PHOTOGRAPH THIS SHEET

DTIC ACCESSION NUMBER

LEVEL

INVENTORY

AD-A148349

DOCUMENT IDENTIFICATION

This document has been approved for public release and sale; its distribution is unlimited.

DISTRIBUTION STATEMENT

ACCESSION FOR		
NTIS	GRA&I	☐
DTIC	TAB	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION /		
AVAILABILITY CODES		
DIST	AVAIL AND/OR SPECIAL	

DISTRIBUTION STAMP

DATE ACCESSIONED

DATE RETURNED

85 09 17 164

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NO.

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDAC

DTIC FORM 70A DEC 83

DOCUMENT PROCESSING SHEET

PREVIOUS EDITION MAY BE USED UNTIL STOCK IS EXHAUSTED.

SCIENCE APPLICATIONS, INC. Hugh W Jones

AD-A148349

FINAL DESIGN REPORT
FOR THE
STUDY ENTITLED
"COSAGE ANALYSIS AND DESIGN REPORT"
VOLUME II -
(VAX COSAGE SOURCE CODE PROCESSED WITH SAI-SDDL)

Contract expiration date:
April 29, 1984

Prepared for:
U.S. Army - Concepts Analysis Agency
Bethesda MD 20014
Mr. Hugh Jones

Prepared by:
Science Applications, Inc.
La Jolla, CA 92038
Mr. Donald A. Heimburger
Ms. Marcia A. Metcalfe
Ms. Suellen S. Worrells
Ms. Diane K. Graham

SAI

VAX COSAGE SOURCE CODE PROCESSED WITH SAI-SDDL

This volume contains the COSAGE SIMSCRIPT source code as it is implemented on the VAX computer. This source code has been processed with SAI-SDDL . This tool provides the formatting and document summary information automatically.

COSAGE PROCESSED SOURCE CODE
VAX VERSION
RUN: 44

APRIL 11, 1984

SCIENCE APPLICATIONS, INC.
LA JOLLA, CA
DONALD A. HEIMBURGER
MARCIA A. METCALFE
SUELLEN S. WORRELLS

```
PAGE  LINE                                   TABLE OF CONTENTS                                                    PAGE I
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

   0    21   TITLE COSAGE DATA STRUCTURE
   1    37     ROUTINE FOR CROSS_REFERENCING
   2    62     PROGRAM REVISIONS
   5   218     ''PROGRAM'' PREAMBLE ''                                  DATA STRUCTURE FOR COSAGE (ORIGINALLY 2865 LINES) \1 D002
   5   250       ''SECTION FOR PERMANENT_ENTITIES
  19  1084       ''SECTION FOR TEMPORARY_ENTITIES
  33  1887       ''SECTION FOR PROCESSES
  39  2199       ''SECTION FOR EVENTS
  41  2350       ''SECTION FOR DEFINITIONS
  50  2861       ''SECTION FOR SUBSTITUTIONS

  54  3018   TITLE TOP-LEVEL ROUTINES
  55  3023     ''PROGRAM'' MAIN                                                                                T001
  56  3036     ROUTINE MAIN1                                                                                   T002
  57  3069     ROUTINE MAIN3                                                                                   T003
  58  3089     ROUTINE CREATE.FORCE                                                                            T004
  60  3157     ROUTINE CREATE.TEAMS                                                                            T005
  61  3184     ROUTINE FEBA.INITIAL  ''                                                                        T006
  63  3260     ROUTINE FILE.FD.SCHD  GIVEN FDC, NEW.START, NEW.STOP, AND NEW.BATS ''                           T007
  65  3358     ROUTINE FILE.KAD.SENSOR                                                                         T008
  66  3416     ROUTINE FORM.TF.LIST(UNT.,UNIT.LIST.,.ERROR) ''                                                 T009
  67  3448     ROUTINE GENERAL.BATTLE                                                                          T010
  71  3634     ROUTINE ORIENTATION GIVEN  ''                                                %15MAR79_%JN \T011
  75  3823     ROUTINE UNIT.ASSIGNMENT                                                                         T012

  76  3874   TITLE MOVEMENT/TERRAIN ROUTINES
  77  3879     ROUTINE ADJUST                                                                                  M001
  80  4005     ROUTINE BLOCK.LOS                                                                               M002
  83  4124     ROUTINE CHANGE.LOC                                                                              M003
  87  4307     ROUTINE END.MOVE                                                                                M004
  89  4395     ROUTINE FA.BN.MOVEMENT                                                                          M005
  93  4587     ROUTINE INITIAL.DETECT                                                                          M006
  94  4613     ROUTINE INITIAL.MOVE                                                                            M007
  95  4637     ROUTINE INIT.REINF GIVEN UNT.CALLING.,UNT.RESPONDING.,ORDR.,  ''                                M008
  96  4663     ROUTINE LINE.OF.SIGHT                                                                           M009
  98  4770     ROUTINE LOCATE.SEARCH.AREA                                                                      M010
 100  4831     ROUTINE LOCATE.SECTOR                                                                           M011
 101  4871     ROUTINE LOS.CHECK ''                                                                            M012
 103  4932     ROUTINE MINE.DELAY                                                                              M013
 106  5060     ROUTINE MIN.MOVE                                                                                M014
 107  5102     ROUTINE NEW.SEGMENT                                                                             M015
 110  5264     ROUTINE POSITION                                                                                M016
 111  5299     ROUTINE PRED.POS                                                                                M017
 112  5346     ROUTINE PREPARE.LIST                                                                            M018
 114  5462     ROUTINE PREP.WITHDRAW                                                                           M019
 115  5496     ROUTINE PROX.CHECK                                                                              M020
 117  5563     ROUTINE PROX.POS                                                M021
 118  5597     ROUTINE REIN.ARRIVE                                                                             M022
 119  5647     ROUTINE RESET.FEBA.SECTOR                                                                       M023
 120  5705     ROUTINE SEARCH                                                  M024
 122  5773     ROUTINE SEGMENT.ADJUST                                                                          M025
 123  5794     ROUTINE TIME.TO.DETECT                                                                          M026
 125  5873     ROUTINE WHAT.NEXT                                                                               M027

 127  5979   TITLE DIRECT FIRE ROUTINES
 128  5984     ROUTINE BTL.CHECK                                                                               DF01
 130  6067     ROUTINE CHECK.DEAD(UNITS.)  ''                                                                  DF02
```

132	6131	ROUTINE CHECK.ENGAGEMENT ''	DF03
133	6153	ROUTINE CHECK.FORCE ''	DF04
134	6211	ROUTINE CHECK.FOR.MINES ''	DF05
137	6332	ROUTINE CHECK.LIST(UNT.,NO.TF,DUM.UNITS.) YIELDING RESULT ''	%30AUG_%RGR \DF06
138	6373	ROUTINE CHECK.PROX GIVEN UNT.,UNT.SECTOR.,CHK.RANGE. ''	DF07
141	6503	ROUTINE CHECK.STREN(UNT.) YIELDING PCT.STREN ''	DF08
142	6521	ROUTINE DEAD.UNIT ''	DF09
145	6692	ROUTINE FIN.BATTLE ''	DF10
146	6739	ROUTINE INTER.BATTLE GIVEN UNT,BATTLE,ORDR. ''	DF11
149	6893	ROUTINE PK.COMPUTE ''	DF12
153	7069	TITLE INDIRECT FIRE ROUTINES	
154	7074	ROUTINE AO.DETECTION ''	IF01
159	7311	ROUTINE ATTRIT.SENSOR ''	IF02
163	7513	ROUTINE BTRY.FM.DEQ GIVEN MISSION ''	IF03
164	7543	ROUTINE BTRY.FM.ENQ ''	IF04
165	7573	ROUTINE CFR.DEGRADE ''	IF05
166	7617	ROUTINE CFR.DETECTION ''	IF06
168	7699	ROUTINE CHK.COMP.TR ''	IF07
169	7722	ROUTINE CHK.FD.TR ''	IF08
170	7764	ROUTINE COMBINE.TRS ''	IF09
171	7797	ROUTINE COMPARE.TRS ''	IF10
172	7843	ROUTINE COPY ''	IF11
173	7882	ROUTINE DUST.EFFECTS ''	IF12
176	8004	ROUTINE EST.COVERAGE GIVEN ''	IF13
180	8219	ROUTINE EST.MIL.WORTH ''	IF14
181	8242	ROUTINE FASCAM.COMPUTATION ''	IF15
182	8286	ROUTINE FA.BN.ASGN ''	IF16
187	8521	ROUTINE FD.EFFECTS.REQ ''	IF17
188	8547	ROUTINE FINAL.COVERAGE GIVEN TARGET,MISSION,ENVIR,POST,TE,FUZE ''	IF18
194	8891	ROUTINE FIND.START.TIME ''	IF19
196	8989	ROUTINE HE.OR.ICM.COMPUTATION ''	IF20
199	9133	ROUTINE ILLUM.COMPUTATION ''	IF21
200	9181	ROUTINE ILLUM.EFFECTS ''	IF22
203	9300	ROUTINE MARGINAL.EFFECTS.ADJ GIVEN MISSION, ''	IF23
205	9385	ROUTINE NOISE.DEGRADE GIVEN PDB YIELDING DET.PROB ''	IF24
206	9428	ROUTINE PDB.DETECTION ''	IF25
208	9489	ROUTINE PGM.MSN.ASGN ''	IF26
212	9701	ROUTINE PIR.DETECTION ''	IF27
214	9791	ROUTINE REM.EFFECTS.COMPUTATION ''	IF28
215	9828	ROUTINE REQUEST.DEF.FASCAM ''	IF29
216	9885	ROUTINE REQUEST.FASCAM ''	IF30
218	9981	ROUTINE REQUEST.ILLUM ''	IF31
222	172	ROUTINE REQUEST.SMOKE ''	IF32
226	391	ROUTINE REQUEST.WD.FASCAM ''	IF33
229	512	ROUTINE RPV.DETECTION ''	IF34
231	623	ROUTINE SIZE.ESTIMATE ''	IF35
233	687	ROUTINE SMOKE.COMPUTATION ''	IF36
234	738	ROUTINE SMOKE.EFFECTS ''	IF37
238	913	ROUTINE SWITCH.FO ''	IF38
239	971	ROUTINE TARGET.ANALYSIS ''	IF39
242	1111	ROUTINE UNIT.ENVIR ''	IF40
246	1307	ROUTINE VOLLEY ''	IF41
248	1366	ROUTINE WEIGHTED.VOLLEYS ''	IF42
250	1459	ROUTINE MINE.EFFECTS ''	IF43
254	1671	ROUTINE FO.DETECTION ''	IF44
260	1964	ROUTINE BTRY.EFFECTS ''	IF45

274	2733	ROUTINE CLEAN.UP.FIRE.MISSIONS ''	IF46
275	2782	TITLE AIR ROUTINES	
276	2787	ROUTINE AC.BOMB.EFFECTS ''	A001
280	2996	ROUTINE AC.DF.EFFECTS ''	A002
283	3167	ROUTINE CAS.EVAL ''	A003
285	3280	ROUTINE CHECK.CAS.CONSTRAINTS ''	A004
289	3473	ROUTINE EMPLOY.HELICOPTERS ''	A005
293	3676	ROUTINE END.CAS.MISSION ''	A006
297	3856	ROUTINE FARRP.CHECK ''	A007
298	3913	ROUTINE HC.COMPUTE.TIMES ''	A008
300	4003	ROUTINE HC.DISENGAGE ''	A009
302	4119	ROUTINE HEL.RANGE.COMPUTE ''	A010
304	4187	ROUTINE REPLACE.HC ''	A011
305	4234	ROUTINE UNIT.PRIORITY ''	A012
307	4297	ROUTINE AD.SHOOT ''	A013
311	4512	ROUTINE INTER.HELO ''	A014
314	4630	ROUTINE FLIGHT.PATH ''	A015
320	4937	TITLE COMBAT SUPPORT ROUTINES	
321	4942	ROUTINE COMPUTE.D ''	C001
322	4964	ROUTINE COMPUTE.WD ''	C002
323	4998	ROUTINE CONTRAST.TO.FREQ ''	C003
324	5034	ROUTINE DECIDE ''	C004
325	5078	ROUTINE DEQ.FEBA.SET ''	C005
326	5118	ROUTINE DESTROY.ORD(UNT.) ''	C006
327	5143	ROUTINE DQ.CMSN.QUEUE ''	C007
328	5165	ROUTINE EMPTY ''	C008
331	5301	ROUTINE ENQ.FEBA.SET GIVEN ''	C009
332	5341	ROUTINE FDC.TR.DEQ ''	C010
333	5357	ROUTINE FDC.TR.ENQ ''	C011
334	5379	ROUTINE FINISH.COMPUTATION ''	C012
335	5413	ROUTINE FRAC.COMPUTE GIVEN A YIELDING FRAC ''	C013
336	5434	ROUTINE GET.TERRAIN ''	C014
337	5457	ROUTINE HC.EMPTY ''	C015
340	5600	ROUTINE PROB.INF ''	C016
341	5634	ROUTINE PROB.TIME ''	C017
342	5662	ROUTINE RANGE.COMPUTE ''	C018
343	5676	ROUTINE SEARCH.COVERAGE ''	C019
344	5732	ROUTINE TEMPERATURE.ATTENUATION ''	C020
345	5785	ROUTINE TERM.CHECK ''	C021
346	5798	TITLE EVENTS	
347	5803	EVENT ACT.REINF ''	E001
349	5897	EVENT AD.ENGAGEMENT ''	E002
355	6227	EVENT ARTY.OCCUPATION ''	E003
356	6253	EVENT BTL.ENDED GIVEN WINNER.,BL.UNITS.,RD.UNITS. ''	E004
358	6341	EVENT CFR.ACTIVATION ''	E005
359	6380	EVENT CFR.OFF ''	E006
360	6405	EVENT CFR.ON ''	E007
362	6477	EVENT CFR.OPERATOR ''	E008
365	6595	EVENT CHANGE.LITE ''	E009
366	6616	EVENT CHANGE.WEATHER ''	E010
367	6634	EVENT DQ.OLD.SORTIE.QUEUE ''	E011
368	6661	EVENT END.SIMULATION ''	E012
369	6678	EVENT ENGAGEMENT ''	E013
371	6775	EVENT FEBA.SORTIE ''	E014

373	6875	EVENT GET.NX.ORD ..	E015
375	6988	EVENT HC.DEPART.BATTLE ..	E016
376	7030	EVENT HELO.ENGAGEMENT ..	E017
380	7229	EVENT INIT.PREPLAN.CAS ..	E018
381	7267	EVENT MOVE ..	E019
382	7280	EVENT OFF.LINE.ATTRITION ..	E020
386	7467	EVENT PDB.ACTIVATION ..	E021
387	7504	EVENT PDB.OPERATOR ..	E022
389	7568	EVENT POSITION.REPORT ..	E023
390	7579	EVENT SCHEDULE.ARTY.MOVEMENT ..	E024
391	7596	EVENT SEND.TEAM ..	E025
392	7623	EVENT SET.DEBUG ..	E026
393	7655	EVENT START.ARTY.MOVEMENT ..	E027
395	7743	EVENT START.BATTLE ..	E028
403	8177	EVENT START.MOVE ..	E029
407	8370	EVENT STOP.ARTY.MOVEMENT ..	E030
408	8404	EVENT UPDATE.LOC ..	E031
412	8599	EVENT ACT.ATK ..	E032
414	8714	EVENT ACT.DEF ..	E033
415	8736	EVENT ACT.MOVCOR ..	E034
416	8766	EVENT ACT.MOVDIS ..	E035
417	8803	EVENT DYNAMIC.ANALYSIS.REPORT SAVING THE EVENT NOTICE ..	E036
418	8842	TITLE PROCESSES	
419	8847	PROCESS AC.ATK.TGT ..	P001
428	9330	PROCESS AIR.OBSERVER ..	P003
435	9686	PROCESS ARTY.ASSESS ..	P004
438	9854	PROCESS FORWARD.OBSERVER ..	P005
443	93	PROCESS HC.ARRIVE.BATTLE ..	P006
450	460	PROCESS HC.RETURN.FARRP ..	P007
454	692	PROCESS HEL.TARGET.ACQUISITION ..	P008
463	1168	PROCESS HOW.REPAIR ..	P009
464	1216	PROCESS MINE.ASSESS ..	P010
467	1356	PROCESS REMOTE.PILOT.VEHICLE ..	P012
470	1482	PROCESS TARGET.REPORT ..	P013
478	1914	PROCESS WITH.DRAW ..	P014
480	2027	PROCESS FIRE.MISSION ..	P015
487	2411	PROCESS ASSESSMENT ..	P016
493	2749	PROCESS SHOOT.OUT ..	P017
504	3350	PROCESS CAS.MISSION ..	P018
510	3658	PROCESS HELICOPTER.FIRE ..	P019
519	4178	TITLE INPUT ROUTINES	
520	4183	ROUTINE MAIN2 ..	I001
523	4317	ROUTINE SYS.INPUT ..	I002
525	4381	ROUTINE PK.INPUT ..	I003
527	4450	ROUTINE CAT.TU.INPUT ..	I004
529	4523	ROUTINE KV.INPUT ..	I005
530	4573	ROUTINE EQ.TE.INPUT ..	I006
532	4638	ROUTINE TYPE.WEAPON.INPUT ..	I007
533	4680	ROUTINE UNIT.INPUT ..	I008
538	4930	ROUTINE MFO.INPUT ..	I009
539	4983	ROUTINE READ.ORDERS ..	I010
542	5114	ROUTINE ORD.DEF ..	I011
543	5139	ROUTINE ORD.ATK ..	I012
544	5158	ROUTINE ORD.REINF ..	I013
545	5179	ROUTINE ORD.MOVDIS ..	I014

546 5203 ROUTINE ORD.MOVCOR '' I015
547 5255 ROUTINE P.E.M.INPUT '' I016
548 5307 ROUTINE TB.INPUT '' I017
550 5387 ROUTINE BTRY.INPUT '' I018
553 5539 ROUTINE FBN.FD.INPUT '' I019
555 5602 ROUTINE MUNS.INPUT '' I020
557 5713 ROUTINE SUBM.INPUT '' I021
558 5750 ROUTINE HE.LA.INPUT '' I022
560 5820 ROUTINE RUL.EN.INPUT '' I023
561 5855 ROUTINE ST.INPUT '' I024
562 5878 ROUTINE MCFR.INPUT '' I025
563 5915 ROUTINE MPDB.INPUT '' I026
564 5953 ROUTINE MAO.INPUT '' I027
565 6011 ROUTINE SENSOR.INPUT '' I028
569 6202 ROUTINE TBF.INPUT '' I029
570 6260 ROUTINE DECISION.INPUT '' I030
571 6282 ROUTINE TT.FACTORS.INPUT '' I031
572 6327 ROUTINE FARRP.INPUT '' I032
575 6449 ROUTINE PGM.INPUT '' I033
576 6494 ROUTINE ILLUM.INPUT '' I034
577 6533 ROUTINE MINE.INPUT '' I035
580 6668 ROUTINE SMOKE.INPUT '' I036
581 6711 ROUTINE VIS.INPUT '' I037
582 6736 ROUTINE TACAIR.INPUT '' I038
587 6978 ROUTINE MADS.INPUT '' I039
588 7011 ROUTINE AC.MUNS.INPUT '' I040
590 7073 ROUTINE TR.INPUT '' I041

591 7114 TITLE OUTPUT ROUTINES
592 7119 ROUTINE AMMO.RPT '' O001
595 7239 ROUTINE ANALYSIS.OUTPUT '' O002
599 7428 ROUTINE BETWEEN.ROUTINE '' O003
604 7665 ROUTINE ERROR.STOP '' O004
605 7679 ROUTINE HEADING ''O005
606 7688 ROUTINE KV.PRINT GIVEN '' O006
609 7842 ROUTINE KV.SCOREBOARD '' O007
611 7930 ROUTINE OUTPUT.ATTRITION GIVEN UNIT,KILLER,VICTIM,N.KILLED,FIRE,MSN '' O008
613 7993 ROUTINE FOR POSITION.OUT '' O009
614 8020 ROUTINE SNAP2 '' O010
618 8201 ROUTINE SNAP.R '' O011
621 8333 ROUTINE TACAIR.DATA.REPORT '' O012
624 8489 ROUTINE OUTPUT.EXPENDITURES '' O013

627 8625 TITLE FUNCTIONS
628 8630 FUNCTION ACT.RANGE '' F001
629 8645 FUNCTION BTRY.AVAILABLE(BTRY, TARGET, DURATION) '' F003
630 8679 FUNCTION COLLISION(.UNIT, .ORDER) '' F004
631 8714 FUNCTION COMBINATIONS(N , M) '' F005
632 8744 FUNCTION EST.RANGE(TARGET , UNIT) '' F006
633 8761 FUNCTION EST.TR.RANGE(TR1 , TR2) '' F007
634 8774 FUNCTION FEBA.BAND(TARGET) '' F008
636 8859 FUNCTION HE.WLA(TARGET,BTRY,MUNITION,RANGE,BAT.NUM,FM) '' F009
638 8957 FUNCTION ICM.WLA(TARGET,BTRY,MUNITION,RANGE) '' F010
640 9026 ROUTINE EXPONENTIAL.F(MU,STREAM) '' (SHOULD BE FUNCTION) \CLEANUP F012
641 9039 ROUTINE NORMAL.F(MU, SIGMA, STREAM)'' (SHOULD BE FUNCTION) \CLEANUP F013
642 9062 ROUTINE WEIBULL.F (SHAPE,SCALE,STREAM) '' (SHOULD BE FUNCTION) \CLEANUP F014

643	9073	TITLE SUPPORT AND COMPUTATION ROUTINES	
644	9078	ROUTINE ANGLE.COMPUTE ''	S001
645	9116	ROUTINE LINE.CIRCLE ''	S002
646	9155	ROUTINE MRT.TO.FREQ ''	S003
647	9180	ROUTINE OPEN.INPUT.OUTPUT.FILES ''	S004
648	9192	ROUTINE PERFORM.INSTRUMENTATION '' \DYN_ANAL	S005
649	9214	''PROGRAM LIB$INIT_TIMER	
650	9221	''PROGRAM LIB$STAT_TIMER(FLAG, CPU.TIME)	
651	9231	TITLE UN-USED & DELETION CANDIDATE MODULES	
652	9236	ROUTINE GAMMA.F(MEAN, K, STREAM) '' (SHOULD BE FUNCTION) \CLEANUP \REQUIRED?	U001
653	9261	PROCESS AIRBORNE.RADAR ''	U002
656	9385	ROUTINE AR.DETECTION GIVEN AR,CANDIDATE,MODEL,TARGET,SENSOR.TYPE ''	U003
657	9443	FUNCTION AR.PROB.DETECT(RANGE,MODEL) ''	U004
658	9474	PROCESS PHOTO.IR.FLIGHT '' \UNNECESSARY?	U005
660	9590	FUNCTION STAY.TIME ''	U006
661	9615	ROUTINE JOHNSON.CRITERIA ''	U007
662	9626	ROUTINE PROXIMITY.REQ '' \REQUIRED?	U008
663	9640	ROUTINE TIME.REQ '' \REQUIRED?	U009
664	9655	''PROGRAM OLDER.VERSION'' PREAMBLE '' DATA STRUCTURE FOR COSAGE (ORIGINALLY 2848 LINES) \1	U010
714	2505	ROUTINE PLAT.COUNT '' \UNNECESSARY?	U011
715		MODULE INVOCATION TREE	
730		CROSS REFERENCE — MODULE	
763		CROSS REFERENCE — VARIABLES, SETS, AND ENTITIES	
172		CROSS REFERENCE — UPDATES MADE BY PREVIOUS PERSONNEL	
177		CROSS REFERENCE — MODIFICATIONS MADE TO PROCESS SOURCE CODE & NOTES FOR LATER OPTIMIZATION	
192		CROSS REFERENCE — COMPUTER MEMORY ALLOCATION AND DEALLOCATION KEYWORDS	

```
37 ROUTINE FOR CROSS_REFERENCING
38
39    THIS MODULE IS INCLUDED TO FACILITATE THE CROSS REFERENCING OF ONE-WORD
40    VARIABLES, SETS, ENTITIES, AND MEMORY ALLOCATION/DEALLOCATION KEYWORDS:
41
42    "CREATE"            "DESTROY"           "RELEASE"           "RESERVE"
43    "USE"               "ALPHA"             "TEXT"              "EXTENDED"
44    $ENVIRONMENT$       $POSTURE$           $BTRY$              $CATEGORY$
45    $MISSION$           $SUBMUNITION$       $EQUIPMENT$         $FDC$
46    $FUZE$              $GROUPING$          $MINEFIELD$         $COLOR$
47    $DECISION$          $SECTOR$            $SIDE$              $UNIT$
48    $BATTLE$            $FARRP$             $FORCE$             $HELICOPTER$
49    $HOW$               $ORDER$             $POINT$             $SEGMENT$
50    $WEAPON$            $ASSESSMENT$        $ATKUNIT$           $ATKORDER$
51    $ATKENEMY$          $DEFUNIT$           $DEFORDER$          $MCUNIT$
52    $MCORDER$           $MDUNIT$            $MDORDER$           $REINUNIT$
53    $REINORDER$         $ENGAGEMENT$        $MOVE$              $TERRAIN$
54    $DEBUG$             $VISIBILITY$        $BMNT$              $EENT$
55    $SHADE$             $ANALYSIS$          $COLLISION$         $COMBINATIONS$
56    $NORTH$             $SOUTH$
57
59 END
```

PROGRAM REVISIONS

```
*****************************************************************************************************
*     The purpose of this module is to outline the modifications made to COSAGE in order to execute on the VAX computer *
* with a particular set of input test data.  The modifications were each the result of a particular test run which     *
* yielded a SIMSCRIPT execution error.  Each revision has been carefully documented and cross-referenced not only by   *
* specific change but also by class of problem.                                                                       *
*****************************************************************************************************
```

CHANGE NUMBER	DATE	TIME.V	PROGRAM MODULE	SIMSCRIPT ERROR	PROGRAM CHANGE
CHG\01	12/18/83	0.000	MOST PROCESSES	ZERO POINTER ERROR 2058	REPLACED LOCAL VARIABLE L.1 WITH GLOBAL VARIABLE PROCESS.V
CHG\02	01/01/84	0.050	FA.BN MOVEMENT	ZERO POINTER ERROR 2058	CHECK FOR ZERO SUBSCRIPT (MIN.BAT & MAX.DAT) PRIOR TO USE
CHG\03	01/02/84	4.130	SWITCH.FO	ZERO POINTER ERROR 2058	CHECK FOR ZERO SUBSCRIPT (FO.CURRENT.TR(FO)) AND PROVIDE ALTERNATE PATH
CHG\04	01/03/84	4.132	SHOOT.OUT	ZERO POINTER ERROR 2058	CHECK FOR ZERO SUBSCRIPT (FIRING.TABLE) PRIOR TO USE
CHG\05	01/03/84	4.134	BTL.ENDED CALLING EMPTY	NUMBER OF ARGUMENTS ERROR 2026	ADDED ARGUMENTS WINNER. AND YES TO CALL TO EMPTY
CHG\06	01/06/84	4.164	FA.BN.ASGN	SUBSCRIPT OUT-OF-RANGE ERROR 2025	WHEN TR.ES.TU = 0, SET TU = 1
CHG\07	01/08/84	4.198	FIRE.MISSION	REFERENCE TO DESTROYED ENTITY ERROR 2067	COMMENT OUT REDUNDANT DESTROY STATEMENT
CHG\08	01/09/84	4.273	FO.DETECTION	SUBSCRIPT OUT-OF-RANGE ERROR 2025	RESEQUENCED ASSIGNMENT (MODEL) AND USE AS A SUBSCRIPT
CHG\09	01/09/84	4.319	EST.COVERAGE	ZERO POINTER ERROR 2058	CORRECTED SPELLING OF TR.EST.RAD TO TR.EST.RADIUS
CHG\10	01/09/84	4.424	FINAL.COVERAGE	STOP ENCOUNTERED	LET FRAC.COVERAGE = 1.0 AND COMMENTED OUT TRACE AND STOP - PROBABLY A PRECISION PROBLEM
CHG\11	01/09/84	4.424	BTRY.EFFECTS OPEN.IO.FILES MAIN1	OUTPUT REQUESTED ON INPUT DEVICE ERROR 2012	SET UNIT NUMBER TO 1 FOR INPUT - FIXED HEADING.V
CHG\12	01/09/84	4.546	FIRE.MISSION	REFERENCE TO DESTROYED ENTITY ERROR 2067	STORE ZERO IN ENTITY POINTER (FM.EX.FIRE.MISSION)
CHG\13	01/10/84	4.533	HE.OR.ICM.COMPUTATION	SUBSCRIPTS OUT-OF-RANGE ERROR 2025	SWITCHED MUNITION ORDER FOR BAT.TYPE = 28 HE IN DATA CHANGED TO ICM MAY STILL BE LOGIC PROBLEM

CHANGE NUMBER	DATE	TIME.V	PROGRAM MODULE	SIMSCRIPT ERROR	PROGRAM CHANGE
CHG\14	01/13/84	4.610	REQUEST.ILLUM	ZERO POINTER ERROR 2058	CHANGE SPELLING OF UN.SENSOR.TYPE TO US.SENSOR.TYPE
CHG\15	01/14/84	4.611	SHOOT.OUT	ZERO POINTER ERROR 2058	CHECK FOR ZERO SUBSCRIPT (FIRING.TABLE)
CHG\16	01/14/84	4.841	FA.BN.MOVEMENT	ZERO POINTER ERROR 2058	CHANGED IF TEST TO EXCLUSIVE GREATER THAN (BY.CUR.FM(.BTRY))
CHG\17	01/17/84	4.875	SHOOT.OUT TACAIR.INPUT	REFERENCE TO DESTROYED ENTITY ERROR 2067	TURNED OFF AIR, HELICOPTERS
CHG\18	01/22/84	4.641	REQUEST.ILLUM	REFERENCE TO DESTROYED ENTITY ERROR 2067	ZEROED POINTER ATTRIBUTES WHEN ENTITY IS DESTROYED
CHG\19	01/31/84	4.913	PGM.MSN.ASGN FA.BN.ASGN CLEAN.UP.FIRE.MISSIONS	REFERENCE TO DESTROYED ENTITY ERROR 2067	MADE SUBROUTINE TO CHECK FOR ZERO AND TRACE; CLEAN UP SETS FOR FIRE.MISSIONS
CHG\20	01/31/84	5.049	REQUEST.ILLUM	REFERENCE TO DESTROYED ENTITY ERROR 2067	TURNED OFF ILLUMINATION; PROBLEM STILL EXISTS
CHG\21	02/01/84	5.166	TARGET.REPORT TARGET.ANALYSIS	ATTEMPT TO DESTROY ENTITY IN SET ERROR 2226	CHECK FOR ZERO (TR.EST.TU OR TR.SENSOR.ID); ADDED FLAG ARGUMENT TO CALL TO TARGET.ANALYSIS
CHG\22	02/01/84	5.608	REQUEST.SMOKE	REFERENCE TO DESTROYED ENTITY ERROR 2067	TURNED OFF SMOKE; PROBLEM STILL EXISTS
CHG\23	02/01/84	7.276	REQUEST.FASCAM	INVALID ARRAY OR ENTITY POINTER ERROR 2066	TURNED OFF MINES; PROBLEM STILL EXISTS
CHG\24	02/01/84	6.363	DUST.EFFECTS	INCORRECT NUMBER OF ARGUMENTS ERROR 2026	DECLARE .XIMPACT AND .YIMPACT REAL
CHG\25	02/02/84	8.165	FO.DETECTION	ZERO POINTER ERROR 2058	CORRECT SPELLING OF BY.PGM.STATUS TO BY.PGM.CAP
CHG\26	02/03/84	9.879	ASSESSMENT	ZERO POINTER UNRESERVED ARRAY ERROR 2058	SUBSTITUTE N.SO.LIST() FOR UE.QUANT()
CHG\27	02/03/84	5.352	EST.COVERAGE	DIVIDE BY ZERO ERROR 2191	CHECK FOR ZERO (REP.TM) AND RETURN WITH FRACT.COVERAGE = 1.0
CHG\28	02/04/84	9.000	ATTRIT.SENSOR	ZERO POINTER ERROR 2058	CHECK FOR ZERO VALUE OF POINTER FO.CURRENT.TR(SENSOR)

CHANGE NUMBER	DATE	TIME.V	PROGRAM MODULE	SIMSCRIPT ERROR	PROGRAM CHANGE
CHG\29	02/06/84	9.687	TARGET.ANALYSIS	SUBSCRIPT OUT-OF-RANGE ERROR 2025	ADDED CODE TO INSURE NON-ZERO VALUE (FEBA)
CHG\30	02/09/84	9.810	FEBA.BAND	SUBSCRIPT OUT-OF-RANGE ERROR 2025	ADD CHECK FOR EMPTY SET LET DFFB=1 AND RETURN LOCATE.SECTOR SHOULD BE CORRECTED
CHG\31	02/12/84	24.000	AMMO.RPT KV.SCOREBOARD	ZERO POINTER ERROR 2058	COMMENTED OUT CALL TO AMMO.RPT

* At this point, SAI analysts had achieved the first 24 hour simulation. It required 4 hours and 40 minutes of VAX 11/780 CPU time. The next revision was necessary when the random number stream was changed from 3 to 6.

CHG\32	03/14/84	9.872	CFR.ON	SUBSCRIPT OUT-OF-RANGE ERROR 2025	ADD CHECK FOR EMPTY SET LET DFFB=1 AND RETURN LOCATE.SECTOR SHOULD BE CORRECTED

* At this point, SAI analysts had achieved a 24 hour simulation with random number steam 6. It required 3 hours and 49 minutes of CPU time. The next revision was necessary when the random number stream was changed from 6 to 10.

CHG\33	03/29/84	8.799	COMBINE.TRS	DIVIDE BY ZERO ERROR 2191	ADDED CHECK FOR TR.CEP LE ZERO IN TARGET.REPORT

* At this point, SAI analysts had achieved a 24 hour simulation with random number stream 10. It required 4 hours and 44 minutes of CPU time.

END

```
''PROGRAM'' PREAMBLE ''                                    DATA STRUCTURE FOR COSAGE (ORIGINALLY 2865 LINES) \1 D002

  NORMALLY MODE IS INTEGER
  DEFINE HELICOPTER TO MEAN HELI.COPTER
  DEFINE TR.START.TIME TO MEAN TR.RECVD.TIME

  GENERATE LIST ROUTINES

  ********************************************************************************************************************
  *     ''                                                                                                           *
  *     ''              COSAGE I   PROGRAMMERS                                                                       *
  *     ''                                                                                                           *
  *     ''    MR   GLENN    R. STOCKTON    APR 77 - AUG 78                                                          *
  *     ''    MAJ  ROBERT   J. HILL  III   APR 77 - OCT 78                                                          *
  *     ''    MAJ  RICHARD     PORTER      APR 77 - JUL 78                                                          *
  *     ''    MAJ  ROBERT   T. CAMPBELL    APR 77 - DEC 78                                                          *
  *     ''    MR   GARY     L. MARTIN      APR 77 - JUN 79      GLM                                                 *
  *     ''    MR   JOE      E. NICHOLS     OCT 78 -             JN & JEN                                            *
  *     ''    LTC  EARL        DARDEN      SEP 77 - JUL 80      ED                                                  *
  *     ''    LTC  JOHN     L. RAFFERTY    JUL 78 - OCT 80      JLR                                                 *
  *     ''    MAJ  RONALD   G. RHOADES     DEC 78 - APR 80      RGR                                                 *
  *     ''    MS   PAMELA   J. KENNEDY     JUN 79 - AUG 79      PJK                                                 *
  *     ''    MR   ANDREW   J. CARRAS      OCT 79 - JUN 80      AJK                                                 *
  *     ''    MR   HUGH     W. JONES       JAN 80 -             HWJ                                                 *
  *     ''    MAJ  ROBERT   W. FEISZLI     APR 80 -             RWF                                                 *
  *     ''    CPT  JAMES    S. KRAMER      JUL 80 -             JSK                                                 *
  *     ''    MR   DANIEL   J. SHEDLOWSKI  SEP 80 -             DJS                                                 *
  *     ''    CPT  ANDREW   G. LOERCH      OCT 80 -             AGL                                                 *
  *     ''    MAJ  FRANK    T. PROCTOR JR. OCT 80 -             FTP                                                 *
  *     ''                                                                                                           *
  ********************************************************************************************************************

''SECTION FOR PERMANENT_ENTITIES
  PERMANENT ENTITIES
      INCLUDE TB.N.FM,
          NITE.OR.DAY,
          MOVEMENT.STATUS

  EVERY AC.MUNS HAS
      AN AM.NAME,
      AN AM.RELY,
      AN AM.RADIUS

  EVERY AC.MUNS, ENVIRONMENT, POSTURE HAS
      AN AMEP.LA.PERS

  EVERY AC.MUNS, ENVIRONMENT, TYPE.EQUIPMENT HAS
      AN AMET.LA.EQUIP

  EVERY AC.TYPE HAS
      AN ACT.EQUIP.ID,
      AN ACT.SUBSTITUTE,
      AN ACT.WEATHER.DEGRADE,
      AN ACT.MIN.ALT,
      AN ACT.NORM.ALT,
      AN ACT.BAI.TA.DELAY, ''MINUTES
      AN ACT.SPEED,  ''HDM/SEC
      AN ACT.PASS.TIME, ''MINUTES
```

```
276         AN ACT.MAX.ALOFT, ''MINUTES
277         AN ACT.PROB.SORTIE.ABORT,
278         AN ACT.NITE.FLY,
279         AN ACT.MIN.PREP.TIME, ''MINUTES
280         AN ACT.MAX.PREP.TIME, ''MINUTES
281         AN ACT.P1.DIST,
282         AN ACT.P2.DIST,
283         AN ACT.P3.DIST,
284         AN ACT.ANGLE.P1.P2,
285         AN ACT.X1,
286         AN ACT.Y1,
287         AN ACT.Z1,
288         AN ACT.X2,
289         AN ACT.Y2,
290         AN ACT.Z2,
291         AN ACT.X3,
292         AN ACT.Y3,
293         AN ACT.Z3
294
295     EVERY AC.TYPE, AC.MUNS HAS
296         AN ATM.DELIV.CEP
297
298     EVERY AO.ELEVATION.BAND HAS
299         AN AO.EB.ALTITUDE
300     BELONGS TO
301         THE AO.EB.SET
302     HAS
303         A P.AO.EB.SET,
304         A S.AO.EB.SET,
305         A M.AO.EB.SET
306
307     EVERY AO.ELEVATION.BAND, AO.RANGE.BAND  HAS
308         AN AO.PROB.LOS,
309         AN AO.VISIBILITY
310
311     EVERY AO.RANGE.BAND HAS
312         AN AO.RB.RANGE,
313     BELONGS TO
314         THE AO.RB.SET
315     HAS
316         A P.AO.RB.SET,
317         A S.AO.RB.SET,
318         A M.AO.RB.SET
319
320     EVERY BTRY HAS
321         A BY.BN,
322         A BY.STATUS,
323         A BY.TYPE,
324         A BY.PGM.FM,
325         A BY.CUR.FM,
326         A BY.N.ROUNDS,
327         A BY.UNIT,
328         A BY.BN.RANK,
329         A BY.FIRE.RATE,
330         A BY.PGM.CAP,
331         A BY.STOP.FASCAM.SUPP  ''TIME.V * 60
332     OWNS
333         A BY.HOW.SET,
```

```
334         A BY.SCHD.LIST,
335         A BY.FM.QUEUE
336     BELONGS TO
337         A BN.BTRY.SET
338     HAS
339         A F.BY.HOW.SET,
340         A L.BY.HOW.SET,
341         A F.BY.SCHD.LIST,
342         A L.BY.SCHD.LIST,
343         A F.BY.FM.QUEUE,
344         A L.BY.FM.QUEUE,
345         A P.BN.BTRY.SET,
346         A S.BN.BTRY.SET,
347         A M.BN.BTRY.SET,
348         A N.BY.HOW.SET,
349         A N.BY.SCHD.LIST,
350         A N.BY.FM.QUEUE
351
352     EVERY CATEGORY HAS
353         A CT.NAME,
354         A CT.GROUP,
355         A CT.MIN.FEBA
356     OWNS
357         A CT.TU.SET
358     BELONGS TO
359         A GP.CAT.SET
360     HAS
361         A F.CT.TU.SET,
362         A L.CT.TU.SET,
363         A N.CT.TU.SET,
364         A P.GP.CAT.SET,
365         A S.GP.CAT.SET,
366         A M.GP.CAT.SET
367
368     EVERY CATEGORY,DIST.FROM.FEBA.BAND, IC.MUNITION HAS
369          A CDI.USAGE.INDICATOR
370
371     EVERY CATEGORY, DIST.FROM.FEBA.BAND, TYPE.BTRY HAS
372          A CDT.MAX.VOLS
373
374     EVERY CATEGORY , POSTURE , MISSION   HAS
375         A CPM.WARNED.FRACT,
376         A CPM.UNWARNED.FRACT
377
378     EVERY CFR.RNG.HACK HAS
379         A CFR.RH.RANGE,
380         A CFR.DET.PROB,
381         A CFR.CIR.ERROR
382     BELONGS TO
383         A MCFR.RH.LIST
384     HAS
385         A P.MCFR.RH.LIST,
386         A S.MCFR.RH.LIST,
387         A M.MCFR.RH.LIST
388
389     EVERY DIST.FROM.FEBA.BAND HAS
390         A DFFB.MAX.RANGE
391
```

```
392     EVERY DIST.FROM.FEBA.BAND, TYPE.UNIT HAS
393         A DT.MAX.BATS
394
395     EVERY ENVIRONMENT HAS
396         A EN.NAME
397
398     EVERY ENVIRONMENT, CATEGORY  HAS
399               AN EC.FRACT
400
401     EVERY ENVIRONMENT, SUBMUNITION HAS
402         A ES.RELY
403
404     EVERY  ENVIRONMENT , POSTURE , SUBMUNITION HAS
405         A EPS.LA.PERS
406
407     EVERY EQUIPMENT HAS
408         AN EQ.AD.INDICATOR,
409         A EQ.NAME,
410         A EQ.KV.ID,
411         A EQ.TE.PTR,
412         A EQ.MAX.SPEED,
413         A EQ.PERSONNEL.LOAD,
414         A EQUIP.PK.PTR,
415         A EQ.PAX.KILL.RATE
416     BELONGS TO
417         A TE.SET
418     HAS
419         A P.TE.SET,
420         A S.TE.SET,
421         A M.TE.SET
422
423     EVERY FASCAM.MUNITION HAS
424         A FMM.ID, ''TEXT NAME
425         A FMM.MAX.RANGE, ''HDM
426         A FMM.RND.WT ''POUNDS
427
428     EVERY FA.BN HAS
429         A FB.MISSION,
430         A FA.BN.UNIT
431     OWNS
432         A BN.BTRY.SET
433     HAS
434         A F.BN.BTRY.SET,
435         A L.BN.BTRY.SET,
436         A N.BN.BTRY.SET
437
438     EVERY FDC HAS
439         A FD.FDC,
440         A FD.MIN.TIME,
441         A FD.MAX.TIME,
442         A FD.CUR.TR,
443         A FD.N.PROCESSED,
444         A FD.N.LOST,
445         A FD.TOT.THRESHOLD
446     OWNS
447         A FD.TR.QUEUE,
448         A FD.COMPLETE.LIST,
449         A FD.SCHD.LIST,
```

```
450         A FD.BN.LIST
451     HAS
452         A F.FD.TR.QUEUE,
453         A L.FD.TR.QUEUE,
454         A F.FD.COMPLETE.LIST,
455         A L.FD.COMPLETE.LIST,
456         A F.FD.SCHD.LIST,
457         A L.FD.SCHD.LIST,
458         A F.FD.BN.LIST,
459         A L.FD.BN.LIST,
460         A N.FD.TR.QUEUE,
461         A N.FD.COMPLETE.LIST,
462         A N.FD.SCHD.LIST,
463         A N.FD.BN.LIST
464
465     EVERY FO.RANGE.BAND HAS
466         A FO.RB.RANGE,
467         A FO.VISIBILITY,
468         A FO.CIR.ERROR
469     BELONGS TO
470         A MFO.RB.SET
471     HAS
472         A P.MFO.RB.SET,
473         A S.MFO.RB.SET,
474         A M.MFO.RB.SET
475
476     EVERY FUZE HAS
477         A FZ.NAME
478
479     EVERY FUZE , HE.MUNITION HAS
480         A FZ.HE.RELY
481
482     EVERY GROUPING HAS
483         A GP.NAME
484     OWNS
485         A GP.CAT.SET
486     HAS
487         A F.GP.CAT.SET,
488         A L.GP.CAT.SET,
489         A N.GP.CAT.SET
490
491     EVERY HE.MUNITION HAS
492         A HE.ID,
493         A HE.WEIGHT,
494         A HE.COST,
495         A HE.VOLLEY.RAD,
496         A HE.ROUND.RAD,
497         A HE.VOL.DUST.RAD, ''HDM
498         A HE.DUST.DURATION, ''MINUTES
499         A HE.MIN.MARG.EFF ''MINUTES
500
501     EVERY HE.MUNITION, TYPE.BTRY OWNS
502         A HE.TB.RH.LIST
503     HAS
504         A F.HE.TB.RH.LIST,
505         A L.HE.TB.RH.LIST,
506         A N.HE.TB.RH.LIST
507
```

```
508     EVERY HE.RANGE.HACK HAS
509         A HE.RH.RANGE,
510         A HE.RH.TOTAL.CPE,
511         A HE.RH.ROUND.CPE
512     BELONGS TO
513         A HE.TB.RH.LIST
514     HAS
515         A P.HE.TB.RH.LIST,
516         A S.HE.TB.RH.LIST,
517         A M.HE.TB.RH.LIST
518
519     EVERY HE.RANGE.HACK, ENVIRONMENT, POSTURE, FUZE HAS
520               AN REPF.LA.PERS
521
522     EVERY HE.RANGE.HACK, TYPE.EQUIPMENT, ENVIRONMENT, FUZE HAS
523               AN RTEF.LA.EQUIP
524
525     EVERY IC.MUNITION HAS
526         A IC.ID,
527         A IC.WEIGHT,
528         A IC.COST,
529         A IC.RELIABILITY,
530         A IC.SUBM.INDEX,
531         A IC.VOLLEY.RAD,
532         A IC.N.SUBM,
533         A IC.MIN.MARG.EFF
534
535     EVERY IC.MUNITION, TYPE.BTRY HAS
536         A IC.TB.SLOPE,
537         A IC.TB.INTERCEPT
538     OWNS
539         A IC.TB.RH.LIST
540     HAS
541         A F.IC.TB.RH.LIST,
542         A L.IC.TB.RH.LIST,
543         A N.IC.TB.RH.LIST
544
545     EVERY IC.RANGE.HACK HAS
546         A IC.RH.RANGE,
547         A IC.RH.TOTAL.CPE,
548         A IC.RH.ROUND.CPE
549     BELONGS TO
550         A IC.TB.RH.LIST
551     HAS
552         A P.IC.TB.RH.LIST,
553         A S.IC.TB.RH.LIST,
554         A M.IC.TB.RH.LIST
555
556     EVERY ILLUM.MUNITION HAS
557         AN ILLUM.ID,
558         AN ILLUM.RADIUS, ''HDM
559         AN ILLUM.MAX.RANGE, ''HDM
560         AN ILLUM.DURATION, ''MINUTES
561         AN ILLUM.RND.WT ''POUNDS
562
563     EVERY MINEFIELD HAS
564         A MF.ID,
565         A MF.COLOR,
```

```
566         A MF.X.HIGH,
567         A MF.X.LOW,
568         A MF.Y.HIGH,
569         A MF.Y.LOW
570     AND OWNS
571         A  MFP.LIST,
572         AN AU.LIST
573     AND HAS
574         A F.MFP.LIST,
575         A L.MFP.LIST,
576         A N.MFP.LIST,
577         A F.AU.LIST,
578         A L.AU.LIST,
579         A N.AU.LIST
580
581     EVERY MISSION HAS
582         A MN.NAME
583
584     EVERY MISSION,COLOR HAS
585         A DECISION,
586         A SUP.MISSION.PRIORITY,
587         A WD.DIST
588
589     EVERY MODEL.AD.SENSOR HAS
590         A MADS.NAME,
591         A MADS.DELAY.TIME, ''SECONDS
592         A MADS.PW.DEGRADE,
593         A MADS.FCM,
594         A MADS.RIPL, ''SECONDS
595         A MADS.WPN.RELOAD.TIME, ''MINUTES
596         A MADS.XMIT.PCT,
597         A MADS.DETECT,
598         A MADS.RDY.RDS,
599         A MADS.RESUP.TIME
600         AND OWNS
601             A MADS.RH.SET
602         AND HAS
603             A F.MADS.RH.SET,
604             A L.MADS.RH.SET,
605             A N.MADS.RH.SET
606
607     EVERY MODEL.AO HAS   ''TIME REFERENCE IS MINUTES
608         A MAO.NAME,
609         A MAO.VELOCITY,    ''                                   HDM PER HOUR
610         A MAO.ALTITUDE,    ''                                     DECAMETERS
611         A MAO.MAGNIFICATION,  ''                                      X POWER
612         A MAO.MIN.PREP,     ''                          DECIMAL MINUTES (REAL)
613         A MAO.MAX.PREP,     ''                          DECIMAL MINUTES (REAL)
614         A MAO.MAX.ALOFT.TIME,      ''                        MINUTES (INTEGER)
615         A MAO.PGM.CAP,   ''                               BINARY, 0 = NO, 1 = YES
616         A MAO.EQ.ID
617
618     EVERY MODEL.AO, AO.RANGE.BAND HAS
619         A MAOR.CIR.ERROR    '' RADIUSIN METERS
620
621     EVERY MODEL.AR HAS
622         A MAR.NAME,
623         A MAR.VELOCITY,
```

```
624         A MAR.MIN.PREP,
625         A MAR.MAX.PREP,
626         A MAR.MAX.ALOFT.TIME,
627         A MAR.MAX.SEARCH.RNG,
628         A MAR.MIN.SEARCH.RNG,
629         A MAR.CEP
630
631     EVERY MODEL.CFR HAS
632         A MCFR.MIN.OFF, ''IN MINUTES
633         A MCFR.MAX.ON, ''IN MINUTES
634         A MCFR.SWEEP.ANGLE, ''IN DEGREES
635         A MCFR.SEARCH.WIDTH, ''WIDTH OF AREA OF RESPONSIBILITY
636         A MCFR.NAME,
637         A MCFR.EQ.ID
638     OWNS
639         A MCFR.RH.LIST
640     HAS
641         A F.MCFR.RH.LIST,
642         A L.MCFR.RH.LIST,
643         A N.MCFR.RH.LIST
644
645     EVERY MODEL.FO HAS
646         A MFO.NAME,
647         A MFO.SEARCH.RATE,    ''IN SQ. METERS / MINUTE
648         A MFO.PGM.CAP,
649         A MFO.EQ.ID
650     OWNS
651         A MFO.RB.SET
652     HAS
653         A F.MFO.RB.SET,
654         A L.MFO.RB.SET,
655         A N.MFO.RB.SET
656
657     EVERY MODEL.FO, NITE.OR.DAY, MOVEMENT.STATUS, TYPE.EQUIPMENT, FO.RANGE.BAND HAS
658         A MNMTR.PROB.DETECT
659
660     EVERY MODEL.PDB HAS
661         A MPDB.KEY.TIME,
662         A MPDB.NAME,
663         A MPDB.EQ.ID
664     OWNS
665         A MPDB.RH.LIST
666     HAS
667         A F.MPDB.RH.LIST,
668         A L.MPDB.RH.LIST,
669         A N.MPDB.RH.LIST
670
671     EVERY MODEL.PIR HAS
672         A MPIR.NAME,
673         A MPIR.VELOCITY,
674         A MPIR.MAX.PROCESS,
675         A MPIR.MIN.PROCESS,
676         A MPIR.CIR.ERROR,
677         A MPIR.HALF.COV.WIDTH
678
679     EVERY MODEL.PIR, MOVEMENT.STATUS, ENVIRONMENT, TYPE.EQUIPMENT HAS
680         A MPMET.PROB.DETECT,
681         A MPMET.PROB.ACQUIRE
```

```
682
683     EVERY MODEL.RPV HAS
684         A MRPV.NAME,
685         A MRPV.VELOCITY,
686         A MRPV.MAX.PREP,
687         A MRPV.MIN.PREP,
688         A MRPV.HALF.COV.WIDTH,
689         A MRPV.MAX.ALOFT.TIME,
690         A MRPV.CIR.ERROR,
691         A MRPV.PGM.CAP
692
693     EVERY MODEL.RPV, MOVEMENT.STATUS, ENVIRONMENT, TYPE.EQUIPMENT HAS
694         A MRMET.PROB.DETECT,
695         A MRMET.PROB.ACQUIRE
696
697     EVERY NITE.OR.DAY, PDB.RNG.HACK HAS
698         A MPDB.DET.PROB, ''TIMES 1000
699         A MPDB.CIR.ERROR ''IN METERS
700
701     EVERY PDB.RNG.HACK HAS
702         A PDB.RH.RANGE
703     BELONGS TO
704         A MPDB.RH.LIST
705     HAS
706         A P.MPDB.RH.LIST,
707         A S.MPDB.RH.LIST,
708         A M.MPDB.RH.LIST
709
710     EVERY POSTURE HAS
711         A PT.NAME
712
713     EVERY SEARCH.POINT HAS ''                                          \EQUIVALENCING
714         A (SP.X.COORD,SP.X.GRID),
715         A (SP.Y.COORD,SP.Y.GRID)
716
717     EVERY SECTOR HAS
718         A SE.BNDRY.INT          ''MEASURE OF SECTOR RIGHT BNDRY
719
720     EVERY SENSOR.TYPE HAS
721         A ST.NAME,              ''FO, AO, CB, CM, IR, PH, SD, FL , SO, MT
722         A ST.MIN.XMIT,    ''IN MINUTES*10.
723         A ST.MAX.XMIT,    ''IN MINUTES*10.
724         A ST.TE.PTR,    ''POINTER TO TYPE EQUIP THAT THIS EQUATES TO
725         A ST.MAX.RANGE
726
727     EVERY SIDE HAS
728         A COST.CRITERIA, ''1 = TRUE(APPLICABLE) FOR ARTY LOGIC, 2 =
729         A FDC.THRESHOLD,   ''NOT USED IN LOGIC %26OCT79_%RGR
730         A BREAK.POINT, ''PCT ON HAND FOR CBT-INEFF  %RGR
731         A SIDE.TU.TOTAL,   ''TOTAL NUM OF TYPE UNITS %RGR
732         A REQ.EFF.MOVING,
733         A REQ.EFF.STA,
734         A ARTY.DECIMATE, ''PCT X 100 ARTY TROOPS FOR CBT-INEFF
735         A ARTY.DEGRADE, ''PCT X 100 ARTY TROOPS FOR MAX FIRE RATE
736         A SD.AIRFIELD,
737         A SD.NR.CAS.MISSIONS,
738         A SD.MAX.SORTIE.TP,
739         A SD.TP.SORTIE,
```

```
740         A SD.ASC.MAX.SORTIE,
741         A SD.ASC.RADIUS,
742         A SD.SORTIES.THIS.TP,
743         A SD.NO.FLY.VIS,
744         A SD.POOR.FLY.VIS,
745         A SD.CAS.BRKPT,
746         A SD.PAX.KV.ID,
747         AND OWNS
748              A SD.ADS.SET,
749         A SD.CMSN.QUEUE,
750              A SD.FPO.LIST,
751              A SD.KAS.SET,
752              A SD.OLD.SORTIE.QUEUE,
753         AN FP.SET,
754         A SIDE.PDB.SET,
755         A SIDE.CFR.SET,
756         AN AVAIL.AO.LIST
757     HAS
758         A F.FP.SET,
759         A L.FP.SET,
760         A N.FP.SET,
761         A F.SIDE.PDB.SET,
762         A L.SIDE.PDB.SET,
763         A F.SIDE.CFR.SET,
764         A L.SIDE.CFR.SET,
765         A F.AVAIL.AO.LIST,
766         A L.AVAIL.AO.LIST,
767         A N.SIDE.PDB.SET,
768         A N.SIDE.CFR.SET,
769         A N.AVAIL.AO.LIST
770
771     EVERY SIDE , GROUPING OWNS
772         A UNIT.SET
773     HAS
774         A F.UNIT.SET,
775         A L.UNIT.SET,
776         A N.UNIT.SET
777
778     EVERY SIDE,KILLER.VICTIM HAS
779         A KV.CEM.WPN.NO,
780         A KV.AMMO.CONSUMED,              ''LBS
781         A KV.INITIAL.DENSITY,
782         A KV.EQ.ID
783
784     EVERY SIDE,KILLER.VICTIM,KILLER.VICTIM  HAS  '' KILLER.SIDE, KILLER.VICTIM %030
785         A KV.SCORE                ''HITS - INCAPACITATING DAMAGE
786
787     EVERY SIDE, MF.BAND HAS
788         A MFB.UPPER.LIMIT,  ''HDM
789         A MFB.DELAY         ''MINUTES
790
791     EVERY SIDE, MISSION HAS
792         A SM.TANK.TE,
793         A SM.MIN.TANK.RATIO,
794         A SM.MAX.TANK.RATIO,
795         A SM.MIN.CEQ,
796         AN ILLUM.RULE,
797         A  MINE.USE.RULE,
```

```
798         A  MINE.WD.RULE
799
800     EVERY SIDE, NITE.OR.DAY, MISSION HAS
801         A SMK.USE.RULE,
802         A SMK.WD.RULE
803
804     EVERY SIDE, SECTOR HAS
805         A SS.REAR
806     OWNS
807         A SS.SET
808     HAS
809         A F.SS.SET,
810         A L.SS.SET,
811         A N.SS.SET
812
813     EVERY SIDE, TYPE.WEAPON HAS
814         A STW.RND.FIRED
815
816     EVERY SMOKE.MUNITION HAS
817         A SMK.ID,
818         A SMK.WIDTH,   ''HDM
819         A SMK.MAX.RANGE,   ''HDM
820         A SMK.BURN.TIME,   ''MINUTES
821         A SMK.RND.WT    ''POUNDS
822
823     EVERY SUBMUNITION HAS
824         A SM.NAME       ''SUBMUNITION NAME
825
826     THE SYSTEM OWNS
827         A COL.SET,
828         A BN.CAN.FM.SET,
829         A BATTLE.SET,
830         AN AO.RB.SET,
831         AN AO.EB.SET,
832         AN IF.RATE.LIST,
833         A DF.RATE.LIST
834
835
836     EVERY TERRAIN.TYPE HAS
837         A TT.LOS.SHAPE,
838         A TT.LOS.SCALE,
839         A TT.NLOS.SHAPE,
840         A TT.NLOS.SCALE,
841         A TT.STATIONARY.LOS.BREAK,
842         A TT.MOVING.LOS.BREAK,
843         A TT.M.S.LOS.BREAK,
844         A MOVE.FIRE.DIST,
845         A DEFILADE.DIST,
846         A MOV.FAC
847
848     EVERY LOS.BAND HAS
849         A BAND.RANGE
850
851     EVERY TERRAIN.TYPE,LOS.BAND HAS
852         A LOS.PROB
853
854     EVERY TYPE.BATTLE.FIELD HAS
855         A TBF.NO.BL.UNITS,
```

```
856         A TBF.NO.RD.UNITS,
857         A TBF.BL.MISSION,
858         A TBF.RD.MISSION,
859         A TBF.WIDTH,
860         A TBF.RD.ARMOR.UNITS,
861         A TBF.BL.ARMOR.UNITS,
862         A TBF.RD.MECH.UNITS,
863         A TBF.BL.MECH.UNITS,
864         A TBF.RD.INF.UNITS,
865         A TBF.BL.INF.UNITS,
866         A TBF.RD.HQ.UNITS,
867         A TBF.BL.HQ.UNITS
868     AND OWNS
869          A TEAM.TYPES,
870          A TB.SORT.LIST
871     HAS
872         A F.TB.SORT.LIST,
873         A L.TB.SORT.LIST,
874         A N.TB.SORT.LIST,
875         A F.TEAM.TYPES,
876         A L.TEAM.TYPES,
877         A N.TEAM.TYPES
878
879     EVERY TYPE.BTRY HAS
880         A TB.NAME,              ''155, 8IN, ETC
881         A TB.HOW.EQ.ID,         ''PTR TO EQUIPMENT
882         A TB.RND.PER.LAUNCH,    ''NUMBER OF TUBES ON LAUNCHER
883         A TB.MIN.HOW,           ''MINIMUM HOW TO FIRE MISSION
884         A TB.SUST.FIRE.RATE,    ''IN ROUNDS-PER-MINUTE X 100
885         A TB.MAX.RANGE,         ''IN DECAMETERS
886         A TB.MAX.RAP.RANGE,     ''IN DECAMETERS
887         A TB.SFAIL.MEAN.RNDS,   ''IN ROUNDS
888         A TB.LFAIL.MEAN.RNDS,   ''IN ROUNDS
889         A TB.SFAIL.REPAIR,      ''IN HOURS * 10.
890         A TB.LFAIL.REPAIR,      ''IN HOURS * 10.
891         A TB.SUPPRESS.TIME,     ''IN MINUTES
892         A TB.MIN.PREP,          ''MIN TIME TO PREP A MSN IN MIN X 10
893         A TB.MAX.PREP,          ''MAX TIME TO PREP A MSN IN MIN X 10
894         A TB.MIN.FEBA,          ''IN HEXADECAMETERS
895         A TB.MAX.FEBA,          ''IN HEXADECAMETERS
896         A TB.MARCH.ORDER,       ''IN MINUTES
897         A TB.OCCUPY,        ''IN MINUTES
898         A TB.SHOOT.SCOOT.IND,   ''1 = YES, 0 = NO
899         A TB.MN.FASCAM.SUPP ,   ''MINUTES OF SUPPRESSION
900         A TB.MX.FASCAM.SUPP     ''MINUTES
901     OWNS
902         A TB.TM.LIST        ''TYPE MUNITION POINTERS
903     HAS
904         A F.TB.TM.LIST,
905         A L.TB.TM.LIST,
906         A N.TB.TM.LIST
907
908     EVERY TYPE.BTRY, TB.N.FM HAS
909         A TB.MW.THRESHOLD
910
911     EVERY TYPE.EQUIPMENT HAS
912         A TE.NAME,
913         A TE.PGM.INDIC,
```

```
914         A TE.PROJECTED.AREA, ''IN SQUARE METERS
915         A TE.MIN.MF.LOSS,
916         A TE.MAX.MF.LOSS,
917         A TE.DELTA.T,  ''DEGREES C * 10
918         A TE.HEIGHT    ''DECIMETERS
919     OWNS A
920         TE.SET
921
922     EVERY TYPE.EQUIPMENT , ENVIRONMENT , SUBMUNITION  HAS
923         A TES.LA.EQUIP    ''LETHAL AREA AGAINST EQUIPMENT - SQ METERS X
924
925     EVERY TYPE.UNIT HAS
926          A TU.ATKING.AC,
927         A TU.AC.PER.MSN,
928         A TU.LEVEL,            ''CO, BTRY, BN, ETC
929         A TU.CAT,             ''PTR TO CATEGORY
930         A TU.MIL.WORTH,
931         A TU.FREQ,          ''PERCENT OF UNITS IN FORCE THAT ARE OF
932         A TU.RADIUS,  ''IN METERS
933         A TU.PRIN.TE,
934         A TU.CRIT.NO,           ''NUMBER OF CRITICAL PIECES OF EQUI
935         A TU.MOV.RATE,   ''BASIC MOVEMENT RATE FOR UNIT IN HDM PER HOUR
936         A TU.SIDE,
937         A TU.SUP.PRIORITY,
938         A TU.OPP.PRIORITY,
939         A TU.MF.FACTOR   ''MODIFIES MINE DELAY
940     OWNS
941         A TU.TE.LIST,
942         A TU.NTE.SET
943     BELONGS TO
944         A CT.TU.SET
945     HAS
946         A F.TU.TE.LIST,
947         A L.TU.TE.LIST,
948         A F.TU.NTE.SET,
949         A L.TU.NTE.SET,
950         A P.CT.TU.SET,
951         A S.CT.TU.SET,
952         A N.TU.TE.LIST,
953         A N.TU.NTE.SET,
954         A M.CT.TU.SET
955
956     EVERY TYPE.WEAPON HAS
957         A TW.ROF.AIR,
958         A TW.AC.DET.TIME, ''SECONDS
959         A TW.NAME,
960         A TW.RATE.OF.FIRE,  ''IN TENTHS OF ROUNDS PER MINUTE
961         A TW.NO.SENSORS,
962         A TW.ROUND.VELOCITY,  ''IN HDM PER SECOND
963         A TW.RND.WEIGHT, ''LBS
964         A TW.MAX.RANGE,  ''IN METERS
965         A TW.MIN.RANGE,  ''IN METERS
966         A TW.BASIC.LOAD,
967         A TW.PK.PTR,
968         A TW.NITE.FAC,
969         A TW.FIRE.OTM.PTR,
970         A TW.TYPE.OF.SENSOR,
971         A TW.SPECTRUM,
```

```
        A TW.HFOV,   ''DEGREES * 10
        A TW.VFOV,   ''DEGREES * 10
        A TW.HFOS,   ''DEGREES
        A TW.VFOS    ''DEGREES

EVERY UNIT HAS ''                                                    \EQUIVALENCING
        A UN.AD.AVAIL,
        A UNIT.NOS,
        A UN.ENGAGED.INDEX,
        A UN.TYPE.UNIT,       ''TYPE UNIT
        A (UN.X.COORD,UN.X.GRID), ''IN HEXADECAMETERS
        A (UN.Y.COORD,UN.Y.GRID), ''IN HEXADECAMETERS
        A UN.RADIUS, ''RADIUS IN METERS - DO NOT USE "333" %RGR
        A UN.LAST.ARTY.ENG, ''TIME LAST BTRY VOL(MIN X 100)
        A UN.TIME.LAST.MOVE,  ''TIME SINCE THE UNIT WAS LAST MOVED
        A UN.PARENT,          ''PARENT UNIT INDEX
        A UN.MISSION,         ''ATTACK, DEFEND, ETC
        A UN.STATUS,          ''ADVANCING, WITHDRAWING, OVERWATCHI
        A UN.COLOR,           ''BLUE OR RED
        A UN.POSITION.INDEX, ''LINK OF PATH IN CLOSE CMBT
        A UN.BATTLE.INDEX,   '' %5DEC79_%RGR
        A UN.BTRY.INDEX,   '' %5DEC79_%RGR
        A UN.PTR,
        A UN.FASCAM.RECVD,
        A UN.DELAY
    AND OWNS
        A UN.EQUIP.LIST,
        A UN.SUB.LIST,
        A UN.PATH,            ''PATH FOR UNIT IN CLOSE COMBAT
        A UN.SEGMENT.LIST,
        A UN.LOS.LIST,     ''VISUAL UNITS IN LOS
        A UN.HC.LOS.LIST,    ''HELICOPTERS IN LOS
        A UN.SENSOR.LIST,
        A MO.LIST
    MAY BELONG TO
        A SS.SET,
        A UN.SUB.LIST,
        A FR.UNIT.SET,
        AN AO.DET.TGT.LIST,
        AN AR.DET.TGT.LIST,
        AN HT.TARGET.LIST,
        A TB.SORT.LIST
    BELONGS TO
        A UNIT.SET
    HAS
        A P.TB.SORT.LIST,
        A S.TB.SORT.LIST,
        A M.TB.SORT.LIST,
        A F.UN.EQUIP.LIST,
        A L.UN.EQUIP.LIST,
        A F.UN.SUB.LIST,
        A L.UN.SUB.LIST,
        A F.UN.PATH,
        A L.UN.PATH,
        A F.UN.SEGMENT.LIST,
        A L.UN.SEGMENT.LIST,
        A F.UN.LOS.LIST,
        A L.UN.LOS.LIST,
```

```
1030        A F.UN.SENSOR.LIST,
1031        A L.UN.SENSOR.LIST,
1032        A P.SS.SET,
1033        A S.SS.SET,
1034        A P.UN.SUB.LIST,
1035        A S.UN.SUB.LIST,
1036        A P.FR.UNIT.SET,
1037        A S.FR.UNIT.SET,
1038        A P.AO.DET.TGT.LIST,
1039        A S.AO.DET.TGT.LIST,
1040        A P.AR.DET.TGT.LIST,
1041        A S.AR.DET.TGT.LIST,
1042        A P.UNIT.SET,
1043        A S.UNIT.SET,
1044        A N.UN.EQUIP.LIST,
1045        A N.UN.SUB.LIST,
1046        A N.UN.PATH,
1047        A N.UN.SEGMENT.LIST,
1048        A N.UN.LOS.LIST,
1049        A N.UN.SENSOR.LIST,
1050        A M.SS.SET,
1051        A M.UN.SUB.LIST,
1052        A M.FR.UNIT.SET,
1053        A M.AO.DET.TGT.LIST,
1054        A M.AR.DET.TGT.LIST,
1055        A M.UNIT.SET,
1056        A P.HT.TARGET.LIST,
1057        A S.HT.TARGET.LIST,
1058        A M.HT.TARGET.LIST,
1059        A F.UN.HC.LOS.LIST,
1060        A L.UN.HC.LOS.LIST,
1061        A N.UN.HC.LOS.LIST,
1062        A N.MO.LIST,
1063        A F.MO.LIST,
1064        A L.MO.LIST
1065
1066    EVERY UNIT, UNIT HAS  ''                                                \NEEDED?
1067        A ACT.RANGE FUNCTION
1068
1069    EVERY PK.BAND HAS              ''PK.VECTOR.NO,PK.MOVE.FACTOR.NO REMOVED
1070        A PK.BAND.RNG
1071
1072    EVERY PK.VECTOR,PK.BAND HAS
1073        A PK.PROB
1074
1075    EVERY PK.MOVE.BAND HAS
1076        A PK.MOV.RNG
1077
1078    EVERY PK.MOVE.FACTOR,PK.MOVE.BAND HAS
1079        A PK.MOV.FAC
1080
1081    EVERY PK.F.MOVE.FACTOR, PK.MOVE.BAND HAS
1082        A PK.F.MOV.FAC
1083
1084 ''SECTION FOR TEMPORARY_ENTITIES
1086
1087    TEMPORARY ENTITIES
1088
```

```
1089    EVERY AAT.TGT HAS
1090        AN AATT.UE.LINK,
1091        AN AATT.CUM.AREA
1092        AND BELONGS TO
1093             AN AATT.LIST
1094        AND HAS
1095             A P.AATT.LIST,
1096             A S.AATT.LIST,
1097             A M.AATT.LIST
1098
1099    EVERY AD.SENSOR HAS
1100        AN ADS.UNIT.PTR,
1101        AN ADS.UNIT.STATUS,
1102        AN ADS.MADS.PTR,
1103        AN ADS.NR.SENSORS,
1104        AN ADS.TIME.DOWN,
1105        AN ADS.RDS.REMAINING
1106        AND BELONGS TO
1107             AN SD.ADS.SET
1108        AND HAS
1109             A P.SD.ADS.SET,
1110             A S.SD.ADS.SET,
1111             A M.SD.ADS.SET
1112
1113    EVERY AA.LINK HAS
1114        A AA.UE.LINK,
1115        A AA.FRACTION,
1116    AND BELONGS TO
1117        A AA.SET
1118    HAS
1119        A P.AA.SET,
1120        A S.AA.SET,
1121        A M.AA.SET
1122
1123    EVERY AO.DET.CANDIDATE HAS
1124        A AO.DC.UNIT,
1125        A AO.DC.LEG.DIST,
1126        A AO.DC.DIST
1127    BELONGS TO
1128        AN AO.CAND.DET.LIST
1129    HAS
1130        A P.AO.CAND.DET.LIST,
1131        A S.AO.CAND.DET.LIST,
1132        A M.AO.CAND.DET.LIST
1133
1134    EVERY AR.DET.CANDIDATE HAS
1135        A AR.DC.UNIT,
1136        A AR.DC.Y.COORD
1137    BELONGS TO
1138        AN AR.CAND.DET.LIST
1139    HAS
1140        A P.AR.CAND.DET.LIST,
1141        A S.AR.CAND.DET.LIST,
1142        A M.AR.CAND.DET.LIST
1143
1144    EVERY ATK.ORDER HAS
1145        AN ENEMY.DO,
1146        AN OWN.DO
```

```
1147
1148    EVERY AWARE.UNIT HAS
1149        AN AU.UNIT.ID
1150    AND BELONGS TO
1151            AN AU.LIST
1152
1153    EVERY BATTLE HAS
1154        A BTL.TIME.OF.DAY,
1155        A BTL.SEQ.NO,
1156        A BTL.TERRAIN.TYPE,      ''TERRAIN.TYPE IS A POINTER
1157        A BTL.FIELD,             ''BTL.FIELD IS THE PRESTORED
1158                                 ''BATTLEFIELD TO BE SELECTED
1159        A BTL.BL.UNITS,
1160        A BTL.RD.UNITS,
1161        A BTL.WIDTH,             ''IN HDM
1162        A RED.HB.PRIORITY,           ''PRIORITY FOR HC SUPPORT
1163        A BLUE.HB.PRIORITY,          ''PRIORITY FOR HC SUPPORT
1164        A BTL.BL.HC.TEAM,        ''CURRENT BLUE SUPPORT TEAM
1165        A BTL.RD.HC.TEAM,        ''CURRENT RED SUPPORT TEAM
1166        A BTL.BL.FARRP,          ''BLUE FARRP
1167        A BTL.RD.FARRP,          ''RED FARRP
1168    AND BELONGS TO
1169        A BATTLE.SET
1170    MAY OWN
1171        A BTL.FORCE.SET
1172    HAS
1173        A P.BATTLE.SET,
1174        A S.BATTLE.SET,
1175        A M.BATTLE.SET,
1176        A F.BTL.FORCE.SET,
1177        A L.BTL.FORCE.SET,
1178        A N.BTL.FORCE.SET
1179
1180    EVERY CFP.OBS.LINK HAS
1181        A COL.OBSTACLE.PTR,
1182        A COL.XENTRY,
1183        A COL.XEXIT,
1184        A COL.YENTRY,
1185        A COL.YEXIT
1186        AND BELONGS TO
1187            A COL.SET
1188        AND HAS
1189            A P.COL.SET,
1190            A S.COL.SET,
1191            A M.COL.SET
1192
1193    EVERY CFP.SEGMENT HAS
1194        A CFPS.XSTART,
1195        A CFPS.YSTART,
1196        A CFPS.XEND,
1197        A CFPS.YEND,
1198        A CFPS.TIME.LENGTH
1199        AND OWNS
1200            A SI.LIST
1201        AND BELONGS TO
1202            A CFPS.LIST
1203        AND HAS
1204            A F.SI.LIST,
```

```
1205            A L.SI.LIST,
1206            A N.SI.LIST,
1207            A P.CFPS.LIST,
1208            A S.CFPS.LIST,
1209            A M.CFPS.LIST
1210
1211    EVERY CF.DET.UNIT HAS
1212        A CF.D.BTRY,
1213        A CF.D.CPE,
1214        A CF.D.PD,
1215        A CF.D.PRIORITY,
1216    BELONGS TO
1217        A CF.OP.Q
1218    HAS
1219        A P.CF.OP.Q,
1220        A S.CF.OP.Q,
1221        A M.CF.OP.Q
1222
1223    EVERY CF.RADAR HAS
1224        A CFR.LAST.ON.OR.OFF,
1225        A CFR.ORIENTATION, ''RADIANS TIME 1000
1226        A CFR.US.LINK,
1227        A CF.OPERATOR
1228    BELONGS TO
1229        A SIDE.CFR.SET
1230    OWNS
1231        A CF.OP.Q
1232    HAS
1233        A P.SIDE.CFR.SET,
1234        A S.SIDE.CFR.SET,
1235        A M.SIDE.CFR.SET,
1236        A F.CF.OP.Q,
1237        A L.CF.OP.Q,
1238        A N.CF.OP.Q
1239
1240
1241    EVERY DEF.ORDER HAS
1242        A REINF.THRESH,
1243        AN EN.DIS.OP,
1244        AN OWN.DIS.OP,
1245        AN ORD.MISSION
1246
1247    EVERY DF.NOISE HAS
1248        A DF.TIME,
1249        A DF.UNIT
1250    BELONGS TO
1251        THE DF.RATE.LIST
1252    HAS
1253        A P.DF.RATE.LIST,
1254        A S.DF.RATE.LIST,
1255        A M.DF.RATE.LIST
1256
1257    EVERY EX.AC.ATK.TGT HAS
1258         AN EAAT.AC.DET.TIME,
1259         AN EAAT.BLIND,
1260         AN EAAT.DETECT.TIME,
1261         AN EAAT.DIST.TO.FP,
1262         AN EAAT.DIST.TO.P3,
```

```
1263          AN EAAT.FIRING.RANGE,
1264          AN EAAT.FIRING.TIME,
1265          AN EAAT.MODEL.ADS,
1266          AN EAAT.NUM.FIRES.P,
1267          AN EAAT.NUM.PASSES,
1268          AN EAAT.P1.TO.P2,
1269          AN EAAT.P2.TO.P3,
1270          AN EAAT.P3.TO.P1,
1271          AN EAAT.RANGE,
1272          AN EAAT.RANGE.AT.FP,
1273          AN EAAT.RESULT,
1274          AN EAAT.ROW,
1275          AN EAAT.SUM.CRIT.EQ,
1276          AN EAAT.TIME.TO.LEA,
1277          AN EAAT.WAIT.TIME
1278
1279     EVERY EX.FIRE.MISSION HAS
1280          AN EFM.PREP.TIME,
1281          AN EFM.FIRE.RATE,
1282          AN EFM.SUPPRESS,
1283          AN EFM.LAST.ARTY.ENGAGE,
1284          AN EFM.OLD.CEP,
1285          AN EFM.SUPP.TIME
1286
1287
1288     EVERY EX.FWD.OBSERVER HAS
1289          AN EFO.CANDIDATE,
1290          AN EFO.ENEMY,
1291          AN EFO.SEARCH.TIME,
1292          AN EFO.FO.UNIT,
1293          AN EFO.X.CORRECT,
1294          AN EFO.Y.CORRECT,
1295          AN EFO.LINK,
1296          AN EFO.MIN.XMIT,
1297          AN EFO.MAX.XMIT,
1298          AN EFO.PERIOD.OF.SEARCH,
1299          AN EFO.X.SEARCH.GRID,
1300          AN EFO.Y.SEARCH.GRID,
1301          AN EFO.START.TIME,
1302          AN EFO.TARGET
1303
1304     EVERY EX.SHOOT.OUT HAS
1305          A ESO.CUM1,
1306          A ESO.CUM2,
1307          A ESO.DUMMY,
1308          A ESO.RANGE,
1309          A ESO.SCR1,
1310          A ESO.SCR2,
1311          A ESO.TGT,
1312          A ESO.TGT.EQUIP,
1313          A ESO.TGT.UNIT,
1314          A ESO.WEAPON,
1315          A ESO.WPN,
1316          A ESO.ACQ.TIME,
1317          A ESO.QUANT,
1318          A ESO..TGT
1319
1320     EVERY EX.TGT.REPORT HAS
```

```
1321        AN ETR.DUPLICATE,
1322        AN ETR.DUR,
1323        AN ETR.MAX.PREP,
1324        AN ETR.NEW.STOP,
1325        AN ETR.RFAF,
1326        AN ETR.START.TIME,
1327        AN ETR.TOT.FOLLOW
1328
1329    EVERY FARRP HAS
1330        AN FP.UNIT,
1331        AN FP.BATTLE,
1332        AN FP.NO.TEAMS,
1333        AN REFUEL.TIME,
1334        A REARM.TIME,
1335        A REFUEL.CAP,
1336        A REARM.CAP,
1337        A P.FP.SET,
1338        AN S.FP.SET,
1339        AN M.FP.SET,
1340        AN F.HT.LIST,
1341        AN L.HT.LIST,
1342        AN N.HT.LIST
1343    AND BELONGS TO
1344        AN FP.SET
1345    AND OWNS
1346        AN HT.LIST
1347
1348    EVERY FD.BN.LINK HAS
1349        A FB.BN              ''FA.BN INDEX
1350    BELONGS TO
1351        A FD.BN.LIST
1352    HAS
1353        A P.FD.BN.LIST,
1354        A S.FD.BN.LIST,
1355        A M.FD.BN.LIST
1356
1357    EVERY FD.SCHD.MSN HAS
1358        A FS.START,        ''START TIME OF SCHEDULED BTRY USAGE * 100
1359        A FS.STOP,         ''STOP TIME OF SCHEDULED BTRY USAGE * 100
1360        A FS.BATS          ''NUMBER OF BTRYS SCHEDULED
1361    MAY BELONG TO
1362        A FD.SCHD.LIST
1363    HAS
1364        A  P.FD.SCHD.LIST,
1365        A  S.FD.SCHD.LIST,
1366        A  M.FD.SCHD.LIST
1367
1368    EVERY FIRING.TABLE HAS
1369        A FT.AC.ATK.TGT,
1370        A FT.CAS.MISSION,
1371        A FT.TGT.UNIT,
1372        A FT.TARGET.EQUIP,
1373        A FT.FIRING.WPN,
1374        A FT.PK,    ''IN FRACTION
1375        A FT.PK.BAR,  ''IN FRACTION
1376        A FT.SCORE1,
1377        A FT.SCORE2
1378    AND BELONGS TO
```

```
1379        A UE.TARGET.LIST
1380    HAS
1381        A P.UE.TARGET.LIST,
1382        A S.UE.TARGET.LIST,
1383        A M.UE.TARGET.LIST
1384
1385    EVERY FLIGHT.LEG HAS
1386        A FL.X.START,
1387        A FL.Y.START,
1388        A FL.X.END,
1389        A FL.Y.END
1390    BELONGS TO
1391        AN RPV.FLIGHT.LEG.LIST,
1392        AN AO.FLIGHT.LEG.LIST,
1393        A PIR.FLIGHT.LEG.LIST
1394    HAS
1395        A P.PIR.FLIGHT.LEG.LIST,
1396        A S.PIR.FLIGHT.LEG.LIST,
1397        A P.AO.FLIGHT.LEG.LIST,
1398        A S.AO.FLIGHT.LEG.LIST,
1399        A P.RPV.FLIGHT.LEG.LIST,
1400        A S.RPV.FLIGHT.LEG.LIST,
1401        A M.RPV.FLIGHT.LEG.LIST,
1402        A M.AO.FLIGHT.LEG.LIST,
1403        A M.PIR.FLIGHT.LEG.LIST
1404
1405    EVERY FORCE HAS
1406        A DECISION.POINT,
1407        A FR.CRIT.NO,
1408        A FR.MISSION,
1409        A FR.SIDE,
1410        A FR.CAS.INDIC
1411    MAY OWN
1412        A FR.UNIT.SET
1413    MAY BELONG TO
1414        A BTL.FORCE.SET
1415    HAS
1416        A F.FR.UNIT.SET,
1417        A L.FR.UNIT.SET,
1418        A N.FR.UNIT.SET,
1419        A P.BTL.FORCE.SET,
1420        A S.BTL.FORCE.SET,
1421        A M.BTL.FORCE.SET
1422
1423    EVERY FO.DET.CANDIDATE HAS
1424        A FO.DC.UNIT,
1425        A FO.DC.RANGE
1426    MAY BELONG TO
1427        AN FO.CAND.DET.LIST
1428    HAS
1429        A P.FO.CAND.DET.LIST,
1430        A S.FO.CAND.DET.LIST,
1431        A M.FO.CAND.DET.LIST
1432
1433    EVERY FP.OBSTACLE HAS
1434        AN FPO.XMIN,
1435        AN FPO.XMAX,
1436        AN FPO.YMIN,
```

```
1437        AN FPO.YMAX
1438        AND BELONGS TO
1439             AN SD.FPO.LIST
1440        AND HAS
1441             A P.SD.FPO.LIST,
1442             A S.SD.FPO.LIST,
1443             A M.SD.FPO.LIST
1444
1445    EVERY HC.TEAM HAS
1446        A HT.FARRP,
1447        A HT.STATUS,
1448        A HT.MOVE.TIME,
1449        A HT.LOITER.TIME,
1450        A HT.ARR.BTL.TIME,
1451        A HT.TERMINATOR,
1452        AN F.HT.MEMBER.LIST,
1453        AN L.HT.MEMBER.LIST,
1454        AN N.HT.MEMBER.LIST,
1455        AN F.HT.TARGET.LIST,
1456        AN L.HT.TARGET.LIST,
1457        AN N.HT.TARGET.LIST,
1458        A P.HT.LIST,
1459        A S.HT.LIST,
1460        AN M.HT.LIST
1461    AND OWNS
1462        AN HT.MEMBER.LIST,  ''MEMBERS OF THE TEAM ''
1463        AN HT.TARGET.LIST   ''TEAM TARGET LIST''
1464    AND BELONGS TO
1465        AN HT.LIST
1466
1467    EVERY HELICOPTER HAS
1468        A HC.TYPE,     ''SCOUT OR ATTACK''
1469        A HC.UE.ID,   ''SAME AS UE.ID''
1470        A HC.BTL.TEAM,    ''BATTLE TEAM
1471        A HC.PAIRED,   ''YES OR NO''
1472        A HC.X,        ''X-COORDINATE''
1473        A HC.Y,        ''Y-COORDINATE''
1474        A HC.ALTITUDE, ''MASKED, UNMASKED, ON GROUND''
1475        A HC.STATUS,   ''DETECTING, ENGAGING, ETC.''
1476        A HC.TIME.ALOFT,        ''MINUTES''
1477        A HC.FIRE.MASK,        ''MINUTES''
1478        A P.HT.MEMBER.LIST,
1479        A S.HT.MEMBER.LIST,
1480        A M.HT.MEMBER.LIST
1481    AND BELONGS TO
1482        A HT.MEMBER.LIST
1483    AND OWNS
1484        AN HC.UN.LOS.LIST
1485
1486    EVERY HOW HAS
1487        A HW.BTRY,     ''OWNING BTRY
1488        A HW.SFAIL.RNDS, ''ROUNDS TILL SHORT TERM FAILURE
1489        A HW.LFAIL.RNDS, ''ROUNDS TILL LONG TERM FAILURE
1490    BELONGS TO
1491        A BY.HOW.SET ''WHEN NOT FAILED
1492    HAS
1493        A P.BY.HOW.SET,
1494        A S.BY.HOW.SET,
```

```
1495        A M.BY.HOW.SET
1496
1497    EVERY IF.VOLLEY HAS
1498        A IF.V.TIME,
1499        A IF.V.BTRY
1500    BELONGS TO
1501        A IF.RATE.LIST
1502    HAS
1503        A P.IF.RATE.LIST,
1504        A S.IF.RATE.LIST,
1505        A M.IF.RATE.LIST
1506
1507    EVERY KEYED.SENSOR HAS
1508        A KS.TYPE.SENSOR,
1509        A KS.SENSOR.ID
1510    MAY BELONG TO
1511        A PDB.KEYED.LIST
1512    HAS
1513        A P.PDB.KEYED.LIST,
1514        A S.PDB.KEYED.LIST,
1515        A M.PDB.KEYED.LIST
1516
1517    EVERY KNOWN.AD.SENSOR HAS
1518        A KAS.AD.UNIT
1519        AND BELONGS TO
1520            A SD.KAS.SET
1521        AND HAS
1522            A P.SD.KAS.SET,
1523            A S.SD.KAS.SET,
1524            A M.SD.KAS.SET
1525
1526    EVERY MADS.RH HAS
1527        AN MRH.RANGE,
1528        AN MRH.MIN.ALT,
1529        AN MRH.PD
1530        AND BELONGS TO
1531            AN MADS.RH.SET
1532        AND HAS
1533            A P.MADS.RH.SET,
1534            A S.MADS.RH.SET,
1535            A M.MADS.RH.SET
1536
1537    EVERY MAN.UNIT HAS
1538        A MU.CRIT.NO,
1539        A MU.REINF.IND, ''YES OR NO INDICATING AVAILABILITY FOR
1540        A MU.CUR.ORDER,
1541        A MU.TF.MEM,
1542        A MU.OFFSET.X,
1543        A MU.OFFSET.Y,
1544        A MU.UNIT.ID
1545    AND OWNS
1546        A MU.TF.LIST,
1547        A MU.ORDER.SET
1548    AND BELONGS TO
1549        A MU.TF.LIST
1550    HAS
1551        A F.MU.TF.LIST,
1552        A L.MU.TF.LIST,
```

```
1553        A F.MU.ORDER.SET,
1554        A L.MU.ORDER.SET,
1555        A N.MU.TF.LIST,
1556        A N.MU.ORDER.SET,
1557        A P.MU.TF.LIST,
1558        A S.MU.TF.LIST,
1559        A M.MU.TF.LIST
1560
1561    EVERY MA.LINK HAS
1562        A MA.UE.LINK,
1563        A MA.CASUALTIES
1564    AND BELONGS TO
1565        A MA.SET
1566    AND HAS
1567        A P.MA.SET,
1568        A S.MA.SET,
1569        A M.MA.SET
1570
1571    EVERY MINE.OBSTACLE HAS
1572        A MO.MINEFIELD,
1573        A MO.X.INTEP,
1574        A MO.Y.INTER
1575    AND BELONGS TO
1576        A MO.LIST
1577    AND HAS
1578        A P.MO.LIST,
1579        A S.MO.LIST,
1580        A M.MO.LIST
1581
1582    EVERY MF.POINT HAS
1583        A MFP.X.COORD,
1584        A MFP.Y.COORD
1585    AND BELONGS TO
1586        A MFP.LIST
1587    AND HAS
1588        A P.MFP.LIST,
1589        A S.MFP.LIST,
1590        A M.MFP.LIST
1591
1592    EVERY MOVCOR.ORDER HAS
1593        A DESTIN.X,
1594        A DESTIN.Y,
1595        A MOV.MISSION,
1596        A TYPE.MOVE,
1597        A THRESH.REIN,
1598        A NX.ORDER, ''WHEN MOVE IS COMPLETE
1599        A NX.ORD.ABOVE, ''WHEN IN PROX TO ENEMY AND ABOVE REINF THR
1600        A NX.ORD.BELOW  ''WHEN IN PROX TO ENEMY AND BELOW REINF TH
1601
1602    EVERY MOVDIS.ORDER HAS
1603        A DIR.OF.MOVE,        ''ADVANC OR WITHDR
1604        A DIST.MOVED,
1605        AN ORD.NEXT,
1606        A MOVE.TYPE
1607
1608    EVERY MOVREINF.ORD HAS
1609        A REINFORCED.UN,
1610        AN ATK.UNITS.PTR
```

```
1611
1612    EVERY OLD.SORTIE HAS
1613        AN OS.QTY
1614        AND BELONGS TO
1615             A SD.OLD.SORTIE.QUEUE
1616        AND HAS
1617             A P.SD.OLD.SORTIE.QUEUE,
1618             A S.SD.OLD.SORTIE.QUEUE,
1619             A M.SD.OLD.SORTIE.QUEUE
1620
1621    EVERY ORDER HAS
1622        AN ORD.TYPE,
1623        AN ORD.ID,
1624        AN ORD.SEQ.NO
1625    MAY BELONG TO A MU.ORDER.SET
1626    HAS
1627        A P.MU.ORDER.SET,
1628        A S.MU.ORDER.SET,
1629        A M.MU.ORDER.SET
1630
1631    EVERY PASSIVE.DETECTION.BASE HAS
1632        A PDB.US.LINK,
1633        A PD.OPERATOR
1634    MAY OWN
1635        A PDB.KEYED.LIST,
1636        A PDB.OP.Q
1637    BELONGS TO
1638        A SIDE.PDB.SET
1639    HAS
1640        A F.PDB.KEYED.LIST,
1641        A L.PDB.KEYED.LIST,
1642        A N.PDB.KEYED.LIST,
1643        A F.PDB.OP.Q,
1644        A L.PDB.OP.Q,
1645        A N.PDB.OP.Q,
1646        A P.SIDE.PDB.SET,
1647        A S.SIDE.PDB.SET,
1648        A M.SIDE.PDB.SET
1649
1650    EVERY PD.DET.UNIT HAS
1651        A PD.D.BTRY,
1652        A PD.D.CEP,
1653        A PD.D.PD,
1654        A PD.D.PRIORITY
1655    BELONGS TO
1656        A PDB.OP.Q
1657    HAS
1658        A P.PDB.OP.Q,
1659        A S.PDB.OP.Q,
1660        A M.PDB.OP.Q
1661
1662    EVERY PATH.POINT HAS
1663        A PP.X.POINT,
1664        AND A PP.Y.POINT
1665    BELONGS TO
1666        A PATH.SET
1667    HAS
1668        A P.PATH.SET,
```

```
1669        A S.PATH.SET,
1670        A M.PATH.SET
1671
1672    EVERY PIR.REC.TARGET HAS
1673        A PIR.RT.UNIT
1674    BELONGS TO
1675        A PIR.RECORD.LIST
1676    OWNS
1677        A PIR.RTD.LIST
1678    HAS
1679        A P.PIR.RECORD.LIST,
1680        A S.PIR.RECORD.LIST,
1681        A M.PIR.RECORD.LIST,
1682        A F.PIR.RTD.LIST,
1683        A L.PIR.RTD.LIST,
1684        A N.PIR.RTD.LIST
1685
1686    EVERY PIR.RTD.LINK HAS
1687        A PIR.RTD.TE,     ''TYPE EQUIPMENT PTR
1688        A PIR.RTD.ELEM.PROB,   ''PROB OF A SINGLE ELEMENT BEING DETECT
1689        A PIR.RTD.QUANT    ''QUANT OF EQUIPMENT DETECTED
1690    BELONGS TO
1691        A PIR.RTD.LIST
1692    HAS
1693        A P.PIR.RTD.LIST,
1694        A S.PIR.RTD.LIST,
1695        A M.PIR.RTD.LIST
1696
1697    EVERY POINT HAS
1698        A P.X,
1699        AND A P.Y,
1700    AND BELONGS TO
1701        AN UN.PATH
1702    HAS
1703        A P.UN.PATH,
1704        A S.UN.PATH,
1705        A M.UN.PATH
1706
1707    EVERY REINF.ORDER HAS
1708        A SUC.REINF.OP,
1709        AN UNSUC.REINF.OP
1710
1711    EVERY RPV.DET.CANDIDATE HAS
1712        AN RPV.DC.UNIT,
1713        AN RPV.DC.LEG.DIST,
1714        AN RPV.DC.DIST
1715    AND BELONGS TO
1716        AN RPV.CAND.DET.LIST
1717    HAS
1718        A P.RPV.CAND.DET.LIST,
1719        A S.RPV.CAND.DET.LIST,
1720        A M.RPV.CAND.DET.LIST
1721
1722    EVERY SENSOR.INTERSECT HAS
1723        AN SI.X.ENTRY,
1724        AN SI.Y.ENTRY,
1725        AN SI.X.EXIT,
1726        AN SI.Y.EXIT,
```

```
1727        AN SI.TIME.TIL.INTERSECT,
1728        AN SI.AD.UNIT,
1729        AN SI.ALTITUDE
1730        AND BELONGS TO
1731             AN SI.LIST
1732        AND HAS
1733            A P.SI.LIST,
1734            A S.SI.LIST,
1735            A M.SI.LIST
1736
1737    EVERY SEGMENT HAS
1738        A SEG.LENGTH,
1739        A SEG.UNIT,
1740        A SEG.TYPE,
1741    AND BELONGS TO
1742        A UN.SEGMENT.LIST
1743    HAS
1744        A P.UN.SEGMENT.LIST,
1745        A S.UN.SEGMENT.LIST,
1746        A M.UN.SEGMENT.LIST
1747
1748    EVERY TB.TM.LINK HAS
1749        A TB.TM.RAP,   ''1=RAP,2=NONRAP
1750        A TB.TM.CLASS,''HE,ICM,PGM,ILLUM,SMOKE
1751        A TB.TM,    ''INDEX TO TYPE MUNITION
1752        A TB.TM.FIRED
1753    BELONGS TO
1754        A TB.TM.LIST
1755    HAS
1756        A P.TB.TM.LIST,
1757        A S.TB.TM.LIST,
1758        A M.TB.TM.LIST
1759
1760    EVERY TR.DET.LINK HAS
1761        A TR.DET.TE,       ''TYPE EQUIPMENT PTR
1762        A TR.DET.ELEM.PROB,  ''PROB OF A SINGLE ELEMENT BEING DETECTE
1763        A TR.DET.QUANT       ''QUANT OF EQUIPMENT DETECTED
1764    BELONGS TO
1765        A TR.DET.LIST
1766    HAS
1767        A P.TR.DET.LIST,
1768        A S.TR.DET.LIST,
1769        A M.TR.DET.LIST
1770
1771    EVERY TU.NTE.LINK HAS
1772        A TU.NTE.ID
1773    BELONGS TO
1774        A TU.NTE.SET
1775    HAS
1776        A P.TU.NTE.SET,
1777        A S.TU.NTE.SET,
1778        A M.TU.NTE.SET
1779
1780    EVERY TU.TE.LINK HAS
1781        A TU.TE.ID,            ''PTR TO TYPE EQUIPMENT
1782        A TU.CRITICAL.EQUIP.INDIC,
1783        A TU.TE.QUANT          ''QUANTITY OF EQUIP WITH ID = TU EQUIP
1784    BELONGS TO
```

```
1785        A TU.TE.LIST
1786    HAS
1787        A P.TU.TE.LIST,
1788        A S.TU.TE.LIST,
1789        A M.TU.TE.LIST
1790
1791    EVERY TYPE.TEAM HAS
1792        A TT.UNIT,
1793        A TT.IN.USE,
1794        A TT.SIDE,
1795        AND A TT.TYPE,          ''INF PLAT, MECH PLAT, ETC
1796    AND OWNS
1797        A PATH.SET,
1798    AND BELONGS TO
1799        A TEAM.TYPES
1800    HAS
1801        A F.PATH.SET,
1802        A L.PATH.SET,
1803        A N.PATH.SET,
1804        A P.TEAM.TYPES,
1805        A S.TEAM.TYPES,
1806        A M.TEAM.TYPES
1807
1808    EVERY UE.LINK HAS
1809        A UE.ID,                ''EQUIPMENT INDEX
1810        A UE.CRITICAL.EQUIP.INDIC,
1811        A UE.QUANT,             ''QUANTITY OF EQUIPMENT
1812    BELONGS TO
1813        A UN.EQUIP.LIST
1814    OWNS
1815        UE.TARGET.LIST,     ''EQUIPMENT IN LOS AND DETECTED
1816        SO.LIST,
1817        UE.WEAPON.SET
1818    MAY OWN
1819        A HF.SO.LIST
1820    HAS
1821        A P.UN.EQUIP.LIST,
1822        A S.UN.EQUIP.LIST,
1823        A M.UN.EQUIP.LIST,
1824        A F.UE.TARGET.LIST,
1825        A L.UE.TARGET.LIST,
1826        A N.UE.TARGET.LIST,
1827        A F.SO.LIST,
1828        A L.SO.LIST,
1829        A N.SO.LIST,
1830        A F.UE.WEAPON.SET,
1831        A L.UE.WEAPON.SET,
1832        A N.UE.WEAPON.SET,
1833        A F.HF.SO.LIST,
1834        A L.HF.SO.LIST,
1835        A N.HF.SO.LIST
1836
1837    EVERY US.LINK HAS
1838        A US.SENSOR.TYPE,
1839        A US.MODEL,
1840        A US.UNIT,
1841        A US.FDC,
1842        A US.ID,
```

```
1843        A US.STATUS,
1844        A US.EQ.ID
1845    BELONGS TO
1846        A UN.SENSOR.LIST,
1847        AN AVAIL.AO.LIST  ''MOVED HERE FROM AO %8DEC80_%RWF
1848    HAS A P.UN.SENSOR.LIST,
1849        A S.UN.SENSOR.LIST,
1850        A M.UN.SENSOR.LIST,
1851        A M.AVAIL.AO.LIST,
1852        A P.AVAIL.AO.LIST,
1853        A S.AVAIL.AO.LIST
1854
1855    EVERY VISIBLE.UNIT HAS
1856        A VU.POINTER,
1857        A VU.STATUS,
1858        A VU.PREV.ENG
1859    AND BELONGS TO
1860        A UN.LOS.LIST
1861    MAY BELONG TO
1862        A UN.HC.LOS.LIST,
1863        A HC.UN.LOS.LIST
1864    HAS
1865        A P.UN.LOS.LIST,
1866        A S.UN.LOS.LIST,
1867        A M.UN.LOS.LIST,
1868        A P.HC.UN.LOS.LIST,
1869        A S.HC.UN.LOS.LIST,
1870        A M.HC.UN.LOS.LIST,
1871        A P.UN.HC.LOS.LIST,
1872        A S.UN.HC.LOS.LIST,
1873        A M.UN.HC.LOS.LIST
1874
1875    EVERY WEAPON HAS
1876        A WPN.ID,  '' POINTER TO TYPE.WEAPON
1877        A (WPN.AC.MUNS,
1878          HC.WPN.TYPE), ''                                    \EQUIVALENCING
1879        A WPN.QUANTITY,
1880        AND SOME WPN.ROUNDS.REMAINING,
1881    AND BELONGS TO
1882        A UE.WEAPON.SET
1883    HAS A P.UE.WEAPON.SET,
1884        A S.UE.WEAPON.SET,
1885        A M.UE.WEAPON.SET
1886
1887 ''SECTION FOR PROCESSES
1889
1890    PROCESSES
1891
1892    EVERY AC.ATK.TGT HAS
1893        AN AAT.CMSN,
1894        AN AAT.AD.UNIT,
1895        AN AAT.RANGE,
1896        AN AAT.VIS.IND,
1897        AN AAT.AC.KILLED.IND,
1898        AN AAT.EX.AATGT
1899        AND OWNS
1900             AN AATT.LIST
1901        AND HAS
```

```
1902            AN F.AATT.LIST,
1903            AN L.AATT.LIST,
1904            AN N.AATT.LIST
1905
1906    EVERY AIRBORNE.RADAR HAS
1907        AN AR.X.START,
1908        AN AR.Y.START,
1909        AN AR.Y.FINISH,
1910        AN AR.DURATION
1911    OWNS
1912        AN AR.DET.TGT.LIST,
1913        AN AR.CAND.DET.LIST
1914    HAS
1915        A F.AR.DET.TGT.LIST,
1916        A L.AR.DET.TGT.LIST,
1917        A F.AR.CAND.DET.LIST,
1918        A L.AR.CAND.DET.LIST,
1919        A N.AR.DET.TGT.LIST,
1920        A N.AR.CAND.DET.LIST
1921
1922    EVERY AIR.OBSERVER HAS
1923        AN AO.VELOCITY,
1924        AN AO.CURRENT.TR,
1925        AN AO.US.LINK,
1926        AN AO.X.START,
1927        AN AO.Y.START
1928    OWNS
1929        AN AO.FLIGHT.LEG.LIST,
1930        AN AO.CAND.DET.LIST,
1931        AN AO.DET.TGT.LIST
1932    HAS
1933        A F.AO.FLIGHT.LEG.LIST,
1934        A L.AO.FLIGHT.LEG.LIST,
1935        A F.AO.CAND.DET.LIST,
1936        A L.AO.CAND.DET.LIST,
1937        A F.AO.DET.TGT.LIST,
1938        A L.AO.DET.TGT.LIST,
1939        A N.AO.FLIGHT.LEG.LIST,
1940        A N.AO.CAND.DET.LIST,
1941        A N.AO.DET.TGT.LIST
1942
1943    EVERY ARTY.ASSESS HAS
1944        A AA.DESTRUCT.INDIC,
1945        A AA.FIRE.MISSION,
1946        A AA.UNIT
1947    AND OWNS
1948        A AA.SET
1949    HAS
1950        A F.AA.SET,
1951        A L.AA.SET,
1952        A N.AA.SET
1953
1954    EVERY ASSESSMENT HAS
1955        A AS.PK,  ''IN PER CENT
1956        A AS.TGT.EQUIP,
1957        A AS.TGT.UNIT,
1958        A AS.FIRING.UNIT,
1959        A AS.SHOOT.OUT,
```

```
1960        A AS.DESTRUCT.INDIC
1961
1962    EVERY CAS.MISSION HAS
1963        A CMSN.SEQ.NR,
1964        A CMSN.TYPE,      ''ON.CALL OR PREPLANNED
1965        A CMSN.Q.FLAG,
1966        A CMSN.SIDE,
1967        A CMSN.AC.TYPE,
1968        A CMSN.NR.AC,
1969        A CMSN.NR.ABORTED,
1970        A CMSN.NR.SURV.AC,
1971        A CMSN.TGT.UNIT,
1972        A CMSN.SCOREBOARD,
1973        A CMSN.REQUEST.TIME, ''HOURS
1974        A CMSN.START.TIME, ''HOURS
1975        A CMSN.TAKE.OFF.TIME, ''HOURS
1976        A CMSN.FLIGHT.TIME, ''HOURS
1977        A CMSN.FIRST.PASS.TIME, ''HOURS
1978        A CMSN.ASP.STATUS, ''BUSY OR IDLE
1979        A CMSN.P1.ADU.RANGE,
1980        A CMSN.P3.ADU.RANGE,
1981        A CMSN.ANGLE
1982        AND OWNS
1983            A CFPS.LIST
1984        AND BELONGS TO
1985            A SD.CMSN.QUEUE
1986        AND HAS
1987            A F.CFPS.LIST,
1988            A L.CFPS.LIST,
1989            A N.CFPS.LIST,
1990            A P.SD.CMSN.QUEUE,
1991            A S.SD.CMSN.QUEUE,
1992            A M.SD.CMSN.QUEUE
1993
1994    EVERY FIRE.MISSION HAS
1995        A FM.TGT,             ''PNTR TO TARGET.REPORT
1996        A FM.BTRY,            ''BATTERY ASSIGNED MISSION
1997        A FM.STATUS,          ''ACTIVE OR QUEUED
1998        A FM.RAP.FLAG,        ''1=RAPMSN,2=NONRAP
1999        A FM.RANGE,           ''ESTIMATED RANGE IN DECAMETERS
2000        A FM.N.VOLS,          ''NUMBER OF VOLLEYS TO BE FIRED
2001        A FM.Q.SIZE,          ''# OF FM'S IN FM.BTRY.QUEUE
2002        A FM.TM,              ''ROUND TYPE TO BE FIRED
2003        A FM.TM.CLASS,        '' "HE" OR "ICM"
2004        A FM.PRIORITY,        ''SAME AS TARGET REPORT
2005        A FM.START.TIME,
2006        A FM.PREP.TIME,       ''FOR PGM MSNS
2007        A FM.TOF.TIME,        ''FOR PGM MSNS
2008        A FM.FIRED.VOLS,      ''VOLLEYS ALREADY FIRED
2009        A FM.EX.FIRE.MISSION
2010    MAY BELONG TO
2011        A TR.FM.LIST,
2012        A BN.CAN.FM.SET,                  ''CANDIDATE FIRE MISSION SET
2013        A BY.SCHD.LIST,
2014        A BY.FM.QUEUE,
2015        A FO.CUR.FM.LIST
2016    HAS
2017        A P.TR.FM.LIST,
```

EQ.NAME (LIE.ID (TB.HOW.EQ.ID (BY.TYPE (FM. BTRY (FM)))))

```
2018        A S.TR.FM.LIST,
2019        A P.BN.CAN.FM.SET,
2020        A S.BN.CAN.FM.SET,
2021        A P.BY.SCHD.LIST,
2022        A S.BY.SCHD.LIST,
2023        A P.BY.FM.QUEUE,
2024        A S.BY.FM.QUEUE,
2025        A P.FO.CUR.FM.LIST,
2026        A S.FO.CUR.FM.LIST,
2027        A M.TR.FM.LIST,
2028        A M.BN.CAN.FM.SET,
2029        A M.BY.SCHD.LIST,
2030        A M.BY.FM.QUEUE,
2031        A M.FO.CUR.FM.LIST
2032
2033    EVERY FORWARD.OBSERVER HAS          ''TIME REFERENCE IS MINUTES
2034        AN FO.US.LINK,
2035        AN FO.REL.DIRECTION,
2036        AN FO.CURRENT.TR,
2037        AN FO.FST.INDIC,
2038        AN FO.EX.FWD.OBSERVER
2039    OWNS
2040        AN FO.CAND.DET.LIST,
2041        AN FO.TGT.RPT.LIST,
2042        AN FO.CUR.FM.LIST
2043    HAS
2044        A F.FO.CAND.DET.LIST,
2045        A L.FO.CAND.DET.LIST,
2046        A F.FO.TGT.RPT.LIST,
2047        A L.FO.TGT.RPT.LIST,
2048        A F.FO.CUR.FM.LIST,
2049        A L.FO.CUR.FM.LIST,
2050        A N.FO.CAND.DET.LIST,
2051        A N.FO.TGT.RPT.LIST,
2052        A N.FO.CUR.FM.LIST
2053
2054    EVERY HC.ARRIVE.BATTLE HAS
2055        AN AB.FARRP,
2056        AN AB.TEAM
2057
2058    EVERY HC.RETURN.FARRP HAS
2059        AN RF.FARRP,
2060        AN RF.TEAM
2061
2062    EVERY HELICOPTER.FIRE HAS
2063        AN ATK.HELICOPTER,
2064        AN HF.ENEMY.UNIT,
2065        A  HEL.IS.KILLED.INDIC,
2066        AN HF.RESULTS,
2067        AN HF.DESTRUCT.INDIC,
2068        AN HF.FIRING.TABLE,
2069        AN HF.RANGE,
2070        AN HF.TEAM,
2071        AN HF.REINFORCE.IND,
2072        A P.HF.SO.LIST,
2073        A S.HF.SO.LIST,
2074        A M.HF.SO.LIST
2075    AND BELONGS TO
```

```
2076        AN HF.SO.LIST
2077
2078    EVERY HEL.TARGET.ACQUISITION HAS
2079        AN HTA.TEAM,
2080        AN ENEMY.UNITS,
2081        AN HTA.BATTLE,
2082        AN HTA.REINFORCE.IND
2083
2084    EVERY HOW.REPAIR HAS
2085        A HR.HOW
2086
2087    EVERY MINE.ASSESS HAS
2088        A MA.UNIT,
2089        A MA.MINEFIELD,
2090        A MA.BATTLE.ENDED,
2091        A MA.MSN
2092    AND OWNS
2093        A MA.SET
2094    AND HAS
2095        A F.MA.SET,
2096        A L.MA.SET,
2097        A N.MA.SET
2098
2099    EVERY PHOTO.IR.FLIGHT HAS
2100        A PIR.US.LINK,
2101        A PIR.X.START,
2102        A PIR.Y.START
2103    OWNS
2104        A PIR.FLIGHT.LEG.LIST,
2105        A PIR.RECORD.LIST
2106    HAS
2107        A F.PIR.FLIGHT.LEG.LIST,
2108        A L.PIR.FLIGHT.LEG.LIST,
2109        A F.PIR.RECORD.LIST,
2110        A L.PIR.RECORD.LIST,
2111        A N.PIR.FLIGHT.LEG.LIST,
2112        A N.PIR.RECORD.LIST
2113
2114    EVERY REMOTE.PILOT.VEHICLE HAS
2115        A RPV.CURRENT.TARGET,
2116        A RPV.US.LINK,
2117        A RPV.X.START,
2118        A RPV.Y.START
2119    OWNS
2120        A RPV.FLIGHT.LEG.LIST,
2121        A RPV.CAND.DET.LIST
2122    HAS
2123        A F.RPV.CAND.DET.LIST,
2124        A L.RPV.CAND.DET.LIST,
2125        A N.RPV.CAND.DET.LIST,
2126        A F.RPV.FLIGHT.LEG.LIST,
2127        A L.RPV.FLIGHT.LEG.LIST,
2128        A N.RPV.FLIGHT.LEG.LIST
2129
2130    EVERY SHOOT.OUT HAS
2131        A SO.AIR.ATK.INDIC,
2132        A FIRING.EQUIP,
2133        AN FIRER.UNIT,
```

```
2134        A DROP.DEAD.INDICATOR,
2135        A SO.RESULTS,
2136        A SO.DESTRUCT.INDIC,
2137        A SO.HELICOPTER,
2138        A SO.FIRING.TABLE,
2139        A SO.EX.SHOOT.OUT,
2140    AND BELONGS TO
2141        AN SO.LIST
2142    HAS
2143        A P.SO.LIST,
2144        A S.SO.LIST,
2145        A M.SO.LIST
2146
2147    EVERY TARGET.REPORT HAS
2148        A TR.MISSION.TYPE,
2149        A TR.FDC,
2150        A TR.FDC.STATUS,
2151        A TR.SENSOR.TYPE,
2152        A TR.SENSOR.ID,
2153        A TR.REP.UNIT,    ''UNIT REPORTING TARGET
2154        A TR.TGT.UNIT,     ''ACTUAL ENEMY UNIT
2155        A TR.MIL.WORTH,   ''TARGET PRIORITY
2156        A TR.PGM.STATUS,  ''PRECISION GUIDED MUNITION TARGET
2157        A TR.REM.EFFECTS,    ''TIMES 10000
2158        A TR.REQ.EFFECTS,    ''TIMES 100
2159        A TR.CUM.EFFECTS,    ''TIMES 10000
2160        A TR.MOVE,         ''MOVEMENT STATUS
2161        A TR.EST.X,        ''ESTIMATED X COORD TO NEAREST DECAMETER
2162        A TR.EST.Y,        ''ESTIMATED Y COORD TO NEAREST DECAMETER
2163        A TR.EST.RADIUS,  ''IN METERS
2164        A TR.EST.TU, ''ESTIMATED TYPE UNIT
2165        A TR.RECVD.TIME,
2166        A TR.ABORT.TIME, ''TIME TR NO LONGER CONSIDERED VALID
2167        A TR.CEP,           ''CIRCULAR ERROR PROBABLE OF TR LOCATION
2168        A TR.TOT.STATUS, ''0 = NONTOT,  1 = TOT, 2 = TOT FOLLOW-ON
2169        A TR.ASGND.BATS,  ''BTRYS ASGD TO FM NOW ED
2170        A TR.EX.TGT.REPORT
2171    OWNS
2172        A TR.FM.LIST,
2173        A TR.DET.LIST
2174    MAY BELONG TO
2175        A FD.TR.QUEUE,          ''QUEUE OF TRS TO BE COMPUTED AT FDC
2176        A FD.COMPLETE.LIST,    ''LIST OF COMPUTED AND STILL ACTIVE TRS AT FDC
2177        A FO.TGT.RPT.LIST    ''LIST OF TR'S THAT AN FO HAS WORKING
2178    HAS
2179        A F.TR.FM.LIST,
2180        A L.TR.FM.LIST,
2181        A F.TR.DET.LIST,
2182        A L.TR.DET.LIST,
2183        A P.FD.TR.QUEUE,
2184        A S.FD.TR.QUEUE,
2185        A P.FD.COMPLETE.LIST,
2186        A S.FD.COMPLETE.LIST,
2187        A P.FO.TGT.RPT.LIST,
2188        A S.FO.TGT.RPT.LIST,
2189        A N.TR.FM.LIST,
2190        A N.TR.DET.LIST,
2191        A M.FD.TR.QUEUE,
```

```
2192        A M.FD.COMPLETE.LIST,
2193        A M.FO.TGT.RPT.LIST
2194
2195    EVERY WITH.DRAW HAS
2196        A WD.UNIT,
2197        A WD.DESTRUCT.INDIC
2198
2199 ''SECTION FOR EVENTS
2201
2202    ''THE'' EVENT NOTICES INCLUDE CHANGE.LITE, END.SIMULATION
2203
2204    EVERY ACT.ATK HAS
2205        AN ATKUNIT,
2206        AN ATKORDER,
2207        AN ATKENEMY
2208
2209    EVERY ACT.DEF HAS
2210        A DEFUNIT,
2211        A DEFORDER
2212
2213    EVERY ACT.MOVCOR HAS
2214        A MCUNIT,
2215        A MCORDER
2216
2217    EVERY ACT.MOVDIS HAS
2218        A MDUNIT,
2219        A MDORDER
2220
2221    EVERY ACT.REINF HAS
2222        A REINUNIT,
2223        A REINORDER,
2224        A EN.UN.PTR
2225
2226    EVERY AD.ENGAGEMENT HAS
2227        AN AE.INTERSECTION,
2228        AN AE.CALLING.PROCESS
2229
2230    EVERY ARTY.OCCUPATION HAS
2231        AN AOC.BTRY
2232
2233    EVERY BTL.ENDED HAS
2234        A BTL.WINNER,
2235        A BTL.BU,
2236        A BTL.RU
2237
2238    EVERY CFR.ACTIVATION HAS
2239        A CFR.ACT.BTRY
2240
2241    EVERY CFR.OFF HAS
2242        A CFR.OFF.ID,
2243        A CFR.OFF.LINK
2244
2245    EVERY CFR.ON HAS
2246        A CFR.ON.ID,
2247        A CFR.ON.LINK
2248
2249    EVERY CFR.OPERATOR HAS
2250        A CFR.SENS.ID,
```

```
2251        A CFR.DET.UN
2252
2253    EVERY CHANGE.WEATHER HAS
2254        A CW.VISIBILITY
2255
2256    EVERY DQ.OLD.SORTIE.QUEUE HAS
2257         A DOSQ.SIDE
2258
2259    EVERY ENGAGEMENT HAS
2260        A DETECTING.UNIT,
2261        A DETECTED.UNIT
2262
2263    EVERY FEBA.SORTIE HAS
2264        A FE.SECTOR,
2265        A FE.SIDE,
2266        A FE.TIME.INTERVAL
2267
2268    EVERY GET.NX.ORD HAS
2269        A GET.UNIT,
2270        A GET.ORDER,
2271        A GET.OTHER,
2272        A GET.FLAG
2273
2274    EVERY HC.DEPART.BATTLE HAS
2275        A DB.FARRP,
2276        A DB.TEAM,
2277        A DB.ENEMY.UNITS
2278
2279    EVERY HELO.ENGAGEMENT HAS
2280        AN HCEN.TEAM,
2281        AN HCEN.FARRP,
2282        AN HCEN.HELICOPTER,
2283        AN HCEN.ENEMY
2284
2285    EVERY INIT.PREPLAN.CAS HAS
2286        AN IPC.TARGET.UNIT,
2287        AN IPC.AC.TYPE,
2288        AN IPC.NR.AC
2289
2290    EVERY MOVE HAS
2291        A MV.UNIT
2292
2293    EVERY PDB.ACTIVATION HAS
2294        A PDB.ACT.BTRY,
2295        A PDB.ACT.ACTIVITY.TYPE
2296
2297    EVERY PDB.OPERATOR HAS
2298        A PDB.SENS.ID,
2299        A PDB.DET.UNIT
2300
2301    ''THE'' EVENT NOTICES INCLUDE SCHEDULE.ARTY.MOVEMENT
2302
2303    EVERY SEND.TEAM HAS
2304        A ST.FARRP,
2305        A ST.TEAM
2306
2307    EVERY START.ARTY.MOVEMENT HAS
2308        A STAM.BTRY,
```

```
2309        A STAM.DIRECTION
2310
2311    EVERY START.BATTLE HAS
2312        A TERRAIN,
2313        A BLUE.MISSION,
2314        A RED.MISSION,
2315        A BLUE.UNITS,
2316        A RED.UNITS,
2317        A ARG.ARRAY
2318
2319    EVERY START.MOVE HAS
2320        AN SM.UNIT,
2321        AN SM.XCOR,
2322        AN SM.YCOR,
2323        AN SM.TYP.MOV,
2324        AN SM.ORDER
2325
2326    EVERY STOP.ARTY.MOVEMENT HAS
2327        A SPAM.BTRY,
2328        A SPAM.DIRECTION
2329
2330    EVERY UPDATE.LOC HAS
2331        A UL.UNIT,
2332        A UL.XCOR,
2333        A UL.YCOR,
2334        A UL.RATE.MOVE,
2335        A UL.TYP.MOV,
2336        A UL.ORDER
2337
2338    ''THE'' EVENT NOTICES INCLUDE POSITION.REPORT AND DYNAMIC.ANALYSIS.REPORT ''                    \DYN_ANAL
2339
2340    ''THE FOLLOWING STATEMENT IS COMMENTED OUT BECAUSE THERE ARE NO
2341    ''EXTERNAL EVENTS                                                                              \VAX_CHG_4
2342    ''THE EXTERNAL EVENTS ARE SET.DEBUG AND OFF.LINE.ATTRITION
2343    ''SEE SET.DEBUG------------------------------------------------------------------------------>(392)
2344    ''SEE OFF.LINE.ATTRITION--------------------------------------------------------------------->(382)
2345
2346    ''THE FOLLOWING STATEMENT IS COMMENTED OUT BECAUSE THERE ARE NO
2347    ''EXTERNAL EVENTS                                                                              \VAX_CHG_4
2348    ''          EXTERNAL EVENT UNITS ARE 5 AND 15
2349
2350 ''SECTION FOR DEFINITIONS
2352
2353    DEFINE ''GLOBAL INTEGER VARIABLES
2354        NUM.PGM.FIRED,
2355        NUM.KILL.PGM,
2356        DEBUG,
2357        NUM.RAD.INCREMENTS,
2358        NUM.ANG.INCREMENTS,
2359        CUM.NO.BATTLES,
2360        BREAK.POINT,
2361        DISTANCE.INCREMENT,
2362        LEFT.BNDRY.INT,
2363        FRONT.DEPTH,
2364        FEBA.WIDTH,
2365        INIT.X.FEBA,
2366        INIT.Y.FEBA,
2367        HQ.SET.BACK,
```

```
2368        DIS.WITH.DRAW,
2369        DIS.ATTACK,
2370        RN.SEED,
2371        R.WIDTH.UNIT,
2372        B.WIDTH.UNIT,
2373        PK.BAND.RNG,
2374        PK.PROB,
2375        PK.VECTOR.NR,
2376        MOVE.FACTOR.NR,
2377        DFFB.MAX.RANGE,
2378        DT.MAX.BATS,
2379        CDT.MAX.VOLS,
2380        CDI.USAGE.INDICATOR,
2381        ACT.BATTLE.RANGE,          ''OPENING BATTLE RANGE IN METERS
2382        AFK.DELAY,
2383        REIN.DELAY,
2384        REIN.PROX,                 ''RANGE OF REINFORCING UNITS (METERS)
2385        REIN.THRESH,
2386        NO.ATTACK.CONFIGURATIONS,  ''RED AND BLUE COMBINED
2387        NO.SCOUT.CONFIGURATIONS,   ''RED AND BLUE COMBINED
2388        RD.ROUNDS.PER.POPUP,       ''MAX ROUNDS PER POPUP
2389        BL.ROUNDS.PER.POPUP,       ''MAX ROUNDS PER POPUP
2390        HC.SWITCH,                 ''OFF/ON SWITCH FOR HELICOPTERS
2391        HC.OUTPUT.SWITCH,          ''ON/OFF SWITCH FOR HC OUTPUT
2392        BL.MAX.FL.TIME,            ''MAX TIME ALOFT FOR BLUE TEAM (MIN)
2393        RD.MAX.FL.TIME,            ''MAX TIME ALOFT FOR RED TEAM (MIN)
2394        BL.HC.SPACING,             ''LATERAL DISTANCE BETWEEN BLUE HC (M)
2395        RD.HC.SPACING,             ''LATERAL DISTANCE BETWEEN RED HC (M)
2396        MIN.NO.SUP.UNITS,          ''MIN NO BLUE UNITS WHICH CAN BE SUPPORTED BY HC
2397        SCAN.RATE,
2398        LOC.UPDATE.FREQ,
2399        REQ.EFF.MOVING,
2400        REQ.EFF.STA,
2401        N.BLUE.TYPE.EQP,
2402        N.RED.TYPE.EQP,
2403        N.B.WPN.TYPE,
2404        N.R.WPN.TYPE,
2405        TERRAIN.PAR,
2406        SADARM.THRESHOLD,
2407        SDM.CB,
2408        SDM.PGM.RNG,
2409        SDM.TM,
2410        SDM.MNV,
2411        SDM.SSPK,
2412        SDM.MAX.RANGE,
2413        SDM.VOLLEY.RADIUS,
2414        TOT.KILL.SDM,
2415        TOT.FIRED.SDM,
2416        GOOD.WEATHER.PROB,
2417        FASCAM.VOLLEYS,
2418        ILLUM.DEBUG,
2419        ILLUM.SWITCH,
2420        MAX.FASCAM.RANGE,
2421        MIN.FASCAM.RANGE,
2422        MAX.ATT.FASCAM,
2423        MAX.WD.FASCAM,
2424        MF.DEBUG,
2425        MF.SWITCH,
```

```
2426        SMK.DEBUG,
2427        SMK.SWITCH,
2428        VISIBILITY,
2429       TACAIR.DEBUG,
2430       CAS.MSN.RPT.FLAG,
2431       TER.W.INC,
2432        NITE.VIS.PCT,
2433        CALP.ON,
2434        TGW.SDM.RANGE,
2435        TGW.ATK.CRITERIA,
2436        TGW.THRESHOLD,
2437        TGW.MIN.RANGE,
2438        TGW.TM,
2439        TGW.FIRED.RNDS,
2440        TGW.TOT.KILLS,
2441        MINEFIELD.PRINT,
2442        PK.DEFILADE
2443        AS INTEGER VARIABLES
2444
2445    DEFINE ''SIGNED INTEGER VARIABLES ******************************************
2446        MOVE.FIRE.DIST,
2447        AAT.AD.UNIT,
2448        AAT.RANGE,
2449        ACT.EQUIP.ID,
2450         ACT.SUBSTITURE,
2451         ACT.WEATHER.DEGRADE,
2452        ACT.MIN.ALT,
2453        ACT.NORM.ALT,
2454        ACT.SPEED,
2455        ACT.PASS.TIME,
2456        ACT.MAX.ALOFT,
2457        ACT.PROB.SORTIE.ABORT,
2458        ACT.P1.DIST,
2459        ACT.P2.DIST,
2460        ACT.P3.DIST,
2461        ACT.X1,
2462        ACT.X2,
2463        ACT.X3,
2464        ACT.Y1,
2465        ACT.Y2,
2466        ACT.Y3,
2467        ACT.Z1,
2468        ACT.Z2,
2469        ACT.Z3,
2470        CFPS.XSTART,
2471        CFPS.XEND,
2472        CFPS.YSTART,
2473        CFPS.YEND,
2474        COL.XENTRY,
2475         COL.XEXIT,
2476       COL.YENTRY,
2477       COL.YEXIT,
2478       CMSN.SEQ.NR,
2479       CMSN.TGT.UNIT,
2480       CMSN.P1.ADU.RANGE,
2481       CMSN.P2.ADU.RANGE,
2482       MRH.PD,
2483       SD.AIRFIELD,
```

```
2484          SD.NR.CAS.MISSIONS,
2485          SD.ASC.RADIUS,
2486          SD.NO.FLY.VIS,
2487          SD.POOR.FLY.VIS,
2488          WPN.AC.MUNS,
2489          BY.STOP.FASCAM.SUPP,
2490          DEFILADE.DIST,
2491          MFP.X.COORD,
2492          MFP.Y.COORD,
2493          TW.PK.PTR,
2494          EQUIP.PK.PTR,
2495          PK.MOV.RNG,
2496          TW.FIRE.OTM.PTR,
2497          THRESH.REIN,
2498          MU.OFFSET.X,
2499          MU.OFFSET.Y,
2500          DESTIN.X,
2501          DESTIN.Y,
2502          UE.QUANT,
2503          UN.TYPE.UNIT,
2504          UN.X.COORD,
2505          UN.Y.COORD,
2506          UN.X.GRID,
2507          UN.Y.GRID,
2508          UN.RADIUS,
2509          UN.FASCAM.RECVD,
2510          UN.DELAY,
2511          SM.XCOR,
2512          SM.YCOR,
2513          UL.XCOR,
2514          UL.YCOR,
2515          BAND.RANGE,
2516          BY.BN.RANK,
2517          BY.FIRE.RATE,
2518          CFR.CIR.ERROR,
2519          CFR.DET.PROB,
2520          CFR.ORIENTATION,
2521          CF.D.BTRY,
2522          CF.D.CPE,
2523          CF.D.PD,
2524          CF.D.PRIORITY,
2525          EQ.MAX.SPEED,
2526          EQ.PERSONNEL.LOAD,
2527          EFO.FO.UNIT,
2528          EFO.X.CORRECT,
2529          EFO.Y.CORRECT,
2530          EFO.X.SEARCH.GRID,
2531          EFO.Y.SEARCH.GRID,
2532          FD.FDC,
2533          FD.MIN.TIME,
2534          FD.MAX.TIME,
2535          HC.X,
2536          HC.Y,
2537          HE.COST,
2538          HE.DUST.DURATION,
2539          HE.ROUND.RAD,
2540          HE.VOL.DUST.RAD,
2541          HE.VOLLEY.RAD,
```

```
2542        HE.WEIGHT,
2543        HE.RH.TOTAL.CPE,
2544        HE.RH.ROUND.CPE,
2545        IC.RELIABILITY,
2546        IC.VOLLEY.RAD,
2547        IC.N.SUBM,
2548        IC.RH.TOTAL.CPE,
2549        IC.RH.ROUND.CPE,
2550        ILLUM.RADIUS,
2551        ILLUM.MAX.RANGE,
2552        ILLUM.DURATION,
2553        MAO.PGM.CAP,
2554        MAR.MIN.PREP,
2555        MAR.MAX.PREP,
2556        MA.CASUALTIES,
2557        MF.COLOR,
2558        MF.X.HIGH,
2559        MF.X.LOW,
2560        MF.Y.HIGH,
2561        MF.Y.LOW,
2562        MO.X.INTER,
2563        MO.Y.INTER,
2564        MAR.MAX.ALOFT.TIME,
2565        MPIR.MAX.PROCESS,
2566        MPIR.MIN.PROCESS,
2567        MPIR.CIR.ERROR,
2568        MRPV.MAX.PREP,
2569        MRPV.MIN.PREP,
2570        MRPV.CIR.ERROR,
2571        US.EQ.ID,
2572        SPAM.DIRECTION,
2573        SP.X.COORD,
2574        SP.Y.COORD,
2575        SP.X.GRID,
2576        SP.Y.GRID,
2577        STAM.DIRECTION,
2578        SMK.WIDTH,
2579        SMK.MAX.RANGE,
2580        SMK.BURN.TIME,
2581        TB.HOW.EQ.ID,
2582        TB.SUST.FIRE.RATE,
2583        TB.MAX.RANGE,
2584        TB.SFAIL.MEAN.RNDS,
2585        TB.LFAIL.MEAN.RNDS,
2586        TB.SFAIL.REPAIR,
2587        TB.LFAIL.REPAIR,
2588        TB.SUPPRESS.TIME,
2589        TB.MIN.PREP,
2590        TB.MAX.PREP,
2591        TB.MIN.FEBA,
2592        TB.MAX.FEBA,
2593        TE.DELTA.T,
2594        TE.HEIGHT,
2595        TU.MIL.WORTH,
2596        TU.FREQ,
2597        TU.RADIUS,
2598        TU.CRIT.NO,
2599        TU.MOV.RATE,
```

```
2600        TU.OPP.PRIORITY,
2601        TU.SUP.PRIORITY,
2602        TU.MF.FACTOR,
2603        TW.MAX.RANGE,
2604        TW.MIN.RANGE,
2605        TW.BASIC.LOAD,
2606        TW.TYPE.OF.SENSOR,
2607        TW.SPECTRUM,
2608        TW.HFOV,
2609        TW.VFOV,
2610        TW.HFOS,
2611        TW.VFOS,
2612        AO.DC.UNIT,
2613        AO.DC.LEG.DIST,
2614        AO.DC.DIST,
2615        FT.TGT.UNIT,
2616        FT.TARGET.EQUIP,
2617        FT.SCORE1,
2618        FT.SCORE2,
2619        DECISION.POINT,
2620        FR.CRIT.NO,
2621        HW.BTRY,
2622        HW.SFAIL.RNDS,
2623        HW.LFAIL.RNDS,
2624        PP.X.POINT,
2625        PP.Y.POINT,
2626        PIR.RTD.ELEM.PROB,
2627        PIR.RTD.QUANT,
2628        P.X,
2629        P.Y,
2630        RPV.DC.UNIT,
2631        RPV.DC.LEG.DIST,
2632        RPV.DC.DIST,
2633        SEG.LENGTH,
2634        SEG.UNIT,
2635        SEG.TYPE,
2636        TR.DET.TE,
2637        TR.DET.ELEM.PROB,
2638        TR.DET.QUANT,
2639        TU.TE.ID,
2640        TU.CRITICAL.EQUIP.INDIC,
2641        UE.ID,
2642        FM.BTRY,
2643        DROP.DEAD.INDICATOR,
2644        TR.REM.EFFECTS,
2645        TR.REP.UNIT,
2646        TR.TGT.UNIT,
2647        TR.MIL.WORTH,
2648        TR.EST.X,
2649        TR.EST.Y,
2650        SI.X.ENTRY,
2651        SI.Y.ENTRY,
2652        SI.X.EXIT,
2653        SI.Y.EXIT,
2654        TR.EST.RADIUS
2655        AS SIGNED INTEGER VARIABLES
2656
2657
```

```
2658    DEFINE ''TEXT VARIABLES  ********************************************** \TEXT
2659        AM.NAME,
2660        MADS.NAME,
2661        CT.NAME,
2662        DIR.OF.MOVE,
2663        EN.NAME,
2664        EQ.NAME,
2665        FB.MISSION,
2666        FMM.ID,
2667        FM.TM.CLASS,
2668        FZ.NAME,
2669        GP.NAME,
2670        HE.ID,
2671        IC.ID,
2672        ILLUM.ID,
2673        KV.WPN.NAME,
2674        MAO.NAME,
2675        MAR.NAME,
2676        MCFR.NAME,
2677        MFO.NAME,
2678        MF.ID,
2679        MN.NAME,
2680        MOVE.TYPE,
2681        MPDB.NAME,
2682        MPIR.NAME,
2683        MRPV.NAME,
2684        ORD.TYPE,
2685        PDB.ACT.ACTIVITY.TYPE,
2686        PGM.MARK,
2687        PT.NAME,
2688        SMK.ID,
2689        SDM.MARK,
2690        SM.NAME,
2691        SM.TYP.MOV,
2692        ST.NAME,
2693        TB.NAME,
2694        TB.TM.CLASS,
2695        TB.WPN.NAME,
2696        TE.NAME,
2697        TR.MISSION.TYPE,
2698        TR.SENSOR.TYPE,
2699        TU.LEVEL,
2700        TW.NAME,
2701        TYPE.MOVE,
2702        UL.TYP.MOV,
2703        WD.NAME,
2704        WS.NAME,
2705        HC.DEBUG,
2706        TGW.MARK
2707        AS TEXT VARIABLES '' \TEXT
2708
2709    DEFINE ''REAL VARIABLES   ********************************************
2710        AA.FRACTION,
2711        ADS.TIME.DOWN,
2712        AS.PK,
2713        ACT.ANGLE.P1.P2,
2714        CFPS.TIME.LENGTH,
2715        CMSN.ANGLE,
```

```
2716        CMSN.FIRST.PASS.TIME,
2717        CMSN.FLIGHT.TIME,
2718        CMSN.REQUEST.TIME,
2719        CMSN.START.TIME,
2720        CMSN.TAKE.OFF.TIME,
2721        SI.TIME.TIL.INTERSECT,
2722        SM.MIN.TANK.RATIO,
2723        SM.MAX.TANK.RATIO,
2724        BMNT,
2725        CFR.LAST.ON.OR.OFF,
2726        DF.TIME,
2727        EFO.SEARCH.TIME,
2728        EFO.PERIOD.OF.SEARCH,
2729        EFO.START.TIME,
2730        EFM.PREP.TIME,
2731        EFM.FIRE.RATE,
2732        EFM.SUPPRESS,
2733        ESO.ACQ.TIME,
2734        EENT,
2735        FE.TIME.INTERVAL,
2736        FM.START.TIME,
2737        FO.REL.DIRECTION,
2738        FT.PK,
2739        FT.PK.BAR,
2740        IC.TB.INTERCEPT,
2741        IC.TB.SLOPE,
2742        IF.V.TIME,
2743        MAO.MAX.PREP,
2744        PGM.RELY,
2745        MAO.MIN.PREP,
2746        MOV.FAC,
2747        NITE.MOV.FAC,
2748        POS.REP.INT,
2749        STOP.SIMULATION.TIME,
2750        TAC.MOV.FAC,
2751        TIME.BETWEEN.ARTY.MOVE,
2752        TR.ABORT.TIME,
2753        TR.RECVD.TIME,
2754        TT.LOS.SHAPE,
2755        TT.LOS.SCALE,
2756        TT.NLOS.SHAPE,
2757        TT.NLOS.SCALE,
2758        UN.TIME.LAST.MOVE,
2759        TW.NITE.FAC,
2760        PK.MOV.FAC,
2761        BL.ATK.FAIL.PROB,       ''ATTACK FAILURE PROBABILITY PER MISSION
2762        BL.SCT.FAIL.PROB,       ''SCOUT FAILURE PROBABILITY PER MISSION
2763        RD.ATK.FAIL.PROB,       ''ATTACK FAILURE PROBABILITY PER MISSION
2764        RD.SCT.FAIL.PROB,       ''SCOUT FAILURE PROBABILITY PER MISSION
2765        PK.F.MOV.FAC,
2766        BL.MIN.HANDOFF.TIME,    ''MIN TIME TO HAND OFF TARGETS
2767        RD.MIN.HANDOFF.TIME,    ''MIN TIME TO HAND OFF TARGETS (SEC)
2768        BL.MAX.HANDOFF.TIME,    ''MAX TIME TO HAND OFF TARGETS (SEC)
2769        RD.MAX.HANDOFF.TIME,    ''MAX TIME TO HAND OFF TARGETS (SEC)
2770        BL.MIN.MASK.TIME,       ''MIN TIME TO REMAIN MASKED (SEC)
2771        RD.MIN.MASK.TIME,       ''MIN TIME TO REMAIN MASKED (SEC)
2772        BL.MAX.MASK.TIME,       ''MAX TIME TO REMAIN MASKED (SEC)
2773        RD.MAX.MASK.TIME,       ''MAX TIME TO REMAIN MASKED (SEC)
```

```
2774        BL.MIN.UNMASK.TIME,      ''MIN TIME TO REMAIN UNMASKED (SEC)
2775        RD.MIN.UNMASK.TIME,      ''MIN TIME TO REMAIN UNMASKED (SEC)
2776        BL.MAX.UNMASK.TIME,      ''MAX TIME TO REMAIN UNMASKED (SEC)
2777        RD.MAX.UNMASK.TIME,      ''MAX TIME TO REMAIN UNMASKED (SEC)
2778        BL.LOW.FRAC.RANGE,       ''SMALLEST FRACTION OF HC'S LONGEST RANGE WPN AT
2779        RD.LOW.FRAC.RANGE,       ''WHICH IT WILL BE POSITIONED FROM CLOSEST OPP UNT
2780        BL.HIGH.FRAC.RANGE,      ''LARGEST FRACTION OF HC'S LONGEST RANGE
2781        RD.HIGH.FRAC.RANGE,      ''WEAPON AT WHICH IT WILL BE POSITIONED
2782        HT.ARR.BTL.TIME,
2783        TGW.AVG.PK,
2784        EAAT.DETECT.TIME,
2785        EAAT.WAIT.TIME,
2786        EAAT.FIRING.TIME,
2787        EAAT.TIME.TO.LEA
2788         AS REAL VARIABLES
2789
2790    ''DEFINE SETS  ********************************************************
2791    DEFINE AO.CAND.DET.LIST AS A SET RANKED BY LOW AO.DC.LEG.DIST
2792    DEFINE AO.DET.TGT.LIST, AR.DET.TGT.LIST  AS FIFO SETS
2793    DEFINE AO.EB.SET AS A SET RANKED BY LOW AO.EB.ALTITUDE
2794    DEFINE AO.FLIGHT.LEG.LIST AS A FIFO SET
2795    DEFINE AO.RB.SET AS A SET RANKED BY LOW AO.RB.RANGE
2796    DEFINE AVAIL.AO.LIST AS A FIFO SET
2797    DEFINE BN.BTRY.SET AS A SET RANKED BY LOW BY.BN.RANK
2798    DEFINE BN.CAN.FM.SET AS A SET RANKED BY LOW FM.Q.SIZE, THEN BY ''                \1
2799        LOW FM.RANGE
2800    DEFINE BY.FM.QUEUE AS A SET RANKED BY HIGH FM.PRIORITY
2801    DEFINE BY.HOW.SET AS A LIFO SET
2802    DEFINE BY.SCHD.LIST AS A SET RANKED BY LOW FM.START.TIME
2803    DEFINE CF.OP.Q. AS A SET RANKED BY HIGH CF.D.PRIORITY
2804    DEFINE COL.SET                              AS A SET RANKED BY LOW COL.XENTRY
2805    DEFINE DF.RATE.LIST AS A FIFO SET
2806    DEFINE FD.COMPLETE.LIST AS A LIFO SET
2807    DEFINE FD.SCHD.LIST AS A SET RANKED BY LOW FS.START
2808    DEFINE FD.TR.QUEUE AS A SET RANKED BY HIGH TR.MIL.WORTH
2809    DEFINE FO.CUR.FM.LIST AS A FIFO SET
2810    DEFINE FO.CAND.DET.LIST AS A SET RANKED BY LOW FO.DC.RANGE
2811    DEFINE MFO.RB.SET AS A SET RANKED BY LOW FO.RB.RANGE
2812    DEFINE FO.TGT.RPT.LIST AS A LIFO SET
2813    DEFINE IF.RATE.LIST AS A FIFO SET
2814    DEFINE FP.SET AS A FIFO SET
2815    DEFINE MU.ORDER.SET AS A SET
2816    DEFINE MU.TF.LIST AS A SET
2817    DEFINE MADS.RH.SET                          AS A SET RANKED BY LOW MRH.RANGE
2818    DEFINE PDB.OP.Q AS A SET RANKED BY HIGH PD.D.PRIORITY
2819    DEFINE RPV.CAND.DET.LIST AS A SET
2820    DEFINE RPV.FLIGHT.LEG.LIST AS A FIFO SET
2821    DEFINE SIDE.PDB.SET AS A FIFO SET
2822    DEFINE SIDE.CFR.SET AS A FIFO SET
2823    DEFINE SS.SET AS A SET  RANKED BY HIGH UN.X.COORD
2824    DEFINE FD.BN.LIST AS A FIFO SET
2825    DEFINE TB.SORT.LIST AS A SET RANKED BY LOW UN.Y.COORD
2826    DEFINE TB.TM.LIST AS A FIFO SET
2827    DEFINE TR.DET.LIST AS A LIFO SET
2828    DEFINE TR.FM.LIST AS A LIFO SET
2829    DEFINE UN.EQUIP.LIST, TU.NTE.SET,  AND TU.TE.LIST AS LIFO SETS
2830    DEFINE  UNIT.SET AND UN.SUB.LIST AS FIFO SETS
2831
```

```
2832    ''DEFINE ARRAYS   ****************************************************
2833    DEFINE AR.DET.COEFF AS A 2-DIMENSIONAL REAL ARRAY
2834    DEFINE SHADE AS A 1-DIMENSIONAL TEXT ARRAY ''                          \TEXT
2835    DEFINE PK.POINTER     AS AN INTEGER 2-DIMENSIONAL ARRAY ''              \VAX
2836    DEFINE PK.DEF.POINTER AS AN INTEGER 2-DIMENSIONAL ARRAY ''              \VAX
2837    DEFINE TGT.OTM        AS AN INTEGER 2-DIMENSIONAL ARRAY ''              \VAX
2838    DEFINE FIRE.OTM       AS AN INTEGER 2-DIMENSIONAL ARRAY ''              \VAX
2839    DEFINE ATK.HC, SCT.HC AS TEXT, 1-DIM ARRAYS ''                         \TEXT
2840    DEFINE ANALYSIS AS A 1-DIMENSIONAL INTEGER ARRAY
2841    DEFINE STY.BLUE.EXP, STY.RED.EXP AS INTEGER 2-DIMENSIONAL ARRAYS
2842    DEFINE PGM.HIT, PGM.KILL AS REAL 2-DIMENSIONAL ARRAYS
2843    DEFINE PGM.LINK, FD.UNIT AS INTEGER 1-DIMENSIONAL ARRAYS
2844    DEFINE ANAL.CTR AS A 2-DIMENSIONAL, INTEGER ARRAY ''                 \DYN_ANAL
2845    DEFINE ANAL.TEXT AS A 1-DIMENSIONAL, TEXT ARRAY ''                   \DYN_ANAL
2846
2847    ''DEFINE FUNCTIONS   *************************************************
2848    DEFINE ACT.RANGE AS A REAL FUNCTION WITH 2 ARGUMENTS
2849    DEFINE AR.PROB.DETECT AS A REAL FUNCTION WITH 2 ARGUMENTS
2850    DEFINE BTRY.AVAILABLE AS AN INTEGER FUNCTION WITH 3 ARGUMENTS
2851    DEFINE COLLISION AS AN INTEGER FUNCTION WITH 2 ARGUMENTS
2852    DEFINE COMBINATIONS AS A REAL FUNCTION WITH 2 ARGUMENTS
2853    DEFINE EST.RANGE AS A REAL FUNCTION WITH 2 ARGUMENTS
2854    DEFINE EST.TR.RANGE AS A REAL FUNCTION WITH 2 ARGUMENTS
2855    DEFINE FEBA.BAND AS AN INTEGER FUNCTION WITH 1 ARGUMENT
2856    DEFINE FILE.FD.SCHD AS A ROUTINE WITH 4 ARGUMENTS
2857    DEFINE HE.WLA AS A REAL FUNCTION WITH 6 ARGUMENTS
2858    DEFINE ICM.WLA AS A REAL FUNCTION WITH 4 ARGUMENTS
2859    DEFINE STAY.TIME AS A REAL FUNCTION WITH 1 ARGUMENT
2860
2861 ''SECTION FOR SUBSTITUTIONS
2863
2864    ''SDDL SUBSTITUTIONS
2865    DEFINE ACTIVATE_A TO MEAN ACTIVATE A ''                                  \1
2866    DEFINE ACTIVATE_AN TO MEAN ACTIVATE AN ''                                \1
2867    DEFINE ACTIVATE_THE TO MEAN ACTIVATE THE ''                              \1
2868    DEFINE CANCEL_THE TO MEAN CANCEL THE ''                                  \1
2869    DEFINE CANCEL_THIS TO MEAN CANCEL THIS ''                                \1
2870    DEFINE ENDPREAMBLE     TO MEAN END
2871    DEFINE ENDMAIN         TO MEAN END
2872    DEFINE EXITMAIN        TO MEAN STOP
2873    DEFINE ENDROUTINE      TO MEAN END
2874    DEFINE EXITROUTINE     TO MEAN RETURN
2875    DEFINE ENDFUNCTION     TO MEAN END
2876    DEFINE EXITFUNCTION    TO MEAN RETURN
2877    DEFINE LEFTROUTINE     TO MEAN LEFT ROUTINE
2878    DEFINE ENDLEFTROUTINE  TO MEAN END
2879    DEFINE EXITLEFTROUTINE TO MEAN RETURN
2880    DEFINE ENDEVENT        TO MEAN END
2881    DEFINE EXITEVENT       TO MEAN RETURN
2882    DEFINE ENDPROCESS      TO MEAN END
2883    DEFINE EXITPROCESS     TO MEAN RETURN
2884    DEFINE ELSEIF          TO MEAN ELSE IF
2885    DEFINE ENDIF           TO MEAN ALWAYS
2886    DEFINE ENDLOOP         TO MEAN REPEAT
2887    DEFINE EXITLOOP        TO MEAN LEAVE
2888    DEFINE LOOP            TO MEAN RESUME SUBSTITUTION
2889    DEFINE REACTIVATE_THE TO MEAN REACTIVATE THE ''                          \1
2890    DEFINE RESCHEDULE_THE TO MEAN RESCHEDULE THE ''                          \1
```

```
2891    DEFINE SCHEDULE_A TO MEAN SCHEDULE A ''                                    \1
2892    DEFINE SCHEDULE_AN TO MEAN SCHEDULE AN ''                                  \1
2893    DEFINE SCHEDULE_THE TO MEAN SCHEDULE THE ''                                \1
2894
2895    DEFINE HOURS   TO MEAN 'UITS
2896    DEFINE MINUTES TO MEAN / 60.0 UNITS
2897    DEFINE SECONDS TO MEAN / 3600.0 UNITS
2898
2899    DEFINE NORTH TO MEAN PI.C/2  ''                                            \OPTIMIZE
2900    DEFINE SOUTH TO MEAN 3.*PI.C/2  ''                                         \OPTIMIZE
2901    DEFINE EAST  TO MEAN 0.0
2902    DEFINE WEST  TO MEAN PI.C
2903
2904    ''CT.GROUP DEFINITIONS
2905    DEFINE MANEUVER  TO MEAN 1
2906    DEFINE ARTILLERY TO MEAN 2
2907    DEFINE SUPPORT   TO MEAN 3
2908
2909    DEFINE FEBA.FLIGHT TO MEAN 1
2910
2911    DEFINE ACTIVE TO MEAN 1
2912    DEFINE HOLD   TO MEAN 0
2913
2914    ''PD.OPERATOR VALUES
2915    DEFINE BUSY   TO MEAN 1
2916    DEFINE IDLE   TO MEAN 0
2917
2918    DEFINE TRUE   TO MEAN 1
2919    DEFINE FALSE  TO MEAN 0
2920    DEFINE SADARM TO MEAN 2
2921    DEFINE TGW TO MEAN 3
2922
2923    DEFINE DAY    TO MEAN 1
2924    DEFINE NITE   TO MEAN 2
2925
2926    ''FUZE TYPES
2927    DEFINE PD     TO MEAN 1
2928    DEFINE VT     TO MEAN 2
2929
2930    ''UN.STATUS
2931    DEFINE ADVANCING   TO MEAN 1
2932    DEFINE WITHDRAWING TO MEAN 2
2933    DEFINE STATIONARY  TO MEAN 3
2934    DEFINE MOVING      TO MEAN 4
2935    DEFINE ADV.TO.WITH TO MEAN 5
2936    DEFINE STA.TO.WITH TO MEAN 6
2937
2938    DEFINE YES TO MEAN 1
2939    DEFINE NO  TO MEAN 2
2940
2941    DEFINE RED.TK.PLT    TO MEAN 1
2942    DEFINE RED.MECH.PLT  TO MEAN 2
2943    DEFINE BLUE.TK.PLT   TO MEAN 3
2944    DEFINE RED.CO.HQ     TO MEAN 4
2945    DEFINE BLUE.MECH.PLT TO MEAN 5
2946    DEFINE BLUE.CO.HQ    TO MEAN 6
2947    DEFINE RED.INF.PLT   TO MEAN 7
2948    DEFINE BLUE.INF.PLT  TO MEAN 8
```

```
2949    DEFINE RED.AH.TEAM   TO MEAN 9
2950    DEFINE BLUE.AH.TEAM  TO MEAN 10
2951
2952    ''MN.NAME AND UN.MISSION DEFINITIONS
2953    DEFINE PATROL TO MEAN 1
2954    DEFINE PROBE  TO MEAN 2
2955    DEFINE ATTACK TO MEAN 3
2956    DEFINE DELAY  TO MEAN 4
2957    DEFINE DEFEND TO MEAN 5
2958    DEFINE AMBUSH TO MEAN 6
2959
2960    DEFINE BLUE TO MEAN 2
2961    DEFINE RED  TO MEAN 1
2962
2963    DEFINE HIT  TO MEAN 1
2964    DEFINE MISS TO MEAN 2
2965
2966    ''TERRAIN.PAR SUBSTITUTIONS
2967    DEFINE FULDA        TO MEAN 21
2968    DEFINE NO.GER.PLAIN TO MEAN 22
2969    DEFINE MIX.TERRAIN  TO MEAN 23
2970
2971    ''HC.TYPE
2972    DEFINE SCOUT TO MEAN 1
2973    ''ATTACK DEFINED ABOVE TO MEAN 3
2974
2975    ''HC.ALTITUDE
2976    DEFINE UNMASKED  TO MEAN 1
2977    DEFINE MASKED    TO MEAN 2
2978    DEFINE ON.GROUND TO MEAN 3
2979
2980    ''HT.STATUS
2981    DEFINE READY                  TO MEAN 1
2982    DEFINE MOVING.TO.BATTLE       TO MEAN 2
2983    DEFINE RETURNING.FROM.BATTLE  TO MEAN 3
2984    DEFINE DETECTING              TO MEAN 4
2985    DEFINE ENGAGING               TO MEAN 5
2986    DEFINE REARMING.REFUELING     TO MEAN 6
2987    DEFINE RELEASED               TO MEAN 7
2988
2989    ''CMSN.Q.FLAG
2990    DEFINE NO.FLY.VIS TO MEAN 1
2991    ''NITE DEFINED ABOVE TO MEAN 2
2992    DEFINE SORTIE.RATE TO MEAN 3
2993    DEFINE AIR.SPACE TO MEAN 4
2994
2995    ''CMSN.TYPE
2996    DEFINE PREPLANNED TO MEAN 1
2997    DEFINE ON.CALL TO MEAN 2
2998
2999    ''MADS.FCM
3000    DEFINE LOOK.SHOOT.LOOK              TO MEAN 1
3001    DEFINE RIPPLE                       TO MEAN 2
3002    DEFINE SALVO                        TO MEAN 3
3003
3004    DEFINE NO.SHOT.MADE                 TO MEAN 3
3005    DEFINE NO.PK.PTR                    TO MEAN 4
3006    DEFINE ATGM   TO MEAN 1 ''ANTI-TANK GUIDED MISSILE''
```

```
3007    DEFINE FFATGM TO MEAN 2  ''FIRE & FORGET ATGM''
3008
3009    '' ADS.UNIT.STATUS
3010    DEFINE NOT.READY TO MEAN 2
3011
3012    '' THE FOLLOWING IS ADDED TO TRACK CPU TIME                                    \DYN_ANAL
3013    DEFINE LIB$INIT_TIMER, LIB$STAT_TIMER AS FORTRAN ROUTINES
3014
3015 ENDPREAMBLE
```

TOP-LEVEL ROUTINES

```
3023 ''PROGRAM'' MAIN                                                                                    T001
3024
3025    NORMALLY MODE IS INTEGER
3026
3027    NOW OPEN.INPUT.OUTPUT.FILES  ''------------------------------------------------------------ \VAX>(647)
3028    CALL PERFORM.INSTRUMENTATION  ''---------------------------------------------------------- \DYN_ANAL>(648)
3029    ADD 1 TO ANAL.CTR(1,1)  ''                                                                    \DYN_ANAL
3030    CALL LIB$INIT_TIMER ''-------------------------------------------------------------------- \DYN_ANAL>(649)
3031    CALL MAIN1-------------------------------------------------------------------------------->( 56)
3032    CALL MAIN2-------------------------------------------------------------------------------->(520)
3033    CALL MAIN3-------------------------------------------------------------------------------->( 57)
3034
3035 ENDMAIN
```

```
3036 ROUTINE MAIN1                                                                                    T002
3037     ADD 1 TO ANAL.CTR(2,1)  ''                                                                  \DYN_ANAL
3038
3039     NORMALLY MODE IS INTEGER
3040
3041     LET DISTANCE.INCREMENT = 16
3042     ''LET MAX.DBANK.V = 2500000                                                                      \VAX
3043     USE UNIT 6 FOR OUTPUT ''                                                                CHG\11   \VAX
3044     LET HEADING.V = 'HEADING'
3045     ''CALL HEADING ROUTINE WHEN EACH NEW OUTPUT PAGE STARTS------------------------------------------>(605)
3046
3047     LET N.NITE.OR.DAY = 2
3048     CREATE EVERY NITE.OR.DAY
3049
3050     LET N.SEARCH.POINT = 1
3051     CREATE EVERY SEARCH.POINT
3052
3053     LET N.SIDE=2
3054     CREATE EVERY SIDE
3055
3056     LET N.GROUPING=4
3057     CREATE EVERY GROUPING
3058     LET GP.NAME(1)="MANEUV"
3059     LET GP.NAME(2)="ARTILL"
3060     LET GP.NAME(3)="SUPPOR"
3061     LET GP.NAME(4)="CBTAVN"
3062
3063     RESERVE SHADE AS 2
3064     LET SHADE(1) = "RED"
3065     LET SHADE(2) = "BLUE"
3066
3067 <--EXITROUTINE
3068 ENDROUTINE
```

```
3069 ROUTINE MAIN3                                                                                                    T003
3070    ADD 1 TO ANAL.CTR(3,1)  ''                                                                                \DYN_ANAL
3071
3072    NORMALLY MODE IS INTEGER
3073
3074    LET NITE.OR.DAY = NITE
3075
3076    SCHEDULE_A SCHEDULE.ARTY.MOVEMENT------------------------------------------------------------------------>(390)
3077    IN TIME.BETWEEN.ARTY.MOVE MINUTES
3078    SCHEDULE_A END.SIMULATION AT STOP.SIMULATION.TIME--------------------------------------------------------->(368)
3079    SCHEDULE_A CHANGE.LITE AT BMNT---------------------------------------------------------------------------->(365)
3080    SCHEDULE_A POSITION.REPORT NOW---------------------------------------------------------------------------->(389)
3081
3082    ''PRINT 1 LINE WITH ZTIME.F,  USED.DBANK.V THUS                                                            \VAX
3083    ''CPU TIME AT SIM START = *****.**  SEC  DBANK AFTER INTIALIZATION = *****
3084
3085    START SIMULATION
3086
3087 <--EXITROUTINE
3088 ENDROUTINE
```

```
3089 ROUTINE CREATE.FORCE                                                    ''                                                    T004
3090    GIVEN
3091    BATTLE,
3092    COLOR,
3093    UNITS,
3094    MISSION
3095    ADD 1 TO ANAL.CTR(4,1)  ''                                                                                                 \DYN_ANAL
3096
3097    DEFINE UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
3098    IF COLOR = BLUE
3099       LET .ENEMY = RED
3100    ELSE
3101       LET .ENEMY = BLUE
3102    ALWAYS
3103
3104    LET NO.UNITS = DIM.F(UNITS(*))
3105    CREATE A FORCE
3106    LET FR.SIDE(FORCE) = COLOR
3107    LET DECISION.POINT(FORCE) = DECISION(MISSION,COLOR)
3108    LET FR.MISSION(FORCE) = MISSION
3109    LET FR.CAS.INDIC(FORCE) = NO
3110    FILE THIS FORCE IN THE BTL.FORCE.SET(BATTLE)
3111    LET CRIT.NO = 0
3112    LOOP
3113    FOR I = 1 TO NO.UNITS
3114    DO THIS ''                                                                                                                 /1
3115       IF FR.CAS.INDIC(FORCE) = NO
3116          ''CHECK IF A MEMBER OF THE FORCE IS THE TARGET OF A CAS.MISSION.
3117          FOR EVERY .CAS IN SD.CMSN.QUEUE(.ENEMY)
3118          WITH CMSN.TGT.UNIT(.CAS) = UNITS(I)
3119          FIND THE FIRST CASE
3120          IF FOUND
3121             LET FR.CAS.INDIC(FORCE) = YES
3122          ELSE
3123             ''CHECK THE ACTIVE MISSION.  IF THE AIRSPACE STATUS IS
3124             ''IDLE THE MISSION IS OVER.
3125             FOR EVERY .CAS IN EV.S(I.CAS.MISSION)
3126             WITH CMSN.TGT.UNIT(.CAS) = UNITS(I) AND
3127             CMSN.ASP.STATUS(.CAS) = BUSY
3128             FIND THE FIRST CASE
3129             IF FOUND
3130                LET FR.CAS.INDIC(FORCE) = YES
3131             ALWAYS
3132          ALWAYS
3133       ALWAYS
3134       ADD TU.CRIT.NO(UN.TYPE.UNIT(UNITS(I))) TO CRIT.NO
3135       IF M.FR.UNIT.SET(UNITS(I)) = 0        '' %5NOV79_%RGR
3136          FILE UNITS(I) IN FR.UNIT.SET(FORCE)
3137       ELSE  '' UNIT IS ALREADY IN FORCE SET
3138          PRINT 1 LINE WITH UNITS(I) THUS
3139          = = = CREATEFORCE BATTLE IS RESTARTED WITH UNIT **** == =
3140       ALWAYS
3141       LOOP FOR EACH .MOVCOR IN EV.S(I.ACT.MOVCOR)
3142       WITH MCUNIT(.MOVCOR) = UNITS(I)
3143       DO THIS ''                                                                                                              \1
3144          CANCEL_THE ACT.MOVCOR CALLED .MOVCOR----------------------------------------------------------------------------->(415)
3145          DESTROY THE ACT.MOVCOR CALLED .MOVCOR
3146       ENDLOOP
```

```
3147        LOOP FOR EACH .ATK IN EV.S(I.ACT.ATK)
3148        WITH ATKUNIT(.ATK) = UNITS(I)
3149        DO THIS ''                                                                                                    \1
3150           CANCEL_THE ACT.ATK CALLED .ATK------------------------------------------------------------------------->(412)
3151           DESTROY THE ACT.ATK CALLED .ATK
3152        ENDLOOP
3153     ENDLOOP
3154     LET FR.CRIT.NO(FORCE) = CRIT.NO
3155 <--EXITROUTINE
3156 ENDROUTINE
```

```
3157 ROUTINE CREATE.TEAMS                                                    ''                                                    T005
3158     GIVEN
3159     COLOR,
3160     TYPE
3161     ADD 1 TO ANAL.CTR(5,1)  ''                                                                                              \DYN_ANAL
3162
3163     NORMALLY MODE IS INTEGER
3164
3165     CREATE A TYPE.TEAM
3166
3167     LET TT.SIDE(TYPE.TEAM)=COLOR
3168     LET TT.TYPE(TYPE.TEAM)=TYPE
3169     FILE THE TYPE.TEAM IN TEAM.TYPES(TYPE.BATTLE.FIELD)
3170
3171     READ NO.PATH.POINTS
3172     LOOP FOR I=1 TO NO.PATH.POINTS
3173     DO
3174        CREATE A PATH.POINT
3175        READ PP.X.POINT(PATH.POINT),    ''IN METERS
3176        PP.Y.POINT(PATH.POINT)      ''IN METERS
3177        LET PP.X.POINT=PP.X.POINT/16    ''CONVERT TO HEXADECAMETERS
3178        LET PP.Y.POINT=PP.Y.POINT/16    ''CONVERT TO HEXADECAMETERS
3179        FILE PATH.POINT IN PATH.SET(TYPE.TEAM)
3180     ENDLOOP
3181
3182 <--EXITROUTINE
3183 ENDROUTINE
```

```
3184 ROUTINE FEBA.INITIAL   ''                                                                    T006
3185    ADD 1 TO ANAL.CTR(6,1) ''                                                                 \DYN_ANAL
3186
3187    DEFINE SIDE, SECTOR, UNIT AS INTEGER VARIABLES
3188
3189    LET ERROR.MSG = FALSE  '' INITIALIZE
3190
3191    CREATE EACH SECTOR
3192    ''TRACE
3193    LET SECTOR.WIDTH = FEBA.WIDTH / N.SECTOR
3194    LET LEFT.BNDRY.INT = INIT.Y.FEBA
3195
3196    LOOP FOR EACH SECTOR,
3197    DO THIS ''                                                                                \1
3198       LET SE.BNDRY.INT( SECTOR ) = LEFT.BNDRY.INT
3199       - SECTOR * SECTOR.WIDTH
3200       IF SECTOR = N.SECTOR   '' %17DEC79_%RGR
3201          LET SE.BNDRY.INT(SECTOR) = 0
3202       ENDIF   '' TO ALLOW FOR ODD N.SECTOR VALUES
3203       LET SS.REAR( RED,SECTOR ) = - INIT.X.FEBA - FRONT.DEPTH
3204       LET SS.REAR( BLUE,SECTOR ) =  INIT.X.FEBA - FRONT.DEPTH
3205    ENDLOOP
3206
3207    LOOP FOR EACH SIDE
3208    DO THIS ''                                                                                \1
3209       LOOP FOR EACH UNIT IN THE UNIT.SET(SIDE,MANEUVER),
3210       WHEN UN.MISSION(UNIT) NE PATROL
3211       AND ( MU.CUR.ORDER(UN.PTR(UNIT)) NE 0  '' IS A HQ, OR BELONGS TO A HQ
3212       OR MU.TF.MEM(UN.PTR(UNIT)) > 0 )   '' %14JAN80_%RGR
3213       DO THIS
3214          IF UN.Y.COORD(UNIT) > INIT.Y.FEBA '' OFF THE GRID
3215             LET ERROR.MSG = TRUE  '' DATA ERROR
3216             LIST UNIT.NOS(UNIT), UN.Y.COORD(UNIT), UNIT
3217             LET UN.Y.COORD(UNIT) = INIT.Y.FEBA - 1 ''GIVE IT A CORRECTION
3218          ENDIF
3219          ''THESE LINES ADDED %7MAR80_%RGR
3220          IF UN.Y.COORD(UNIT) LE 0      '' OFF THE GRID %8MAY80_%JLR
3221             LET ERROR.MSG = TRUE  '' DATA ERROR
3222             LIST UNIT.NOS(UNIT), UN.Y.COORD(UNIT), UNIT
3223             LET UN.Y.COORD(UNIT) = 1      '' GIVE IT A CORRECTION
3224          ENDIF
3225          CALL LOCATE.SECTOR  GIVEN UN.Y.COORD(UNIT) YIELDING SECTOR------------------------------------------------------------>(100)
3226          ''LIST UNIT, UNIT.NOS(UNIT), SECTOR
3227          IF UN.COLOR(UNIT) = BLUE      ''%5MAY80_%JLR
3228             IF UN.X.COORD(UNIT) GT SS.REAR(SIDE,SECTOR)
3229                FILE UNIT IN THE SS.SET(SIDE,SECTOR)
3230             ENDIF
3231          ELSE   ''SIDE IS RED
3232             IF - UN.X.COORD(UNIT) GT SS.REAR(SIDE,SECTOR)
3233                FILE UNIT IN THE SS.SET(SIDE,SECTOR)
3234             ENDIF
3235          ALWAYS ''%5MAY80_%JLR
3236          ''LIST UNIT, UNIT.NOS(UNIT)
3237       ENDLOOP '' THRU EACH UNIT IN THE SET
3238    ENDLOOP  '' THRU EACH SIDE
3239
3240    ''CHECK TO INSURE THAT SECTOR SETS ARE NON-EMPTY
3241
```

```
3242     LOOP FOR EACH SIDE,
3243     DO THIS '' %18OCT79_%RGR                                                                                    \1
3244        LOOP FOR EACH SECTOR,
3245        DO  '' %18OCT79_%RGR                                                                                     \1
3246           IF SS.SET( SIDE,SECTOR )  IS EMPTY
3247              CALL RESET.FEBA.SECTOR  GIVEN SIDE AND SECTOR------------------------------------------------------>(119)
3248           ENDIF
3249        ENDLOOP
3250     ENDLOOP
3251
3252     IF ERROR.MSG = TRUE    ''  ADDED %7MAR80_%RGR
3253        SKIP 1 LINE
3254        PRINT 1 LINE THUS
3255        = =  DATA ERROR EXISTS IN THE UNIT DATA FILE - CORRECT  = =
3256        SKIP 1 LINE
3257     ENDIF
3258
3259 ENDROUTINE
```

```
3260 ROUTINE FILE.FD.SCHD  GIVEN FDC, NEW.START, NEW.STOP, AND NEW.BATS ''                T007
3261    ADD 1 TO ANAL.CTR(7,1)  ''                                                       \DYN_ANAL
3262    NORMALLY MODE IS INTEGER
3263    DEFINE FDC AS AN INTEGER VARIABLE
3264    '' IF DEBUG=TRUE,
3265       '' PRINT 1 LINE WITH FDC,NEW.START,NEW.STOP,NEW.BATS THUS
3266       '' = = = FILE.FD.SCHD FDC=***, START=****, STOP=****, BATS=**** = = =
3267    '' ENDIF
3268    LOOP FOR EACH ITEM IN FD.SCHD.LIST( FDC ),
3269    DO THIS ''                                                                       \1
3270       IF FS.STOP( ITEM ) < INT.F( TIME.V * 100. )
3271          REMOVE ITEM FROM FD.SCHD.LIST( FDC )
3272          DESTROY THE FD.SCHD.MSN CALLED ITEM
3273    <------CYCLE
3274       OTHERWISE
3275       IF NEW.START >= FS.STOP( ITEM )          ''CASE A
3276    <------CYCLE
3277       OTHERWISE
3278       LET OLD.START = FS.START( ITEM )
3279       LET OLD.STOP = FS.STOP( ITEM )
3280       LET OLD.BATS = FS.BATS( ITEM )
3281       IF NEW.STOP <= OLD.START                 ''CASE B
3282          CREATE A FD.SCHD.MSN CALLED NEW.ITEM
3283          LET FS.START( NEW.ITEM ) = NEW.START
3284          LET FS.STOP( NEW.ITEM ) = NEW.STOP
3285          LET FS.BATS( NEW.ITEM ) = NEW.BATS
3286          FILE NEW.ITEM IN FD.SCHD.LIST( FDC )
3287 <----------EXITROUTINE
3288       OTHERWISE
3289       IF NEW.START <= OLD.START
3290          IF NEW.STOP < OLD.STOP             ''CASE C
3291             LET FS.BATS( ITEM ) = OLD.BATS + NEW.BATS
3292             LET FS.STOP( ITEM ) = NEW.STOP
3293             CREATE A FD.SCHD.MSN CALLED NEW.ITEM1
3294             LET FS.START( NEW.ITEM1 ) = NEW.STOP
3295             LET FS.STOP( NEW.ITEM1 ) = OLD.STOP
3296             LET FS.BATS( NEW.ITEM1 ) = OLD.BATS
3297             FILE NEW.ITEM1 IN FD.SCHD.LIST( FDC )
3298             IF NEW.START - OLD.START = 0   ''CASE D
3299 <----------------EXITROUTINE
3300             OTHERWISE
3301             CREATE A FD.SCHD.MSN CALLED NEW.ITEM2
3302             LET FS.START( NEW.ITEM2 ) = NEW.START
3303             LET FS.STOP( NEW.ITEM2 ) = OLD.START
3304             LET FS.BATS( NEW.ITEM2 ) = NEW.BATS
3305             FILE NEW.ITEM2 IN FD.SCHD.LIST( FDC )
3306 <--------------EXITROUTINE
3307          OTHERWISE
3308          LET FS.BATS( ITEM ) = OLD.BATS + NEW.BATS
3309          IF NEW.START NE OLD.START
3310             ''CASE F OR G
3311             CREATE A FD.SCHD.MSN CALLED NEW.ITEM1
3312             LET FS.START( NEW.ITEM1 ) = NEW.START
3313             LET FS.STOP( NEW.ITEM1 ) = OLD.START
3314             LET FS.BATS( NEW.ITEM1 ) = NEW.BATS
3315             FILE NEW.ITEM1 IN FD.SCHD.LIST( FDC )
3316          ELSE
3317          ENDIF
```

```
3318          IF OLD.STOP -NEW.STOP = 0
3319             ''CASE E OR G
3320 <------------EXITROUTINE
3321          OTHERWISE
3322          ''CASE F OR H
3323          CALL FILE.FD.SCHD GIVEN FDC, OLD.STOP,------------------------------------------------------------------------------>( 63)
3324          NEW.STOP, AND NEW.BATS
3325 <----------EXITROUTINE
3326       OTHERWISE        ''NEW.START > OLD.STAR1
3327       IF NEW.STOP < OLD.STOP      ''CASE I
3328          LET FS.START( ITEM ) = NEW.START
3329          LET FS.STOP( ITEM ) = NEW.STOP
3330          LET FS.BATS( ITEM ) = OLD.BATS + NEW.BATS
3331          CREATE A FD.SCHD.MSN CALLED NEW.ITEM1
3332          CREATE A FD.SCHD.MSN CALLED NEW.ITEM2
3333          LET FS.START( NEW.ITEM1 ) = OLD.START
3334          LET FS.START( NEW.ITEM2 ) = NEW.STOP
3335          LET FS.STOP( NEW.ITEM1 ) = NEW.START
3336          LET FS.STOP( NEW.ITEM2 ) = OLD.STOP
3337          LET FS.BATS( NEW.ITEM1 ) = OLD.BATS
3338          LET FS.BATS( NEW.ITEM2 ) = OLD.BATS
3339          FILE NEW.ITEM1 IN FD.SCHD.LIST( FDC )
3340          FILE NEW.ITEM2 IN FD.SCHD.LIST( FDC )
3341 <----------EXITROUTINE
3342       OTHERWISE
3343       LET FS.START( ITEM ) = NEW.START
3344       LET FS.BATS( ITEM ) = OLD.BATS + NEW.BATS
3345       CREATE A FD.SCHD.MSN CALLED NEW.ITEM
3346       LET FS.START( NEW.ITEM ) = OLD.START
3347       LET FS.STOP( NEW.ITEM ) = NEW.START
3348       LET FS.BATS( NEW.ITEM ) = OLD.BATS
3349       FILE NEW.ITEM IN FD.SCHD.LIST( FDC )
3350       IF OLD.STOP - NEW.STOP = 0     ''CASE J
3351 <----------EXITROUTINE
3352       OTHERWISE      ''CASE K
3353       CALL FILE.FD.SCHD GIVEN FDC, OLD.STOP,------------------------------------------------------------------------------>( 63)
3354       NEW.STOP, NEW.BATS
3355 <--------EXITROUTINE
3356    ENDLOOP
3357 ENDROUTINE
```

```
3358 ROUTINE FILE.KAD.SENSOR                                                    T008
3359    GIVEN
3360    .AD.UNIT,
3361    .SENS.TYPE
3362    ADD 1 TO ANAL.CTR(8,1)  ''                                              \DYN_ANAL
3363
3364    NORMALLY MODE IS INTEGER
3365
3366    LET .SIDE = UN.COLOR(.AD.UNIT)
3367    FOR EVERY .KAS IN SD.KAS.SET(.SIDE)
3368    WITH KAS.AD.UNIT(.KAS) = .AD.UNIT
3369    FIND THE FIRST CASE
3370    IF FOUND
3371       ''THIS AD UNIT WAS ALREADY REPORTED.
3372 <------RETURN
3373    OTHERWISE
3374
3375    ''FILE THIS SENSOR IN THE LIST AND CREATE A NEW
3376    ''FLIGHT PATH OBSTACLE.
3377    CREATE A KNOWN.AD.SENSOR CALLED .KAS
3378    FILE .KAS IN THE SD.KAS.SET(.SIDE)
3379    LET KAS.AD.UNIT(.KAS) = .AD.UNIT
3380
3381    LET .RANGE = MRH.RANGE(L.MADS.RH.SET(.SENS.TYPE))
3382    CREATE AN FP.OBSTACLE CALLED .FP
3383    LET FPO.XMIN(.FP) = UN.X.COORD(.AD.UNIT) - .RANGE
3384    LET FPO.XMAX(.FP) = UN.X.COORD(.AD.UNIT) + .RANGE
3385    LET FPO.YMIN(.FP) = UN.Y.COORD(.AD.UNIT) - .RANGE
3386    LET FPO.YMAX(.FP) = UN.Y.COORD(.AD.UNIT) + .RANGE
3387
3388    ''WHEN THE NEW OBSTACLE OVERLAPS AN EXISTING ONE, MERGE THEM.
3389    LOOP UNTIL .FP IS IN THE SD.FPO.LIST
3390    DO
3391       FOR EVERY .OBS IN SD.FPO.LIST(.SIDE)
3392       WITH (FPO.XMIN(.FP) LE FPO.XMIN(.OBS) LE FPO.XMAX(.FP) OR
3393       FPO.XMIN(.FP)  LE FPO.XMAX(.OBS) LE FPO.XMAX(.FP) OR
3394       FPO.XMIN(.OBS) LE FPO.XMIN(.FP)  LE FPO.XMAX(.OBS) OR
3395       FPO.XMIN(.OBS) LE FPO.XMAX(.FP)  LE FPO.XMAX(.OBS)) AND
3396       (FPO.YMIN(.FP) LE FPO.YMIN(.OBS) LE FPO.YMAX(.FP) OR
3397       FPO.YMIN(.FP)  LE FPO.YMAX(.OBS) LE FPO.YMAX(.FP) OR
3398       FPO.YMIN(.OBS) LE FPO.YMIN(.FP)  LE FPO.YMAX(.OBS) OR
3399       FPO.YMIN(.OBS) LE FPO.YMAX(.FP)  LE FPO.YMAX(.OBS))
3400       FIND THE FIRST CASE
3401       IF NONE
3402          FILE .FP IN SD.FPO.LIST(.SIDE)
3403       ELSE
3404          ''MERGE THE POINTS.
3405          LET FPO.XMIN(.FP) = MIN.F(FPO.XMIN(.FP), FPO.XMIN(.OBS))
3406          LET FPO.XMAX(.FP) = MAX.F(FPO.XMAX(.FP), FPO.XMAX(.OBS))
3407          LET FPO.YMIN(.FP) = MIN.F(FPO.YMIN(.FP), FPO.YMIN(.OBS))
3408          LET FPO.YMAX(.FP) = MAX.F(FPO.YMAX(.FP), FPO.YMAX(.OBS))
3409          REMOVE .OBS FROM SD.FPO.LIST(.SIDE)
3410          DESTROY THE FP.OBSTACLE CALLED .OBS
3411       ALWAYS
3412    ENDLOOP
3413
3414 <--RETURN
3415 END
```

```
3416 ROUTINE FORM.TF.LIST(UNT.,UNIT.LIST.,.ERROR) ''                                                    T009
3417     ADD 1 TO ANAL.CTR(9,1)  ''                                                                    \DYN_ANAL
3418
3419     ''THIS ROUTINE PLACES ALL SUBORDINATE UNITS OF A
3420     ''  SPECIFIED UNIT IN THAT UNIT'S TASK FORCE LIST.
3421     NORMALLY MODE IS INTEGER
3422     DEFINE UNIT AS A VARIABLE
3423     LOOP FOR EVERY UNIT IN UN.SUB.LIST(UNT.)
3424     DO ''                                                                                              \1
3425        IF CT.GROUP(TU.CAT(UN.TYPE.UNIT(UNIT))) NE MANEUVER
3426           PRINT 1 LINE WITH UNIT.NOS(UNIT) THUS
3427           ERROR ##  SUBORDINATE UNIT ****** IS NOT A MANEUVER UNIT
3428           LET .ERROR = 1
3429        ELSE
3430           IF MU.TF.LIST(UN.PTR(UNIT)) IS NOT EMPTY
3431              OR MU.ORDER.SET(UN.PTR(UNIT)) IS NOT EMPTY
3432              OR MU.TF.MEM(UN.PTR(UNIT)) > 0
3433              PRINT 2 LINES WITH UNIT.NOS(UNIT), UNIT.NOS(UNT.) THUS
3434              ERROR # SUBORDINATE UNIT ***** ALREADY HAS ORDERS
3435              SO DOES PARENT UNIT *****
3436              LET .ERROR = 1
3437           ALWAYS
3438           FILE UN.PTR(UNIT) IN MU.TF.LIST(UN.PTR(UNIT.LIST.))
3439           LET MU.TF.MEM(UN.PTR(UNIT))=UNIT.LIST.
3440           LET MU.OFFSET.X(UN.PTR(UNIT))=UN.X.COORD(UNIT.LIST)-UN.X.COORD(UNIT)
3441           LET MU.OFFSET.Y(UN.PTR(UNIT))=UN.Y.COORD(UNIT.LIST)-UN.Y.COORD(UNIT)
3442           IF UN.SUB.LIST(UNIT) IS NOT EMPTY
3443              CALL FORM.TF.LIST(UNIT,UNIT.LIST.,.ERROR)------------------------------------------------------------------------------------>( 66)
3444           ENDIF
3445        ALWAYS
3446     ENDLOOP
3447 ENDROUTINE
```

```
3448 ROUTINE GENERAL.BATTLE                                                    T010
3449    GIVEN
3450    BLUE.UNITS,
3451    RED.UNITS
3452    ADD 1 TO ANAL.CTR(10,1)  ''                                            \DYN_ANAL
3453
3454    NORMALLY MODE IS INTEGER
3455    DEFINE BLUE.UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
3456    DEFINE RED.UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
3457    DEFINE UNIT AS AN INTEGER VARIABLE
3458    LET NO.BLUE.UNITS = DIM.F(BLUE.UNITS(*))
3459    LET NO.RED.UNITS = DIM.F(RED.UNITS(*))
3460    LOOP
3461    FOR EVERY TYPE.TEAM OF TEAM.TYPES(1)
3462    UNTIL TEAM.TYPES(1) IS EMPTY
3463    DO THE FOLLOWING
3464       LOOP
3465       FOR EVERY PATH.POINT OF PATH.SET(TYPE.TEAM)
3466       UNTIL PATH.SET(TYPE.TEAM) IS EMPTY
3467       DO THE FOLLOWING
3468          REMOVE THE FIRST PATH.POINT FROM PATH.SET(TYPE.TEAM)
3469          DESTROY THE PATH.POINT
3470       ENDLOOP
3471       REMOVE THE FIRST TYPE.TEAM FROM TEAM.TYPES(1)
3472       DESTROY THE TYPE.TEAM
3473    ENDLOOP
3474    IF TBF.BL.MISSION(1) = DEFEND
3475       OR TBF.BL.MISSION(1) = DELAY
3476       OR TBF.BL.MISSION(1) = AMBUSH
3477       LET A = DIS.WITH.DRAW
3478       IF TBF.RD.MISSION(1) = DEFEND
3479          OR TBF.RD.MISSION(1) = DELAY
3480          OR TBF.RD.MISSION(1) = AMBUSH
3481          LET B = DIS.WITH.DRAW
3482          LOOP
3483          FOR I = 1 TO NO.BLUE.UNITS
3484          DO THE FOLLOWING
3485             ADD UN.X.COORD(BLUE.UNITS(I)) TO X.BLUE
3486             ADD UN.Y.COORD(BLUE.UNITS(I)) TO Y.BLUE
3487          ENDLOOP
3488          LET X.BLUE = X.BLUE/NO.BLUE.UNITS
3489          LET Y.BLUE = Y.BLUE/NO.BLUE.UNITS
3490          LOOP
3491          FOR I = 1 TO NO.RED.UNITS
3492          DO THE FOLLOWING
3493             ADD UN.X.COORD(RED.UNITS(I)) TO X.RED
3494             ADD UN.Y.COORD(RED.UNITS(I)) TO Y.RED
3495          ENDLOOP
3496          LET X.RED = X.RED/NO.RED.UNITS
3497          LET Y.RED = Y.RED/NO.RED.UNITS
3498          LET D.X = X.RED - X.BLUE
3499          LET D.Y = Y.RED - Y.BLUE
3500          LET R = (D.X**2 + D.Y**2)**.5
3501          LET C = B + R
3502       ELSE
3503          LET B = DIS.ATTACK
3504          LET C = DIS.WITH.DRAW
3505       ALWAYS
```

```
3506     ELSE
3507        LET A = DIS.ATTACK
3508        IF TBF.RD.MISSION(1) = DEFEND
3509           OR TBF.RD.MISSION(1) = DELAY
3510           OR TBF.RD.MISSION(1) = AMBUSH
3511           LET B = DIS.WITH.DRAW
3512           LET C = DIS.ATTACK
3513        ELSE
3514           LET B = DIS.ATTACK
3515           LET C = DIS.WITH.DRAW
3516        ALWAYS
3517     ALWAYS
3518     LOOP FOR I = 1 TO NO.BLUE.UNITS
3519     DO THIS ''                                                    \1'
3520        FILE BLUE.UNITS(I) IN TB.SORT.LIST(1)
3521     ENDLOOP
3522     LOOP
3523     FOR I = 1 TO NO.BLUE.UNITS
3524     DO THE FOLLOWING
3525        REMOVE THE FIRST UNIT FROM TB.SORT.LIST(1)
3526        LET BLUE.UNITS(I) = UNIT
3527        CREATE A TYPE.TEAM
3528        LET TT.IN.USE(TYPE.TEAM) = NO
3529        LET TT.SIDE(TYPE.TEAM) = BLUE
3530        LET TT.TYPE(TYPE.TEAM) = UN.TYPE.UNIT(BLUE.UNITS(I))
3531        LET TT.UNIT(TYPE.TEAM) = BLUE.UNITS(I)
3532        FILE THIS TYPE.TEAM IN TEAM.TYPES(1)
3533        CREATE A PATH.POINT
3534        IF UN.SUB.LIST(BLUE.UNITS(I)) IS NOT EMPTY
3535           LET PP.X.POINT(PATH.POINT) = 0
3536        ELSE
3537           LET PP.X.POINT(PATH.POINT) = HQ.SET.BACK
3538        ALWAYS
3539        FILE THIS PATH.POINT IN PATH.SET(TYPE.TEAM)
3540        CREATE A PATH.POINT
3541        IF UN.SUB.LIST(BLUE.UNITS(I)) IS NOT EMPTY
3542           LET PP.X.POINT(PATH.POINT) = A
3543           ADD 1 TO NO.BLUE.HQ
3544        ELSE
3545           LET PP.X.POINT(PATH.POINT) = HQ.SET.BACK + A
3546           ADD 1 TO NO.BLUE.MAN
3547        ALWAYS
3548        FILE THIS PATH.POINT IN PATH.SET(TYPE.TEAM)
3549     ENDLOOP
3550     LOOP FOR I = 1 TO NO.RED.UNITS
3551     DO THIS ''                                                    \1
3552        FILE RED.UNITS(I) IN TB.SORT.LIST(1)
3553     ENDLOOP
3554     LOOP
3555     FOR I = 1 TO NO.RED.UNITS
3556     DO THE FOLLOWING
3557        REMOVE THE FIRST UNIT FROM TB.SORT.LIST(1)
3558        LET RED.UNITS(I) = UNIT
3559        CREATE A TYPE.TEAM
3560        LET TT.IN.USE(TYPE.TEAM) = NO
3561        LET TT.SIDE(TYPE.TEAM) = RED
3562        LET TT.TYPE(TYPE.TEAM) = UN.TYPE.UNIT(RED.UNITS(I))
3563        LET TT.UNIT(TYPE.TEAM) = RED.UNITS(I)
```

```
3564        FILE THIS TYPE.TEAM IN TEAM.TYPES(1)
3565        CREATE A PATH.POINT
3566        IF UN.SUB.LIST(RED.UNITS(I)) IS NOT EMPTY
3567           LET PP.X.POINT(PATH.POINT) = 2*HQ.SET.BACK + C + B
3568        ELSE
3569           LET PP.X.POINT(PATH.POINT) = HQ.SET.BACK + C + B
3570        ALWAYS
3571        FILE THIS PATH.POINT IN PATH.SET(TYPE.TEAM)
3572        CREATE A PATH.POINT
3573        IF UN.SUB.LIST(RED.UNITS(I)) IS NOT EMPTY
3574           LET PP.X.POINT(PATH.POINT) = 2*HQ.SET.BACK + C
3575           ADD 1 TO NO.RED.HQ
3576        ELSE
3577           LET PP.X.POINT(PATH.POINT) = HQ.SET.BACK + C
3578           ADD 1 TO NO.RED.MAN
3579        ALWAYS
3580        FILE THIS PATH.POINT IN PATH.SET(TYPE.TEAM)
3581     ENDLOOP
3582     LET B.HQ.W = (NO.BLUE.HQ +1)*B.WIDTH.UNIT
3583     LET B.MAN.W = (NO.BLUE.MAN + 1) * B.WIDTH.UNIT
3584     LET R.HQ.W = (NO.RED.HQ + 1) * R.WIDTH.UNIT
3585     LET R.MAN.W = (NO.RED.MAN + 1) * R.WIDTH.UNIT
3586     LET TBF.WIDTH(1) = MAX.F(B.HQ.W, B.MAN.W, R.HQ.W, R.MAN.W)
3587     LET D.B.H = TBF.WIDTH(1) / (NO.BLUE.HQ + 1)
3588     LET B.HQ.INC = D.B.H
3589     LET D.B.M = TBF.WIDTH(1) / (NO.BLUE.MAN + 1)
3590     LET B.MAN.INC = D.B.M
3591     LET D.R.H = TBF.WIDTH(1) / (NO.RED.HQ + 1)
3592     LET R.HQ.INC = D.R.H
3593     LET D.R.M = TBF.WIDTH(1) / (NO.RED.MAN + 1)
3594     LET R.MAN.INC = D.R.M
3595     LOOP
3596     FOR EVERY TYPE.TEAM OF TEAM.TYPES(1)
3597     DO THE FOLLOWING
3598        IF TT.SIDE(TYPE.TEAM) = BLUE
3599           IF UN.SUB.LIST(TT.UNIT(TYPE.TEAM)) IS NOT EMPTY
3600              LOOP
3601              FOR EVERY PATH.POINT OF PATH.SET(TYPE.TEAM)
3602              DO THE FOLLOWING
3603                 LET PP.Y.POINT(PATH.POINT) = D.B.H
3604              ENDLOOP
3605              ADD B.HQ.INC TO D.B.H
3606           ELSE
3607              LOOP
3608              FOR EVERY PATH.POINT OF PATH.SET(TYPE.TEAM)
3609              DO THE FOLLOWING
3610                 LET PP.Y.POINT(PATH.POINT) = D.B.M
3611              ENDLOOP
3612              ADD B.MAN.INC TO D.B.M
3613           ALWAYS
3614        ELSE
3615           IF UN.SUB.LIST(TT.UNIT(TYPE.TEAM)) IS NOT EMPTY
3616              LOOP
3617              FOR EVERY PATH.POINT OF PATH.SET(TYPE.TEAM)
3618              DO THE FOLLOWING
3619                 LET PP.Y.POINT(PATH.POINT) = D.R.H
3620              ENDLOOP
3621              ADD R.HQ.INC TO D.R.H
```

```
3622           ELSE
3623              LOOP
3624              FOR EVERY PATH.POINT OF PATH.SET(TYPE.TEAM)
3625              DO THE FOLLOWING
3626                 LET PP.Y.POINT(PATH.POINT) = D.R.M
3627              ENDLOOP
3628              ADD R.MAN.INC TO D.R.M
3629           ALWAYS
3630        ALWAYS
3631     ENDLOOP
3632 <--EXITROUTINE
3633 ENDROUTINE
```

```
3634 ROUTINE ORIENTATION GIVEN   ''                                          %15MAR79_%JN \T011
3635    BATTLE.
3636    YIELDING
3637    RANGE.RATIO,
3638    X.BLUE,
3639    Y.BLUE,
3640    X.RED,
3641    Y.RED,
3642    X,
3643    Y,
3644    AND THETA
3645    ADD 1 TO ANAL.CTR(11,1)  ''                                                \DYN_ANAL
3646    ''ORIENTATION
3647    DEFINE A,THETA,THETA.ONE,THETA.TWO,THETA.THREE,THETA.FOUR,
3648    ANGLE,D,R,RANGE,TYPE.RANGE,RANGE.RATIO AS REAL VARIABLES ''%26MAR78_%LM
3649    NORMALLY MODE IS INTEGER
3650    ''THIS ROUTINE COMPUTES THE TRANSFORMATION FROM THE
3651    ''PRESTORED BATTLE TO THE ACTUAL BATTLE.SPECIFICALLY IT:
3652    ''(A) COMPUTES THE ROTATION ANGLE OF THE PRESTORED
3653    ''BATTLE TO THE ACTUAL BATTLE
3654    ''(B) COMPUTES THE SHIFT OF THE ORIGIN OF THE PRESTORED
3655    ''BATTLE TO THE ACTUAL BATTLE
3656    ''(C) COMPUTES THE RATIO OF THE ACTUAL RANGE TO THE
3657    ''PRESTORED RANGE WHICH IS USED TO ADJUST THE
3658    ''OPENING RANGE
3659    ''(D) COMPUTES THE CENTERS OF THE FORCES
3660    ''
3661    ''FIND THE ACTUAL CENTER OF THE BLUE UNITS AND
3662    ''STORE IT IN X.BLUE AND Y.BLUE
3663    ''
3664    LOOP
3665    FOR EVERY FORCE OF BTL.FORCE.SET(BATTLE.)
3666    DO THE FOLLOWING
3667       LOOP
3668       FOR EVERY UNIT. OF FR.UNIT.SET(FORCE.)
3669       DO THE FOLLOWING
3670          IF UN.COLOR(UNIT.) = BLUE
3671             ADD UN.X.COORD(UNIT.) TO X.BLUE
3672             ADD UN.Y.COORD(UNIT.) TO Y.BLUE
3673             ADD 1 TO NO.BLUE.UNITS
3674          ELSE
3675             ADD UN.X.COORD(UNIT.) TO X.RED
3676             ADD UN.Y.COORD(UNIT.) TO Y.RED
3677             ADD 1 TO NO.RED.UNITS
3678          ALWAYS
3679       ENDLOOP
3680    ENDLOOP
3681    LET X.BLUE = X.BLUE / NO.BLUE.UNITS
3682    LET Y.BLUE = Y.BLUE / NO.BLUE.UNITS
3683    LET X.RED = X.RED / NO.RED.UNITS
3684    LET Y.RED = Y.RED / NO.RED.UNITS
3685    ''
3686    ''RANGE IS THE CENTER TO CENTER RANGE OF THE TWO FORCES
3687    ''
3688    LET RANGE = ((Y.RED - Y.BLUE)**2 + (X.RED - X.BLUE)**2)**.5  ''             \OPTIMIZE
3689    ''
3690    ''FIND THE CENTER OF THE UNITS IN THE
3691    ''PRESTORED BATTLE AND STORE THEM IN
```

```
3692    ''XX.BLUE, XX.RED, YY.BLUE, AND YY.RED
3693    ''
3694    LOOP
3695    FOR EVERY TYPE.TEAM OF TEAM.TYPES(BTL.FIELD(BATTLE.))
3696    DO
3697       IF TT.SIDE(TYPE.TEAM) = BLUE ,
3698          IF TBF.BL.MISSION(BTL.FIELD(BATTLE.)) = ATTACK
3699             OR TBF.BL.MISSION(BTL.FIELD(BATTLE.)) = PATROL
3700             OR TBF.BL.MISSION(BTL.FIELD(BATTLE.)) = PROBE,
3701             LET XX.BLUE = XX.BLUE + PP.X.POINT(F.PATH.SET(TYPE.TEAM))
3702             LET YY.BLUE = YY.BLUE + PP.Y.POINT(F.PATH.SET(TYPE.TEAM))
3703          ELSE
3704             LET XX.BLUE = XX.BLUE + PP.X.POINT(L.PATH.SET(TYPE.TEAM))
3705             LET YY.BLUE = YY.BLUE + PP.Y.POINT(L.PATH.SET(TYPE.TEAM))
3706          ALWAYS
3707       ELSE
3708          IF TBF.RD.MISSION(BTL.FIELD(BATTLE.)) = ATTACK
3709             OR TBF.RD.MISSION(BTL.FIELD(BATTLE.)) = PATROL
3710             OR TBF.RD.MISSION(BTL.FIELD(BATTLE.)) = PROBE,
3711             LET XX.RED = XX.RED + PP.X.POINT(F.PATH.SET(TYPE.TEAM))
3712             LET YY.RED = YY.RED + PP.Y.POINT(F.PATH.SET(TYPE.TEAM))
3713          ELSE
3714             LET XX.RED = XX.RED + PP.X.POINT(L.PATH.SET(TYPE.TEAM))
3715             LET YY.RED = YY.RED + PP.Y.POINT(L.PATH.SET(TYPE.TEAM))
3716          ALWAYS
3717       ALWAYS
3718    ENDLOOP
3719    LET XX.BLUE = XX.BLUE / NO.BLUE.UNITS
3720    LET YY.BLUE = YY.BLUE / NO.BLUE.UNITS
3721    LET XX.RED = XX.RED / NO.RED.UNITS
3722    LET YY.RED = YY.RED / NO.RED.UNITS
3723    ''
3724    ''TYPE.RANGE IS THE CENTER TO CENTER RANGE
3725    ''IN THE PRESTORED BATTLE
3726    ''
3727    LET TYPE.RANGE = ((YY.RED - YY.BLUE)**2 + (XX.RED - XX.BLUE)**2)**.5
3728    '' %15MAR79
3729    IF TYPE.RANGE = 0  '' %15MAR
3730       LET TYPE.RANGE = RANGE '' CENTER TO CENTER DIS = RANGE %15MAR
3731    ENDIF
3732    ''
3733     'COMPUTE THE SLOPES OF THE CONNECTING LINES
3734    ''
3735    LET DY = Y.RED - Y.BLUE
3736    LET DX = X.RED - X.BLUE
3737    LET DYY = YY.RED - YY.BLUE
3738    LET DXX = XX.RED - XX.BLUE
3739    CALL ANGLE.COMPUTE----------------------------------------------------------------------------------------------------->(644)
3740    GIVEN
3741    DX,
3742    DY
3743    YIELDING
3744    THETA.ONE
3745    CALL ANGLE.COMPUTE----------------------------------------------------------------------------------------------------->(644)
3746    GIVEN
3747    DXX,
3748    DYY
3749    YIELDING
```

```
3750    THETA.TWO
3751    LET THETA = THETA.ONE. - THETA.TWO.
3752    ''
3753    IF TBF.BL.MISSION(BTL.FIELD(BATTLE.)) = ATTACK
3754       OR TBF.BL.MISSION(BTL.FIELD(BATTLE.)) = PATROL
3755       OR TBF.BL.MISSION(BTL.FIELD(BATTLE.)) = PROBE,
3756       IF TBF.RD.MISSION(BTL.FIELD(BATTLE.)) = ATTACK
3757          OR TBF.RD.MISSION(BTL.FIELD(BATTLE.)) = PATROL
3758          OR TBF.RD.MISSION(BTL.FIELD(BATTLE.)) = PROBE,
3759          LET DYYY = .5*(YY.BLUE + YY.RED)
3760          LET DXXX = .5*(XX.BLUE + XX.RED)
3761          LET X = .5*(X.BLUE + X.RED)
3762          LET Y = .5*(Y.BLUE + Y.RED)
3763       ELSE
3764          LET DYYY = YY.RED
3765          LET DXXX = XX.RED
3766          LET X = X.RED
3767          LET Y = Y.RED
3768       ALWAYS
3769    ELSE
3770       IF TBF.RD.MISSION(BTL.FIELD(BATTLE.)) = ATTACK
3771          OR TBF.RD.MISSION(BTL.FIELD(BATTLE.)) = PATROL
3772          OR TBF.RD.MISSION(BTL.FIELD(BATTLE.)) = PROBE,
3773          LET DYYY = YY.BLUE
3774          LET DXXX = XX.BLUE
3775          LET X = X.BLUE
3776          LET Y = Y.BLUE
3777       ELSE
3778          LET DYYY = .5*(YY.BLUE + YY.RED)
3779          LET DXXX = .5*(XX.BLUE + XX.RED)
3780          LET X = .5*(X.BLUE + X.RED)
3781          LET Y = .5*(Y.BLUE + Y.RED)
3782       ALWAYS
3783    ALWAYS
3784    CALL ANGLE.COMPUTE----------------------------------------------------------------------------------------->(644)
3785    GIVEN
3786    DXXX,
3787    DYYY
3788    YIELDING
3789    THETA.THREE
3790    LET D = (DXXX**2 + DYYY**2)**.5
3791    LET A = PI.C - THETA.THREE - THETA
3792    ''X AND Y ARE THE COORDINATES OF THE TRANSLATED ORIGIN
3793    ''
3794    LET X = X + D * COS.F(A)
3795    LET Y = Y - D * SIN.F(A)
3796    LET RANGE.RATIO = RANGE / TYPE.RANGE
3797    ''
3798    ''PLACE THE CENTER OF THE FORCES IN                                                                          \1
3799    ''X.BLUE, Y.BLUE, X.RED, AND Y.RED
3800    ''
3801    CALL ANGLE.COMPUTE----------------------------------------------------------------------------------------->(644)
3802    GIVEN
3803    XX.BLUE,
3804    YY.BLUE
3805    YIELDING
3806    THETA.FOUR
3807    LET ANGLE = THETA + THETA.FOUR
```

```
3808     LET R = (XX.BLUE**2 + YY.BLUE**2)**.5
3809     LET X.BLUE = X + R * COS.F(ANGLE)
3810     LET Y.BLUE = Y + R * SIN.F(ANGLE)
3811     CALL ANGLE.COMPUTE------------------------------------------------------------------------------------------------------>(644)
3812     GIVEN
3813     XX.RED,
3814     YY.RED
3815     YIELDING
3816     THETA.FOUR
3817     LET ANGLE = THETA + THETA.FOUR
3818     LET R = (XX.RED**2 + YY.RED**2)**.5
3819     LET X.RED = X + R * COS.F(ANGLE)
3820     LET Y.RED = Y + R * SIN.F(ANGLE)
3821 <--EXITROUTINE
3822 ENDROUTINE
```

```
3823 ROUTINE UNIT.ASSIGNMENT                                                                                    T012
3824    GIVEN
3825    SIDE,
3826    UNITS,
3827    TYPE.BATTLE.FIELD,
3828    THETA,
3829    X,
3830    Y
3831    ADD 1 TO ANAL.CTR(12,1)                                                                                 \DYN_ANAL
3832
3833    NORMALLY MODE IS INTEGER
3834    DEFINE ANGLE, THETA, THETA.ONE AS REAL VARIABLES
3835    DEFINE UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
3836
3837    LET NO.UNITS = DIM.F(UNITS(*))
3838    LOOP FOR I = 1 TO NO.UNITS
3839    UNLESS TEAM.TYPES(TYPE.BATTLE.FIELD) IS EMPTY
3840    DO THE FOLLOWING
3841       FOR EVERY TT OF TEAM.TYPES(TYPE.BATTLE.FIELD)
3842       WITH TT.IN.USE(TT) = NO  AND
3843       TT.SIDE(TT) = SIDE  AND
3844       TT.TYPE(TT) = UN.TYPE.UNIT(UNITS(I))
3845       FIND THE FIRST CASE
3846       IF NONE
3847          LIST ATTRIBUTES OF TYPE.TEAM CALLED TT
3848          LIST UN.TYPE.UNIT(UNITS(I)), SIDE, I, UNITS(I)
3849          CALL ERROR.STOP------------------------------------------------------------------------------------>(604)
3850       ALWAYS
3851
3852       LET TT.IN.USE(TT) = YES
3853       LOOP FOR EVERY PP OF PATH.SET(TT)
3854       UNLESS PATH.SET(TT) IS EMPTY
3855       DO THE FOLLOWING
3856          CREATE A POINT
3857          CALL ANGLE.COMPUTE--------------------------------------------------------------------------------->(644)
3858          GIVEN
3859          PP.X.POINT(PP),
3860          PP.Y.POINT(PP)
3861          YIELDING
3862          THETA.ONE
3863          LET ANGLE = THETA + THETA.ONE
3864          LET R = (PP.Y.POINT(PP)**2 + PP.X.POINT(PP)**2)**.5                                               \OPTIMIZE
3865          LET P.X(POINT)= X + R * COS.F(ANGLE)
3866          LET P.Y(POINT)  = Y + R * SIN.F(ANGLE)
3867          FILE THE POINT IN THE UN.PATH(UNITS(I))
3868       ENDLOOP
3869    ENDLOOP
3870
3871 <--EXITROUTINE
3872 ENDROUTINE
```

MOVEMENT/TERRAIN ROUTINES

```
3879 ROUTINE ADJUST                                                    ''          M001
3880    GIVEN
3881    UNITS,
3882    X.0,
3883    Y.0,
3884    RANGE.RATIO
3885    ADD 1 TO ANAL.CTR(13,1)  ''                                              \DYN_ANAL
3886
3887    ''THIS ROUTINE ADJUSTS THE OPENING RANGES AT THE START OF A BATTLE.
3888    ''THE ARRAY CALLED UNITS CONTAINS THE UNITS THAT ARE TO BE MOVED.
3889    ''THE POINT, (X.0,Y.0), IS TH CENTER OF THE OPPOSING FORCE AND IS
3890    ''THE BASE POINT.RANGE.RATIO IS THE FRACTION THAT THE RANGES
3891    ''ARE REDUCED.
3892
3893    NORMALLY MODE IS REAL
3894    DEFINE X.0, Y.0, I, NO.UNITS, POINT AS INTEGER VARIABLES
3895    DEFINE UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
3896
3897    LET NO.UNITS = DIM.F(UNITS(*))
3898
3899    ''THE ROUTINE ONLY REDUCES THE RANGE—NO ATTEMPT IS MADE TO
3900    ''INCREASE THE RANGES.
3901
3902    IF RANGE.RATIO > 1.0,
3903       LIST RANGE.RATIO
3904 <———EXITROUTINE
3905    OTHERWISE
3906
3907    LOOP FOR I = 1 TO NO.UNITS
3908    DO
3909
3910       ''RANGE IS THE DESIRED OPENING RANGE FOR EACH UNIT.
3911
3912       LET RANGE = (((X.0 - UN.X.COORD(UNITS(I)))**2 + ''                    \OPTIMIZE
3913       (Y.0 - UN.Y.COORD(UNITS(I)))**2)**.5)*RANGE.RATIO
3914       LOOP FOR EVERY POINT OF UN.PATH(UNITS(I))
3915       DO
3916          ''DISTANCE IS HOW FAR EACH POINT ALONG THE PATH
3917          ''IS FROM THE BASE POINT
3918
3919          LET DISTANCE = ((X.0 - P.X(POINT))**2 +
3920          (Y.0 - P.Y(POINT))**2)**.5
3921
3922          ''THE ROUTINE SEARCHES EACH POINT ALONG THE PATH AND COMPARES
3923          ''THE DISTANCE TO THE RANGE.
3924          ''(1)IF DISTANCE = RANGE, THE UNIT IS PLACED AT THAT POINT.
3925
3926          ''(2)IF DISTANCE IS LESS THAN RANGE, IT INFERS THAT THE
3927          ''PREVIOUS POINT WAS MORE THAN THE RANGE AND THE UNIT SHOULD
3928          ''BE PLACED BETWEEN THE TWO.
3929
3930          ''(3)IF DISTANCE IS GREATER THAN RANGE, THE ROUTINE CYCLES
3931          ''TO THE NEXT POINT ON THE PATH.
3932
3933          ''(4) IF THE PATH ENDS BEFORE (1) OR (2) OCCURS, THE UNIT IS
3934          ''PLACED ON THE LAST PATH POINT.
3935
3936          IF DISTANCE EQ RANGE,
```

```
3937            LET UN.X.COORD(UNITS(I)) = P.X(POINT)
3938            LET UN.Y.COORD(UNITS(I)) = P.Y(POINT)
3939            IF UN.POSITION.INDEX(UNITS(I)) GE N.UN.PATH(UNITS(I)),
3940               LET UN.STATUS(UNITS(I)) = STATIONARY
3941            ALWAYS
3942         <------LEAVE
3943         OTHERWISE
3944
3945         IF DISTANCE LT RANGE,
3946            LET Y.1 = P.Y(POINT)
3947            LET X.1 = P.X(POINT)
3948            LET Y.2 = P.Y(P.UN.PATH(POINT))
3949            LET X.2 = P.X(P.UN.PATH(POINT))
3950            LET DELTA.Y = Y.1 - Y.2
3951            LET DELTA.X = X.1 - X.2
3952            IF DELTA.X EQ 0,
3953               LET UN.X.COORD(UNITS(I))=X.1
3954               LET X.3 = X.2 - X.0
3955               LET UN.Y.COORD(UNITS(I)) = Y.0
3956               - SQRT.F(RANGE**2 - X.3**2)
3957         <----------EXITLOOP
3958            OTHERWISE
3959
3960            LET SLOPE = DELTA.Y / DELTA.X
3961            LET Y.3 = Y.2 - SLOPE*X.2 - Y.0
3962            LET A = 1 + SLOPE**2
3963            LET B = 2*Y.3*SLOPE - 2*X.0
3964            LET C = X.0**2 + Y.3**2 - RANGE**2
3965            LET T = B**2 - 4*A*C
3966            IF T LT 0,
3967               LET UN.X.COORD(UNITS(I)) = X.1
3968               LET X.3 = X.2 - X.0
3969               LET UN.Y.COORD(UNITS(I)) = Y.0
3970               - SQRT.F(RANGE**2 - X.3**2)
3971               SUBTRACT 1 FROM UN.POSITION.INDEX(UNITS(I))
3972         <----------LEAVE
3973            OTHERWISE
3974
3975            LET R.T = SQRT.F(T)
3976            LET X = (-B + R.T)/(2*A)
3977            LET Y = SLOPE*(X-X.2) + Y.2
3978            IF X LT MIN.F(X.1,X.2) OR
3979               X GT MAX.F(X.1,X.2) OR
3980               Y LT MIN.F(Y.1,Y.2) OR
3981               Y GT MAX.F(Y.1,Y.2),
3982               LET X = (-B - R.T)/(2*A)
3983               LET Y = SLOPE*(X - X.2) + Y.2
3984            ALWAYS
3985            LET UN.X.COORD(UNITS(I)) = X
3986            LET UN.Y.COORD(UNITS(I)) = Y
3987            SUBTRACT 1 FROM UN.POSITION.INDEX(UNITS(I))
3988         <------LEAVE
3989         OTHERWISE
3990
3991         ''THE DISTANCE IS GREATER THAN THE RANGE,
3992         IF UN.POSITION.INDEX(UNITS(I)) GE N.UN.PATH(UNITS(I)),
3993            LET UN.X.COORD(UNITS(I)) = P.X(POINT)
3994            LET UN.Y.COORD(UNITS(I)) = P.Y(POINT)
```

```
3995              LET UN.STATUS(UNITS(I)) = STATIONARY
3996          <-----LEAVE
3997             OTHERWISE
3998
3999             ADD 1 TO UN.POSITION.INDEX(UNITS(I))
4000          ENDLOOP
4001       ENDLOOP
4002
4003 <--EXITROUTINE
4004 ENDROUTINE
```

```
4005 ROUTINE BLOCK.LOS                                                    M002
4006     GIVEN
4007     .UNIT,
4008     .ENEMY,
4009     .DURATION    ''MINUTES
4010     ADD 1 TO ANAL.CTR(14,1)  ''                                      \DYN_ANAL
4011
4012     ''THIS ROUTINE IS CALLED FROM SMOKE EFFECTS AND
4013     ''DUST EFFECTS WHEN LOS IS BROKEN
4014
4015     NORMALLY MODE IS INTEGER
4016
4017     FOR EVERY .SEG OF UN.SEGMENT.LIST(.UNIT)
4018     WITH SEG.UNIT(.SEG) = .ENEMY
4019     FIND THE FIRST CASE
4020     IF FOUND
4021        LET .MOVE.RATE = TU.MOV.RATE(UN.TYPE.UNIT(.UNIT))
4022        * TAC.MOV.FAC
4023        * MOV.FAC(BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(.UNIT)))
4024        IF TIME.V LT BMNT OR TIME.V GT EENT
4025           LET .MOVE.RATE = .MOVE.RATE * NITE.MOV.FAC
4026        ALWAYS
4027        IF UN.STATUS(.UNIT) = WITHDRAWING
4028           LET .MOVE.RATE = 3 * TU.MOV.RATE(UN.TYPE.UNIT(.UNIT))
4029        ALWAYS
4030        LET .NLOS = .MOVE.RATE * .DURATION /
4031        MINUTES.V
4032        LET .NLOS = MAX.F(.NLOS, 1)
4033        IF SEG.TYPE(.SEG)= NO
4034           LET SEG.LENGTH(.SEG) = MAX.F(SEG.LENGTH(.SEG), .NLOS)
4035        ELSE
4036           LET SEG.TYPE(.SEG) = NO
4037           LET SEG.LENGTH(.SEG)= .NLOS
4038        ALWAYS
4039     ALWAYS
4040
4041     FOR EVERY .SEG OF UN.SEGMENT.LIST(.ENEMY)
4042     WITH SEG.UNIT(.SEG) = .UNIT
4043     FIND THE FIRST CASE
4044     IF FOUND
4045        LET .NLOS = TU.MOV.RATE(UN.TYPE.UNIT(.ENEMY)) * .DURATION /
4046        MINUTES.V
4047        LET .NLOS = MAX.F(.NLOS, 1)
4048        IF SEG.TYPE(.SEG) = NO
4049           LET SEG.LENGTH(.SEG) = MAX.F(SEG.LENGTH(.SEG), .NLOS)
4050        ELSE
4051           LET SEG.TYPE(.SEG) = NO
4052           LET SEG.LENGTH(.SEG) = .NLOS
4053        ALWAYS
4054     ALWAYS
4055
4056     FOR EVERY .VU OF UN.LOS.LIST(.UNIT)
4057     WITH VU.POINTER(.VU) = .ENEMY
4058     FIND THE FIRST CASE
4059     IF FOUND
4060        REMOVE .VU FROM THE UN.LOS.LIST(.UNIT)
4061        DESTROY THE VISIBLE.UNIT CALLED .VU
4062        FOR EVERY .VU IN UN.LOS.LIST(.ENEMY)
```

```
4063        WITH VU.POINTER(.VU) = .UNIT
4064        FIND THE FIRST CASE
4065        IF FOUND
4066           REMOVE THE .VU FROM THE UN.LOS.LIST(.ENEMY)
4067           DESTROY THE VISIBLE.UNIT CALLED .VU
4068        ALWAYS
4069
4070        LOOP FOR EVERY .ENG OF EV.S(I.ENGAGEMENT)
4071        WITH (DETECTED.UNIT(.ENG) = .UNIT AND
4072        DETECTING.UNIT(.ENG) = .ENEMY) OR
4073        (DETECTED.UNIT(.ENG) = .ENEMY AND
4074        DETECTING.UNIT(.ENG) = .UNIT)
4075        DO
4076           CANCEL_THE ENGAGEMENT CALLED .ENG------------------------------------------------------------------------------>(369)
4077           DESTROY THE ENGAGEMENT CALLED .ENG
4078        ENDLOOP
4079
4080        LOOP FOR EACH .ASSESS OF EV.S(I.ASSESSMENT)
4081        WITH (AS.TGT.UNIT(.ASSESS) = .UNIT AND
4082        AS.FIRING.UNIT(.ASSESS) = .ENEMY) OR
4083        (AS.TGT.UNIT(.ASSESS)= .ENEMY AND
4084        AS.FIRING.UNIT(.ASSESS) = .UNIT)
4085        DO
4086           IF AS.DESTRUCT.INDIC(.ASSESS) NE YES
4087              INTERRUPT ASSESSMENT CALLED .ASSESS------------------------------------------------------------------------->(487)
4088              LET AS.DESTRUCT.INDIC(.ASSESS) = YES
4089              REACTIVATE_THE ASSESSMENT CALLED .ASSESS NOW---------------------------------------------------------------->(487)
4090           ALWAYS
4091        ENDLOOP
4092
4093        LOOP FOR EVERY .UE.LINK OF UN.EQUIP.LIST(.UNIT)
4094        DO
4095           LOOP FOR EVERY .FT OF UE.TARGET.LIST(.UE.LINK)
4096           DO
4097              LOOP FOR EVERY .SO OF SO.LIST(.UE.LINK)
4098              WITH SO.FIRING.TABLE(.SO) = .FT
4099              DO
4100                 LET SO.FIRING.TABLE(.SO) = 0
4101              ENDLOOP
4102              REMOVE THE .FT FROM THE UE.TARGET.LIST(.UE.LINK)
4103              DESTROY THE FIRING.TABLE CALLED .FT
4104           ENDLOOP
4105        ENDLOOP
4106
4107        LOOP FOR EVERY .UE.LINK OF UN.EQUIP.LIST(.ENEMY)
4108        DO
4109           LOOP FOR EVERY .FT OF UE.TARGET.LIST(.UE.LINK)
4110           DO
4111              LOOP FOR EVERY .SO OF SO.LIST(.UE.LINK)
4112              WITH SO.FIRING.TABLE(.SO) = .FT
4113              DO
4114                 LET SO.FIRING.TABLE(.SO) = 0
4115              ENDLOOP
4116              REMOVE THE .FT FROM THE UE.TARGET.LIST(.UE.LINK)
4117              DESTROY THE FIRING.TABLE CALLED .FT
4118           ENDLOOP
4119        ENDLOOP
4120     ALWAYS
```

```
4121
4122 <--EXITROUTINE
4123 ENDROUTINE
```

```
4124 ROUTINE CHANGE.LOC                                                                                    M003
4125    GIVEN
4126    UNIT
4127    ADD 1 TO ANAL.CTR(15,1)  ''                                                                        \DYN_ANAL
4128
4129    ''CHANGE$LOC CALLED FROM BTRY.EFFECTS AND MOVE
4130
4131    NORMALLY MODE IS INTEGER
4132    DEFINE POINT,PNT AS INTEGER VARIABLES
4133    DEFINE DELTA.TIME, MOVE.RATE AS REAL VARIABLES
4134
4135    LET DELTA.TIME. = TIME.V - UN.TIME.LAST.MOVE(UNIT)
4136    IF UN.DELAY(UNIT) GT 0
4137       ''A MINEFIELD HAS DELAYED THIS UNIT.
4138       IF DELTA.TIME GT UN.DELAY(UNIT)/MINUTES.V
4139          ''SOME MOVEMENT WILL OCCUR
4140          SUBTRACT UN.DELAY(UNIT)/MINUTES.V FROM DELTA.TIME
4141          LET UN.DELAY(UNIT) = 0
4142       ELSE
4143          ''THE UNIT IS STILL CLEARING THE MINEFIELD
4144          SUBTRACT DELTA.TIME*MINUTES.V FROM UN.DELAY(UNIT)
4145       ALWAYS
4146    ALWAYS
4147    LET UN.TIME.LAST.MOVE(UNIT) = TIME.V
4148    LET MOVE.RATE = TU.MOV.RATE(UN.TYPE.UNIT(UNIT)) * TAC.MOV.FAC *
4149    MOV.FAC(BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT)))
4150    IF TIME.V LT BMNT OR TIME.V GT EENT
4151       LET MOVE.RATE = MOVE.RATE * NITE.MOV.FAC
4152    ALWAYS
4153    IF UN.STATUS(UNIT) = WITHDRAWING
4154       LET MOVE.RATE = 3.0 * TU.MOV.RATE(UN.TYPE.UNIT(UNIT))
4155    ALWAYS
4156    LET DISTANCE =  MOVE.RATE * DELTA.TIME
4157    IF UN.DELAY LE 0
4158       IF UN.STATUS(UNIT) = ADVANCING
4159          OR UN.STATUS(UNIT) = ADV.TO.WITH
4160          CALL COMPUTE.D------------------------------------------------------------------------------>(321)
4161          GIVEN
4162          UNIT
4163          YIELDING
4164          D,
4165          POINT
4166          LOOP UNTIL D GT DISTANCE OR
4167          UN.POSITION.INDEX(UNIT) GE N.UN.PATH(UNIT)
4168          DO THE FOLLOWING
4169             ADD 1 TO UN.POSITION.INDEX(UNIT)
4170             LET UN.X.COORD(UNIT) = P.X(POINT)
4171             LET UN.Y.COORD(UNIT) = P.Y(POINT)
4172             SUBTRACT D FROM DISTANCE
4173             CALL SEGMENT.ADJUST------------------------------------------------------------------------>(122)
4174             GIVEN
4175             UNIT,
4176             D
4177             IF UN.POSITION.INDEX(UNIT) GE N.UN.PATH(UNIT),
4178          <------LEAVE
4179             OTHERWISE
4180             CALL COMPUTE.D------------------------------------------------------------------------------>(321)
4181             GIVEN
```

```
4182            UNIT
4183            YIELDING
4184            D,
4185            POINT
4186         ENDLOOP
4187         CALL DEQ.FEBA.SET------------------------------------------------------------------------------------------>(325)
4188         GIVEN
4189         UNIT
4190         YIELDING
4191         OLD.SECTOR
4192         CALL ENQ.FEBA.SET------------------------------------------------------------------------------------------>(331)
4193         GIVEN
4194         UNIT,
4195         OLD.SECTOR
4196         YIELDING
4197         NEW.SECTOR
4198         IF UN.POSITION.INDEX(UNIT) GE N.UN.PATH(UNIT),
4199            LET UN.POSITION.INDEX(UNIT) = N.UN.PATH(UNIT)
4200            IF UN.STATUS(UNIT) = ADVANCING
4201               LET UN.STATUS(UNIT) = STATIONARY
4202            ELSE
4203               LET UN.STATUS(UNIT) = STA.TO.WITH
4204            ALWAYS
4205            CALL LOS.CHECK------------------------------------------------------------------------------------------>(101)
4206            GIVEN
4207            UNIT
4208            LOOP UNTIL UN.SEGMENT.LIST(UNIT) IS EMPTY
4209            DO THE FOLLOWING
4210               REMOVE THE FIRST SEGMENT FROM UN.SEGMENT.LIST(UNIT)
4211               DESTROY THE SEGMENT
4212            ENDLOOP
4213 <----------EXITROUTINE
4214         OTHERWISE
4215         IF DISTANCE GT 0,
4216            CALL POSITION------------------------------------------------------------------------------------------>(110)
4217            GIVEN
4218            UNIT,
4219            D,
4220            DISTANCE,
4221            POINT
4222         ALWAYS
4223      ELSE ''UN.STATUS(UNIT) = WITHDRAWING
4224         CALL COMPUTE.WD------------------------------------------------------------------------------------------>(322)
4225         GIVEN
4226         UNIT
4227         YIELDING
4228         D,
4229         POINT
4230         LOOP UNTIL D GT DISTANCE OR
4231         UN.POSITION.INDEX(UNIT) LE 0
4232         DO THE FOLLOWING
4233            LET UN.X.COORD(UNIT) = P.X(POINT)
4234            LET UN.Y.COORD(UNIT) = P.Y(POINT)
4235            SUBTRACT D FROM DISTANCE
4236            SUBTRACT 1 FROM UN.POSITION.INDEX(UNIT)
4237            IF UN.POSITION.INDEX(UNIT) LE 0,
4238         <------LEAVE
4239            OTHERWISE
```

```
4240            CALL SEGMENT.ADJUST----------------------------------------------------------->(122)
4241            GIVEN
4242            UNIT,
4243            D
4244            CALL COMPUTE.WD--------------------------------------------------------------->(322)
4245            GIVEN
4246            UNIT
4247            YIELDING
4248            D,
4249            POINT
4250         ENDLOOP
4251         CALL DEQ.FEBA.SET------------------------------------------------------------------>(325)
4252         GIVEN
4253         UNIT
4254         YIELDING
4255         OLD.SECTOR
4256         CALL ENQ.FEBA.SET------------------------------------------------------------------>(331)
4257         GIVEN
4258         UNIT,
4259         OLD.SECTOR
4260         YIELDING
4261         NEW.SECTOR
4262         IF UN.POSITION.INDEX(UNIT) LE 0,
4263            LET UN.POSITION.INDEX(UNIT) = 1
4264            LET UN.STATUS(UNIT) = STATIONARY
4265            CALL LOS.CHECK---------------------------------------------------------------->(101)
4266            GIVEN
4267            UNIT
4268            ''CHECK TO SEE IF THE BATTLE IS OVER
4269            CALL TERM.CHECK--------------------------------------------------------------->(345)
4270            GIVEN
4271            UNIT
4272            LOOP UNTIL UN.SEGMENT.LIST(UNIT) IS EMPTY
4273            DO THE FOLLOWING
4274               REMOVE THE FIRST SEGMENT FROM UN.SEGMENT.LIST(UNIT)
4275               DESTROY THE SEGMENT
4276            ENDLOOP
4277 <----------EXITROUTINE
4278         OTHERWISE
4279         IF DISTANCE GT 0,
4280            FOR EVERY PNT OF UN.PATH(UNIT)
4281            WITH P.X(PNT) = UN.X.COORD(UNIT)
4282            AND  P.Y(PNT) = UN.Y.COORD(UNIT)
4283            FIND THE FIRST CASE
4284            IF FOUND,
4285               SUBTRACT 1 FROM UN.POSITION.INDEX(UNIT)
4286            ALWAYS
4287            CALL POSITION----------------------------------------------------------------->(110)
4288            GIVEN
4289            UNIT,
4290            D,
4291            DISTANCE,
4292            POINT
4293         ALWAYS
4294      ALWAYS
4295   ALWAYS
4296
4297   ''SCHEDULE THE NEXT MOVE
```

```
4298     CALL LOS.CHECK------------------------------------------------------------------------------------------>(101)
4299     GIVEN
4300     UNIT
4301     CALL MIN.MOVE-------------------------------------------------------------------------------------------->(106)
4302     GIVEN
4303     UNIT
4304
4305 <--EXITROUTINE
4306 ENDROUTINE
```

```
4307 ROUTINE END.MOVE                                                                                    M004
4308    GIVEN
4309    UNT.,
4310    ORDR.
4311    ADD 1 TO ANAL.CTR(16,1)  ''                                                                      \DYN_ANAL
4312
4313    ''THIS ROUTINE DOES THE BOOKKEEPING A THE END OF A MOVE.
4314    ''  IT DETERMINES THE SEQUENCE NO OF THE UNIT'S NEXT ORDER
4315    ''  AND PLACES THE UNIT IN ITS PROPER SECTOR.
4316
4317    NORMALLY MODE IS INTEGER
4318    DEFINE SECT AS A VARIABLE
4319    DEFINE MAN.UNIT AS A VARIABLE
4320
4321    LOOP FOR EVERY MAN.UNIT IN MU.TF.LIST(UN.PTR(UNT.))
4322    DO
4323       CALL DEQ.FEBA.SET------------------------------------------------------------------------------>(325)
4324       GIVEN
4325       MU.UNIT.ID(MAN.UNIT)
4326       YIELDING
4327       OLD.SECTOR
4328       LET UN.X.COORD(MU.UNIT.ID(MAN.UNIT)) = UN.X.COORD(UNT.)-
4329       MU.OFFSET.X(MAN.UNIT)
4330       LET UN.Y.COORD(MU.UNIT.ID(MAN.UNIT)) = UN.Y.COORD(UNT.)-
4331       MU.OFFSET.Y(MAN.UNIT)
4332       IF UN.RADIUS(MU.UNIT.ID(MAN.UNIT)) NE 333
4333          CALL ENQ.FEBA.SET---------------------------------------------------------------------->(331)
4334          GIVEN
4335          MU.UNIT.ID(MAN.UNIT),
4336          OLD.SECTOR
4337          YIELDING
4338          SECT.
4339       ALWAYS
4340    ENDLOOP
4341
4342    CALL DEQ.FEBA.SET------------------------------------------------------------------------------>(325)
4343    GIVEN
4344    UNT.
4345    YIELDING
4346    OLD.SECTOR
4347    IF UN.RADIUS(UNT.) NE 333
4348       CALL ENQ.FEBA.SET---------------------------------------------------------------------->(331)
4349       GIVEN
4350       UNT.,
4351       OLD.SECTOR
4352       YIELDING
4353       SECT.
4354    ALWAYS
4355
4356    ''EMPTY LIST OF BARRIER MINES THAT WOULD HAVE BEEN CROSSED.
4357    LOOP FOR EVERY .MO OF MO.LIST(UNT.)
4358    DO
4359       REMOVE .MO FROM THE MO.LIST(UNT.)
4360       DESTROY THE MINE.OBSTACLE CALLED .MO
4361    ENDLOOP
4362
4363    IF ORD.TYPE(ORDR.) = "MOVCOR"
4364       LET NEXT. = NX.ORDER(ORD.ID(ORDR.))
```

```
4365     ELSE
4366        IF ORD.TYPE(ORDR.) = "MOVDIS"
4367           LET NEXT. = ORD.NEXT(ORD.ID(ORDR.))
4368        ELSE
4369           IF ORD.TYPE(ORDR.) = "MOVREI"
4370              CALL REIN.ARRIVE------------------------------------------------------------------------------------------>(118)
4371              GIVEN
4372              UNT.,
4373              ORDR.
4374 <------------EXITROUTINE
4375           OTHERWISE
4376           ''THE SEQUENCE OF THE NEXT ORDER, WHICH IS DEFEND
4377           LET NEXT. = 1
4378        ALWAYS
4379     ALWAYS
4380
4381     SCHEDULE_A GET.NX.ORD-------------------------------------------------------------------------------------------->(373)
4382     GIVEN
4383     UNT.,
4384     NEXT.,
4385     0,
4386     0 NOW ''                                                                                                          \1
4387     SKIP 1 LINE
4388     PRINT 1 LINE WITH UNIT.NOS(UNT.),ORD.SEQ.NO(ORDR.),
4389     ORD.TYPE(ORDR.), TIME.V THUS
4390     MOVEMENT OF UNIT ****** UNDER ORDER #** (******)  TERMINATED AT **.** HOURS
4391     SKIP 1 LINE
4392
4393 <--EXITROUTINE
4394 ENDROUTINE
```

```
4395 ROUTINE FA.BN.MOVEMENT                    ''                                                    M005
4396    GIVEN .FA.BN  AND .BTRY
4397
4398    ADD 1 TO ANAL.CTR(17,1)  ''                                                                  \DYN_ANAL
4399
4400    NORMALLY MODE IS INTEGER
4401    DEFINE .BTRY, .SECTOR, .SIDE AND .FA.BN AS INTEGER VARIABLES
4402    DEFINE MARCH.ORDER TO MEAN 3
4403    IF .FA.BN = 0 AND .BTRY = 0
4404       PRINT 1 LINE WITH TIME.V THUS
4405       == FA$BN$MOVE == BOTH ARGUMENTS 0 AT TIME.V = ****.*****
4406 <------EXITROUTINE
4407    OTHERWISE
4408    ''WHEN A BTRY SHOULD BE MOVING, PICK THE ONE NEEDING TO MOVE MOST
4409    IF .FA.BN NE 0
4410       LET .UNIT = BY.UNIT(F.BN.BTRY.SET(.FA.BN))
4411       LET .SIDE = UN.COLOR(.UNIT)
4412       IF .SIDE GT 2 OR .SIDE LT 1
4413          TRACE   STOP
4414       OTHERWISE
4415       IF .SIDE = BLUE
4416          LET BACK = -1
4417          LET FORWARD = +1
4418       ELSE
4419          LET BACK = +1
4420          LET FORWARD = -1
4421       ENDIF
4422       LOOP FOR EACH .BTRY IN BN.BTRY.SET(.FA.BN)
4423       DO THIS ''                                                                                 \1
4424          LET .UNIT = BY.UNIT(.BTRY)
4425          LET GP = CT.GROUP(TU.CAT(UN.TYPE.UNIT(.UNIT)))
4426          IF GP NE ARTILLERY
4427       <------CYCLE
4428          ENDIF
4429          IF BY.CUR.FM(.BTRY) NE 0 AND TB.SHOOT.SCOOT.IND(BY.TYPE(.BTRY)) GT 0
4430       <------CYCLE
4431          OTHERWISE
4432          IF TIME.V <= UN.TIME.LAST.MOVE(.UNIT) + TIME.BETWEEN.ARTY.MOVE/60.
4433             OR BY.STATUS(.BTRY) > 0
4434       <------CYCLE
4435          OTHERWISE
4436          CALL LOCATE.SECTOR GIVEN UN.Y.COORD(BY.UNIT(.BTRY)) YIELDING .SECTOR------------------------------------------------------>(100)
4437          IF SS.SET(.SIDE,.SECTOR) IS EMPTY
4438             IF .SECTOR+1 <= N.SECTOR AND
4439                SS.SET(.SIDE,.SECTOR+1) IS NOT EMPTY
4440                LET .SECTOR = .SECTOR + 1
4441             ELSE
4442                IF .SECTOR - 1 > 0 AND
4443                   SS.SET(.SIDE,.SECTOR-1) IS NOT EMPTY
4444                   LET .SECTOR = .SECTOR - 1
4445                ELSE
4446 <----------------EXITROUTINE
4447                ENDIF
4448             ENDIF
4449          ENDIF
4450          IF .SIDE = BLUE
4451             LET SECTOR.LEADER = F.SS.SET(.SIDE,.SECTOR)
4452          ELSE
```

```
4453            LET SECTOR.LEADER = L.SS.SET(.SIDE,.SECTOR)
4454         ALWAYS
4455         LET RANGE = FORWARD*(UN.X.COORD(SECTOR.LEADER)-UN.X.COORD(BY.UNIT(.BTRY)))
4456         COMPUTE MAX.BAT AS MAX(.BTRY),MAX.DIST AS MAXIMUM OF RANGE
4457         COMPUTE MIN.BAT AS MIN(.BTRY),MIN.DIST AS MINIMUM OF RANGE
4458      ENDLOOP
4459      IF MIN.BAT = 0 OR MAX.BAT = 0, ''                                          CHG\02   \ZERO_SUB
4460 <--------RETURN '' NO .BTRY TO BE MOVED
4461      OTHERWISE
4462      IF MIN.DIST <= TB.MIN.FEBA(BY.TYPE(MIN.BAT)) AND MIN.BAT > 0
4463         LET BACK = BACK + (TB.MIN.FEBA(BY.TYPE(MIN.BAT))-MIN.DIST +
4464         2*(TB.MAX.FEBA(BY.TYPE(MIN.BAT))-TB.MIN.FEBA(BY.TYPE(MIN.BAT)))/3)
4465         SCHEDULE_A START.ARTY.MOVEMENT GIVEN MIN.BAT, BACK------------------------------------------------------------------------>(393)
4466         IN REAL.F(TB.MARCH.ORDER(BY.TYPE(MIN.BAT))) MINUTES
4467         IF ANALYSIS(1) = TRUE
4468            IF ANALYSIS(1) = TRUE
4469               USE UNIT 42 FOR OUTPUT
4470               PRINT 1 LINE WITH MIN.BAT, BY.STATUS(MIN.BAT), TIME.V THUS
4471               BTRY *** CHANGES FROM STATUS * TO STATUS 3 AT **.*****
4472               USE UNIT 6 FOR OUTPUT
4473            ALWAYS
4474         ALWAYS
4475         LET BY.STATUS(MIN.BAT) = MARCH.ORDER
4476         FOR EACH FM IN THE BY.FM.QUEUE(MIN.BAT)
4477         LET FM.N.VOLS(FM) = 0
4478         IF BY.CUR.FM(MIN.BAT) NE 0
4479            LET FM.N.VOLS(BY.CUR.FM(MIN.BAT)) = FM.FIRED.VOLS(BY.CUR.FM(MIN.BAT))
4480         ALWAYS
4481      ELSE
4482         IF MAX.DIST >= TB.MAX.FEBA(BY.TYPE(MAX.BAT)) AND MAX.BAT > 0
4483            LET FORWARD = FORWARD + (MAX.DIST-TB.MAX.FEBA(BY.TYPE(MAX.BAT)) +
4484            2*(TB.MAX.FEBA(BY.TYPE(MAX.BAT))-TB.MIN.FEBA(BY.TYPE(MAX.BAT)))/3)
4485            SCHEDULE_A START.ARTY.MOVEMENT GIVEN MAX.BAT, FORWARD------------------------------------------------------------------->(393)
4486            IN REAL.F(TB.MARCH.ORDER(BY.TYPE(MAX.BAT))) MINUTES
4487            IF ANALYSIS(1) = TRUE
4488               USE UNIT 42 FOR OUTPUT
4489               PRINT 1 LINE WITH MAX.BAT, BY.STATUS(MAX.BAT), TIME.V THUS
4490               BTRY *** CHANGES FROM STATUS * TO STATUS 3 AT **.*****
4491               USE UNIT 6 FOR OUTPUT
4492            ALWAYS
4493            LET BY.STATUS(MAX.BAT) = MARCH.ORDER
4494            FOR EACH FM IN THE BY.FM.QUEUE(MAX.BAT)
4495            LET FM.N.VOLS(FM) = 0
4496            IF BY.CUR.FM(MAX.BAT) NE 0
4497               LET FM.N.VOLS(BY.CUR.FM(MAX.BAT)) = FM.FIRED.VOLS(BY.CUR.FM(MAX.BAT))
4498            ALWAYS
4499         ENDIF
4500      ENDIF
4501   ELSE  '' SHOOT AND SCOOT MOVE FOR .BTRY
4502      IF BY.STATUS(.BTRY) NE 0
4503 <--------RETURN
4504      OTHERWISE
4505      LET .SIDE = UN.COLOR(BY.UNIT(.BTRY))
4506      IF .SIDE GT 2 OR .SIDE LT 1
4507         TRACE   STOP
4508      OTHERWISE
4509      IF .SIDE = BLUE
4510         LET BACK = -1
```

```
4511          LET FORWARD = +1
4512       ELSE
4513          LET BACK = +1
4514          LET FORWARD = -1
4515       ENDIF
4516       CALL LOCATE.SECTOR GIVEN UN.Y.COORD(BY.UNIT(.BTRY)) YIELDING .SECTOR------------------------------------------------------>(100)
4517       IF SS.SET(.SIDE,.SECTOR) IS EMPTY
4518          IF .SECTOR+1 <= N.SECTOR AND
4519             SS.SET(.SIDE,.SECTOR+1) IS NOT EMPTY
4520             LET .SECTOR = .SECTOR + 1
4521          ELSE
4522             IF .SECTOR - 1 > 0 AND
4523                SS.SET(.SIDE,.SECTOR-1) IS NOT EMPTY
4524                LET .SECTOR = .SECTOR -1
4525             ELSE
4526 <-------------EXITROUTINE
4527             ENDIF
4528          ENDIF
4529       ENDIF
4530       IF .SIDE = BLUE
4531          LET SECTOR.LEADER = F.SS.SET(.SIDE,.SECTOR)
4532       ELSE
4533          LET SECTOR.LEADER = L.SS.SET(.SIDE,.SECTOR)
4534       ALWAYS
4535       LET RANGE = FORWARD*(UN.X.COORD(SECTOR.LEADER)-UN.X.COORD(BY.UNIT(.BTRY)))
4536       FOR EACH FM IN BY.FM.QUEUE(.BTRY)
4537       LET FM.N.VOLS(FM) = 0
4538       IF BY.CUR.FM(.BTRY) > 0 ''                                                                    \ZERO_SUB CHG\16
4539          LET FM.N.VOLS(BY.CUR.FM(.BTRY)) = FM.FIRED.VOLS(BY.CUR.FM(.BTRY))
4540       ALWAYS
4541       IF RANGE LE TB.MIN.FEBA(BY.TYPE(.BTRY))
4542          LET BACK = BACK * (TB.MIN.FEBA(BY.TYPE(.BTRY)) - RANGE +
4543          2*(TB.MAX.FEBA(BY.TYPE(.BTRY)) - TB.MIN.FEBA(BY.TYPE(.BTRY)))/3)
4544          SCHEDULE_A START.ARTY.MOVEMENT GIVEN .BTRY AND BACK IN------------------------------------------------------------>(393)
4545          REAL.F(TB.MARCH.ORDER(BY.TYPE(.BTRY))) MINUTES
4546          IF ANALYSIS(1) =TRUE
4547             USE UNIT 42 FOR OUTPUT
4548             PRINT 1 LINE WITH .BTRY, BY.STATUS(.BTRY), TIME.V THUS
4549             BTRY *** CHANGES FROM STATUS * TO STATUS 3 AT **.*****
4550             USE UNIT 6 FOR OUTPUT
4551          ALWAYS
4552          LET BY.STATUS(.BTRY) = MARCH.ORDER
4553 <--------EXITROUTINE
4554       OTHERWISE
4555       IF RANGE GE TB.MAX.FEBA(BY.TYPE(.BTRY))
4556          LET FORWARD = FORWARD * (RANGE - TB.MAX.FEBA(BY.TYPE(.BTRY)) +
4557          2*(TB.MAX.FEBA(BY.TYPE(.BTRY)) - TB.MIN.FEBA(BY.TYPE(.BTRY)))/3)
4558          SCHEDULE_A START.ARTY.MOVEMENT GIVEN .BTRY AND FORWARD------------------------------------------------------------>(393)
4559          IN REAL.F(TB.MARCH.ORDER(BY.TYPE(.BTRY))) MINUTES
4560          IF ANALYSIS(1) =TRUE
4561             USE UNIT 42 FOR OUTPUT
4562             PRINT 1 LINE WITH .BTRY, BY.STATUS(.BTRY), TIME.V THUS
4563             BTRY *** CHANGES FROM STATUS * TO STATUS 3 AT **.*****
4564             USE UNIT 6 FOR OUTPUT
4565          ALWAYS
4566          LET BY.STATUS(.BTRY) = MARCH.ORDER
4567 <--------EXITROUTINE
4568       OTHERWISE
```

```
4569        IF (TB.MAX.FEBA(BY.TYPE(.BTRY)) - RANGE) LT
4570           (RANGE - TB.MIN.FEBA(BY.TYPE(.BTRY)))
4571           LET .MOVE = FORWARD * 62.5 ''HDM
4572        ELSE
4573           LET .MOVE = BACK * 62.5 ''HDM
4574        ALWAYS
4575        SCHEDULE_A START.ARTY.MOVEMENT GIVEN .BTRY AND .MOVE IN----------->(393)
4576        REAL.F(TB.MARCH.ORDER(BY.TYPE(.BTRY))) MINUTES
4577        IF ANALYSIS(1) =TRUE
4578           USE UNIT 42 FOR OUTPUT
4579           PRINT 1 LINE WITH .BTRY, BY.STATUS(.BTRY), TIME.V THUS
4580           BTRY *** CHANGES FROM STATUS * TO STATUS 3 AT **.*****
4581           USE UNIT 6 FOR OUTPUT
4582        ALWAYS
4583        LET BY.STATUS(.BTRY) = MARCH.ORDER
4584     ALWAYS
4585 <--EXITROUTINE
4586 ENDROUTINE
```

```
4587 ROUTINE INITIAL.DETECT                                                                                    M006
4588    GIVEN
4589    BATTLE
4590    ADD 1 TO ANAL.CTR(18,1)  ''                                                                    \DYN_ANAL
4591
4592    NORMALLY MODE IS INTEGER
4593    ''THIS ROUTINE GOES THROUGH THE ENTIRE LIST OF
4594    ''UNITS THAT ARE VISIBLE TO EACH OTHER AND CALLS
4595    ''THE ROUTINE THAT SCHEDULES DETECTIONS
4596    LOOP
4597    FOR EACH FORCE OF BTL.FORCE.SET(BATTLE)
4598    DO THE FOLLOWING
4599       LOOP
4600       FOR EACH UNIT OF FR.UNIT.SET(FORCE)
4601       DO THE FOLLOWING
4602          LOOP
4603          FOR EACH VISIBLE.UNIT OF UN.LOS.LIST(UNIT)
4604          DO THE FOLLOWING
4605             CALL TIME.TO.DETECT GIVEN------------------------------------------------------------------------>(123)
4606             UNIT,
4607             VU.POINTER(VISIBLE.UNIT)
4608          ENDLOOP
4609       ENDLOOP
4610    ENDLOOP
4611 <--EXITROUTINE
4612 ENDROUTINE
```

```
4613 ROUTINE INITIAL.MOVE                                                    ''                                                    M007
4614    GIVEN
4615    BATTLE
4616    ADD 1 TO ANAL.CTR(19,1)  ''                                                                                          \DYN_ANAL
4617
4618    NORMALLY MODE IS INTEGER
4619    ''THIS ROUTINE SCHEDULES ALL OF THE INITIAL MOVES FOR BOTH
4620    ''FORCES IN A BATTLE
4621    LOOP
4622    FOR EACH FORCE OF BTL.FORCE.SET(BATTLE)
4623    DO THE FOLLOWING
4624       LOOP
4625       FOR EACH UNIT OF FR.UNIT.SET(FORCE)
4626       DO THE FOLLOWING
4627          IF UN.STATUS(UNIT) NE STATIONARY
4628             CALL MIN.MOVE GIVEN UNIT------------------------------------------------------------------------------------>(106)
4629          ALWAYS
4630          '' CALL POSITION.OUT---------------------------------------------------------------------------------------->(613)
4631          '' GIVEN
4632          '' UNIT
4633       ENDLOOP
4634    ENDLOOP
4635 <--EXITROUTINE
4636 ENDROUTINE
```

```
4637 ROUTINE INIT.REINF GIVEN UNT.CALLING.,UNT.RESPONDING.,ORDR.,  ''                    M008
4638    EN.UNITS.
4639    ADD 1 TO ANAL.CTR(20,1)  ''                                                     \DYN_ANAL
4640
4641    ''THIS ROUTINE INITIATES THE MOVE OF A UNIT TO REINFORCE ANOTHER
4642    ''    UNIT.  IT CREATES AN ORDER, ADDS IT TO THE ORDER SET OF
4643    ''    THE REINFORCING UNIT AND SCHEDULES A START.MOVE IN AN APPRO-
4644    ''    PRIATE TIME DELAY.
4645
4646    NORMALLY MODE IS INTEGER
4647    DEFINE TYPE. AS A TEXT VARIABLE ''                                              \TEXT
4648    DEFINE EN.UNITS. AS A 1-DIMENSIONAL ARRAY
4649    CREATE AN ORDER
4650    LET ORD.TYPE(ORDER)="MOVREI"
4651    CREATE A MOVREINF.ORD CALLED ORD.ID(ORDER)
4652    LET REINFORCED.UN(ORD.ID(ORDER))=UNT.CALLING.
4653    LET ATK.UNITS.PTR(ORD.ID(ORDER))=EN.UNITS.(*)
4654    LET ORD.SEQ.NO(ORDER)=ORD.SEQ.NO(L.MU.ORDER.SET
4655    (UN.PTR(UNT.RESPONDING)))+1
4656    FILE ORDER IN MU.ORDER.SET(UN.PTR(UNT.RESPONDING.))
4657    LET TYPE.="REINF"
4658    LET MU.CUR.ORDER(UN.PTR(UNT.RESPONDING.))=ORD.SEQ.NO(ORDER)
4659    LET UN.MISSION(UNT.RESPONDING.)=UN.MISSION(UNT.CALLING)
4660    SCHEDULE_A START.MOVE(UNT.RESPONDING.,UN.X.COORD(UNT.CALLING.),----------->(403)
4661    UN.Y.COORD(UNT.CALLING.),TYPE.,ORDER) IN REIN.DELAY MINUTES
4662 ENDROUTINE
```

```
4663 ROUTINE LINE.OF.SIGHT                                                    ''      M009
4664     GIVEN
4665     RED.UNITS,
4666     BLUE.UNITS,
4667     .TERRAIN
4668     ADD 1 TO ANAL.CTR(21,1)  ''                                                  \DYN_ANAL
4669
4670     ''THIS ROUTINE COMPUTES WHICH UNITS ARE IN LINE OF SIGHT AT
4671     ''THE START OF A BATTLE WHEN LOS EXISTS BETWEEN TWO UNITS,
4672     ''THEY ARE PLACED IN EACH OTHERS LIST OF VISIBLE UNITS
4673
4674     NORMALLY MODE IS INTEGER
4675     DEFINE SHAPE,SCALE,W AS REAL VARIABLES
4676     DEFINE BLUE.UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
4677     DEFINE RED.UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
4678
4679     LET NO.BLUE.UNITS = DIM.F(BLUE.UNITS(*))
4680     LET NO.RED.UNITS = DIM.F(RED.UNITS(*))
4681     LOOP FOR I = 1 TO NO.BLUE.UNITS
4682     DO
4683        LOOP FOR J = 1 TO NO.RED.UNITS
4684        DO
4685           CALL RANGE.COMPUTE------------------------------------------------------------------------------------------------------>(342)
4686           GIVEN
4687           BLUE.UNITS(I),
4688           RED.UNITS(J)
4689           YIELDING
4690           RANGE
4691           IF UN.STATUS(RED.UNITS(J)) = STATIONARY
4692              IF UN.STATUS(BLUE.UNITS(I)) = STATIONARY
4693                 LET RNG = TT.STATIONARY.LOS.BREAK(.TERRAIN)
4694              ELSE
4695                 LET RNG = TT.M.S.LOS.BREAK(.TERRAIN)
4696              ALWAYS
4697           ELSE
4698              IF UN.STATUS(BLUE.UNITS(I)) = STATIONARY
4699                 LET RNG = TT.M.S.LOS.BREAK(.TERRAIN)
4700              ELSE
4701                 LET RNG. = TT.MOVING.LOS.BREAK(.TERRAIN)
4702              ALWAYS
4703           ALWAYS
4704           FOR EVERY LOS.BAND
4705           WITH BAND.RANGE(LOS.BAND) GT RANGE
4706           FIND THE FIRST CASE
4707           IF NONE
4708              LET LOS = NO
4709           ELSE
4710              ''LOS.PROB STORED AS [0, 50]
4711              IF (LOS.PROB(.TERRAIN,LOS.BAND)/50)
4712                 GE RANDOM.F(RN.SEED)
4713                 OR RANGE LE RNG.
4714                 LET LOS = YES
4715                 CREATE A VISIBLE.UNIT
4716                 LET VU.POINTER(VISIBLE.UNIT) = RED.UNITS(J)
4717                 LET VU.STATUS(VISIBLE.UNIT) = NO
4718                 FILE THIS VISIBLE.UNIT IN UN.LOS.LIST(BLUE.UNITS(I))
4719                 CREATE A VISIBLE.UNIT
4720                 LET VU.POINTER(VISIBLE.UNIT) = BLUE.UNITS(I)
```

```
4721                LET VU.STATUS(VISIBLE.UNIT) = NO
4722                FILE THIS VISIBLE.UNIT IN UN.LOS.LIST(RED.UNITS(J))
4723             ELSE
4724                LET LOS = NO
4725             ALWAYS
4726          ALWAYS
4727          IF UN.STATUS(BLUE.UNITS(I)) NE STATIONARY
4728             CREATE A SEGMENT
4729             LET SEG.TYPE(SEGMENT) = LOS
4730             LET SEG.UNIT(SEGMENT) = RED.UNITS(J)
4731             IF LOS = YES
4732                LET SHAPE = TT.LOS.SHAPE(.TERRAIN)
4733                LET SCALE = TT.LOS.SCALE(.TERRAIN)
4734             ELSE
4735                LET SHAPE = TT.NLOS.SHAPE(.TERRAIN)
4736                LET SCALE = TT.NLOS.SCALE(.TERRAIN)
4737             ALWAYS
4738             ''UTILIZE WEIBULL.F FUNCTION IN NEXT DRAW ---------- \1>(642)
4739             LET W = WEIBULL.F(SHAPE,SCALE,RN.SEED)
4740             LET SEG.LENGTH(SEGMENT) = MAX.F(2.,(W*TER.W.INC./16.))
4741             IF LOS = NO AND SEG.LENGTH(SEGMENT) GT RANGE-RNG
4742                LET SEG.LENGTH(SEGMENT) = RANGE - RNG
4743             ALWAYS
4744             FILE THIS SEGMENT IN THE UN.SEGMENT.LIST(BLUE.UNITS(I))
4745          ALWAYS
4746          IF UN.STATUS(RED.UNITS(J)) NE STATIONARY
4747             CREATE A SEGMENT
4748             LET SEG.TYPE(SEGMENT) = LOS
4749             LET SEG.UNIT(SEGMENT) = BLUE.UNITS(I)
4750             IF LOS = YES
4751                LET SHAPE = TT.LOS.SHAPE(.TERRAIN)
4752                LET SCALE = TT.LOS.SCALE(.TERRAIN)
4753             ELSE
4754                LET SHAPE = TT.NLOS.SHAPE(.TERRAIN)
4755                LET SCALE = TT.NLOS.SCALE(.TERRAIN)
4756             ALWAYS
4757             ''UTILIZE WEIBULL.F FUNCTION IN NEXT DRAW ---------- \1>(642)
4758             LET W = WEIBULL.F(SHAPE,SCALE,RN.SEED)
4759             LET SEG.LENGTH(SEGMENT) = MAX.F(2.,(W*TER.W.INC./16.))
4760             IF LOS = NO AND SEG.LENGTH(SEGMENT) GT RANGE-RNG
4761                LET SEG.LENGTH(SEGMENT) = RANGE - RNG
4762             ALWAYS
4763             FILE THIS SEGMENT IN THE UN.SEGMENT.LIST(RED.UNITS(J))
4764          ALWAYS
4765       ENDLOOP
4766    ENDLOOP
4767
4768 <--EXITROUTINE
4769 ENDROUTINE
```

```
4770 ROUTINE LOCATE.SEARCH.AREA                                                    M010
4771    GIVEN
4772    FO
4773    YIELDING
4774    SEARCH.X.GRID,
4775    SEARCH.Y.GRID, FO.X.CORRECT, FO.Y.CORRECT
4776    ADD 1 TO ANAL.CTR(22,1)  ''                                          \DYN_ANAL
4777    NORMALLY MODE IS INTEGER
4778    DEFINE DIRECTION, R, ANGLE,  SEARCH.DIRECTION, RADIUS,
4779    FO.DISPLACEMENT.DIR,
4780    MAX.RANGE, MAX.FOV.RANGE, FRACT.MAX.RANGE AS REAL VARIABLES
4781    DEFINE SIDE AS AN INTEGER VARIABLE
4782    LET LINK = FO.US.LINK(FO)
4783    LET UN =  US.UNIT(LINK)
4784    LET RADIUS = UN.RADIUS(UN)/16.
4785    LET SIDE = UN.COLOR( UN )
4786    IF SIDE = BLUE
4787       LET DIRECTION = 0
4788    ELSE
4789       LET DIRECTION =  PI.C
4790    ENDIF
4791    LET R = BETA.F( 8.0 , 8.0 , RN.SEED )
4792    LET ANGLE = - PI.C + R*2*PI.C ''                                     \OPTIMIZE
4793    LET SEARCH.DIRECTION  = DIRECTION + ANGLE
4794    IF SEARCH.DIRECTION < 0
4795       ADD 2. * PI.C TO SEARCH.DIRECTION
4796    ENDIF
4797    LET MAX.RANGE = FO.RB.RANGE(L.MFO.RB.SET(US.MODEL(LINK)))
4798    LET MAX.FOV.RANGE = (MAX.RANGE/2) * ( COS.F( ANGLE ) + 1. )
4799    LET FRACT.MAX.RANGE = MAX.FOV.RANGE / MAX.RANGE
4800    LET R = 10.0
4801    UNTIL R <  FRACT.MAX.RANGE
4802    LET R = BETA.F( 3.0 , 4.0 , RN.SEED )
4803    LET SEARCH.RANGE = R * MAX.RANGE
4804    LET FO.DISPLACEMENT.DIR = DIRECTION + FO.REL.DIRECTION(FO)
4805    IF FO.DISPLACEMENT.DIR < 0
4806       ADD 2. * PI.C TO FO.DISPLACEMENT.DIR
4807    ENDIF
4808    LET FO.X.CORRECT = INT.F( RADIUS * COS.F(FO.DISPLACEMENT.DIR))
4809    LET FO.Y.CORRECT = INT.F( RADIUS * SIN.F(FO.DISPLACEMENT.DIR))
4810    LET SP.X.COORD(1) = UN.X.COORD(UN) + FO.X.CORRECT +
4811    INT.F(SEARCH.RANGE * COS.F(SEARCH.DIRECTION))
4812    IF SP.X.COORD(1) < 0
4813       LET SP.X.COORD(1) = 0
4814    ELSE
4815       IF SP.X.COORD(1) > 2*FRONT.DEPTH
4816          LET SP.X.COORD(1) = 2*FRONT.DEPTH
4817       ENDIF
4818    ENDIF
4819    LET SP.Y.COORD(1) = UN.Y.COORD(UN) + FO.Y.CORRECT +
4820    INT.F(SEARCH.RANGE * SIN.F(SEARCH.DIRECTION))
4821    IF SP.Y.COORD(1) < 0
4822       LET SP.Y.COORD(1) = 0
4823    ELSE
4824       IF SP.Y.COORD(1) > FEBA.WIDTH
4825          LET SP.Y.COORD(1) = FEBA.WIDTH
4826       ENDIF
4827    ENDIF
```

```
4828     LET SEARCH.X.GRID = SP.X.GRID(1)
4829     LET SEARCH.Y.GRID = SP.Y.GRID(1)
4830 ENDROUTINE
```

```
4831 ROUTINE LOCATE.SECTOR                                                    ''                M011
4832    GIVEN Y.COORD
4833    YIELDING SECTOR
4834
4835    ADD 1 TO ANAL.CTR(23,1)  ''                                                             \DYN_ANAL
4836    NORMALLY MODE IS INTEGER
4837
4838    LET B1 = LEFT.BNDRY.INT
4839
4840    LOOP FOR EACH SECTOR,
4841    DO THIS ''                                                                              \1
4842       IF SECTOR > 1
4843          LET B1 = SE.BNDRY.INT( SECTOR - 1 )
4844       ENDIF
4845       LET B2 = SE.BNDRY.INT( SECTOR )
4846       IF B1 >= Y.COORD > B2
4847    <------EXITLOOP  ''RETURN WITH THE SECTOR
4848       OTHERWISE
4849
4850       IF Y.COORD >= INIT.Y.FEBA       '' ITS OFF THE UPPER GRID
4851          ''PRINT 1 LINE THUS
4852          ''=  =  A UNIT IS OFF THE UPPER  GRID AND ASSIGNED SECTOR 1 =  =
4853          LET SECTOR = 1
4854    <------EXITLOOP
4855       ENDIF
4856       IF Y.COORD <= 0      '' ITS OFF THE LOWER GRID
4857          LET SECTOR = N.SECTOR    '' LOWEST SECTOR
4858          ''PRINT 1 LINE THUS
4859          ''=  = A UNIT IS OFF THE LOWER GRID AND ASSIGNED N.SECTOR =  =
4860    <------EXITLOOP
4861       ENDIF
4862       '' THIS ROUTINE MODIFIED %7MAR80_%RGR
4863    ENDLOOP
4864
4865    IF DEBUG=TRUE,
4866       PRINT 1 LINE WITH Y.COORD, SECTOR THUS
4867       LOCATE.SECTOR Y.COORD = ***** AND SECTOR ASSIGNED IS **
4868    ENDIF
4869
4870 ENDROUTINE
```

```
4871 ROUTINE LOS.CHECK ''                                                                                    M012
4872    GIVEN
4873    UNIT
4874    ''LOS$CHECK
4875    ADD 1 TO ANAL.CTR(24,1)  ''                                                                          \DYN_ANAL
4876    NORMALLY MODE IS INTEGER
4877    FOR EVERY FORCE OF BTL.FORCE.SET(UN.BATTLE.INDEX(UNIT))
4878    WITH FR.SIDE(FORCE) NE UN.COLOR(UNIT)
4879    FIND THE FIRST CASE
4880    IF NONE,
4881       CALL ERROR.STOP  ''------------------------------------------------------------------------------- \1>(604)
4882    ENDIF
4883    LOOP
4884    FOR EVERY OTHER.UNIT OF FR.UNIT.SET(FORCE)
4885    DO THE FOLLOWING
4886       FOR EVERY VISIBLE.UNIT OF UN.LOS.LIST(OTHER.UNIT)
4887       WITH VU.POINTER(VISIBLE.UNIT) = UNIT
4888       FIND THE FIRST CASE
4889       IF NONE,
4890          IF UN.STATUS(UNIT) = STATIONARY
4891             OR UN.STATUS(UNIT) = STA.TO.WITH
4892             IF UN.STATUS(OTHER.UNIT) = STATIONARY
4893                OR UN.STATUS(OTHER.UNIT) = STA.TO.WITH
4894                LET RNG. = TT.STATIONARY.LOS.BREAK(BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT)))
4895             ELSE
4896                LET RNG. = TT.M.S.LOS.BREAK(BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT)))
4897             ALWAYS
4898          ELSE
4899             IF UN.STATUS(OTHER.UNIT) = STATIONARY
4900                OR UN.STATUS(OTHER.UNIT) = STA.TO.WITH
4901                LET RNG. = TT.M.S.LOS.BREAK(BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT)))
4902             ELSE
4903                LET RNG. = TT.MOVING.LOS.BREAK(BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT)))
4904             ALWAYS
4905          ALWAYS
4906          CALL RANGE.COMPUTE-------------------------------------------------------------------------------->(342)
4907          GIVEN
4908          UNIT,
4909          OTHER.UNIT
4910          YIELDING
4911          RANGE
4912          IF RANGE LE RNG.
4913             CREATE A VISIBLE.UNIT
4914             LET VU.POINTER(VISIBLE.UNIT) = UNIT
4915             LET VU.STATUS(VISIBLE.UNIT) = NO
4916             FILE THIS VISIBLE.UNIT IN THE UN.LOS.LIST(OTHER.UNIT)
4917             CREATE A VISIBLE.UNIT
4918             LET VU.POINTER(VISIBLE.UNIT) = OTHER.UNIT
4919             LET VU.STATUS(VISIBLE.UNIT) = NO
4920             FILE THIS VISIBLE.UNIT IN THE UN.LOS.LIST(UNIT)
4921          ALWAYS
4922       ALWAYS
4923       CALL CHECK.ENGAGEMENT----------------------------------------------------------------------------->(132)
4924       GIVEN
4925       OTHER.UNIT
4926    ENDLOOP
4927    CALL CHECK.ENGAGEMENT-------------------------------------------------------------------------------->(132)
4928    GIVEN
```

```
4929    UNIT
4930 <--EXITROUTINE
4931 ENDROUTINE
```

```
4932 ROUTINE MINE.DELAY                                                                                    M013
4933     GIVEN
4934     .UNIT,
4935     .XCOR,
4936     .YCOR,
4937     .TYPE
4938
4939     YIELDING
4940     .DELAY
4941
4942     ADD 1 TO ANAL.CTR(25,1)  ''                                                                       \DYN_ANAL
4943     ''THIS ROUTINE COMPUTES THE DELAY IN MINUTES CAUSED BY
4944     ''A MINEFIELD - ALSO, BARRIER MINEFIELDS WILL HAVE A
4945     ''FASCAM MISSION CALLED TO RESEED THEM
4946
4947     NORMALLY MODE IS INTEGER
4948     DEFINE .MFB.DELAY AS A REAL VARIABLE
4949     DEFINE BARRIER TO MEAN 1
4950
4951     IF UN.COLOR(.UNIT) = BLUE
4952        LET .KILLER.COLOR = RED
4953     ELSE
4954        LET .KILLER.COLOR = BLUE
4955     ALWAYS
4956
4957     IF UN.STATUS(.UNIT) = STATIONARY OR UN.STATUS(.UNIT) = STA.TO.WITH
4958        IF MF.DEBUG = TRUE
4959           PRINT 1 LINE WITH
4960           .UNIT,
4961           UN.STATUS(.UNIT) THUS
4962           ---MINE.DELAY FOR UNIT=**** - UNIT NOT MOVING, STATUS=**
4963        ALWAYS
4964 <------EXITROUTINE
4965     OTHERWISE
4966
4967     ''FIND THE CLOSEST FORWARD OBSERVER
4968     LET .CLOSEST.FO = 0
4969     LET .MIN.DIS = INF.C
4970     LOOP
4971     FOR EVERY .FO IN EV.S(I.FORWARD.OBSERVER)
4972     WITH UN.COLOR(US.UNIT(FO.US.LINK(.FO))) = .KILLER.COLOR
4973     DO
4974        LET .FO.UNIT = US.UNIT(FO.US.LINK(.FO))
4975        LET .DELTAX = .XCOR - UN.X.COORD(.FO.UNIT)
4976        LET .DELTAY = .YCOR - UN.Y.COORD(.FO.UNIT)
4977        LET .DIS = SQRT.F(.DELTAX**2 + .DELTAY**2)  ''                                                 \OPTIMIZE
4978        IF .DIS LT .MIN.DIS
4979           LET .MIN.DIS = .DIS
4980           LET .CLOSEST.FO = .FO.UNIT
4981        ALWAYS
4982     ENDLOOP
4983     LET .FO.UNIT = .CLOSEST.FO
4984
4985     IF .CLOSEST.FO NE 0 AND .TYPE = BARRIER
4986        CALL REQUEST.FASCAM------------------------------------------------------------------------------>(216)
4987        GIVEN
4988        .FO.UNIT,
4989        .FO.UNIT,
```

```
4990        .XCOR,
4991        .YCOR,
4992        0
4993     ALWAYS
4994     LET .MINE.DIST = ABS.F(.XCOR - UN.X.COORD(.FO.UNIT))
4995
4996     FOR EVERY MF.BAND CALLED .MFB
4997     WITH MFB.UPPER.LIMIT(.KILLER.COLOR, .MFB) GE .MINE.DIST
4998     FIND THE FIRST CASE
4999     IF NONE
5000        LET .MFB = 3
5001     ALWAYS
5002     LET .MFB.DELAY = MFB.DELAY(.KILLER.COLOR, .MFB)
5003     IF .MFB.DELAY GT 0.0
5004        LET .DELAY = (BETA.F(1.,5.,RN.SEED))*.MFB.DELAY  ''MIN
5005     ALWAYS
5006     LET .DELAY = .DELAY * TU.MF.FACTOR(UN.TYPE.UNIT(.UNIT))
5007
5008     IF MF.DEBUG = TRUE
5009        PRINT 1 LINE WITH
5010        .DELAY,
5011        .UNIT,
5012        .XCOR,
5013        .YCOR,
5014        .TYPE THUS
5015        ---MINE.DELAY OF *** MIN, UNIT= **** AT (*******, *******), MINE TYPE= **
5016     ALWAYS
5017
5018     IF .TYPE NE BARRIER
5019        ''DELAYS FOR BARRIER MINEFIELDS ARE ASSESSED IN
5020        ''UPDATE LOC AND START ARTY MOVEMENT
5021        IF UN.BATTLE.INDEX(.UNIT) GT 0
5022           ''THIS WILL CAUSE DELAYS IN CHANGE LOC
5023           LET UN.DELAY(.UNIT) = MAX.F(UN.DELAY(.UNIT), .DELAY)
5024        ELSE
5025           ''THIS IS EITHER MOPMS OR ARTY FASCAM THAT ARRIVED
5026           ''AFTER A BATTLE FINISHED - SINCE THE UNIT IS STILL
5027           ''MOVING, ADJUST THE UPDATE LOC FOR THE TASK
5028           ''FORCE
5029           IF CT.GROUP(TU.CAT(UN.TYPE.UNIT(.UNIT))) = ARTILLERY
5030              FOR EVERY .SPAM IN EV.S(I.STOP.ARTY.MOVEMENT)
5031              WITH BY.UNIT(SPAM.BTRY(.SPAM)) = .UNIT
5032              FIND THE FIRST CASE
5033              IF NONE
5034                 CALL ERROR.STOP----------------------------------------------------------------------------------------->(604)
5035              ALWAYS
5036              REMOVE .SPAM FROM EV.S(I.STOP.ARTY.MOVEMENT)
5037              ADD .DELAY/MINUTES.V TO TIME.A(.SPAM)
5038              FILE .SPAM IN EV.S(I.STOP.ARTY.MOVEMENT)
5039           ELSE
5040              ''THE UNIT MOVES WITH A TASK FORCE
5041              IF MU.TF.MEM(UN.PTR(.UNIT)) GT 0
5042                 LET .TF.LEADER = MU.TF.MEM(UN.PTR(.UNIT))
5043              ELSE
5044                 LET .TF.LEADER = .UNIT
5045              ALWAYS
5046              FOR EVERY .UL IN EV.S(I.UPDATE.LOC)
5047              WITH UL.UNIT(.UL) = .TF.LEADER
```

```
5048              FIND THE FIRST CASE
5049              IF FOUND
5050                 REMOVE .UL FROM EV.S(I.UPDATE.LOC)
5051                 ADD .DELAY/MINUTES.V TO TIME.A(.UL)
5052                 FILE .UL IN EV.S(I.UPDATE.LOC)
5053              ALWAYS
5054           ALWAYS
5055        ALWAYS
5056     ALWAYS
5057
5058 <--EXITROUTINE
5059 ENDROUTINE
```

```
5060 ROUTINE MIN.MOVE                                                                                    M014
5061     GIVEN
5062     UNIT
5063
5064     ADD 1 TO ANAL.CTR(26,1)  ''                                                                  \DYN_ANAL
5065     ''CALLED FROM CHANGE.LOC AND WITHDRAW
5066
5067     NORMALLY MODE IS INTEGER
5068     DEFINE DELTA.TIME,
5069     MOVE.FACTOR, NITE.DAY.FACTOR AS REAL VARIABLES
5070
5071     LET MIN.SEGMENT = INF.C
5072     LOOP FOR EACH SEGMENT OF UN.SEGMENT.LIST(UNIT)
5073     DO THE FOLLOWING
5074        LET MIN.SEGMENT = MIN.F(MIN.SEGMENT,SEG.LENGTH(SEGMENT))
5075        IF MIN.SEGMENT < 0
5076           LET A.MIN.SEG = MIN.SEGMENT
5077           LET MIN.SEGMENT = ABS.F(A.MIN.SEG)
5078           PRINT 1 LINE WITH UNIT.NOS(UNIT), A.MIN.SEG, MIN.SEGMENT THUS
5079           = = = MINIMUM MOVE UNIT ******'S MOVE OF ***** CHANGED TO ***** HDM = = =
5080        ALWAYS
5081     ENDLOOP
5082
5083     LET MOVE.FACTOR = MOV.FAC(BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT)))
5084     LET NITE.DAY.FACTOR = 1.
5085     IF TIME.V LT BMNT OR TIME.V GT EENT
5086        LET NITE.DAY.FACTOR = NITE.MOV.FAC
5087     ALWAYS
5088     LET MOVE.FACTOR = MOVE.FACTOR * NITE.DAY.FACTOR * TAC.MOV.FAC
5089     IF UN.STATUS(UNIT) = WITHDRAWING
5090        LET MOVE.FACTOR = 3.
5091     ALWAYS
5092     LET MIN.SEGMENT = MAX.F(6,MIN.SEGMENT)
5093     LET DELTA.TIME = MIN.SEGMENT/(TU.MOV.RATE(UN.TYPE.UNIT(UNIT)) * MOVE.FACTOR)
5094     ADD UN.DELAY(UNIT)/MINUTES.V TO DELTA.TIME
5095     SCHEDULE_A MOVE------------------------------------------------------------------------------------------------------------------------>(381)
5096     GIVEN
5097     UNIT
5098     IN DELTA.TIME HOURS
5099
5100 <--EXITROUTINE
5101 ENDROUTINE
```

```
5102 ROUTINE NEW.SEGMENT                                                                                    M015
5103    GIVEN
5104    UNIT,
5105    SEGMENT
5106
5107
5108    ADD 1 TO ANAL.CTR(27,1)  ''                                                                         \DYN_ANAL
5109    NORMALLY MODE IS INTEGER
5110    DEFINE SHAPE,SCALE,W AS REAL VARIABLES
5111    DEFINE SHOOT.OUT AS AN INTEGER VARIABLE '' 25APR79_%JN
5112
5113    CALL RANGE.COMPUTE------------------------------------------------------------------------------------------------------------>(342)
5114    GIVEN
5115    UNIT,
5116    SEG.UNIT(SEGMENT)
5117    YIELDING
5118    RANGE
5119
5120    IF SEG.TYPE(SEGMENT) = YES,
5121       IF UN.STATUS(UNIT) = STATIONARY
5122          IF UN.STATUS(SEG.UNIT(SEGMENT)) = STATIONARY
5123             LET RNG. = TT.STATIONARY.LOS.BREAK(BTL.TERRAIN.TYPE
5124             (UN.BATTLE.INDEX(UNIT)))    '' 16 REMOVED %8FEB80_%RGR
5125          ELSE
5126             LET RNG. = TT.M.S.LOS.BREAK(BTL.TERRAIN.TYPE
5127             (UN.BATTLE.INDEX(UNIT)))     '' %8FEB80_%RGR
5128          ALWAYS
5129       ELSE
5130          IF UN.STATUS(SEG.UNIT(SEGMENT)) = STATIONARY
5131             LET RNG. = TT.M.S.LOS.BREAK(BTL.TERRAIN.TYPE
5132             (UN.BATTLE.INDEX(UNIT)))
5133          ELSE
5134             LET RNG. = TT.MOVING.LOS.BREAK(BTL.TERRAIN.TYPE
5135             (UN.BATTLE.INDEX(UNIT)))
5136          ALWAYS
5137       ALWAYS
5138       IF RANGE LE RNG.
5139          LET SEG.LENGTH(SEGMENT) = RNG.
5140 <--------EXITROUTINE
5141       OTHERWISE
5142       LET SEG.TYPE(SEGMENT) = NO
5143       IF DEBUG = TRUE
5144          PRINT 4 LINES WITH
5145          TIME.V,
5146          UN.BATTLE.INDEX(UNIT),
5147          UNIT.NOS(UNIT),
5148          UNIT.NOS(SEG.UNIT(SEGMENT))
5149          THUS
5150          ==========LOS BROKEN==========
5151          TIME.V = **.**** BATTLE = ******
5152          BETWEEN
5153          UNITS ****** AND ******
5154       ENDIF
5155       FOR EACH ENGAGE.MENT OF EV.S(I,ENGAGEMENT)
5156       WITH DETECTED.UNIT(ENGAGE.MENT) = UNIT AND
5157       DETECTING.UNIT(ENGAGE.MENT) = SEG.UNIT(SEGMENT)
5158       FIND THE FIRST CASE    '' %14FEB80
5159       IF FOUND
```

```
5160          CANCEL_THE ENGAGEMENT CALLED ENGAGE.MENT------------ ------------>(369)
5161          DESTROY THE ENGAGEMENT CALLED ENGAGE.MENT
5162       ENDIF        '' %14FEB80_%RGR
5163       FOR EACH ENGAGE.MENT OF EV.S(I.ENGAGEMENT)
5164       WITH DETECTED.UNIT(ENGAGE.MENT) = SEG.UNIT(SEGMENT)
5165       AND DETECTING.UNIT(ENGAGE.MENT) = UNIT
5166       FIND THE FIRST CASE     ''  %14FEB80_%RGR
5167       IF FOUND
5168          CANCEL_THE ENGAGEMENT CALLED ENGAGE.MENT------------------------------------------------------------>(369)
5169          DESTROY THE ENGAGEMENT CALLED ENGAGE.MENT
5170       ENDIF      ''   %14FEB80_%RGR
5171       LOOP
5172       FOR EACH ASSESSMENT OF EV.S(I.ASSESSMENT)
5173       WITH (AS.TGT.UNIT(ASSESSMENT) = UNIT AND
5174       AS.FIRING.UNIT(ASSESSMENT) = SEG.UNIT(SEGMENT))
5175       OR (AS.TGT.UNIT(ASSESSMENT) = SEG.UNIT(SEGMENT)
5176       AND AS.FIRING.UNIT(ASSESSMENT) = UNIT)
5177       DO THE FOLLOWING
5178          IF DEBUG = TRUE
5179             PRINT 2 LINES WITH
5180             TIME.V,
5181             UN.BATTLE.INDEX(UNIT)
5182             THUS
5183             ==========ROUNDS LOST DUE TO BREAK IN LOS==========
5184             TIME.V = **.******** BATTLE = ******
5185          ENDIF
5186          IF AS.DESTRUCT.INDIC(ASSESSMENT) NE YES
5187             INTERRUPT ASSESSMENT------------------------------------------------------------------------------>(487)
5188             LET TIME.A(ASSESSMENT) = -RINF.C
5189             LET AS.DESTRUCT.INDIC(ASSESSMENT) = YES
5190             RESUME ASSESSMENT--------------------------------------------------------------------------------->(487)
5191          ALWAYS
5192       ENDLOOP
5193       FOR EACH VISIBLE.UNIT OF UN.LOS.LIST(UNIT)
5194       WITH VU.POINTER(VISIBLE.UNIT) = SEG.UNIT(SEGMENT)
5195       FIND THE FIRST CASE
5196       IF FOUND
5197          REMOVE THE VISIBLE.UNIT FROM UN.LOS.LIST(UNIT)
5198          DESTROY THE VISIBLE.UNIT
5199       ENDIF
5200       FOR EACH VISIBLE.UNIT OF UN.LOS.LIST(SEG.UNIT(SEGMENT))
5201       WITH VU.POINTER(VISIBLE.UNIT) = UNIT
5202       FIND THE FIRST CASE
5203       IF FOUND
5204          REMOVE THE VISIBLE.UNIT FROM UN.LOS.LIST(SEG.UNIT(SEGMENT))
5205          DESTROY THE VISIBLE.UNIT
5206       ENDIF
5207       LOOP
5208       FOR EVERY UE.LINK OF UN.EQUIP.LIST(SEG.UNIT(SEGMENT))
5209       DO THE FOLLOWING
5210          LOOP
5211          FOR EVERY FIRING.TABLE OF UE.TARGET.LIST(UE.LINK)
5212          WITH FT.TGT.UNIT(FIRING.TABLE) = UNIT
5213          DO THE FOLLOWING
5214             LOOP
5215             FOR EVERY SHOOT.OUT OF SO.LIST(UE.LINK)
5216             WITH SO.FIRING.TABLE(SHOOT.OUT) = FIRING.TABLE
5217             DO THE FOLLOWING
```

```
5218                LET SO.FIRING.TABLE(SHOOT.OUT) = 0
5219             ENDLOOP
5220             REMOVE THE FIRING.TABLE FROM UE.TARGET.LIST(UE.LINK)
5221             DESTROY THE FIRING.TABLE
5222          ENDLOOP
5223       ENDLOOP
5224       LOOP
5225       FOR EVERY UE.LINK OF UN.EQUIP.LIST(UNIT)
5226       DO THE FOLLOWING
5227          LOOP
5228          FOR EVERY FIRING.TABLE OF UE.TARGET.LIST(UE.LINK)
5229          WITH FT.TGT.UNIT(FIRING.TABLE) = SEG.UNIT(SEGMENT)
5230          DO THE FOLLOWING
5231             LOOP
5232             FOR EVERY SHOOT.OUT OF SO.LIST(UE.LINK)
5233             WITH SO.FIRING.TABLE(SHOOT.OUT) = FIRING.TABLE
5234             DO THE FOLLOWING
5235                LET SO.FIRING.TABLE(SHOOT.OUT) = 0
5236             ENDLOOP
5237             REMOVE THE FIRING.TABLE FROM UE.TARGET.LIST(UE.LINK)
5238             DESTROY THE FIRING.TABLE
5239          ENDLOOP
5240       ENDLOOP
5241       LET SHAPE = TT.NLOS.SHAPE(BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT)))
5242       LET SCALE = TT.NLOS.SCALE(BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT)))
5243    ELSE
5244       LET SEG.TYPE(SEGMENT) = YES
5245       LET SHAPE = TT.LOS.SHAPE(BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT)))
5246       LET SCALE = TT.LOS.SCALE(BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT)))
5247       CREATE A VISIBLE.UNIT
5248       LET VU.POINTER(VISIBLE.UNIT) = UNIT
5249       LET VU.STATUS(VISIBLE.UNIT) = NO
5250       FILE THIS VISIBLE.UNIT IN THE UN.LOS.LIST(SEG.UNIT(SEGMENT))
5251       CREATE A VISIBLE.UNIT
5252       LET VU.POINTER(VISIBLE.UNIT) = SEG.UNIT(SEGMENT)
5253       LET VU.STATUS(VISIBLE.UNIT) = NO
5254       FILE THIS VISIBLE.UNIT IN THE UN.LOS.LIST(UNIT)
5255    ALWAYS
5256    ''UTILIZE WEIBULL.F FUNCTION IN NEXT DRAW ------------------------------------------------------------------ \1>(642)
5257    LET W = WEIBULL.F(SHAPE,SCALE,RN.SEED)
5258    LET SEG.LENGTH(SEGMENT) = MAX.F(2.,(W*TER.W.INC./16.))
5259    IF SEG.TYPE(SEGMENT) = NO AND SEG.LENGTH(SEGMENT) GT RANGE-RNG
5260       LET SEG.LENGTH(SEGMENT) = RANGE-RNG
5261    ALWAYS
5262 <--EXITROUTINE
5263 ENDROUTINE
```

```
5264 ROUTINE POSITION                                                                                                    M016
5265     GIVEN
5266     UNIT,
5267     D,
5268     DISTANCE,
5269     POINT
5270
5271     ADD 1 TO ANAL.CTR(28,1)                                                                                    \DYN_ANAL
5272     NORMALLY MODE IS INTEGER
5273     DEFINE RATIO. AS A REAL VARIABLE
5274     DEFINE NEW.SECTOR AS AN INTEGER VARIABLE
5275
5276     LET DELTA.X = P.X(POINT)-UN.X.COORD(UNIT)
5277     LET DELTA.Y = P.Y(POINT)-UN.Y.COORD(UNIT)
5278     LET RATIO. = DISTANCE/D
5279     ADD RATIO.*DELTA.X TO UN.X.COORD(UNIT)
5280     ADD RATIO.*DELTA.Y TO UN.Y.COORD(UNIT)
5281     CALL SEGMENT.ADJUST------------------------------------------------------------------------------------>(122)
5282     GIVEN
5283     UNIT,
5284     DISTANCE
5285     CALL DEQ.FEBA.SET--------------------------------------------------------------------------------------->(325)
5286     GIVEN
5287     UNIT
5288     YIELDING
5289     OLD.SECTOR
5290     CALL ENQ.FEBA.SET--------------------------------------------------------------------------------------->(331)
5291     GIVEN
5292     UNIT,
5293     OLD.SECTOR
5294     YIELDING
5295     NEW.SECTOR
5296
5297 <--EXITROUTINE
5298 ENDROUTINE
```

```
5299 ROUTINE PRED.POS                                                                                                    M017
5300    GIVEN
5301    UNIT,
5302    DELAY.TIME
5303    YIELDING
5304    D.X,
5305    D.Y
5306
5307    ADD 1 TO ANAL.CTR(29,1)  ''                                                                                \DYN_ANAL
5308    NORMALLY MODE IS INTEGER
5309    DEFINE THETA, DELAY.TIME AS REAL VARIABLES
5310
5311    IF UN.STATUS(UNIT) = STATIONARY  OR
5312       UN.STATUS(UNIT) = STA.TO.WITH
5313       LET D.X = 0
5314       LET D.Y = 0
5315 <------EXITROUTINE
5316    OTHERWISE
5317
5318    IF UN.STATUS(UNIT) = ADVANCING  OR
5319       UN.STATUS(UNIT) = ADV.TO.WITH
5320       LET N = 0
5321    ELSE
5322       LET N = 1
5323    ALWAYS
5324    LOOP FOR EVERY POINT OF UN.PATH(UNIT)
5325    DO THE FOLLOWING
5326       IF N GE UN.POSITION.INDEX(UNIT)
5327    <------LEAVE
5328       OTHERWISE
5329
5330       ADD 1 TO N
5331    ENDLOOP
5332    LET D = TU.MOV.RATE(UN.TYPE.UNIT(UNIT)) * DELAY.TIME / 60
5333    LET DELTA.X = P.X(POINT) - UN.X.COORD(UNIT)
5334    LET DELTA.Y = P.Y(POINT) - UN.Y.COORD(UNIT)
5335    CALL ANGLE.COMPUTE-------------------------------------------------------------------------------------------->(644)
5336    GIVEN
5337    DELTA.X,
5338    DELTA.Y
5339    YIELDING
5340    THETA
5341    LET D.X = D * COS.F(THETA)
5342    LET D.Y = D * SIN.F(THETA)
5343
5344 <--EXITROUTINE
5345 ENDROUTINE
```

```
5346 ROUTINE PREPARE.LIST                                                    M018
5347     GIVEN
5348     EN.UNITS
5349     YIELDING
5350     DEF.UNITS
5351
5352     ADD 1 TO ANAL.CTR(30,1)  ''                                        \DYN_ANAL
5353     ''THIS ROUTINE PREPARES A LIST OF ALL DEFENDING UNITS
5354     ''IN A BATTLE.  THIS LIST CONSISTS OF UNITS FOUND IN
5355     ''PROXIMITY OF THE ATTACKERS AND OTHER UNITS BELONGING
5356     ''TO THE SAME TASK FORCE.  IF THOSE UNITS ARE NOT
5357     ''CURRENTLY EXECUTING A DEFEND ORDER, A DEFEND ORDER
5358     ''IS FOUND IN THEIR ORDER SET AND THAT BECOMES THEIR
5359     ''CURRENT ORDER.
5360
5361     NORMALLY MODE IS INTEGER
5362     DEFINE EN.UNITS., DEF.UNITS. AS 1-DIMENSIONAL ARRAYS
5363     DEFINE DUM.UNITS. AS A 1-DIMENSIONAL ARRAY
5364
5365     RESERVE DUM.UNITS. AS 100
5366
5367     LET NO.TF=0
5368     LOOP FOR COUNT = 1 TO DIM.F(EN.UNITS.(*))
5369     DO
5370        IF MU.TF.MEM(UN.PTR(EN.UNITS.(COUNT)))=0
5371           LET UNT.=EN.UNITS.(COUNT)
5372        ELSE
5373           ''UNIT IS A MEMBER OF A TASK FORCE
5374           LET A.TF.LDR = EN.UNITS.(COUNT)
5375           LET B.TF.LDR = UN.PTR(A.TF.LDR)
5376           LET UNT.=MU.TF.MEM(B.TF.LDR)
5377        ALWAYS
5378        CALL CHECK.LIST------------------------------------------------------------------------------------------>(137)
5379        GIVEN
5380        UNT.,
5381        NO.TF,
5382        DUM.UNITS.(*)
5383        YIELDING
5384        RESULT
5385        FOR EVERY UP.L IN EV.S(I.UPDATE.LOC)
5386        WITH UL.UNIT(UP.L) = UNT.
5387        FIND THE FIRST CASE
5388        IF FOUND
5389           CANCEL_THE UPDATE.LOC CALLED UP.L--------------------------------------------------------------------->(408)
5390           DESTROY THE UPDATE.LOC CALLED UP.L
5391        ALWAYS
5392        FOR EVERY S.M IN EV.S(I.START.MOVE)
5393        WITH SM.UNIT(S.M) = UNT.
5394        FIND THE FIRST CASE
5395        IF FOUND
5396           CANCEL_THE START.MOVE CALLED S.M---------------------------------------------------------------------->(403)
5397           DESTROY THE START.MOVE CALLED S.M
5398        ALWAYS
5399        IF RESULT > 0
5400           ADD RESULT TO NO.TF
5401           IF UNT. GE 3000
5402              '' MAX SUBSCRIPT VALUE IS 3000
5403              LET POINT.ER = UNT.
```

```
5404            LET UNT. = MU.UNIT.ID(POINT.ER)
5405            PRINT 1 LINE WITH UNT., POINT.ER THUS
5406            = = =PREPARE.LIST UNIT **** (POINTER******) WAS IN ERROR = = =
5407         ALWAYS
5408         FOR EVERY .ORDER IN MU.ORDER.SET(UN.PTR(UNT.))
5409         WITH ORD.SEQ.NO(.ORDER)=MU.CUR.ORDER(UN.PTR(UNT.))
5410         FIND THE FIRST CASE
5411         IF FOUND
5412            ''UTILIZE COLLISION FUNCTION IN NEXT TEST ---------------------------------------------------------- \1>(630)
5413            IF ORD.TYPE(.ORDER) = "MOVCOR"  AND
5414               COLLISION(UNT.,.ORDER)=YES
5415   <-----------CYCLE
5416            OTHERWISE
5417
5418            IF ORD.TYPE(.ORDER) NE "DEF"
5419               FOR EVERY ..ORDER IN MU.ORDER.SET(UN.PTR(UNT.))
5420               WITH ORD.TYPE(..ORDER) = "DEF"
5421               FIND THE FIRST CASE
5422               IF FOUND
5423                  LET MU.CUR.ORDER(UN.PTR(UNT.)) =
5424                  ORD.SEQ.NO(..ORDER)
5425                  LET UN.MISSION(UNT.) =
5426                  ORD.MISSION(ORD.ID(..ORDER))
5427                  LOOP FOR EACH MU
5428                  IN MU.TF.LIST(UN.PTR(UNT.))
5429                  DO
5430                     LET UN.MISSION(MU.UNIT.ID(MU)) =
5431                     ORD.MISSION(ORD.ID(..ORDER))
5432                  ENDLOOP
5433               ELSE
5434                  CALL ERROR.STOP-------------------------------------------------------------------------------->(604)
5435               ALWAYS
5436            ALWAYS
5437         ELSE
5438            CALL ERROR.STOP-------------------------------------------------------------------------------->(604)
5439         ALWAYS
5440      ALWAYS
5441   ENDLOOP
5442   IF NO.TF > 100
5443      PRINT 1 LINE WITH NO.TF AND TIME.V THUS
5444      $$$$$$ BATTLE WITH NO.TF = **** AT TIME ***.****
5445   ALWAYS
5446   IF NO.TF > 150
5447      TRACE
5448 <-----STOP
5449   OTHERWISE
5450
5451   RESERVE DEF.UNITS. AS NO.TF
5452
5453   LOOP FOR COUNT= 1 TO NO.TF
5454   DO
5455      LET DEF.UNITS.(COUNT)=DUM.UNITS.(COUNT)
5456   ENDLOOP
5457
5458   RELEASE DUM.UNITS.(*)
5459
5460 <--EXITROUTINE
5461 ENDROUTINE
```

```
5462 ROUTINE PREP.WITHDRAW                                                                                    M019
5463    GIVEN
5464    UNIT
5465
5466    NORMALLY MODE IS INTEGER
5467
5468    ADD 1 TO ANAL.CTR(31,1)  ''                                                                  \DYN_ANAL
5469    IF UN.STATUS(UNIT) = WITHDRAWING
5470       OR UN.STATUS(UNIT) = ADV.TO.WITH
5471       OR UN.STATUS(UNIT) = STA.TO.WITH
5472       CALL ERROR.STOP-------------------------------------------------------------------------------->(604)
5473    ALWAYS
5474
5475    IF UN.STATUS(UNIT) = STATIONARY
5476       LET UN.TIME.LAST.MOVE(UNIT) = TIME.V
5477       LET UN.STATUS(UNIT) = STA.TO.WITH
5478    ELSE
5479       ''UN.STATUS(UNIT) = ADVANCING
5480       LET UN.STATUS(UNIT) = ADV.TO.WITH
5481       FOR EACH MOVE OF EV.S(I.MOVE)
5482       WITH MV.UNIT(MOVE) = UNIT
5483       FIND THE FIRST CASE
5484       IF NONE
5485          CALL ERROR.STOP------------------------------------------------------------------------------>(604)
5486       ALWAYS
5487
5488       CANCEL_THE MOVE------------------------------------------------------------------------------->(381)
5489       SCHEDULE_THE MOVE '' ------------------------------------------------------------ \CLEANUP>(381)
5490       GIVEN
5491       UNIT NOW ''                                                                                      \1
5492    ALWAYS
5493
5494 <--EXITROUTINE
5495 ENDROUTINE
```

```
5496 ROUTINE PROX.CHECK                                                                                    M020
5497     GIVEN
5498     CHECK.UNIT,
5499     RANGE
5500
5501     ADD 1 TO ANAL.CTR(32,1)  ''                                                                       \DYN_ANAL
5502     ''THIS ROUTINE WAS RECODED %19MAR80_%RGR TO ALLOW THE
5503     ''RANGE AT WHICH A DEFENDING UNIT BEGINS TO WITHDRAW TO VARY BY
5504     ''THE TYPE TERRAIN THAT THE BATTLE IS FOUGHT ON. IF THE UNIT HAS
5505     ''A DEFEND MISSION, IT IS EXPECTED TO HOLD THAT POSTION UNTILL THE
5506     ''LAST MOMENT BEFORE WITHDRAWING. CONSEQUENTLY THE TERRAIN RANGE
5507     ''VALUE IS FACTORED DOWN TO ENABLE THE ATTACKER TO CLOSE WITH
5508     ''THE DEFENDER. IF THE UNIT MISSION IS TO DELAY, THEN IT BEGINS
5509     ''IT'S WITHDRAW AT THE TERRAIN RANGE VALUE. THE PREVIOUSLY USED
5510     ''VALUE WAS 1500 METERS SPECIFIED AS A GLOBIAL VARIABLE SET IN
5511     ''THE MAIN ROUTINE. THIS VALUE DID NOT ALLOW FOR TERRAIN TYPE OR
5512     ''THE MISSION OF THE DEFENSING UNIT.
5513
5514     NORMALLY MODE IS INTEGER
5515     DEFINE MIN.R, MAX.R AS REAL VARIABLES
5516
5517     IF UN.STATUS(CHECK.UNIT) EQ WITHDRAWING
5518        OR UN.STATUS(CHECK.UNIT) EQ STA.TO.WITH
5519        OR UN.STATUS(CHECK.UNIT) EQ ADV.TO.WITH
5520 <------EXITROUTINE
5521     OTHERWISE
5522
5523     IF UN.POSITION.INDEX(CHECK.UNIT) = 1
5524 <------EXITROUTINE
5525     ALWAYS
5526     LET BTL.NUMBER = UN.BATTLE.INDEX(CHECK.UNIT)
5527     LET T.TYPE = BTL.TERRAIN.TYPE(BTL.NUMBER)
5528     LET RANGE.1 = TT.STATIONARY.LOS.BREAK(T.TYPE)
5529     IF UN.MISSION(CHECK.UNIT) = DEFEND
5530        LET MAX.R = RANGE.1 * 0.667
5531        LET MIN.R = RANGE.1 * 0.333
5532        LET BREAK.OFF.RANGE = UNIFORM.F(MIN.R,MAX.R,RN.SEED)
5533     ELSE   '' THE MISSION IS NOT DEFEND
5534        IF UN.MISSION(CHECK.UNIT) = DELAY
5535           LET BREAK.OFF.RANGE = RANGE.1
5536        ELSE   '' THE MISSION IS OTHER THAN DELAY OR DEFEND
5537 <----------EXITROUTINE    '' BECAUSE ATTACKER'S ARE ADDRESSED IN CHECK.FORCE
5538        ALWAYS   '' FOR THOSE WHO ARE DEFENDERS
5539     ALWAYS
5540
5541     IF RANGE <= WD.DIST(UN.MISSION(CHECK.UNIT),UN.COLOR(CHECK.UNIT))/16.
5542        PRINT 2 LINES WITH
5543        TIME.V,
5544        BTL.SEQ.NO(UN.BATTLE.INDEX(CHECK.UNIT)),
5545        UNIT.NOS(CHECK.UNIT), RANGE*16
5546        THUS
5547        ----------RANGE PROXIMITY THRESHOLD BROKEN----------
5548        TIME.V = **.******** BATTLE = ******  UNIT = ******  RANGE = ******
5549
5550        CALL PREP.WITHDRAW------------------------------------------------------------------------------------------------------------>(114)
5551        GIVEN
5552        CHECK.UNIT
5553        ACTIVATE_A WITH.DRAW---------------------------------------------------------------------------------------------------------->(478)
```

```
5554        GIVEN
5555        CHECK.UNIT NOW ''                                                                    \1
5556        CALL CHECK.FORCE------------------------------------------------------------------->(133)
5557        GIVEN
5558        CHECK.UNIT
5559     ALWAYS
5560
5561 <--EXITROUTINE
5562 ENDROUTINE
```

```
5563 ROUTINE PROX.POS                                                    '' M021
5564    GIVEN
5565    UNT.,
5566    ORDR.,
5567    EN.UNITS.
5568
5569    ADD 1 TO ANAL.CTR(33,1)  ''                                                    \DYN_ANAL
5570    ''THIS ROUTINE IS CALLED WHEN A MOVING COMBAT UNIT IS IN CLOSE
5571    ''  PROXIMITY TO AN ENEMY UNIT.  IT CHECKS THE STRENGTH OF THE
5572    ''  MOVING UNIT AND SELECTS THE APPROPRIATE NEXT ORDER.
5573
5574    NORMALLY MODE IS INTEGER
5575    DEFINE EN.UNITS. AS AN INTEGER 1-DIMENSIONAL ARRAY
5576
5577    CALL CHECK.STREN------------------------------------------------------------------------------------------>(141)
5578    GIVEN
5579    UNT.
5580    YIELDING
5581    PCT.ON.HAND
5582    IF PCT.ON.HAND.>=THRESH.REIN(ORD.ID(ORDR.))
5583       LET NEXT. = NX.ORD.ABOVE(ORD.ID(ORDR.))
5584    ELSE
5585       LET NEXT. = NX.ORD.BELOW(ORD.ID(ORDR.))
5586    ALWAYS
5587    LET FLAG = 1
5588    SCHEDULE_A GET.NX.ORD-------------------------------------------------------------------------------------->(373)
5589    GIVEN
5590    UNT.,
5591    NEXT.,
5592    EN.UNITS.(*),
5593    FLAG NOW ''                                                                    \1
5594
5595 <--EXITROUTINE
5596 ENDROUTINE
```

```
5597 ROUTINE REIN.ARRIVE                                                                                    M022
5598     GIVEN
5599     UNT.,
5600     ORDR.
5601
5602     ADD 1 TO ANAL.CTR(34,1)  ''                                                                        \DYN_ANAL
5603     ''THIS ROUTINE IS ACTIVATED WHEN A REINFORCING UNIT ARRIVES AT THE
5604     ''     LOCATION OF THE REINFORCED UNIT.  THE REINFORCING UNIT OR UNITS
5605     ''     ARE ADDED TO THE TASK FORCE LIST OF THE REINFORCED UNIT AND
5606     ''     ASSUME THE ORDER SET OF THE REINFORCED UNIT.  IF THE REIN-
5607     ''     FORCED UNIT IS ENGAGED IN CLOSE COMBAT AT THE TIME THE
5608     ''     BATTLE IS INTERRUPTED SO THAT THE REINFORCING UNITS CAN
5609     ''     BECOME A PART OF IT.
5610
5611     NORMALLY MODE IS INTEGER
5612     DEFINE EN.UNITS. AS A 1-DIMENSIONAL ARRAY
5613
5614     LET UNIT.REIN.=REINFORCED.UN(ORD.ID(ORDR.))
5615     LET EN.UNITS.(*)=ATK.UNITS.PTR(ORD.ID(ORDR.))
5616     IF UN.BATTLE.INDEX(UNIT.REIN.)>0
5617        RELEASE EN.UNITS.(*)
5618        CALL INTER.BATTLE------------------------------------------------------------------------------>(146)
5619        GIVEN
5620        UNT.,
5621        UN.BATTLE.INDEX(UNIT.REIN),
5622        ORDR.
5623     ELSE
5624        FOR EVERY ORDER IN MU.ORDER.SET(UN.PTR(UNIT.REIN.))
5625        WITH ORD.SEQ.NO(ORDER)=MU.CUR.ORDER(UN.PTR(UNIT.REIN.))
5626        FIND THE FIRST CASE
5627        IF FOUND
5628           IF ORD.TYPE(ORDER)="REINF"
5629              LET NEXT.=SUC.REINF.OP(ORD.ID(ORDER))
5630              LET FLAG = 1
5631              SCHEDULE_A GET.NX.ORD------------------------------------------------------------------->(373)
5632              GIVEN
5633              UNIT.REIN.,
5634              NEXT.,
5635              EN.UNITS.(*),
5636              FLAG NOW ''                                                                                \1
5637           ELSE
5638              CALL ERROR.STOP------------------------------------------------------------------------->(604)
5639           ALWAYS
5640        ELSE
5641           CALL ERROR.STOP------------------------------------------------------------------------->(604)
5642        ALWAYS
5643     ALWAYS
5644
5645 <--EXITROUTINE
5646 ENDROUTINE
```

```
5647 ROUTINE RESET.FEBA.SECTOR                                                                                    M023
5648    GIVEN
5649    SIDE,
5650    OLD.SECTOR
5651
5652    ADD 1 TO ANAL.CTR(35,1)  ''                                                                              \DYN_ANAL
5653    NORMALLY MODE IS INTEGER
5654    DEFINE UNIT, SIDE , SECTOR, OLD.SECTOR AS INTEGER VARIABLES
5655
5656    IF DEBUG=TRUE,
5657       PRINT 1 LINE WITH SIDE AND OLD.SECTOR THUS
5658       RESET.FEBA.SECTOR SIDE=***, OLD.SECTOR=***
5659    ALWAYS
5660
5661    LOOP UNTIL SS.SET( SIDE,OLD.SECTOR ) IS NOT EMPTY,
5662    DO
5663       LET CANDIDATE.REAR = SS.REAR(SIDE,OLD.SECTOR)
5664       - FRONT.DEPTH
5665       IF SIDE = RED
5666          IF CANDIDATE.REAR LT (-2)*INIT.X.FEBA
5667    <----------LEAVE
5668          OTHERWISE
5669       ELSE
5670          IF CANDIDATE.REAR LT 0
5671    <----------LEAVE
5672          OTHERWISE
5673       ALWAYS
5674       LET SS.REAR(SIDE,OLD.SECTOR) = CANDIDATE.REAR
5675       LOOP FOR EACH UNIT IN THE UNIT.SET(SIDE,MANEUVER)
5676       WHILE UN.MISSION(UNIT) NE PATROL
5677       WHILE MU.CUR.ORDER(UN.PTR(UNIT)) NE 0 OR
5678       MU.TF.MEM(UN.PTR(UNIT))>0
5679       WHILE M.SS.SET(UNIT) = 0
5680       DO THIS
5681          CALL LOCATE.SECTOR------------------------------------------------------------------------------------>(100)
5682          GIVEN
5683          UN.Y.COORD(UNIT)
5684          YIELDING
5685          SECTOR
5686          IF UN.COLOR(UNIT) = RED
5687             LET UN.FWD.ADV = - UN.X.COORD(UNIT)
5688          ELSE
5689             LET UN.FWD.ADV = UN.X.COORD(UNIT)
5690          ALWAYS
5691          IF SECTOR = OLD.SECTOR
5692             IF UN.FWD.ADV  >  SS.REAR(SIDE,OLD.SECTOR)
5693                FILE UNIT IN SS.SET(SIDE,OLD.SECTOR)
5694             ALWAYS
5695          ELSE
5696             IF UN.FWD.ADV  >  SS.REAR(SIDE,SECTOR)
5697                FILE UNIT IN SS.SET(SIDE,SECTOR)
5698             ALWAYS
5699          ALWAYS
5700       ENDLOOP
5701    ENDLOOP
5702
5703 <--EXITROUTINE
5704 ENDROUTINE
```

```
5705 ROUTINE SEARCH                                                    ''M024
5706     GIVEN
5707     .DETECTING.WPN,
5708     .RANGE.HDM,
5709     .FRAC.TARGET.VISIBLE,
5710     .DETECTED.TE,
5711     .NUM.SEARCHES
5712     YIELDING
5713     .TIME
5714
5715     ADD 1 TO ANAL.CTR(36,1)  ''                                                    \DYN_ANAL
5716     NORMALLY MODE IS REAL
5717     DEFINE .DETECTING.WPN, .DETECTED.TE, .RANGE, .RANGE.HDM, .NUM.SEARCHES
5718     AS INTEGER VARIABLES
5719
5720     IF .NUM.SEARCHES = 0
5721        LET .TIME = RINF.C
5722 <------RETURN
5723     OTHERWISE
5724
5725     LET .RANGE = .RANGE.HDM ''HEXADECAMETERS'' * 16
5726     IF TW.TYPE.OF.SENSOR(.DETECTING.WPN) = 1
5727        ''EYE.
5728        CALL CONTRAST.TO.FREQ-------------------------------------------------------->(323)
5729        GIVEN
5730        .RANGE
5731        YIELDING
5732        .FREQUENCY
5733     ELSE
5734        ''TANK THERMAL SIGHT (TTS) OR TOW INFRA-RED.
5735        CALL TEMPERATURE.ATTENUATION------------------------------------------------>(344)
5736        GIVEN
5737        .DETECTED.TE,
5738        .RANGE
5739        YIELDING
5740        .DELTA.T.PRIME
5741
5742        CALL MRT.TO.FREQ------------------------------------------------------------>(646)
5743        GIVEN
5744        .DELTA.T.PRIME
5745        YIELDING
5746        .FREQUENCY
5747     ALWAYS
5748
5749     CALL JOHNSON.CRITERIA-------------------------------------------------------->(661)
5750     YIELDING
5751     .NO.BARS
5752
5753     CALL PROB.INF---------------------------------------------------------------->(340)
5754     GIVEN
5755     .NO.BARS,
5756     .FREQUENCY,
5757     .RANGE,
5758     .FRAC.TARGET.VISIBLE,
5759     .DETECTED.TE
5760     YIELDING
5761     .PROBABILITY.INF
5762
```

```
5763     CALL PROB.TIME------------------------------------------------------------------------------------------>(341)
5764     GIVEN
5765     .DETECTING.WPN,
5766     .PROBABILITY.INF,
5767     .NUM.SEARCHES
5768     YIELDING
5769     .TIME
5770
5771 <--EXITROUTINE
5772 ENDROUTINE
```

```
5773 ROUTINE SEGMENT.ADJUST                                                                                    M025
5774    GIVEN
5775    UNIT,
5776    D
5777
5778    ADD 1 TO ANAL.CTR(37,1)                                                                                \DYN_ANAL
5779    NORMALLY MODE IS INTEGER
5780
5781    LOOP FOR EVERY SEGMENT OF UN.SEGMENT.LIST(UNIT)
5782    DO THE FOLLOWING
5783       SUBTRACT D FROM SEG.LENGTH(SEGMENT)
5784       IF SEG.LENGTH(SEGMENT) LE 0,
5785          CALL NEW.SEGMENT-------------------------------------------------------------------------------->(107)
5786          GIVEN
5787          UNIT,
5788          SEGMENT
5789       ALWAYS
5790    ENDLOOP
5791
5792 <--EXITROUTINE
5793 ENDROUTINE
```

```
5794 ROUTINE TIME.TO.DETECT                                                                                    M026
5795    GIVEN
5796    .DETECTING.UNIT,
5797    .DETECTED.UNIT
5798
5799    ADD 1 TO ANAL.CTR(38,1)                                                                                \DYN_ANAL
5800    NORMALLY MODE IS INTEGER
5801    DEFINE .TIME.TO.DET, .DETECT.TIME AS REAL VARIABLES
5802
5803    CALL RANGE.COMPUTE------------------------------------------------------------------------------------>(342)
5804    GIVEN
5805    .DETECTED.UNIT,
5806    .DETECTING.UNIT
5807    YIELDING
5808    .RANGE
5809    IF .RANGE = 0
5810       LET .RANGE = 1
5811       PRINT 1 LINE WITH .DETECTING.UNIT, .DETECTED.UNIT THUS
5812       ===TIME.TO.DETE  RANGE BETWEEN **** AND **** IS ZERO
5813    ALWAYS
5814    LET .TIME.TO.DET = RINF.C
5815
5816    LOOP FOR EVERY .UEL OF UN.EQUIP.LIST(.DETECTED.UNIT)
5817    WITH UE.QUANT(.UEL) GT 0
5818    DO
5819       LOOP FOR EVERY .LINK IN UN.EQUIP.LIST(.DETECTING.UNIT)
5820       WITH UE.QUANT(.LINK) GT 0
5821       DO
5822          LOOP FOR EVERY .WEAPON OF UE.WEAPON.SET(.LINK)
5823          WITH WPN.QUANTITY(.WEAPON) GT 0 AND
5824          TW.NO.SENSORS(WPN.ID(.WEAPON)) GT 0 AND
5825          TW.MIN.RANGE(WPN.ID(.WEAPON)) LE .RANGE AND
5826          .RANGE LE TW.MAX.RANGE(WPN.ID(.WEAPON))
5827          DO
5828             LET .COUNT = UE.QUANT(.LINK)
5829             * WPN.QUANTITY(.WEAPON)
5830             * TW.NO.SENSORS(WPN.ID(.WEAPON))
5831             CALL SEARCH------------------------------------------------------------------------------>(120)
5832             GIVEN
5833             WPN.ID(.WEAPON),
5834             .RANGE,
5835             RANDOM.F(RN.SEED),
5836             EQ.TE.PTR(UE.ID(.UEL)),
5837             .COUNT
5838             YIELDING
5839             .DETECT.TIME
5840             LET .TIME.TO.DET = MIN.F(.TIME.TO.DET,
5841             .DETECT.TIME)
5842          ENDLOOP
5843       ENDLOOP
5844    ENDLOOP
5845
5846    IF DEBUG = YES
5847       PRINT 1 LINE WITH .DETECTING.UNIT, .DETECTED.UNIT,
5848       .RANGE, .TIME.TO.DET  THUS
5849       ===TIME.TO.DETE DETECTING=****, DETECTED=****, RANGE=******,TIME=****.***
5850    ALWAYS
5851
```

```
5852     FOR EVERY ENGAGEMENT OF EV.S(I.ENGAGEMENT)
5853     WITH DETECTING.UNIT(ENGAGEMENT) = .DETECTING.UNIT
5854     AND DETECTED.UNIT(ENGAGEMENT) = .DETECTED.UNIT
5855     FIND THE FIRST CASE
5856     IF FOUND,
5857        CANCEL_THE ENGAGEMENT----------------------------------------------------------------------------------------->(369)
5858        RESCHEDULE_THE ENGAGEMENT------------------------------------------------------------------------------------->(369)
5859        GIVEN
5860        .DETECTING.UNIT,
5861        .DETECTED.UNIT
5862        IN .TIME.TO.DET SECONDS
5863     ELSE
5864        SCHEDULE_AN ENGAGEMENT---------------------------------------------------------------------------------------->(369)
5865        GIVEN
5866        .DETECTING.UNIT,
5867        .DETECTED.UNIT
5868        IN .TIME.TO.DET SECONDS
5869     ALWAYS
5870
5871 <--EXITROUTINE
5872 ENDROUTINE
```

```
5873 ROUTINE WHAT.NEXT                                                                                    M027
5874     GIVEN
5875     WIN.,
5876     BTL.UNITS.
5877
5878     ADD 1 TO ANAL.CTR(39,1)  ''                                                                      \DYN_ANAL
5879     ''THIS ROUTINE FINDS THE POINTER TO THE NEXT ORDER IN THE
5880     ''  ORDER SET OF UNITS COMING OUT OF A SMALL UNIT BATTLE.
5881     ''  IT THEN SCHEDULES A GET.NX.ORD TO SELECT AND EXECUTE
5882     ''  THE ORDER.
5883
5884     NORMALLY MODE IS INTEGER
5885     DEFINE BTL.UNITS. AS A 1-DIMENSIONAL ARRAY
5886     DEFINE ORDER AS A VARIABLE
5887
5888     LET DUMMY = 0
5889     LET FLAG = 0
5890     LOOP FOR I = 1 TO DIM.F(BTL.UNITS.(*))
5891     DO
5892        LET MU = UN.PTR(BTL.UNITS.(I))
5893        LET PARENT = UN.PARENT(BTL.UNITS.(I))
5894        LET TF.LDR = MU.TF.MEM(MU)
5895        IF PARENT NE TF.LDR AND TF.LDR GT 0
5896           IF MU.ORDER.SET(MU) IS NOT EMPTY
5897              REMOVE MU FROM MU.TF.LIST(UN.PTR(TF.LDR))
5898              LET MU.TF.MEM(MU) = 0
5899           ELSE
5900              IF UN.PTR(PARENT) = 0
5901                 CALL ERROR.STOP------------------------------------------------------------------------------------------------------>(604)
5902              ELSE
5903                 IF MU.ORDER.SET(UN.PTR(PARENT)) IS NOT EMPTY
5904                    REMOVE MU FROM MU.TF.LIST(UN.PTR(TF.LDR))
5905                    FILE MU IN MU.TF.LIST(UN.PTR(PARENT))
5906                    LET MU.TF.MEM(MU) = PARENT
5907                 ELSE
5908                    LET GRAND.PARENT = UN.PARENT(PARENT)
5909                    IF UN.PTR(GRAND.PARENT) = 0
5910                       CALL ERROR.STOP------------------------------------------------------------------------------------------------->(604)
5911                    ELSE
5912                       IF MU.ORDER.SET(UN.PTR(GRAND.PARENT))
5913                          IS NOT EMPTY
5914                          AND GRAND.PARENT NE TF.LDR
5915                          REMOVE MU FROM
5916                          MU.TF.LIST(UN.PTR(TF.LDR))
5917                          FILE MU IN
5918                          MU.TF.LIST(UN.PTR(GRAND.PARENT))
5919                          LET MU.TF.MEM(MU) = GRAND.PARENT
5920                       ALWAYS
5921                    ALWAYS
5922                 ALWAYS
5923              ALWAYS
5924           ALWAYS
5925        ALWAYS
5926     ENDLOOP
5927
5928     LOOP FOR I=1 TO DIM.F(BTL.UNITS.(*))
5929     DO
5930        IF MU.TF.MEM(UN.PTR(BTL.UNITS.(I)))=0
```

```
5931          FOR EVERY ORDER IN MU.ORDER.SET(UN.PTR(BTL.UNITS.(I)))
5932          WITH ORD.SEQ.NO(ORDER) =
5933          MU.CUR.ORDER(UN.PTR(BTL.UNITS.(I)))
5934          FIND THE FIRST CASE
5935          IF FOUND
5936             IF ORD.TYPE(ORDER)="DEF"
5937                IF WIN=YES
5938                   LET NEXT.=EN.DIS.OP(ORD.ID.(ORDER))
5939                ELSE
5940                   LET NEXT.=OWN.DIS.OP(ORD.ID.(ORDER))
5941                ALWAYS
5942             ELSE
5943                IF ORD.TYPE(ORDER)="ATK"
5944                   IF WIN=YES
5945                      LET NEXT.=ENEMY.DO(ORD.ID(ORDER))
5946                   ELSE
5947                      LET NEXT.=OWN.DO(ORD.ID(ORDER))
5948                   ALWAYS
5949                ELSE
5950                   PRINT 1 LINE WITH BTL.UNITS.(I),
5951                   UNIT.NOS(BTL.UNITS.(I)) THUS
5952                   ****(UNIT ******) HAS NO ATK OR DEF ORDER
5953                   LIST ATTRIBUTES OF EACH ORDER
5954                   IN MU.ORDER.SET(UN.PTR(BTL.UNITS.(I)))
5955                   PRINT 1 LINE WITH
5956                   MU.CUR.ORDER(UN.PTR(BTL.UNITS.(I))) THUS
5957                   CURRENT ORDER = ******
5958                   CALL ERROR.STOP------------------------------------------------------------------------------>(604)
5959                ALWAYS
5960             ALWAYS
5961             SCHEDULE_A GET.NX.ORD------------------------------------------------------------------------------>(373)
5962             GIVEN
5963             BTL.UNITS.(I),
5964             NEXT.,
5965             DUMMY,
5966             FLAG NOW ''                                                                                    \1
5967          ELSE
5968             PRINT 1 LINE WITH
5969             BTL.UNITS.(I), UNIT.NOS(BTL.UNITS.(I)) THUS
5970             ****( UNIT ******) IS TF.LDR W/O ORDERS
5971             CALL ERROR.STOP------------------------------------------------------------------------------>(604)
5972          ALWAYS
5973       ALWAYS
5974    ENDLOOP
5975
5976 <--EXITROUTINE
5977 ENDROUTINE
```

DIRECT FIRE ROUTINES

```
5984 ROUTINE BTL.CHECK                                                                                                    DF01
5985    GIVEN
5986    BC.FARRP,
5987    SCT.UE.ID,
5988    ATK.UE.ID
5989
5990    ADD 1 TO ANAL.CTR(40,1)  ''                                                                                \DYN_ANAL
5991    ''THIS ROUTINE ATTEMPTS TO ASSIGN A DISENGAGED FARRP TO THE HIGHEST
5992    ''PRIORITY BATTLE THAT IT'S HELICOPTERS CAN REACH.
5993
5994    NORMALLY MODE IS INTEGER
5995
5996    LET BC.SIDE = UN.COLOR(FP.UNIT(BC.FARRP))
5997    LET NEW.PRIORITY = 0
5998    LET NEW.BTL = 0
5999
6000    IF HC.DEBUG = "YES"
6001       PRINT 1 LINE WITH BC.FARRP AS FOLLOWS
6002       ----BTL.CHECK----  FARRP=******
6003    ALWAYS
6004
6005    IF BC.SIDE = BLUE
6006       LOOP FOR EVERY BTL IN THE BATTLE.SET
6007       WITH BLUE.HB.PRIORITY(BTL) > 0
6008       DO THE FOLLOWING
6009          CALL HC.COMPUTE.TIMES------------------------------------------------------------------------------------>(298)
6010          GIVING
6011          BC.FARRP,
6012          BTL,
6013          SCT.UE.ID,
6014          ATK.UE.ID
6015          YIELDING
6016          FLIGHT.TIME,
6017          LOITER.TIME
6018          IF BLUE.HB.PRIORITY(BTL) > NEW.PRIORITY
6019             AND LOITER.TIME > 0
6020             LET NEW.PRIORITY = BLUE.HB.PRIORITY(BTL)
6021             LET NEW.BTL = BTL
6022          ALWAYS
6023       ENDLOOP
6024    ELSE
6025       LOOP FOR EVERY BTL IN THE BATTLE.SET
6026       WITH RED.HB.PRIORITY(BTL) > 0
6027       DO THE FOLLOWING
6028          CALL HC.COMPUTE.TIMES------------------------------------------------------------------------------------>(298)
6029          GIVING
6030          BC.FARRP,
6031          BTL,
6032          SCT.UE.ID,
6033          ATK.UE.ID
6034          YIELDING
6035          FLIGHT.TIME,
6036          LOITER.TIME
6037          IF RED.HB.PRIORITY(BTL) > NEW.PRIORITY
6038             AND LOITER.TIME > 0
6039             LET NEW.PRIORITY = RED.HB.PRIORITY(BTL)
6040             LET NEW.BTL = BTL
6041          ALWAYS
```

```
6042        ENDLOOP
6043     ALWAYS
6044
6045     IF NEW.BTL > 0
6046        LET FP.BATTLE(BC.FARRP) = NEW.BTL
6047        IF BC.SIDE = BLUE
6048           LET BLUE.HB.PRIORITY(NEW.BTL) = -1
6049           LET BTL.BL.FARRP(NEW.BTL) = BC.FARRP
6050        ELSE
6051           LET RED.HB.PRIORITY(NEW.BTL) = -1
6052           LET BTL.RD.FARRP(NEW.BTL) = BC.FARRP
6053        ALWAYS
6054        CALL EMPLOY.HELICOPTERS------------------------------------------------------------------------------------------------------------>(289)
6055        GIVING
6056        BC.FARRP,
6057        NEW.BTL
6058
6059        IF HC.DEBUG = "YES"
6060           PRINT 1 LINE WITH BC.FARRP, NEW.BTL AS FOLLOWS
6061           ----BTL.CHECK----  FARRP=******  NEW.BTL=******
6062        ALWAYS
6063     ALWAYS
6064
6065 <--EXITROUTINE
6066 ENDROUTINE
```

```
6067 ROUTINE CHECK.DEAD(UNITS.)  ''                                                    DF02
6068    YIELDING STATUS., NEW.UNITS
6069
6070    ADD 1 TO ANAL.CTR(41,1)  ''                                                    \DYN_ANAL
6071    ''  THIS ROUTINE CHECKS A LIST OF BATTLEING MANUEVER UNITS TO DETERMINE
6072    ''  WHICH IF ANY ARE DECIMATED.  FOR DECIMATED UNITS
6073    ''  IT CALLS DEAD.UNIT, AND IT PLACES NON-DECIMATED
6074    ''  UNITS ON A NEW LIST AND RETURNS IT TO THE CALLING ROUTINE(BTL.ENDED)       \1
6075
6076    NORMALLY MODE IS INTEGER
6077    DEFINE UNITS. AND NEW.UNITS AS 1-DIMENSIONAL ARRAYS
6078    LET NEW.COUNT=DIM.F(UNITS.(*))
6079    LOOP FOR I = 1 TO DIM.F(UNITS.(*))
6080    DO  ''                                                                         \1
6081       LET SIDE = UN.COLOR(UNITS.(I)) ''%26OCT79_%RGR
6082       LET FULL.STRENGTH=TU.CRIT.NO(UN.TYPE.UNIT(UNITS.(I)))
6083       IF FULL.STRENGTH GT 0  '' %10JAN80_%RGR
6084          '' THIS UNIT DOES HAVE CRITICAL EQUIPMENT
6085          LET CUR.STRENGTH=MU.CRIT.NO(UN.PTR(UNITS.(I)))
6086          LET FRACTION=(CUR.STRENGTH/FULL.STRENGTH)*100
6087       ELSE
6088    <------CYCLE ''                                                               \CLEANUP
6089       ENDIF
6090       IF FRACTION < BREAK.POINT(SIDE) '' %26OCT79_%RGR
6091          CALL DEAD.UNIT(UNITS.(I))------------------------------------------------------------------------------------------>(142)
6092          LET UN.RADIUS(UNITS.(I)) = 333 '' %11JAN80_%RGR
6093          SUBTRACT 1 FROM NEW.COUNT
6094          LET UNITS.(I)=-1
6095       ENDIF
6096    ENDLOOP
6097    LOOP FOR I=1 TO DIM.F(UNITS.(*)), WHEN UNITS.(I) > 0
6098    DO  ''                                                                         \1
6099       LET TYPE.U.PTR = UN.TYPE.UNIT(UNITS.(I))  '' %10JAN80_%RGR
6100       LET F.STRENGTH=TU.CRIT.NO(TYPE.U.PTR)
6101       IF F.STRENGTH = 0  '' THIS UNIT MAY NOT HAVE ANY CRITICAL EQUIP
6102          AND MU.TF.LIST(UN.PTR(UNITS.(I))) IS EMPTY
6103          AND UN.RADIUS(UNITS.(I)) NE 333 '' %11JAN80_%RGR
6104          CALL DEAD.UNIT(UNITS.(I))------------------------------------------------------------------------------------------>(142)
6105          SKIP 1 LINE
6106          PRINT 1 LINE WITH UNIT.NOS(UNITS.(I)) THUS '' %14JAN80_%RGR
6107          = = CHECK.DEAD CALLED DEAD.UNIT WITH UNIT ****** AND 2D COND IS TRUE = = =
6108          SKIP 1 LINE
6109          SUBTRACT 1 FROM NEW.COUNT
6110          LET UNITS.(I)=-1
6111       ENDIF
6112    ENDLOOP
6113    LET STATUS.=YES
6114    IF NEW.COUNT=0
6115       LET STATUS.=NO
6116       RESERVE NEW.UNITS AS 1
6117       RELEASE UNITS.(*)
6118 <------EXITROUTINE
6119    OTHERWISE
6120    RESERVE NEW.UNITS AS NEW.COUNT
6121    LET COUNT=1
6122    LOOP FOR I=1 TO DIM.F(UNITS.(*))
6123    DO  ''                                                                         \1
6124       IF UNITS(I) > 0
```

```
6125          LET NEW.UNITS(COUNT)=UNITS.(I)
6126          ADD 1 TO COUNT
6127       ENDIF
6128    ENDLOOP
6129    RELEASE UNITS.(*)
6130 ENDROUTINE
```

```
6131 ROUTINE CHECK.ENGAGEMENT                                                                                    ''                                                                                
```

```
6153 ROUTINE CHECK.FORCE                                                                                    DF04
6154    GIVEN
6155    CHECK.UNIT
6156
6157    ADD 1 TO ANAL.CTR(43,1)  ''                                                                         \DYN_ANAL
6158    NORMALLY MODE IS INTEGER
6159    FOR EVERY FORCE OF THE BTL.FORCE.SET(UN.BATTLE.INDEX(CHECK.UNIT))
6160    WITH FR.SIDE(FORCE) = UN.COLOR(CHECK.UNIT)
6161    FIND THE FIRST CASE
6162    IF NONE,
6163       CALL ERROR.STOP-------------------------------------------------------------------------------->(604)
6164    ENDIF
6165    IF DECISION.POINT(FORCE) = 101
6166 <------EXITROUTINE
6167    OTHERWISE
6168    LOOP
6169    FOR EVERY UNIT OF FR.UNIT.SET(FORCE)
6170    DO THE FOLLOWING
6171       LET U.FRACT = (MU.CRIT.NO(UN.PTR(UNIT))/
6172       TU.CRIT.NO(UN.TYPE.UNIT(UNIT))) * 100
6173       IF U.FRACT >= BREAK.POINT(FR.SIDE(FORCE))
6174          ADD MU.CRIT.NO(UN.PTR(UNIT)) TO CRIT.NO
6175       ENDIF
6176    ENDLOOP
6177    LET FRACTION = (CRIT.NO/FR.CRIT.NO(FORCE))*100
6178    IF FRACTION LE DECISION.POINT(FORCE)
6179       PRINT 2 LINES WITH
6180       TIME.V,
6181       BTL.SEQ.NO(UN.BATTLE.INDEX(CHECK.UNIT)),
6182       SHADE(FR.SIDE(FORCE)), FRACTION
6183       THUS
6184       ==========FORCE WITHDRAWING==========
6185       TIME.V = **.******** BATTLE = ******  SIDE = ******  STRENGTH = ***%
6186       LOOP
6187       FOR EVERY UNIT OF FR.UNIT.SET(FORCE)
6188       WITH UN.STATUS(UNIT) EQ ADVANCING
6189       OR (UN.STATUS(UNIT) EQ STATIONARY
6190       AND  UN.POSITION.INDEX(UNIT) NE 1)
6191       DO THE FOLLOWING
6192          CALL PREP.WITHDRAW-------------------------------------------------------------------------->(114)
6193          GIVEN
6194          UNIT
6195          ACTIVATE_A WITH.DRAW-------------------------------------------------------------------------->(478)
6196          GIVEN
6197          UNIT NOW ''                                                                                   \1
6198       ENDLOOP
6199       ''INSURE ONLY ONE WINNER
6200       FOR EVERY OTHER.FORCE OF THE BTL.FORCE.SET(UN.BATTLE.INDEX(CHECK.UNIT))
6201       WITH FR.SIDE(OTHER.FORCE) NE UN.COLOR(CHECK.UNIT)
6202       FIND THE FIRST CASE
6203       IF NONE,
6204          CALL ERROR.STOP-------------------------------------------------------------------------->(604)
6205       ENDIF
6206       LET DECISION.POINT(FORCE) = 101
6207       LET DECISION.POINT(OTHER.FORCE) = -1
6208    ALWAYS
6209 <--EXITROUTINE
6210 ENDROUTINE
```

```
.11 ROUTINE CHECK.FOR.MINES                                                    ''                    DF05
5212    GIVEN
6213    .UNIT,
6214    .XEND,
6215    .YEND
6216
6217    YIELDING
6218
6219    .MINE.FLAG
6220
6221    ADD 1 TO ANAL.CTR(44,1)  ''                                                                  \DYN_ANAL
6222    ''THIS ROUTINE IS CALLED FROM START MOVE AND START ARTY
6223    ''MOVEMENT TO FIND BARRIER MINES IN THE PATH OF THE TASK FORCE
6224
6225    NORMALLY MODE IS INTEGER
6226    DEFINE .SLOPE, .SLOPE2 AS REAL VARIABLES
6227
6228    LET .MINE.FLAG = 0
6229    LET .XLEFT = MIN.F(UN.X.COORD(.UNIT), .XEND)
6230    LET .XRIGHT = MAX.F(UN.X.COORD(.UNIT), .XEND)
6231    LET .YLOW = MIN.F(UN.Y.COORD(.UNIT), .YEND)
6232    LET .YHIGH = MAX.F(UN.Y.COORD(.UNIT), .YEND)
6233    LET .XMID = (.XLEFT + .XRIGHT) / 2.
6234    LET .YMID = (.YLOW + .YHIGH) / 2.
6235    ''SLOPE OF UNIT'S PATH
6236    IF .XEND NE UN.X.COORD(.UNIT)
6237       LET .SLOPE = (.YEND-UN.Y.COORD(.UNIT)) /
6238       (.XEND - UN.X.COORD(.UNIT))
6239    ELSE
6240       LET .SLOPE = RINF.C
6241    ALWAYS
6242
6243    LOOP
6244    FOR EVERY MINEFIELD CALLED .MF
6245    WITH MF.COLOR(.MF) NE UN.COLOR(.UNIT)
6246    DO
6247       ''IGNORE FIELDS PREVIOUSLY HIT
6248       FOR EVERY .AU IN AU.LIST(.MF)
6249       WITH AU.UNIT.ID(.AU) = .UNIT
6250       FIND THE FIRST CASE
6251       IF FOUND
6252    <------CYCLE
6253       OTHERWISE
6254       IF AU.LIST(.MF) IS NOT EMPTY
6255          LET .MINE.FLAG = 1
6256       ALWAYS
6257
6258       ''MAKE A CRUDE ELIMINATION
6259       IF MF.X.LOW(.MF) GT .XRIGHT OR
6260          MF.X.HIGH(.MF) LT .XLEFT OR
6261          MF.Y.LOW(.MF) GT .YHIGH OR
6262          MF.Y.HIGH(.MF) LT .YLOW
6263    <------CYCLE
6264       OTHERWISE
6265
6266       ''TRY A FINER ELIMINATION
6267       IF (MF.X.LOW(.MF) GT .XMID AND MF.Y.HIGH(.MF) LT .YMID) OR
6268          (MF.X.HIGH(.MF) LT .XMID AND MF.Y.LOW(.MF) GT .YMID)
```

```
6269     <-----CYCLE
6270        OTHERWISE
6271
6272        ''CHECK THE POINT OF INTERSECTION WITH EACH SEGMENT
6273        ''OF THE MINEFIELD
6274        LET .HIT = NO
6275        LOOP
6276        FOR EACH .MFP IN MFP.LIST(.MF)
6277        UNTIL S.MFP.LIST(.MFP) = 0 OR
6278        .HIT = YES
6279        DO
6280           IF MFP.X.COORD(.MFP) NE MFP.X.COORD(S.MFP.LIST(.MFP))
6281              LET .SLOPE2 = (MFP.Y.COORD(S.MFP.LIST(.MFP)) -
6282              MFP.Y.COORD(.MFP)) /
6283              (MFP.X.COORD(S.MFP.LIST(.MFP))-
6284              MFP.X.COORD(.MFP))
6285           ELSE
6286              LET .SLOPE2 = RINF.C
6287           ALWAYS
6288           IF .SLOPE = .SLOPE2
6289              ''UNIT IS MOVING PARALLEL TO MINEFIELD
6290        <-----CYCLE
6291           OTHERWISE
6292           IF .SLOPE = RINF.C OR .SLOPE2 = RINF.C
6293              IF .SLOPE = RINF.C
6294                 LET .XINTER = .XEND
6295                 LET .YINTER = .SLOPE2 * (.XINTER -
6296                 MFP.X.COORD(.MFP)) + MFP.Y.COORD(.MFP)
6297              ELSE
6298                 LET .XINTER = MFP.X.COORD(.MFP)
6299                 LET .YINTER = .SLOPE * (.XINTER - .XEND)
6300                 + .YEND
6301              ALWAYS
6302           ELSE
6303              LET .XINTER = (.SLOPE * .XEND
6304              - .SLOPE2 * MFP.X.COORD(.MFP)
6305              + MFP.Y.COORD(.MFP)-.YEND) / (.SLOPE - .SLOPE2)
6306              LET .YINTER = .SLOPE * (.XINTER - .XEND) + .YEND
6307           ALWAYS
6308           LET .LEFTX = MIN.F(MFP.X.COORD(.MFP),
6309           MFP.X.COORD(S.MFP.LIST(.MFP)))
6310           LET .RIGHTX = MAX.F(MFP.X.COORD(.MFP),
6311           MFP.X.COORD(S.MFP.LIST(.MFP)))
6312           LET .LOWY = MIN.F(MFP.Y.COORD(.MFP),
6313           MFP.Y.COORD(S.MFP.LIST(.MFP)))
6314           LET .HIGHY = MAX.F(MFP.Y.COORD(.MFP),
6315           MFP.Y.COORD(S.MFP.LIST(.MFP)))
6316           IF .LEFTX LE .XINTER AND .XINTER LE .RIGHTX AND
6317              .LOWY LE .YINTER AND .YINTER LE .HIGHY AND
6318              .XLEFT LE .XINTER AND .XINTER LE .XRIGHT AND
6319              .YLOW LE .YINTER AND .YINTER LE .YHIGH
6320              CREATE A MINE.OBSTACLE CALLED .MO
6321              LET MO.MINEFIELD(.MO) = .MF
6322              LET MO.X.INTER(.MO) = .XINTER
6323              LET MO.Y.INTER(.MO) = .YINTER
6324              FILE THE .MO IN THE MO.LIST(.UNIT)
6325              LET .HIT = YES
6326           ALWAYS
```

```
6327        ENDLOOP
6328     ENDLOOP
6329
6330 <--EXITROUTINE
6331 ENDROUTINE
```

```
6332 ROUTINE CHECK.LIST(UNT.,NO.TF,DUM.UNITS.) YIELDING RESULT ''                %30AUG_%RGR \DF06
6333
6334    ADD 1 TO ANAL.CTR(45,1)  ''                                              \DYN_ANAL
6335    ''THIS ROUTINE CHECKS TO SEE IF A PARTICULAR UNIT IS ALREADY
6336    ''  ON THE LIST OF UNITS TO BE IN A BATTLE.  IF IT IS NOT
6337    ''  IT IS ADDED TO THE LIST ALONG WITH ALL OF THE UNITS
6338    ''  IN ITS TASK FORCE.
6339
6340    NORMALLY MODE IS INTEGER
6341    DEFINE DUM.UNITS. AS A 1-DIMENSIONAL ARRAY
6342    DEFINE MAN.UNIT AS A VARIABLE
6343    LOOP FOR COUNT=1 TO NO.TF
6344    DO ''                                                                    \1
6345       IF DUM.UNITS.(COUNT)=UNT.
6346          LET RESULT=0
6347 <--------EXITROUTINE
6348       OTHERWISE
6349    ENDLOOP
6350    LET RESULT=1
6351    LET COUNT=NO.TF+1
6352    IF UNT. GE 2000,
6353       LET POINT.ER = UNT.
6354       LET UNT. = MU.UNIT.ID(POINT.ER)
6355    ELSE
6356    ALWAYS      ''LINES ADDED %11SEP79_%RGR
6357    LET DUM.UNITS.(COUNT)=UNT.
6358    LET A.TF.MEM = UN.PTR(UNT.)    '' %30AUG_%RGR
6359    IF DEBUG = TRUE,
6360       PRINT 1 LINE WITH  UNT., A.TF.MEM    THUS
6361       = = = CHECK.LIST UNIT ****** (POINTER ******) IS FORMING A TASK FORCE = = =
6362    ENDIF
6363    LOOP FOR EACH MAN.UNIT IN MU.TF.LIST(A.TF.MEM)
6364    DO ''                                                                    \1
6365       ADD 1 TO COUNT
6366       ADD 1 TO RESULT
6367       IF DEBUG = TRUE,
6368          LIST COUNT, RESULT, A.TF.MEM
6369       ENDIF
6370       LET DUM.UNITS.(COUNT)=MU.UNIT.ID(MAN.UNIT)
6371    ENDLOOP
6372 ENDROUTINE
```

```
6373 ROUTINE CHECK.PROX GIVEN UNT.,UNT.SECTOR.,CHK.RANGE.   ''                                    DF07
6374     YIELDING PROX.TEST.,EN.UNITS.,BATTLE.NUM
6375
6376     ADD 1 TO ANAL.CTR(46,1)  ''                                                             \DYN_ANAL
6377     ''THIS ROUTINE IS CALLED AT EVERY UPDATE OF A UNIT'S LOCATION
6378     ''  DURING AN AGGRESSIVE MOVE.  IT CHECKS THE UNIT'S PROXIMITY
6379     ''  TO EVERY ENEMY MANEUVER UNIT IN ADJACENT SECTORS.  THE
6380     ''  CRITICAL RANGE AGAINST WHICH THE PROXIMITY IS MEASURED IS
6381     ''  PASSED AS AN ARGUMENT (CHK.RANGE) TO THE ROUTINE.  A VAR-
6382     ''  IABLE (PROX.TEST) IS SET TO YES OR NO AND RETURNED TO THE
6383     ''  CALLING ROUTINE.  ALSO RETURNED IS THE LIST OF ENEMY UNITS
6384     ''  TO WHICH PROXIMITY EXISTS.
6385
6386     NORMALLY MODE IS INTEGER
6387     DEFINE EN.UNITS. AS AN INTEGER 1-DIMENSIONAL ARRAY
6388     DEFINE DUM.UNITS. AS A 1-DIMENSIONAL ARRAY
6389     DEFINE SEARCH.SECTOR AS A 1-DIMENSIONAL INTEGER ARRAY
6390     RESERVE DUM.UNITS. AS 100    '' CHANGED FROM 15 TO 30 %8APR80_%RGR
6391     LET ENEMY. = RED
6392     IF UN.COLOR(UNT.)=RED
6393        LET ENEMY. = BLUE
6394     ENDIF
6395     LET COUNT=0
6396     RESERVE SEARCH.SECTOR(*) AS 3
6397     LET NUM.SECT = 0
6398     IF SS.SET(ENEMY.,UNT.SECTOR.) IS NOT EMPTY
6399        ADD 1 TO NUM.SECT
6400        LET SEARCH.SECTOR(NUM.SECT) = UNT.SECTOR.
6401     ALWAYS
6402     IF UNT.SECTOR.+1 <= N.SECTOR AND
6403        SS.SET(ENEMY.,UNT.SECTOR.+1) IS NOT EMPTY
6404        ADD 1 TO NUM.SECT
6405        LET SEARCH.SECTOR(NUM.SECT) = UNT.SECTOR.+1
6406     ALWAYS
6407     IF UNT.SECTOR.-1 > 0 AND
6408        SS.SET(ENEMY.,UNT.SECTOR-1) IS NOT EMPTY
6409        ADD 1 TO NUM.SECT
6410        LET SEARCH.SECTOR(NUM.SECT) = UNT.SECTOR.-1
6411     ALWAYS
6412     IF NUM.SECT = 0
6413        LET PROX.TEST. = NO
6414        RESERVE EN.UNITS.(*) AS 1
6415        RELEASE DUM.UNITS.(*)
6416        RELEASE SEARCH.SECTOR(*)
6417 <------EXITROUTINE
6418     OTHERWISE
6419     LET CLOSE = INF.C
6420     LOOP FOR I = 1 TO NUM.SECT
6421     DO ''                                                                                   \1
6422        LOOP FOR EVERY .UNIT IN SS.SET(ENEMY.,SEARCH.SECTOR(I))
6423        DO ''                                                                                \1
6424           IF MU.TF.LIST(UN.PTR(UNT.)) IS EMPTY
6425              ''UTILIZE ACT.RANGE IN NEXT ASSIGNMENT ------------------------------------- \1>(628)
6426              LET RANGE. = ACT.RANGE(UNT.,.UNIT)
6427              IF RANGE. <= CHK.RANGE.
6428                 LET FRACTION = (MU.CRIT.NO(UN.PTR(.UNIT))/
6429                 TU.CRIT.NO(UN.TYPE.UNIT(.UNIT))) * 100.
6430                 IF FRACTION > BREAK.POINT(ENEMY.)
```

```
6431                  ADD 1 TO COUNT
6432                  LET DUM.UNITS.(COUNT) = .UNIT
6433                  IF RANGE. LT CLOSE
6434                     LET UN = .UNIT
6435                     LET CLOSE = RANGE.
6436                  ALWAYS
6437               ALWAYS
6438            ALWAYS
6439         ELSE
6440            LOOP FOR EVERY M.U IN MU.TF.LIST(UN.PTR(UNT.))
6441            DO ''                                                                    \1
6442               ''UTILIZE ACT.RANGE IN NEXT ASSIGNMENT ------------------------------ \1>(628)
6443               LET RANGE. = ACT.RANGE(MU.UNIT.ID(M.U),.UNIT)
6444               IF RANGE. <= CHK.RANGE.
6445                  LET FRACTION = (MU.CRIT.NO(UN.PTR(.UNIT))/
6446                  TU.CRIT.NO(UN.TYPE.UNIT(.UNIT))) * 100.
6447                  IF FRACTION > BREAK.POINT(ENEMY.)
6448                     ADD 1 TO COUNT
6449                     LET DUM.UNITS.(COUNT) = .UNIT
6450                     IF RANGE. LT CLOSE
6451                        LET UN = .UNIT
6452                        LET CLOSE = RANGE.
6453                     ALWAYS
6454               <--------EXITLOOP
6455                  OTHERWISE
6456               ALWAYS
6457            ENDLOOP
6458         ALWAYS
6459      ENDLOOP
6460   ENDLOOP
6461   RELEASE SEARCH.SECTOR(*)
6462   LET PROX.TEST.=NO
6463   IF COUNT > 100
6464      TRACE   STOP
6465   OTHERWISE
6466   IF COUNT > 0
6467      LET PROX.TEST=YES
6468      IF UN > 0
6469         IF UN.BATTLE.INDEX(UN) > 0
6470            LET BATTLE.NUM = UN.BATTLE.INDEX(UN)
6471         ELSE
6472            LET BATTLE.NUM = 0
6473         ENDIF
6474      ENDIF
6475      IF BATTLE.NUM = 0
6476         LET RE.COUNT = 0
6477         LOOP FOR I = 1 TO COUNT
6478         DO ''                                                                       \1
6479            IF UN.BATTLE.INDEX(DUM.UNITS.(I)) > 0
6480               LET DUM.UNITS.(I) = 0
6481            ELSE
6482               ADD 1 TO RE.COUNT
6483            ENDIF
6484         ENDLOOP
6485      ELSE
6486         LET RE.COUNT = COUNT
6487      ENDIF
6488      RESERVE EN.UNITS. AS RE.COUNT
```

```
6489        LET J = 0
6490        LOOP FOR I=1 TO COUNT
6491        DO ''                                                    \1
6492           IF DUM.UNITS.(I) > 0
6493              ADD 1 TO J
6494              LET EN.UNITS.(J)=DUM.UNITS.(I)
6495           ENDIF
6496        ENDLOOP
6497        RELEASE DUM.UNITS.(*)
6498 <------EXITROUTINE
6499     OTHERWISE
6500     RESERVE EN.UNITS. AS 1
6501     RELEASE DUM.UNITS.(*)
6502 ENDROUTINE
```

```
6503 ROUTINE CHECK.STREN(UNT.) YIELDING PCT.STREN ''                                    DF08
6504
6505    ADD 1 TO ANAL.CTR(47,1)  ''                                                     \DYN_ANAL
6506    ''THIS ROUTINE COMPUTES THE PERCENT OF CRITICAL EQUIPMENT ON-HAND
6507    ''  IN A TASK FORCE.  THIS FIGURE IS USED IN DETERMINING APPROP-
6508    ''  RIATE ORDERS FOR THE FORCE.
6509
6510    NORMALLY MODE IS INTEGER
6511    DEFINE MAN.UNIT AS A VARIABLE
6512    LET NO.ON.HAND=MU.CRIT.NO(UN.PTR(UNT.))
6513    LET NO.AUTH=TU.CRIT.NO(UN.TYPE.UNIT(UNT.))
6514    LOOP FOR EVERY MAN.UNIT IN MU.TF.LIST(UN.PTR(UNT.))
6515    DO ''                                                                           \1
6516       ADD MU.CRIT.NO(MAN.UNIT) TO NO.ON.HAND
6517       ADD TU.CRIT.NO(UN.TYPE.UNIT(MU.UNIT.ID(MAN.UNIT))) TO NO.AUTH
6518    ENDLOOP
6519    LET PCT.STREN=100*NO.ON.HAND/NO.AUTH
6520 ENDROUTINE
```

```
6521 ROUTINE DEAD.UNIT                                                                                    ''                                                                                
```

```
6579          REMOVE NEW.TF FROM MU.TF.LIST(MAN.UNT)
6580          LET TF.LDR = MU.UNIT.ID(NEW.TF)
6581          LET NEW.MU = UN.PTR(TF.LDR)
6582          IF NEW.MU NE NEW.TF
6583             CALL ERROR.STOP----------------------------------------------------------------------->(604)
6584          ALWAYS
6585
6586          PRINT 1 LINE WITH UNIT.NOS(TF.LDR), NEW.MU THUS
6587          UNIT ****** (POINTER *****) TAKES OVER TASK FORCE
6588
6589          LET MU.TF.MEM(NEW.MU)=0
6590          LOOP UNTIL MU.TF.LIST(MAN.UNT) IS EMPTY
6591          DO
6592             REMOVE THE FIRST MAN.UNIT FROM MU.TF.LIST(MAN.UNT)
6593             LET MU.TF.MEM(MAN.UNIT)=TF.LDR
6594             FILE MAN.UNIT IN MU.TF.LIST(NEW.MU)
6595          ENDLOOP
6596          IF MU.ORDER.SET(NEW.MU) IS EMPTY
6597             LET MU.CUR.ORDER(NEW.MU)=MU.CUR.ORDER(MAN.UNT)
6598             LOOP UNTIL MU.ORDER.SET(MAN.UNT) IS EMPTY
6599             DO
6600                REMOVE FIRST ORDER FROM MU.ORDER.SET(MAN.UNT)
6601                FILE THE ORDER IN MU.ORDER.SET(NEW.MU)
6602             ENDLOOP
6603          ALWAYS
6604
6605          ''REPLACE NEW TF LDR IN AU.LIST OF MINEFIELDS
6606          LOOP FOR EVERY MINEFIELD CALLED .MF
6607          DO
6608             LOOP FOR EVERY .AU IN AU.LIST(.MF)
6609             WITH AU.UNIT.ID(.AU) = UNT.
6610             DO
6611                LET AU.UNIT.ID(.AU) = TF.LDR
6612             ENDLOOP
6613          ENDLOOP
6614       ELSE
6615          CALL DESTROY.ORD---------------------------------------------------------------------->(326)
6616          GIVEN
6617          UNT.
6618       ALWAYS
6619    ALWAYS
6620
6621    IF UN.SUB.LIST(UNT.) IS NOT EMPTY
6622       IF UN.PARENT(TF.LDR) = UNT.
6623          LET NEW.PARENT = TF.LDR
6624       ELSE
6625          LET NEW.PARENT=F.UN.SUB.LIST(UNT.)
6626       ALWAYS
6627
6628       PRINT 1 LINE WITH UNIT.NOS(NEW.PARENT), UNIT.NOS(UNT.) THUS
6629       UNIT ****** ASSUMES CONTROL OF UNIT ******'S SUBORDINATES
6630       SKIP 1 LINE
6631
6632       LET UN.PARENT(NEW.PARENT)=UN.PARENT(UNT.)
6633       REMOVE NEW.PARENT FROM UN.SUB.LIST(UNT.)
6634       IF UN.PARENT(NEW.PARENT) NE 0
6635          FILE NEW.PARENT IN UN.SUB.LIST(UN.PARENT(NEW.PARENT))
6636       ALWAYS
```

```
6637        LOOP UNTIL UN.SUB.LIST(UNT.) IS EMPTY
6638        DO
6639           REMOVE THE FIRST UNIT FROM UN.SUB.LIST(UNT.)
6640           LET UN.PARENT(UNIT)=NEW.PARENT
6641           FILE UNIT IN UN.SUB.LIST(NEW.PARENT)
6642        ENDLOOP
6643        LET MU = UN.PTR(NEW.PARENT)
6644        IF MU.ORDER.SET(MU) IS EMPTY
6645           LOOP UNTIL MU.ORDER.SET(MAN.UNT) IS EMPTY
6646           DO
6647              REMOVE FIRST ORDER FROM MU.ORDER.SET(MAN.UNT)
6648              FILE THE ORDER IN MU.ORDER.SET(MU)
6649           ENDLOOP
6650           LET MU.CUR.ORDER(MU) = MU.CUR.ORDER(MAN.UNT)
6651        ALWAYS
6652     ALWAYS
6653
6654     LET MU.TF.MEM(MAN.UNT) = 0
6655     IF UN.PARENT(UNT.) NE 0
6656        REMOVE UNT. FROM UN.SUB.LIST(UN.PARENT(UNT.))
6657        LET UN.PARENT(UNT.) = 0  ''UNIT IS DEAD, DON'T DO IT TWICE
6658     ALWAYS
6659
6660     CREATE AN ORDER
6661     LET ORD.TYPE(ORDER)="DEF"
6662     LET ORD.SEQ.NO(ORDER)=1
6663
6664     CREATE A DEF.ORDER CALLED ORD.ID(ORDER)
6665     LET ORD.MISSION(ORD.ID(ORDER))=DELAY
6666     LET REINF.THRESH(ORD.ID(ORDER)) = 20
6667     LET EN.DIS.OP(ORD.ID(ORDER))=1
6668     LET OWN.DIS.OP(ORD.ID(ORDER))=2
6669     FILE ORDER IN  MU.ORDER.SET(MAN.UNT)
6670     LET UN.MISSION(UNT.) = ORD.MISSION(ORD.ID(ORDER))
6671     LET UN.STATUS(UNT.) = STATIONARY
6672
6673     CREATE AN ORDER
6674     LET ORD.TYPE(ORDER)="MOVDIS"
6675     LET ORD.SEQ.NO(ORDER)=2
6676
6677     CREATE A MOVDIS.ORDER CALLED ORD.ID(ORDER)
6678     LET DIR.OF.MOVE(ORD.ID(ORDER))="WITHDR"
6679     LET DIST.MOVED(ORD.ID(ORDER))=20000
6680     '' UNIT MOVES 20 KM TO THE REAR FROM PRESENT LOCATION
6681     LET MOVE.TYPE(ORD.ID(ORDER))="ADMIN"
6682     LET ORD.NEXT(ORD.ID(ORDER))=1
6683     FILE ORDER IN MU.ORDER.SET(MAN.UNT)
6684
6685     LET MU.CUR.ORDER(MAN.UNT)=2
6686     SCHEDULE_AN ACT.MOVDIS------------------------------------------------------------------------------------------>(416)
6687     GIVEN
6688     UNT.,
6689     L.MU.ORDER.SET(MAN.UNT)  NOW   ''                                                                              \1
6690 <--EXITROUTINE
6691 ENDROUTINE
```

```
6692 ROUTINE FIN.BATTLE                                                                                    DF10
6693    GIVEN .BATTLE, WITH.DRAW.FORCE
6694
6695    ADD 1 TO ANAL.CTR(49,1)  ''                                                                        \DYN_ANAL
6696    NORMALLY MODE IS INTEGER
6697    DEFINE .BLUE.UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
6698    DEFINE .RED.UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
6699    LET .BLUE.UNITS(*) = BTL.BL.UNITS(.BATTLE)
6700    LET .RED.UNITS(*) = BTL.RD.UNITS(.BATTLE)
6701    LOOP
6702    FOR EVERY .FORCE OF BTL.FORCE.SET(.BATTLE)
6703    DO THE FOLLOWING
6704       IF .FORCE NE WITH.DRAW.FORCE
6705          LET OTHER.FORCE = .FORCE
6706       ALWAYS
6707       LET FN = 0
6708       LOOP
6709       FOR EVERY .UNIT OF FR.UNIT.SET(.FORCE)
6710       DO THE FOLLOWING
6711          LET N = MU.CRIT.NO(UN.PTR(.UNIT))
6712          LET D = TU.CRIT.NO(UN.TYPE.UNIT(.UNIT))
6713          IF D = 0
6714       <------CYCLE
6715          OTHERWISE
6716          LET .SIDE = UN.COLOR(.UNIT)  '' %26OCT79_%RGR
6717          IF (N/D) * 100 LE BREAK.POINT(.SIDE)
6718       <------CYCLE
6719          OTHERWISE
6720          ADD N TO FN
6721       ENDLOOP
6722       IF (FN / FR.CRIT.NO(.FORCE)) * 100 LE
6723          DECISION(FR.MISSION(.FORCE),FR.SIDE(.FORCE))
6724          LET DECISION.POINT(.FORCE) = 101
6725       ELSE
6726          LET DECISION.POINT(.FORCE) = -1
6727       ALWAYS
6728    ENDLOOP
6729    IF OTHER.FORCE = 0
6730       TRACE      CALL ERROR.STOP
6731    ALWAYS
6732    LET WINNER = FR.SIDE(OTHER.FORCE)
6733    IF DECISION.POINT(OTHER.FORCE) = 101
6734       LET WINNER = 0
6735    ALWAYS
6736    SCHEDULE_A BTL.ENDED GIVING WINNER, .BLUE.UNITS(*), .RED.UNITS(*) NOW----------------------------------------------------------->(356)
6737 <--EXITROUTINE
6738 ENDROUTINE
```

```
6739 ROUTINE INTER.BATTLE GIVEN UNT,BATTLE,ORDR.  ''                                                    DF11
6740
6741    ADD 1 TO ANAL.CTR(50,1)  ''                                                                     \DYN_ANAL
6742    ''THIS ROUTINE IS CALLED WHEN A UNIT REACHES PROXIMITY TO A BATTLE
6743    ''  AREA DURING THE COURSE OF THE BATTLE.  THE BATTLE IS INTERRUPTED,
6744    ''  THE ARRIVING UNITS ARE ADDED TO THE LIST OF UNITS IN THE BATTLE,
6745    ''  AND THE BATTLE RE-STARTED.
6746
6747    NORMALLY MODE IS INTEGER
6748    DEFINE BL.UNITS,RD.UNITS,NEW.RED,NEW.BLUE,NEW.UNIT.LIST
6749    AS 1-DIMENSIONAL ARRAYS
6750    DEFINE ARG.ARRAY AS A 1-DIMENSIONAL INTEGER ARRAY
6751    DEFINE ORDER AS AN INTEGER VARIABLE
6752    LET RD.UNITS(*)=BTL.RD.UNITS(BATTLE)
6753    LET BL.UNITS(*)=BTL.BL.UNITS(BATTLE)
6754    LET DUMMY = 0
6755    LET FLAG = 0
6756    LET TERR=BTL.TERRAIN.TYPE(BATTLE)
6757    LET MU = UN.PTR(UNT)
6758    CALL CHECK.STREN(UNT) YIELDING PCT.ON.HAND------------------------------------------------------>(141)
6759    IF PCT.ON.HAND >= THRESH.REIN(ORD.ID(ORDR.))
6760       LET NEXT = NX.ORD.ABOVE(ORD.ID(ORDR.))
6761    ELSE
6762       LET NEXT = NX.ORD.BELOW(ORD.ID(ORDR.))
6763    ENDIF
6764    FOR EVERY ORDER IN MU.ORDER.SET(MU)
6765    WITH ORD.SEQ.NO(ORDER) = NEXT
6766    FIND THE FIRST CASE
6767    IF FOUND
6768       IF ORD.TYPE(ORDER) NE "ATK"
6769          SCHEDULE_A GET.NX.ORD (UNT, NEXT, DUMMY, FLAG) NOW------------------------------------->(373)
6770 <--------EXITROUTINE
6771       OTHERWISE  ''                                                    ELSE CHANGED TO OTHERWISE \1
6772       PRINT 2 LINES WITH UNIT.NOS(UNT),ORD.SEQ.NO(ORDR.),ORD.TYPE(ORDR.),
6773       TIME.V,NEXT,ORD.TYPE(ORDER) THUS
6774       UNIT ****** LEAVING ORDER #** (******) AT TIME = **.**
6775       NEXT ORDER IS #** (******)
6776       LET MU.CUR.ORDER(MU) = NEXT
6777    ELSE
6778       TRACE
6779       CALL ERROR.STOP  ''------------------------------------------------------------------ \1>(604)
6780    ENDIF
6781    IF ORD.TYPE(ORDER) = "REINF"
6782       SCHEDULE_A GET.NX.ORD(UNT,NEXT,DUMMY,FLAG) NOW------------------------------------------->(373)
6783 <-------EXITROUTINE
6784    OTHERWISE
6785    CALL INTER.HELO--------------------------------------------------------------------------------->(311)
6786    GIVEN
6787    BATTLE
6788    YIELDING
6789    ARG.ARRAY(*)
6790    CALL SWITCH.FO GIVEN BATTLE,2------------------------------------------------------------------>(238)
6791    PRINT 2 LINES WITH
6792    TIME.V,
6793    BTL.SEQ.NO(BATTLE), TERR THUS
6794    ============== BATTLE INTERRUPTED ==========
6795    TIME.V = **.*******      BATTLE = ******    TERRAIN = ****
6796    LOOP FOR EACH FORCE IN BTL.FORCE.SET(BATTLE)
```

```
6797    WHEN FR.SIDE(FORCE) = UN.COLOR(UNT)
6798    FIND THE FIRST CASE
6799    IF NONE
6800       TRACE
6801       CALL ERROR.STOP  ''------------------------------------------------------------------------------------------ \1>(604)
6802    ENDIF
6803    LET TF.LDR = 0
6804    LOOP FOR EACH .UNIT IN FR.UNIT.SET(FORCE)
6805    DO THIS ''                                                                                                      \1
6806       IF MU.TF.MEM(UN.PTR(.UNIT)) = 0
6807          LET TF.LDR = .UNIT
6808    <------EXITLOOP
6809       OTHERWISE
6810    ENDLOOP
6811    IF TF.LDR = 0
6812       TRACE
6813 <------STOP  ''                                                                                                    \1
6814    OTHERWISE
6815    LET MU.TF.LDR = UN.PTR(TF.LDR)
6816    CALL EMPTY GIVEN BATTLE,0,NO-------------------------------------------------------------------------------------->(328)
6817    IF UN.COLOR(UNT)=BLUE
6818       LET NUM=DIM.F(BL.UNITS(*))
6819       RESERVE NEW.UNIT.LIST AS NUM + N.MU.TF.LIST(UN.PTR(UNT)) + 1
6820       LOOP FOR I=1 TO NUM
6821       DO  ''                                                                                                       \1
6822          LET NEW.UNIT.LIST(I)=BL.UNITS(I)
6823          LET PARENT = UN.PARENT(BL.UNITS(I))
6824          IF PARENT > 0 AND UN.PTR(PARENT) > 0
6825             LET UN.Y.COORD(BL.UNITS(I)) = UN.Y.COORD(PARENT) -
6826             MU.OFFSET.Y(UN.PTR(BL.UNITS(I)))/10
6827          ENDIF
6828       ENDLOOP
6829       LET J=NUM+1
6830       LET NEW.UNIT.LIST(J)=UNT
6831       FILE UN.PTR(UNT) IN MU.TF.LIST(MU.TF.LDR)
6832       LET MU.TF.MEM(UN.PTR(UNT)) = TF.LDR
6833       LOOP FOR EVERY MU IN MU.TF.LIST(UN.PTR(UNT))
6834       DO  ''                                                                                                       \1
6835          REMOVE MU FROM MU.TF.LIST(UN.PTR(UNT))
6836          LET MU.TF.MEM(MU) = TF.LDR
6837          FILE MU IN MU.TF.LIST(MU.TF.LDR)
6838          ADD 1 TO J
6839          LET NEW.UNIT.LIST(J)=MU.UNIT.ID(MU)
6840       ENDLOOP
6841       CALL CHECK.DEAD(RD.UNITS(*)) YIELDING RD.STATUS, NEW.RED(*)------------------------------------------------->(130)
6842       IF RD.STATUS = NO
6843          TRACE
6844          CALL ERROR.STOP  ''------------------------------------------------------------------------------------- \1>(604)
6845       ENDIF
6846       CALL CHECK.DEAD(NEW.UNIT.LIST(*)) YIELDING BL.STATUS, NEW.BLUE(*)------------------------------------------->(130)
6847       IF BL.STATUS = NO
6848          TRACE
6849          CALL ERROR.STOP  ''------------------------------------------------------------------------------------- \1>(604)
6850       ENDIF
6851       SCHEDULE_A START.BATTLE GIVEN TERR,UN.MISSION(NEW.BLUE(1)),-------------------------------------------------->(395)
6852       UN.MISSION(NEW.RED(1)),NEW.BLUE(*),NEW.RED(*),ARG.ARRAY(*) NOW ''                                             \1
6853       RELEASE BL.UNITS(*)
6854    ELSE
```

```
6855        LET NUM=DIM.F(RD.UNITS(*))
6856        RESERVE NEW.UNIT.LIST AS NUM + N.MU.TF.LIST(UN.PTR(UNT)) + 1
6857        LOOP FOR I=1 TO NUM
6858        DO ''                                                                                    \1
6859           LET NEW.UNIT.LIST(I)=RD.UNITS(I)
6860           LET PARENT = UN.PARENT(RD.UNITS(I))
6861           IF PARENT > 0 AND UN.PTR(PARENT) > 0
6862              LET UN.Y.COORD(RD.UNITS(I)) = UN.Y.COORD(PARENT) -
6863              MU.OFFSET.Y(UN.PTR(RD.UNITS(I)))/10
6864           ENDIF
6865        ENDLOOP
6866        LET J=NUM+1
6867        LET NEW.UNIT.LIST(J)=UNT
6868        FILE UN.PTR(UNT) IN MU.TF.LIST(MU.TF.LDR)
6869        LET MU.TF.MEM(UN.PTR(UNT)) = TF.LDR
6870        LOOP FOR EVERY MU IN MU.TF.LIST(UN.PTR(UNT))
6871        DO  ''                                                                                   \1
6872           REMOVE MU FROM MU.TF.LIST(UN.PTR(UNT))
6873           LET MU.TF.MEM(MU) = TF.LDR
6874           FILE MU IN MU.TF.LIST(MU.TF.LDR)
6875           ADD 1 TO J
6876           LET NEW.UNIT.LIST(J)=MU.UNIT.ID(MU)
6877        ENDLOOP
6878        CALL CHECK.DEAD(NEW.UNIT.LIST(*)) YIELDING RD.STATUS, NEW.RED(*)------------------------------------------------------>(130)
6879        IF RD.STATUS = NO
6880           TRACE
6881           CALL ERROR.STOP  ''------------------------------------------------------------------------------------------ \1>(604)
6882        ENDIF
6883        CALL CHECK.DEAD(BL.UNITS(*)) YIELDING BL.STATUS, NEW.BLUE(*)------------------------------------------------------>(130)
6884        IF BL.STATUS = NO
6885           TRACE
6886           CALL ERROR.STOP    ''------------------------------------------------------------------------------------------ \1>(604)
6887        ENDIF
6888        SCHEDULE_A START.BATTLE GIVEN TERR,UN.MISSION(NEW.BLUE(1)),------------------------------------------------------>(395)
6889        UN.MISSION(NEW.RED(1)),NEW.BLUE(*),NEW.RED(*), ARG.ARRAY(*) NOW ''                       \1
6890        RELEASE RD.UNITS(*)
6891     ENDIF
6892  ENDROUTINE
```

```
6893 ROUTINE PK.COMPUTE                                                    DF12
6894    GIVEN
6895    SHOOTER,
6896    TARGET,
6897    RANGE,
6898    FIRER.UNIT,
6899    TARGET.UNIT
6900    YIELDING
6901    PK,
6902    SELECTED.WPN
6903
6904    ADD 1 TO ANAL.CTR(51,1)                                        \DYN_ANAL
6905    NORMALLY MODE IS INTEGER
6906    DEFINE DD,
6907    KK, PK, N.FAC, EE.REAL, EE.DIF, PK.ADJ.MOV.FAC, EE.FAC, EE.ABS
6908    AS REAL VARIABLES
6909    DEFINE BTL. AS AN INTEGER VARIABLE
6910
6911    LET BTL. = UN.BATTLE.INDEX(TARGET.UNIT)
6912    IF BTL. NE 0 AND
6913       FIRER.UNIT NE SD.AIRFIELD(UN.COLOR(FIRER.UNIT)) AND
6914       TARGET.UNIT NE SD.AIRFIELD(UN.COLOR(TARGET.UNIT))
6915       IF UN.STATUS(FIRER.UNIT) EQ ADVANCING
6916          OR UN.STATUS(FIRER.UNIT) EQ WITHDRAWING
6917          IF RANGE LT MOVE.FIRE.DIST(BTL.TERRAIN.TYPE(BTL.))
6918             LET FIRE.ON.MOVE = YES
6919          ALWAYS
6920       ALWAYS
6921
6922       IF UN.STATUS(TARGET.UNIT) = ADVANCING
6923          AND RANGE LT MOVE.FIRE.DIST(BTL.TERRAIN.TYPE(BTL.))
6924          LET TGT.ON.MOVE = YES
6925       ELSE
6926          IF UN.STATUS(TARGET.UNIT) = WITHDRAWING
6927             LET TGT.ON.MOVE = YES
6928          ALWAYS
6929       ALWAYS
6930       IF UN.STATUS(TARGET.UNIT) EQ ADVANCING
6931          OR UN.STATUS(TARGET.UNIT) EQ WITHDRAWING
6932          IF RANGE GT DEFILADE.DIST(BTL.TERRAIN.TYPE(BTL.))
6933             LET DEFILADE = YES
6934          ALWAYS
6935       ELSE
6936          LET DEFILADE = YES
6937       ALWAYS
6938    ALWAYS
6939
6940    LET BETA = EQUIP.PK.PTR(UE.ID(TARGET))
6941    LET PK = 0.
6942    LET SELECTED.WPN = 0
6943    LOOP FOR EVERY WEAPON OF UE.WEAPON.SET(SHOOTER)
6944    UNLESS TE.NAME(EQ.TE.PTR(UE.ID(SHOOTER))) = "MINES" OR
6945    TE.NAME(EQ.TE.PTR(UE.ID(TARGET))) = "MINES"
6946    DO THE FOLLOWING
6947       LET MAX = TW.MAX.RANGE(WPN.ID(WEAPON))
6948       LET MIN = TW.MIN.RANGE(WPN.ID(WEAPON))
6949       LET KK = 0.
6950       LET ALPHA = TW.PK.PTR(WPN.ID(WEAPON))
```

```
6951        IF MAX GT RANGE AND
6952           MIN LT RANGE AND
6953           ALPHA GT 0
6954           IF DEFILADE = YES
6955              LET TRIAL.PK.PTR = PK.DEF.POINTER(ALPHA,BETA)
6956           ELSE
6957              LET TRIAL.PK.PTR = PK.POINTER(ALPHA,BETA)
6958           ALWAYS
6959           IF TRIAL.PK.PTR NE 0
6960              FOR EVERY PK.VECTOR
6961              WITH PK.VECTOR EQ TRIAL.PK.PTR
6962              FIND THE FIRST CASE
6963              IF NONE,
6964                 CALL ERROR.STOP------------------------------------------------------------------------------------------>(604)
6965              ALWAYS
6966              LET DD = (MAX-MIN)/10
6967              LET EE.REAL = (RANGE-MIN)/DD
6968              LET EE.INT = EE.REAL
6969              LET EE.DIF = EE.REAL-EE.INT
6970              IF EE.DIF GE 0
6971                 IF (EE.REAL GE 10.)
6972                    AND (EE.INT EQ 10)
6973                    LET HIGH = EE.INT
6974                 ELSE
6975                    LET HIGH = EE.INT+1
6976                 ALWAYS
6977                 LET LOW = EE.INT
6978                 LET EE.FAC = EE.DIF
6979              ELSE
6980                 LET HIGH = EE.INT
6981                 LET LOW = EE.INT-1
6982                 LET EE.ABS = ABS.F(EE.DIF)
6983                 LET EE.FAC = 1.-EE.ABS
6984              ALWAYS
6985              FOR EVERY PK.BAND
6986              WITH PK.BAND.RNG(PK.BAND) EQ HIGH
6987              FIND THE FIRST CASE
6988              IF NONE,
6989                 CALL ERROR.STOP------------------------------------------------------------------------------------------>(604)
6990              ALWAYS
6991              LET HH = PK.PROB(PK.VECTOR,PK.BAND)
6992              FOR EVERY PK.BAND
6993              WITH PK.BAND.RNG(PK.BAND) EQ LOW
6994              FIND THE FIRST CASE
6995              IF NONE,
6996                 CALL ERROR.STOP------------------------------------------------------------------------------------------>(604)
6997              ALWAYS
6998              LET LL = PK.PROB(PK.VECTOR, PK.BAND)
6999              LET KK = .02*(EE.FAC*(HH-LL)+LL)
7000              IF (TIME.V LT BMNT)
7001                 OR (TIME.V GT EENT)
7002                 LET N.FAC = TW.NITE.FAC(WPN.ID(WEAPON))
7003              ELSE
7004                 LET N.FAC = 1
7005              ALWAYS
7006              LET KK = KK*N.FAC
7007              LET F = TW.FIRE.OTM.PTR(WPN.ID(WEAPON))
7008              LET PK.ADJ.MOV.FAC = 1
```

```
7009             IF TGT.ON.MOVE = YES
7010                IF TGT.OTM(ALPHA,BETA) = YES
7011                   FOR EVERY PK.MOVE.FACTOR
7012                   WITH PK.MOVE.FACTOR = F
7013                   FIND THE FIRST CASE
7014                   IF NONE
7015                      LET PK.ADJ.MOV.FAC = 1
7016                   ELSE
7017                      FOR EVERY PK.MOVE.BAND
7018                      WITH PK.MOV.RNG(PK.MOVE.BAND) GE RANGE
7019                      FIND THE FIRST CASE
7020                      IF NONE
7021                         LET PK.ADJ.MOV.FAC = 1
7022                      ELSE
7023                         LET PK.ADJ.MOV.FAC =
7024                         PK.MOV.FAC(PK.MOVE.FACTOR,PK.MOVE.BAND)
7025                      ALWAYS
7026                      LET KK = KK*PK.ADJ.MOV.FAC
7027                   ALWAYS
7028                ALWAYS
7029             ELSE
7030                IF FIRE.ON.MOVE = YES  AND
7031                   FIRE.OTM(ALPHA,BETA) = YES
7032                   FOR EVERY PK.F.MOVE.FACTOR
7033                   WITH PK.F.MOVE.FACTOR = F
7034                   FIND THE FIRST CASE
7035                   IF NONE
7036                      LET PK.ADJ.MOV.FAC = 1
7037                   ELSE
7038                      FOR EVERY PK.MOVE.BAND
7039                      WITH PK.MOV.RNG(PK.MOVE.BAND) GE RANGE
7040                      FIND THE FIRST CASE
7041                      IF NONE
7042                         LET PK.ADJ.MOV.FAC = 1
7043                      ELSE
7044                         LET PK.ADJ.MOV.FAC =
7045                         PK.F.MOV.FAC(PK.F.MOVE.FACTOR,
7046                         PK.MOVE.BAND)
7047                      ALWAYS
7048                      LET KK = KK*PK.ADJ.MOV.FAC
7049                   ALWAYS
7050                ALWAYS
7051             ALWAYS
7052          ALWAYS
7053       ALWAYS
7054       IF KK GT PK
7055          LET PK = KK
7056          LET SELECTED.WPN = WEAPON
7057       ELSE
7058          IF PK= 0
7059             LET SELECTED.WPN = 0
7060          ALWAYS
7061       ALWAYS
7062    ENDLOOP
7063
7064    LET PK.DEFILADE = DEFILADE      ''  %MAY83_%HWJ% FOR ATCAL
7065
7066 <--EXITROUTINE
```

```
7067 ENDROUTINE
```

INDIRECT FIRE ROUTINES

```
7074 ROUTINE AO.DETECTION                    ''                                                                    IF01
7075    GIVEN AO,
7076    VELOCITY,
7077    CANDIDATE,
7078    MODEL,
7079    ALTITUDE,
7080    SENSOR.TYPE,
7081    MAX.RANGE,
7082    MAX.DIST,
7083    TRANSITION.DIST,
7084    TARGET     ''%30DEC80_%RWF
7085
7086
7087    ADD 1 TO ANAL.CTR(52,1)  ''                                                                          \DYN_ANAL
7088    NORMALLY MODE IS INTEGER
7089    DEFINE SLANT.RANGE, MIN.PROB.LOS, MAX.PROB.LOS,
7090    EXPOSURE.DIST, TIME.EXPOSED, TIME.SCORE,
7091    AVER.PROB.LOS, EFF.TIME.SCORE, MIN.APPAR.SIZE,
7092    MAX.APPAR.SIZE, AVER.APPAR.SIZE, EFF.EXPOSED.SIZE,
7093    ELEM.PROB.DETECT, PERPENDICULAR.DIST  AS REAL VARIABLES
7094    DEFINE MAX.RANGE, MAX.DIST, TRANSITION.DIST
7095    AS REAL 1-DIMENSIONAL ARRAYS ''                                                                      \VAX
7096    DEFINE UN.PCT.OPEN, UN.PCT.WOOD, UN.PCT.TOWN AS REAL VARIABLES
7097    DEFINE DISTINCTIVENESS AS A REAL VARIABLE  '' %19MAR80_%RGR
7098    IF DEBUG = TRUE
7099       PRINT 3 LINES WITH AO, VELOCITY,CANDIDATE, MODEL, ALTITUDE,
7100       SENSOR.TYPE, TARGET THUS  ''%30DEC80_%RWF
7101       &&&&& ARG'S COMING TO AO.DETECTION ARE: AO ******, VELOCITY ****,
7102       &&&&& CANDIDATE ******, MODEL *****, ALT *****, TYPE SENS ***,
7103       &&&&& AND TARGET ****** = = =
7104    ENDIF
7105
7106    LET PERPENDICULAR.DIST = ABS.F(AO.DC.DIST(CANDIDATE))
7107    LET TGT = AO.DC.UNIT(CANDIDATE)
7108    LET LINK = AO.US.LINK(AO)
7109    LET AIR.STRIP = US.UNIT(LINK)
7110
7111    IF DEBUG = TRUE
7112       PRINT 1 LINE WITH AO, CANDIDATE, PERPENDICULAR.DIST*16 THUS
7113       &&&&& FOR AO ******, CANDIDATE ******, PERP.DIST = ****.** METERS
7114       PRINT 1 LINE WITH AO, UNIT.NOS(TGT) THUS
7115       &&&&& AO.DETECTION  AO = ****** , TARGET UNIT IS ****** = = =
7116    ENDIF
7117
7118    CALL UNIT.ENVIR(TGT) YIELDING UN.PCT.OPEN, UN.PCT.WOOD, UN.PCT.TOWN------------------------------------------------------------>(242)
7119
7120    LET DISTINCTIVENESS = UN.PCT.OPEN
7121
7122
7123    LOOP FOR EACH EQUIP IN UN.EQUIP.LIST(TGT),
7124    WHEN UE.QUANT(EQUIP) > 0
7125    DO THIS                                                  ''                                                  \1
7126       LET EQUIP.TYPE = EQ.TE.PTR(UE.ID(EQUIP))
7127       LET SLANT.RANGE = SQRT.F((PERPENDICULAR.DIST*16)**2+ALTITUDE**2) '' METERS                          \OPTIMIZE
7128       IF SLANT.RANGE > MAX.RANGE(EQUIP.TYPE)
7129    <------CYCLE
7130       OTHERWISE
7131       IF PERPENDICULAR.DIST*16 > TRANSITION.DIST(EQUIP.TYPE)
```

```
7132          LET EXPOSURE.DIST =MAX.DIST(EQUIP.TYPE)
7133          LET TIME.EXPOSED = 2.*SQRT.F(ABS.F(EXPOSURE.DIST**2 -
7134          (PERPENDICULAR.DIST*16)**2))/(VELOCITY*16)
7135          LET TIME.EXPOSED = TIME.EXPOSED*3600 ''CONVERT TO SECONDS
7136       ELSE
7137          LET EXPOSURE.DIST =SQRT.F((PERPENDICULAR.DIST*16)**2+
7138          ((PERPENDICULAR.DIST*16)+30.48)**2)
7139          LET TIME.EXPOSED = 2.*((PERPENDICULAR.DIST*16)+30.48)/(VELOCITY*16)
7140          LET TIME.EXPOSED = TIME.EXPOSED * 3600 ''CONVERT TO SECONDS
7141       ENDIF
7142       IF DEBUG = TRUE
7143          PRINT 2 LINES WITH AO, CANDIDATE, EQ.NAME(UE.ID(EQUIP)),DISTINCTIVENESS,
7144          TRANSITION.DIST(EQUIP.TYPE),  EXPOSURE.DIST, TIME.EXPOSED THUS
7145          &&&&& FOR AO ******, CANDIDATE ******, AND EQUIP ******, DIST=**.*****
7146          &&&&& TRANS.DIST = ****.**, EXP DIST = ****.**, TIME.EXP = *****.*****
7147       ENDIF
7148       IF TIME.EXPOSED > 5.0
7149          LET TIME.SCORE = 1.0
7150       ELSE
7151          LET TIME.SCORE = SQRT.F(TIME.EXPOSED/5.)
7152       ENDIF
7153       '' THE FOLLOWING CODE INTERPOLATES BETWEEN RANGES AND ELEVATIONS
7154       '' TO DETERMINE THE LOS PROBABILITY
7155       FOR EACH AO.ELEV.BD IN THE AO.EB.SET
7156       WITH AO.EB.ALTITUDE(AO.ELEV.BD) GE ALTITUDE/16
7157       FIND THE FIRST CASE
7158       IF NONE
7159    <------CYCLE
7160       OTHERWISE
7161       FOR EACH AO.RNG.BD IN THE AO.RB.SET
7162       WITH AO.RB.RANGE(AO.RNG.BD) GE PERPENDICULAR.DIST
7163       FIND THE FIRST CASE
7164       IF NONE
7165    <------CYCLE
7166       OTHERWISE
7167       IF AO.RNG.BD GT 1
7168          LET MAX = ((PERPENDICULAR.DIST-AO.RB.RANGE(AO.RNG.BD-1))/
7169          (AO.RB.RANGE(AO.RNG.BD)-AO.RB.RANGE(AO.RNG.BD-1)))*
7170          (AO.PROB.LOS(AO.ELEV.BD,AO.RNG.BD)-
7171          AO.PROB.LOS(AO.ELEV.BD,AO.RNG.BD-1)) +
7172          AO.PROB.LOS(AO.ELEV.BD,AO.RNG.BD-1)
7173       ELSE
7174          LET MAX = AO.PROB.LOS(AO.ELEV.BD,AO.RNG.BD)
7175       ENDIF
7176       IF AO.ELEV.BD GT 1
7177          IF AO.RNG.BD GT 1
7178             LET MIN = ((PERPENDICULAR.DIST-AO.RB.RANGE(AO.RNG.BD-1))/
7179             (AO.RB.RANGE(AO.RNG.BD)-AO.RB.RANGE(AO.RNG.BD-1)))*
7180             (AO.PROB.LOS(AO.ELEV.BD-1,AO.RNG.BD)-
7181             AO.PROB.LOS(AO.ELEV.BD-1,AO.RNG.BD-1)) +
7182             AO.PROB.LOS(AO.ELEV.BD-1,AO.RNG.BD-1)
7183          ELSE LET MIN = AO.PROB.LOS(AO.ELEV.BD-1,AO.RNG.BD)
7184          ENDIF
7185       ELSE LET MIN = MAX
7186       ENDIF
7187       IF MIN = MAX
7188          LET MAX.PROB.LOS = MAX/100
7189       ELSE
```

```
7190          LET MAX.PROB.LOS = ((ALTITUDE/16 - AO.EB.ALTITUDE(AO.ELEV.BD-1))/
7191          (AO.EB.ALTITUDE(AO.ELEV.BD)-AO.EB.ALTITUDE(AO.ELEV.BD-1))
7192          * (MAX - MIN) + MIN)/100
7193       ENDIF
7194       FOR EACH AO.RNG.BD IN THE AO.RB.SET
7195       WITH AO.RB.RANGE(AO.RNG.BD) GE EXPOSURE.DIST/16
7196       FIND THE FIRST CASE
7197       IF NONE
7198    <------CYCLE
7199       OTHERWISE
7200       IF AO.RNG.BD GT 1
7201          LET MAX = ((EXPOSURE.DIST/16-AO.RB.RANGE(AO.RNG.BD-1))/
7202          (AO.RB.RANGE(AO.RNG.BD)-AO.RB.RANGE(AO.RNG.BD-1)))*
7203          (AO.PROB.LOS(AO.ELEV.BD,AO.RNG.BD)-
7204          AO.PROB.LOS(AO.ELEV.BD,AO.RNG.BD-1)) +
7205          AO.PROB.LOS(AO.ELEV.BD,AO.RNG.BD-1)
7206       ELSE
7207          LET MAX = AO.PROB.LOS(AO.ELEV.BD,AO.RNG.BD)
7208       ENDIF
7209       IF AO.ELEV.BD GT 1
7210          IF AO.RNG.BD GT 1
7211             LET MIN = ((EXPOSURE.DIST/16-AO.RB.RANGE(AO.RNG.BD-1))/
7212             (AO.RB.RANGE(AO.RNG.BD)-AO.RB.RANGE(AO.RNG.BD-1)))*
7213             (AO.PROB.LOS(AO.ELEV.BD-1,AO.RNG.BD)-
7214             AO.PROB.LOS(AO.ELEV.BD-1,AO.RNG.BD-1)) +
7215             AO.PROB.LOS(AO.ELEV.BD-1,AO.RNG.BD-1)
7216          ELSE LET MIN = AO.PROB.LOS(AO.ELEV.BD-1,AO.RNG.BD)
7217          ENDIF
7218       ELSE LET MIN = MAX
7219       ENDIF
7220       IF MIN = MAX
7221          LET MIN.PROB.LOS = MIN/100
7222       ELSE
7223          LET MIN.PROB.LOS = ((ALTITUDE/16 - AO.EB.ALTITUDE(AO.ELEV.BD-1))/
7224          (AO.EB.ALTITUDE(AO.ELEV.BD)-AO.EB.ALTITUDE(AO.ELEV.BD-1))
7225          * (MAX - MIN) + MIN)/100
7226       ENDIF
7227       LET AVER.PROB.LOS = (MAX.PROB.LOS + MIN.PROB.LOS)/2.
7228       LET EFF.TIME.SCORE = AVER.PROB.LOS * TIME.SCORE
7229       LET MIN.APPAR.SIZE = (TE.PROJECTED.AREA(EQUIP.TYPE)*1000.**2.
7230       *MAO.MAGNIFICATION(MODEL)**2 ) /
7231       (EXPOSURE.DIST**2.+ ALTITUDE**2)
7232       IF PERPENDICULAR.DIST = 0.    ''%11JAN81_%RWF
7233          LET MAX.APPAR.SIZE=(TE.PROJECTED.AREA(EQUIP.TYPE)*1000.**2.
7234          *MAO.MAGNIFICATION(MODEL)**2 ) / 1.
7235       ELSE ''                                                    CHANGED OTHERWISE TO ELSE \1
7236          LET MAX.APPAR.SIZE = (TE.PROJECTED.AREA(EQUIP.TYPE)*1000.**2.
7237          *MAO.MAGNIFICATION(MODEL)**2 ) /(PERPENDICULAR.DIST*16)**2.
7238       ALWAYS
7239       LET AVER.APPAR.SIZE = (MAX.APPAR.SIZE+MIN.APPAR.SIZE)/2.
7240       LET EFF.EXPOSED.SIZE = SQRT.F(AVER.APPAR.SIZE)*DISTINCTIVENESS
7241       *EFF.TIME.SCORE
7242       IF DEBUG = TRUE
7243          PRINT 1 LINE WITH AVER.PROB.LOS, MAX.APPAR.SIZE,MIN.APPAR.SIZE THUS
7244          &&&&& PLOS = *.*****, MAX.APP.SIZE =***.****, MIN.APP.SIZE = ***.****
7245          PRINT 1 LINE WITH AO, CANDIDATE, EFF.EXPOSED.SIZE,AVER.APPAR.SIZE THUS
7246          &&&&& AO ****** EFF.EXP.SIZE FOR CAND ****** = ***.*** AVG.APP.SIZE ***.***
7247       ENDIF
```

```
7248        IF EFF.EXPOSED.SIZE LE 0.0
7249           LET EFF.EXPOSED.SIZE = 1.0
7250        ENDIF
7251        LET ELEM.PROB.DETECT = 1.00 - ( EXP.F(-.0548*EFF.EXPOSED.SIZE) )
7252        IF DEBUG = TRUE
7253           PRINT 1 LINE WITH AO, CANDIDATE, ELEM.PROB.DETECT THUS
7254           &&&&& AO ****** ELEM.PROB.DETECT FOR CANDIDATE ****** IS **.**** = = =
7255        ENDIF
7256        IF ELEM.PROB.DETECT GT 0
7257           LET NUMBER.DETECTED = BINOMIAL.F(UE.QUANT(EQUIP),ELEM.PROB.DETECT,RN.SEED)
7258           IF DEBUG = TRUE
7259              PRINT 1 LINE WITH NUMBER.DETECTED AND EQ.NAME(UE.ID(EQUIP)) THUS
7260              &&&&& NUM DET = ***** FOR EQUIP ******
7261           ENDIF
7262        ELSE
7263           LET NUMBER.DETECTED = 0
7264        ENDIF
7265        IF NUMBER.DETECTED > 0
7266           IF TR.TGT.UNIT(TARGET) = 0
7267              LET TR.TGT.UNIT(TARGET) = TGT
7268              LET TR.SENSOR.TYPE(TARGET) = ST.NAME(SENSOR.TYPE)
7269              LET TR.SENSOR.ID(TARGET) = AO
7270              LET TR.FDC(TARGET) = US.FDC(LINK)
7271              LET TR.REP.UNIT(TARGET) = AIR.STRIP
7272              LET TR.CEP(TARGET) = MAOR.CIR.ERROR(MODEL,AO.RNG.BD)
7273              ''UTILIZE NORMAL.F FUNCTION IN NEXT 2 CALCULATIONS----------------------------------------------------------------------->(641)
7274              LET TR.EST.X( TARGET ) = UN.X.COORD( TGT ) +
7275              NORMAL.F(0.0,1.0,1) * TR.CEP(TARGET) / (1.1774 * 16.)
7276              LET TR.EST.Y( TARGET ) = UN.Y.COORD( TGT ) +
7277              NORMAL.F(0.0,1.0,1) * TR.CEP(TARGET) / (1.1774 * 16.)
7278           ENDIF
7279           FOR EACH TDL IN TR.DET.LIST(TARGET)
7280           WHEN TR.DET.TE(TDL) = EQUIP.TYPE
7281           FIND THE FIRST CASE
7282           IF NONE
7283              CREATE A TR.DET.LINK CALLED TDL
7284              FILE TDL IN TR.DET.LIST(TARGET)
7285              LET TR.DET.TE(TDL) = EQUIP.TYPE
7286              LET TR.DET.ELEM.PROB(TDL) = INT.F(100. * ELEM.PROB.DETECT)
7287           ENDIF
7288           ADD NUMBER.DETECTED TO TR.DET.QUANT(TDL)  ''%30DEC80_%RWF
7289           IF TE.PGM.INDIC(TR.DET.TE(TDL)) GE 1
7290              ADD NUMBER.DETECTED TO NUM.DET.PGM
7291              IF DEBUG = TRUE
7292                 PRINT 1 LINE WITH NUM.DET.PGM THUS
7293                 &&&&& **** SADARM TGTS DETECTED
7294              ENDIF
7295           ENDIF
7296        ENDIF
7297     ENDLOOP
7298     IF NUM.DET.PGM GT SADARM.THRESHOLD AND TGT IS NOT IN
7299        AN FR.UNIT.SET AND MAO.PGM.CAP(MODEL) = 2
7300        LET TR.PGM.STATUS(TARGET) = 2
7301        LET TR.FDC(TARGET) = US.FDC(LINK)
7302        IF FD.FDC(TR.FDC(TARGET)) NE 0
7303           LET TR.FDC(TARGET) = FD.FDC(TR.FDC(TARGET))
7304        ENDIF
7305        IF FD.FDC(TR.FDC(TARGET)) NE 0
```

```
7306          LET TR.FDC(TARGET) = FD.FDC(TR.FDC(TARGET))
7307        ENDIF
7308        LET TR.MIL.WORTH(TARGET) = 1999
7309      ENDIF
7310 ENDROUTINE
```

```
7311 ROUTINE ATTRIT.SENSOR                                                                    ''                                                                
```

```
7369          REMOVE LINK FROM AVAIL.AO.LIST(SIDE)
7370          IF DEBUG = TRUE
7371             PRINT 2 LINES WITH LINK, SENSOR, SIDE THUS
7372             = = = LINK ****** FOR AO ****** REMOVED
7373             FROM AO LIST OF SIDE ** = = =
7374          ALWAYS
7375       ALWAYS
7376    ALWAYS
7377
7378    IF ST.NAME(US.SENSOR.TYPE(LINK)) = "FO"
7379       IF FO.CURRENT.TR(SENSOR) NE 0 ''                                    CHG\28 \DEBUG
7380          IF TR.PGM.STATUS(FO.CURRENT.TR(SENSOR)) = TRUE
7381 <----------EXITROUTINE
7382          OTHERWISE
7383       ALWAYS ''                                                           CHG\28 \DEBUG
7384
7385       IF FO.CURRENT.TR(SENSOR) NE 0
7386          DESTROY THE TARGET.REPORT CALLED FO.CURRENT.TR(SENSOR)
7387       ALWAYS
7388       LOOP FOR EACH TR IN FO.TGT.RPT.LIST(SENSOR)
7389       DO
7390          REMOVE TR FROM FO.TGT.RPT.LIST(SENSOR)
7391          LET TR.ABORT.TIME(TR) = -RINF.C
7392          LET TR.SENSOR.TYPE(TR) = "KILLED"
7393       ENDLOOP
7394       LOOP FOR EACH FO.DC IN FO.CAND.DET.LIST(SENSOR)
7395       DO
7396          REMOVE FO.DC FROM FO.CAND.DET.LIST(SENSOR)
7397          DESTROY THE FO.DET.CANDIDATE CALLED FO.DC
7398       ENDLOOP
7399       LOOP FOR EACH FM IN FO.CUR.FM.LIST(SENSOR)
7400       DO
7401          REMOVE FM FROM FO.CUR.FM.LIST(SENSOR)
7402          LET FM.N.VOLS(FM) = FM.FIRED.VOLS(FM)
7403       ENDLOOP
7404       IF SENSOR IS IN AN EV.S
7405          INTERRUPT FORWARD.OBSERVER CALLED SENSOR----------->(438)
7406       ALWAYS
7407       LET TIME.A(SENSOR)=-RINF.C
7408       LET US.STATUS (LINK)= 9999
7409       IF FO.CURRENT.TR(SENSOR) NE 0 ''                                    CHG\28 \DEBUG
7410          IF TR.PGM.STATUS(FO.CURRENT.TR(SENSOR)) NE TRUE
7411             RESUME FORWARD.OBSERVER CALLED SENSOR----------->(438)
7412          ALWAYS
7413       ALWAYS ''                                                           CHG\28 \DEBUG
7414    ALWAYS
7415
7416    IF ST.NAME(US.SENSOR.TYPE(LINK)) = "FL" OR
7417       ST.NAME(US.SENSOR.TYPE(LINK)) = "SD"
7418       LOOP FOR EACH KS IN PDB.KEYED.LIST(SENSOR)
7419       DO
7420          REMOVE KS FROM PDB.KEYED.LIST(SENSOR)
7421          DESTROY THE KEYED.SENSOR CALLED KS
7422       ENDLOOP
7423       FOR EACH OPERATOR IN EV.S(I.PDB.OPERATOR)
7424       WHEN PDB.SENS.ID(OPERATOR) = SENSOR
7425       FIND THE FIRST CASE
7426       IF FOUND
```

```
7427          CANCEL_THE PDB.OPERATOR CALLED OPERATOR------------------------------------------------------------------------------------------>(387)
7428          DESTROY THE PDB.OPERATOR CALLED OPERATOR
7429       ALWAYS
7430       LOOP FOR EACH DET.UNIT IN PDB.OP.Q(SENSOR)
7431       DO
7432          REMOVE DET.UNIT FROM PDB.OP.Q(SENSOR)
7433          DESTROY THE PD.DET.UNIT CALLED DET.UNIT
7434       ENDLOOP
7435       REMOVE SENSOR FROM SIDE.PDB.SET(SIDE)
7436       DESTROY THE PASSIVE.DETECTION.BASE CALLED SENSOR
7437    ALWAYS
7438
7439    IF ST.NAME(US.SENSOR.TYPE(LINK)) = "CM" OR
7440       ST.NAME(US.SENSOR.TYPE(LINK)) = "CB"
7441       LOOP FOR EACH PDB IN SIDE.PDB.SET(SIDE)
7442       DO
7443          LOOP FOR EACH KS IN PDB.KEYED.LIST(PDB)
7444          DO
7445             IF KS.SENSOR.ID(KS) = SENSOR
7446                REMOVE KS FROM PDB.KEYED.LIST(PDB)
7447                DESTROY THE KEYED.SENSOR CALLED KS
7448             ALWAYS
7449          ENDLOOP
7450       ENDLOOP
7451       LOOP FOR EACH CFRON IN EV.S(I.CFR.ON)
7452       DO
7453          IF CFR.ON.ID(CFRON) = SENSOR
7454             CANCEL_THE CFR.ON CALLED CFRON------------------------------------------------------------------------------------------>(360)
7455             DESTROY THE CFR.ON CALLED CFRON
7456          ALWAYS
7457       ENDLOOP
7458       FOR EACH CFROFF IN EV.S(I.CFR.OFF)
7459       WHEN CFR.OFF.ID(CFROFF) = SENSOR
7460       FIND THE FIRST CASE
7461       IF FOUND
7462          CANCEL_THE CFR.OFF CALLED CFROFF------------------------------------------------------------------------------------------>(359)
7463          DESTROY THE CFR.OFF CALLED CFROFF
7464       ALWAYS
7465       FOR EACH OPERATOR IN EV.S(I.CFR.OPERATOR)
7466       WHEN CFR.SENS.ID(OPERATOR) = SENSOR
7467       FIND THE FIRST CASE
7468       IF FOUND
7469          CANCEL_THE CFR.OPERATOR CALLED OPERATOR------------------------------------------------------------------------------------------>(362)
7470          DESTROY THE CFR.OPERATOR CALLED OPERATOR
7471       ALWAYS
7472       LOOP FOR EACH DET.UNIT IN CF.OP.Q(SENSOR)
7473       DO
7474          REMOVE DET.UNIT FROM CF.OP.Q(SENSOR)
7475          DESTROY THE CF.DET.UNIT CALLED DET.UNIT
7476       ENDLOOP
7477       REMOVE SENSOR FROM SIDE.CFR.SET(SIDE)
7478
7479       IF DEBUG = TRUE,
7480          PRINT 1 LINE WITH SENSOR THUS
7481          + CFR ****** HAS BEEN DESTROYED                +++
7482       ALWAYS
7483
7484       DESTROY THE CF.RADAR CALLED SENSOR
```

```
7485     ALWAYS
7486
7487     IF ST.NAME(US.SENSOR.TYPE(LINK)) = "AD"
7488        FOR EVERY .AD IN SD.ADS.SET(US.UNIT(LINK))
7489        WITH ADS.UNIT.PTR(.AD) = US.UNIT(LINK)
7490        FIND THE FIRST CASE
7491        IF NONE
7492           TRACE
7493 <----------STOP
7494        OTHERWISE
7495        SUBTRACT 1 FROM ADS.NR.SENSORS(.AD)
7496     ALWAYS
7497
7498     REMOVE LINK FROM UN.SENSOR.LIST(UNIT)
7499
7500     IF DEBUG = TRUE
7501        PRINT 2 LINES WITH LINK, SENSOR, UNIT.NOS(UNIT) THUS
7502        = = = LINK ****** FOR AO ****** REMOVED
7503        FROM SENSOR LIST OF UNIT ******** = = =
7504     ALWAYS
7505
7506     IF ST.NAME(US.SENSOR.TYPE(LINK)) NE "AO" AND
7507        ST.NAME(US.SENSOR.TYPE(LINK)) NE "FO"
7508        DESTROY THE US.LINK CALLED LINK
7509     ALWAYS
7510
7511 <--EXITROUTINE
7512 ENDROUTINE
```

```
7513 ROUTINE BTRY.FM.DEQ GIVEN MISSION     ''                                                                    IF03
7514    ADD 1 TO ANAL.CTR(54,1)   ''                                                                           \DYN_ANAL
7515    ''THIS ROUTINE INSURES THAT WHEN A FIRE UNIT FINISHES
7516    ''WITH A FIRE MISSION, THAT THE HIGHEST PRIORITY FM
7517    ''AWAITING PROCESSING IS FIRED NEXT
7518    ''UNLESS THE BATTERY IS GOING TO MOVE,
7519    ''WHEN ALL THE REMAINING FIRE MISSIONS IN THE
7520    ''QUEUE ARE KILLED OFF (BY SETTING N.VOLS TO 0)
7521    DEFINE BTRY AS AN INTEGER VARIABLE
7522    NORMALLY MODE IS INTEGER
7523    LET BTRY = FM.BTRY( MISSION )
7524    '' IF DEBUG=TRUE,
7525       '' PRINT 1 LINE WITH MISSION AND BTRY THUS
7526       '' = = = BTRY.FM.DEQ MISSION=******, BTRY=*** = = =
7527    '' ENDIF
7528    LET BY.CUR.FM( BTRY ) = F.BY.FM.QUEUE( BTRY ) ''%31JAN79_GLM
7529    IF BY.FM.QUEUE( BTRY ) IS EMPTY
7530       ''
7531       IF ANALYSIS(1) = TRUE
7532          USE UNIT 42 FOR OUTPUT
7533          PRINT 1 LINE WITH BTRY, TIME.V THUS
7534          BTRY ** CHANGES FROM STATUS 9 TO STATUS 8 AT **.*****
7535          USE UNIT 6 FOR OUTPUT
7536       ALWAYS
7537       ''
7538 <-----EXITROUTINE
7539    OTHERWISE
7540    LET FM.STATUS( BY.CUR.FM( BTRY )) = ACTIVE
7541    REACTIVATE_THE FIRE.MISSION CALLED BY.CUR.FM( BTRY ) NOW ''%1FEB79----------------------------------------------------------------------->(480)
7542 ENDROUTINE
```

```
7543 ROUTINE BTRY.FM.ENQ  ''
7544    GIVEN NEW.MISSION AND BTRY ''%31JAN79_%GLM
7545    ADD 1 TO ANAL.CTR(55,1)  ''
7546    NORMALLY MODE IS INTEGER
7547    ''THIS ROUTINE DETERMINES WHETHER THE NEW FIRE MISSION
7548    ''SHOULD BE QUEUED UP OR WHETHER IT SHOULD INTERRUPT
7549    ''THE CURRENT FIRE MISSION AND THEN BE STARTED ITSELF
7550    '' IF DEBUG=TRUE,
7551       '' PRINT 1 LINE WITH NEW.MISSION AND BTRY THUS
7552       '' = = = ENTER BTRY.FM.ENQ NEW.MISSION=******, BTRY=*** = = =
7553    '' ENDIF
7554    IF BY.CUR.FM(BTRY) IS ZERO, ''THE BATTERY IS IDLE
7555       ''
7556       IF ANALYSIS(1) = TRUE
7557          USE UNIT 42 FOR OUTPUT
7558          PRINT 1 LINE WITH BTRY, TIME.V THUS
7559          BTRY ** CHANGES FROM STATUS 8 TO STATUS 9 AT **.*****
7560          USE UNIT 6 FOR OUTPUT
7561       ALWAYS
7562       ''
7563       LET BY.CUR.FM(BTRY) = NEW.MISSION
7564       LET FM.STATUS(NEW.MISSION) = ACTIVE
7565    ELSE ''THE BATTERY ALREADY HAS A FIRE MISSION ACTIVE
7566       LET FM.STATUS(NEW.MISSION) = HOLD
7567    ENDIF
7568    '' IF DEBUG=TRUE,
7569       '' PRINT 1 LINE WITH BY.CUR.FM AND N.BY.FM.QUEUE THUS
7570       '' = = = EXITN BTRY.FM.ENQ CUR.MISSION=******, N.QUEUE=*** = = =
7571    '' ENDIF
7572 ENDROUTINE
```

```
7573 ROUTINE CFR.DEGRADE                                                                    IF05
7574    GIVEN
7575    PROB,
7576    RADAR,
7577    MODEL
7578    YIELDING DET.PROB
7579    ADD 1 TO ANAL.CTR(56,1)  ''                                                         \DYN_ANAL
7580    NORMALLY MODE IS INTEGER
7581    DEFINE RADAR, MODEL, RATE, VOLLEY, ENEMY, WIDTH AS INTEGER VARIABLES
7582    DEFINE PROB, DET.PROB AS REAL VARIABLES
7583    LET RATE = 0
7584    LET WIDTH = INT.F(MCFR.SEARCH.WIDTH(MODEL)/2.)
7585    IF UN.COLOR(RADAR) = BLUE
7586       LET X1 = UN.X.COORD(RADAR)
7587       LET X2 = X1 + L.MCFR.RH.LIST(MODEL)
7588    ELSE
7589       LET X2 = UN.X.COORD(RADAR)
7590       LET X1 = X2 - L.MCFR.RH.LIST(MODEL)
7591    ENDIF
7592    LET Y1 = UN.Y.COORD(RADAR) - WIDTH
7593    LET Y2 = UN.Y.COORD(RADAR) + WIDTH
7594    LOOP FOR EACH VOLLEY IN IF.RATE.LIST
7595    DO THIS                                                                             \1
7596       LET ENEMY = BY.UNIT(IF.V.BTRY(VOLLEY))
7597       IF (TIME.V-IF.V.TIME(VOLLEY))*MINUTES.V > 1.
7598          REMOVE VOLLEY FROM IF.RATE.LIST
7599          DESTROY THE IF.VOLLEY CALLED VOLLEY
7600       ELSE
7601          IF X1 <= UN.X.COORD(ENEMY) <= X2 AND
7602             Y1 <= UN.Y.COORD(ENEMY) <= Y2
7603             ADD 1 TO RATE
7604          ENDIF
7605       ENDIF
7606    ENDLOOP
7607    IF RATE >= 50
7608       LET DET.PROB = PROB * 0.2
7609    ELSE
7610       IF RATE >= 35
7611          LET DET.PROB = PROB * (1.53 - (REAL.F(RATE)*0.4/15.)) ''                      \OPTIMIZE
7612       ELSE
7613          LET DET.PROB = PROB * (1.0-(REAL.F(RATE)*0.4/35.))
7614       ENDIF
7615    ENDIF
7616 ENDROUTINE
```

```
7617 ROUTINE CFR.DETECTION                                                    ''                                                    IF06
7618     GIVEN
7619     CFR,
7620     BTRY,
7621     RANGE
7622     ADD 1 TO ANAL.CTR(57,1)  ''                                                                                              \DYN_ANAL
7623     NORMALLY MODE IS INTEGER
7624     DEFINE BTRY AS AN INTEGER VARIABLE
7625     DEFINE DIRECTION, GAMMA, DET.PROB, DELTA.X, DELTA.Y AS REAL VARIABLES
7626     DEFINE G1AMMA, G2AMMA, PROB AS REAL VARIABLES
7627     LET LINK = CFR.US.LINK(CFR)
7628     LET MODEL = US.MODEL(LINK)
7629     LET RADAR = US.UNIT(LINK)
7630     FOR EACH DETECTION IN CF.OP.Q(CFR)
7631     WHEN CF.D.BTRY(DETECTION) = BTRY
7632     FIND THE FIRST CASE
7633     IF FOUND    '' THIS IS A DUPLICATE SENSING, SO
7634 <-------EXITROUTINE
7635     OTHERWISE  '' COMPUTE DIRECTION FROM CFR TO ENEMY BTRY
7636     LET DELTA.X = REAL.F(UN.X.COORD(BY.UNIT(BTRY)) - UN.X.COORD(RADAR))
7637     LET DELTA.Y = REAL.F(UN.Y.COORD(BY.UNIT(BTRY)) - UN.Y.COORD(RADAR))
7638     IF DELTA.X = 0.0
7639        IF DELTA.Y > 0.0
7640           LET DIRECTION = PI.C/2.
7641        ELSE
7642           LET DIRECTION = 3.*PI.C/2.
7643        ENDIF
7644     ELSE
7645        IF DELTA.Y = 0.0
7646           IF DELTA.X > 0.0
7647              LET DIRECTION = 0.
7648           ELSE
7649              LET DIRECTION = PI.C
7650           ENDIF
7651        ELSE
7652           LET DIRECTION = ARCTAN.F(DELTA.Y,DELTA.X)
7653           IF DIRECTION < 0.0
7654              ADD 2.0*PI.C TO DIRECTION
7655           ENDIF
7656        ENDIF
7657     ENDIF
7658     ''  COMPUTE ANGLES WHICH DEFINE EXTENT OF RADAR FAN
7659     LET GAMMA = REAL.F(MCFR.SWEEP.ANGLE(MODEL))*PI.C/360.
7660     LET G1AMMA=REAL.F(CFR.ORIENTATION(CFR))/100.-GAMMA
7661     LET G2AMMA=REAL.F(CFR.ORIENTATION(CFR))/100.+GAMMA
7662     IF G1AMMA<0. AND DIRECTION>PI.C
7663        LET DIRECTION=DIRECTION-2.*PI.C  ''                                                                                   \OPTIMIZE
7664     ELSE
7665        IF G2AMMA>2.*PI.C
7666           LET G2AMMA=G2AMMA-2.*PI.C
7667           LET G1AMMA=G1AMMA-2.*PI.C
7668           IF DIRECTION>PI.C
7669              LET DIRECTION=DIRECTION-2.*PI.C
7670           ENDIF
7671        ENDIF
7672     ENDIF
7673     IF G1AMMA <= DIRECTION <= G2AMMA    '' ENEMY BTRY IS W/I FAN
7674        FOR EACH RH IN MCFR.RH.LIST(MODEL)
```

```
7675        WHEN CFR.RH.RANGE(RH) >= RANGE
7676        FIND THE FIRST CASE
7677        IF FOUND
7678           LET PROB = REAL.F(CFR.DET.PROB(RH))/100.
7679           LET CPE = CFR.CIR.ERROR(RH)
7680        ELSE
7681  <-------EXITROUTINE
7682        ENDIF
7683        CALL CFR.DEGRADE GIVING PROB, RADAR, MODEL-------------------------------------------------------------->(165)
7684        YIELDING DET.PROB
7685        IF RANDOM.F(1) > DET.PROB    '' THERE IS NO DETECTION, SO
7686  <-------EXITROUTINE
7687        OTHERWISE
7688        CREATE A CF.DET.UNIT CALLED THIS.DET
7689        LET CF.D.BTRY(THIS.DET) = BTRY
7690        LET CF.D.CPE(THIS.DET) = CPE
7691        LET CF.D.PD(THIS.DET) = INT.F(DET.PROB*100.)
7692        LET CF.D.PRIORITY(THIS.DET) = TU.MIL.WORTH(UN.TYPE.UNIT(BY.UNIT(BTRY)))
7693        FILE THIS.DET IN CF.OP.Q(CFR)
7694        IF CF.OPERATOR(CFR) = IDLE
7695           SCHEDULE_A CFR.OPERATOR GIVING CFR, THIS.DET NOW-------------------------------------------------->(362)
7696        ENDIF
7697     ENDIF
7698  ENDROUTINE
```

```
7699 ROUTINE CHK.COMP.TR ''                                                                                    IF07
7700    GIVEN TARGET, FDC, SENSOR ''%31JAN79_%GLM
7701    YIELDING DUPLICATE
7702    ADD 1 TO ANAL.CTR(58,1)  ''                                                                            \DYN_ANAL
7703    NORMALLY MODE IS INTEGER
7704    '' IF DEBUG = TRUE,
7705       '' PRINT 1 LINE WITH TARGET, FDC THUS
7706       '' = = = CHK.COMP.TR  TARGET = ****** , FDC = ****  = = =
7707    '' ENDIF
7708    LOOP FOR EACH TR IN FD.COMPLETE.LIST( FDC ),
7709    WHEN TR.TGT.UNIT( TR ) = TR.TGT.UNIT( TARGET ),
7710    DO THIS
7711       CALL COMPARE.TRS GIVEN TARGET, TR, SENSOR------------------------------------------------------------------------------------------------------>(171)
7712       YIELDING DUPLICATE
7713       IF DUPLICATE= TRUE
7714          IF SENSOR = 0
7715             LET TR.ABORT.TIME( TR ) = MAX.F( TR.ABORT.TIME( TR ),
7716             TR.ABORT.TIME( TARGET ) )
7717          ENDIF
7718 <--------EXITROUTINE
7719       OTHERWISE
7720    ENDLOOP
7721 ENDROUTINE
```

```
7722 ROUTINE CHK.FD.TR                                                                                    IF08
7723    GIVEN
7724    TARGET,
7725    FDC,
7726    SENSOR
7727    YIELDING
7728    DUPLICATE
7729
7730    ADD 1 TO ANAL.CTR(59,1)  ''                                                                       \DYN_ANAL
7731    ''THIS ROUTINE PREVENTS DUPLICATION OF HE OR ICM MISSIONS
7732
7733    NORMALLY MODE IS INTEGER
7734
7735    LOOP FOR EACH TR IN FD.TR.QUEUE( FDC ) IN REVERSE ORDER
7736    UNLESS TR = FD.CUR.TR(FDC) OR
7737    TR.MISSION.TYPE(TR) = "FASCAM" OR
7738    TR.MISSION.TYPE(TR) = "ILLUM" OR
7739    TR.MISSION.TYPE(TR) = "SMOKE"
7740    DO THIS
7741       IF TR.TGT.UNIT( TR ) = TR.TGT.UNIT( TARGET )
7742          CALL COMPARE.TRS------------------------------------------------------------------------------>(171)
7743          GIVEN
7744          TARGET,
7745          TR,
7746          SENSOR
7747          YIELDING
7748          DUPLICATE
7749          IF DUPLICATE = TRUE
7750             ''KEEP THE TGT ALREADY IN QUEUE
7751             IF SENSOR = 0
7752                CALL COMBINE.TRS------------------------------------------------------------------------>(170)
7753                GIVEN
7754                TR,
7755                TARGET
7756             ALWAYS
7757 <----------EXITROUTINE
7758          OTHERWISE
7759       ALWAYS
7760    ENDLOOP
7761
7762 <--EXITROUTINE
7763 ENDROUTINE
```

```
7764 ROUTINE COMBINE.TRS                                                    ''          IF09
7765     GIVEN
7766     TARGET,
7767     TR
7768
7769     ADD 1 TO ANAL.CTR(60,1)  ''                                                    \DYN_ANAL
7770     NORMALLY MODE IS REAL
7771     DEFINE TARGET, TR, LINK1, LINK2 AS INTEGER  VARIABLES
7772
7773     LET I.CEP1.SQ = 1.0 / TR.CEP( TARGET ) ** 2 ''                                 \OPTIMIZE
7774     LET I.CEP2.SQ = 1.0 / TR.CEP( TR ) ** 2
7775     LET W1 = I.CEP1.SQ / ( I.CEP1.SQ  +  I.CEP2.SQ )
7776     LET W2 = I.CEP2.SQ / ( I.CEP1.SQ  +  I.CEP2.SQ )
7777     LET TR.CEP( TARGET ) = W1 * TR.CEP( TARGET ) + W2 * TR.CEP( TR )
7778     LET TR.EST.X( TARGET ) = W1 * TR.EST.X( TARGET ) + W2 * TR.EST.X( TR )
7779     LET TR.EST.Y( TARGET ) = W1 * TR.EST.Y( TARGET ) + W2 * TR.EST.Y( TR )
7780     LOOP FOR EACH LINK2 IN TR.DET.LIST( TR ),
7781     DO
7782        REMOVE LINK2 FROM TR.DET.LIST( TR )
7783        FOR EACH LINK1 IN TR.DET.LIST( TARGET )
7784        WHEN TR.DET.TE( LINK1 ) = TR.DET.TE( LINK2 )
7785        FIND THE FIRST CASE
7786        IF FOUND
7787           LET TR.DET.QUANT( LINK1 ) = MAX.F( TR.DET.QUANT( LINK1 ),
7788           TR.DET.QUANT( LINK2 ) )
7789           DESTROY THE TR.DET.LINK CALLED LINK2
7790     <-----CYCLE
7791        OTHERWISE
7792        FILE LINK2 IN TR.DET.LIST( TARGET )
7793     ENDLOOP
7794
7795 <--EXITROUTINE
7796 ENDROUTINE
```

```
7797 ROUTINE COMPARE.TRS                                                                                    IF10
7798    GIVEN
7799    TARGET,
7800    TR,
7801    SENSOR
7802    YIELDING
7803    DUPLICATE
7804
7805    ADD 1 TO ANAL.CTR(61,1)  ''                                                                         \DYN_ANAL
7806    NORMALLY MODE IS INTEGER
7807    DEFINE REQ.TIME AS A REAL VARIABLE
7808
7809    LET DUPLICATE = FALSE
7810    IF TR.MOVE( TARGET ) = TR.MOVE( TR )
7811       IF SENSOR NE 0 AND
7812          TR.SENSOR.ID( TARGET ) NE TR.SENSOR.ID( TR)
7813 <--------EXITROUTINE
7814       OTHERWISE
7815
7816       CALL TIME.REQ-------------------------------------------------------------------------------------->(663)
7817       GIVEN
7818       TARGET,
7819       TR
7820       YIELDING
7821       REQ.TIME
7822       IF TR.RECVD.TIME( TARGET ) - TR.RECVD.TIME( TR ) < REQ.TIME
7823          IF SENSOR EQ 0 AND
7824             TU.CAT(TR.EST.TU(TARGET)) NE TU.CAT(TR.EST.TU(TR))
7825 <-----------EXITROUTINE
7826          OTHERWISE
7827
7828          CALL PROXIMITY.REQ-----------------------------------------------------------------------------------------
```

```
7843 ROUTINE COPY                                                          ''                IF11
7844     GIVEN
7845     .COPY.FROM,
7846     .COPY.TO
7847
7848     ADD 1 TO ANAL.CTR(62,1)  ''                                                         \DYN_ANAL
7849     ''THIS ROUTINE IS USED TO COPY ALL THE ATTRIBUTES OF ONE
7850     ''TARGET REPORT INTO ANOTHER - IT IS CALLED WHEN A REQUEST
7851     ''FOR ADDITIONAL FIRE(RFAF) IS MADE OR A TOT FOLLOW ON IS REQUIRED -
7852     ''IN BOTH CASES, THE FDC MUST USE THE SAME ATTRIBUTES IN ORDER
7853     ''TO DELIVER FIRES ON THE SAME TARGET LOCATION
7854
7855     NORMALLY MODE IS INTEGER
7856
7857     LET TR.MISSION.TYPE(.COPY.TO) = TR.MISSION.TYPE(.COPY.FROM)
7858     LET TR.FDC(.COPY.TO)          = TR.FDC(.COPY.FROM)
7859     LET TR.FDC.STATUS(.COPY.TO)   = TR.FDC.STATUS(.COPY.FROM)
7860     LET TR.SENSOR.TYPE(.COPY.TO)  = TR.SENSOR.TYPE(.COPY.FROM)
7861     LET TR.SENSOR.ID(.COPY.TO)    = TR.SENSOR.ID(.COPY.FROM)
7862     LET TR.REP.UNIT(.COPY.TO)     = TR.REP.UNIT(.COPY.FROM)
7863     LET TR.TGT.UNIT(.COPY.TO)     = TR.TGT.UNIT(.COPY.FROM)
7864     LET TR.MIL.WORTH(.COPY.TO)    = TR.MIL.WORTH(.COPY.FROM)
7865     LET TR.PGM.STATUS(.COPY.TO)   = TR.PGM.STATUS(.COPY.FROM)
7866     LET TR.REM.EFFECTS(.COPY.TO)  = TR.REM.EFFECTS(.COPY.FROM)
7867     LET TR.REQ.EFFECTS(.COPY.TO)  = TR.REQ.EFFECTS(.COPY.FROM)
7868     LET TR.CUM.EFFECTS(.COPY.TO)  = TR.CUM.EFFECTS(.COPY.FROM)
7869     LET TR.MOVE(.COPY.TO)         = TR.MOVE(.COPY.FROM)
7870     LET TR.EST.X(.COPY.TO)        = TR.EST.X(.COPY.FROM)
7871     LET TR.EST.Y(.COPY.TO)        = TR.EST.Y(.COPY.FROM)
7872     LET TR.EST.RADIUS(.COPY.TO)   = TR.EST.RADIUS(.COPY.FROM)
7873     LET TR.EST.TU(.COPY.TO)       = TR.EST.TU(.COPY.FROM)
7874     LET TR.RECVD.TIME(.COPY.TO)   = TR.RECVD.TIME(.COPY.FROM)
7875     LET TR.ABORT.TIME(.COPY.TO)   = TR.ABORT.TIME(.COPY.FROM)
7876     LET TR.CEP(.COPY.TO)          = TR.CEP(.COPY.FROM)
7877     LET TR.TOT.STATUS(.COPY.TO)   = TR.TOT.STATUS(.COPY.FROM)
7878     LET TR.ASGND.BATS(.COPY.TO)   = TR.ASGND.BATS(.COPY.FROM)
7879
7880 <--EXITROUTINE
7881 ENDROUTINE
```

```
7882 ROUTINE DUST.EFFECTS                                                                                    IF12
7883    GIVEN
7884    .UNIT,
7885    .XIMPACT,
7886    .YIMPACT,
7887    .RADIUS,
7888    .DURATION
7889
7890    ADD 1 TO ANAL.CTR(63,1)  ''                                                                  \DYN_ANAL
7891    ''THIS ROUTINE IS CALLED FROM FINAL COVERAGE TO DEGRADE
7892    ''VISIBILITY DUE TO DUST FROM HE MUNITIONS
7893
7894    NORMALLY MODE IS INTEGER
7895    DEFINE .A, .C, .XIMPACT, .YIMPACT AS REAL VARIABLES ''                          CHG\24 \ARG_MODE_ERROR
7896
7897    IF .RADIUS LE 0 OR .DURATION LE 0 OR .UNIT IS NOT IN FR.UNIT.SET
7898 <------EXITROUTINE
7899    OTHERWISE
7900
7901    ''UPDATE UNIT POSITIONS
7902    LOOP FOR EACH .FORCE IN BTL.FORCE.SET(UN.BATTLE.INDEX(.UNIT))
7903    DO
7904       LOOP FOR EACH .UN IN FR.UNIT.SET(.FORCE)
7905       WITH UN.STATUS(.UN) = ADVANCING OR
7906       UN.STATUS(.UN) = WITHDRAWING OR
7907       UN.STATUS(.UN) = ADV.TO.WITH
7908       DO
7909          FOR EVERY .MOVE IN EV.S(I.MOVE)
7910          WITH MV.UNIT(.MOVE) = .UN
7911          FIND THE FIRST CASE
7912          IF NONE
7913             CALL ERROR.STOP------------------------------------------------------------------------------------------------------------>(604)
7914          ALWAYS
7915          CANCEL_THE MOVE CALLED .MOVE---------------------------------------------------------------------------------------------->(381)
7916          DESTROY THE MOVE CALLED .MOVE
7917          CALL CHANGE.LOC------------------------------------------------------------------------------------------------------------>( 83)
7918          GIVEN
7919          .UN
7920       ENDLOOP
7921    ENDLOOP
7922
7923    FOR EVERY .FORCE OF BTL.FORCE.SET(UN.BATTLE.INDEX(.UNIT))
7924    WITH FR.SIDE(.FORCE) = UN.COLOR(.UNIT)
7925    FIND THE FIRST CASE
7926    IF NONE
7927       CALL ERROR.STOP------------------------------------------------------------------------------------------------------------>(604)
7928    ALWAYS
7929    FOR EVERY .ENEMY.FORCE OF BTL.FORCE.SET(UN.BATTLE.INDEX(.UNIT))
7930    WITH .ENEMY.FORCE NE .FORCE
7931    FIND THE FIRST CASE
7932    IF NONE
7933       CALL ERROR.STOP------------------------------------------------------------------------------------------------------------>(604)
7934    ALWAYS
7935
7936    ''CHECK LINES OF SIGHT TO THE TARGET UNIT
7937    LET .FLAG = NO
7938    LOOP FOR EACH .ENEMY IN FR.UNIT.SET(.ENEMY.FORCE)
7939    UNLESS UN.X.COORD(.UNIT) = UN.X.COORD(.ENEMY) AND
```

```
7940    UN.Y.COORD(.UNIT) = UN.Y.COORD(.ENEMY)
7941    DO
7942       IF UN.X.COORD(.UNIT) NE UN.X.COORD(.ENEMY)
7943          ''DISTANCE BETWEEN LINE AX+BY+C=0 AND (X1,Y1) IS
7944          ''ABS((A*X1+B*Y1+C)/SQRT(A**2+B**2))
7945          LET .A = (UN.Y.COORD(.UNIT) - UN.Y.COORD(.ENEMY)) /
7946          (UN.X.COORD(.UNIT) - UN.X.COORD(.ENEMY))
7947          ''B = -1
7948          LET .C = UN.Y.COORD(.UNIT) - .A * UN.X.COORD(.UNIT)
7949          LET .DIS = ABS.F(.A*.XIMPACT - .YIMPACT + .C) /
7950          SQRT.F(.A**2 + 1)
7951       ELSE
7952          LET .DIS = ABS.F(UN.Y.COORD(.UNIT) - UN.Y.COORD(.ENEMY))
7953       ALWAYS
7954       IF .DIS LE .RADIUS
7955          CALL BLOCK.LOS----------------------------------------------------------------------------------------->( 80)
7956          GIVEN
7957          .UNIT,
7958          .ENEMY,
7959          .DURATION
7960          LET .FLAG = YES
7961          PRINT 1 LINE WITH .UNIT, .ENEMY, .DURATION THUS
7962          = = = DUST BLOCKED LOS BETWEEN UNIT **** AND **** FOR **** MIN = = =
7963       ALWAYS
7964    ENDLOOP
7965
7966    IF .FLAG = YES
7967       ''AT LEAST ONE PAIR OF UNITS WAS AFFECTED
7968       LOOP FOR EACH .ENEMY IN FR.UNIT.SET(.ENEMY.FORCE)
7969       WITH UN.STATUS(.ENEMY) = ADVANCING OR
7970       UN.STATUS(.ENEMY) = WITHDRAWING OR
7971       UN.STATUS(.ENEMY) = ADV.TO.WITH
7972       DO
7973          FOR EVERY .MOVE OF EV.S(I.MOVE)
7974          WITH MV.UNIT(.MOVE) = .ENEMY
7975          FIND THE FIRST CASE
7976          IF NONE
7977             CALL ERROR.STOP------------------------------------------------------------------------------------->(604)
7978          ALWAYS
7979          CANCEL_THE MOVE CALLED .MOVE------------------------------------------------------------------------->(381)
7980          DESTROY THE MOVE CALLED .MOVE
7981          CALL MIN.MOVE---------------------------------------------------------------------------------------->(106)
7982          GIVEN
7983          .ENEMY
7984       ENDLOOP
7985       IF UN.STATUS(.UNIT) = ADVANCING OR
7986          UN.STATUS(.UNIT) = WITHDRAWING OR
7987          UN.STATUS(.UNIT) = ADV.TO.WITH
7988          FOR EVERY .MOVE OF EV.S(I.MOVE)
7989          WITH MV.UNIT(.MOVE) = .UNIT
7990          FIND THE FIRST CASE
7991          IF NONE
7992             CALL ERROR.STOP------------------------------------------------------------------------------------->(604)
7993          ALWAYS
7994          CANCEL_THE MOVE CALLED .MOVE------------------------------------------------------------------------->(381)
7995          DESTROY THE MOVE CALLED .MOVE
7996          CALL MIN.MOVE---------------------------------------------------------------------------------------->(106)
7997          GIVEN
```

```
7998          .UNIT
7999        ALWAYS
8000     ALWAYS
8001
8002 <--EXITROUTINE
8003 ENDROUTINE
```

```
8004 ROUTINE EST.COVERAGE GIVEN      ''                                                    IF13
8005     ''                                                                %JUN79 - USE WITH WT.VOL
8006     ''THIS ROUTINE COMPUTES THE EXPECTED FRACTION OF A TAREGT
8007     ''THAT WILL BE COVERED BY THE EFFECTS FROM A BATTERY VOLLEY
8008     MISSION,
8009     CLASS,
8010     ROUND
8011     YIELDING FRACT.COVERAGE AND RNG.HACK  '' RANGE HACK RQD IN WEIGHT VOLS
8012     ADD 1 TO ANAL.CTR(64,1)  ''                                                    \DYN_ANAL
8013     NORMALLY MODE IS REAL
8014     DEFINE TARGET, RANGE, ROUND, MISSION,
8015     RAD.OF.EFFECTS, ICHK, RNG.HACK,
8016     RND.INT.CPE,TOT.INT.CPE,RDIF1,RDIF2,CDIF1,
8017     MIN.RNG.INT,MAX.RNG.INT,MIN.CPE.INT,MAX.CPE.INT AS INTEGER VARIABLES
8018     DEFINE CLASS AS A TEXT VARIABLE ''                                    CHG\27 \TEXT
8019     DEFINE ZERO.REP.TMS AS A SAVED, INTEGER VARIABLE ''                        \DEBUG
8020     LET TARGET = FM.TGT(MISSION)
8021     LET RANGE = FM.RANGE(MISSION)
8022     IF DEBUG=TRUE,
8023        PRINT 1 LINE WITH TARGET, MISSION, RANGE THUS
8024        = = = EST.COVERAGE TARGET=******, MISSION=******  RANGE= ******
8025     ENDIF
8026     IF CLASS = "HE"
8027        FOR EACH RNG.HACK IN HE.TB.RH.LIST( ROUND, BY.TYPE(FM.BTRY(MISSION))),
8028        WHEN HE.RH.RANGE( RNG.HACK ) >= RANGE,
8029        FIND THE FIRST CASE
8030        IF NONE
8031           LET RNG.HACK = 0
8032           IF DEBUG=TRUE,
8033              PRINT 1 LINE THUS
8034              = = = EST.COV = OUT OF HE RANGE - EXIT
8035           ENDIF
8036 <--------EXITROUTINE
8037        OTHERWISE
8038        ''THE FOLLOWING TWO IF STATEMENTS PICK OUT SELECTIVE INDICES FOR CERTAIN
8039        ''VALUES FROM THE MUNITION RECORDS..... %AUG80_%HWJ
8040        LET ICHK=MOD.F(RNG.HACK,2)
8041        IF ICHK > 0
8042           LET MIN.RNG.INT = RNG.HACK
8043           LET MIN.CPE.INT = RNG.HACK
8044        ENDIF
8045        IF ICHK = 0
8046           LET MAX.CPE.INT = RNG.HACK
8047           LET MIN.CPE.INT = RNG.HACK - 1
8048           LET MAX.RNG.INT = RNG.HACK
8049           LET MIN.RNG.INT   RNG.HACK - 1
8050        ENDIF
8051        ''COMPUTE INTERPOLATED VALUE FOR TOTAL HE CPE BETWEEN 0 AND MIN RNG
8052        IF RANGE <= HE.RH.RANGE(MIN.RNG.INT)
8053           LET RDIF1=HE.RH.RANGE(MIN.RNG.INT)-RANGE
8054           LET RDIF2=HE.RH.RANGE(MIN.RNG.INT)
8055           LET CDIF1=HE.RH.TOTAL.CPE(MIN.RNG.INT)
8056           LET DIV = RDIF1/RDIF2
8057           LET TOT.INT.CPE=HE.RH.TOTAL.CPE(MIN.CPE.INT)-(DIV*CDIF1)
8058           ''COMPUTE INTERPOLATED VALUE FOR ROUND HE CPE BETWEEN 0 AND MIN
8059           LET CDIF1=HE.RH.ROUND.CPE(MIN.CPE.INT)
8060           LET RND.INT.CPE=HE.RH.ROUND.CPE(MIN.CPE.INT)-(DIV*CDIF1)
8061        ELSE
```

```
8062          ''COMPUTE INTERPOLATED VALUE FOR TOTAL HE CPE BETWEEN HE.RH.RANGE.(MIN & MAX)
8063          LET RDIF1=HE.RH.RANGE(MAX.RNG.INT)-RANGE
8064          LET RDIF2=HE.RH.RANGE(MAX.RNG.INT)-HE.RH.RANGE(MIN.RNG.INT)
8065          LET CDIF1=HE.RH.TOTAL.CPE(MAX.CPE.INT)-HE.RH.TOTAL.CPE(MIN.CPE.INT)
8066          LET DIV = RDIF1/RDIF2
8067          LET TOT.INT.CPE=HE.RH.TOTAL.CPE(MAX.CPE.INT)-(DIV*CDIF1)
8068          ''COMPUTE INTERPOLATED VALUE FOR ROUND HE CPE BETWEEN HE.RH.RANGE(MIN & MAX)
8069          LET CDIF1=HE.RH.ROUND.CPE(MAX.CPE.INT)-HE.RH.ROUND.CPE(MIN.CPE.INT)
8070          LET RND.INT.CPE=HE.RH.ROUND.CPE(MAX.CPE.INT)-(DIV*CDIF1)
8071       ENDIF
8072       IF DEBUG=TRUE,
8073          PRINT 4 LINES WITH HE.RH.TOTAL.CPE(RNG.HACK) AND HE.RH.ROUND.CPE(RNG.HACK) AND
8074          TOT.INT.CPE AND RND.INT.CPE AND HE.RH.RANGE(MIN.RNG.INT) AND
8075          HE.RH.RANGE(MAX.RNG.INT) AND ICHK AND RNG.HACK THUS
8076          $$$$ OLD TOT HE CPE = ********, OLD RND HE CPE = ********, $$$$
8077          $$$$ NEW TOT HE CPE = ********, NEW RND HE CPE = ********,   $$$$
8078          $$$$  MIN RNG   = ********,    MAX RNG   = ********,   $$$$
8079          $$$$  ICHK = ***,   RNG.HACK = ****    $$$$
8080       ENDIF
8081       LET RAD.OF.EFFECTS = HE.VOLLEY.RAD( ROUND )
8082    ELSE   ''CLASS = "ICM"
8083       FOR EACH RNG.HACK IN IC.TB.RH.LIST( ROUND, BY.TYPE(FM.BTRY(MISSION))),
8084       WHEN IC.RH.RANGE( RNG.HACK ) >= RANGE,
8085       FIND THE FIRST CASE
8086       IF NONE
8087          LET RNG.HACK = 0
8088          IF DEBUG=TRUE,
8089             PRINT 1 LINE THUS
8090             = = = EST.COV. = ICM OUT OF RANGE - EXIT
8091          ENDIF
8092 <--------EXITROUTINE
8093       OTHERWISE
8094       ''THE FOLLOWING TWO IF STATEMENTS PICK OUT SELECTIVE INDICES FOR CERTAIN
8095       ''VALUES FROM THE MUNITION RECORDS..... %AUG80_%HWJ
8096       LET ICHK=MOD.F(RNG.HACK,2)
8097       IF ICHK > 0
8098          LET MIN.RNG.INT = RNG.HACK
8099          LET MIN.CPE.INT = RNG.HACK
8100       ENDIF
8101       IF ICHK = 0
8102          LET MAX.CPE.INT = RNG.HACK
8103          LET MIN.CPE.INT = RNG.HACK - 1
8104          LET MAX.RNG.INT = RNG.HACK
8105          LET MIN.RNG.INT = RNG.HACK - 1
8106       ENDIF
8107       ''COMPUTE INTERPOLATED VALUE FOR TOTAL IC CPE BETWEEN 0 AND MIN RNG
8108       IF RANGE <= IC.RH.RANGE(MIN.RNG.INT)
8109          LET RDIF1=IC.RH.RANGE(MIN.RNG.INT)-RANGE
8110          LET RDIF2=IC.RH.RANGE(MIN.RNG.INT)
8111          LET CDIF1=IC.RH.TOTAL.CPE(MIN.RNG.INT)
8112          LET DIV = RDIF1/RDIF2
8113          LET TOT.INT.CPE=IC.RH.TOTAL.CPE(MIN.CPE.INT)-(DIV*CDIF1)
8114          ''COMPUTE INTERPOLATED VALUE FOR ROUND IC CPE BETWEEN 0 AND MIN
8115          LET CDIF1=IC.RH.ROUND.CPE(MIN.CPE.INT)
8116          LET RND.INT.CPE=IC.RH.ROUND.CPE(MIN.CPE.INT)-(DIV*CDIF1)
8117       ELSE
8118          ''COMPUTE INTERPOLATED VALUE FOR TOTAL IC CPE BETWEEN IC.RH.RANGE.(MIN & MAX)
8119          LET RDIF1=IC.RH.RANGE(MAX.RNG.INT)-RANGE
```

```
8120          LET RDIF2=IC.RH.RANGE(MAX.RNG.INT)-IC.RH.RANGE(MIN.RNG.INT)
8121          LET CDIF1=IC.RH.TOTAL.CPE(MAX.CPE.INT)-IC.RH.TOTAL.CPE(MIN.CPE.INT)
8122          LET DIV = RDIF1/RDIF2
8123          LET TOT.INT.CPE=IC.RH.TOTAL.CPE(MAX.CPE.INT)-(DIV*CDIF1)
8124          ''COMPUTE INTERPOLATED VALUE FOR ROUND IC CPE BETWEEN IC.RH.RANGE(MIN & MAX)
8125          LET CDIF1=IC.RH.ROUND.CPE(MAX.CPE.INT)-IC.RH.ROUND.CPE(MIN.CPE.INT)
8126          LET RND.INT.CPE=IC.RH.ROUND.CPE(MAX.CPE.INT)-(DIV*CDIF1)
8127       ENDIF
8128       IF DEBUG=TRUE,
8129          PRINT 4 LINES WITH IC.RH.TOTAL.CPE(RNG.HACK) AND IC.RH.ROUND.CPE(RNG.HACK) AND
8130          TOT.INT.CPE AND RND.INT.CPE AND IC.RH.RANGE(MIN.RNG.INT) AND
8131          IC.RH.RANGE(MAX.RNG.INT) AND ICHK AND RNG.HACK THUS
8132          $$$$ OLD TOT IC CPE = ********, OLD RND IC CPE = ********, $$$$
8133          $$$$ NEW TOT IC CPE = ********, NEW RND IC CPE = ********   $$$$
8134          $$$$  MIN RNG   = ********,    MAX RNG   = ********,   $$$$
8135          $$$$  ICHK = ***,   RNG.HACK = ****     $$$$
8136       ENDIF
8137       LET RAD.OF.EFFECTS = IC.VOLLEY.RAD( ROUND )
8138    ENDIF
8139    IF CLASS = "ICM"
8140       LET K = 2.
8141    ELSE
8142       LET K = 5.
8143    ALWAYS
8144    LET REP.PREC = 0.573 * RND.INT.CPE * K
8145    LET DEP.PREC = 0.573 * RND.INT.CPE * K
8146    LET REP.TM = SQRT.F((0.573*TOT.INT.CPE)**2
8147    + (0.573*TR.CEP(FM.TGT(MISSION)))**2)
8148    LET DEP.TM = REP.TM
8149    LET LENGTH.TGT = SQRT.F(PI.C) * TR.EST.RADIUS(TARGET)  ''                    CHG\09
8150    LET WIDTH.TGT = SQRT.F(PI.C) * TR.EST.RADIUS(TARGET)   ''                    \MISSPELLING
8151    IF CLASS = "ICM"
8152       LET LENGTH.1RND.PAT = SQRT.F(PI.C) * ((RANGE*16./1000.)
8153       * IC.TB.SLOPE(ROUND,BY.TYPE(FM.BTRY(MISSION)))
8154       + IC.TB.INTERCEPT(ROUND,BY.TYPE(FM.BTRY(MISSION))))
8155       + REP.PREC
8156       LET WIDTH.1RND.PAT = SQRT.F(PI.C) * ((RANGE*16./1000.)
8157       * IC.TB.SLOPE(ROUND,BY.TYPE(FM.BTRY(MISSION)))
8158       + IC.TB.INTERCEPT(ROUND,BY.TYPE(FM.BTRY(MISSION))))
8159       + DEP.PREC
8160    ELSE ''HE
8161       ''UTILIZE HE.WLA FUNCTION IN NEXT CALCULATION ———————————————————————————— \1>(636)
8162       LET LETHAL.AREA = HE.WLA(TARGET,FM.BTRY(MISSION),ROUND,RANGE,2,MISSION)
8163       LET LENGTH.1RND.PAT = 2. * SQRT.F(LETHAL.AREA * 0.307/PI.C)
8164       + REP.PREC
8165       LET WIDTH.1RND.PAT =(2. * SQRT.F(LETHAL.AREA * 0.307/PI.C))/0.307
8166       + DEP.PREC
8167    ALWAYS
8168    LET AREA.ADJ.PAT = LENGTH.1RND.PAT * WIDTH.1RND.PAT
8169    LET LVP = LENGTH.1RND.PAT + RAD.OF.EFFECTS/SQRT.F(PI.C)
8170    LET WVP = WIDTH.1RND.PAT + RAD.OF.EFFECTS/SQRT.F(PI.C)
8171    IF TIME.V GT 5.65 AND TIME.V LT 5.67
8172       PRINT 1 LINE WITH N.BY.HOW.SET(FM.BTRY(MISSION)),FM.BTRY(MISSION),
8173       TB.RND.PER.LAUNCH(BY.TYPE(FM.BTRY(MISSION))) THUS
8174       >>>>> *** TUBES IN BTRY ****, **** TUBES PER LAUNCHER
8175    ALWAYS
8176    LET OVERLAP = REAL.F(N.BY.HOW.SET(FM.BTRY(MISSION))
8177    *TB.RND.PER.LAUNCH(BY.TYPE(FM.BTRY(MISSION))))
```

```
8178     * AREA.ADJ.PAT/(LVP*WVP)
8179     IF OVERLAP > 1.0
8180        LET OVERLAP = 1.0
8181     ALWAYS
8182     IF REP.TM IS ZERO, ''                                                    CHG\27 \DEBUG
8183        ADD 1 TO ZERO.REP.TMS
8184        LET FRACT.COVERAGE = 1.0
8185        WRITE AS "*****************************************", /
8186        WRITE ZERO.REP.TMS AS "EST.COVERAGE WITH REP.TM=0, OCCURRENCE=",
8187        I 6, /
8188        WRITE AS "*****************************************", /
8189 <------RETURN
8190     OTHERWISE
8191     LET A = (LVP + LENGTH.TGT)/(2.96*REP.TM)
8192     CALL FRAC.COMPUTE GIVEN A YIELDING FRAC------------------------------------------------------------------------>(335)
8193     LET FL = FRAC
8194     LET A = ABS.F(LVP - LENGTH.TGT)/(2.96*REP.TM)
8195     CALL FRAC.COMPUTE GIVEN A YIELDING FRAC------------------------------------------------------------------------>(335)
8196     SUBTRACT FRAC FROM FL
8197     LET FRACT.COVERAGE = 2.96*REP.TM*FL/LENGTH.TGT
8198     LET  A = (WVP + WIDTH.TGT)/(2.96*DEP.TM)
8199     CALL FRAC.COMPUTE GIVEN A YIELDING FRAC------------------------------------------------------------------------>(335)
8200     LET FW = FRAC
8201     LET  A = ABS.F(WVP - WIDTH.TGT)/(2.96*DEP.TM)
8202     CALL FRAC.COMPUTE GIVEN A YIELDING FRAC------------------------------------------------------------------------>(335)
8203     SUBTRACT FRAC FROM FW
8204     LET FRACT.COVERAGE = FRACT.COVERAGE * 2.96 * DEP.TM * FW/WIDTH.TGT
8205     ''
8206     '' WHEN THE FRACTION OF COVERAGE IS LARGER THAN 1.0 IT MUST BE BECAUSE
8207     ''    THE DAMAGE AREA IS LARGER THAN THE TARGET == THEREFORE THE
8208     ''    FRACTIONAL COVERAGE IS SET TO 1.0
8209     ''
8210     IF FRACT.COVERAGE > 1.0 AND LVP*WVP > LENGTH.TGT*WIDTH.TGT
8211        LET FRACT.COVERAGE = 1.0
8212     ALWAYS
8213     IF DEBUG = TRUE
8214        PRINT 1 LINE WITH FRACT.COVERAGE THUS
8215        === EST.COV === FRAC.COV = *.*******
8216     ALWAYS
8217 <--RETURN
8218 ENDROUTINE
```

```
8219 ROUTINE EST.MIL.WORTH                    ''                                    IF14
8220     GIVEN TARGET,
8221     FEBA.DISTANCE
8222     YIELDING MIL.WORTH
8223     '' CALLED BY TARGET.ANALYSIS
8224     ADD 1 TO ANAL.CTR(65,1)   ''                                               \DYN_ANAL
8225     NORMALLY MODE IS INTEGER
8226     DEFINE DIST.FACTOR AS A REAL VARIABLE
8227     LET UN.MIL.WORTH = TU.MIL.WORTH(UN.TYPE.UNIT(TR.TGT.UNIT(TARGET)))
8228     IF FEBA.DISTANCE <= 1   '' UNIT IS WITHIN 16 METERS OF THE FEBA-
8229        '' A NEGATIVE VALUE MAY BE PASSED IF THE
8230        '' UNIT IS TEMPORARILY AHEAD OF THE FEBA %RGR
8231        '' IN THE ATTACK OR ON PATROL
8232        LET DIST.FACTOR = 1.0
8233     ELSE   '' THE FEBA DISTANCE IS GREATER THAN 16 METERS
8234        LET DIST.FACTOR = 1 / SQRT.F(REAL.F(FEBA.DISTANCE))
8235     ALWAYS
8236     LET MIL.WORTH = INT.F(UN.MIL.WORTH * DIST.FACTOR)
8237     IF DEBUG = TRUE
8238        PRINT 1 LINE WITH TARGET, MIL.WORTH AND FEBA.DISTANCE*16 THUS
8239        = = = EST.MIL.WORTH  TARGET ****** IS WORTH ******, AT  ****** METERS = = =
8240     ENDIF
8241 ENDROUTINE
```

```
8242 ROUTINE FASCAM.COMPUTATION                                                                    IF15
8243     GIVEN
8244      MSN
8245
8246     ''THIS ROUTINE IS CALLED BY FA.BN.ASGN TO DETERMINE WHICH
8247     ''FASCAM MUNITION IS TO BE USED
8248
8249     ADD 1 TO ANAL.CTR(66,1)  ''                                                               \DYN_ANAL
8250     NORMALLY MODE IS INTEGER
8251
8252     LET .TARGET = FM.TGT(.MSN)
8253     LET .TB = BY.TYPE(FM.BTRY(.MSN))
8254     ''UTILIZE FEBA.BAND FUNCTION IN NEXT ASSIGNMENT ———————————————————————————————————— \1>(634)
8255     LET .DFFB = FEBA.BAND(.TARGET)
8256     LET .CAT = TU.CAT(UN.TYPE.UNIT(TR.TGT.UNIT(.TARGET)))
8257
8258     FOR EVERY .LINK IN TB.TM.LIST(.TB)
8259     WITH TB.TM.CLASS(.LINK) = "FASCAM" AND
8260     FMM.MAX.RANGE(TB.TM(.LINK)) GE FM.RANGE(.MSN)
8261     FIND THE FIRST CASE
8262     IF FOUND
8263        LET .FASCAM = TB.TM(.LINK)
8264        LET FM.TM(.MSN) = .FASCAM
8265        LET FM.TM.CLASS(.MSN) = "FASCAM"
8266        LET .REQ.VOLS = FASCAM.VOLLEYS * TR.REM.EFFECTS(.TARGET) / 100
8267        LET FM.N.VOLS(.MSN) = MIN.F(.REQ.VOLS,
8268        CDT.MAX.VOLS(.CAT, .DFFB, .TB))
8269        LET TR.REM.EFFECTS(.TARGET) = TR.REM.EFFECTS(.TARGET)
8270        * (1. - (FM.N.VOLS(.MSN) / .REQ.VOLS))
8271     ALWAYS
8272
8273     IF MF.DEBUG = TRUE
8274        PRINT 2 LINES WITH
8275        TIME.V,
8276        TR.REP.UNIT(.TARGET),
8277        TR.TGT.UNIT(.TARGET),
8278        .MSN, .TARGET, FM.BTRY(.MSN),
8279        .TB, .FASCAM, FASCAM.VOLLEYS, TR.REM.EFFECTS(.TARGET) THUS
8280        ===FASCAM.COMPU AT **.***HRS, REP UNT= ****, TGT= ****, MSN=******,
8281        TGT RPT=******, BTRY=***, TB=***, MUNS=***, VOLS=***, REM EFF=****
8282     ALWAYS
8283
8284 <—EXITROUTINE
8285 ENDROUTINE
```

```
   ROUTINE FA.BN.ASGN                                                                                                    IF16
8287    GIVEN
8288    TARGET,
8289    BN,
8290    DURATION
8291
8292    ADD 1 TO ANAL.CTR(67,1)  ''                                                                                   \DYN_ANAL
8293    ''THIS ROUTINE MODELS THE PROCESS WITHIN AN FDC OF ASSIGNING THE
8294    ''AVAILALE FIRE UNITS IN A BATTALION TO FIRE ON A TARGET REPORT.
8295    ''THE ROUND TO FIRE BY EACH FIRE UNIT IS SELECTED AND THE QUANTITY
8296    ''OF THAT ROUND TO FIRE IS COMPUTED. FIRE MISSIONS ARE GENERATED
8297    ''AND SENT TO THE APROPRIATE FIRE UNITS. THE ESTIMATED
8298    ''FIRE MISSION EFFECTS ARE COMPUTED AND APPLIED AGAINST THE
8299    ''REQUIRED EFFECTS FOR THE TARGET REPORT IN ORDER TO COMPUTE
8300    ''THE REMAINING EFFECTS NECESSARY TO MEET THE "ATTACK CRITERIA".
8301
8302    NORMALLY MODE IS INTEGER
8303    DEFINE TARGET,BN,BTRY AS INTEGER VARIABLES
8304    DEFINE EST.MSN.EFFECTS AS A REAL VARIABLE
8305    DEFINE RAP.MSN TO MEAN 1
8306    DEFINE NON.RAP TO MEAN 2
8307    DEFINE TOT.FOLLOW TO MEAN 2
8308    DEFINE PERSONNEL TO MEAN 1
8309
8310    ''UTILIZE FEBA.BAND FUNCTION IN NEXT ASSIGMENT ———————————————————————————————————————————————————————— \1>(634)
8311    LET DFFB = FEBA.BAND(TARGET)
8312    IF DFFB = 33
8313       LET TR.REM.EFFECTS(TARGET) = 0
8314 <———EXITROUTINE
8315    OTHERWISE
8316
8317    LET TU = TR.EST.TU(TARGET)
8318    IF TU IS ZERO, ''                                                                                  CHG\06  \ZERO_SUB
8319       LIST ATTRIBUTES OF TARGET.REPORT CALLED TARGET  ''                                                      \REMOVE?
8320       WRITE AS /
8321       LET TU = 1
8322    ALWAYS
8323
8324    ''RANK AVAILABLE FIRE UNITS BY PROXIMITY TO THE TARGET REPORT LOCATION.
8325    LOOP FOR EACH BTRY IN BN.BTRY.SET(BN),
8326    DO ''                                                                                                          \1
8327       IF BY.STATUS(BTRY) GE 2  ''EITHER OCCUPYING OR ALREADY IN POSITION
8328          IF DEBUG = TRUE
8329             PRINT 1 LINE WITH BY.STATUS(BTRY) THUS
8330             *** FA.BN.ASGN BTRY NOT AVAIL, STATUS = ***
8331          ALWAYS
8332    <———CYCLE
8333       OTHERWISE
8334       IF N.BY.HOW.SET(BTRY) LT TB.MIN.HOW(BY.TYPE(BTRY))
8335          IF DEBUG = TRUE
8336             PRINT 1 LINE WITH N.BY.HOW.SET(BTRY),TB.MIN.HOW(BY.TYPE(BTRY)) THUS
8337             *** FA.BN.ASGN BTRY NOT AVAIL, N.BTRY.HOW.SET *****  TB.MIN.HOW *****
8338          ALWAYS
8339    <———CYCLE
8340       OTHERWISE
8341       IF (TIME.V*MINUTES.V - UN.LAST.ARTY.ENG(BY.UNIT(BTRY))) LT
8342          TB.SUPPRESS.TIME(BY.TYPE(BTRY)) AND
8343          TIME.V * MINUTES.V LE BY.STOP.FASCAM.SUPP(BTRY)
```

```
8344            IF DEBUG = TRUE
8345               PRINT 2 LINE WITH TIME.V*MINUTES.V,UN.LAST.ARTY.ENG(BY.UNIT(BTRY)),
8346               TB.SUPPRESS.TIME(BY.TYPE(BTRY)),BY.STOP.FASCAM.SUPP(BTRY)
8347               THUS
8348               = = = FA.BN.ASGN BTRY NOT AVAIL, TIME (MIN) ****.**** LAST.ENG.TIME ****.****
8349               TB.SUPPRESS.TIME ****, FASCAM.SUPP.TIME *****.****
8350            ALWAYS
8351      <-----CYCLE
8352         OTHERWISE
8353         IF DEBUG = TRUE
8354            PRINT 1 LINE WITH BTRY, TARGET, TIME.V THUS
8355            = = = FA.BN.ASGN  BTRY *** AVAILABLE FOR TARGET ****** AT **.*****
8356         ALWAYS
8357
8358         ''THERE MUST BE ENOUGH TUBES AND THE BTRY MUST NOT BE
8359         ''SUPPRESSED.
8360
8361         ''FIRE MISSION MUST BE MORE IMPORTANT THAN MINIMUM MILITARY
8362         ''WORTH THRESHOLD ASSOCIATED WITH THE NUMBER OF FIRE MISSIONS
8363         ''ALREADY WAITING TO BE FIRED BY THIS FIRE UNIT.
8364
8365         LET .SIDE = UN.COLOR(BY.UNIT(BTRY))
8366         FOR EACH TE.LINK IN THE TU.TE.LIST(UN.TYPE.UNIT(BY.UNIT(BTRY)))
8367         WITH TU.TE.ID(TE.LINK) = PERSONNEL
8368         FIND THE FIRST CASE
8369         IF NONE
8370            PRINT 1 LINE WITH UN.TYPE.UNIT(BY.UNIT(BTRY)) THUS
8371            = = ERROR = = NO PERSONNEL IN TYPE UNIT *****
8372            TRACE
8373      <--------STOP
8374         OTHERWISE
8375         FOR EACH EQ IN THE UN.EQUIP.LIST(BY.UNIT(BTRY)) WITH
8376         EQ.TE.PTR(UE.ID(EQ)) = PERSONNEL
8377         FIND THE FIRST CASE
8378         IF NONE
8379            PRINT 1 LINE WITH BY.UNIT(BTRY) THUS
8380            = = ERROR = = NO PERSONNEL IN UNIT ****
8381            TRACE
8382      <--------STOP
8383         OTHERWISE
8384         IF  REAL.F(UE.QUANT(EQ)/TU.TE.QUANT(TE.LINK)) LT
8385            REAL.F(ARTY.DECIMATE(.SIDE)/100) ''NOT ENOUGH TROOPS ON HAND
8386            IF DEBUG = TRUE
8387               PRINT 1 LINE WITH UE.QUANT(EQ),EQ.NAME(UE.ID(EQ)),TU.TE.QUANT(TE.LINK),
8388               TE.NAME(TU.TE.ID(TE.LINK)),REAL.F(UE.QUANT(UE.ID(EQ))/TU.TE.QUANT(TE.LINK))
8389               THUS
8390               &%$# NOT ASGN'D, *** ****** IN UNIT VS *** ****** ORIG, RATIO = ******.**
8391            ALWAYS
8392      <-----CYCLE
8393         OTHERWISE
8394         LET .NUM.IN.QUEUE = MIN.F(5, N.BY.FM.QUEUE(BTRY) +1)
8395         IF TR.MIL.WORTH(TARGET) < TB.MW.THRESHOLD(BY.TYPE(BTRY),
8396            .NUM.IN.QUEUE)
8397      <-----CYCLE
8398         OTHERWISE
8399
8400         IF TR.TOT.STATUS( TARGET ) = TRUE  OR  TR.TGT.UNIT(TARGET) = 0
8401            ''THIS FIRE MISSION REQUIRES SCHEDULING
```

```
8402          ''UTILIZE BTRY.AVAILABLE FUNCTION IN NEXT TEST ———————————————————————————————————————— \1>(629)
8403          IF BTRY.AVAILABLE(BTRY, TARGET, DURATION)=FALSE
8404    <————————CYCLE
8405          OTHERWISE
8406       ALWAYS
8407
8408       ''WHEN BTRY IS NOT FIRING TOT, DON'T USE IT FOR FOLLOW ON
8409       IF TR.TOT.STATUS( TARGET ) = TOT.FOLLOW AND
8410          TR.TGT.UNIT(FM.TGT(BY.CUR.FM(BTRY))) NE TR.TGT.UNIT(TARGET)
8411    <———CYCLE
8412       OTHERWISE
8413
8414       CREATE A FIRE.MISSION CALLED FM
8415       IF DEBUG = TRUE
8416          PRINT 1 LINE WITH BTRY,TARGET,FM,TIME.V THUS
8417          = = = FA.BN.ASGN  BTRY ***  TARGET ****** FM ****** CREATED AT **.*****
8418       ALWAYS
8419       LET FM.Q.SIZE(FM) = N.BY.FM.QUEUE(BTRY)
8420       LET FM.PRIORITY(FM) = TR.MIL.WORTH(TARGET)
8421       LET FM.BTRY(FM) = BTRY
8422       LET FM.TGT(FM) = TARGET
8423       ''UTILIZE EST.RANGE FUNCTION IN NEXT ASSIGNMENT ———————————————————————————————————————— \1>(632)
8424       LET FM.RANGE(FM) = INT.F( EST.RANGE(TARGET,BY.UNIT(BTRY)) )
8425       IF FM.RANGE(FM)<=TB.MAX.RANGE(BY.TYPE(BTRY))
8426          LET FM.RAP.FLAG(FM) = NON.RAP
8427          FILE THE FM IN THE BN.CAN.FM.SET
8428       ELSE
8429          IF FM.RANGE(FM)<=TB.MAX.RAP.RANGE(BY.TYPE(BTRY)) AND
8430             TR.MISSION.TYPE(TARGET) NE "SMOKE" AND
8431             TR.MISSION.TYPE(TARGET) NE "ILLUM" AND
8432             TR.MISSION.TYPE(TARGET) NE "FASCAM"
8433             LET FM.RAP.FLAG(FM) = RAP.MSN
8434             FILE THE FM IN THE BN.CAN.FM.SET
8435          ELSE ''THE TARGET IS OUT OF RANGE OF THIS BTRY
8436             NOW CLEAN.UP.FIRE.MISSIONS ''———————————————————————————————————— CHG\19  \DEBUG>(274)
8437             GIVEN FM
8438             DESTROY THE FIRE.MISSION CALLED FM
8439             IF DEBUG = TRUE
8440                PRINT 1 LINE WITH BTRY,TARGET,FM,TIME.V THUS
8441                = = = FA.BN.ASGN  BTRY ***  TARGET ****** FM ****** DESTROY AT **.*****
8442             ALWAYS
8443          ALWAYS
8444       ALWAYS
8445    ENDLOOP
8446
8447    ''ASSIGN FIRE MISSIONS IN RANKED ORDER
8448    LET BAT.NUM = 0
8449    LOOP UNTIL BN.CAN.FM.SET IS EMPTY,
8450    DO
8451       IF N.TR.FM.LIST(TARGET) GE DT.MAX.BATS(DFFB,TU)
8452          LET TR.REM.EFFECTS(TARGET) = 0
8453       ALWAYS
8454       REMOVE THE FIRST FM FROM BN.CAN.FM.SET
8455       LET BAT.NUM = BAT.NUM + 1
8456       IF TR.REM.EFFECTS( TARGET ) <= 0  , ''THIS FM NOT NEEDED
8457          NOW CLEAN.UP.FIRE.MISSIONS ''———————————————————————————————————— CHG\19  \DEBUG>(274)
8458          GIVEN FM
8459          DESTROY THE FIRE.MISSION CALLED FM
```

```
8460        ELSE
8461           ''COMPUTE FIRE MISSION DETAILS
8462           LET HE.OR.ICM.=FALSE
8463           IF TR.MISSION.TYPE(TARGET) ="ILLUM"
8464              CALL ILLUM.COMPUTATION------------------------------------------------------------------------------>(199)
8465              GIVEN
8466              FM
8467           ELSE
8468              IF TR.MISSION.TYPE(TARGET) ="SMOKE"
8469                 CALL SMOKE.COMPUTATION---------------------------------------------------------------------------->(233)
8470                 GIVEN
8471                 FM
8472              ELSE
8473                 IF TR.MISSION.TYPE(TARGET) ="FASCAM"
8474                    CALL FASCAM.COMPUTATION------------------------------------------------------------------------>(181)
8475                    GIVEN
8476                    FM
8477                 ELSE
8478                    CALL HE.OR.ICM.COMPUTATION--------------------------------------------------------------------->(196)
8479                    GIVEN
8480                    BAT.NUM,
8481                    FM
8482                    YIELDING
8483                    EST.MSN.EFFECTS
8484                    LET HE.OR.ICM.=TRUE
8485                 ALWAYS
8486              ALWAYS
8487           ALWAYS
8488           IF FM.N.VOLS(FM) = 0
8489              NOW CLEAN.UP.FIRE.MISSIONS ''------------------------------------------------------ CHG\19  \DEBUG>(274)
8490              GIVEN FM
8491              DESTROY THE FIRE.MISSION CALLED FM
8492           ELSE
8493              FILE FM IN TR.FM.LIST(TARGET)
8494              IF TR.START.TIME( TARGET ) > TIME.V  AND
8495                 TR.TOT.STATUS( TARGET ) NE TOT.FOLLOW
8496                 FILE FM IN BY.SCHD.LIST( FM.BTRY( FM ) )
8497                 LET FM.START.TIME( FM ) = TR.START.TIME( TARGET )
8498                 ACTIVATE_THE FIRE.MISSION CALLED FM------------------------------------------------------------>(480)
8499                 AT FM.START.TIME(FM)
8500              ELSE
8501                 ''ACCOUNT FOR FIRE MISSION PROCESSING TIME
8502                 LET PROCESS.TIME =
8503                 FD.MAX.TIME(TR.FDC(TARGET))*BAT.NUM/3.
8504                 ACTIVATE_THE FIRE.MISSION CALLED FM------------------------------------------------------------>(480)
8505                 IN PROCESS.TIME/10. MINUTES
8506              ALWAYS
8507
8508              IF HE.OR.ICM.=TRUE
8509                 ''DETERMINE EXPECTED EFFTS, ADJUST REQUIRED EFFTS
8510                 CALL REM.EFFECTS.COMPUTATION------------------------------------------------------------------->(214)
8511                 GIVEN
8512                 EST.MSN.EFFECTS,
8513                 TARGET
8514              ALWAYS
8515           ALWAYS
8516        ALWAYS
8517     ENDLOOP
```

```
8518
8519 <--EXITROUTINE
8520 ENDROUTINE
```

```
8521 ROUTINE FD.EFFECTS.REQ                                                    ''                                                    IF17
8522     GIVEN
8523     TARGET,
8524     SIDE
8525     YIELDING
8526     REQ.EFFECTS
8527
8528     ADD 1 TO ANAL.CTR(68,1)  ''                                                                                          \DYN_ANAL
8529     ''THIS ROUTINE COMPUTES THE ATTACK CRITERIA FOR THE TARGET REPORT
8530
8531     NORMALLY MODE IS INTEGER
8532
8533     IF TR.MOVE( TARGET ) = TRUE   '' TARGET IS MOVING
8534        LET REQ.EFFECTS = REQ.EFF.MOVING(SIDE)
8535     ELSE
8536        LET REQ.EFFECTS = REQ.EFF.STA(SIDE)
8537     ALWAYS
8538
8539     IF TR.MISSION.TYPE(TARGET) = "ILLUM" OR
8540        TR.MISSION.TYPE(TARGET) = "SMOKE" OR
8541        TR.MISSION.TYPE(TARGET) = "FASCAM"
8542        LET REQ.EFFECTS = 1
8543     ALWAYS
8544
8545 <--EXITROUTINE
8546 ENDROUTINE
```

```
8547 ROUTINE FINAL.COVERAGE GIVEN TARGET,MISSION,ENVIR,POST,TE,FUZE ''                    IF18
8548     ''THIS ROUTINE COMPUTES EXPECTED COVERAGE OF A TARGET
8549     ''BY A BATTERY VOLLEY GIVEN A SPECIFIC TARGET LOCATION ERROR
8550     YIELDING FRAC.COVERAGE AND AX
8551     ADD 1 TO ANAL.CTR(69,1)  ''                                                     \DYN_ANAL
8552     NORMALLY MODE IS REAL
8553     DEFINE ENVIR,POST,TE,FUZE AS INTEGER VARIABLES
8554     DEFINE PERSONNEL TO MEAN 1
8555     DEFINE TARGET, MISSION, RAD.OF.TARGET, RNG.HACK,
8556     RAD.OF.EFFECTS, RANGE, ROUND,
8557     RND.INT.CPE,TOT.INT.CPE,RDIF1,RDIF2,CDIF1,ICHK,
8558     MIN.RNG.INT, MAX.RNG.INT, MIN.CPE.INT, MAX.CPE.INT, I AND J
8559     AS INTEGER VARIABLES
8560     DEFINE OVERLAP.COORD AS A 1-DIMENSIONAL REAL ARRAY
8561     DEFINE CLASS AS A TEXT VARIABLE ''                                              \TEXT
8562     LET RAD.OF.TARGET = UN.RADIUS( TR.TGT.UNIT( TARGET ) )
8563     LET ROUND = FM.TM( MISSION )
8564     LET CLASS = FM.TM.CLASS( MISSION )
8565     IF CLASS = "SDM"
8566        LET ROUND = SDM.TM
8567     ENDIF
8568     LET RANGE = FM.RANGE(MISSION)
8569     IF CLASS = "HE"  OR CLASS = "SDM"
8570        FOR EACH RNG.HACK IN HE.TB.RH.LIST( ROUND, BY.TYPE(FM.BTRY(MISSION))),
8571        WHEN HE.RH.RANGE( RNG.HACK ) >= RANGE,
8572        FIND THE FIRST CASE
8573        IF NONE
8574           PRINT 1 LINE WITH RANGE AND BY.TYPE(FM.BTRY(MISSION)) THUS
8575           ==== ERROR ==== FINAL COV, CAN'T FIND RNG.HACK FOR RANGE = ******* TB *****
8576           FOR EACH HE.RANGE.HACK IN THE HE.TB.RH.LIST(ROUND,BY.TYPE(FM.BTRY(MISSION)))
8577           LIST ATTRIBUTES OF HE.RANGE.HACK
8578           TRACE
8579 <---------STOP
8580        OTHERWISE
8581        ''THE FOLLOWING TWO IF STATEMENTS PICK OUT SELECTIVE INDICES FOR CERTAIN
8582        ''VALUES FROM THE MUNITION RECORDS..... %AUG80_%HWJ
8583        LET ICHK=MOD.F(RNG.HACK,2)
8584        IF ICHK > 0
8585           LET MIN.RNG.INT = RNG.HACK
8586           LET MIN.CPE.INT = RNG.HACK
8587        ENDIF
8588        IF ICHK = 0
8589           LET MAX.CPE.INT = RNG.HACK
8590           LET MIN.CPE.INT = RNG.HACK - 1
8591           LET MAX.RNG.INT = RNG.HACK
8592           LET MIN.RNG.INT = RNG.HACK - 1
8593        ENDIF
8594        ''COMPUTE INTERPOLATED VALUE FOR TOTAL HE CPE BETWEEN 0 AND MIN RNG
8595        IF RANGE <= HE.RH.RANGE(MIN.RNG.INT)
8596           LET RDIF1=HE.RH.RANGE(MIN.RNG.INT)-RANGE
8597           LET RDIF2=HE.RH.RANGE(MIN.RNG.INT)
8598           LET CDIF1=HE.RH.TOTAL.CPE(MIN.RNG.INT)
8599           LET DIV = RDIF1/RDIF2
8600           LET TOT.INT.CPE=HE.RH.TOTAL.CPE(MIN.CPE.INT)-(DIV*CDIF1)
8601           IF TOT.INT.CPE = 0
8602              LET TOT.INT.CPE = 1 ''FOR SHORT RANGE MISSIONS
8603           ALWAYS
8604           ''COMPUTE INTERPOLATED VALUE FOR ROUND HE CPE BETWEEN 0 AND MIN
```

```
8605          LET CDIF1=HE.RH.ROUND.CPE(MIN.CPE.INT)
8606          LET RND.INT.CPE=HE.RH.ROUND.CPE(MIN.CPE.INT)-(DIV*CDIF1)
8607       ELSE
8608          ''COMPUTE INTERPOLATED VALUE FOR TOTAL HE CPE BETWEEN HE.RH.RANGE.(MIN & MAX)
8609          LET RDIF1=HE.RH.RANGE(MAX.RNG.INT)-RANGE
8610          LET RDIF2=HE.RH.RANGE(MAX.RNG.INT)-HE.RH.RANGE(MIN.RNG.INT)
8611          LET CDIF1=HE.RH.TOTAL.CPE(MAX.CPE.INT)-HE.RH.TOTAL.CPE(MIN.CPE.INT)
8612          LET DIV = RDIF1/RDIF2
8613          LET TOT.INT.CPE=HE.RH.TOTAL.CPE(MAX.CPE.INT)-(DIV*CDIF1)
8614          IF TOT.INT.CPE = 0
8615             LET TOT.INT.CPE = 1 ''FOR SHORT RANGE MISSIONS
8616          ALWAYS
8617          ''COMPUTE INTERPOLATED VALUE FOR ROUND HE CPE BETWEEN HE.RH.RANGE(MIN & MAX)
8618          LET CDIF1=HE.RH.ROUND.CPE(MAX.CPE.INT)-HE.RH.ROUND.CPE(MIN.CPE.INT)
8619          LET RND.INT.CPE=HE.RH.ROUND.CPE(MAX.CPE.INT)-(DIV*CDIF1)
8620       ENDIF
8621       IF DEBUG=TRUE,
8622          PRINT 4 LINES WITH HE.RH.TOTAL.CPE(RNG.HACK) AND HE.RH.ROUND.CPE(RNG.HACK) AND
8623          TOT.INT.CPE AND RND.INT.CPE AND HE.RH.RANGE(MIN.RNG.INT) AND
8624          HE.RH.RANGE(MAX.RNG.INT) AND ICHK AND RNG.HACK THUS
8625          $$FC OLD TOT HE CPE = ********, OLD RND HE CPE = ********, $$FC
8626          $$FC NEW TOT HE CPE = ********, NEW RND HE CPE = ********,   $$FC
8627          $$FC  MIN RNG   = ********,    MAX RNG   = ********,    $$FC
8628          $$FC  ICHK = ***,   RNG.HACK = ****   $$FCC
8629       ENDIF
8630       LET RAD.OF.EFFECTS = HE.VOLLEY.RAD( ROUND )
8631       IF CLASS = "SDM"
8632          LET RAD.OF.EFFECTS = SDM.VOLLEY.RADIUS
8633       ENDIF
8634    ELSE   ''CLASS = "ICM"
8635       FOR EACH RNG.HACK IN IC.TB.RH.LIST( ROUND, BY.TYPE(FM.BTRY(MISSION))),
8636       WHEN IC.RH.RANGE( RNG.HACK ) >= RANGE,
8637       FIND THE FIRST CASE
8638       ''THE FOLLOWING TWO IF STATEMENTS PICK OUT SELECTIVE INDICES FOR CERTAIN
8639       ''VALUES FROM THE MUNITION RECORDS..... %AUG80_%HWJ
8640       LET ICHK=MOD.F(RNG.HACK,2)
8641       IF ICHK > 0
8642          LET MIN.RNG.INT = RNG.HACK
8643          LET MIN.CPE.INT = RNG.HACK
8644       ENDIF
8645       IF ICHK = 0
8646          LET MAX.CPE.INT = RNG.HACK
8647          LET MIN.CPE.INT = RNG.HACK - 1
8648          LET MAX.RNG.INT = RNG.HACK
8649          LET MIN.RNG.INT = RNG.HACK - 1
8650       ENDIF
8651       ''COMPUTE INTERPOLATED VALUE FOR TOTAL IC CPE BETWEEN 0 AND MIN RNG
8652       IF RANGE <= IC.RH.RANGE(MIN.RNG.INT)
8653          LET RDIF1=IC.RH.RANGE(MIN.RNG.INT)-RANGE
8654          LET RDIF2=IC.RH.RANGE(MIN.RNG.INT)
8655          LET CDIF1=IC.RH.TOTAL.CPE(MIN.RNG.INT)
8656          LET DIV = RDIF1/RDIF2
8657          LET TOT.INT.CPE=IC.RH.TOTAL.CPE(MIN.CPE.INT)-(DIV*CDIF1)
8658          IF TOT.INT.CPE = 0
8659             LET TOT.INT.CPE = 1 ''FOR SHORT RANGE MISSIONS
8660          ALWAYS
8661          ''COMPUTE INTERPOLATED VALUE FOR ROUND IC CPE BETWEEN 0 AND MIN
8662          LET CDIF1=IC.RH.ROUND.CPE(MIN.CPE.INT)
```

```
8663          LET RND.INT.CPE=IC.RH.ROUND.CPE(MIN.CPE.INT)-(DIV*CDIF1)
8664       ELSE
8665          ''COMPUTE INTERPOLATED VALUE FOR TOTAL IC CPE BETWEEN IC.RH.RANGE.(MIN & MAX)
8666          LET RDIF1=IC.RH.RANGE(MAX.RNG.INT)-RANGE
8667          LET RDIF2=IC.RH.RANGE(MAX.RNG.INT)-IC.RH.RANGE(MIN.RNG.INT)
8668          LET CDIF1=IC.RH.TOTAL.CPE(MAX.CPE.INT)-IC.RH.TOTAL.CPE(MIN.CPE.INT)
8669          LET DIV = RDIF1/RDIF2
8670          LET TOT.INT.CPE=IC.RH.TOTAL.CPE(MAX.CPE.INT)-(DIV*CDIF1)
8671          IF TOT.INT.CPE = 0
8672             LET TOT.INT.CPE = 1 ''FOR SHORT RANGE MISSIONS
8673          ALWAYS
8674          ''COMPUTE INTERPOLATED VALUE FOR ROUND IC CPE BETWEEN IC.RH.RANGE(MIN & MAX)
8675          LET CDIF1=IC.RH.ROUND.CPE(MAX.CPE.INT)-IC.RH.ROUND.CPE(MIN.CPE.INT)
8676          LET RND.INT.CPE=IC.RH.ROUND.CPE(MAX.CPE.INT)-(DIV*CDIF1)
8677       ENDIF
8678       IF DEBUG=TRUE,
8679          PRINT 4 LINES WITH IC.RH.TOTAL.CPE(RNG.HACK) AND IC.RH.ROUND.CPE(RNG.HACK) AND
8680          TOT.INT.CPE AND RND.INT.CPE AND IC.RH.RANGE(MIN.RNG.INT) AND
8681          IC.RH.RANGE(MAX.RNG.INT) AND ICHK AND RNG.HACK THUS
8682          $$FC OLD TOT IC CPE = ********, OLD RND IC CPE = ********, $$FC
8683          $$FC NEW TOT IC CPE = ********, NEW RND IC CPE = ********,   $$FC
8684          $$FC  MIN RNG   = ********,     MAX RNG   = ********,     $$FC
8685          $$FC  ICHK = ***,      RNG.HACK = ****        $$FC
8686       ENDIF
8687       LET RAD.OF.EFFECTS = IC.VOLLEY.RAD( ROUND )
8688    ENDIF
8689    IF CLASS = "ICM"
8690       LET K = 2.
8691    ELSE
8692       LET K = 5.
8693    ALWAYS
8694    LET REP.PREC = 0.573 * RND.INT.CPE * K
8695    LET DEP.PREC = 0.573 * RND.INT.CPE * K
8696    LET REP.TM = SQRT.F((0.573*TOT.INT.CPE)**2 +(0.573*TR.CEP(TARGET))**2)
8697    LET DEP.TM = SQRT.F((0.573*TOT.INT.CPE)**2 +(0.573*TR.CEP(TARGET))**2)
8698    LET LENGTH.TGT = SQRT.F(PI.C) * RAD.OF.TARGET
8699    LET WIDTH.TGT = SQRT.F(PI.C) * RAD.OF.TARGET
8700    IF CLASS = "ICM"
8701       LET ROUND.RELY = REAL.F(IC.RELIABILITY(ROUND)/100.)
8702       LET LENGTH.1RND.PAT = SQRT.F(PI.C) * ((RANGE*16./1000.)
8703       * IC.TB.SLOPE(ROUND,BY.TYPE(FM.BTRY(MISSION)))
8704       + IC.TB.INTERCEPT(ROUND,BY.TYPE(FM.BTRY(MISSION))))
8705       LET WIDTH.1RND.PAT = LENGTH.1RND.PAT
8706       IF TE = PERSONNEL
8707          LET LETHAL.AREA = REAL.F(EPS.LA.PERS(ENVIR,POST,IC.SUBM.INDEX(ROUND))/10.)
8708       ELSE
8709          LET LETHAL.AREA = REAL.F(TES.LA.EQUIP(TE,ENVIR,IC.SUBM.INDEX(ROUND))/10.)
8710       ALWAYS
8711       LET PR.1RND.PAT.DAM = 1. - EXP.F((-IC.N.SUBM(ROUND)
8712       * (ES.RELY(ENVIR,IC.SUBM.INDEX(ROUND))/100.)
8713       * LETHAL.AREA)/(LENGTH.1RND.PAT*WIDTH.1RND.PAT))
8714       LET EX.LETH.AREA = LENGTH.1RND.PAT * WIDTH.1RND.PAT * PR.1RND.PAT.DAM
8715    ELSE
8716       LET ROUND.RELY = REAL.F(FZ.HE.RELY(FUZE,ROUND))/100.
8717       IF TE = PERSONNEL
8718          LET LETHAL.AREA = REAL.F(REPF.LA.PERS(RNG.HACK,ENVIR,POST,FUZE))
8719       ELSE
8720          LET LETHAL.AREA = REAL.F(RTEF.LA.EQUIP(RNG.HACK,TE,ENVIR,FUZE))/10.
```

```
8721        ALWAYS
8722        LET LENGTH.1RND.PAT = 2.0 * SQRT.F(LETHAL.AREA*0.307/PI.C)
8723        LET WIDTH.1RND.PAT = LENGTH.1RND.PAT/0.307
8724        LET EX.LETH.AREA = LETHAL.AREA
8725     ALWAYS
8726
8727     IF CLASS NE "SDM"
8728
8729        IF EX.LETH.AREA = 0.0
8730           LET AX = 0.0
8731           LET FRAC.COVERAGE = 0.0
8732 <----------RETURN
8733        OTHERWISE
8734        LET LENGTH.ADJ.PAT = LENGTH.1RND.PAT + REP.PREC
8735        LET WIDTH.ADJ.PAT = WIDTH.1RND.PAT + DEP.PREC
8736        LET AREA.ADJ.PAT = LENGTH.ADJ.PAT * WIDTH.ADJ.PAT
8737        LET LENGTH.VOL = SQRT.F(PI.C) * RAD.OF.EFFECTS
8738        LET WIDTH.VOL = SQRT.F(PI.C) * RAD.OF.EFFECTS
8739        LET LENGTH.DAM = LENGTH.VOL + LENGTH.ADJ.PAT
8740        LET WIDTH.DAM = WIDTH.VOL + WIDTH.ADJ.PAT
8741        LET AREA.DAM = LENGTH.DAM * WIDTH.DAM
8742
8743     ELSE ''SADARM MISSION
8744
8745        LET LENGTH.DAM = SQRT.F(PI.C) * RAD.OF.EFFECTS
8746        LET WIDTH.DAM = SQRT.F(PI.C) * RAD.OF.EFFECTS
8747        LET AREA.DAM = LENGTH.DAM * WIDTH.DAM
8748
8749     ALWAYS
8750     ''
8751     ''
8752     ''THE NEXT BLOCK OF CODE COMPUTES THE FRACTION OF OVERLAP OF THE
8753     ''DAMAGE AREA WITH THE TARGET AREA
8754     ''
8755     LET X.DAM = REAL.F(TR.EST.X(TARGET))
8756     LET Y.DAM = REAL.F(TR.EST.Y(TARGET))
8757     LET X.TGT = REAL.F(UN.X.COORD(TR.TGT.UNIT(TARGET)))
8758     LET Y.TGT = REAL.F(UN.Y.COORD(TR.TGT.UNIT(TARGET)))
8759     IF (ABS.F(X.DAM-X.TGT) > .5 * (LENGTH.DAM + LENGTH.TGT))
8760        OR (ABS.F(Y.DAM-Y.TGT) > .5 * (WIDTH.DAM + WIDTH.TGT))
8761        ''NO COVERAGE
8762        LET AX = 0.0
8763        LET FRAC.COVERAGE = 0.0
8764 <------RETURN
8765     OTHERWISE
8766     RESERVE OVERLAP.COORD(*) AS 4
8767     LET OVERLAP.COORD(1) = Y.DAM - (.5 * WIDTH.DAM)
8768     LET OVERLAP.COORD(2) = Y.DAM + (.5 * WIDTH.DAM)
8769     LET OVERLAP.COORD(3) = Y.TGT - (.5 * WIDTH.TGT)
8770     LET OVERLAP.COORD(4) = Y.TGT + (.5 * WIDTH.TGT)
8771     FOR I = 1 TO 3,
8772     DO ''                                                    \1
8773        FOR J = 1 TO 4
8774        COMPUTE ICHK AS MIN(J) OF OVERLAP.COORD(J)
8775        IF I = 1
8776           LET OVERLAP.COORD(ICHK) = RINF.C
8777        ELSE IF I = 2
8778           LET OVERLAP.WIDTH = -OVERLAP.COORD(ICHK)
```

```
8779          LET OVERLAP.COORD(ICHK) = RINF.C
8780       ELSE IF I = 3
8781          ADD OVERLAP.COORD(ICHK) TO OVERLAP.WIDTH
8782       ALWAYS ALWAYS ALWAYS
8783    ENDLOOP ''                                                    REPEAT CHANGED TO ENDLOOP \1
8784    LET OVERLAP.COORD(1) = X.DAM - (.5 * LENGTH.DAM)
8785    LET OVERLAP.COORD(2) = X.DAM + (.5 * LENGTH.DAM)
8786    LET OVERLAP.COORD(3) = X.TGT - (.5 * LENGTH.TGT)
8787    LET OVERLAP.COORD(4) = X.TGT + (.5 * LENGTH.TGT)
8788    FOR I = 1 TO 3
8789    DO  ''                                                                                \1
8790       FOR J = 1 TO 4
8791       COMPUTE ICHK AS MIN(J) OF OVERLAP.COORD(J)
8792       IF I = 1
8793          LET OVERLAP.COORD(ICHK) = RINF.C
8794       ELSE IF I = 2
8795          LET OVERLAP.LENGTH = -OVERLAP.COORD(ICHK)
8796          LET OVERLAP.COORD(ICHK) = RINF.C
8797       ELSE IF I = 3
8798          ADD OVERLAP.COORD(ICHK) TO OVERLAP.LENGTH
8799       ALWAYS ALWAYS ALWAYS
8800    ENDLOOP  ''                                                   REPEAT CHANGED TO ENDLOOP \1
8801    LET FRAC.COVERAGE = (OVERLAP.LENGTH * OVERLAP.WIDTH)/(LENGTH.TGT
8802    * WIDTH.TGT)
8803    RELEASE OVERLAP.COORD
8804    IF CLASS = "SDM"
8805       LET AX = 0.
8806 <------RETURN
8807    OTHERWISE
8808    LET OVERLAP = N.BY.HOW.SET(FM.BTRY(MISSION)) *
8809    TB.RND.PER.LAUNCH(BY.TYPE(FM.BTRY(MISSION))) *
8810    AREA.ADJ.PAT/AREA.DAM
8811    IF OVERLAP > 1.0
8812       LET OVERLAP = 1.0
8813    ALWAYS
8814    IF OVERLAP = 0.
8815       LET AX = 0.
8816    ELSE
8817       LET AX = OVERLAP * LOG.E.F(1.0 - (EX.LETH.AREA
8818       * N.BY.HOW.SET(FM.BTRY(MISSION))
8819       * TB.RND.PER.LAUNCH(BY.TYPE(FM.BTRY(MISSION)))
8820       * ROUND.RELY)/(AREA.DAM*OVERLAP))
8821    ALWAYS
8822    IF DEBUG = TRUE
8823       PRINT 4 LINES WITH OVERLAP,EX.LETH.AREA, N.BY.HOW.SET(FM.BTRY(MISSION)),
8824       ROUND.RELY, AREA.DAM, AX ,(1.-EXP.F(AX)),LENGTH.1RND.PAT,WIDTH.1RND.PAT,
8825       PR.1RND.PAT.DAM,CLASS,LENGTH.TGT,WIDTH.TGT,TE THUS
8826       ===FINAL COV=== OF = *.**** XLA = ****.**** N.HOW = ** RELY = **.*****
8827       AREA.DAM = *****.****  AX = ****.*** PK = **.*****
8828       L.1RND *****.*****  W.1RND *****.***** PR.1RND **.******
8829       CLASS *****  L.TGT ******.***  W.TGT ******.***  TE *****
8830    ALWAYS
8831    LET FL = 0.0
8832    LET A1 = (LENGTH.DAM + LENGTH.TGT)/(2.96 * REP.TM)
8833    LET A2 = ABS.F(LENGTH.DAM - LENGTH.TGT)/(2.96 * REP.TM)
8834    CALL FRAC.COMPUTE GIVEN A1 YIELDING FRAC------------------------------------------------------------------------------>(335)
8835    ADD FRAC TO FL
8836    CALL FRAC.COMPUTE GIVEN A2 YIELDING FRAC------------------------------------------------------------------------------>(335)
```

```
8837    SUBTRACT FRAC FROM FL
8838    LET ECR = 2.96 * REP.TM * FL/LENGTH.TGT
8839    IF ECR > 1.0 ''DAMAGE LENGTH CONTAINS TGT LENGTH
8840       LET ECR = 1.0
8841    ALWAYS
8842    LET B1 = (WIDTH.DAM + WIDTH.TGT)/(2.96 *DEP.TM)
8843    LET B2 = ABS.F(WIDTH.DAM - WIDTH.TGT)/(2.96 *DEP.TM)
8844    CALL FRAC.COMPUTE GIVEN B1 YIELDING FRAC------------------------------------------------------------------------------------>(335)
8845    ADD FRAC TO FW
8846    CALL FRAC.COMPUTE GIVEN B2 YIELDING FRAC------------------------------------------------------------------------------------>(335)
8847    SUBTRACT FRAC FROM FW
8848    LET ECD = 2.96 * DEP.TM * FW/WIDTH.TGT
8849    IF ECD > 1.0 ''DAMAGE WIDTH CONTAINS TGT WIDTH
8850       LET ECD = 1.0
8851    ALWAYS
8852    LET FRAC.COVERAGE = ECR * ECD * FRAC.COVERAGE
8853    IF FRAC.COVERAGE > 1.0
8854       ''PRINT 7 LINES WITH OVERLAP,EX.LETH.AREA, N.BY.HOW.SET(FM.BTRY(MISSION)),                                          CHG\10
8855       ''ROUND.RELY, AREA.DAM, AX ,(1.-EXP.F(AX)),LENGTH.1RND.PAT,WIDTH.1RND.PAT,
8856       ''PR.1RND.PAT.DAM,CLASS,LENGTH.TGT,WIDTH.TGT,TE,REP.TM,A1,A2,B1,B2,
8857       ''FL,FW,ECR,ECD,WIDTH.DAM,LENGTH.DAM THUS
8858       ''===FINAL COV=== OF = *.**** XLA = ****.**** N.HOW = ** RELY = **.*****
8859       ''AREA.DAM = *****.****  AX = ****.*** PK = **.*****
8860       ''L.1RND *****.*****  W.1RND *****.***** PR.1RND **.******
8861       ''CLASS *****  L.TGT ******.***  W.TGT ******.***  TE ******
8862       ''REP.TM ****.*** A1 **.**** A2 ***.**** B1 ***.**** B2 ***.****
8863       ''FL ****.**** FW ****.**** ECR ****.*** ECD ****.****
8864       ''WIDTH.DAM *****.*****  LENGTH.DAM *****.*****
8865       ''PRINT 1 LINE WITH FRAC.COVERAGE THUS
8866       ''=== ERROR === FRAC.COVERAGE = *********.****
8867       ''TRACE
8868 <-----''STOP                                                                    \EFFECT_OF_ZERO_SUB_CORRECTION?
8869       LET FRAC.COVERAGE = 1.0 ''                                                    CHG\10   \TO_LET_CONTINUE
8870    ALWAYS ''                                                  TO COMPILE, HAD TO REPLACE OTHERWISE WITH ALWAYS
8871
8872
8873    IF CLASS = "HE"  AND
8874       TR.TGT.UNIT(TARGET) IS IN A FR.UNIT.SET
8875       ''CHECK THE EFFECT ON VISIBILITY OF THE
8876       ''DUST THROWN UP BY THE ROUNDS
8877       LET .SIGMA = TOT.INT.CPE / .675
8878       ''UTILIZE NORMAL.F FUNCTION IN NEXT DRAWS ------------------------------------------------------------------------ \1>(641)
8879       LET .XIMPACT = NORMAL.F(X.TGT, .SIGMA, RN.SEED)
8880       LET .YIMPACT = NORMAL.F(Y.TGT, .SIGMA, RN.SEED)
8881       CALL DUST.EFFECTS-----------------------------------------------------------------------------------------
```

```
8891 ROUTINE FIND.START.TIME                                        ''          IF19
     GIVEN
8893 EARLIEST,
8894 LATEST,
8895 DURATION,
8896 FDC
8897 YIELDING START.TIME
8898 ADD 1 TO ANAL.CTR(70,1)  ''                                                \DYN_ANAL
8899 NORMALLY MODE IS INTEGER
8900 DEFINE START.TIME AS A REAL VARIABLE
8901 '' IF DEBUG = TRUE,
8902    '' PRINT 1 LINE WITH FDC THUS
8903    '' = = = FIND.START.TIME  FDC = **** = = =
8904 '' ENDIF
8905 IF FS.START( F.FD.SCHD.LIST( FDC ) )  >  (EARLIEST + DURATION)
8906    LET START.TIME = REAL.F(EARLIEST)/100.          ''FIRE TOT ASAP
8907 <------EXITROUTINE
8908 OTHERWISE
8909 LOOP FOR EACH ITEM IN FD.SCHD.LIST( FDC ),
8910 DO THIS ''                                                              \1
8911    IF FS.STOP( ITEM ) <= INT.F( TIME.V * 100. )
8912       REMOVE THIS ITEM FROM FD.SCHD.LIST( FDC )
8913       DESTROY THE FD.SCHD.MSN CALLED ITEM
8914 <------CYCLE
8915    OTHERWISE
8916    IF FS.START( ITEM + 1 ) <= EARLIEST
8917 <------CYCLE
8918    OTHERWISE
8919    IF FS.STOP( ITEM ) < EARLIEST
8920       LET GAP = EARLIEST - FS.START( ITEM - 1 )
8921    ELSE
8922       LET GAP = FS.STOP( ITEM ) - FS.START( ITEM + 1 )
8923    ENDIF
8924    IF FS.START( ITEM + 1 ) > LATEST
8925       LET GAP = GAP -  ( FS.START( ITEM + 1 ) - LATEST )
8926 <---------EXITROUTINE
8927    OTHERWISE
8928    LET START.TIME = REAL.F(FS.STOP( ITEM ))/100.
8929 <------EXITROUTINE
8930 ENDLOOP
8931 ''NO OPEN PERIODS AVAILABLE
8932 LOOP FOR EACH ITEM IN FD.SCHD.LIST( FDC )
8933 WHEN FS.START( ITEM ) > EARLIEST,
8934 DO THIS
8935    IF FS.START( ITEM + 1 ) > LATEST
8936 <------CYCLE
8937    OTHERWISE
8938    LET N = 0
8939    LET USAGE = 0
8940    LET START = FS.START( ITEM )
8941    LOOP UNTIL TOTAL.TIME >= DURATION,
8942    DO THIS ''                                                           \1
8943       LET TOTAL.TIME = FS.STOP( ITEM + N ) - START
8944       LET DELTA.TIME = FS.STOP( ITEM + N ) - FS.START( ITEM + N )
8945       IF TOTAL.TIME >=  DURATION
8946          LET USAGE = USAGE + ( DELTA.TIME - TOTAL.TIME
8947          + DURATION ) * FS.BATS( ITEM + N )
8948    <------EXITLOOP
```

```
8949          OTHERWISE
8950          LET USAGE = USAGE + DELTA.TIME * FS.BATS(ITEM + N)
8951          LET TOTAL.TIME = FS.START( ITEM + N + 1 ) - START
8952          LET N = N + 1
8953       ENDLOOP
8954       IF USAGE < BEST.USAGE
8955          LET BEST.TIME = START
8956          LET BEST.USAGE = USAGE
8957       ENDIF
8958       LET N = 0
8959       LET TOTAL.TIME = 0
8960       LET START = FS.STOP( ITEM )
8961       LET USAGE = 0
8962       LOOP UNTIL TOTAL.TIME >= DURATION,
8963       DO THIS ''                                                    \1
8964          LET TOTAL.TIME = FS.START( ITEM + N + 1 ) - START
8965          IF TOTAL.TIME >= DURATION
8966       <-----EXITLOOP
8967          OTHERWISE
8968          LET DELTA.TIME = FS.STOP( ITEM + N + 1 )
8969          - FS.START( ITEM + N + 1 )
8970          LET TOTAL.TIME = FS.STOP( ITEM + N + 1 )
8971          - START
8972          IF TOTAL.TIME >= DURATION
8973             LET USAGE = USAGE + ( DELTA.TIME
8974             - TOTAL.TIME + DURATION )
8975             * FS.BATS( ITEM + N + 1 )
8976          ELSE
8977             LET USAGE = USAGE + DELTA.TIME
8978             * FS.BATS( ITEM + N + 1 )
8979             LET N = N + 1
8980          ENDIF
8981       ENDLOOP
8982       IF USAGE < BEST.USAGE
8983          LET BEST.TIME = START
8984          LET BEST.USAGE = USAGE
8985       ENDIF
8986    ENDLOOP
8987    LET START.TIME = REAL.F(BEST.TIME)/100.
8988 ENDROUTINE
```

```
8989 ROUTINE HE.OR.ICM.COMPUTATION                                                    ''                                                    IF20
8990     GIVEN
8991     BAT.NUM,
8992     FM
8993     YIELDING
8994     EST.MSN.EFFECTS
8995
8996     ADD 1 TO ANAL.CTR(71,1)  ''                                                                                                  \DYN_ANAL
8997     ''DETERMINE BEST ICM AND BEST HE ROUND TYPES
8998     ''EXCEPT FOR TOT WHEN ONLY HE IS USED ON 1ST VOLLEY
8999
9000     NORMALLY MODE IS INTEGER
9001     DEFINE BEST.WLA.HE, BEST.WLA.ICM, ICM.W.VOLS, HE.W.VOLS, HE.FRACT.COV,
9002     ICM.FRACT.COV, EST.FRACT.COV, BEST.WLA, AND EST.MSN.EFFECTS
9003     AS REAL VARIABLES
9004     DEFINE BTRY AS AN INTEGER VARIABLE
9005     DEFINE TOT.FOLLOW TO MEAN 2
9006
9007     LET BTRY = FM.BTRY(FM)
9008     LET TARGET = FM.TGT(FM)
9009     LET RANGE = FM.RANGE(FM)
9010     LET TU=TR.EST.TU(TARGET)
9011     LET CAT=TU.CAT(TU)
9012     ''UTILIZE FEBA.BAND FUNCTION IN NEXT ASSIGMENT ——————————————————————————————————————————————————————————————————————————————— \1>(634)
9013     LET DFFB = FEBA.BAND(TARGET)
9014     IF DFFB = 33
9015        LET EST.MSN.EFFECTS = 0
9016        LET FM.N.VOLS(FM) = 0
9017 <———EXITROUTINE
9018     OTHERWISE
9019     LET TB= BY.TYPE(BTRY)
9020
9021     IF DEBUG=TRUE,
9022        PRINT 2 LINES WITH BAT.NUM, FM, CAT,DFFB,TB    THUS
9023        = = =HE.OR.ICM.COMPUTATION   BAT=***, FM=******  = = =
9024        = = =                  CAT=***, DFFB=****** , TB =   **  ═══
9025     ALWAYS
9026
9027     LOOP FOR EACH TML IN TB.TM.LIST(BY.TYPE(BTRY))
9028     DO
9029        IF TB.TM.RAP(TML)=FM.RAP.FLAG(FM)
9030           AND (TB.TM.CLASS(TML) = "HE" OR
9031           TB.TM.CLASS(TML) = "ICM")
9032           LET ROUND = TB.TM(TML)
9033           IF TB.TM.CLASS(TML)="ICM",
9034              IF TR.TOT.STATUS( TARGET ) = TRUE
9035 <————————CYCLE
9036              OTHERWISE
9037              ''UTILIZE ICM.WLA FUNCTION IN NEXT COMPUTATION ——————————————————————————————————————————————————————————————————————— \1>(638)
9038              COMPUTE BEST.ICM AS THE MAXIMUM(ROUND)
9039              AND BEST.WLA.ICM AS THE MAXIMUM OF
9040              ICM.WLA( TARGET, BTRY, ROUND, RANGE)
9041           ELSE ''ROUND IS HE TYPE
9042              ''UTILIZE HE.WLA FUNCTION IN NEXT COMPUTATION ——————————————————————————————————————————————————————————————————————— \1>(636)
9043              COMPUTE BEST.HE AS THE MAXIMUM(ROUND)
9044              AND BEST.WLA.HE AS THE MAXIMUM OF
9045              HE.WLA( TARGET, BTRY, ROUND, RANGE, BAT.NUM, FM )
9046           ALWAYS
```

```
9047        ALWAYS
9048     ENDLOOP
9049
9050     IF BEST.WLA.HE = BEST.WLA.ICM = 0
9051        LET EST.MSN.EFFECTS = 0
9052  <------EXITROUTINE
9053     OTHERWISE
9054
9055     ''PICK HE OR ICM
9056     IF BEST.WLA.ICM NE 0
9057        LET ICM = ROUND
9058        IF CDI.USAGE.INDICATOR(CAT,DFFB,ICM) NE 0  ''                                                    CHG\13  FIXED IN DATA
9059           ''MUST BE ABLE TO SHOOT ICM
9060           IF TR.TOT.STATUS( TARGET ) NE TRUE   '' NOT TOT VOLLEY
9061              CALL WEIGHTED.VOLLEYS------------------------------------------------------------------------------------------>(248)
9062              GIVEN
9063              BTRY,
9064              BEST.ICM,
9065              "ICM",
9066              BEST.WLA.ICM,
9067              FM
9068              YIELDING
9069              ICM.W.VOLS,
9070              ICM.N.VOLS,
9071              ICM.FRACT.COV
9072           ELSE
9073              LET ICM.W.VOLS = RINF.C    ''DON'T USE ICM ON TOT
9074           ALWAYS
9075        ALWAYS
9076     ELSE
9077        LET ICM.W.VOLS = RINF.C
9078     ALWAYS
9079
9080     IF BEST.WLA.HE NE 0
9081        CALL WEIGHTED.VOLLEYS------------------------------------------------------------------------------------------>(248)
9082        GIVEN
9083        BTRY,
9084        BEST.HE,
9085        "HE",
9086        BEST.WLA.HE,
9087        FM
9088        YIELDING
9089        HE.W.VOLS,
9090        HE.N.VOLS,
9091        HE.FRACT.COV
9092     ELSE
9093        LET HE.W.VOLS = RINF.C
9094     ALWAYS
9095
9096     IF 0<ICM.W.VOLS<=HE.W.VOLS          ''IF ICM CHEAPEST,
9097        AND FM.TM.CLASS(FM) NE "HE"   ''AND CLASS NOT SET TO HE BY
9098        LET FM.TM.CLASS(FM) = "ICM"       ''HE.WLA, THIS IS NOT A TOWN
9099        LET FM.TM(FM) = BEST.ICM          ''TARGET,SO USE ICM
9100        LET TR.MISSION.TYPE(TARGET) = "ICM"
9101        LET EST.FRACT.COV = ICM.FRACT.COV
9102        LET BEST.WLA = BEST.WLA.ICM
9103        LET FM.N.VOLS(FM) = MIN.F(ICM.N.VOLS,CDT.MAX.VOLS(CAT,DFFB,TB))
9104     ELSE ''USE THE HE ROUND
```

```
9105        LET FM.TM.CLASS(FM) = "HE"
9106        LET TR.MISSION.TYPE(TARGET) = "HE"
9107        LET FM.TM(FM) = BEST.HE
9108        LET EST.FRACT.COV = HE.FRACT.COV
9109        LET BEST.WLA = BEST.WLA.HE
9110        IF TR.TOT.STATUS( TARGET ) NE TRUE
9111           LET FM.N.VOLS(FM) = MIN.F(HE.N.VOLS,CDT.MAX.VOLS(CAT,DFFB,TB))
9112        ELSE
9113           LET FM.N.VOLS( FM ) = 1
9114        ALWAYS
9115     ALWAYS
9116
9117     IF TR.TOT.STATUS( TARGET )=TOT.FOLLOW  AND
9118        FM.N.VOLS(FM) = CDT.MAX.VOLS(CAT,DFFB,TB)
9119        LET FM.N.VOLS( FM ) = FM.N.VOLS( FM ) - 1
9120     ALWAYS
9121
9122     PERFORM MARGINAL.EFFECTS.ADJ----------------------------------------------------------------------------------------------------->(203)
9123     GIVEN
9124     FM,
9125     EST.FRACT.COV,
9126     BEST.WLA,
9127     BAT.NUM
9128     YIELDING
9129     EST.MSN.EFFECTS
9130
9131 <--EXITROUTINE
9132 ENDROUTINE
```

```
9133 ROUTINE ILLUM.COMPUTATION                                                    ''                                                IF21
9134     GIVEN
9135     .MSN
9136
9137     ADD 1 TO ANAL.CTR(72,1)  ''                                                                                          \DYN_ANAL
9138     NORMALLY MODE IS INTEGER
9139
9140     ''THIS ROUTINE IS CALLED BY FA BN ASGN TO DETERMINE WHICH
9141     ''ILLUMINATION MUNITION IS TO BE USED
9142
9143     LET .TARGET = FM.TGT(.MSN)
9144     LET .TB = BY.TYPE(FM.BTRY(.MSN))
9145     ''UTILIZE FEBA.BAND FUNCTION IN NEXT ASSIGNMENT ———————————————————————————————————————————————————————————————————— \1>(634)
9146     LET .DFFB = FEBA.BAND(.TARGET)
9147     LET .CAT = TU.CAT(UN.TYPE.UNIT(TR.TGT.UNIT(.TARGET)))
9148
9149     FOR EACH .LINK IN TB.TM.LIST(.TB)
9150     WITH TB.TM.CLASS(.LINK) = "ILLUM" AND
9151     ILLUM.MAX.RANGE(TB.TM(.LINK)) GE FM.RANGE(.MSN)
9152     FIND THE FIRST CASE
9153     IF FOUND
9154        LET .IM = TB.TM(.LINK)
9155        LET FM.TM(.MSN) = .IM
9156        LET FM.TM.CLASS(.MSN)= "ILLUM"
9157        ''LIGHTING AN AREA  - SQUARE OF RADII NEEDED
9158        LET .REQ.VOLS = (TR.EST.RADIUS(.TARGET)/ILLUM.RADIUS(.IM))**2  ''                                                 \OPTIMIZE
9159        * TR.REM.EFFECTS(.TARGET) / 100
9160        IF .REQ.VOLS = 0
9161           LET .REQ.VOLS = 1
9162           LET FM.N.VOLS(.MSN) = 1
9163        ELSE
9164           LET FM.N.VOLS(.MSN) = MIN.F(.REQ.VOLS,
9165           CDT.MAX.VOLS(.CAT, .DFFB, .TB))
9166        ALWAYS
9167        LET TR.REM.EFFECTS(.TARGET) = TR.REM.EFFECTS(.TARGET)
9168        * (1. - (FM.N.VOLS(.MSN) / .REQ.VOLS))
9169     ALWAYS
9170
9171     IF ILLUM.DEBUG = TRUE
9172        PRINT 2 LINES WITH TIME.V, .MSN, .TARGET, FM.BTRY(.MSN),
9173        .TB, .DFFB, .CAT, .IM, FM.N.VOLS(.MSN),
9174        TR.EST.RADIUS(.TARGET), TR.REM.EFFECTS(.TARGET) THUS
9175        ——ILLUM.COMPUT AT **.****HRS, MSN=******, TARGET=******, BTRY=***, TB=***,
9176        DFFB=****, CAT=****, MUNITION=***, VOLS=***, RAD=*****, REM EFF=****
9177     ALWAYS
9178
9179 <—EXITROUTINE
9180 ENDROUTINE
```

```
                                                                                                                    IF22
9181 ROUTINE ILLUM.EFFECTS
9182     GIVEN
9183     .MSN
9184
9185     ''THIS ROUTINE IS CALLED BY FIRE MISSION TO DETERMINE
9186     ''THE EFFECTS OF ILLUMINATION ROUNDS
9187
9188     ADD 1 TO ANAL.CTR(73,1)  ''                                                                    \DYN_ANAL
9189     NORMALLY MODE IS INTEGER
9190
9191     LET .TR = FM.TGT(.MSN)
9192     LET .REQUESTOR = TR.REP.UNIT(.TR)
9193     LET .TARGET = TR.TGT.UNIT(.TR)
9194
9195     IF .REQUESTOR IS NOT IN A FR.UNIT.SET OR NITE.OR.DAY = DAY
9196        ''EITHER THE BATTLE STOPPED OR DAY HAS STARTED
9197 <------EXITROUTINE
9198     OTHERWISE
9199
9200     ''ILLUMINATION CONTINUES UNTIL WITHDRAWAL, DAYLIGHT,
9201     ''OR BATTLE END
9202     IF UN.STATUS(.REQUESTOR) NE WITHDRAWING AND
9203        UN.STATUS(.REQUESTOR) NE STA.TO.WITH
9204        IF ILLUM.DEBUG = TRUE
9205           PRINT 2 LINES WITH
9206           TIME.V,
9207           .MSN,
9208           .REQUESTOR,
9209           ILLUM.RULE(UN.COLOR(.REQUESTOR),
9210           UN.MISSION(.REQUESTOR)),
9211           ILLUM.DURATION(FM.TM(.MSN)) THUS
9212           ----ILLUM.EFFECT AT **.***HRS - REPEAT MSN ****** - REQUESTOR=****,
9213           RULE= **, AFTER *** MIN BURN TIME
9214        ALWAYS
9215        CALL REQUEST.ILLUM------------------------------------------------------------------------------>(218)
9216        GIVEN
9217        .REQUESTOR,
9218        ILLUM.RULE(UN.COLOR(.REQUESTOR),
9219        UN.MISSION(.REQUESTOR)),
9220        ILLUM.DURATION(FM.TM(.MSN))
9221     ALWAYS
9222
9223     FOR EVERY .FORCE IN BTL.FORCE.SET(UN.BATTLE.INDEX(.REQUESTOR))
9224     WITH FR.SIDE(.FORCE) = UN.COLOR(.REQUESTOR)
9225     FIND THE FIRST CASE
9226     IF NONE
9227        CALL ERROR.STOP--------------------------------------------------------------------------------->(604)
9228     ALWAYS
9229     FOR EVERY .ENEMY.FORCE
9230     IN BTL.FORCE.SET(UN.BATTLE.INDEX(.REQUESTOR))
9231     WITH .ENEMY.FORCE NE .FORCE
9232     FIND THE FIRST CASE
9233     IF NONE
9234        CALL ERROR.STOP--------------------------------------------------------------------------------->(604)
9235     ALWAYS
9236
9237     IF UN.MISSION(.REQUESTOR) LE 3 AND .REQUESTOR NE .TARGET
9238        LET .RULE = 1  ''ATTACKER ILLUMINATES ALL DEFENDERS
```

```
9239        LET .LITED = .ENEMY.FORCE
9240        LET .UNLIT = .FORCE
9241        LET .LIT.UNIT = 0
9242     ALWAYS
9243
9244     IF UN.MISSION(.REQUESTOR) LE 3 AND .REQUESTOR = .TARGET
9245        LET .RULE = 2  ''ATTACKER ILLUMINATES SELF
9246        LET .LITED = .FORCE
9247        LET .UNLIT = .ENEMY.FORCE
9248        LET .LIT.UNIT = .REQUESTOR
9249     ALWAYS
9250
9251     IF UN.MISSION(.REQUESTOR) GE 4 AND
9252        .REQUESTOR NE .TARGET
9253        LET .RULE = 3  ''DEFENDER ILLUMINATES CLOSEST ATTACKER
9254        LET .LITED = .ENEMY.FORCE
9255        LET .UNLIT = .FORCE
9256        LET .LIT.UNIT = .TARGET
9257     ALWAYS
9258
9259     IF .UNLIT = 0
9260        TRACE
9261 <------EXITROUTINE
9262     OTHERWISE
9263
9264     LET NITE.OR.DAY = DAY
9265     IF ILLUM.DEBUG = TRUE
9266        PRINT 1 LINE WITH TIME.V, .RULE, .MSN, .TR, .REQUESTOR,
9267        .RULE AS FOLLOWS
9268        ---ILLUM.EFFECT AT **.***HRS,RULE=**,MSN=******,TR=******,TGT=****,REQ=****
9269     ALWAYS
9270     LOOP
9271     FOR EVERY  UNIT IN FR.UNIT.SET(.UNLIT)
9272     DO
9273        LOOP
9274        FOR EVERY .ENEMY IN FR.UNIT.SET(.LITED)
9275        WITH .LIT.UNIT = .ENEMY OR .LIT.UNIT = 0
9276        DO
9277           FOR EVERY .VU IN UN.LOS.LIST(.UNIT)
9278           WITH VU.POINTER(.VU) = .ENEMY AND
9279           VU.STATUS(.VU) = NO
9280           FIND THE FIRST CASE
9281           IF FOUND
9282              CALL TIME.TO.DETECT------------------------------------------------------------------------------------------>(123)
9283              GIVEN
9284              .UNIT,
9285              .ENEMY
9286           ALWAYS
9287        ENDLOOP
9288     ENDLOOP
9289     IF ILLUM.DEBUG = TRUE
9290        PRINT 1 LINE WITH
9291        .MSN,
9292        .TR THUS
9293        ---ILLUM.EFFECT  END - MSN=******, TR=******
9294     ALWAYS
9295
9296     LET NITE.OR.DAY = NITE
```

```
9297
9298 <--EXITROUTINE
9299 ENDROUTINE
```

```
9300 ROUTINE MARGINAL.EFFECTS.ADJ  GIVEN  MISSION,     ''                                    IF23
9301     EST.FRACT.COV,
9302     BEST.WLA,
9303     BAT.NUM
9304     YIELDING  EST.MSN.EFFECTS
9305     ''THIS ROUTINE BACKS OFF THE NUMBER OF ROUNDS THAT ARE TO BE
9306     ''FIRED SUCH THAT NO ADDITIONAL VOLLEY IS FIRED THAT DOES NOT
9307     ''CONTRIBUTED A MINIMUM MARGINAL EFFECT TO THE FIRE MISSION
9308     ADD 1 TO ANAL.CTR(74,1)   ''                                                      \DYN_ANAL
9309     NORMALLY MODE IS INTEGER
9310     DEFINE SIDE AND BTRY AS INTEGER VARIABLES
9311     DEFINE EST.FRACT.COV, BEST.WLA, EST.MSN.EFFECTS, FIRST.EFFECTS, X,
9312     FOLLOW.EFFECTS, MAX.MINUS, FOLLOW.MINUS, MIN.MARG,
9313     MARGINAL.EFFECTS, AND MAX.EFFECTS AS REAL VARIABLES
9314     '' IF DEBUG=TRUE,
9315        '' PRINT 1 LINE WITH EST.FRACT.COV,BEST.WLA,BAT.NUM THUS
9316        '' = = = MARGINAL.EFFECTS.ADJ EST.FRACT.COV=*.***, WLA=***.***, BAT=*** = = =
9317     '' ENDIF
9318     LET TARGET = FM.TGT( MISSION )
9319     LET NVOLS = FM.N.VOLS( MISSION )
9320     LET BTRY = FM.BTRY( MISSION )
9321     LET ROUND = FM.TM( MISSION )
9322     LET RANGE = FM.RANGE( MISSION )
9323     LET TUBES.AVAILABLE = N.BY.HOW.SET( BTRY )*TB.RND.PER.LAUNCH(BY.TYPE(BTRY))
9324     LET EST.MSN.EFFECTS = 0.0
9325     LET SIDE = UN.COLOR(BY.UNIT(BTRY))
9326     IF FM.TM.CLASS( MISSION ) = "ICM"
9327        LET RADIUS = IC.VOLLEY.RAD( ROUND )
9328     ELSE
9329        LET RADIUS = HE.VOLLEY.RAD( ROUND )
9330     ENDIF
9331     IF FM.TM.CLASS(MISSION) = "HE"
9332        LET MIN.MARG = HE.MIN.MARG.EFF(ROUND)/10000
9333     ELSE IF FM.TM.CLASS(MISSION) = "ICM"
9334        LET MIN.MARG = IC.MIN.MARG.EFF(ROUND)/10000
9335     ALWAYS ALWAYS
9336     LET X = BEST.WLA * TUBES.AVAILABLE * EST.FRACT.COV /
9337     ( PI.C * ( RADIUS ** 2 ) )
9338     ''FOR HE MUNITIONS THE FIRST EFFECTS ARE BASED UPON THE
9339     ''ASSUMPTION THAT PERSONNEL IN THE TARGET AREA ARE ALWAYS                          \1
9340     ''UNWARNED WHEN THE FIRST VOLLEY OF THE
9341     ''FIRST FIRE UNIT (FIRST TO DELIVER ON THE TARGET) ARRIVES .
9342     ''THEREAFTER, THE PERSONNEL ARE ASSUMED TO BE WARNED FOR
9343     ''ALL OTHER FIRE MISSION ASSOCIATED WITH THE
9344     ''SAME TARGET REPORT
9345     LET FIRST.EFFECTS = 1.0 - EXP.F( - X )
9346     IF FIRST.EFFECTS < MIN.MARG
9347        ''WHEN THE FIRST VOLLEY DOESN'T MEET MINIMUM MARGINAL                           \1
9348        ''EFFECTIVENESS, THEN DON'T FIRE ANYTHING!
9349        LET FM.N.VOLS( MISSION ) = 0
9350 <------EXITROUTINE
9351     OTHERWISE
9352     IF BAT.NUM = 1  AND  FM.TM.CLASS(MISSION) = "HE"
9353        ''UTILIZE HE.WLA FUNCTION IN NEXT CALCULATION ------------------------------------ \1>(636)
9354        LET BEST.WLA = REAL.F(HE.WLA(TARGET,BTRY,ROUND,RANGE,2,
9355        MISSION))/10.
9356     ENDIF
9357     LET X = ( NVOLS - 1 ) * BEST.WLA * TUBES.AVAILABLE *
```

```
9358     EST.FRACT.COV / ( PI.C * ( RADIUS ** 2 ) )  ''                    \OPTIMIZE
9359     LET FOLLOW.EFFECTS = 1.0 - EXP.F( - X )
9360     LET MAX.EFFECTS = FIRST.EFFECTS + FOLLOW.EFFECTS *
9361     ( 1.0 - FIRST.EFFECTS )
9362     IF NVOLS <= 1
9363        LET FM.N.VOLS(MISSION) = NVOLS
9364     ELSE
9365        LOOP FOR I BACK FROM NVOLS TO 2 BY 1,
9366        DO THIS ''                                                      \1
9367           LET X = ( I - 2 ) * BEST.WLA * TUBES.AVAILABLE *
9368           EST.FRACT.COV / ( PI.C * ( RADIUS ** 2 ))
9369           LET FOLLOW.MINUS = 1.0 - EXP.F( - X )
9370           LET MAX.MINUS = FIRST.EFFECTS + FOLLOW.MINUS *
9371           ( 1.0 - FIRST.EFFECTS )
9372           LET MARGINAL.EFFECTS = MAX.EFFECTS - MAX.MINUS
9373           IF MARGINAL.EFFECTS >= MIN.MARG
9374        <-----EXITLOOP
9375           OTHERWISE
9376           LET MAX.EFFECTS = MAX.MINUS
9377        ENDLOOP
9378        IF MARGINAL.EFFECTS < MIN.MARG
9379           LET I = I - 1
9380        ENDIF
9381        LET FM.N.VOLS( MISSION ) = I
9382     ENDIF
9383     LET EST.MSN.EFFECTS = MAX.EFFECTS
9384  ENDROUTINE
```

```
9385 ROUTINE NOISE.DEGRADE GIVEN PDB YIELDING DET.PROB  ''                                  IF24
9386    ADD 1 TO ANAL.CTR(75,1)  ''                                                        \DYN_ANAL
9387    NORMALLY MODE IS INTEGER
9388    DEFINE VOL AS AN INTEGER VARIABLE
9389    DEFINE DET.PROB, EXP AS REAL VARIABLES
9390    LET LINK = PDB.US.LINK(PDB)
9391    LET RATE = 0
9392    LOOP FOR EACH NOISE IN DF.RATE.LIST
9393    DO THIS ''                                                                          \1
9394       IF (TIME.V - DF.TIME(NOISE)) * MINUTES.V > 1.0
9395          REMOVE NOISE FROM DF.RATE.LIST
9396          DESTROY THE DF.NOISE CALLED NOISE
9397       ELSE
9398          ''UTILIZE ACT.RANGE FUNCTION IN NEXT TEST ------------------------------------ \1>(628)
9399          IF ACT.RANGE(DF.UNIT(NOISE),US.UNIT(LINK)) <= 15000./16.  ''                  \OPTIMIZE
9400             ADD 1 TO RATE
9401          ENDIF
9402       ENDIF
9403    ENDLOOP
9404    LOOP FOR EACH VOL IN IF.RATE.LIST
9405    DO THIS ''                                                                          \1
9406       IF (TIME.V - IF.V.TIME(VOL)) * MINUTES.V > 1.
9407          REMOVE VOL FROM IF.RATE.LIST
9408          DESTROY THE IF.VOLLEY CALLED VOL
9409       ELSE
9410          ''UTILIZE ACT.RANGE FUNCTION IN NEXT TEST ------------------------------------ \1>(628)
9411          IF ACT.RANGE(BY.UNIT(IF.V.BTRY(VOL)),US.UNIT(LINK))
9412             <= 15000./16.  ''                                                          \OPTIMIZE
9413             ADD 1 TO RATE
9414          ENDIF
9415       ENDIF
9416    ENDLOOP
9417    IF RATE < 5
9418       LET DET.PROB = 0.1566
9419    ELSE
9420       IF RATE < 10
9421          LET EXP = 3.0
9422       ELSE
9423          LET EXP = 0.0895 * RATE + 2.105
9424       ALWAYS
9425       LET DET.PROB = 1. - ( 1. - ( 2. / RATE ) ) ** ( 1. / EXP )
9426    ENDIF
9427 ENDROUTINE
```

```
9428 ROUTINE PDB.DETECTION                                                                                    IF25
9429    GIVEN
9430    PDB,
9431    BTRY,
9432    RADAR.TYPE,
9433    RANGE
9434    ADD 1 TO ANAL.CTR(76,1)  ''                                                                          \DYN_ANAL
9435    NORMALLY MODE IS INTEGER
9436    DEFINE DET.PROB, KEY.TIME AS REAL VARIABLES
9437    DEFINE RADAR.TYPE AS A TEXT VARIABLE ''                                                              \TEXT
9438    DEFINE BTRY AS AN INTEGER VARIABLE
9439    LET LINK = PDB.US.LINK(PDB)
9440    LET MODEL = US.MODEL(LINK)
9441    LET ST = US.SENSOR.TYPE(LINK)
9442    FOR EACH RH IN MPDB.RH.LIST(MODEL)
9443    WHEN RANGE <= PDB.RH.RANGE(RH)
9444    FIND THE FIRST CASE
9445    IF NONE
9446 <------EXITROUTINE
9447    ENDIF
9448    IF ST.NAME(ST) = "FL"
9449       LET DET.PROB = MPDB.DET.PROB(NITE.OR.DAY, RH)/100.
9450    ELSE '' ST.NAME(ST) = "SD"
9451       CALL NOISE.DEGRADE GIVING PDB YIELDING DET.PROB------------------------------------------------------>(205)
9452    ENDIF
9453    IF RANDOM.F(1) <= DET.PROB
9454       CREATE A PD.DET.UNIT CALLED THIS.UNIT
9455       LET PD.D.BTRY(THIS.UNIT) = BTRY
9456       LET PD.D.CEP(THIS.UNIT) = MPDB.CIR.ERROR(NITE.OR.DAY,RH)
9457       LET PD.D.PD(THIS.UNIT) = INT.F(DET.PROB * 100.)
9458       LET PD.D.PRIORITY(THIS.UNIT) = TU.MIL.WORTH(UN.TYPE.UNIT(BY.UNIT(BTRY)))
9459       FILE THIS.UNIT IN PDB.OP.Q(PDB)
9460       IF PD.OPERATOR(PDB) = IDLE
9461          SCHEDULE_A PDB.OPERATOR GIVING PDB, THIS.UNIT NOW------------------------------------------------->(387)
9462       ENDIF
9463    ELSE
9464       LOOP FOR EACH KS IN PDB.KEYED.LIST(PDB)
9465       DO THIS
9466          LET SENS.ID = KS.SENSOR.ID(KS)
9467          LET TS = KS.TYPE.SENSOR(KS)
9468          LET SENS.LINK = CFR.US.LINK(SENS.ID)
9469          LET MOD = US.MODEL(SENS.LINK)
9470          IF ST.NAME(TS) = RADAR.TYPE
9471             IF US.STATUS(SENS.LINK) = HOLD
9472                IF TIME.V-CFR.LAST.ON.OR.OFF(SENS.ID)> REAL.F(MCFR.MIN.OFF(MOD))/
9473                   (10.*MINUTES.V)
9474                   LET KEY.TIME = UNIFORM.F(0.0,REAL.F(MPDB.KEY.TIME(MODEL))/10.,1)
9475                   LOOP FOR EACH CFRON IN EV.S(I.CFR.ON)
9476                   DO THIS  ''                                                                           \1
9477                      IF CFR.ON.ID(CFRON) = SENS.ID
9478                         CANCEL_THE CFR.ON CALLED CFRON------------------------------------------------->(360)
9479                         DESTROY THE CFR.ON CALLED CFRON
9480                      ENDIF
9481                   ENDLOOP
9482                   SCHEDULE_A CFR.ON GIVING SENS.ID, SENS.LINK IN KEY.TIME MINUTES---------------------->(360)
9483                ENDIF
9484             ENDIF
9485          ENDIF
```

```
9486        ENDLOOP
9487      ENDIF
9488 ENDROUTINE
```

```
9489 ROUTINE PGM.MSN.ASGN                                                                                    IF26
9490     GIVEN
9491     TARGET,
9492     FDC
9493
9494     ADD 1 TO ANAL.CTR(77,1)  ''                                                                         \DYN_ANAL
9495     NORMALLY MODE IS INTEGER
9496     DEFINE W AS A REAL VARIABLE
9497     DEFINE TAC.FAC AND VIS.FAC AS REAL VARIABLES
9498     DEFINE SHAPE AND SCALE AS REAL VARIABLES
9499     DEFINE PROCESS.TIME AS A REAL VARIABLE
9500     DEFINE PERSONNEL TO MEAN 1
9501
9502     '' CHOOSE A BTRY TO FIRE THE MISSION
9503     IF TR.PGM.STATUS(TARGET) = SADARM AND
9504        (UN.STATUS(TR.TGT.UNIT(TARGET)) EQ ADVANCING
9505        OR UN.STATUS(TR.TGT.UNIT(TARGET)) EQ WITHDRAWING)
9506 <------RETURN  ''DON'T SHOOT SADARM AT A MOVING TARGET
9507     OTHERWISE
9508
9509     LET LOW.MIL.WORTH = 1999
9510     LOOP FOR EACH BN.LINK IN FD.BN.LIST(FDC),
9511     UNTIL LOW.MIL.WORTH = 0
9512     DO
9513        LET BN = FB.BN(BN.LINK)
9514        LOOP FOR EACH BTRY IN BN.BTRY.SET(BN)
9515        WITH BY.PGM.CAP(BTRY) = TR.PGM.STATUS(TARGET) AND
9516        BY.STATUS(BTRY) < 2  ''EITHER OCCUPYING OR IN POSITION
9517        AND N.BY.HOW.SET(BTRY) GE TB.MIN.HOW(BY.TYPE(BTRY))
9518        AND (TIME.V*MINUTES.V -UN.LAST.ARTY.ENG(BY.UNIT(BTRY))) GE
9519        (TB.SUPPRESS.TIME(BY.TYPE(BTRY))) AND
9520        TIME.V*MINUTES.V GT BY.STOP.FASCAM.SUPP(BTRY)
9521        DO THE FOLLOWING
9522           ''FIRE MISSION MUST BE MORE IMPORTANT THAN MINIMUM MILITARY
9523           ''WORTH THRESHOLD ASSOCIATED WITH THE NUMBER OF FIRE MISSIONS
9524           ''ALREADY WAITING TO BE FIRED BY THIS FIRE UNIT.
9525           LET .NUM.IN.QUEUE = MIN.F(5, N.BY.FM.QUEUE(BTRY) +1)
9526           ''UTILIZE EST.RANGE FUNCTION IN NEXT TEST ---------------------------------------------------- \1>(632)
9527           IF TR.MIL.WORTH(TARGET) <
9528              TB.MW.THRESHOLD(BY.TYPE(BTRY), .NUM.IN.QUEUE) OR
9529              EST.RANGE(TARGET,BY.UNIT(BTRY)) GT
9530              TB.MAX.RANGE(BY.TYPE(BTRY))
9531        <------CYCLE
9532           OTHERWISE
9533           LET SIDE = UN.COLOR(BY.UNIT(BTRY))
9534           FOR EACH TE.LINK IN THE TU.TE.LIST(UN.TYPE.UNIT(BY.UNIT(BTRY)))
9535           WITH TU.TE.ID(TE.LINK) = PERSONNEL
9536           FIND THE FIRST CASE
9537           IF NONE
9538              PRINT 1 LINE WITH UN.TYPE.UNIT(BY.UNIT(BTRY)) THUS
9539              * * ERROR * * NO PERSONNEL IN TYPE UNIT *****
9540              TRACE
9541 <------------STOP
9542           OTHERWISE
9543           FOR EACH EQ IN THE UN.EQUIP.LIST(BY.UNIT(BTRY)) WITH
9544           EQ.TE.PTR(UE.ID(EQ)) = PERSONNEL
9545           FIND THE FIRST CASE
9546           IF NONE
```

```
9547              PRINT 1 LINE WITH BY.UNIT(BTRY) THUS
9548              = = ERROR = = NO PERSONNEL IN UNIT ****
9549              TRACE
9550 <------------STOP
9551           OTHERWISE
9552           IF REAL.F(UE.QUANT(EQ)/TU.TE.QUANT(TE.LINK)) LT
9553              REAL.F(ARTY.DECIMATE(SIDE)/100)
9554        <------CYCLE
9555           OTHERWISE
9556           ''UTILIZE EST.RANGE FUNCTION IN NEXT TEST ---------------------------------------------------------- \1>(632)
9557           IF TR.PGM.STATUS(TARGET) = SADARM AND
9558              EST.RANGE(TARGET, BY.UNIT(BTRY))*16  GT SDM.MAX.RANGE
9559        <------CYCLE
9560           OTHERWISE
9561           IF BY.CUR.FM(BTRY) = 0
9562              LET LOW.MIL.WORTH = 0
9563              LET .CHOSEN.BTRY = BTRY
9564        <------LEAVE
9565           OTHERWISE
9566           IF TR.MIL.WORTH(FM.TGT(BY.CUR.FM(BTRY))) LT LOW.MIL.WORTH
9567              LET LOW.MIL.WORTH = TR.MIL.WORTH(FM.TGT(BY.CUR.FM(BTRY)))
9568              LET .CHOSEN.BTRY = BTRY
9569           ALWAYS
9570        ENDLOOP
9571     ENDLOOP
9572
9573     IF .CHOSEN.BTRY = 0
9574        IF DEBUG = TRUE
9575           PRINT 1 LINE WITH TARGET THUS
9576           $#$# NO BTRY AVAILABLE FOR TGT *******
9577        ALWAYS
9578 <------EXITROUTINE
9579     OTHERWISE
9580
9581     LET BTRY = .CHOSEN.BTRY
9582     IF BY.CUR.FM(BTRY) GT 0 AND LOW.MIL.WORTH LT 1999
9583        ''MUST INTERRUPT A BTRY THAT IS FIRING
9584        LET FM.N.VOLS(BY.CUR.FM(BTRY))=FM.FIRED.VOLS(BY.CUR.FM(BTRY))
9585     ALWAYS
9586
9587     IF DEBUG = TRUE
9588        PRINT 1 LINE WITH TARGET, BTRY, BY.STATUS(BTRY), BY.PGM.CAP(BTRY),
9589        N.BY.HOW.SET(BTRY) THUS
9590        $#$# TGT = *******, BTRY = *******, STATUS = *, CAP = *, N.HOW = ***
9591     ALWAYS
9592
9593     CREATE A FIRE.MISSION CALLED FM
9594
9595     IF DEBUG = TRUE
9596        PRINT 1 LINE WITH TARGET, TIME.V, TR.MIL.WORTH(TARGET) THUS
9597        $#$#$ FIRE MSN CREATED FOR TARGET  ******* AT TIME **.******, MIL WOR ****
9598     ALWAYS
9599
9600     LET FM.PRIORITY(FM) = TR.MIL.WORTH(TARGET)
9601     LET FM.BTRY(FM) = BTRY
9602     LET FM.TGT(FM) = TARGET
9603     ''UTILIZE EST.RANGE FUNCTION IN NEXT DRAW ---------------------------------------------------------- \1>(632)
9604     LET FM.RANGE(FM) = EST.RANGE(TARGET, BY.UNIT(BTRY))
```

```
9605     IF TR.PGM.STATUS(TARGET) = TRUE
9606        LET FM.TM.CLASS(FM) = "PGM"
9607     ELSE
9608        LET FM.TM.CLASS(FM) = "SDM"
9609     ALWAYS
9610
9611     LET FM.TM(FM) = 1
9612     IF FM.TM.CLASS(FM) = "PGM"
9613        CALL GET.TERRAIN----------------------------------------------------------------------------------------->(336)
9614        YIELDING
9615        .TERRAIN
9616        LET SHAPE = TT.LOS.SHAPE(.TERRAIN)
9617        LET SCALE = TT.LOS.SCALE(.TERRAIN)
9618        ''UTILIZE WEIBULL.F FUNCTION IN NEXT DRAW ------------------------------------------------------ \1>(642)
9619        LET W =(WEIBULL.F(SHAPE,SCALE,RN.SEED))* TER.W.INC./16.
9620        LET TGTUNITYPE = UN.TYPE.UNIT(TR.TGT.UNIT(TARGET))
9621        LET TAC.FAC = TAC.MOV.FAC
9622        LET VIS.FAC = 1.0
9623        IF NITE.OR.DAY = NITE
9624           LET VIS.FAC = NITE.MOV.FAC
9625        ALWAYS
9626        LET PGM.LOS.TIME = (W/(TU.MOV.RATE(TGTUNITYPE)*VIS.FAC*TAC.FAC))
9627        *3600
9628        IF W>0.0 AND (UN.STATUS(TR.TGT.UNIT(TARGET)) NE ADVANCING  OR
9629           UN.STATUS(TR.TGT.UNIT(TARGET)) NE WITHDRAWING)
9630           LET PGM.LOS.TIME = 99999
9631        ALWAYS
9632     ELSE
9633        LET PGM.LOS.TIME = 99999
9634     ALWAYS
9635
9636     LET PGM.PREP.TIME = RANDI.F(60, 90, RN.SEED)
9637     IF FM.RANGE(FM) GT 14500/16 ''                                                          \OPTIMIZE
9638        LET PGM.TOF.TIME = RANDI.F(62,71,RN.SEED)
9639     ELSE
9640        IF FM.RANGE(FM) GT 8500/16
9641           LET PGM.TOF.TIME = RANDI.F(39, 54, RN.SEED)
9642        ELSE
9643           LET PGM.TOF.TIME = RANDI.F(23, 31, RN.SEED)
9644        ALWAYS
9645     ALWAYS
9646
9647     LET PGM.MISSION.TIME = PGM.TOF.TIME + PGM.PREP.TIME
9648     LET FM.PREP.TIME(FM) = PGM.PREP.TIME
9649     LET FM.TOF.TIME(FM) = PGM.TOF.TIME
9650     IF PGM.LOS.TIME <= PGM.MISSION.TIME
9651        IF DEBUG = TRUE
9652           PRINT 1 LINE WITH TARGET, PGM.LOS.TIME, PGM.MISSION.TIME THUS
9653           $#$#$ TARGET ******* , LOS TIME *******, MISSION TIME ********
9654           PRINT 1 LINE WITH W, TU.MOV.RATE(TGTUNITYPE) THUS
9655           $#$# W = ****.*****,  MOVE RATE = ******
9656        ALWAYS
9657
9658        NOW CLEAN.UP.FIRE.MISSIONS ''------------------------------------------------------ CHG\19  \DEBUG>(274)
9659        GIVEN FM
9660        DESTROY THE FIRE.MISSION CALLED FM
9661 <------RETURN
9662     OTHERWISE
```

```
9663
9664     ''THE FOLLOWING CODE COMPUTES NUMBER OF CLGP'S TO BE FIRED
9665     IF FM.TM.CLASS(FM) = "PGM"
9666        LOOP FOR EACH LINK IN TR.DET.LIST(TARGET)
9667        DO THIS
9668           IF TE.PGM.INDIC(TR.DET.TE(LINK)) = TRUE
9669              ADD TR.DET.QUANT(LINK) TO NUM
9670           ALWAYS
9671        ENDLOOP
9672        LET FM.N.VOLS(FM) = NUM + 1
9673     ELSE  ''COMPUTE THE NUMBER OF SADARM VOLLEYS TO BE FIRED
9674        FOR EACH LINK IN TR.DET.LIST(TARGET)
9675        WITH TE.PGM.INDIC(TR.DET.TE(LINK)) GE TRUE
9676        ADD TR.DET.QUANT(LINK) TO NUM
9677        LET FM.N.VOLS(FM) = ((NUM*SDM.MNV/100.)/(SDM.SSPK*3.
9678        *N.BY.HOW.SET(BTRY))) * 100
9679        IF FM.N.VOLS(FM) LE 0
9680           LET FM.N.VOLS(FM) = 1
9681        ALWAYS
9682     ALWAYS
9683     FILE FM IN TR.FM.LIST(TARGET)
9684
9685     IF DEBUG = TRUE
9686        PRINT 1 LINE WITH TARGET THUS
9687        $#$# TGT *******, FM FILED IN TR.FM.LIST
9688     ALWAYS
9689
9690     ''ACCOUNT FOR FIRE MISSION PROCESSING TIME
9691     LET PROCESS.TIME = (FD.MAX.TIME(TR.FDC(TARGET))/9.) * RANDOM.F(RN.SEED)
9692     ACTIVATE_THE FIRE.MISSION CALLED FM IN PROCESS.TIME/60. MINUTES----------->(480)
9693
9694     IF DEBUG = TRUE
9695        PRINT 1 LINE WITH TARGET AND PROCESS.TIME THUS
9696        $#$# FM ACTIVATED FOR TGT ******* IN *****.*  MIN
9697     ALWAYS
9698
9699 <--EXITROUTINE
9700 ENDROUTINE
```

```
9701 ROUTINE PIR.DETECTION                                                                                    IF27
9702    GIVEN
9703    PIR, UN, FRACT.COVERED, REC.TARGET
9704
9705    ADD 1 TO ANAL.CTR(78,1)  ''                                                                          \DYN_ANAL
9706    NORMALLY MODE IS INTEGER
9707    DEFINE  ELEM.DETECT.PROB, ELEM.ACQUIRE.PROB, FRACT.COVERED  AS A REAL VARIABLE
9708    DEFINE  SIDE, SENSOR.TYPE AS INTEGER VARIABLES
9709    DEFINE UN.PCT.OPEN, UN.PCT.WOOD, UN.PCT.TOWN AS REAL VARIABLES
9710    DEFINE UN.ENVIR.FRACT AS A REAL 1-DIMENSIONAL ARRAY
9711    RESERVE UN.ENVIR.FRACT(*) AS N.ENVIRONMENT   '' ALWAYS 3
9712
9713    '' IF DEBUG = TRUE,
9714       '' PRINT 1 LINE WITH PIR, REC.TARGET THUS
9715       '' = = = PIR.DETECTION  PIR = ****** ,REC.TARGET = ******  = = =
9716    '' ENDIF
9717    LET LINK = PIR.US.LINK(PIR)
9718    LET SENSOR = US.UNIT(LINK)
9719    LET MODEL = US.MODEL(LINK)
9720    LET SENSOR.TYPE = US.SENSOR.TYPE(LINK)
9721    LET SIDE = UN.COLOR( SENSOR )
9722
9723    ''THIS ROUTINE WAS CHANGED %15JAN80_%RGR TO REMOVE REFERENCES
9724    ''TO THE COMPOUND ENTITY FOR UNIT,ENVIRONMENT PERCENTS. INSTEAD
9725    ''THE ROUTINE UNIT.ENVIR IS CALLED TO COMPUTE THE VALUES. REFER
9726    ''TO THAT ROUTINE.
9727
9728    CALL UNIT.ENVIR(UN) YIELDING UN.PCT.OPEN, UN.PCT.WOOD, UN.PCT.TOWN------------------------------------------------------------------------------>(242)
9729    LOOP FOR EACH ENVIRONMENT CALLED ENVIR
9730    DO THIS  ''                                                                                           \1
9731       IF EN.NAME(ENVIR) = "OPEN"
9732          LET UN.ENVIR.FRACT(ENVIR) = UN.PCT.OPEN
9733       ELSEIF EN.NAME(ENVIR) = "WOODS"
9734          LET UN.ENVIR.FRACT(ENVIR) = UN.PCT.WOOD
9735       ELSEIF EN.NAME(ENVIR) = "TOWN"
9736          LET UN.ENVIR.FRACT(ENVIR) = UN.PCT.TOWN
9737       ELSE
9738       ALWAYS ALWAYS ENDIF
9739    ENDLOOP
9740
9741    LOOP FOR EACH EQUIP IN UN.EQUIP.LIST( UN )
9742    WHEN UE.QUANT( EQUIP ) > 0
9743    DO THIS
9744       LET QUANT = UE.QUANT(EQUIP)*FRACT.COVERED
9745       LET EQUIP.TYPE = EQ.TE.PTR( UE.ID( EQUIP ) )
9746       LOOP FOR EACH ENVIRONMENT CALLED ENVIR
9747       DO THIS  ''                                                                                        \1
9748          ADD MPMET.DETECT.PROB(MODEL,MOVEMENT,ENVIR,EQUIP.TYPE)*
9749          UN.ENVIR.FRACT(ENVIR) TO ELEM.DETECT.PROB
9750          ADD MPMET.ACQUIRE.PROB(MODEL,MOVEMENT,ENVIR,EQUIP.TYPE)*
9751          UN.ENVIR.FRACT(ENVIR) TO ELEM.ACQUIRE.PROB
9752       ENDLOOP
9753       IF  ELEM.DETECT.PROB = 0.0  OR  ELEM.ACQUIRE.PROB = 0.0
9754    <------CYCLE
9755       OTHERWISE
9756       LET DETECT.QUANT = 0
9757       LOOP FOR I = 1 TO QUANT
9758       DO THIS ''                                                                                         \1
```

```
9759          IF ELEM.DETECT.PROB <= RANDOM.F(1)
9760             LET DETECT.QUANT = DETECT.QUANT + 1
9761          ENDIF
9762       ENDLOOP
9763       IF DETECT.QUANT = 0
9764    <------CYCLE
9765       OTHERWISE
9766       LET ACQUIRED.QUANT = 0
9767       LOOP FOR I = 1 TO DET.QUANT
9768       DO THIS   ''                                                    \1
9769          IF ELEM.ACQUIRE.PROB <= RANDOM.F(1)
9770             ADD 1 TO ACQUIRED.QUANT
9771          ENDIF
9772       ENDLOOP
9773       IF ACQUIRED.QUANT = 0
9774    <------CYCLE
9775       OTHERWISE
9776       FOR EACH DET.LINK IN PIR.RTD.LIST( REC.TARGET )
9777       WHEN PIR.RTD.TE( DET.LINK ) = EQUIP.TYPE
9778       FIND THE FIRST CASE
9779       IF NONE
9780          CREATE A PIR.RTD.LINK CALLED DET.LINK
9781          FILE DET.LINK IN PIR.RTD.LIST( REC.TARGET )
9782          LET PIR.RTD.TE( DET.LINK ) = EQUIP.TYPE
9783          LET PIR.RTD.ELEM.PROB(DET.LINK)=INT.F(100.* ELEM.DETECT.PROB*ELEM.ACQUIRE.PROB)
9784          LET PIR.RTD.QUANT( DET.LINK ) = 0
9785       ELSE
9786          ADD ACQUIRED.QUANT TO PIR.RTD.QUANT(DET.LINK)
9787       ENDIF
9788    ENDLOOP
9789    RELEASE UN.ENVIR.FRACT(*)    '' %21JAN80_%RGR
9790 ENDROUTINE
```

```
9791 ROUTINE REM.EFFECTS.COMPUTATION                                    ''                    IF28
9792    GIVEN
9793    EST.MSN.EFFECTS,
9794    TARGET
9795
9796    ''THIS ROUTINE COMPUTES CUMULATIVE EFFECTS AND REMAINING
9797    ''EFFECTS ON THIS TARGET REPORT AFTER THE LAST MISSION
9798
9799    ADD 1 TO ANAL.CTR(79,1)  ''                                                    \DYN_ANAL
9800    NORMALLY MODE IS INTEGER
9801    DEFINE TOTAL.EFF.OLD, TOTAL.EFF.NEW, REQ.EFFECTS, EST.MSN.EFFECTS
9802    AS REAL VARIABLES
9803
9804    IF DEBUG=TRUE,
9805       PRINT 1 LINE WITH EST.MSN.EFFECTS, TARGET THUS
9806       = = = REM.EFFECTS.COMPUTATION EST.EFFECTS=***.***, TARGET=****** = = =
9807    ALWAYS
9808
9809    LET TOTAL.EFF.OLD = REAL.F(TR.CUM.EFFECTS( TARGET ) ) / 10000.
9810    LET REQ.EFFECTS = REAL.F(TR.REQ.EFFECTS( TARGET ) ) / 100.
9811    LET TOTAL.EFF.NEW = TOTAL.EFF.OLD + EST.MSN.EFFECTS * (1.-TOTAL.EFF.OLD)
9812    IF TOTAL.EFF.NEW >= REQ.EFFECTS
9813       LET TR.REM.EFFECTS(TARGET) = 0
9814    ELSE
9815       LET TR.REM.EFFECTS(TARGET) = INT.F(10000.*
9816       (REQ.EFFECTS - TOTAL.EFF.NEW) / (1.0 - TOTAL.EFF.NEW))
9817    ALWAYS
9818    LET TR.CUM.EFFECTS( TARGET ) = INT.F( 10000. *  TOTAL.EFF.NEW )
9819
9820    IF DEBUG=TRUE,
9821       PRINT 1 LINE WITH TOTAL.EFF.OLD, EST.MSN.EFFECTS, TOTAL.EFF.NEW
9822       THUS
9823       = = = REM.EFF = PREVIOUS = ***.***  THIS MSN = ***.***  TOTAL = ***.***
9824    ALWAYS
9825
9826 <--EXITROUTINE
9827 ENDROUTINE
```

```
9828 ROUTINE REQUEST.DEF.FASCAM                                                    ''                                                IF29
9829    GIVEN
9830    .UNITA,
9831    .UNITB
9832
9833    ''THIS ROUTINE IS CALLED BY ENGAGEMENT TO SEE IF FASCAM
9834    ''SHOULD BE REQUESTED BY THE DEFENDER
9835
9836    ADD 1 TO ANAL.CTR(80,1)  ''                                                                                          \DYN_ANAL
9837    NORMALLY MODE IS INTEGER
9838
9839    IF MF.SWITCH NE 1
9840 <------EXITROUTINE
9841    OTHERWISE
9842
9843    IF UN.MISSION(.UNITA) GE 4 AND
9844       (UN.STATUS(.UNITB) = MOVING OR
9845       UN.STATUS(.UNITB) = ADVANCING)
9846       LET .DEFENDER = .UNITA
9847       LET .ATTACKER = .UNITB
9848    ALWAYS
9849    IF UN.MISSION(.UNITB) GE 4 AND
9850       (UN.STATUS(.UNITA) = MOVING OR
9851       UN.STATUS(.UNITA) = ADVANCING)
9852       LET .DEFENDER = .UNITB
9853       LET .ATTACKER = .UNITA
9854    ALWAYS
9855    IF .DEFENDER = 0 OR .ATTACKER = 0
9856 <------EXITROUTINE
9857    OTHERWISE
9858
9859    IF MINE.USE.RULE(UN.COLOR(.DEFENDER), UN.MISSION(.DEFENDER))
9860       GT 0 AND
9861       UN.FASCAM.RECVD(.ATTACKER) LT MAX.ATT.FASCAM AND
9862       UN.DELAY(.ATTACKER) LE 0
9863       ''THE ATTACKER HAS NOT RECEIVED THE MAX PERMISSIBLE
9864       ''FASCAM ROUNDS AND IS NOT CURRENTLY AFFECTED BY MINES
9865       CALL RANGE.COMPUTE------------------------------------------------------------------------------------------------------>(342)
9866       GIVEN
9867       .DEFENDER,
9868       .ATTACKER
9869       YIELDING
9870       .RANGE
9871       IF .RANGE GT MIN.FASCAM.RANGE AND
9872          .RANGE LE MAX.FASCAM.RANGE
9873          CALL REQUEST.FASCAM--------------------------------------------------------------------------------------------------->(216)
9874          GIVEN
9875          .DEFENDER,
9876          .ATTACKER,
9877          0,
9878          0,
9879          2
9880       ALWAYS
9881    ALWAYS
9882
9883 <--EXITROUTINE
9884 ENDROUTINE
```

```
9885 ROUTINE REQUEST.FASCAM                                                                                    IF30
9886     GIVEN
9887     .REQUESTOR,
9888     .TARGET,
9889     .XCOR,
9890     .YCOR,
9891     .RULE
9892
9893     ''THIS ROUTINE STARTS A FASCAM FIRE MISSION - IT IS CALLED FROM
9894     ''REQUEST WD FASCAM AND REQUEST DEF FASCAM AGAINST A TARGET OR
9895     ''FROM MINE DELAY TO RESEED A BARRIER MINEFIELD
9896     ''                                                                    CHG\23  SHUT OFF WITH DATA
9897
9898     ADD 1 TO ANAL.CTR(81,1)  ''                                                          \DYN_ANAL
9899     NORMALLY MODE IS INTEGER
9900
9901     IF .XCOR = 0 AND .YCOR = 0
9902        LET .XCOR = UN.X.COORD(.TARGET)
9903        LET .YCOR = UN.Y.COORD(.TARGET)
9904
9905        ''DON'T DUPLICATE MISSIONS AGAINST THE SAME TARGET                                          \1
9906        ''CHECK FOR TARGET REPORTS NOT YET BEGUN
9907        FOR EVERY .TR IN EV.S(I.TARGET.REPORT)
9908        WITH TR.MISSION.TYPE(.TR) = "FASCAM" AND
9909        TR.TGT.UNIT(.TR) = .TARGET
9910        FIND THE FIRST CASE
9911        IF FOUND
9912 <--------EXITROUTINE
9913        OTHERWISE
9914        ''CHECK FOR REPORTS ALREADY HAVING A FIRE MISSION
9915        FOR EVERY .FM IN EV.S(I.FIRE.MISSION)
9916        WITH TR.MISSION.TYPE(FM.TGT(.FM)) = "FASCAM" AND
9917        TR.TGT.UNIT(FM.TGT(.FM)) = .TARGET
9918        FIND THE FIRST CASE
9919        IF FOUND
9920 <--------EXITROUTINE
9921        OTHERWISE
9922     ALWAYS
9923
9924     ''FIND AN FDC
9925     FOR EVERY .LINK IN UN.SENSOR.LIST(.REQUESTOR)
9926     WITH ST.NAME(US.SENSOR.TYPE(.LINK)) NE "AD"
9927     FIND THE FIRST CASE
9928     IF FOUND
9929        LET .FDC = US.FDC(.LINK)
9930     ELSE
9931        FOR EVERY .FORCE IN
9932        BTL.FORCE.SET(UN.BATTLE.INDEX(.REQUESTOR))
9933        WITH FR.SIDE(.FORCE) = UN.COLOR(.REQUESTOR)
9934        FIND THE FIRST CASE
9935        IF NONE
9936           CALL ERROR.STOP------------------------------------------------------------------------------------------>(604)
9937        ALWAYS
9938        FOR EVERY .UNIT IN FR.UNIT.SET(.FORCE)
9939        FOR EVERY .LINK IN UN.SENSOR.LIST(.UNIT)
9940        WITH ST.NAME(US.SENSOR.TYPE(.LINK)) NE "AD"
9941        FIND THE FIRST CASE
9942        IF FOUND
```

```
9943          LET .FDC = US.FDC(.LINK)
9944       ELSE
9945          FOR EVERY UNIT CALLED .UNIT
9946          WITH UN.COLOR(.UNIT) = UN.COLOR(.REQUESTOR)
9947          FOR EVERY .LINK IN UN.SENSOR.LIST(.UNIT)
9948          WITH ST.NAME(US.SENSOR.TYPE(.LINK)) NE "AD"
9949          FIND THE FIRST CASE
9950          IF NONE
9951             CALL ERROR.STOP------------------------------------------------------------------------------------------------------>(604)
9952          ALWAYS
9953          LET .FDC = US.FDC(.LINK)
9954       ALWAYS
9955    ALWAYS
9956
9957    ACTIVATE_A TARGET.REPORT--------------------------------------------------------------------------------------------------->(470)
9958    CALLED .TR NOW ''                                                                                                          \1
9959
9960    LET TR.MISSION.TYPE(.TR) = "FASCAM"
9961    LET TR.FDC(.TR) = .FDC
9962    LET TR.REP.UNIT(.TR) = .REQUESTOR
9963    LET TR.TGT.UNIT(.TR) = .TARGET
9964    LET TR.MIL.WORTH(.TR) = 1999
9965    LET TR.PGM.STATUS(.TR) = FALSE
9966    LET TR.EST.X(.TR) = .XCOR
9967    LET TR.EST.Y(.TR) = .YCOR
9968    LET TR.RECVD.TIME(.TR) = TIME.V
9969    LET TR.ABORT.TIME(.TR) = TIME.V + .5
9970    LET TR.TOT.STATUS(.TR) = 0
9971    ADD 1 TO UN.FASCAM.RECVD(.TARGET)
9972
9973    IF MF.DEBUG = TRUE
9974       PRINT 2 LINES WITH .RULE, .TR, .REQUESTOR, .TARGET THUS
9975       ===REQUEST.FASC  RULE= **, TARGET REPORT= ******,
9976       REQUESTOR= ****, TARGET= ****    = = =
9977    ALWAYS
9978
9979 <--EXITROUTINE
9980 ENDROUTINE
```

```
9981 ROUTINE REQUEST.ILLUM                                                                                    ''                                                                                
```

```
39    IF .RULE = 1
40       ''ALL OF .ENEMY FORCE WILL BE LIT
41       LOOP
42       FOR EVERY .ENEMY IN FR.UNIT.SET(.ENEMY.FORCE)
43       DO
44          ADD TU.RADIUS(UN.TYPE.UNIT(.ENEMY)) TO .RADIUS
45       ENDLOOP
46    ALWAYS
47    IF .RULE = 2
48       LET .RADIUS = TU.RADIUS(UN.TYPE.UNIT(.REQUESTOR))
49    ALWAYS
50    IF .RULE = 3
51       LET .RADIUS = TU.RADIUS(UN.TYPE.UNIT(.TARGET))
52    ALWAYS
53
54    ''FIND AN FDC
55    FOR EVERY .LINK IN UN.SENSOR.LIST(.REQUESTOR)
56    WITH ST.NAME(US.SENSOR.TYPE(.LINK)) NE "AD"
57    FIND THE FIRST CASE
58    IF FOUND
59       LET .FDC = US.FDC(.LINK)
60    ELSE
61       FOR EVERY .FORCE IN
62       BTL.FORCE.SET(UN.BATTLE.INDEX(.REQUESTOR))
63       WITH FR.SIDE(.FORCE) = UN.COLOR(.REQUESTOR)
64       FIND THE FIRST CASE
65       IF NONE
66          CALL ERROR.STOP----------------------------------------------------------------------------------------------------->(604)
67       ALWAYS
68       FOR EVERY .UNIT IN FR.UNIT.SET(.FORCE)
69       FOR EVERY .LINK IN UN.SENSOR.LIST(.UNIT)
70       WITH ST.NAME(US.SENSOR.TYPE(.LINK)) NE "AD"
71       FIND THE FIRST CASE
72       IF FOUND
73          LET .FDC = US.FDC(.LINK)
74       ELSE
75          FOR EVERY UNIT CALLED .UNIT
76          WITH UN.COLOR(.UNIT) = UN.COLOR(.REQUESTOR)
77          FOR EVERY .LINK IN UN.SENSOR.LIST(.UNIT)
78          WITH ST.NAME(US.SENSOR.TYPE(.LINK))  NE "AD" ''                                    CHG\14  \MISSPELLED
79          FIND THE FIRST CASE
80          IF NONE
81             CALL ERROR.STOP----------------------------------------------------------------------------------------------->(604)
82          ALWAYS
83          LET .FDC = US.FDC(.LINK)
84       ALWAYS
85    ALWAYS
86
87    ''CHECK IF THIS DUPLICATES A MISSION IN PROGRESS
88    LET .DUPLICATE = NO
89    IF .DELAY = 0
90       ''CHECK TARGET REPORTS NOT YET BEGUN
91       LOOP
92       FOR EVERY .TR IN EV.S(I.TARGET.REPORT)
93       WITH TR.MISSION.TYPE(.TR) = "ILLUM" AND
94       RSA.A(.TR) = 0 AND
95       UN.BATTLE.INDEX(TR.REP.UNIT(.TR)) =
96       UN.BATTLE.INDEX(.REQUESTOR) AND
```

```
 97       UN.COLOR(TR.REP.UNIT(.TR)) = UN.COLOR(.REQUESTOR)
 98       UNTIL .DUPLICATE = YES
 99       DO
100          IF TR.TGT.UNIT(.TR) = TR.REP.UNIT(.TR)
101             ''LIGHTING SELF
102             IF TR.REP.UNIT(.TR) = .REQUESTOR
103                LET .DUPLICATE = YES
104             ALWAYS
105          ELSE
106             IF TR.TGT.UNIT(.TR) = .TARGET OR .RULE = 1
107                LET .DUPLICATE = YES
108             ALWAYS
109          ALWAYS
110       ENDLOOP
111       ''CHECK TARGET REPORTS WHICH ALREADY HAVE MISSIONS                                                    CHG\18
112       LOOP
113       FOR EVERY .FM IN EV.S(I.FIRE.MISSION)  ''                                          CHG\20 TURNED OFF ILLUM
114       WITH TR.MISSION.TYPE(FM.TGT(.FM)) = "ILLUM" AND
115       UN.BATTLE.INDEX(TR.REP.UNIT(FM.TGT(.FM))) =
116       UN.BATTLE.INDEX(.REQUESTOR) AND
117       UN.COLOR(TR.REP.UNIT(FM.TGT(.FM))) =
118       UN.COLOR(.REQUESTOR)
119       UNTIL .DUPLICATE = YES
120       DO
121          LET .TR = FM.TGT(.FM)
122          IF TR.TGT.UNIT(.TR) = TR.REP.UNIT(.TR)
123             ''LIGHTING SELF
124             IF TR.REP.UNIT(.TR) = .REQUESTOR
125                LET .DUPLICATE = YES
126             ALWAYS
127          ELSE
128             IF TR.TGT.UNIT(.TR) = .TARGET OR .RULE = 1
129                LET .DUPLICATE = YES
130             ALWAYS
131          ALWAYS
132       ENDLOOP
133    ALWAYS
134
135    IF .DUPLICATE = NO
136       ACTIVATE_A TARGET.REPORT------------------------------------------------------------------------------------------------------>(470)
137       CALLED .TR
138       IN .DELAY MINUTES
139
140       LET TR.MISSION.TYPE(.TR) = "ILLUM"
141       LET TR.FDC(.TR) = .FDC
142       LET TR.REP.UNIT(.TR) = .REQUESTOR
143       LET TR.TGT.UNIT(.TR) = .TARGET
144       LET TR.MIL.WORTH(.TR) = 1999
145       LET TR.PGM.STATUS(.TR) = FALSE
146       LET TR.EST.X(.TR) = .XAIM
147       LET TR.EST.Y(.TR) = .YAIM
148       LET TR.EST.RADIUS(.TR)= .RADIUS
149       LET TR.RECVD.TIME(.TR) = TIME.V
150       LET TR.ABORT.TIME(.TR) = TIME.V + .5
151       LET TR.TOT.STATUS(.TR)= 0
152
153       IF ILLUM.DEBUG = TRUE
154          PRINT 2 LINES WITH TIME.V, .RULE, .TR, .REQUESTOR,
```

```
155           .TARGET THUS
156           ===REQUEST.ILLU AT **.***HRS, RULE= **, TARGET REPORT= ******,
157           REQUESTOR= ******, TARGET= ******
158        ALWAYS
159     ELSE
160        IF MF.DEBUG = TRUE
161           PRINT 1 LINE WITH
162           TIME.V,
163           .RULE,
164           .REQUESTOR,
165           .TARGET THUS
166           ===REQUEST.ILLU AT **.***HRS - DUPLICATE - RULE=**,UNIT=****,TGT=****
167        ALWAYS
168     ALWAYS
169
170 <--EXITROUTINE
171 ENDROUTINE
```

```
172 ROUTINE REQUEST.SMOKE                                                                                    IF32
173    GIVEN
174    .REQUESTOR,
175    .RULE,
176    .DELAY
177
178    ''THIS ROUTINE IS CALLED FROM ENGAGEMENT, WITHDRAW, AND
179    ''SMOKE.EFFECTS TO START A SMOKE MISSION
180
181    ADD 1 TO ANAL.CTR(83,1)  ''                                                                          \DYN_ANAL
182    NORMALLY MODE IS INTEGER
183
184    IF SMK.SWITCH = 0 OR .RULE = 0  ''                                                  CHG\22  SHUT OFF WITH DATA
185 <------EXITROUTINE
186    OTHERWISE
187
188    FOR EVERY .FORCE IN BTL.FORCE.SET(UN.BATTLE.INDEX(.REQUESTOR))
189    WITH FR.SIDE(.FORCE) = UN.COLOR(.REQUESTOR)
190    FIND THE FIRST CASE
191    IF NONE
192       CALL ERROR.STOP-------------------------------------------------------------------------------------->(604)
193    ALWAYS
194    FOR EVERY .ENEMY.FORCE IN BTL.FORCE.SET(UN.BATTLE.INDEX(.REQUESTOR))
195    WITH .ENEMY.FORCE NE .FORCE
196    FIND THE FIRST CASE
197    IF NONE
198       CALL ERROR.STOP-------------------------------------------------------------------------------------->(604)
199    ALWAYS
200
201    IF .RULE = 1 OR .RULE = 4
202       ''SMOKE IS PLACED RELATIVE TO SELF
203       LET .XAIM = UN.X.COORD(.REQUESTOR)
204       LET .YAIM = UN.Y.COORD(.REQUESTOR)
205       LET .TARGET = .REQUESTOR
206    ELSE
207       ''SMOKE IS PLACED RELATIVE TO CLOSEST .ENEMY
208       LET .MIN.DIS = INF.C
209       LOOP
210       FOR EVERY .ENEMY IN FR.UNIT.SET(.ENEMY.FORCE)
211       DO
212          CALL RANGE.COMPUTE------------------------------------------------------------------------------->(342)
213          GIVEN
214          .REQUESTOR,
215          .ENEMY
216          YIELDING
217          .RANGE
218          IF .RANGE LT .MIN.DIS
219             LET .MIN.DIS = .RANGE
220             LET .TARGET = .ENEMY
221          ALWAYS
222       ENDLOOP
223       LET .XAIM = UN.X.COORD(.TARGET)
224       LET .YAIM = UN.Y.COORD(.TARGET)
225    ALWAYS
226
227    ''SET AN FDC
228    FOR EVERY .LINK IN UN.SENSOR.LIST(.REQUESTOR)
229    WITH ST.NAME(US.SENSOR.TYPE(.LINK)) NE "AD"
```

```
230     FIND THE FIRST CASE
231     IF FOUND
232        LET .FDC = US.FDC(.LINK)
233     ELSE
234        FOR EVERY .FORCE IN
235        BTL.FORCE.SET(UN.BATTLE.INDEX(.REQUESTOR))
236        WITH FR.SIDE(.FORCE) = UN.COLOR(.REQUESTOR)
237        FIND THE FIRST CASE
238        IF NONE
239           CALL ERROR.STOP------------------------------------------------------------------------------------------>(604)
240        ALWAYS
241        FOR EVERY .UNIT IN FR.UNIT.SET(.FORCE)
242        FOR EVERY .LINK IN UN.SENSOR.LIST(.UNIT)
243        WITH ST.NAME(US.SENSOR.TYPE(.LINK)) NE "AD"
244        FIND THE FIRST CASE
245        IF FOUND
246           LET .FDC = US.FDC(.LINK)
247        ELSE
248           FOR EVERY UNIT CALLED .UNIT
249           WITH UN.COLOR(.UNIT) = UN.COLOR(.REQUESTOR)
250           FOR EVERY .LINK IN UN.SENSOR.LIST(.UNIT)
251           WITH ST.NAME(US.SENSOR.TYPE(.LINK)) NE "AD"
252           FIND THE FIRST CASE
253           IF NONE
254              CALL ERROR.STOP------------------------------------------------------------------------------------->(604)
255           ALWAYS
256           LET .FDC = US.FDC(.LINK)
257        ALWAYS
258     ALWAYS
259
260     IF .RULE = 2 OR .RULE = 4
261        ''ONLY A SINGLE UNIT IS TO BE BLOCKED
262        IF .RULE = 2
263           ''ATTACKERS BLOCK CLOSEST DEFENDER
264           LET .RADIUS = TU.RADIUS(UN.TYPE.UNIT(.TARGET))
265        ELSE
266           ''DEFENDER BLOCKS SELF UPON WITHDRAWAL
267           LET .RADIUS = TU.RADIUS(UN.TYPE.UNIT(.REQUESTOR))
268        ALWAYS
269     ELSE
270        ''CONCEAL THE ENTIRE FORCE
271        IF .RULE = 1
272           ''ATTACKERS BLOCK OWN FORCE
273           LET .CONCEAL = .FORCE
274        ELSE
275           ''DEFENDERS BLOCK ALL BUT CLOSEST ATTACKER
276           LET .CONCEAL = .ENEMY.FORCE
277        ALWAYS
278        LOOP
279        FOR EVERY .UNIT IN FR.UNIT.SET(.CONCEAL)
280        DO
281           ADD TU.RADIUS(UN.TYPE.UNIT(.UNIT)) TO .RADIUS
282        ENDLOOP
283     ALWAYS
284
285     ''CHECK IF THIS DUPLICATES A MISSION ALREADY SCHEDULED
286     LET .DUPLICATE = NO
287     IF .DELAY = 0
```

```
''CHECK TARGET REPORTS NOT YET BEGUN
LOOP
FOR EVERY .TR IN EV.S(I.TARGET.REPORT)
WITH TR.MISSION.TYPE(.TR) = "SMOKE" AND
RSA.A(.TR) = 0 AND
UN.BATTLE.INDEX(TR.REP.UNIT(.TR)) =
UN.BATTLE.INDEX(.REQUESTOR) AND
UN.COLOR(TR.REP.UNIT(.TR)) = UN.COLOR(.REQUESTOR)
UNTIL .DUPLICATE = YES
DO
   IF TR.TGT.UNIT(.TR) = TR.REP.UNIT(.TR)
      IF .RULE = 1
         ''THE ENTIRE FORCE IS ALREADY SCREENED
         LET .DUPLICATE = YES
      ELSE
         ''RULE 4 - SCREENING SELF
         IF TR.REP.UNIT(.TR) = .REQUESTOR
            LET .DUPLICATE = YES
         ALWAYS
      ALWAYS
   ELSE
      IF TR.TGT.UNIT(.TR) = .TARGET
         LET .DUPLICATE = YES
      ELSE
         IF .RULE = 3
            ''ALL BUT ONE OF ENEMY SCREENED
            LET .DUPLICATE = YES
         ALWAYS
      ALWAYS
   ALWAYS
ENDLOOP
''CHECK TARGET REPORTS ALREADY BEGUN                                    CHG\22 PROBLEM LOCATION
LOOP
FOR EVERY .FM IN EV.S(I.FIRE.MISSION)
WITH TR.MISSION.TYPE(FM.TGT(.FM)) = "SMOKE" AND
UN.BATTLE.INDEX(TR.REP.UNIT(FM.TGT(.FM))) =
UN.BATTLE.INDEX(.REQUESTOR) AND
UN.COLOR(TR.REP.UNIT(FM.TGT(.FM))) =
UN.COLOR(.REQUESTOR)
UNTIL .DUPLICATE = YES
DO
   LET .TR = FM.TGT(.FM)
   IF TR.TGT.UNIT(.TR) = TR.REP.UNIT(.TR)
      IF .RULE = 1
         ''THE ENTIRE FORCE IS ALREADY SCREENED
         LET .DUPLICATE = YES
      ELSE
         ''RULE 4 - SCREENING SELF
         IF TR.REP.UNIT(.TR) = .REQUESTOR
            LET .DUPLICATE = YES
         ALWAYS
      ALWAYS
   ELSE
      IF TR.TGT.UNIT(.TR) = .TARGET
         LET .DUPLICATE = YES
      ELSE
         IF .RULE = 3
            ''ALL BUT ONE OF ENEMY SCREENED
```

```
346                  LET .DUPLICATE = YES
347               ALWAYS
348            ALWAYS
349         ALWAYS
350      ENDLOOP
351   ALWAYS
352
353   IF .DUPLICATE = NO
354      ACTIVATE_A TARGET.REPORT------------------------------------------------------------------------------------------>(470)
355      CALLED .TR
356      IN .DELAY MINUTES
357
358      LET TR.MISSION.TYPE(.TR) = "SMOKE"
359      LET TR.FDC(.TR) = .FDC
360      LET TR.REP.UNIT(.TR) = .REQUESTOR
361      LET TR.TGT.UNIT(.TR) = .TARGET
362      LET TR.MIL.WORTH(.TR) = 1999
363      LET TR.PGM.STATUS(.TR) = FALSE
364      LET TR.EST.X(.TR) = .XAIM
365      LET TR.EST.Y(.TR) = .YAIM
366      LET TR.EST.RADIUS(.TR) = .RADIUS
367      LET TR.RECVD.TIME(.TR) = TIME.V
368      LET TR.ABORT.TIME(.TR) = TIME.V + .5
369      LET TR.TOT.STATUS(.TR) = 0
370
371      IF SMK.DEBUG = TRUE
372         PRINT 2 LINES WITH TIME.V, .RULE, .TR, .REQUESTOR,
373         .TARGET,
374         .DELAY THUS
375         ===REQUEST.SMOK AT **.***HRS  RULE= **, TARGET REPORT= ******,
376         REQUESTOR= ****, TARGET= ****, SCHED. IN **** MIN = = =
377      ALWAYS
378   ELSE
379      IF SMK.DEBUG = TRUE
380         PRINT 1 LINE WITH
381         TIME.V,
382         .RULE,
383         .REQUESTOR,
384         .TARGET THUS
385         = = =REQUEST SMOKE, **.***HRS -DUP., IGNORED- RULE=**,UNIT=****,TGT=****
386      ALWAYS
387   ALWAYS
388
389 <--EXITROUTINE
390 ENDROUTINE
```

```
391 ROUTINE REQUEST.WD.FASCAM                                                                    ''                                                                
```

```
449          .MIN.UNIT,
450          0,
451          0,
452          3
453       ALWAYS
454       IF .MIN.UNIT GT 0 AND .MIN.DIS LE MIN.FASCAM.RANGE
455          ''TOO CLOSE FOR ARTY - TRY MOPMS
456          FOR EVERY .EQ IN UN.EQUIP.LIST(.UNIT)
457          WITH EQ.NAME(UE.ID(.EQ)) = "MOPMS" AND
458          UE.QUANT(.EQ) GT 0
459          FIND THE FIRST CASE
460          IF FOUND
461             SUBTRACT 1 FROM UE.QUANT(.EQ)
462             CALL MINE.EFFECTS------------------------------------------------------------------------------------------>(250)
463             GIVEN
464             .MIN.UNIT,
465             UN.X.COORD(.MIN.UNIT),
466             UN.Y.COORD(.MIN.UNIT),
467             2,  ''MOPMS
468             0   ''NO FIRE MISSION
469             FOR EVERY TYPE.WEAPON CALLED .TW
470             WITH TW.NAME(.TW) = "MOPMS"
471             FIND THE FIRST CASE
472             IF FOUND
473                ADD 1 TO STW.RND.FIRED(UN.COLOR(.UNIT), .TW)
474             ALWAYS
475          ALWAYS
476       ALWAYS
477    ALWAYS
478
479    LET .MIN.DIS = INF.C
480    LOOP
481    FOR EVERY .ENEMY IN FR.UNIT.SET(.ENEMY.FORCE)
482    DO
483       CALL RANGE.COMPUTE----------------------------------------------------------------------------------------->(342)
484       GIVEN
485       .UNIT,
486       .ENEMY
487       YIELDING
488       .RANGE
489       IF .RANGE LT .MIN.DIS
490          LET .MIN.DIS = .RANGE
491          LET .MIN.UNIT = .ENEMY
492       ALWAYS
493    ENDLOOP
494    IF MINE.USE.RULE(FR.SIDE(.ENEMY.FORCE), FR.MISSION(.ENEMY.FORCE))
495       NE 0 AND
496       FR.MISSION(.ENEMY.FORCE) LE 3 AND
497       UN.FASCAM.RECVD(.UNIT) LT MAX.WD.FASCAM AND
498       .MIN.DIS GT MIN.FASCAM.RANGE AND
499       .MIN.DIS LE MAX.FASCAM.RANGE
500       ''THE .ENEMY WILL CALL FOR FASCAM TO SLOW WITHDRAWAL
501       CALL REQUEST.FASCAM---------------------------------------------------------------------------------------->(216)
502       GIVEN
503       .MIN.UNIT,
504       .UNIT,
505       0,
506       0,
```

```
507       1
508     ALWAYS
509
510 <--EXITROUTINE
511 ENDROUTINE
```

```
512 ROUTINE RPV.DETECTION                                                                                    ''                                                                                
```

```
570       LOOP FOR I = 1 TO QUANT
571       DO THIS  ''                                                                                    \1
572          IF ELEM.DETECT.PROB <= RANDOM.F(1)
573             LET DETECT.QUANT = DETECT.QUANT + 1
574          ENDIF
575       ENDLOOP
576       IF DETECT.QUANT = 0
577    <------CYCLE
578       OTHERWISE
579       LET ACQUIRED.QUANT = 0
580       LOOP FOR I = 1 TO DET.QUANT
581       DO THIS  ''                                                                                    \1
582          IF ELEM.ACQUIRE.PROB <= RANDOM.F(1)
583             ADD 1 TO ACQUIRED.QUANT
584          ENDIF
585       ENDLOOP
586       IF ACQUIRED.QUANT = 0
587    <------CYCLE
588       OTHERWISE
589       IF TR.TGT.UNIT(TARGET) = 0
590          LET TR.TGT.UNIT(TARGET) = UN
591          LET TR.SENSOR.TYPE(TARGET) =ST.NAME(SENSOR.TYPE)
592          LET TR.SENSOR.ID(TARGET) = RPV
593          LET TR.FDC(TARGET) = US.FDC(LINK)
594          LET TR.REP.UNIT(TARGET) = LAUNCH.SITE
595          LET TR.CEP(TARGET) = MRPV.CIR.ERROR(MODEL)
596          ''UTILIZE NORMAL.F FUNCTION IN NEXT 2 CALCULATIONS ------------------------------------------ \1>(641)
597          LET TR.EST.X( TARGET ) = UN.X.COORD( UN ) +
598          NORMAL.F(0.0,1.0,1) * TR.CEP(TARGET) / (1.1774 * 16.) ''                                       \OPTIMIZE
599          LET TR.EST.Y( TARGET ) = UN.Y.COORD( UN ) +
600          NORMAL.F(0.0,1.0,1) * TR.CEP(TARGET) / (1.1774 * 16.)
601       ENDIF
602       '' WRITE"///200", TIME.V, RPV,SENSOR.TYPE,MODEL, "RPV",RANGE,
603       '' UN, UN.TYPE(UN), TR.FDC(TARGET),TARGET,TR.MOVE(TARGET)
604       '' AS BINARY USING UNIT 7
605       IF MRPV.PGM.CAP(MODEL) =TRUE AND TE.PGM.INDIC(EQUIP.TYPE) =TRUE
606          LET TR.PGM.STATUS(TARGET) = TRUE
607          FOR EACH DET.LINK IN TR.DET.LIST( TARGET )
608          WHEN TR.DET.TE( DET.LINK ) = EQUIP.TYPE
609          FIND THE FIRST CASE
610          IF NONE
611             CREATE A TR.DET.LINK CALLED DET.LINK
612             FILE DET.LINK IN TR.DET.LIST( TARGET )
613             LET TR.DET.TE( DET.LINK ) = EQUIP.TYPE
614             LET TR.DET.ELEM.PROB(DET.LINK)=INT.F(100.* ELEM.DETECT.PROB*ELEM.ACQUIRE.PROB)
615             LET TR.DET.QUANT( DET.LINK ) = 0
616          ELSE
617             ADD ACQUIRED.QUANT TO TR.DET.QUANT(DET.LINK)
618          ENDIF
619       ENDIF
620    ENDLOOP
621    RELEASE UN.ENVIR.FRACT(*)    '' %21JAN80_%RGR
622 ENDROUTINE
```

```
623 ROUTINE SIZE.ESTIMATE                                                    ''                                                    IF35
624    GIVEN TARGET
625    AND TYPE.UN
626    YIELDING SIZE.MEASURE
627
628    ADD 1 TO ANAL.CTR(86,1)  ''                                                                                              \DYN_ANAL
629    ''CALLED BY TARGET.ANALYSIS
630    NORMALLY MODE IS INTEGER
631    DEFINE P, Q, SIZE.MEASURE AS REAL VARIABLES
632    DEFINE COMB.RESULT, SM.LOG.VALUE AS REAL VARIABLES
633    LET .SIDE = TU.SIDE(TYPE.UN)
634    LET SIZE.MEASURE = 1.0
635    LET COUNT = 0
636    LOOP FOR EACH TR.DET.LINK IN TR.DET.LIST( TARGET ),
637    DO THIS  ''                                                                                                              \1
638       FOR EACH TU.TE.LINK IN TU.TE.LIST( TYPE.UN ),
639       WHEN TR.DET.TE( TR.DET.LINK ) = TU.TE.ID( TU.TE.LINK ),
640       FIND THE FIRST CASE,
641       IF NONE
642    <------CYCLE
643       OTHERWISE
644       ADD 1 TO COUNT
645       LET N = TU.TE.QUANT( TU.TE.LINK )
646       LET M = TR.DET.QUANT( TR.DET.LINK )
647       IF M > N
648          OR TU.FREQ(TYPE.UN)  <= 0
649          LET SIZE.MEASURE = 0.0
650 <--------EXITROUTINE
651       OTHERWISE
652       LET DIFF = N - M
653       LET P = TR.DET.ELEM.PROB(TR.DET.LINK)/100.
654       IF P >= 1.0
655          LET P = 0.99
656       ALWAYS
657       IF P <= 0.0
658          LET P = 0.01
659       ALWAYS
660       LET Q = 1.0 - P
661       IF Q <= 0.0
662          TRACE  ''                                                                                                          \1
663          CALL ERROR.STOP------------------------------------------------------------------------------------------------->(604)
664       ALWAYS
665       ''UTILIZE COMBINATIONS FUNCTION IN NEXT ASSIGMENT ------------------------------------------------------------  \1>(631)
666       LET COMB.RESULT = COMBINATIONS(N,M)
667       IF P <= 0.0
668          TRACE  ''                                                                                                          \1
669          CALL ERROR.STOP------------------------------------------------------------------------------------------------->(604)
670       ALWAYS
671       LET SM.LOG.VALUE = LOG.E.F(SIZE.MEASURE)
672       + COMB.RESULT
673       + LOG.E.F(REAL.F(TU.FREQ(TYPE.UN)))
674       - LOG.E.F(REAL.F(SIDE.TU.TOTAL(.SIDE)))
675       + M * LOG.E.F(P)
676       + DIFF * LOG.E.F(Q)
677       LET SIZE.MEASURE = EXP.F(SM.LOG.VALUE)
678       IF SIZE.MEASURE <= 0.0
679          LET SIZE.MEASURE = 0.0
680 <--------EXITROUTINE
```

```
681       OTHERWISE
682    ENDLOOP
683    IF COUNT = 0
684       LET SIZE.MEASURE = 0.0
685    ALWAYS
686 ENDROUTINE
```

```
687 ROUTINE SMOKE COMPUTATION                                                    IF36
688    GIVEN
689    .MSN
690
691    ADD 1 TO ANAL.CTR(87,1)  ''                                               \DYN_ANAL
692    ''THIS ROUTINE IS CALLED FROM FA BN ASGN TO DETERMINE
693    ''WHICH SMOKE MUNITION IS TO BE USED
694
695    NORMALLY MODE IS INTEGER
696
697    LET .TARGET = FM.TGT(.MSN)
698    LET .TB = BY.TYPE(FM.BTRY(.MSN))
699    ''UTILIZE FEBA.BAND FUNCTION IN NEXT ASSIGNMENT ------------------------------ \1>(634)
700    LET .DFFB = FEBA.BAND(.TARGET)
701    IF TR.TGT.UNIT(.TARGET) = TR.REP.UNIT(.TARGET)
702       LET .CAT = TU.CAT(UN.TYPE.UNIT(TR.REP.UNIT(.TARGET)))
703    ELSE
704       LET .CAT = TU.CAT(UN.TYPE.UNIT(TR.TGT.UNIT(.TARGET)))
705    ALWAYS
706
707    FOR EACH .LINK IN TB.TM.LIST(.TB)
708    WITH TB.TM.CLASS(.LINK) = "SMOKE" AND
709    SMK.MAX.RANGE(TB.TM(.LINK)) GE FM.RANGE(.MSN)
710    FIND THE FIRST CASE
711    IF FOUND
712       LET .SMOKE = TB.TM(.LINK)
713       LET FM.TM(.MSN) = .SMOKE
714       LET FM.TM.CLASS(.MSN) = "SMOKE"
715       LET .REQ.VOLS = (2*TR.EST.RADIUS(.TARGET) / SMK.WIDTH(.SMOKE))
716       * TR.REM.EFFECTS(.TARGET) / 100
717       IF .REQ.VOLS = 0
718          LET .REQ.VOLS = 1
719          LET FM.N.VOLS(.MSN) = 1
720       ELSE
721          LET FM.N.VOLS(.MSN) = MIN.F(.REQ.VOLS,
722          CDT.MAX.VOLS(.CAT, .DFFB, .TB))
723       ALWAYS
724       LET TR.REM.EFFECTS(.TARGET) = TR.REM.EFFECTS(.TARGET)
725       * (1. - (FM.N.VOLS(.MSN) / .REQ.VOLS))
726    ALWAYS
727
728    IF SMK.DEBUG = TRUE
729       PRINT 2 LINES WITH TIME.V, .MSN, .TARGET, FM.BTRY(.MSN),
730       .TB, .DFFB, .CAT, .SMOKE, FM.N.VOLS(.MSN),
731       TR.EST.RADIUS(.TARGET), TR.REM.EFFECTS(.TARGET) THUS
732       ===SMOKE.COMPUT AT **.****HRS, MSN=******, TARGET=******, BTRY=***, TB=***,
733       DFFB=****, CAT=****, MUNITION=***, VOLS=***, RAD=******, REM EFF=****
734    ALWAYS
735
736 <--EXITROUTINE
737 ENDROUTINE
```

```
738 ROUTINE SMOKE.EFFECTS                                                                                    IF37
739     GIVEN
740     .MSN
741
742     ADD 1 TO ANAL.CTR(88,1)  ''                                                                          \DYN_ANAL
743     ''THIS ROUTINE IS CALLED BY FIRE MISSION TO INTERRUPT
744     ''LOS AFTER DELIVERY OF SMOKE
745
746     NORMALLY MODE IS INTEGER
747
748     LET .TR = FM.TGT(.MSN)
749     LET .REQUESTOR = TR.REP.UNIT(.TR)
750     IF .REQUESTOR IS NOT IN FR.UNIT.SET
751        IF SMK.DEBUG = TRUE
752           PRINT 1 LINE WITH
753           TIME.V,
754           .MSN,
755           .TR,
756           .REQUESTOR THUS
757           ===SMOKE.EFFECT AT **.***HRS - BTL OVER - MSN=******,TR=******,UN=****
758        ALWAYS
759
760        ''THE BATTLE HAS ALREADY ENDED
761 <------EXITROUTINE
762     OTHERWISE
763
764     ''SEE IF MORE SMOKE IS DESIRED------------------------------------------------------------------------------------------------------------------>(    )
765     IF UN.STATUS(.REQUESTOR) NE WITHDRAWING AND
766        UN.STATUS(.REQUESTOR) NE STA.TO.WITH
767        ''ONLY RULE 4 - COVER WITHDRAWAL - IS NOT REPEATED
768        IF SMK.DEBUG = TRUE
769           PRINT 2 LINES WITH
770           TIME.V,
771           .MSN,
772           .REQUESTOR,
773           SMK.USE.RULE(UN.COLOR(.REQUESTOR),
774           NITE.OR.DAY, UN.MISSION(.REQUESTOR)),
775           SMK.BURN.TIME(FM.TM(.MSN)) THUS
776           ===SMOKE.EFFECT AT **.***HRS, MSN ****** TO BE REPEATED,REQUESTOR=****,
777           RULE=**, AFTER ****MIN BURN TIME
778        ALWAYS
779        CALL REQUEST.SMOKE------------------------------------------------------------------------------------------------------------------------>(222)
780        GIVEN
781        .REQUESTOR,
782        SMK.USE.RULE(UN.COLOR(.REQUESTOR), NITE.OR.DAY,
783        UN.MISSION(.REQUESTOR)),
784        SMK.BURN.TIME(FM.TM(.MSN))
785     ALWAYS
786
787     ''UPDATE ALL UNIT POSITIONS
788     LOOP
789     FOR EVERY .FORCE IN BTL.FORCE.SET(UN.BATTLE.INDEX(.REQUESTOR))
790     DO
791        LOOP
792        FOR EVERY .UNIT IN FR.UNIT.SET(.FORCE)
793        WITH UN.STATUS(.UNIT) = ADVANCING OR
794        UN.STATUS(.UNIT) = WITHDRAWING OR
795        UN.STATUS(.UNIT) = ADV.TO.WITH
```

```
796         DO
797            FOR EVERY .MOVE OF EV.S(I.MOVE)
798            WITH MV.UNIT(.MOVE) = .UNIT
799            FIND THE FIRST CASE
800            IF NONE
801               CALL ERROR.STOP------------------------------------------------------------------------------------------>(604)
802            ALWAYS
803            CANCEL_THE MOVE CALLED .MOVE-----------------------------------------------------------------------------------------------
```

```
854             ALWAYS
855
856          ENDLOOP
857       ENDLOOP
858    ELSE
859       IF TR.TGT.UNIT(.TR) NE TR.REP.UNIT(.TR)
860          LET .RULE = 2   ''ATTACKERS .COVER CLOSEST DEFENDER
861          LET .COVER = TR.TGT.UNIT(.TR)
862          LET .OTHER.FORCE = .FORCE
863       ELSE
864          LET .RULE = 4   ''DEFENDER COVERS WITHDRAWAL
865          LET .COVER = .REQUESTOR
866          LET .OTHER.FORCE = .ENEMY.FORCE
867       ALWAYS
868       LOOP
869       FOR EVERY .ENEMY IN FR.UNIT.SET(.OTHER.FORCE)
870       DO
871          CALL BLOCK.LOS------------------------------------------------------------------------------------------------------------>( 80)
872          GIVEN
873          .COVER,
874          .ENEMY,
875          .DURATION
876
877          IF SMK.DEBUG = TRUE
878             PRINT 1 LINE WITH TIME.V, .RULE, .MSN, .TR,
879             .UNIT, .ENEMY,
880             .DURATION THUS
881             ===SMOKE.EFFECT AT **.***HRS,RULE=*,MSN******,TR******,UN****,TGT****,***MIN
882          ALWAYS
883
884       ENDLOOP
885    ALWAYS
886
887    ''THIS WILL HAVE CHANGED THE SEGMENT LENGTHS
888    LOOP
889    FOR EVERY .FORCE IN BTL.FORCE.SET(UN.BATTLE.INDEX(.REQUESTOR))
890    DO
891       LOOP
892       FOR EVERY .UNIT IN FR.UNIT.SET(.FORCE)
893       WITH UN.STATUS(.UNIT) = ADVANCING OR
894       UN.STATUS(.UNIT) = WITHDRAWING OR
895       UN.STATUS(.UNIT) = ADV.TO.WITH
896       DO
897          FOR EVERY .MOVE IN EV.S(I.MOVE)
898          WITH MV.UNIT(.MOVE) = .UNIT
899          FIND THE FIRST CASE
900          IF NONE
901             CALL ERROR.STOP------------------------------------------------------------------------------------------------------------>(604)
902          ALWAYS
903          CANCEL_THE MOVE CALLED .MOVE------------------------------------------------------------------------------------------------>(381)
904          DESTROY THE MOVE CALLED .MOVE
905          CALL MIN.MOVE------------------------------------------------------------------------------------------------------------>(106)
906          GIVEN
907          .UNIT
908       ENDLOOP
909    ENDLOOP
910
911 <--EXITROUTINE
```

```
912 ENDROUTINE
```

```
913 ROUTINE SWITCH.FO                                                                                                    ''  IF38
914     GIVEN
915     BATTLE,
916     STATUS
917
918     ADD 1 TO ANAL.CTR(89,1)  ''                                                                                          \DYN_ANAL
919     NORMALLY MODE IS INTEGER
920     DEFINE   START TO MEAN 1
921     DEFINE   STOP TO MEAN 2
922     DEFINE   TARGET.OF.OPPORT TO MEAN 0
923     DEFINE   BATTLE,FORCE,LINK,DET.CAND, AND UNIT AS INTEGER VARIABLES
924
925     LOOP FOR EACH FORCE IN BTL.FORCE.SET(BATTLE)
926     DO
927        LOOP FOR EACH UNIT IN FR.UNIT.SET(FORCE)
928        DO
929           LOOP FOR EACH LINK IN UN.SENSOR.LIST(UNIT)
930           WHEN ST.NAME(US.SENSOR.TYPE(LINK)) ="FO"
931           DO
932              LET FO =US.ID(LINK)
933              IF STATUS = START
934                 IF FO.CURRENT.TR(FO) IS NOT ZERO, ''                                                  CHG\03  \ZERO_SUB
935                    IF FO IS IN AN EV.S AND
936                       TR.PGM.STATUS(FO.CURRENT.TR(FO)) NE TRUE
937                       INTERRUPT FORWARD.OBSERVER CALLED FO------------------------------------------------------>(438)
938                    ALWAYS
939                    LOOP FOR EACH DET.CAND IN FO.CAND.DET.LIST(FO)
940                    DO
941                       REMOVE  DET.CAND FROM FO.CAND.DET.LIST(FO)
942                       DESTROY THE FO.DET.CANDIDATE CALLED DET.CAND
943                    ENDLOOP
944                    LET  US.STATUS(LINK)= BATTLE
945                    IF TR.PGM.STATUS(FO.CURRENT.TR(FO)) NE TRUE
946                       LET  TIME.A(FO) =  -RINF.C
947                       RESUME FORWARD.OBSERVER CALLED FO------------------------------------------------------>(438)
948                    ALWAYS
949                 ELSE ''                                                                               CHG\03  \ZERO_SUB
950                    IF FO IS IN AN EV.S,
951                       INTERRUPT FORWARD.OBSERVER CALLED FO------------------------------------------------------>(438)
952                       LET TIME.A(FO) = -RINF.C
953                       RESUME FORWARD.OBSERVER CALLED FO------------------------------------------------------>(438)
954                    ALWAYS
955                    LOOP FOR EACH DET.CAND IN FO.CAND.DET.LIST(FO)
956                    DO
957                       REMOVE  DET.CAND FROM FO.CAND.DET.LIST(FO)
958                       DESTROY THE FO.DET.CANDIDATE CALLED DET.CAND
959                    ENDLOOP
960                    LET  US.STATUS(LINK)= BATTLE
961                 ALWAYS
962              ELSE
963                 LET  US.STATUS(LINK)= TARGET.OF.OPPORT
964              ALWAYS
965           ENDLOOP
966        ENDLOOP
967     ENDLOOP
968
969 <--EXITROUTINE
970 ENDROUTINE
```

```
971 ROUTINE TARGET.ANALYSIS                                                                                    IF39
972    GIVEN
973    TARGET,
974    MIL.FLAG ''                                                                                  CHG\21  \DEBUG
975
976    ADD 1 TO ANAL.CTR(90,1)  ''                                                                    \DYN_ANAL
977    ''THIS ROUTINE PROVIDES A METHOD OF DETERMINING THE TARGET
978    ''TYPE BASED UPON THE NON-TYPE-EQUIPMENT LIST OF THE TYPE
979    ''UNIT AND STATISTICALLY ESTABLISHES THE TARGET TYPE
980    ''WITHOUT GOING DIRECTLY TO THE TARGETED UNIT'S ATTRIBUTES
981    ''WHICH WOULD PROVIDE PERFECT INFORMATION.
982
983    NORMALLY MODE IS INTEGER
984    DEFINE TYPE.UN AS AN INTEGER VARIABLE
985    DEFINE SIDE AND SECTOR AS INTEGER VARIABLES
986    DEFINE SIZE.MEASURE AS A REAL VARIABLE
987    DEFINE MANUEVER.SUPPORT TO MEAN 2000
988    DEFINE TARGET.ACQUISITION TO MEAN 0
989    DEFINE MIL.FLAG AS AN INTEGER VARIABLE  ''                                                   CHG\21  \DEBUG
990    DEFINE .INCREMENT AS AN INTEGER VARIABLE  ''                                                 CHG\29  \DEBUG
991
992    LET TGT = TR.TGT.UNIT( TARGET )
993    LET SIGN = 1
994
995    IF DEBUG = TRUE,
996       PRINT 1 LINE WITH TARGET, UNIT.NOS(TGT) THUS
997       = = = TARGET.ANALYSIS TARGET = ******(UNIT ******)  = = =
998    ALWAYS
999
1000   CALL LOCATE.SECTOR------------------------------------------------------------------------------------------>(100)
1001   GIVEN
1002   UN.Y.COORD(TGT)
1003   YIELDING
1004   SECTOR
1005   LET SIDE = UN.COLOR(TGT)
1006
1007   ''THE FOLLOWING 10 LINES OF CODE CORRECT THE ZERO FEBA PROBLEM                                CHG\29 \DEBUG
1008   LET .INCREMENT = -1
1009   UNTIL SS.SET(SIDE,SECTOR) IS NOT EMPTY,
1010   DO
1011      ADD .INCREMENT TO SECTOR
1012      IF SECTOR IS ZERO,
1013         LET .INCREMENT = 1
1014         LET SECTOR = 2 ''ALREADY CHECKED 1
1015      ALWAYS
1016   ENDLOOP ''REPEAT UNTIL SET IS NOT EMPTY
1017
1018   IF SIDE = BLUE
1019      LET FEBA =  F.SS.SET( SIDE , SECTOR )
1020      LET ADV = UN.X.COORD(TGT)
1021      LET FEBA.ADV = UN.X.COORD(FEBA)
1022   ELSE
1023      LET FEBA =  L.SS.SET( SIDE , SECTOR )
1024      LET ADV = - UN.X.COORD(TGT)
1025      LET FEBA.ADV = - UN.X.COORD(FEBA)
1026   ALWAYS
1027   IF ADV > FEBA.ADV
1028      LET SIGN = - SIGN
```

```
1029     ALWAYS
1030     ''UTILIZE EST.RANGE FUNCTION IN NEXT ASSIGNMENT ———————————————————————————————————————— \1>(632)
1031     LET RANGE = EST.RANGE(TARGET , FEBA)
1032     LET DIST.FEBA = SIGN * RANGE
1033
1034     LOOP FOR EACH CATEGORY CALLED CAT,
1035     DO
1036        IF DIST.FEBA < CT.MIN.FEBA( CAT ) AND
1037           CT.MIN.FEBA( CAT ) > 0
1038     <———CYCLE
1039        OTHERWISE
1040
1041        LOOP FOR EACH TYPE.UN IN CT.TU.SET(CAT)
1042        WITH TU.SIDE(TYPE.UN) = SIDE  AND
1043        TU.FREQ(TYPE.UN) > 0
1044        DO THIS
1045           LET NOT.THIS.TU = FALSE  '' INITIALIZE
1046           LOOP FOR EACH TDL IN TR.DET.LIST( TARGET ),
1047           DO
1048              LOOP FOR EACH NTE IN TU.NTE.SET(TYPE.UN)
1049              DO
1050                 IF TR.DET.TE( TDL ) = TU.NTE.ID(NTE)
1051                    LET NOT.THIS.TU = TRUE
1052              <———LEAVE
1053                 OTHERWISE
1054              ENDLOOP
1055              IF NOT.THIS.TU = TRUE
1056           <———LEAVE
1057              OTHERWISE
1058           ENDLOOP
1059           IF NOT.THIS.TU = TRUE
1060        <———CYCLE
1061           OTHERWISE
1062
1063           CALL SIZE.ESTIMATE————————————————————————————————————————————————————————————————————>(231)
1064           GIVEN
1065           TARGET,
1066           TYPE.UN
1067           YIELDING
1068           SIZE.MEASURE
1069           COMPUTE MAX.TU AS THE MAX( TYPE.UN ) OF SIZE.MEASURE
1070
1071           ''THIS COMPUTATION ACCUMULATES DATA ON EACH TYPE UNIT AS
1072           ''SIZE.MEASURE IS RETURNED. ONCE THE LOOP-THRU OF ALL
1073           ''CANDIDATE TYPE UNITS IS COMPLETE, THE ONE THAT
1074           ''HAS ACCUMULATED THE LARGEST STATISTICAL VALUES IS
1075           ''SELECTED AS THE TYPE UNIT FOR THE TARGETING
1076           ''PROCESS.
1077
1078        ENDLOOP
1079     ENDLOOP
1080
1081     IF MAX.TU = 0
1082        PRINT 1 LINE WITH UNIT.NOS(TGT), TU.LEVEL(UN.TYPE.UNIT(TGT))
1083        THUS
1084        = = TARGET ANALYSIS > SIZE ESTIMATE COULD NOT IDENTIFY UNIT •••••• (••••••)
1085        LET TR.EST.TU(TARGET) = MAX.TU
1086 <———EXITROUTINE
```

```
1087     OTHERWISE
1088
1089     IF DEBUG = TRUE,
1090        PRINT 1 LINE WITH TARGET,UNIT.NOS(TGT), MAX.TU THUS
1091        = = = TARGET ******(UNIT ******)IS ANALYZED TO BE TYPE UNIT *** = = =
1092     ALWAYS
1093
1094     LET TR.EST.TU( TARGET ) = MAX.TU
1095     LET TR.EST.RADIUS( TARGET ) = TU.RADIUS( MAX.TU )
1096     IF MIL.FLAG = 1  ''                                                                                    CHG\21 \DEBUG
1097 <-------RETURN
1098     OTHERWISE
1099     IF TR.MIL.WORTH( TARGET ) < MANUEVER.SUPPORT - 1
1100        CALL EST.MIL.WORTH-------------------------------------------------------------------------------------------------->(180)
1101        GIVEN
1102        TARGET,
1103        DIST.FEBA
1104        YIELDING
1105        MIL.WORTH
1106        LET TR.MIL.WORTH( TARGET ) = MIL.WORTH
1107     ALWAYS
1108
1109 <--EXITROUTINE
1110 ENDROUTINE
```

```
1111 ROUTINE UNIT.ENVIR                                                          ''                                                IF40
1112    GIVEN
1113    TGT.UNIT
1114    YIELDING
1115    UN.PCT.OPEN,
1116    UN.PCT.WOOD,
1117    UN.PCT.TOWN
1118
1119    ADD 1 TO ANAL.CTR(91,1)  ''                                                                                              \DYN_ANAL
1120    ''THIS FUNCTION REPLACES THE DATA ORIGINALLY INPUT DIRECTLY
1121    ''IN THE P.E.M.INPUT ROUTINE. THIS FUNCTION COMPUTES
1122    ''THE PERCENT OF AN ARTILLERY-TARGETED UNIT THAT IS IN ONE OF
1123    ''THREE ENVIRONMENTS OF OPEN, TOWN, OR WOODS. THE VALUES ARE
1124    ''COMPUTED WHEN REQUIRED . THE VALUES OF A GIVEN UNIT WILL BE
1125    ''UPDATED EACH TIME THE UNIT IS TARGETED. IF THE UNIT IS NEVER
1126    ''TARGETED, THE VALUE IS NOT COMPUTED.
1127    ''THIS ROUTINE IS CALLED FROM BTRY.EFFECTS, PIR.DETECTION, AO.DETECTION,
1128    ''AND RPV.DETECTION. THE VALUES ARE TRANSFERED AS REAL
1129    ''VARIABLES FROM 0.00 TO 1.00 TO THE CALLING ROUTINE.
1130    ''FACTORS CONSIDERED ARE THE UNIT'S MOVEMENT STATUS, THE
1131    ''UNIT'S GROUP, BE IT MANEUVER, ARTILLERY OR HQ/SUPPORT,
1132    ''AND THE TYPE OF TERRAIN THE UNIT IS PRESENTLY LOCATED ON.
1133
1134    NORMALLY  MODE IS INTEGER
1135    DEFINE TER.TYPE AS AN INTEGER VARIABLE
1136    DEFINE UNIT.M.FACTOR,
1137    OPEN.PCT, WOOD.PCT, TOWN.PCT,
1138    UN.PCT.OPEN, UN.PCT.TOWN, UN.PCT.WOOD,
1139    O.ST.DEV, W.ST.DEV, T.ST.DEV AS REAL VARIABLES
1140    DEFINE OPEN TO MEAN 1
1141    DEFINE WOODS TO MEAN 2
1142    DEFINE TOWN TO MEAN 3
1143
1144    IF DEBUG = TRUE
1145       SKIP 1 LINE
1146       PRINT 1 LINE WITH UNIT.NOS(TGT.UNIT) THUS
1147       = = = UNIT ******'S ENVIRONMENTAL FACTORS ARE = = =
1148    ALWAYS
1149
1150    ''THESE LINES ACCOUNT FOR MOVING UNITS
1151    IF UN.STATUS(TGT.UNIT) = 1    '' ADVANCING
1152       OR UN.STATUS(TGT.UNIT) = 2  ''WITHDRAWING
1153       OR UN.STATUS(TGT.UNIT) = 4  ''MOVING
1154       ''UTILIZE NORMAL.F FUNCTION IN NEXT DRAW ——————————————————————————————————————————————————————————————————————————— \1>(641)
1155       LET UNIT.M.FACTOR = NORMAL.F(1.30,0.20,RN.SEED)
1156       IF UNIT.M.FACTOR < 0.0
1157          LET UNIT.M.FACTOR = 0.0
1158       ALWAYS
1159    ELSE  '' UN.STATUS(TGT.UNIT) = 3  STATIONARY
1160       LET UNIT.M.FACTOR = 1.00
1161    ALWAYS
1162
1163    ''THESE LINES ACCOUNT FOR THE TERRAIN THE UNIT IS ON
1164    LET CAT = TU.CAT(UN.TYPE.UNIT(TGT.UNIT))
1165    IF TER.W.INC = 1
1166       LET OPEN.PCT = EC.FRACT(OPEN,CAT)/100.
1167       LET WOOD.PCT = EC.FRACT(WOODS,CAT)/100.
1168       LET TOWN.PCT = EC.FRACT(TOWN,CAT)/100.
```

```
1169    ELSE
1170       CALL GET.TERRAIN------------------------------------------------------------------------------------------>(336)
1171       YIELDING
1172       TER.TYPE
1173       IF TER.TYPE = 3   ''TANKS AVOID
1174          OR TER.TYPE = 5
1175          OR TER.TYPE = 10
1176          OR TER.TYPE = 11
1177          LET OPEN.PCT = 0.17
1178          LET WOOD.PCT = 0.83
1179       ELSE
1180          IF TER.TYPE = 2 ''GOOD TANK TERRAIN
1181             OR TER.TYPE = 4
1182             OR TER.TYPE = 7
1183             OR TER.TYPE = 9
1184             LET OPEN.PCT = 0.83
1185             LET WOOD.PCT = 0.17
1186          ELSE
1187             IF TER.TYPE = 1  ''ACCEPTABLE TANK TERRAIN
1188                LET OPEN.PCT = 0.67
1189                LET WOOD.PCT = 0.33
1190             ELSE
1191                IF TER.TYPE = 6  ''FLAT AND FORRESTED
1192                   OR TER.TYPE = 8
1193                   LET OPEN.PCT = 0.33
1194                   LET WOOD.PCT = 0.67
1195                ELSE   '' TER.TYPE > 11 , SHOULD OTHER TYPES BE USED
1196                   LET OPEN.PCT = 0.50
1197                   LET WOOD.PCT = 0.50
1198                ALWAYS
1199             ALWAYS
1200          ALWAYS
1201       ALWAYS
1202       LET TOWN.PCT = 0.15  '' AN ARBITRARY VALUE
1203    ALWAYS
1204
1205    ''THESE LINES ACCOUNT FOR THE UNIT GROUP
1206    LET GROUPING = CT.GROUP(CAT) '' 1 = MANEU, 2 = ARTY, 3 = HQ
1207    LET O.ST.DEV = 0.10 * OPEN.PCT  '' COMPUTE A DEVIATION
1208    LET W.ST.DEV = 0.10 * WOOD.PCT
1209    LET T.ST.DEV = 0.10 * TOWN.PCT
1210
1211    ''THESE LINES COMPUTE RANDOM VALUES FOR THE ENVIRONMENT
1212    IF GROUPING = MANEUVER '' 1, INFANTRY OR ARMOR
1213       ''UTILIZE NORMAL.F FUNCTION IN NEXT DRAW ------------------------------------------------------------ \1>(641)
1214       LET UN.PCT.OPEN = NORMAL.F(OPEN.PCT,O.ST.DEV,RN.SEED) *
1215       UNIT.M.FACTOR
1216       IF UN.PCT.OPEN < 1.00
1217          IF UN.PCT.OPEN < 0.0
1218             LET UN.PCT.OPEN = 0.0
1219          ALWAYS
1220          LET UN.PCT.WOOD = UNIFORM.F(0.00,1.0-UN.PCT.OPEN,RN.SEED)
1221          LET UN.PCT.TOWN = 1.0 - (UN.PCT.OPEN + UN.PCT.WOOD)
1222       ELSE
1223          LET UN.PCT.OPEN = 1.00
1224          LET UN.PCT.WOOD = 0.00
1225          LET UN.PCT.TOWN = 0.00
1226       ALWAYS
```

```
1227    ELSE
1228       IF GROUPING = ARTILLERY  '' 2, INCLUDING FA, ADA AND MORTARS
1229          ''UTILIZE NORMAL.F FUNCTION IN NEXT DRAW ---------- \1>(641)
1230          LET UN.PCT.WOOD = NORMAL.F(WOOD.PCT,W.ST.DEV,RN.SEED) *
1231          ( 2.00 - UNIT.M.FACTOR )
1232          IF UN.PCT.WOOD > 1.0
1233             LET UN.PCT.WOOD = 1.0
1234          ELSE
1235             IF UN.PCT.WOOD < 0.0
1236                LET UN.PCT.WOOD = 0.0
1237             ALWAYS
1238          ALWAYS
1239          IF UN.PCT.WOOD < WOOD.PCT
1240             ''UTILIZE NORMAL.F FUNCTION IN NEXT DRAW ---------- \1>(641)
1241             LET UN.PCT.OPEN = NORMAL.F(OPEN.PCT,O.ST.DEV,RN.SEED) *
1242             UNIT.M.FACTOR
1243             IF UN.PCT.OPEN > 1.0
1244                LET UN.PCT.OPEN = 1.0
1245             ELSE
1246                IF UN.PCT.OPEN < 0.0
1247                   LET UN.PCT.OPEN = 0.0
1248                ALWAYS
1249             ALWAYS
1250             IF UN.PCT.WOOD + UN.PCT.OPEN < 1.00
1251                LET UN.PCT.TOWN = 1.00 - (UN.PCT.OPEN + UN.PCT.WOOD)
1252             ELSE
1253                LET UN.PCT.OPEN = 1.00 - UN.PCT.WOOD
1254                LET UN.PCT.TOWN = 0.00
1255             ALWAYS
1256          ELSE
1257             LET UN.PCT.OPEN = 1.00 - UN.PCT.WOOD
1258             LET UN.PCT.TOWN = 0.00
1259          ALWAYS
1260       ELSE
1261          ''GROUPING IS OTHER THAN MANUEVER OR ARTILLERY
1262          ''THIS TYPE OF UNIT DOES NOT MOVE
1263          ''UTILIZE NORMAL.F FUNCTION IN NEXT DRAW ---------- \1>(641)
1264          LET UN.PCT.TOWN = NORMAL.F(TOWN.PCT,T.ST.DEV,RN.SEED)
1265          IF UN.PCT.TOWN > 1.0
1266             LET UN.PCT.TOWN = 1.0
1267          ELSE
1268             IF UN.PCT.TOWN < 0.0
1269                LET UN.PCT.TOWN = 0.0
1270             ALWAYS
1271          ALWAYS
1272          ''UTILIZE NORMAL.F FUNCTION IN NEXT DRAW ---------- \1>(641)
1273          LET UN.PCT.WOOD = NORMAL.F(WOOD.PCT,W.ST.DEV,RN.SEED)
1274          IF UN.PCT.WOOD > 1.0
1275             LET UN.PCT.WOOD = 1.0
1276          ELSE
1277             IF UN.PCT.WOOD < 0.0
1278                LET UN.PCT.WOOD = 0.0
1279             ALWAYS
1280          ALWAYS
1281          IF UN.PCT.TOWN + UN.PCT.WOOD > 1.00
1282             LET UN.PCT.WOOD = 1.00 - UN.PCT.TOWN
1283             LET UN.PCT.OPEN = 0.00
1284          ELSE
```

```
1285              LET UN.PCT.OPEN = 1.00 - (UN.PCT.TOWN + UN.PCT.WOOD)
1286           ALWAYS
1287        ALWAYS
1288     ALWAYS
1289
1290     ''CONVERT VALUES  FOR PRINTING
1291     LET A.UNPCT.OPEN = UN.PCT.OPEN * 100.00
1292     LET A.UNPCT.WOOD = UN.PCT.WOOD * 100.00
1293     LET A.UNPCT.TOWN = UN.PCT.TOWN * 100.00
1294
1295     IF DEBUG = TRUE
1296        PRINT 3 LINES WITH
1297        A.UNPCT.OPEN, A.UNPCT.WOOD, A.UNPCT.TOWN THUS
1298        IT HAS ***.** PERCENT OF THE TROOPS IN THE OPEN
1299        IT HAS ***.** PERCENT OF THE TROOPS IN THE WOODS
1300        IT HAS ***.** PERCENT OF THE TROOPS IN BUILT-UP AREAS
1301        SKIP 1 LINE
1302     ALWAYS
1303
1304
1305 <--EXITROUTINE
1306 ENDROUTINE
```

```
1307 ROUTINE VOLLEY                                                                                                    IF41
1308    GIVEN
1309    BTRY
1310
1311    ADD 1 TO ANAL.CTR(92,1)  ''                                                                                    \DYN_ANAL
1312    NORMALLY MODE IS INTEGER
1313    DEFINE SOUND.TIME,SOUND.RANGE AS REAL VARIABLES
1314    DEFINE A.FLASH, A.SOUND AS TEXT VARIABLES ''                                                                   \TEXT
1315    DEFINE BTRY, SECTOR AS INTEGER VARIABLES
1316
1317    CALL LOCATE.SECTOR------------------------------------------------------------------------------------------>(100)
1318    GIVEN
1319    UN.Y.COORD(BY.UNIT(BTRY))
1320    YIELDING
1321    SECTOR
1322    IF UN.COLOR(BY.UNIT(BTRY)) = BLUE
1323       LET  ENEMY = RED
1324    ELSE
1325       LET ENEMY = BLUE
1326    ALWAYS
1327    IF SS.SET(ENEMY, SECTOR) IS EMPTY
1328       ADD 1 TO SECTOR
1329       IF SECTOR > N.SECTOR
1330          SUBTRACT 2 FROM SECTOR
1331       ALWAYS
1332       IF SS.SET(ENEMY, SECTOR) IS EMPTY
1333 <--------EXITROUTINE
1334       OTHERWISE
1335    ALWAYS
1336    IF ENEMY = RED
1337       LET FEBA = L.SS.SET(ENEMY, SECTOR)
1338    ELSE
1339       LET FEBA = F.SS.SET(ENEMY, SECTOR)
1340    ALWAYS
1341    ''UTILIZE ACT.RANGE FUNCTION IN NEXT ASSIGMENT ------------------------------------------------------------ \1>(628)
1342    LET SOUND.RANGE = ACT.RANGE(BY.UNIT(BTRY),FEBA)
1343    LET SOUND.TIME = 16.*SOUND.RANGE/350.
1344    LET A.FLASH = "FL"
1345    SCHEDULE_A PDB.ACTIVATION------------------------------------------------------------------------------------>(386)
1346    GIVEN
1347    BTRY,                                                                                                          \1
1348    A.FLASH NOW ''
1349    LET A.SOUND = "SD"
1350    SCHEDULE_A PDB.ACTIVATION------------------------------------------------------------------------------------>(386)
1351    GIVEN
1352    BTRY,
1353    A.SOUND
1354    IN SOUND.TIME SECONDS
1355    CREATE A IF.VOLLEY CALLED IF.V
1356    LET IF.V.TIME(IF.V) = TIME.V
1357    LET IF.V.BTRY(IF.V) = BTRY
1358    FILE IF.V IN IF.RATE.LIST
1359    SCHEDULE_A CFR.ACTIVATION------------------------------------------------------------------------------------>(358)
1360    GIVEN
1361    BTRY
1362    IN 15.0 SECONDS
1363
1364 <--EXITROUTINE
```

```
1365 ENDROUTINE
```

```
1366 ROUTINE WEIGHTED.VOLLEYS                                                                                    IF42
1367    GIVEN
1368    BTRY,
1369    BEST.MUNITION,
1370    CLASS,
1371    BEST.LETH.AREA,
1372    MISSION
1373    YIELDING
1374    WEIGHTED.VOLS,
1375    NUMBER.VOLLEYS,
1376    FRACTION.COVERED
1377
1378    ADD 1 TO ANAL.CTR(93,1)  ''                                                                          \DYN_ANAL
1379    ''THIS ROUTINE COMPUTES THE TOTAL COST(DOLLARS) OR
1380    ''THE TOTAL WEIGHT OF THE ROUNDS OF THIS TYPE THAT
1381    ''WOULD BE NECESSARY FOR THIS FIRE UNIT TO DELIVER ON THE
1382    ''TARGET IN ORDER TO ACHIEVE THE REQUIRED EFFECTS OR
1383    ''"ATTACK CRITERIA"
1384
1385    NORMALLY MODE IS INTEGER
1386    DEFINE CLASS AS A TEXT VARIABLE ''                                                                    \TEXT
1387    DEFINE SIDE AND MISSION AS INTEGER VARIABLES
1388    DEFINE FRACTION.COVERED,  REAL.VOLLEYS, BEST.LETH.AREA, WEIGHTED.VOLS,
1389    BN.REM.EFFECTS AS REAL VARIABLES
1390
1391    LET TARGET = FM.TGT(MISSION)
1392
1393    IF DEBUG=TRUE,
1394       PRINT 1 LINE WITH TARGET,BTRY,BEST.MUNITION,BEST.LETH.AREA,MISSION
1395       THUS
1396       = = = WEIGHTED.VOLS TGT=******, BTRY=***, MUN=***, LA=******, MSN=****** = = =
1397    ALWAYS
1398
1399    CALL EST.COVERAGE------------------------------------------------------------------------------------------>(176)
1400    GIVEN
1401    MISSION,
1402    CLASS,
1403    BEST.MUNITION
1404    YIELDING
1405    FRACTION.COVERED,
1406    RNG.HACK
1407    IF RNG.HACK = 0
1408       IF DEBUG = TRUE
1409          PRINT 1 LINE WITH RNG.HACK THUS
1410          = = = WEIGHT.VOLS === EXIT = = RNG.HACK = ********
1411       ALWAYS
1412 <-----EXITROUTINE
1413    OTHERWISE
1414
1415    LET TUBES.AVAILABLE = N.BY.HOW.SET(BTRY)*TB.RND.PER.LAUNCH(BY.TYPE(BTRY))
1416    LET BN.REM.EFFECTS = REAL.F(TR.REM.EFFECTS( TARGET )) / 10000.
1417    IF BN.REM.EFFECTS GE 1.00
1418       PRINT 1 LINE WITH BN.REM.EFFECTS THUS
1419       = = = WEIGHT.VOLS  BN.REM.EFFECTS = **.*** HIGH = = =
1420       LET BN.REM.EFFECTS = 0.995
1421    ALWAYS
1422
1423    LET SIDE = UN.COLOR(BY.UNIT(BTRY))
```

```
1424     IF  CLASS  = "ICM"
1425        LET VOLLEY.RAD.OF.EFFECTS = IC.VOLLEY.RAD( BEST.MUNITION )
1426        IF COST.CRITERIA(SIDE) = TRUE
1427           LET COST = IC.COST( BEST.MUNITION )
1428        ELSE
1429           LET COST = IC.WEIGHT( BEST.MUNITION )
1430        ALWAYS
1431     ELSE     ''THIS IS A HE MUNITION
1432        LET VOLLEY.RAD.OF.EFFECTS = HE.VOLLEY.RAD( BEST.MUNITION )
1433        IF COST.CRITERIA(SIDE) = TRUE
1434           LET COST = HE.COST( BEST.MUNITION )
1435        ELSE
1436           LET COST = HE.WEIGHT( BEST.MUNITION )
1437        ALWAYS
1438     ALWAYS
1439     IF FRACTION.COVERED <= 0.000001
1440        LET NUMBER.VOLLEYS = 99999
1441        LET WEIGHTED.VOLS = REAL.VOLLEYS * COST   ''TOTAL DOLLARS OR WEIGHT
1442 <-----EXITROUTINE
1443     OTHERWISE
1444
1445     LET REAL.VOLLEYS = -PI.C * ( VOLLEY.RAD.OF.EFFECTS ** 2 ) * ''                \OPTIMIZE
1446     LOG.E.F( 1.0 - BN.REM.EFFECTS ) /
1447     ( TUBES.AVAILABLE * BEST.LETH.AREA * FRACTION.COVERED )
1448     IF DEBUG = TRUE,
1449        PRINT 2 LINES WITH -PI.C,VOLLEY.RAD.OF.EFFECTS,BN.REM.EFFECTS,
1450        REAL.VOLLEYS THUS
1451        = = = WEIGHTED.VOLS -PI.C = *.*** , VOLLEY.RAD.OF.EFFECTS = ****,
1452        BN.REM.EFFECTS = *.****, REAL.VOLLEYS = ***.***     = = =
1453     ALWAYS
1454     LET NUMBER.VOLLEYS = INT.F(REAL.VOLLEYS+.5) ''TOTAL ROUNDS
1455     LET WEIGHTED.VOLS = REAL.VOLLEYS * COST   ''TOTAL DOLLARS OR WEIGHT
1456
1457 <--EXITROUTINE
1458 ENDROUTINE
```

```
1459 ROUTINE MINE.EFFECTS                                                    ''
1460     GIVEN                                                                          IF43
1461     .UNIT,
1462     .XCOR,
1463     .YCOR,
1464     .TYPE,
1465     FM,
1466     .MINE.FLAG
1467
1468     YIELDING
1469     .DELAY
1470
1471     ADD 1 TO ANAL.CTR(94,1)  ''
1472     ''THIS ROUTINE DIRECTS CASUALTY ASSESSMENT AND DELAY TIME                  \DYN_ANAL
1473     ''COMPUTATIONS
1474
1475     NORMALLY MODE IS INTEGER
1476     DEFINE .FRAC.CASUALTIES AS A REAL VARIABLE
1477     DEFINE .NAME AS A TEXT VARIABLE ''
1478     DEFINE BARRIER TO MEAN 1                                                   \TEXT
1479     DEFINE MOPMS TO MEAN 2
1480     DEFINE ARTY.MINE TO MEAN 3
1481
1482     IF UN.COLOR(.UNIT) = BLUE
1483        LET .KILLER.COLOR = RED
1484     ELSE
1485        LET .KILLER.COLOR = BLUE
1486     ALWAYS
1487     IF .TYPE = BARRIER
1488        LET .NAME = "MINEFD"
1489     ALWAYS
1490     IF .TYPE = MOPMS
1491        LET .NAME= "MOPMS"
1492     ALWAYS
1493     IF .TYPE = ARTY.MINE
1494        LET .NAME = "FASCAM"
1495     ALWAYS
1496
1497     FOR EVERY TYPE.EQUIPMENT CALLED .TE
1498     WITH TE.NAME(.TE) = "MINES"
1499     FIND THE FIRST CASE
1500     IF FOUND
1501        IF .KILLER.COLOR = BLUE
1502           LET .I1 = 1
1503           LET .I2 = N.BLUE.TYPE.EQP
1504        ELSE
1505           LET .I1 = N.BLUE.TYPE.EQP + 1
1506           LET .I2 = N.EQUIPMENT
1507        ALWAYS
1508        FOR .KILLER.EQ = .I1 TO .I2
1509        WITH EQ.NAME(.KILLER.EQ) = .NAME
1510        FIND THE FIRST CASE
1511        IF FOUND
1512           LET .KILLER = EQ.KV.ID(.KILLER.EQ)
1513        ELSE
1514           PRINT 1 LINE WITH .TYPE THUS
1515           ==MINE.EFFECTS - NO KV ID FOR A MINE OF TYPE *
1516        ALWAYS
```

```
1517    ALWAYS
1518
1519    CALL MINE.DELAY------------------------------------------------------------------------------------------>(103)
1520    GIVEN
1521    .UNIT,
1522    .XCOR,
1523    .YCOR,
1524    .TYPE
1525    YIELDING
1526    .DELAY
1527
1528    ''COMPUTE LOSSES
1529    IF .UNIT IS IN A FR.UNIT.SET
1530       ''ATTRITION IS HANDLED THRU SHOOT OUTS
1531       ACTIVATE_A MINE.ASSESS-------------------------------------------------------------------------------->(464)
1532       CALLED .MA
1533       GIVEN
1534       .UNIT,
1535       .KILLER,
1536       NO NOW ''                                                                                              \1
1537       LET MA.MSN(.MA) = FM
1538
1539       IF MF.DEBUG = TRUE
1540          PRINT 1 LINE WITH .UNIT, .MA THUS
1541          ===MINE.EFFECTS  TGT UNIT= ****, MINE ASSESS= ******
1542       ALWAYS
1543
1544    ELSE
1545       ''ATTRITION IS HANDLED BELOW
1546       LET .MA = 0
1547       IF MF.DEBUG = TRUE
1548          PRINT 1 LINE WITH
1549          .UNIT,
1550          .DELAY THUS
1551          ===MINE.EFFECTS - UNIT=**** NOT IN BATTLE, DELAYED *** MIN
1552       ALWAYS
1553    ALWAYS
1554
1555    IF .MINE.FLAG LE
1556       LOOP FOR EACH  EQ IN UN.EQUIP.LIST(.UNIT)
1557       WITH UE.QUANT(.EQ) GT 0
1558       DO
1559          LET ..TE = EQ.TE.PTR(UE.ID(.EQ))
1560          LET .VICTIM = EQ.KV.ID(UE.ID(.EQ))
1561          LET .CASUALTIES = UNIFORM.F(REAL.F(TE.MIN.MF.LOSS(..TE)),
1562          REAL.F(TE.MAX.MF.LOSS(..TE)), RN.SEED)
1563          LET .CASUALTIES = MIN.F(.CASUALTIES, UE.QUANT(.EQ))
1564
1565          IF .MA GT 0
1566             CREATE A MA.LINK CALLED .MAL
1567             FILE .MAL IN THE MA.SET(.MA)
1568             LET MA.UE.LINK(.MAL) = .EQ
1569             LET MA.CASUALTIES(.MAL) = .CASUALTIES
1570          ELSE
1571             ''COMPUTE ATTRITION
1572             LET .FRAC.CASUALTIES = .CASUALTIES / UE.QUANT(.EQ)
1573             SUBTRACT .CASUALTIES FROM UE.QUANT(.EQ)
1574             IF UN.PTR(.UNIT) GT 0 AND
```

```
1575              UE.CRITICAL.EQUIP.INDIC(.EQ) = TRUE
1576              SUBTRACT .CASUALTIES FROM MU.CRIT.NO(UN.PTR(.UNIT))
1577           ALWAYS
1578           IF .KILLER GT 0 AND .VICTIM GT 0
1579              ADD .CASUALTIES TO
1580              KV.SCORE(.KILLER.COLOR, .KILLER, .VICTIM)
1581              IF ANALYSIS(5) = TRUE
1582                 IF .CASUALTIES > 0
1583                    FOR .I = 1 TO .CASUALTIES
1584                    WRITE TIME.V,.NAME,.TE,EQ.NAME(UE.ID(.EQ)),
1585                    ..TE,.KILLER.COLOR,.FRAC.CASUALTIES
1586                    AS S 1,"1",S 1,D(7,4),S 1,"    1",S 1,
1587                    T 6,S 3,I 1,S 3,"MINES ",S 3,T 6,S 3,''                    \TEXT
1588                    I 1,S 3,"999999",S 4,"0",S 4,
1589                    "9999900000",S 4,I 1,S 4,D(4,3),S 4,
1590                    "1MF",/ USING UNIT 55
1591                 ALWAYS
1592
1593                 FOR .I = 1 TO (UE.QUANT(.EQ)-.CASUALTIES)
1594                 WRITE TIME.V,.NAME,.TE,EQ.NAME(UE.ID(.EQ)),
1595                 ..TE,.KILLER.COLOR,.FRAC.CASUALTIES
1596                 AS S 1,"0",S 1,D(7,4),S 1,"    1",S 1,
1597                 T 6,S 3,I 1,S 3,"MINES ",S 3,T 6,S 3,''                    \TEXT
1598                 I 1,S 3,"999999",S 4,"0",S 4,
1599                 "9999900000",S 4,I 1,S 4,D(4,3),S 4,
1600                 "1MF",/ USING UNIT 55
1601              ALWAYS
1602           ALWAYS
1603
1604           LET .GROUPING = CT.GROUP(TU.CAT(UN.TYPE.UNIT(.UNIT)))
1605           IF .GROUPING = ARTILLERY AND .CASUALTIES GT 0 AND
1606              UN.STATUS(.UNIT) NE STATIONARY AND
1607              UN.STATUS(.UNIT) NE STA.TO.WITH
1608              ''HOW ONLY ATTRITTED IF MOVING
1609              LET .TGT.BTRY = UN.BTRY.INDEX(.UNIT)
1610           THEN IF EQ.TE.PTR(TB.HOW.EQ.ID(BY.TYPE(.TGT.BTRY)))=..TE
1611              LET .TOT.HOW.LOSS = 0
1612              LOOP UNTIL .TOT.HOW.LOSS GE .CASUALTIES OR
1613              BY.HOW.SET(.TGT.BTRY) IS EMPTY
1614              DO
1615                 LOOP FOR EACH .HOW IN BY.HOW.SET(.TGT.BTRY)
1616                 UNTIL .TOT.HOW.LOSS GE .CASUALTIES
1617                 DO
1618                    IF RANDOM.F(RN.SEED) LT .FRAC.CASUALTIES
1619                       ADD 1 TO .TOT.HOW.LOSS
1620                       REMOVE .HOW FROM BY.HOW.SET(.TGT.BTRY)
1621                       DESTROY THE HOW CALLED .HOW
1622                    ALWAYS
1623                 ENDLOOP
1624              ENDLOOP
1625           THEN IF N.BY.HOW.SET(.TGT.BTRY) LT
1626              TB.MIN.HOW(BY.TYPE(.TGT.BTRY))
1627              LET UN.TIME.LAST.MOVE(.UNIT) = TIME.V -
1628              TIME.BETWEEN.ARTY.MOVE
1629              CALL FA.BN.MOVEMENT----------------------------------------------------------------------------------------->( 89)
1630              GIVEN
1631              BY.BN(.TGT.BTRY)
1632           ALWAYS
```

```
1633            IF ANALYSIS(6) GT 0 AND .KILLER.EQ GT 0 AND FM GT 0
1634               CALL OUTPUT.ATTRITION------------------------------------------------------------------------------------------>(611)
1635               GIVEN
1636               .UNIT,
1637               UE.ID(.KILLER.EQ),
1638               .CASUALTIES,
1639               .NAME,
1640               FM
1641            ALWAYS
1642            IF MF.DEBUG = TRUE
1643               PRINT 1 LINE WITH
1644               .UNIT,
1645               UE.ID(.EQ),
1646               .CASUALTIES,
1647               UE.QUANT(.EQ) THUS
1648               ===MINE.EFFECTS UNIT=****, EQ=****, KILLED=****, REM=****
1649            ALWAYS
1650         ALWAYS
1651
1652         LOOP FOR EACH .LINK IN UN.SENSOR.LIST(.UNIT)
1653         WHEN US.EQ.ID(.LINK) = UE.ID(.EQ)
1654         DO
1655            CALL ATTRIT.SENSOR---------------------------------------------------------------------------------------------->(159)
1656            GIVEN
1657            .LINK,
1658            .FRAC.CASUALTIES
1659         ENDLOOP
1660      ENDLOOP
1661   ALWAYS
1662
1663
1664   IF MF.DEBUG = TRUE
1665      PRINT 1 LINE WITH .UNIT AND .DELAY THUS
1666      ===MINE.EFFECTS = UNIT ***** DELAYED ********** MINS
1667   ALWAYS
1668
1669 <--EXITROUTINE
1670 ENDROUTINE
```

```
1671 ROUTINE FO.DETECTION                                                    ''                    IF44
1672    GIVEN
1673    CANDIDATE,
1674    FO,
1675    TARGET,
1676    FO.X.CORRECT,
1677    FO.Y.CORRECT
1678
1679    ADD 1 TO ANAL.CTR(95,1)  ''                                                                \DYN_ANAL
1680    NORMALLY MODE IS INTEGER
1681    DEFINE PROB.LINE.OF.SITE, DEGRADATION, ELEM.PROB.DETECT AS REAL VARIABLES
1682    DEFINE NORM1, NORM2 AS REAL VARIABLES
1683    DEFINE SIDE, SENSOR.TYPE AS INTEGER VARIABLES
1684    DEFINE TARGET.OF.OPPORTUNITY TO MEAN 0
1685
1686    LET LINK = FO.US.LINK(FO)
1687    LET SENSOR = US.UNIT(LINK)
1688    LET TGT =FO.DC.UNIT( CANDIDATE )                                                           CHG\08
1689    LET MODEL = US.MODEL(LINK)  ''                                                             \ZERO_SUB
1690    IF DEBUG = TRUE OR MFO.NAME(MODEL) = "RPV" ''
1691       SKIP 1 LINE
1692       PRINT 1 LINE WITH FO, TARGET, TIME.V THUS
1693       = = = FO.DETECTION  FO = ****** ,TARGET = ****** ,  ***.*** HOURS  = = =
1694    ALWAYS
1695    IF MFO.NAME(MODEL) = "RPV"
1696       PRINT 1 LINE WITH TIME.V, US.UNIT(LINK) THUS
1697       = = FO.DET = RPV DETECTION AT **.***** BY UNIT ********
1698    ALWAYS
1699    LET SENSOR.TYPE = US.SENSOR.TYPE(LINK)
1700    LET SIDE = UN.COLOR( SENSOR )
1701    LET RANGE = SQRT.F((UN.X.COORD(TGT)-(UN.X.COORD(SENSOR)+FO.X.CORRECT)) ** 2  ''            \OPTIMIZE
1702    + (UN.Y.COORD(TGT)-(UN.Y.COORD(SENSOR)+FO.Y.CORRECT)) ** 2)  ''                            \OPTIMIZE
1703    IF UN.STATUS(TGT) = ADVANCING OR
1704       UN.STATUS(TGT) = WITHDRAWING OR
1705       UN.STATUS(TGT) = MOVING
1706       LET MOVEMENT = 1
1707    ELSE
1708       LET MOVEMENT = 2
1709    ALWAYS
1710    IF RANGE > FO.RB.RANGE(L.MFO.RB.SET(MODEL))
1711       IF DEBUG = TRUE OR MFO.NAME(MODEL) = "RPV"
1712          PRINT 1 LINE THUS
1713          = = = FO.DETECT = TARGET OUT OF RANGE - EXIT
1714       ALWAYS
1715 <-----EXITROUTINE
1716    OTHERWISE
1717
1718    IF US.STATUS(FO.US.LINK(FO)) = TARGET.OF.OPPORTUNITY
1719       LET PROB.LINE.OF.SITE = .54 * EXP.F(-.0037 * RANGE * 1.6) ''DCM
1720       IF PROB.LINE.OF.SITE < RANDOM.F(RN.SEED)
1721          IF DEBUG = TRUE OR MFO.NAME(MODEL) = "RPV"
1722             PRINT 1 LINE WITH PROB.LINE.OF.SITE THUS
1723             = = = PLOS = *.****  NO LOS - EXIT
1724          ALWAYS
1725 <--------EXITROUTINE
1726       OTHERWISE
1727    ELSE
1728       LET MANUEVER.SUPPORT = TRUE
```

```
1729        FOR EACH VISIBLE.UNIT IN UN.LOS.LIST(SENSOR)
1730        WHEN VU.POINTER(VISIBLE.UNIT) = TGT
1731        FIND THE FIRST CASE
1732        IF NONE
1733           IF DEBUG = TRUE OR MFO.NAME(MODEL) = "RPV"
1734              PRINT 1 LINE THUS
1735              = = = NO LOS(BATTLE) - EXIT
1736           ALWAYS
1737  <--------EXITROUTINE
1738        OTHERWISE
1739     ALWAYS
1740
1741     LOOP FOR EACH RNG.BND IN MFO.RB.SET(MODEL)
1742     WITH FO.RB.RANGE(RNG.BND) GE RANGE
1743     FIND THE FIRST CASE
1744     IF NONE
1745        IF DEBUG = TRUE OR MFO.NAME(MODEL) = "RPV"
1746           PRINT 1 LINE THUS
1747           = = = FO.DETECTION = TGT OUT OF RANGE
1748        ALWAYS
1749  <-----EXITROUTINE
1750     OTHERWISE
1751
1752     LET DEGRADATION = FO.VISIBILITY(RNG.BND)/100.
1753     IF NITE.OR.DAY = NITE
1754        LET DEGRADATION = 1. - (1. - DEGRADATION) * NITE.VIS.PCT/100
1755     ALWAYS
1756     IF DEGRADATION < RANDOM.F(RN.SEED)
1757        IF DEBUG = TRUE OR MFO.NAME(MODEL) = "RPV"
1758           PRINT 1 LINE WITH DEGRADATION THUS
1759           = = = DEGRADATION = *.****  - EXIT
1760        ALWAYS
1761  <-----EXITROUTINE
1762     OTHERWISE
1763
1764     LOOP FOR EACH EQUIP IN UN.EQUIP.LIST( TGT )
1765     WHEN UE.QUANT( EQUIP ) > 0
1766     DO THIS
1767        LET EQUIP.TYPE = EQ.TE.PTR( UE.ID( EQUIP ) )
1768        LET ELEM.PROB.DETECT = MNMTR.PROB.DETECT(MODEL,NITE.OR.DAY,MOVEMENT,
1769        EQUIP.TYPE,RNG.BND)/100.
1770        IF  ELEM.PROB.DETECT = 0.0
1771     <-----CYCLE
1772        OTHERWISE
1773
1774        LET DETECT.QUANT = 0
1775        LET NUMBER.IN.LOS = RANDI.F(1, UE.QUANT(EQUIP), RN.SEED)
1776        LOOP FOR I = 1 TO NUMBER.IN.LOS
1777        DO
1778           IF ELEM.PROB.DETECT >= RANDOM.F(RN.SEED)
1779              LET DETECT.QUANT = DETECT.QUANT + 1
1780           ALWAYS
1781        ENDLOOP
1782        IF DETECT.QUANT = 0
1783     <-----CYCLE
1784        OTHERWISE
1785
1786        FOR EACH TDL IN TR.DET.LIST( TARGET )
```

```
1787        WHEN TR.DET.TE( TDL ) = EQUIP.TYPE
1788        FIND THE FIRST CASE
1789        IF NONE
1790           CREATE A TR.DET.LINK CALLED TDL
1791           FILE TDL IN TR.DET.LIST( TARGET )
1792           LET TR.DET.TE( TDL ) = EQUIP.TYPE
1793           LET TR.DET.ELEM.PROB(TDL) = INT.F(100. * ELEM.PROB.DETECT)
1794           IF TR.PGM.STATUS( TARGET ) = FALSE  AND
1795              MFO.PGM.CAP(MODEL) = TRUE AND
1796              TE.PGM.INDIC( EQUIP.TYPE ) = TRUE AND
1797              RANGE*16 LE SDM.PGM.RNG AND
1798              RANDOM.F(RN.SEED) LE GOOD.WEATHER.PROB/100
1799              AND MFO.NAME(MODEL) NE "RPV"
1800              ''THIS CODE PUT IN TO PLAY LIMITED USE OF COPPERHEAD
1801              ''BAD WEATHER AND SHOULD BE REPLACED WHEN
1802              ''THE VISIBILITY ENHANCEMENT IS DONE
1803              LET TR.PGM.STATUS( TARGET ) = TRUE
1804              IF DEBUG = TRUE OR MFO.NAME(MODEL) = "RPV"
1805                 PRINT 1 LINE WITH TARGET THUS
1806                 $#$#$#$#$# PGM MISSION ASSIGNED FOR TARGET ********* #$#$#$#$#$#
1807              ALWAYS
1808           ALWAYS
1809
1810
1811           ''THE FOLLOWING CODE DEPICTS RPV DETECTIONS FOR CUHD
1812
1813           IF TR.PGM.STATUS(TARGET) = FALSE AND MFO.PGM.CAP(MODEL) = TRUE
1814              AND MFO.NAME(MODEL) = "RPV"
1815              IF TE.PGM.INDIC(EQUIP.TYPE) = TRUE AND (UN.STATUS(TGT) =
1816                 ADVANCING OR UN.STATUS(TGT) = WITHDRAWING)
1817                 LET TR.PGM.STATUS(TARGET) = TRUE
1818                 PRINT 1 LINE WITH TARGET AND TIME.V THUS
1819                 $$$$$ RPV CUHD MISSION FOR TARGET ****** AT TIME **.*****
1820              ALWAYS
1821           ALWAYS
1822           LET TR.DET.QUANT( TDL ) = 0
1823        ALWAYS
1824        LET TR.DET.QUANT( TDL ) = TR.DET.QUANT( TDL ) + DETECT.QUANT
1825     ENDLOOP
1826
1827     IF TR.DET.LIST( TARGET ) IS EMPTY
1828        IF DEBUG = TRUE OR MFO.NAME(MODEL) = "RPV"
1829           PRINT 1 LINE THUS
1830           = = = FO.DETECT = TR.DET.LIST IS EMPTY - EXIT
1831        ALWAYS
1832 <------EXITROUTINE
1833     OTHERWISE
1834
1835     LET TR.FDC(TARGET)=US.FDC(LINK)
1836     IF TR.PGM.STATUS(TARGET)=TRUE AND FD.FDC(TR.FDC(TARGET)) GT 0
1837        LET TR.FDC(TARGET)=FD.FDC(TR.FDC(TARGET))
1838     ALWAYS
1839     LET TR.SENSOR.TYPE(TARGET) = ST.NAME(SENSOR.TYPE)
1840     LET TR.SENSOR.ID   ( TARGET ) = FO
1841     LET TR.REP.UNIT( TARGET ) = SENSOR
1842     LET TR.TGT.UNIT( TARGET ) = TGT
1843     IF UN.STATUS(TGT) = STA.TO.WITH  OR
1844        UN.STATUS(TGT) = ADV.TO.WITH  OR
```

```
1845        UN.STATUS(TGT) = STATIONARY
1846        LET TR.MOVE(TARGET) = FALSE
1847     ELSE
1848        LET TR.MOVE(TARGET) = TRUE
1849        IF TGT IS IN A FR.UNIT.SET
1850           CALL PRED.POS ------------------------------------------------------------------------------------------>(111)
1851           GIVEN
1852           TGT,
1853           3.0
1854           YIELDING
1855           DELTA.X,
1856           DELTA.Y
1857        ALWAYS
1858     ALWAYS
1859
1860     IF DEBUG = TRUE OR MFO.NAME(MODEL) = "RPV"
1861        PRINT 1 LINE WITH TR.SENSOR.ID(TARGET),FO.FST.INDIC(FO),
1862        TR.TGT.UNIT(TARGET), RANGE*16 THUS
1863        = = = FO ******(*) HAS DETECTED UNIT **** AT ***** METERS = = =
1864        SKIP 1 LINE
1865     ALWAYS
1866
1867     LET TR.CEP( TARGET ) = FO.CIR.ERROR( RNG.BND )
1868     IF TR.CEP(TARGET) = 0
1869        PRINT 1 LINE WITH RANGE, MODEL, TARGET,TR.REP.UNIT(TARGET)
1870        THUS
1871        = = = FO.DETECT AT RNG ******* FO MODEL *** TGT ****** UNIT ***** 0 CEP
1872        LET TR.CEP(TARGET) = 1
1873     ALWAYS
1874     ''UTILIZE NORMAL.F FUNCTION IN NEXT 2 DRAWS ------------------------------------------------------------ \1>(641)
1875     LET NORM1 = NORMAL.F(0.0,1.0,1)
1876     LET NORM2 = NORMAL.F(0.0,1.0,1)
1877     LET TR.EST.X( TARGET ) = UN.X.COORD( TGT ) + DELTA.X +
1878     NORM1 * TR.CEP(TARGET) / (1.1774 * 16.)
1879     LET TR.EST.Y( TARGET ) = UN.Y.COORD( TGT ) + DELTA.Y +
1880     NORM2 * TR.CEP(TARGET) / (1.1774 * 16.)
1881
1882     IF MANUEVER.SUPPORT = TRUE
1883        IF TR.PGM.STATUS(TARGET) = TRUE
1884           LET TR.MIL.WORTH(TARGET) = 2001
1885        ELSE
1886           LET TR.MIL.WORTH(TARGET) = 2000
1887        ALWAYS
1888     ELSE
1889        IF TR.PGM.STATUS(TARGET) = TRUE
1890           LET TR.MIL.WORTH(TARGET) = 1999
1891        ALWAYS
1892     ALWAYS
1893
1894     IF TR.PGM.STATUS(TARGET) = FALSE AND MFO.PGM.CAP(MODEL) GE TRUE
1895        AND RANGE*16 GT SDM.PGM.RNG AND TGT IS NOT IN AN FR.UNIT.SET
1896        AND UN.STATUS(TGT) NE WITHDRAWING AND UN.STATUS(TGT) NE ADVANCING
1897        FOR EACH TDL IN TR.DET.LIST(TARGET)
1898        WITH TE.PGM.INDIC(TR.DET.TE(TDL)) GE TRUE
1899        ADD TR.DET.QUANT(TDL) TO NUM.DET.SDM
1900        IF NUM.DET.SDM GE SADARM.THRESHOLD
1901           LET TR.FDC(TARGET) = US.FDC(LINK)
1902           LET .FDC = TR.FDC(TARGET)
```

```
1903          PRINT 1 LINE WITH TARGET, .FDC THUS
1904          = = = FO.DETECTION POTENTIAL SDM TGT ******** AT FDC *****
1905          UNTIL .FDC = 0,
1906          DO   ''                                                                    \1
1907             FOR EACH .BN IN FD.BN.LIST(.FDC),
1908             DO  ''                                                                  \1
1909                FOR EACH .BTRY IN BN.BTRY.SET(FB.BN(.BN))
1910                WITH BY.PGM.CAP(.BTRY) = 2 '' STATUS BECAME CAP          CHG\25 \ERROR_CORRECTION
1911                ''BY.PGM.CAP=2 EQUALS SADARM
1912                FIND THE FIRST CASE
1913                IF FOUND
1914                   LET TR.PGM.STATUS(TARGET) = 2 '' EQUALS SADARM TARGET
1915                   LET TR.FDC(TARGET) = .FDC
1916                   LET .FDC = 0
1917             <------LEAVE
1918                OTHERWISE
1919             ENDLOOP
1920             IF TR.PGM.STATUS(TARGET) NE 2
1921                LET .FDC = FD.FDC(.FDC)
1922             ALWAYS
1923          ENDLOOP
1924          IF TR.PGM.STATUS(TARGET) GT 1
1925             LET TR.MIL.WORTH(TARGET) = 1999
1926          ALWAYS
1927          PRINT 1 LINE WITH TR, TR.FDC(TARGET), TR.PGM.STATUS(TARGET) THUS
1928          = = = FO.DETECTION SDM TGT ****** GOES TO FDC ***** PGM STATUS ****
1929       ALWAYS
1930    ALWAYS
1931
1932
1933
1934    ''THE FOLLOWING CODE IS FOR RPV SADARM MISSIONS
1935
1936    IF TR.PGM.STATUS(TARGET) = FALSE AND MFO.PGM.CAP(MODEL) = TRUE
1937       AND MFO.NAME(MODEL) = "RPV" AND UN.STATUS(TGT) NE ADVANCING
1938       AND UN.STATUS(TGT) NE WITHDRAWING
1939       FOR EACH TDL IN TR.DET.LIST(TARGET) WITH
1940       TE.PGM.INDIC(TR.DET.TE(TDL)) GE TRUE
1941       ADD TR.DET.QUANT(TDL) TO NUM.DET.SDM
1942       IF NUM.DET.SDM GE SADARM.THRESHOLD
1943          LET TR.PGM.STATUS(TARGET) = 2
1944          PRINT 1 LINE WITH TARGET AND TIME.V THUS
1945          $$$$$ RPV SADARM MISSION FOR TARGET ****** AT TIME **.*****
1946          LET TR.FDC(TARGET) = US.FDC(LINK)
1947          IF FD.FDC(TR.FDC(TARGET)) NE 0
1948             LET TR.FDC(TARGET) = FD.FDC(TR.FDC(TARGET))
1949          ENDIF
1950          IF FD.FDC(TR.FDC(TARGET)) NE 0
1951             LET TR.FDC(TARGET) = FD.FDC(TR.FDC(TARGET))
1952          ENDIF
1953          IF FD.FDC(TR.FDC(TARGET)) NE 0
1954             LET TR.FDC(TARGET) = FD.FDC(TR.FDC(TARGET))
1955          ENDIF
1956          LET TR.MIL.WORTH(TARGET) = 1999
1957       ENDIF
1958    ALWAYS
1959
1960
```

```
1961
1962 <—EXITROUTINE
1963 ENDROUTINE
```

```
1964 ROUTINE BTRY.EFFECTS                                                                                    ''                                                                                
```

```
2022        TR.EST.Y(TARGET)
2023        YIELDING
2024        SECTOR
2025        FOR EACH UN IN SS.SET(ENEMY.SECTOR)
2026        COMPUTE MIN.DIST.2 AS MINIMUM, UNIT AS MIN(UN) OF
2027        ((UN.X.COORD(UN)-TR.EST.X(TARGET))**2 +  ''                                    \OPTIMIZE
2028        (UN.Y.COORD(UN)-TR.EST.Y(TARGET))**2 )  ''                                     \OPTIMIZE
2029        LET TR.TGT.UNIT(TARGET) = UNIT
2030     ALWAYS
2031
2032     IF TR.PGM.STATUS(TARGET) = TRUE
2033        LET EXT.RND=1
2034     ALWAYS
2035
2036     CALL GET.TERRAIN------------------------------------------------------------------------------------------>(336)
2037     YIELDING
2038     .TERRAIN
2039     IF TR.PGM.STATUS(TARGET) EQ FALSE
2040        CALL UNIT.ENVIR------------------------------------------------------------------------------------------>(242)
2041        GIVEN
2042        UNIT
2043        YIELDING
2044        UN.PCT.OPEN,
2045        UN.PCT.WOOD,
2046        UN.PCT.TOWN
2047        LOOP FOR EACH ENVIRONMENT CALLED ENVIR
2048        DO
2049           IF EN.NAME(ENVIR) = "OPEN"
2050              LET UN.ENVIR.FRACT(ENVIR) = UN.PCT.OPEN
2051           ELSE
2052              IF EN.NAME(ENVIR) = "WOODS"
2053                 LET UN.ENVIR.FRACT(ENVIR) = UN.PCT.WOOD
2054              ELSE
2055                 IF EN.NAME(ENVIR) = "TOWN"
2056                    LET UN.ENVIR.FRACT(ENVIR) = UN.PCT.TOWN
2057                 ALWAYS
2058              ALWAYS
2059           ALWAYS
2060        ENDLOOP
2061     ALWAYS
2062
2063     IF UNIT IS IN A FR.UNIT.SET
2064        IF UN.BATTLE.INDEX(UNIT) NE 9999
2065           IF UN.STATUS(UNIT)  EQ ADVANCING
2066              OR UN.STATUS(UNIT)  EQ WITHDRAWING
2067              FOR EVERY MOVE OF EV.S(I.MOVE)
2068              WITH MV.UNIT(MOVE)=UNIT
2069              FIND THE FIRST CASE
2070              IF NONE,
2071                 CALL ERROR.STOP------------------------------------------------------------------------------>(604)
2072              ALWAYS
2073              CANCEL_THE MOVE------------------------------------------------------------------------------->(381)
2074              DESTROY THIS MOVE
2075              CALL CHANGE.LOC------------------------------------------------------------------------------->( 83)
2076              GIVEN
2077              UNIT
2078           ALWAYS
2079           CREATE AN ARTY.ASSESS CALLED ARTY.ASS
```

```
2080          LET AA.DESTRUCT.INDIC(ARTY.ASS) = NO
2081          LET AA.UNIT(ARTY.ASS) = UNIT
2082          LET AA.FIRE.MISSION(ARTY.ASS) = MSN
2083       ALWAYS
2084    ALWAYS
2085
2086    IF UNIT IS NOT IN FR.UNIT.SET AND
2087       UN.BATTLE.INDEX(UNIT) NE 9999 AND
2088       TR.PGM.STATUS(TARGET) GE TRUE AND
2089       DEBUG = TRUE
2090       PRINT 1 LINE WITH TARGET THUS
2091       $#$# NO ARTY.ASS CREATED FOR TGT *******
2092    ALWAYS
2093
2094    LET MISSION = UN.MISSION(UNIT)
2095    IF MISSION <= 0 OR
2096       MISSION > N.MISSION
2097       LET MISSION = DEFEND
2098    ALWAYS
2099    IF UN.COLOR(BY.UNIT(BTRY))=RED
2100       LET SIDE = RED
2101       LET ENEMY=BLUE
2102    ELSE
2103       LET SIDE = BLUE
2104       LET ENEMY=RED
2105    ALWAYS
2106
2107
2108    LET TU = UN.TYPE.UNIT( UNIT )
2109    LET CAT = TU.CAT(TU)
2110    LET TUBES = N.BY.HOW.SET( BTRY )*TB.RND.PER.LAUNCH(BY.TYPE(BTRY))
2111    LET RANGE = FM.RANGE(MSN)
2112    LET TB = BY.TYPE( BTRY )
2113    LET MUNITION = FM.TM( MSN )
2114    IF TR.PGM.STATUS(TARGET) NE TRUE
2115       ''
2116       ''
2117       ''
2118       '' OUTPUT FOR AMSAA TO UNIT 51
2119       ''
2120       ''
2121       IF ANALYSIS(4) = TRUE
2122          IF TR.PGM.STATUS(FM.TGT(MSN)) = 0
2123             IF FM.TM.CLASS(MSN) = "HE"
2124                LET ROUND.NAME = HE.ID(FM.TM(MSN))
2125                IF EC.FRACT(OPEN,TU.CAT(TR.EST.TU(FM.TGT(MSN)))) > 50
2126                   LET FUZE = VT
2127                ELSE
2128                   LET FUZE = PD
2129                ALWAYS
2130                LET ROUND.RELY = REAL.F(FZ.HE.RELY(FUZE,FM.TM(MSN)))/100.
2131                LET N.TGTS = HE.VOLLEY.RAD(FM.TM(MSN))
2132             ELSE
2133                LET ROUND.NAME = IC.ID(FM.TM(MSN))
2134                LET ROUND.RELY = IC.RELIABILITY(FM.TM(MSN))
2135                LET N.TGTS = IC.VOLLEY.RAD(FM.TM(MSN))
2136             ALWAYS
2137             USE UNIT 51 FOR OUTPUT
```

```
2138            SKIP 1 OUTPUT LINE
2139            PRINT 8 LINES WITH MSN,TB.NAME(TB),TU.LEVEL(UN.TYPE.UNIT(UNIT)),
2140            EQ.NAME(TB.HOW.EQ.ID(TB)),
2141            TU.RADIUS(UN.TYPE.UNIT(UNIT)), RANGE*16, ROUND.NAME, ROUND.RELY,
2142            FM.FIRED.VOLS(MSN),UN.ENGAGED.INDEX(UNIT),N.BY.HOW.SET(BTRY)
2143            *TB.RND.PER.LAUNCH(BY.TYPE(BTRY)), N.TGTS THUS
2144
2145            FIRE MISSION *******   ******* FIRES AT ****** WITH ******
2146            TGT RADIUS         ******* METERS
2147            RANGE              ******* METERS
2148            ROUND               ****** (RELIABILITY  *.****  )
2149            NUMBER VOLLEYS       *****  ENGAGED *
2150            RNDS PER VOLLEY      *****
2151            VOLLEY RADIUS        ***** METERS
2152
2153            IF FM.TM.CLASS(MSN) = "ICM"
2154               WRITE (REAL.F(RANGE)*0.016*IC.TB.SLOPE(FM.TM(MSN),TB)
2155               + IC.TB.INTERCEPT(FM.TM(MSN),TB), IC.N.SUBM(FM.TM(MSN))
2156               AS S 4,"ROUND RAD  ",D(7,4),S 2, "NUM SUBM  ",I 4,/
2157            ALWAYS  ''                                                          \1
2158         ALWAYS
2159         USE UNIT 6 FOR OUTPUT
2160      ALWAYS
2161   ALWAYS
2162
2163   IF TR.PGM.STATUS(TARGET) = FALSE
2164      IF FM.TM.CLASS( MSN ) = "HE"
2165         LET ROUND.NAME = HE.ID(MUNITION)
2166         FOR EACH RNG.HACK IN HE.TB.RH.LIST( MUNITION, TB )
2167         WHEN HE.RH.RANGE( RNG.HACK ) >= RANGE
2168         FIND THE FIRST CASE
2169         IF NONE
2170            CALL ERROR.STOP------------------------------------------------------------------------------------------------------------------>(604)
2171         ALWAYS
2172         IF EC.FRACT( OPEN, TU.CAT(TR.EST.TU( TARGET ))) > 50''VT FUZE WAS
2173            LET FUZE = VT
2174         ELSE
2175            LET FUZE = PD
2176         ALWAYS
2177         LET PI.RAD.2 = PI.C * HE.VOLLEY.RAD( MUNITION ) ** 2  ''                                                \OPTIMIZE
2178         LET ROUND.RELY = REAL.F( FZ.HE.RELY(FUZE, MUNITION)) / 100.
2179         LET QUANT = FM.FIRED.VOLS( MSN ) * TUBES
2180         LET NUM.ROUNDS = QUANT
2181      ELSE  ''FM.TM.CLASS = "ICM"
2182         LET ROUND.NAME = IC.ID(MUNITION)
2183         FOR EACH RNG.HACK IN IC.TB.RH.LIST( MUNITION, TB )
2184         WHEN IC.RH.RANGE( RNG.HACK ) >= RANGE
2185         FIND THE FIRST CASE
2186         IF NONE
2187            CALL ERROR.STOP------------------------------------------------------------------------------------------------------------------>(604)
2188         ALWAYS
2189         LET PI.RAD.2 = PI.C * IC.VOLLEY.RAD( MUNITION ) ** 2
2190         LET NUM.ROUNDS = FM.FIRED.VOLS( MSN ) * TUBES
2191         LET QUANT = NUM.ROUNDS * IC.N.SUBM( MUNITION )
2192         LET ROUND.RELY = REAL.F( IC.RELIABILITY( MUNITION ) ) / 100.
2193         LET ST=IC.SUBM.INDEX(MUNITION)
2194      ALWAYS
2195   ELSE
```

```
2196        IF TR.PGM.STATUS(TARGET) = TRUE ''COPPERHEAD MSN
2197           LET ROUND.NAME="PGM"
2198           LET NUM.ROUNDS = FM.FIRED.VOLS(MSN)
2199           LET ROUND.RELY = 1.0
2200           IF DEBUG = TRUE
2201              PRINT 1 LINE WITH TARGET AND NUM.ROUNDS THUS
2202              $#$#$# PGM MISSION FIRED ON TARGET ********, **** ROUNDS
2203           ALWAYS
2204
2205           ADD NUM.ROUNDS TO NUM.PGM.FIRED
2206        ELSE ''SADARM MSN
2207           LET ROUND.NAME = "SDM"
2208           LOOP FOR EACH EQ IN UN.EQUIP.LIST(UNIT)
2209           WITH TE.PGM.INDIC(EQ.TE.PTR(UE.ID(EQ))) GE TRUE,
2210           DO
2211              ADD UE.QUANT(EQ) TO COUNT
2212           ENDLOOP
2213           ADD (FM.N.VOLS(MSN) * TUBES) TO TOT.FIRED.SDM
2214           LET NUM.ROUNDS = FM.N.VOLS(MSN) * TUBES
2215           IF DEBUG = TRUE
2216              PRINT 1 LINE WITH FM.N.VOLS(MSN)*TUBES AND TARGET
2217              THUS
2218              $#$# ******* ROUNDS OF SDM FIRED FOR TGT *******
2219           ALWAYS
2220        ALWAYS
2221     ALWAYS
2222
2223     LOOP FOR EACH EQ IN UN.EQUIP.LIST(UNIT)
2224     WHEN UE.QUANT(EQ) > 0
2225     DO THIS
2226        LET TE = EQ.TE.PTR(UE.ID(EQ))
2227        LET VICTIM = EQ.KV.ID(UE.ID(EQ))
2228        LET FRAC.CAS = 0.0
2229
2230
2231
2232        LET VICTIM = EQ.KV.ID(UE.ID(EQ)) ''%8FEB79_%GLM
2233        LET FRAC.CAS = 0.0
2234        LET FDS1 = 0.0
2235        LET FDSN = 0.0
2236        LET FDP1 = 0.0
2237        LET FDPN = 0.0
2238        LET FDF1 = 0.0
2239        LET FDFN = 0.0
2240        IF TR.PGM.STATUS(TARGET) EQ FALSE
2241
2242           IF ANALYSIS(4) = TRUE
2243              WRITE UE.QUANT(EQ),TE.NAME(TE) AS I 5,S 4, "TGT EQ ",T 6,/ USING UNIT 51 ''                    \TEXT
2244           ALWAYS
2245           LOOP FOR EACH ENVIRONMENT CALLED ENVIR,
2246           DO THIS   ''                                                                                      \1
2247              IF TE NE PERSONNEL
2248                 ''
2249                 ''
2250                 ''MAKE SURE FRAC.CAS IS INITIALIZED TO ZERO
2251                 ''
2252                 IF ANALYSIS(4) = TRUE
2253                    WRITE FRAC.CAS AS S 3,"FRAC.CAS FOR EQUIP ",S 2,D(9,6),/ USING UNIT 51
```

```
2254              ALWAYS
2255              ''
2256              IF FM.TM.CLASS(MSN) = "HE"
2257                 IF ANALYSIS(4) = TRUE
2258                    WRITE EN.NAME(ENVIR),UN.ENVIR.FRACT(ENVIR),REAL.F(RTEF.LA.EQUIP(RNG.HACK,
2259                    TE,ENVIR,FUZE))/10. AS S 4,
2260                    T 6,"FRACT = ",D(6,3), S 4,"LA = ",D(8,2),/ USING UNIT 51 ''                    \TEXT
2261                 ALWAYS
2262              ELSE ''ICM
2263                 IF ANALYSIS(4) = TRUE
2264                    WRITE EN.NAME(ENVIR),UN.ENVIR.FRACT(ENVIR),TES.LA.EQUIP(TE,ENVIR,ST)/10.,
2265                    REAL.F(ES.RELY(ENVIR,ST)/100.) AS S 2,T 6,S 2,"FRACT = ",D(5,3), ''             \TEXT
2266                    S 2,"LA = ",D(8,2),S 2,"SUBM RELY =",D(5,3),/ USING UNIT 51
2267                 ALWAYS
2268              ALWAYS
2269              ''
2270              ''
2271              LET POST = 0
2272              IF UN.ENVIR.FRACT(ENVIR) = 0.
2273        <-------CYCLE ''NO NEED TO DO CALCULATION FOR ZERO FRACTION
2274              ALWAYS
2275              CALL FINAL.COVERAGE GIVEN TARGET, MSN, ENVIR, POST, TE, FUZE------------------------------------------------------>(188)
2276              YIELDING COVER AND X
2277              IF COVER LT 0.0001
2278                 LET COVER = 0.0
2279              ALWAYS
2280              LET ..X = REAL.F(FM.FIRED.VOLS(MSN))*X
2281              LET ..PK = COVER*(1.0 - EXP.F(..X))*UN.ENVIR.FRACT(ENVIR)
2282              LET FRAC.CAS = FRAC.CAS + ..PK
2283              USE UNIT 51 FOR OUTPUT
2284              IF ANALYSIS(4) = TRUE
2285                 PRINT 1 LINE WITH ..PK,COVER,X THUS
2286                 ..PK = **.******** COVER =  **.******** X = ***.********
2287              ALWAYS
2288
2289
2290              USE UNIT 6 FOR OUTPUT
2291
2292              IF ANALYSIS(4) = TRUE
2293                 WRITE FRAC.CAS AND COVER AS S 3,"FRAC.CAS =",S 1,D(9,6),
2294                 S 2,"COVER =",S 1,D(9,6),/ USING UNIT 51
2295              ALWAYS
2296
2297           ELSE ''PERSONNEL
2298              ''
2299              ''
2300              ''AMSAA OUTPUT
2301              ''
2302              ''
2303              ''MAKE SURE FRAC.CAS IS INITIALIZED TO ZERO
2304              ''
2305              IF ANALYSIS(4) = TRUE
2306                 WRITE FRAC.CAS AS S 3,"FRAC.CAS FOR PERS ",S 2,D(9,6),/ USING UNIT 51
2307                 ''
2308                 WRITE EN.NAME(ENVIR),UN.ENVIR.FRACT(ENVIR) AS S 2,
2309                 T 6,S 4,"FRACT = ",D(5,3),/ USING UNIT 51 ''                                    \TEXT
2310              ALWAYS
2311
```

```
2312              IF FM.TM.CLASS(MSN) = "HE"
2313                 FOR EACH POSTURE CALLED POST
2314                 IF ANALYSIS(4) = TRUE
2315                    WRITE PT.NAME(POST),REAL.F(CPM.UNWARNED.FRACT(CAT,POST,MISSION)/100.),
2316                    REAL.F(CPM.WARNED.FRACT(CAT,POST,MISSION)/100.),
2317                    REAL.F(REPF LA.PERS(RNG.HACK,ENVIR,POST,FUZE)) AS
2318                    S 4,T 6,S 2,"FRAC 1ST & SUBSQ ",D(6,4),S 2,D(6,4),S 2,"LA ",D(8,2),/ ''                    \TEXT
2319                    USING UNIT 51
2320                 ALWAYS
2321              ELSE ''ICM
2322                 FOR EACH POSTURE CALLED POST
2323                 IF ANALYSIS(4) = TRUE
2324                    WRITE PT.NAME(POST),REAL.F(CPM.UNWARNED.FRACT(CAT,POST,MISSION)/100.),
2325                    REAL.F(CPM.WARNED.FRACT(CAT,POST,MISSION)/100.),
2326                    REAL.F(EPS.LA.PERS(ENVIR,POST,ST)/10.) AS
2327                    S 4,T 6,S 2,"FRAC 1ST & SUBSQ ",D(6,4),S 2,D(6,4),S 2,"LA ",D(8,2),/ ''                    \TEXT
2328                    USING UNIT 51
2329                 ALWAYS
2330              ALWAYS
2331              ''
2332              ''
2333              LET SN = REAL.F(CPM.WARNED.FRACT(CAT,STAND,MISSION))/100.
2334              LET PN = REAL.F(CPM.WARNED.FRACT(CAT,PRONE,MISSION))/100.
2335              LET FN = REAL.F(CPM.WARNED.FRACT(CAT,FOX,MISSION))/100.
2336              IF TIME.V*MINUTES.V - UN.LAST.ARTY.ENG(UNIT) > 10.
2337                 LET S1 = REAL.F(CPM.UNWARNED.FRACT(CAT,STAND,MISSION))/100.
2338                 LET P1 = REAL.F(CPM.UNWARNED.FRACT(CAT,PRONE,MISSION))/100.
2339                 LET F1 = REAL.F(CPM.UNWARNED.FRACT(CAT,FOX,MISSION))/100.
2340              ELSE
2341                 LET S1 = SN
2342                 LET P1 = PN
2343                 LET F1 = FN
2344              ALWAYS
2345              CALL FINAL.COVERAGE GIVEN TARGET,MSN,ENVIR,STAND,TE,FUZE------------------------------------------>(188)
2346              YIELDING COVER AND X
2347              IF COVER LT 0.0001
2348                 LET COVER = 0.0
2349              ALWAYS
2350              ADD COVER*(1.0-EXP.F(X))*UN.ENVIR.FRACT(ENVIR) TO FDS1
2351              ADD COVER*(1.0-EXP.F(FM.FIRED.VOLS(MSN)*X))*UN.ENVIR.FRACT(ENVIR)
2352              TO FDSN
2353              CALL FINAL.COVERAGE GIVEN TARGET,MSN,ENVIR,PRONE,TE,FUZE------------------------------------------>(188)
2354              YIELDING COVER AND X
2355              IF COVER LT 0.0001
2356                 LET COVER = 0.0
2357              ALWAYS
2358              ADD COVER*(1.0-EXP.F(X))*UN.ENVIR.FRACT(ENVIR) TO FDP1
2359              ADD COVER*(1.0-EXP.F(FM.FIRED.VOLS(MSN)*X))*UN.ENVIR.FRACT(ENVIR)
2360              TO FDPN
2361              CALL FINAL.COVERAGE GIVEN TARGET,MSN,ENVIR,FOX,TE,FUZE------------------------------------------>(188)
2362              YIELDING COVER AND X
2363              IF COVER LT 0.0001
2364                 LET COVER = 0.0
2365              ALWAYS
2366              ADD COVER*(1.0-EXP.F(X))*UN.ENVIR.FRACT(ENVIR) TO FDF1
2367              ADD COVER*(1.0-EXP.F(FM.FIRED.VOLS(MSN)*X))*UN.ENVIR.FRACT(ENVIR)
2368              TO FDFN
2369           ALWAYS
```

```
2370          ENDLOOP
2371          IF TE = PERSONNEL
2372             IF FM.FIRED.VOLS(MSN) > 1
2373                LET T.FAC = PN - (S1 - SN)
2374                LET X.FAC = SN * (1.0 - FDS1)
2375                IF T.FAC LT 0.
2376                   LET Y.FAC = PN * (1.0 -FDS1)
2377                ELSE
2378                   LET Y.FAC = S1 * (1.0 - FDS1) + T.FAC * (1.0 -FDP1) - X.FAC
2379                ALWAYS
2380                LET Z.FAC = S1 * (1.0 - FDS1) + P1 * (1.0 -FDP1) + F1 * (1.0 - FDF1)
2381                - X.FAC - Y.FAC
2382                LET FRAC.CAS = 1.0 -(((1.0 - FDSN)/(1.0 - FDS1)) * X.FAC
2383                + ((1.0 - FDPN)/(1.0 - FDP1)) * Y.FAC
2384                + ((1.0 - FDFN)/(1.0 - FDF1)) * Z.FAC)
2385             ELSE ''ONLY 1 VOLLEY FIRED
2386                LET FRAC.CAS = (S1*FDS1) + (P1*FDP1) + (F1*FDF1)
2387             ALWAYS
2388          ALWAYS
2389
2390
2391          IF ANALYSIS(4) = TRUE
2392             WRITE FRAC.CAS,COVER AS /,S 6,"FRAC.CAS = ",D(9,6),S 3,"COVER = ",D(9,6),/
2393             USING UNIT 51
2394          ALWAYS
2395       ELSE  ''MISSION IS PGM
2396          IF TR.PGM.STATUS(TARGET) > 1 AND TE.PGM.INDIC(TE) = 0
2397    <-------CYCLE
2398          OTHERWISE
2399          IF TR.PGM.STATUS(TARGET) = 1 AND TE.PGM.INDIC(TE) NE 1
2400    <-------CYCLE
2401          OTHERWISE
2402
2403          IF TR.PGM.STATUS(TARGET) =1
2404             LET XDIFF=UN.X.COORD(TR.TGT.UNIT(TARGET)) -
2405             UN.X.COORD(TR.REP.UNIT(TARGET))
2406             LET YDIFF=UN.Y.COORD(TR.TGT.UNIT(TARGET)) -
2407             UN.Y.COORD(TR.REP.UNIT(TARGET))
2408             LET FO.RANGE = (SQRT.F((XDIFF**2)+(YDIFF**2)))*16.0
2409             IF UN.STATUS(TR.TGT.UNIT(TARGET)) = STATIONARY OR
2410                UN.STATUS(TR.TGT.UNIT(TARGET)) = ADV.TO.WITH OR
2411                UN.STATUS(TR.TGT.UNIT(TARGET)) = STA.TO.WITH
2412                LET MOVEMENT = 2
2413             ELSE
2414                LET MOVEMENT = 0
2415             ALWAYS
2416             IF FO.RANGE GT (DEFILADE.DIST(.TERRAIN)) * 16.0
2417                LET DEFILADE = YES
2418             ELSE
2419                LET DEFILADE = NO
2420             ALWAYS
2421             ''DETERMINE PR(HIT/LASE)
2422             LET J=MOVEMENT + DEFILADE
2423             LET ROW = FO.RANGE/500.0
2424             LET ILOW = TRUNC.F(ROW)
2425             IF ILOW = 0
2426                LET ILOW = 1
2427             ALWAYS
```

```
2428            LET IHI = ILOW + 1
2429            IF IHI > 12
2430               LET IHI = 12
2431               LET ILOW = 12
2432            ALWAYS
2433            LET P.HIT = PGM.HIT(ILOW,J)+FRAC.F(ROW)*(PGM.HIT(IHI,J)-
2434            PGM.HIT(ILOW,J))
2435
2436            IF DEBUG = TRUE
2437               PRINT 1 LINE WITH FO.RANGE,MOVEMENT,DEFILADE,P.HIT,
2438               TARGET    THUS
2439               $#$# FO.RANGE = ****.**, MOVE=*,DEF=*, P.HIT = *.****, TGT *******
2440            ALWAYS
2441
2442            ''DETERMINE PR(KILL/HIT) AND PK
2443            FOR II = 1 TO DIM.F(PGM.LINK(*))
2444            WITH PGM.LINK(II) = UE.ID(EQ)
2445            FIND THE FIRST CASE
2446            IF NONE
2447               PRINT 1 LINE WITH EQ.NAME(UE.ID(EQ)) THUS
2448               ==== ERROR ====  NO PGM PK DATA FOR ******
2449               CALL ERROR.STOP------------------------------------------------------------------------------------------------------>(604)
2450            ALWAYS
2451
2452            LET PK = P.HIT*PGM.KILL(II,J)*ROUND.RELY
2453            ''DETERMINE FRACTION OF CASUALTIES
2454            LET DET.QUANT = NUM.ROUNDS - 1
2455            LET T.EQPS =0
2456            LOOP FOR EACH EQUIP IN UN.EQUIP.LIST(TR.TGT.UNIT(TARGET))
2457            WITH TE.PGM.INDIC(EQ.TE.PTR(UE.ID(EQUIP))) = TRUE
2458            ''THIS COUNTS TOTAL NUMBER OF PGM TARGETS IN UNIT
2459            ''WHETHER THEY ARE DETECTED OR NOT
2460            ADD UE.QUANT(EQUIP) TO T.EQPS
2461            IF T.EQPS LE DET.QUANT
2462               LET T.EQPS = DET.QUANT
2463            ALWAYS
2464            LET N.TGTS=INT.F((DET.QUANT/T.EQPS)*UE.QUANT(EQ))
2465
2466            IF DEBUG = TRUE
2467               PRINT 1 LINE WITH DET.QUANT, T.EQPS, UE.QUANT(EQ)
2468               THUS
2469               $#$# DET.QUANT = ****, T.EQPS = ****, UE.QUANT = ****
2470            ALWAYS
2471
2472            ''DETERMINE NUMBER OF KILLS
2473            LET N.KILLED = 0
2474            LOOP FOR II = 1 TO N.TGTS,
2475            DO
2476               IF RANDOM.F(RN.SEED) LE PK
2477                  ADD 1 TO N.KILLED
2478               ALWAYS
2479            ENDLOOP
2480            IF N.KILLED LT N.TGTS AND EXT.RND=1
2481               LET EXT.RND = 0
2482               IF RANDOM.F(RN.SEED) LE PK
2483                  ADD 1 TO N.KILLED
2484               ALWAYS
2485            ALWAYS
```

```
2486            LET FRAC.CAS=N.KILLED/UE.QUANT(EQ)
2487         ELSE  ''SADARM EVALUATION OF KILLS
2488            IF COUNT = 0
2489    <----------CYCLE
2490            OTHERWISE
2491
2492            ''COMPUTE NUMBER OF TARGETS OF THIS EQUIP AVAILABLE FOR
2493            ''KILLING
2494            CALL FINAL.COVERAGE GIVEN TARGET,MSN,1,0,TE,2----------->(188)
2495            YIELDING COVER AND X
2496            LET AV.TGTS = COVER * UE.QUANT(EQ)
2497            '' COMPUTE NUMBER OF PROJOS FOR THESE TGTS
2498            LET AV.PROJ = FM.N.VOLS(MSN) * TUBES*3.*AV.TGTS/COUNT
2499            IF AV.TGTS LT 1
2500    <----------CYCLE
2501            OTHERWISE
2502
2503            RESERVE SAD.KILL(*) AS AV.TGTS
2504
2505            LOOP FOR III = 1 TO AV.PROJ,
2506            DO
2507               IF RANDOM.F(RN.SEED) LE SDM.SSPK/100
2508                  LET SAD.KILL(RANDI.F(1,AV.TGTS,RN.SEED)) = 1
2509               ALWAYS
2510            ENDLOOP
2511            LET TOTAL.CAS = 0
2512            LOOP FOR III = 1 TO AV.TGTS,
2513            DO
2514               ADD SAD.KILL(III) TO TOT.KILL.SDM
2515               ADD SAD.KILL(III) TO TOTAL.CAS
2516            ENDLOOP
2517
2518            PRINT 1 LINE WITH TOTAL.CAS, EQ.NAME(UE.ID(EQ)) THUS
2519            &&&&& ***** ****** KILLED WITH SADARM
2520
2521            RELEASE SAD.KILL(*)
2522
2523         ALWAYS
2524      ALWAYS
2525
2526      IF DEBUG = TRUE AND
2527         TR.PGM.STATUS(TARGET) GE TRUE
2528         PRINT 1 LINE WITH PK, N.TGTS, N.KILLED, TARGET THUS
2529         $#$# PK = *.****, N.TGTS = ****, N.KILLED = ***, TGT *******
2530      ALWAYS
2531
2532      IF ARTY.ASS > 0
2533         CREATE AN AA.LINK CALLED AAL
2534         LET AA.UE.LINK(AAL) = EQ
2535         LET AA.FRACTION(AAL)= FRAC.CAS
2536         FILE AAL IN AA.SET(ARTY.ASS)
2537      ELSE
2538         LET I = I + 1
2539         IF TR.PGM.STATUS(TARGET) LE TRUE
2540            IF FRAC.CAS > 0.0
2541               LET TOTAL.CAS = BINOMIAL.F(UE.QUANT(EQ),FRAC.CAS,
2542               RN.SEED)
2543               IF TR.PGM.STATUS(TARGET) = TRUE
```

```
2544                  LET TOTAL.CAS = N.KILLED
2545               ALWAYS
2546            ELSE
2547               LET TOTAL.CAS = 0
2548            ALWAYS
2549         ALWAYS
2550         IF ARTY.ASS LE 0  AND TR.PGM.STATUS(TARGET) = TRUE
2551            PRINT 1 LINE WITH TOTAL.CAS, TARGET,
2552            EQ.NAME(UE.ID(EQ))   THUS
2553            $#$# *** KILLED WITH PGM ON TGT ******* EQ ******
2554
2555            ADD TOTAL.CAS TO NUM.KILL.PGM
2556         ALWAYS
2557         LET UE.QUANT( EQ ) = UE.QUANT( EQ ) - TOTAL.CAS
2558
2559         CALL OUTPUT.ATTRITION------------------------------------------------------------------------------------------------------>(611)
2560         GIVEN
2561         UNIT,
2562         TB.HOW.EQ.ID(BY.TYPE(BTRY)),
2563         UE.ID(EQ),
2564         TOTAL.CAS,
2565         "INDIRECT",
2566         MSN
2567
2568         IF UN.PTR(UNIT) > 0 AND UE.CRITICAL.EQUIP.INDIC(EQ) = YES
2569            SUBTRACT TOTAL.CAS FROM MU.CRIT.NO(UN.PTR(UNIT))
2570         ALWAYS
2571         IF KILLER>0 AND VICTIM>0,
2572            ADD TOTAL.CAS TO KV.SCORE(SIDE,KILLER,VICTIM)
2573            IF TR.PGM.STATUS(TARGET) GT TRUE AND
2574               DEBUG = TRUE
2575               PRINT 1 LINE WITH SIDE, KILLER, VICTIM,
2576               KV.SCORE(SIDE, KILLER, VICTIM),
2577               TOTAL.CAS, TARGET THUS
2578               $#$# SIDE **, KILLER ********, VICTIM *******, KV *****, CAS **** TGT *******
2579            ALWAYS
2580         ALWAYS
2581
2582         LET UN.GROUP = TU.CAT(UN.TYPE.UNIT(UNIT))   '' %22JAN80_%RGR
2583         LET GROUPING = CT.GROUP(UN.GROUP)
2584         IF GROUPING = ARTILLERY   '' 2, FA AND MORTAR SECTIONS
2585            LET TGT.BTRY = UN.BTRY.INDEX(UNIT)
2586         THEN IF TGT.BTRY > 0
2587            IF TIME.V - REAL.F(UN.LAST.ARTY.ENG(UNIT))/MINUTES.V GE
2588               REAL.F(TB.SUPPRESS.TIME(BY.TYPE(TGT.BTRY)))/MINUTES.V
2589               IF BY.CUR.FM(TGT.BTRY) GT 0
2590                  LET .FM = BY.CUR.FM(TGT.BTRY)
2591                  IF DEBUG = TRUE
2592                     PRINT 1 LINE WITH .FM THUS
2593                     FIRE MISSION ******  IS STOPPED DUE TO SUPPRESSION
2594                  ALWAYS
2595                  LET FM.N.VOLS(.FM) = FM.FIRED.VOLS(.FM)
2596               ALWAYS
2597            ALWAYS
2598         THEN IF UN.STATUS(UNIT) EQ 3 '' STATIONARY
2599            LET UN.TIME.LAST.MOVE(UNIT) = TIME.V - TIME.BETWEEN.ARTY.MOVE
2600            '' THIS CHANGES THE TIME OF THE UNIT'S LAST MOVE
2601            '' SO THAT IT CAN BE MOVED IMMEDIATELY
```

```
2602            LET FA.BN = BY.BN(TGT.BTRY)    '' HIGHER HEADQUARTERS
2603            CALL FA.BN.MOVEMENT GIVEN FA.BN, 0------------------------------------------------------------------------------------------>( 89)
2604         THEN IF EQ.TE.PTR(TB.HOW.EQ.ID(BY.TYPE(TGT.BTRY))) = TE
2605            AND TOTAL.CAS > 0
2606            LET TOT.HOW.LOSS = 0
2607            LOOP FOR EACH HOW IN BY.HOW.SET(TGT.BTRY)
2608            DO THIS    ''                                                                                                          \1
2609               IF RANDOM.F( RN.SEED ) < FRAC.CAS
2610                  ADD 1 TO TOT.HOW.LOSS
2611                  IF TOT.HOW.LOSS > TOTAL.CAS
2612            <----------LEAVE
2613                  OTHERWISE
2614                  REMOVE HOW FROM BY.HOW.SET(TGT.BTRY)
2615                  DESTROY THIS HOW
2616               ALWAYS
2617            ENDLOOP
2618         THEN IF N.BY.HOW.SET(TGT.BTRY) LT
2619            TB.MIN.HOW(BY.TYPE(TGT.BTRY))
2620            SKIP 1 LINE
2621            PRINT 1 LINE WITH UNIT.NOS(UNIT) THUS
2622            = = = FIRING BATTERY/MORTAR SECTION UNIT ***** IS DECIMATED = = =
2623            SKIP 1 LINE
2624         ALWAYS
2625
2626      ALWAYS
2627      FOR EACH LINK IN UN.SENSOR.LIST(UNIT)
2628      WHEN US.EQ.ID(LINK) = UE.ID(EQ)
2629      CALL ATTRIT.SENSOR----------------------------------------------------------------------------------------------------------->(159)
2630      GIVEN
2631      LINK,
2632      FRAC.CAS
2633      ''
2634      ''
2635      '' ATCAL WRITES FOR THE THEATER LEVEL ARTY CALCULATIONS...
2636      ''    %MAY83_%HWJ FOR ATCAL
2637      ''
2638      ''
2639      IF TR.PGM.STATUS(TARGET) = TRUE
2640         LET TUBES = 1
2641         LET FRAC.CAS = PK
2642      ALWAYS
2643      IF TR.PGM.STATUS(TARGET) GT TRUE
2644         LET FRAC.CAS = TOTAL.CAS/(UE.QUANT(EQ) + TOTAL.CAS)
2645      ALWAYS
2646      ''IF UN.COLOR(BY.UNIT(BTRY)) = RED                                                                                           \DEBUG
2647         ''LET SIDE = RED
2648         ''WRITE TIME.V,SIDE,FM.FIRED.VOLS(MSN),TUBES,TOTAL.CAS,
2649         ''FRAC.CAS,EQ.NAME(UE.ID(EQ)),TE.NAME(TE),EQ.TE.PTR(UE.ID(EQ)),
2650         ''FM.TM.CLASS(MSN),ROUND.NAME,EQ.NAME(TB.HOW.EQ.ID(TB)),
2651         ''TB.NAME(TB),UN.PCT.OPEN,UN.PCT.WOOD,UN.PCT.TOWN,COVER,
2652         ''UE.QUANT(EQ) AS
2653         ''S 1, D(7,4), S 3, I 5, S 3, I 5, S 3, I 5, S 3,
2654         ''I 5, S 3, D(10,8), S 3, T 6, S 1, T 6, S 3, I 5, S 3, /,                                                                \TEXT
2655         ''T 4, S 3, T 6, S 3, T 6, S 3, T 6, S 3, D(5,3), S 3,                                                                    \TEXT
2656         ''D(5,3), S 3, D(5,3), S 3, D(6,4), S 2 , I 5 ,/ USING UNIT 16
2657      ''ELSE
2658         ''LET SIDE = BLUE
2659         ''WRITE TIME.V,SIDE,FM.FIRED.VOLS(MSN),TUBES,TOTAL.CAS,
```

```
2660          ''FRAC.CAS,EQ.NAME(UE.ID(EQ)),TE.NAME(TE),EQ.TE.PTR(UE.ID(EQ)),
2661          ''FM.TM.CLASS(MSN),ROUND.NAME,EQ.NAME(TB.HOW.EQ.ID(TB)),
2662          ''TB.NAME(TB),UN.PCT.OPEN,UN.PCT.WOOD,UN.PCT.TOWN,COVER,
2663          ''UE.QUANT(EQ) AS
2664          ''S 1, D(7,4), S 3, I 5, S 3, I 5, S 3, I 5, S 3,
2665          ''I 5, S 3, D(10,8), S 3, T 6, S 1, T 6, S 3, I 5, S 3, /.                    \TEXT
2666          ''T 4, S 3, T 6, S 3, T 6, S 3, T 6, S 3, D(5,3), S 3,                        \TEXT
2667          ''D(5,3), S 3, D(5,3), S 3, D(6,4), S 2 , I 5,/ USING UNIT 16
2668       ''ENDIF
2669    ENDLOOP
2670
2671    IF UN.COLOR(BY.UNIT(BTRY)) = RED    ''  %WARF REPORT DATA
2672       LET SIDE = RED
2673       WRITE TIME.V, SIDE, FIRE.PLAN, RANGE, FM.FIRED.VOLS(MSN),
2674       BTRY, BY.UNIT(BTRY), TB, TUBES, UNIT, TU, CAT,
2675       FM.TM.CLASS(MSN), MUNITION, ROUND.NAME, FUZE
2676       AS S 1, D(7,4), S 2, I 5, S 8, I 5, S 4, I 5, S 2, I 5,
2677       S 2, I 5, S 3, I 5, S 2, I 5, S 2, I 5, S 2, I 5,
2678       S 2, I 3, S 2, I 2, S 3, T 4, S 2, I 3, S 3, T 6, ''                          \TEXT
2679       S 2, I 2 USING UNIT 4
2680       WRITE AS / USING UNIT 4
2681       WRITE RNG.HACK, ROUND.RELY, NUM.ROUNDS, ST, QUANT,
2682       COVER, PI.RAD.2, MSN, TOTAL.CAS, MISSION
2683       AS  S 7, I 5, S 5, D(5,3), S 2,   I 5,  S 2, I 5,
2684       S 2, I 5, S 2, D(6,2), S 6, D(9,2), S 3, I 6,
2685       S 2, I 5,  S 2, I 5  USING UNIT 4
2686       WRITE AS / USING UNIT 4
2687    ELSE   ''SIDE = BLUE , DO NOT WRITE A REPORT
2688       LET SIDE = BLUE
2689    ALWAYS
2690
2691    ''OUTPUT RECORD FOR SUMMARY REPORT
2692    WRITE SIDE, TB, TB.NAME(TB), TU.LEVEL(TU), TE.NAME(TU.PRIN.TE(TU)),
2693    UN.PCT.OPEN, UN.PCT.WOOD, UN.PCT.TOWN, M.FR.UNIT.SET(UNIT),
2694    TR.MOVE(TARGET), RANGE*16, ROUND.NAME, FUZE, FM.FIRED.VOLS(MSN),
2695    NUM.ROUNDS, TIME.V AS
2696    I 2, I 3, S 1, T 6, S 1, T 6, S 1, 3 D(4,2), I 2, I 2, ''                        \TEXT
2697    I 8, S 1, T 6, I 2, I 3, I 4, D(6,2), / USING UNIT 7  ''                        \TEXT
2698
2699    IF ARTY.ASS > 0
2700       ACTIVATE_THE ARTY.ASSESS CALLED ARTY.ASS NOW------------------------------------------------------------------------------------>(435)
2701    ALWAYS
2702
2703    IF FIRE.PLAN = TRUE
2704       LET TR.TGT.UNIT(TARGET)=0
2705    ALWAYS
2706    LET UN.LAST.ARTY.ENG(UNIT)=TIME.V*MINUTES.V
2707
2708    IF UN.STATUS(UNIT) = STATIONARY AND
2709       CT.GROUP(TU.CAT(UN.TYPE.UNIT(UNIT))) = ARTILLERY
2710       FOR EVERY LINK IN TB.TM.LIST(BY.TYPE(BTRY))
2711       WITH TB.TM.CLASS(LINK) = "FASCAM" AND
2712       FMM.MAX.RANGE(TB.TM(LINK)) GT FM.RANGE(MSN)
2713       FIND THE FIRST CASE
2714       IF FOUND
2715          ''THE LAST VOLLEY IS ASSUMED TO HAVE FASCAM TO SUPPRESS
2716          ''THE ARTY UNIT
2717          LET MAX.SUPP = TB.MX.FASCAM.SUPP(BY.TYPE(UN.BTRY.INDEX(UNIT)))
```

```
2718          LET MIN.SUPP = TB.MN.FASCAM.SUPP(BY.TYPE(UN.BTRY.INDEX(UNIT)))
2719          IF MAX.SUPP GT 0.
2720             LET END.SUPP = TIME.V * MINUTES.V
2721             + RANDI.F(MIN.SUPP, MAX.SUPP, RN.SEED)
2722             LET BY.STOP.FASCAM.SUPP(UN.BTRY.INDEX(UNIT)) =
2723             MAX.F(END.SUPP,
2724             BY.STOP.FASCAM.SUPP(UN.BTRY.INDEX(UNIT)))
2725          ALWAYS
2726       ALWAYS
2727    ALWAYS
2728
2729    RELEASE UN.ENVIR.FRACT(*)
2730
2731 <--EXITROUTINE
2732 ENDROUTINE
```

```
2733 ROUTINE CLEAN.UP.FIRE.MISSIONS ''                                          IF46
2734     GIVEN FM
2735
2736     DEFINE FM, BTRY, TGT, SENSOR AS INTEGER VARIABLES  ''                  CHG\19
2737
2738     ''CLEANUP BATTERY QUEUES
2739     LET BTRY = FM.BTRY(FM)
2740     IF FM IS IN THE BY.FM.QUEUE
2741        REMOVE THE FM FROM THE BY.FM.QUEUE(BTRY)
2742        WRITE AS "FM REMOVED FROM BY.FM.QUEUE", /
2743     ALWAYS
2744
2745     IF FM IS IN THE BY.SCHD.LIST
2746        REMOVE THE FM FROM THE BY.SCHD.LIST(BTRY)
2747        WRITE AS "FM REMOVED FROM BY.SCHD.LIST", /
2748     ALWAYS
2749
2750     ''CLEANUP TARGET REPORT LISTS
2751     LET TGT = FM.TGT(FM)
2752     IF TGT IS ZERO
2753        WRITE AS "TGT IS ZERO", /
2754        LIST ATTRIBUTES OF FIRE.MISSION CALLED FM
2755        WRITE AS /
2756        TRACE
2757 <-----STOP
2758     OTHERWISE
2759     IF FM IS IN THE TR.FM.LIST
2760        REMOVE THE FM FROM THE TR.FM.LIST
2761        WRITE AS "FM REMOVED FROM TR.FM.LIST", /
2762     ALWAYS
2763
2764     ''CLEANUP FORWARD OBSERVER LIST
2765     LET SENSOR = TR.SENSOR.ID(TGT)
2766     IF SENSOR IS ZERO
2767        WRITE AS "SENSOR IS ZERO", /
2768        LIST ATTRIBUTES OF FIRE.MISSION CALLED FM
2769        LIST ATTRIBUTES OF TARGET.REPORT CALLED TGT
2770        WRITE AS /
2771        TRACE
2772 <-----STOP
2773     OTHERWISE
2774     IF FM IS IN THE FO.CUR.FM.LIST,
2775        REMOVE THE FM FROM THE FO.CUR.FM.LIST(SENSOR)
2776        WRITE AS "FM REMOVED FROM FO.CUR.FM.LIST", /
2777     ALWAYS
2778
2779 <--RETURN
2780 END
```

AIR ROUTINES

```
2787 ROUTINE AC.BOMB.EFFECTS                                                    ''                                  A001
2788     GIVEN
2789     .AC.ATK.TGT,
2790     .WPN,
2791     .ROW
2792
2793     ADD 1 TO ANAL.CTR(97,1)  ''                                                                              \DYN_ANAL
2794     IF TACAIR.DEBUG = 1
2795        PRINT 1 LINE WITH TIME.V, .AC.ATK.TGT THUS
2796        = = =  AC.BOMB.EFFECTS CALLED AT  ***.*** BY ******
2797     ALWAYS
2798     NORMALLY MODE IS INTEGER
2799     DEFINE .T, .T.SQ, .SIGMA, .COVERAGE, .X, .THETA1, .THETA2,
2800     .RAD, .RAD.SQ, .AREA,
2801     .FRAC.CAS AS REAL VARIABLES
2802     DEFINE .ENVIR.FRAC AS A 1-DIM REAL ARRAY
2803     DEFINE .SCOREBOARD AS A 2-DIM INTEGER ARRAY  ''                                                          \VAX
2804
2805     RESERVE .ENVIR.FRAC AS 3
2806
2807     LET .TGT = CMSN.TGT.UNIT(AAT.CMSN(.AC.ATK.TGT))
2808     LET .MUN = WPN.AC.MUNS(.WPN)
2809     LET .TYPE.WPN = WPN.ID(.WPN)
2810     LET .CMSN = AAT.CMSN(.AC.ATK.TGT)
2811     LET .FIRER.EQ = ACT.EQUIP.ID(CMSN.AC.TYPE(.CMSN))
2812     LET .SIDE = CMSN.SIDE(.CMSN)
2813     LET .SCOREBOARD(*,*) = CMSN.SCOREBOARD(.CMSN)
2814     ''UPDATE AMMO EXPENDITURES.
2815     LET .ROUNDS = TW.RATE.OF.FIRE(.TYPE.WPN)
2816     ADD .ROUNDS TO STW.RND.FIRED(.SIDE, .TYPE.WPN)
2817     ADD .ROUNDS * TW.RND.WEIGHT(.TYPE.WPN)
2818     TO KV.AMMO.CONSUMED(.SIDE, EQ.KV.ID(.FIRER.EQ))
2819
2820     IF .MUN GT 0
2821        ''DETERMINE THE COVERAGE BY THE BOMBS
2822        CALL UNIT.ENVIR------------------------------------------------------------------------------------------>(242)
2823        GIVEN
2824        .TGT
2825        YIELDING
2826        .ENVIR.FRAC(1),
2827        .ENVIR.FRAC(2),
2828        .ENVIR.FRAC(3)
2829        ''DETERMINE THE COVERAGE.
2830        LET .TGT.RADIUS = TU.RADIUS(UN.TYPE.UNIT(.TGT))
2831        LET .R1 = MIN.F(AM.RADIUS(.MUN), .TGT.RADIUS)
2832        LET .R2 = MAX.F(AM.RADIUS(.MUN), .TGT.RADIUS)
2833        LET .R1.SQ = .R1**2
2834        LET .R2.SQ = .R2**2
2835        LET .T = .R1 / .R2
2836        LET .T.SQ = .T ** 2
2837        LET .SIGMA = ATM.DELIV.CEP(CMSN.AC.TYPE(.CMSN), .MUN)
2838        IF VISIBILITY LT SD.POOR.FLY.VIS(.SIDE)
2839           LET .SIGMA = .SIGMA /
2840           (ACT.WEATHER.DEGRADE(CMSN.AC.TYPE(.CMSN)) / 100)
2841        ALWAYS
2842        ''UTILIZE NORMAL.F FUNCTION IN THE NEXT 2 DRAWS ------------------------------------------------------ \1>(641)
2843        LET .DELTAX = NORMAL.F(0.0, .SIGMA, RN.SEED)
2844        LET .DELTAY = NORMAL.F(0.0, .SIGMA, RN.SEED)
```

```
2845        LET .RAD.SQ  = .DELTAX **2 + .DELTAY ** 2
2846        LET .RAD = SQRT.F(.RAD SQ)
2847        IF .RAD GT .R1 + .R2
2848           ''NO COVERAGE OF TARGET
2849  <---------RETURN
2850        OTHERWISE
2851
2852        IF .RAD LE .R2 - .R1
2853           LET .COVERAGE = ( .R1 / .TGT.RADIUS) ** 2
2854        ELSE
2855           LET .THETA1 = ARCCOS.F((.RAD.SQ + .R1.SQ - .R2.SQ)
2856           / (2. * .RAD * .R1))
2857           LET .THETA2 = ARCCOS.F((.RAD.SQ - .R1.SQ + .R2.SQ)
2858           / (2. * .RAD * .R2))
2859           IF .RAD.SQ LE .R2.SQ + .R1.SQ
2860              IF .R2 = .TGT.RADIUS
2861                 LET .COVERAGE = (.THETA2 + .T.SQ * (PI.C - .THETA1)
2862                 - .T * SIN.F(.THETA1 - .THETA2)) / PI.C
2863              ELSE
2864                 LET .COVERAGE = ((.THETA2 / .T.SQ) + (PI.C - .THETA1)
2865                 - SIN.F(.THETA1 - .THETA2)/.T) / PI.C
2866              ALWAYS
2867           ELSE
2868              IF .R2 = .TGT.RADIUS
2869                 LET .COVERAGE = (.THETA2 + .T.SQ * .THETA1
2870                 -.T * SIN.F(.THETA1 + .THETA2)) / PI.C
2871              ELSE
2872                 LET .COVERAGE = ((.THETA2 / .T.SQ) + .THETA1
2873                 - SIN.F(.THETA1 - .THETA2)/.T) / PI.C
2874              ALWAYS
2875           ALWAYS
2876        ALWAYS
2877        LET .AREA = PI.C * AM.RADIUS(.MUN) ** 2
2878        IF TACAIR.DEBUG = 1
2879           LIST .AREA, .MUN, AM.RADIUS(.MUN)
2880        ALWAYS
2881     ALWAYS
2882
2883     ''DETERMINE THE LOSSES
2884     LOOP FOR EACH .LINK IN UN.EQUIP.LIST(.TGT)
2885     DO
2886        ADD 1 TO .COLUMN
2887        IF UE.QUANT(.LINK) LE 0
2888  <------CYCLE
2889        OTHERWISE
2890        LET .TE = EQ.TE.PTR(UE.ID(.LINK))
2891        LET .KILLED = 0
2892        LET .FRAC.CAS = 0.0
2893        IF .MUN GT 0
2894           ''BOMBS
2895           LOOP FOR EACH ENVIRONMENT CALLED .EN
2896           DO
2897              IF .TE = 1 ''PERSONNEL
2898                 LOOP FOR EACH POSTURE CALLED .P
2899                 DO
2900                    LET .X = .ROUNDS * AMEP.LA.PERS(.MUN,.EN, .P)
2901                    *(AM.RELY(.MUN)/100)* .COVERAGE/.AREA
2902                    LET .FRAC.CAS = .FRAC.CAS
```

```
2903                  + (1.0 - EXP.F(-.X)) * .ENVIR.FRAC(.EN)
2904                  * CPM.WARNED.FRACT(
2905                  TU.CAT(UN.TYPE.UNIT(.TGT)),
2906                  .P, UN.MISSION(.TGT))
2907               ENDLOOP
2908            ELSE
2909               ''EQUIPMENT OTHER THAN PERSONNEL
2910               LET .X = .ROUNDS * AMET.LA.EQUIP(.MUN,.EN, .TE)
2911               *(AM.RELY(.MUN)/100)* .COVERAGE / .AREA
2912               LET .FRAC.CAS = .FRAC.CAS
2913               + (1.0 - EXP.F(-.X)) * .ENVIR.FRAC(.EN)
2914            ALWAYS
2915         ENDLOOP
2916      ELSE
2917         ''MINES
2918         LET .KILLED = UNIFORM.F(REAL.F(TE.MIN.MF.LOSS(.TE)),
2919         REAL.F(TE.MAX.MF.LOSS(.TE)), RN.SEED)
2920         LET .KILLED = MIN.F(.KILLED, UE.QUANT(.LINK))
2921         LET .FRAC.CAS = .KILLED / UE.QUANT(.LINK)
2922      ALWAYS
2923      IF .FRAC.CAS LE 0.00001
2924   <------CYCLE
2925      OTHERWISE
2926
2927      IF SO.LIST(.LINK) IS EMPTY
2928         IF .MUN GT 0
2929            LET .KILLED = BINOMIAL.F(UE.QUANT(.LINK),
2930            .FRAC.CAS, RN.SEED)
2931         ALWAYS
2932         SUBTRACT .KILLED FROM UE.QUANT(.LINK)
2933      ELSE
2934         LOOP FOR EVERY .SO IN SO.LIST(.LINK)
2935         WITH .FRAC.CAS GT RANDOM.F(RN.SEED) AND
2936         DROP.DEAD.INDICATOR(.SO) NE YES
2937         DO
2938            LET DROP.DEAD.INDICATOR(.SO) = YES
2939            INTERRUPT SHOOT.OUT CALLED .SO------------------------------------------------------------------------>(493)
2940            REACTIVATE_THE SHOOT.OUT CALLED .SO------------------------------------------------------------------->(493)
2941            IN 1 / 60 HOURS
2942            IF .MUN GT 0
2943               ADD 1 TO .KILLED
2944            ALWAYS
2945            SUBTRACT 1 FROM UE.QUANT(.LINK)
2946         ENDLOOP
2947      ALWAYS
2948      IF UN.PTR(.TGT) GT 0 AND
2949         UE.CRITICAL.EQUIP.INDIC(.LINK) = TRUE
2950         SUBTRACT .KILLED FROM MU.CRIT.NO(UN.PTR(.TGT))
2951      ALWAYS
2952
2953      IF CT.GROUP(TU.CAT(UN.TYPE.UNIT(.TGT))) = ARTILLERY
2954         LET .TGT.BTRY = UN.BTRY.INDEX(.TGT)
2955      THEN IF .TGT.BTRY GT 0 AND
2956         EQ.TE.PTR(TB.HOW.EQ.ID(BY.TYPE(.TGT.BTRY))) = .TE
2957         LET .HOW.LOSS = BINOMIAL.F(N.BY.HOW.SET(.TGT.BTRY),
2958         .FRAC.CAS, RN.SEED)
2959         LOOP FOR .I = 1 TO .HOW.LOSS
2960         DO
```

```
2961              REMOVE THE FIRST HOW FROM BY.HOW.SET(.TGT.BTRY)
2962              DESTROY THIS HOW
2963           ENDLOOP
2964           IF N.BY.HOW.SET(.TGT.BTRY) LT TB.MIN.HOW(BY.TYPE(.TGT.BTRY))
2965              LET UN.TIME.LAST.MOVE(.TGT) = TIME.V -
2966              TIME.BETWEEN.ARTY.MOVE
2967              CALL FA.BN.MOVEMENT------------------------------------------------------------------------------------------>( 89)
2968              GIVEN
2969              BY.BN(.TGT.BTRY)
2970           ALWAYS
2971        ALWAYS
2972        FOR EACH US.LINK IN UN.SENSOR.LIST(.TGT)
2973        WITH US.EQ.ID(US.LINK) = UE.ID(.LINK)
2974        CALL ATTRIT.SENSOR------------------------------------------------------------------------------------------>(159)
2975        GIVEN
2976        US.LINK,
2977        .FRAC.CAS
2978        ADD .KILLED TO KV.SCORE(.SIDE,
2979        EQ.KV.ID(.FIRER.EQ), EQ.KV.ID(UE.ID(.LINK)))
2980        ADD .KILLED TO .SCOREBOARD(.ROW, .COLUMN)
2981     ENDLOOP
2982
2983     IF WPN.AC.MUNS(.WPN) LT 0
2984        CALL MINE.DELAY---------------------------------------------------------------------------------------------->(103)
2985        GIVEN
2986        .TGT,
2987        UN.X.COORD(.TGT),
2988        UN.Y.COORD(.TGT),
2989        4
2990     ALWAYS
2991
2992     RELEASE .ENVIR.FRAC
2993
2994 <--RETURN
2995 END
```

```
2996 ROUTINE AC.DF.EFFECTS                                                    ''                A002
2997     GIVEN
2998     .AC.ATK.TGT,
2999     .WPN,
3000     .ROW,
3001     .NUM.FIRES.PER.PASS
3002
3003
3004     ADD 1 TO ANAL.CTR(98,1)  ''                                                            \DYN_ANAL
3005     IF TACAIR.DEBUG = 1
3006        PRINT 1 LINE WITH TIME.V, .AC.ATK.TGT THUS
3007        = = =  AC.DF.EFFECTS CALLED AT  ***.*** BY ******
3008     ALWAYS
3009     NORMALLY MODE IS INTEGER
3010     DEFINE .PK, .DELTA, .ORDINATE AS REAL VARIABLES
3011     DEFINE .SCOREBOARD AS A 2-DIM INTEGER ARRAY  ''                                        \VAX
3012     ''CONSIDERS ATTACKING AIRCRAFT SHOOTING MORE THAN ONCE PER PASS
3013     LOOP FOR .FIRE = 1 TO .NUM.FIRES.PER.PASS
3014     DO  ''                                                                                 \1
3015
3016        ''SELECT A TARGET BASED ON ITS AREA. PERSONNEL
3017        ''ARE NOT FIRED ON WITH MISSILES.
3018        LOOP FOR EACH .AATT OF AATT.LIST(.AC.ATK.TGT)
3019        DO
3020           LET .PROJ.AREA = UE.QUANT(AATT.UE.LINK(.AATT))
3021           * TE.PROJECTED.AREA(EQ.TE.PTR(UE.ID(AATT.UE.LINK(.AATT))))
3022           ADD .PROJ.AREA TO .TOTAL.AREA
3023           LET AATT.CUM.AREA(.AATT) = .TOTAL.AREA
3024        ENDLOOP
3025
3026        IF .TOTAL.AREA LE 1
3027           PRINT 1 LINE WITH .TOTAL.AREA THUS
3028           ↑↑↑↑   ↑↑↑↑    ↑↑↑↑   NO EQUIPMENT IN TARGET UNIT  TOTAL AREA = ***
3029 <---------RETURN
3030
3031        OTHERWISE
3032
3033        LET .SCORE = RANDI.F(1, .TOTAL.AREA, RN.SEED)
3034        FOR EACH .AATT OF AATT.LIST (.AC.ATK.TGT)
3035        WITH AATT.CUM.AREA(.AATT) GE .SCORE
3036        FIND THE FIRST CASE
3037        IF NONE
3038           TRACE
3039 <---------STOP
3040        OTHERWISE
3041
3042        ''FIRE ON THE SELECTED EQ.
3043        LET .TYPE.WPN = WPN.ID(.WPN)
3044        LET .TGT = CMSN.TGT.UNIT(AAT.CMSN(.AC.ATK.TGT))
3045        LET .TGT.LINK = AATT.UE.LINK(.AATT)
3046        LET .TGT.EQ = UE.ID(.TGT.LINK)
3047        LET .FIRER.EQ = ACT.EQUIP.ID(CMSN.AC.TYPE(AAT.CMSN(.AC.ATK.TGT)))
3048        LET .SIDE = CMSN.SIDE(AAT.CMSN(.AC.ATK.TGT))
3049        ADD TW.RATE.OF.FIRE(.TYPE.WPN)*WPN.QUANTITY(.WPN) TO
3050        STW.RND.FIRED(.SIDE, .TYPE.WPN)
3051        ADD TW.RATE.OF.FIRE(.TYPE.WPN)  * TW.RND.WEIGHT(.TYPE.WPN)*
3052        WPN.QUANTITY(.WPN) TO KV.AMMO.CONSUMED(.SIDE, EQ.KV.ID(
3053        .FIRER.EQ))
```

```
3054        IF .SIDE = BLUE
3055           ADD TW.RATE.OF.FIRE(.TYPE.WPN)*WPN.QUANTITY(.WPN) TO
3056           STY.BLUE.EXP(.TYPE.WPN,   .TGT.EQ - N.BLUE.TYPE.EQP)
3057        ELSE
3058           ADD TW.RATE.OF.FIRE(.TYPE.WPN)*WPN.QUANTITY(.WPN) TO
3059           STY.RED.EXP(.TYPE.WPN - N.B.WPN.TYPE, .TGT.EQ)
3060        ALWAYS
3061
3062        LET .PK.PTR = PK.POINTER(TW.PK.PTR(.TYPE.WPN), EQUIP.PK.PTR(.TGT.EQ))
3063        IF .PK.PTR = 0 OR
3064           TW.MIN.RANGE(.TYPE.WPN) GT AAT.RANGE(.AC.ATK.TGT) OR
3065           TW.MAX.RANGE(.TYPE.WPN) LT AAT.RANGE(.AC.ATK.TGT)
3066 <---------RETURN
3067        OTHERWISE
3068
3069        ''INTERPOLATE THE PK BETWEEN GIVEN VALUES
3070        LET .DELTA = (TW.MAX.RANGE (.TYPE.WPN) - TW.MIN.RANGE(.TYPE.WPN))/10
3071        LET .ORDINATE = (AAT.RANGE(.AC.ATK.TGT) - TW.MIN.RANGE(.TYPE.WPN))/
3072        .DELTA
3073        LET .LOW.POINT = TRUNC.F(.ORDINATE)
3074        LET .HIGH.POINT = .LOW.POINT + 1
3075        LET .DELTA = .ORDINATE - .LOW.POINT
3076        IF .DELTA = 0. AND .LOW.POINT GT 0
3077           LET .PK = 2. * PK.PROB(.PK.PTR, .LOW.POINT)/100.
3078        ELSE
3079
3080           FOR EVERY PK.BAND
3081           WITH PK.BAND.RNG(PK.BAND) = .LOW.POINT
3082           FIND THE FIRST CASE
3083           IF NONE
3084              TRACE
3085 <-------------STOP
3086           OTHERWISE
3087           LET .LPK = PK.PROB(.PK.PTR, PK.BAND)
3088
3089           FOR EVERY PK.BAND
3090           WITH PK.BAND.RNG(PK.BAND) = .HIGH.POINT
3091           FIND THE FIRST CASE
3092           IF NONE
3093              TRACE
3094 <-------------STOP
3095           OTHERWISE
3096           LET .HPK = PK.PROB(.PK.PTR, PK.BAND)
3097
3098           LET .PK = 2. * (.LPK + .DELTA * (.HPK - .LPK)) /  100.
3099           ''THE 100 CONVERTS .PK FROM A PERCENTAGE TO A FRACTION.
3100           ''THE 2 CONVERTS .PK FROM A RANGE OF 0 TO 50 TO A
3101           ''RANGE OF 0 TO 100 (SEE PK.INPUT).
3102        ALWAYS
3103
3104        IF VISIBILITY LT SD.POOR.FLY.VIS(.SIDE)
3105           LET .PK = .PK * ACT.WEATHER.DEGRADE(CMSN.AC.TYPE
3106           (AAT.CMSN(.AC.ATK.TGT))) / 100
3107        ALWAYS
3108        LET .PK = 1. - (1. - .PK)**(TW.RATE.OF.FIRE(.TYPE.WPN)*
3109        WPN.QUANTITY(.WPN))
3110        IF RANDOM.F(RN.SEED) LT .PK AND
3111           UE.QUANT(.TGT.LINK) GT 0
```

```
3112          ''THE TARGET WAS HIT.
3113          IF SO.LIST(.TGT.LINK) IS EMPTY
3114             LET .KILLED = YES
3115          ELSE
3116             LET .VICTIM = RANDI.F(1, UE.QUANT(.TGT.LINK),RN.SEED)
3117             LET .N = 1
3118             FOR EVERY .SO OF SO.LIST(.TGT.LINK)
3119             UNTIL .N = .VICTIM  OR .N  = N.SO.LIST(.TGT.LINK)
3120             ADD 1 TO .N
3121             IF DROP.DEAD.INDICATOR(.SO) NE YES
3122                LET DROP.DEAD.INDICATOR(.SO) = YES
3123                INTERRUPT SHOOT.OUT CALLED .SO------------------------------------------------------------------------------>(493)
3124                REACTIVATE_THE SHOOT.OUT CALLED .SO------------------------------------------------------------------------>(493)
3125                IN 1/60 HOURS
3126                LET .KILLED = YES
3127             ALWAYS
3128          ALWAYS
3129          IF .KILLED = YES
3130             SUBTRACT 1 FROM UE.QUANT(.TGT.LINK)
3131             IF UN.PTR(.TGT) GT 0 AND
3132                UE.CRITICAL.EQUIP.INDIC(.TGT.LINK) = TRUE
3133                SUBTRACT 1 FROM MU.CRIT.NO(UN.PTR(.TGT))
3134             ALWAYS
3135             IF CT.GROUP(TU.CAT(UN.TYPE.UNIT(.TGT))) = ARTILLERY
3136                LET .TGT.BTRY = UN.BTRY.INDEX(.TGT)
3137             THEN IF .TGT.BTRY GT 0 AND
3138                EQ.TE.PTR(TB.HOW.EQ.ID(BY.TYPE(.TGT.BTRY))) =
3139                EQ.TE.PTR(.TGT.EQ) AND
3140                BY.HOW.SET(.TGT.BTRY) IS NOT EMPTY
3141                REMOVE THE FIRST HOW FROM BY.HOW.SET(.TGT.BTRY)
3142                DESTROY THIS HOW
3143                IF N.BY.HOW.SET(.TGT.BTRY) LT TB.MIN.HOW(BY.TYPE
3144                   (.TGT.BTRY))
3145                   LET UN.TIME.LAST.MOVE(.TGT) = TIME.V -
3146                   TIME.BETWEEN.ARTY.MOVE
3147                   CALL FA.BN.MOVEMENT------------------------------------------------------------------------------------>( 89)
3148                   GIVEN
3149                   BY.BN(.TGT.BTRY)
3150                ALWAYS
3151             ALWAYS
3152
3153             ADD 1 TO KV.SCORE (.SIDE,
3154             EQ.KV.ID(.FIRER.EQ), EQ.KV.ID(.TGT.EQ))
3155             LET .SCOREBOARD(*,*) = CMSN.SCOREBOARD(AAT.CMSN(.AC.ATK.TGT))
3156             LET .COLUMN = 1
3157             FOR EVERY UE.LINK IN UN.EQUIP.LIST(.TGT)
3158             UNTIL UE.LINK = .TGT.LINK
3159             ADD 1 TO .COLUMN
3160             ADD 1 TO .SCOREBOARD(.ROW, .COLUMN)
3161          ALWAYS
3162       ALWAYS
3163    ENDLOOP
3164
3165 <--RETURN
3166 END
```

```
3167 ROUTINE CAS.EVAL                                                          A003
3168    GIVEN
3169     .BATTLE
3170
3171    ADD 1 TO ANAL.CTR(99,1)                                                \DYN_ANAL
3172    NORMALLY MODE IS INTEGER
3173    DEFINE .RATIO AS A REAL VARIABLE
3174
3175    IF N.AC.TYPE = 0
3176       ''AIRCRAFT NOT BEING MODELED OR ALREADY REQUESTED.
3177 <------RETURN
3178    OTHERWISE
3179
3180
3181    ''COUNT THE TANKS AND CRITICAL EQUIPMENT ON EACH SIDE.
3182    LOOP FOR EACH .FORCE IN BTL.FORCE.SET(.BATTLE)
3183    DO
3184       LET .TANK.TE = SM.TANK.TE(FR.SIDE(.FORCE),FR.MISSION(.FORCE))
3185       LET .MAX.CE = 0
3186       LET .MAX.UNIT = 0
3187       LET .TK.COUNT = 0
3188       LET .CE.COUNT = 0
3189       LOOP FOR EACH .UNIT IN FR.UNIT.SET(.FORCE)
3190       DO
3191          LET .UNIT.CE = 0
3192          LOOP FOR EACH .LINK IN UN.EQUIP.LIST(.UNIT)
3193          DO
3194             IF EQ.TE.PTR(UE.ID(.LINK)) = .TANK.TE
3195                ADD UE.QUANT(.LINK) TO .TK.COUNT
3196             ALWAYS
3197             IF UE.CRITICAL.EQUIP.INDIC(.LINK) = YES
3198                ADD UE.QUANT(.LINK) TO .CE.COUNT
3199                ADD UE.QUANT(.LINK) TO .UNIT.CE
3200             ALWAYS
3201          ENDLOOP
3202          IF .UNIT.CE GT .MAX.CE
3203             LET .MAX.CE = .UNIT.CE
3204             LET .MAX.UNIT = .UNIT
3205          ALWAYS
3206       ENDLOOP
3207       IF FR.SIDE(.FORCE) = BLUE
3208          LET .BLUE.TKS = .TK.COUNT
3209          LET .BLUE.CEQ = .CE.COUNT
3210          LET .BLUE.TGT = .MAX.UNIT
3211          LET .BLUE.FORCE = .FORCE
3212       ELSE
3213          LET .RED.TKS = .TK.COUNT
3214          LET .RED.CEQ = .CE.COUNT
3215          LET .RED.TGT = .MAX.UNIT
3216          LET .RED.FORCE = .FORCE
3217       ALWAYS
3218    ENDLOOP
3219    ''WHEN FR.CAS.INDIC = YES THE FORCE IS ALREADY TARGETED.                \1
3220    LOOP FOR EACH .FORCE IN BTL.FORCE.SET(.BATTLE)
3221    UNLESS FR.CAS.INDIC(.FORCE) = YES
3222    DO
3223       LET .ENEMY = FR.SIDE(.FORCE)
3224       IF .ENEMY = BLUE
```

```
3225          LET .SIDE = RED
3226          LET .MISSION = FR.MISSION(.RED.FORCE)
3227       ELSE
3228          LET .SIDE = BLUE
3229          LET .MISSION = FR.MISSION(.BLUE.FORCE)
3230       ALWAYS
3231       IF .SIDE = BLUE
3232          IF .RED.TKS = 0
3233             IF .BLUE.TKS GT 0
3234                LET .RATIO = RINF.C
3235             ELSE
3236                LET .RATIO = 1.
3237             ALWAYS
3238          ELSE
3239             LET .RATIO = .BLUE.TKS / .RED.TKS
3240          ALWAYS
3241          LET .CEQ = .RED.CEQ
3242          LET .TARGET = .RED.TGT
3243       ELSE
3244          IF .BLUE.TKS = 0
3245             IF .RED.TKS GT 0
3246                LET .RATIO = RINF.C
3247             ELSE
3248                LET .RATIO = 1.
3249             ALWAYS
3250          ELSE
3251             LET .RATIO = .RED.TKS / .BLUE.TKS
3252          ALWAYS
3253          LET .CEQ = .BLUE.CEQ
3254          LET .TARGET = .BLUE.TGT
3255       ALWAYS
3256       IF .CEQ GE SM.MIN.CEQ(.SIDE, .MISSION) AND
3257          (.RATIO GT SM.MAX.TANK.RATIO(.SIDE,.MISSION) OR
3258          .RATIO LT SM.MIN.TANK.RATIO(.SIDE, .MISSION))
3259          CREATE A CAS.MISSION CALLED .CAS
3260          ADD 1 TO SD.NR.CAS.MISSIONS(.SIDE)
3261          LET CMSN.SEQ.NR(.CAS) = SD.NR.CAS.MISSIONS(.SIDE)
3262          LET CMSN.TYPE(.CAS) = ON.CALL
3263          LET CMSN.SIDE(.CAS) = .SIDE
3264          LET CMSN.AC.TYPE(.CAS) =
3265          TU.ATKING.AC(UN.TYPE.UNIT(.TARGET))
3266          LET CMSN.NR.AC(.CAS) =
3267          TU.AC.PER.MSN(UN.TYPE.UNIT(.TARGET))
3268          LET CMSN.TGT.UNIT(.CAS) = .TARGET
3269          LET CMSN.REQUEST.TIME(.CAS) = TIME.V
3270          LET CMSN.ASP.STATUS(.CAS) = IDLE
3271
3272          CALL CHECK.CAS.CONSTRAINTS----------------------------------------------------------------------------------------->(285)
3273          GIVEN
3274          .CAS
3275       ALWAYS
3276    ENDLOOP
3277
3278 <--RETURN
3279 END
```

```
3280 ROUTINE CHECK.CAS.CONSTRAINTS                    ''                                    A004
3281     GIVEN
3282     .CAS
3283
3284     ADD 1 TO ANAL.CTR(100,1)  ''                                                       \DYN_ANAL
3285     NORMALLY MODE IS INTEGER
3286
3287     LET .SIDE = CMSN.SIDE(.CAS)
3288     IF TACAIR.DEBUG = 1
3289        PRINT 2 LINES WITH TIME.V, .CAS, .SIDE THUS     ''                              \1
3290        AT ***.***, ROUTINE CHECK.CAS.CONSTRAINTS CALLED
3291        .CAS = ******     .SIDE = **
3292     ALWAYS
3293
3294     ''WHEN THIS IS A NEW MISSION MAKE SURE ENOUGH
3295     ''AIRCRAFT ARE AVAILABLE
3296     IF .CAS IS NOT IN THE SD.CMSN.QUEUE
3297        FOR EVERY .LINK IN UN.EQUIP.LIST(SD.AIRFIELD(.SIDE))
3298        WITH UE.ID(.LINK) = ACT.EQUIP.ID(CMSN.AC.TYPE(.CAS))
3299        FIND THE FIRST CASE
3300        IF NONE
3301           TRACE
3302 <---------STOP
3303        OTHERWISE
3304
3305        IF UE.QUANT(.LINK) GE CMSN.NR.AC(.CAS)
3306           ''THERE ARE ENOUGH OF THE FIRST CHOICE AIRCRAFT.
3307           SUBTRACT CMSN.NR.AC(.CAS) FROM UE.QUANT(.LINK)
3308        ELSE
3309           IF TACAIR.DEBUG=1
3310              PRINT 1 LINE WITH EQ.NAME(ACT.EQUIP.ID(CMSN.AC.TYPE
3311              (.CAS))) THUS
3312              ==========INSUFFICIENT NUMBER OF **** AVAILABLE==========
3313           ALWAYS
3314           ''TRY THE SUBSTITUTE AIRCRAFT.
3315           LET .SUBS = ACT.SUBSTITUTE(CMSN.AC.TYPE(.CAS))
3316           IF .SUBS = 0
3317              ''USE ALL OF THE REMAINING FIRST CHOICE AIRCRAFT
3318              LET CMSN.NR.AC(.CAS) = UE.QUANT(.LINK)
3319              IF TACAIR.DEBUG=1
3320                 PRINT 1 LINE WITH EQ.NAME(ACT.EQUIP.ID(CMSN.AC.TYPE
3321                 (.CAS))) AND UE.QUANT(.LINK) AS FOLLOWS
3322                 **** HAS NO SUBSTITUTE, FLY THE MISSION WITH ** NUMBER OF AIRCRAFT
3323              ALWAYS
3324              LET UE.QUANT(.LINK) = 0
3325           ELSE
3326              ''FIND THE SUBSTITUTE UE.LINK
3327              FOR EVERY .SUBS.LINK IN UN.EQUIP.LIST(SD.AIRFIELD(.SIDE))
3328              WITH UE.ID(.SUBS.LINK) = ACT.EQUIP.ID(.SUBS)
3329              FIND THE FIRST CASE
3330              IF NONE
3331                 TRACE
3332 <-------------STOP
3333              OTHERWISE
3334              IF TACAIR.DEBUG=1
3335                 PRINT 1 LINE WITH EQ.NAME(ACT.EQUIP.ID(.SUBS)
3336                 THUS
3337                 ==========TRY USING **** AS A SUBSTITUTE AIRCRAFT==========
```

```
3338            ALWAYS
3339
3340            LET .SUBSTITUTE=.SUBS
3341            LET .SUBSTITUTE.LINK=.SUBS.LINK
3342            ''USE THE TYPE WITH THE MOST AIRCRAFT
3343            IF UE.QUANT(.SUBS.LINK) LT CMSN.NR.AC(.CAS)
3344               ''SEE IF THE SUBSTITUTE HAS A SUBSTITUTE----------------------------------------------------------------------->(    )
3345               LET .SUBS.SUBS=ACT.SUBSTITUTE(.SUBS)
3346               IF .SUBS.SUBS NE 0
3347                  FOR EVERY .SUBS.SUBS.LINK IN UN.EQUIP.LIST(
3348                  SD.AIRFIELD(.SIDE))
3349                  WITH UE.ID(.SUBS.SUBS.LINK)=ACT.EQUIP.ID
3350                  (.SUBS.SUBS)
3351                  FIND THE FIRST CASE
3352                  IF NONE
3353                     TRACE
3354 <-----------------STOP
3355                  OTHERWISE
3356                  IF TACAIR.DEBUG=1
3357                     PRINT 1 LINE WITH EQ.NAME(ACT.EQUIP.ID(
3358                     .SUBS.SUBS)) THUS
3359                     ==========TRY USING **** AS ANOTHER SUBSTITUTE AIRCRAFT=========
3360                  ALWAYS
3361                  IF UE.QUANT(.SUBS.LINK) LT UE.QUANT(
3362                     .SUBS.SUBS.LINK)
3363                     LET .SUBSTITUTE = .SUBS.SUBS
3364                     LET .SUBSTITUTE.LINK = .SUBS.SUBS.LINK
3365                  ALWAYS
3366               ALWAYS
3367               IF TACAIR.DEBUG = 1
3368                  PRINT 1 LINE WITH EQ.NAME(ACT.EQUIP.ID(
3369                  .SUBSTITUTE)) THUS
3370                  ==========THE SUBSTITUTE AIRCRAFT TO BE CHECKED IS **** =========
3371               ALWAYS
3372            ALWAYS
3373            IF UE.QUANT(.LINK) GE UE.QUANT(.SUBSTITUTE.LINK)
3374               LET CMSN.NR.AC(.CAS) = UE.QUANT(.LINK)
3375               LET UE.QUANT(.LINK) = 0
3376            ELSE
3377               LET CMSN.AC.TYPE(.CAS) = .SUBSTITUTE
3378               LET CMSN.NR.AC(.CAS) = MIN.F(CMSN.NR.AC(.CAS),
3379               UE.QUANT(.SUBSTITUTE.LINK))
3380               SUBTRACT CMSN.NR.AC(.CAS) FROM UE.QUANT(
3381               .SUBSTITUTE.LINK)
3382            ALWAYS
3383            IF TACAIR.DEBUG=1
3384               PRINT 1 LINE WITH CMSN.SEQ.NR(.CAS), EQ.NAME(
3385               ACT.EQUIP.ID(CMSN.AC.TYPE(
3386               .CAS))) AND UE.QUANT(.SUBSTITUTE.LINK)
3387               THUS
3388               ====FOR CAS MISSION *** USE AIRCRAFT **** WITH *** REMAINING====
3389            ALWAYS
3390         ALWAYS
3391         IF CMSN.NR.AC(.CAS) = 0
3392            PRINT 2 LINES WITH CMSN.SEQ.NR(.CAS),
3393            SHADE(.SIDE), UNIT.NOS(CMSN.TGT.UNIT(.CAS)),
3394            TIME.V  THUS
3395            CAS MISSION *** OF SIDE **** AGAINST UNIT NOS ****** CANCELED
```

```
3396            AT TIME.V = **.*** HRS DUE TO LACK OF AIRCRAFT
3397
3398            DESTROY THE CAS.MISSION CALLED .CAS
3399 <----------RETURN
3400         OTHERWISE
3401      ALWAYS
3402      IF .LINK = .SUBSTITUTE.LINK AND TACAIR.DEBUG=1
3403         PRINT 1 LINE WITH CMSN.SEQ.NR(.CAS),
3404         EQ.NAME(ACT.EQUIP.ID(CMSN.AC.TYPE(
3405         .CAS))) AND UE.QUANT(.LINK) THUS
3406         ====FOR MISSION *** USING AIRCRAFT TYPE **** WITH *** REMAINING====
3407      ALWAYS
3408      IF UN.BATTLE.INDEX(CMSN.TGT.UNIT(.CAS)) GT 0
3409         FOR EVERY FORCE IN BTL.FORCE.SET(UN.BATTLE.INDEX
3410         (CMSN.TGT.UNIT(.CAS)))
3411         WITH FR.SIDE(FORCE) = UN.COLOR(CMSN.TGT.UNIT(.CAS))
3412         LET FR.CAS.INDIC(FORCE) = YES
3413      ALWAYS
3414   ALWAYS
3415
3416   ''CHECK FLYING CONDITIONS
3417   IF VISIBILITY LE SD.NO.FLY.VIS(.SIDE)
3418      LET .FLAG = NO.FLY.VIS
3419   ELSE
3420      IF NITE.OR.DAY = NITE
3421         IF ACT.NITE.FLY(CMSN.AC.TYPE(.CAS)) = 0
3422            LET .FLAG = NITE
3423         ALWAYS
3424      ALWAYS
3425   ALWAYS
3426
3427   IF .FLAG = 0
3428      ''CHECK THE SORTIES PER TIME PERIOD CONSTRAINT
3429      IF SD.SORTIES.THIS.TP(.SIDE) + CMSN.NR.AC(.CAS)
3430         GT SD.MAX.SORTIE.TP(.SIDE)
3431         LET .FLAG = SORTIE.RATE
3432      ALWAYS
3433   ALWAYS
3434
3435   IF .FLAG = 0
3436      ''CHECK THE SORTIES PER AIRSPACE CONSTRAINT
3437      LOOP FOR EVERY .MSN IN EV.S(I.CAS.MISSION)
3438      WITH CMSN.SIDE(.MSN) = .SIDE AND
3439      CMSN.ASP.STATUS(.MSN) = BUSY
3440      DO
3441         LET .DELTAX = UN.X.COORD(CMSN.TGT.UNIT(.CAS))
3442         - UN.X.COORD(CMSN.TGT.UNIT(.MSN))
3443         LET .DELTAY = UN.Y.COORD(CMSN.TGT.UNIT(.CAS))
3444         - UN.Y.COORD(CMSN.TGT.UNIT(.MSN))
3445         LET .DISTANCE = SQRT.F(REAL.F((.DELTAX**2 + .DELTAY**2)))
3446         IF .DISTANCE LE SD.ASC.RADIUS(.SIDE)
3447            ADD CMSN.NR.SURV.AC.(.MSN) TO .COUNT
3448         ALWAYS
3449      ENDLOOP
3450      IF .COUNT + CMSN.NR.AC(.CAS) GT SD.ASC.MAX.SORTIE(.SIDE)
3451         LET .FLAG = AIR.SPACE
3452      ALWAYS
3453   ALWAYS
```

```
3454
3455     LET CMSN.Q.FLAG(.CAS) = .FLAG
3456     IF .FLAG = 0
3457        IF .CAS IS IN THE SD.CMSN.QUEUE
3458           REMOVE .CAS FROM THE SD.CMSN.QUEUE(.SIDE)
3459        ALWAYS
3460        ACTIVATE_THE CAS.MISSION CALLED .CAS NOW------------------------------------------------------------------------------------------>(504)
3461     ELSE
3462        IF .CAS IS NOT IN THE SD.CMSN.QUEUE
3463           FILE .CAS IN THE SD.CMSN.QUEUE(.SIDE)
3464        ALWAYS
3465        PRINT 2 LINES WITH CMSN.SEQ.NR(.CAS), SHADE(.SIDE),
3466        TIME.V, .FLAG, UNIT.NOS(CMSN.TGT.UNIT(.CAS)) THUS
3467        CAS.MISSION *** OF SIDE **** CANNOT BE FLOWN AT TIME.V = **.***
3468        REASON = *, UNIT.NOS OF TARGET = ******
3469     ALWAYS
3470
3471 <--RETURN
3472 END
```

```
3473 ROUTINE EMPLOY.HELICOPTERS                                                                                    A005
3474    GIVEN
3475    FARRP,
3476    BATTLE
3477
3478    ADD 1 TO ANAL.CTR(101,1)  ''                                                                               \DYN_ANAL
3479    ''THIS ROUTINE DIVIDES A FARRP'S HELICOPTERS INTO TEAMS AND SCHEDULES
3480    ''THE TEAMS FOR MOVEMENT TO A BATTLE.
3481
3482    NORMALLY MODE IS INTEGER
3483    DEFINE FARRP, BATTLE AS INTEGER VARIABLES
3484    DEFINE SCT, ATK AS INTEGER, 1-DIM ARRAYS
3485    DEFINE LEAVE.TIME AS A REAL VARIABLE
3486
3487    RESERVE SCT, ATK AS 3
3488
3489    IF HC.DEBUG = "YES"
3490       PRINT 1 LINE WITH FARRP, BATTLE AS FOLLOWS
3491       ----EMPLOY.HELIC----  FARRP=******  BATTLE=******
3492    ALWAYS
3493
3494    ''COUNT SCOUT AND ATTACK HELICOPTERS AND ZERO UE.QUANT(LINK)
3495    LOOP FOR EVERY LINK IN THE UN.EQUIP.LIST(FP.UNIT(FARRP))
3496    DO THE FOLLOWING
3497       LOOP FOR I = 1 TO NO.SCOUT.CONFIGURATIONS
3498       DO THE FOLLOWING
3499          IF EQ.NAME(UE.ID(LINK)) = SCT.HC(I)
3500             ADD UE.QUANT(LINK) TO NO.SCOUT
3501             LET UE.QUANT(LINK) = 0
3502             LET SCT.UE.ID = UE.ID(LINK)
3503       <-----EXITLOOP
3504          ALWAYS
3505       ENDLOOP
3506       LOOP FOR I= 1 TO NO.ATTACK.CONFIGURATIONS
3507       DO THE FOLLOWING
3508          IF EQ.NAME(UE.ID(LINK)) = ATK.HC(I)
3509             ADD UE.QUANT(LINK) TO NO.ATTACK
3510             LET UE.QUANT(LINK) = 0
3511             LET ATK.UE.ID = UE.ID(LINK)
3512       <-----EXITLOOP
3513          ALWAYS
3514       ENDLOOP
3515    ENDLOOP
3516
3517    IF HC.DEBUG = "YES"
3518       PRINT 1 LINE WITH NO.SCOUT, NO.ATTACK AS FOLLOWS
3519       ----TRACE FROM EMPLOY.HELICOPTERS----   NO.SCOUT=****  NO.ATTACK=****
3520    ALWAYS
3521
3522    IF NO.ATTACK LE 0
3523       TRACE
3524 <-----STOP
3525    OTHERWISE
3526
3527    CALL HC.COMPUTE.TIMES-------------------------------------------------------------------------------------->(298)
3528    GIVING
3529    FARRP,
3530    BATTLE.
```

```
3531     SCT.UE.ID,
3532     ATK.UE.ID
3533     YIELDING
3534     FLIGHT.TIME,
3535     LOITER.TIME
3536
3537     ''DIVIDE HELICOPTERS INTO TEAMS
3538     IF NO.ATTACK < 3
3539        IF NO.ATTACK = 1
3540           LET FP.NO.TEAMS(FARRP) = 1
3541           CREATE A HELICOPTER
3542           LET HC.TYPE(HELICOPTER) = ATTACK
3543           LET HC.UE.ID(HELICOPTER) = ATK.UE.ID
3544           CREATE AN HC.TEAM
3545           LET HC.BTL.TEAM(HELICOPTER) = HC.TEAM
3546           LET HT.STATUS(HC.TEAM) = READY
3547           LET HT.LOITER.TIME(HC.TEAM) = LOITER.TIME
3548           LET HT.MOVE.TIME(HC.TEAM) = FLIGHT.TIME
3549           LET HT.FARRP(HC.TEAM) = FARRP
3550           FILE THE HELICOPTER IN THE HT.MEMBER.LIST(HC.TEAM)
3551           FILE THE HC.TEAM IN THE HT.LIST(FARRP)
3552           IF NO.SCOUT GT 0
3553              LOOP FOR I = 1 TO NO.SCOUT
3554              DO THE FOLLOWING
3555                 CREATE A HELICOPTER
3556                 LET HC.TYPE(HELICOPTER) = SCOUT
3557                 LET HC.UE.ID(HELICOPTER) = SCT.UE.ID
3558                 LET HC.BTL.TEAM(HELICOPTER) = HC.TEAM
3559                 FILE THE HELICOPTER IN THE HT.MEMBER.LIST(HC.TEAM)
3560              ENDLOOP
3561           ALWAYS
3562           SCHEDULE_A SEND.TEAM------------------------------------------------------------------------------------------------------>(391)
3563           GIVING
3564           FARRP,
3565           HC.TEAM
3566           IN .001 MINUTES
3567           IF HC.DEBUG = "YES"
3568              PRINT 1 LINE WITH FARRP, HC.TEAM AS FOLLOWS
3569              ----EMPLOY.HELIC----  FARRP=******  TEAM=******
3570           ALWAYS
3571        ELSE
3572           ''TWO ATTACK HELICOPTERS
3573           LET FP.NO.TEAMS(FARRP) = 2
3574           LET SCT(1) = INT.F(NO.SCOUT / 2)
3575           LET SCT(2) = NO.SCOUT - SCT(1)
3576           LOOP FOR J = 1 TO 2
3577           DO THE FOLLOWING
3578              CREATE A HELICOPTER
3579              LET HC.TYPE(HELICOPTER) = ATTACK
3580              LET HC.UE.ID(HELICOPTER) = ATK.UE.ID
3581              CREATE AN HC.TEAM
3582              LET HC.BTL.TEAM(HELICOPTER) = HC.TEAM
3583              LET HT.FARRP(HC.TEAM) = FARRP
3584              LET HT.STATUS(HC.TEAM) = READY
3585              LET HT.LOITER.TIME(HC.TEAM) = LOITER.TIME
3586              LET HT.MOVE.TIME(HC.TEAM) = FLIGHT.TIME
3587              FILE THE HELICOPTER IN THE HT.MEMBER.LIST(HC.TEAM)
3588              FILE THE HC.TEAM IN THE HT.LIST(FARRP)
```

```
3589            IF SCT(J) > 0
3590               LOOP FOR I = 1 TO SCT(J)
3591               DO THE FOLLOWING
3592                  CREATE A HELICOPTER
3593                  LET HC.TYPE(HELICOPTER) = SCOUT
3594                  LET HC.UE.ID(HELICOPTER) = SCT.UE.ID
3595                  LET HC.BTL.TEAM(HELICOPTER) = HC.TEAM
3596                  FILE THE HELICOPTER IN THE
3597                  HT.MEMBER.LIST(HC.TEAM)
3598               ENDLOOP
3599            ALWAYS
3600            IF J = 1
3601               LET LEAVE.TIME = .001
3602            ELSE
3603               LET LEAVE.TIME = LOITER.TIME
3604            ALWAYS
3605            SCHEDULE_A SEND.TEAM------------------------------------------------------------------------------------------>(391)
3606            GIVING
3607            FARRP,
3608            HC.TEAM
3609            IN LEAVE.TIME MINUTES
3610            IF HC.DEBUG = "YES"
3611               PRINT 1 LINE WITH FARRP, HC.TEAM, LEAVE.TIME
3612               AS FOLLOWS
3613               ----EMPLOY.HELIC----  FARRP=****** TEAM****** LEAVE.TIME=*****.***
3614            ALWAYS
3615         ENDLOOP
3616      ALWAYS
3617   ELSE
3618      LET FP.NO.TEAMS(FARRP) = 3
3619      LET SCT(3) = INT.F(NO.SCOUT / 3)
3620      LET SCT(2) = SCT(3)
3621      LET SCT(1) = NO.SCOUT - (SCT(3) + SCT(2))
3622      LET ATK(3) = INT.F(NO.ATTACK / 3)
3623      LET ATK(2) = ATK(3)
3624      LET ATK(1) = NO.ATTACK - (ATK(3) + ATK(2))
3625      LOOP FOR J = 1 TO 3
3626      DO THE FOLLOWING
3627         CREATE AN HC.TEAM
3628         FILE THE HC.TEAM IN THE HT.LIST(FARRP)
3629         LET HT.FARRP(HC.TEAM) = FARRP
3630         LET HT.STATUS(HC.TEAM) = READY
3631         LET HT.LOITER.TIME(HC.TEAM) = LOITER.TIME
3632         LET HT.MOVE.TIME(HC.TEAM) = FLIGHT.TIME
3633         LOOP FOR I = 1 TO ATK(J)
3634         DO THE FOLLOWING
3635            CREATE A HELICOPTER
3636            LET HC.TYPE(HELICOPTER) = ATTACK
3637            LET HC.UE.ID(HELICOPTER) = ATK.UE.ID
3638            LET HC.BTL.TEAM(HELICOPTER) = HC.TEAM
3639            FILE THE HELICOPTER IN THE HT.MEMBER.LIST(HC.TEAM)
3640         ENDLOOP
3641         IF SCT(J) > 0
3642            LOOP FOR I = 1 TO SCT(J)
3643            DO THE FOLLOWING
3644               CREATE A HELICOPTER
3645               LET HC.TYPE(HELICOPTER) = SCOUT
3646               LET HC.UE.ID(HELICOPTER) = SCT.UE.ID
```

```
3647                LET HC.BTL.TEAM(HELICOPTER) = HC.TEAM
3648                FILE THE HELICOPTER IN THE HT.MEMBER.LIST(HC.TEAM)
3649             ENDLOOP
3650          ALWAYS
3651          IF J = 1
3652             LET LEAVE.TIME = .001
3653          ELSE
3654             IF J = 2
3655                LET LEAVE.TIME = LOITER.TIME
3656             ELSE
3657                LET LEAVE.TIME = 2 * LOITER.TIME
3658             ALWAYS
3659          ALWAYS
3660          SCHEDULE_A SEND.TEAM------------------------------------------------------------------------------------------------------------>(391)
3661          GIVING
3662          FARRP,
3663          HC.TEAM
3664          IN LEAVE.TIME MINUTES
3665          IF HC.DEBUG = "YES"
3666             PRINT 1 LINE WITH FARRP, HC.TEAM, LEAVE.TIME AS FOLLOWS
3667             ----EMPLOY.HELIC---- FARRP=****** TEAM=****** LEAVE.TIME=*****.***
3668          ALWAYS
3669       ENDLOOP
3670    ALWAYS
3671
3672    RELEASE SCT, ATK
3673
3674 <--EXITROUTINE
3675 ENDROUTINE
```

```
3676 ROUTINE END.CAS.MISSION                                                    ''                                                    A006
3677     GIVEN
3678     .CMSN,
3679     .STATUS
3680
3681     ADD 1 TO ANAL.CTR(102,1)  ''                                                                                          \DYN_ANAL
3682     NORMALLY MODE IS INTEGER
3683     DEFINE .AFLD TO MEAN SD.AIRFIELD(CMSN.SIDE(.CMSN))
3684     DEFINE .TGT TO MEAN CMSN.TGT.UNIT(.CMSN)
3685     DEFINE .ALPHA AS A TEXT VARIABLE ''                                                                                       \TEXT
3686     DEFINE .SCOREBOARD AS A 2-DIM INTEGER ARRAY  ''                                                                           \VAX
3687
3688     ''EMPTY ALL SETS BELONGING TO THE MISSION
3689     LOOP FOR EVERY .SEG IN CFPS.LIST(.CMSN)
3690     DO
3691        LOOP FOR EVERY .LINK IN SI.LIST(.SEG)
3692        DO
3693           REMOVE .LINK FROM THE SI.LIST(.SEG)
3694           DESTROY THE SENSOR.INTERSECT CALLED .LINK
3695        ENDLOOP
3696        REMOVE .SEG FROM THE CFPS.LIST(.CMSN)
3697        DESTROY THE CFP.SEGMENT CALLED .SEG
3698     ENDLOOP
3699
3700     FOR EACH AD.ENGAGEMENT IN EV.S(I.AD.ENGAGEMENT)
3701     WITH AE.CALLING.PROCESS(AD.ENGAGEMENT) EQ .CMSN
3702     CANCEL_THE AD.ENGAGEMENT------------------------------------------------------------------------------------------------>(349)
3703
3704 <--''STOP THE TARGET UNIT FROM FIRING ON THE AIRCRAFT
3705     FOR EVERY .CAS IN EV.S(I.CAS.MISSION)
3706     WITH CMSN.TGT.UNIT(.CAS) = .CAS AND
3707     CMSN.ASP.STATUS(.CAS) = BUSY AND
3708     CMSN.FIRST.PASS.TIME(.CAS) GT 0.0
3709     FIND THE FIRST CASE
3710     IF FOUND
3711        ''ANOTHER MISSION IS FIRING ON THE SAME TARGET. THE
3712        ''SHOOTOUTS CANNOT BE ENDED.
3713        LET .FLAG = NO
3714     ELSE
3715        LET .FLAG = YES
3716     ALWAYS
3717     LOOP FOR EVERY .LINK IN UN.EQUIP.LIST(.TGT)
3718     DO
3719        LOOP FOR EVERY .FT IN UE.TARGET.LIST(.LINK)
3720        WITH FT.CAS.MISSION(.FT) = .CMSN
3721        DO
3722           REMOVE .FT FROM UE.TARGET.LIST(.LINK)
3723           DESTROY THE FIRING.TABLE CALLED .FT
3724        ENDLOOP
3725        LOOP FOR EVERY .SO IN SO.LIST(.LINK)
3726        DO
3727           LET .KILLED = NO
3728           IF .FLAG = NO OR UN.BATTLE.INDEX(.TGT) GT 0
3729              IF SO.FIRING.TABLE(.SO) GT 0 AND
3730                 FT.CAS.MISSION(SO.FIRING.TABLE(.SO)) = .CMSN
3731                 LET SO.FIRING.TABLE(.SO) = 0
3732                 IF .FLAG = YES
3733                    ''FREE THE AD EQUIPMENT TO FIRE
```

```
3734                  ''AT GROUND UNITS
3735                  LET SO.AIR.ATK.INDIC(.SO) = NO
3736               ALWAYS
3737               LET .KILLED = YES
3738            ALWAYS
3739         ELSE
3740            ''ALL FIGHTING IS OVER
3741            INTERRUPT SHOOT.OUT CALLED .SO------------------------------------------------------------------------------------------>(493)
3742            LET SO.DESTRUCT.INDIC(.SO) = YES
3743            REMOVE .SO FROM SO.LIST(.LINK)
3744            REACTIVATE_THE SHOOT.OUT CALLED .SO NOW---------------------------------------------------------------------------------->(493)
3745            LET .KILLED = YES
3746         ALWAYS
3747         IF .KILLED = YES
3748            LOOP FOR EVERY ASSESSMENT IN EV.S(I.ASSESSMENT)
3749            WITH AS.SHOOT.OUT(ASSESSMENT) = .SO
3750            DO
3751               INTERRUPT ASSESSMENT-------------------------------------------------------------------------------------------------->(487)
3752               LET AS.DESTRUCT.INDIC(ASSESSMENT) = YES
3753               REACTIVATE_THE ASSESSMENT NOW----------------------------------------------------------------------------------------->(487)
3754            ENDLOOP
3755         ALWAYS
3756      ENDLOOP
3757   ENDLOOP
3758
3759   IF CAS.MSN.RPT.FLAG = 1
3760      ''WRITE A REPORT.
3761      START NEW PAGE
3762      LET .SCOREBOARD(*,*)   = CMSN.SCOREBOARD(.CMSN)
3763      IF CMSN.TYPE(.CMSN) = PREPLANNED
3764         LET .ALPHA = "PREPLN"
3765      ELSE
3766         LET .ALPHA = "ON.CLL"
3767      ALWAYS
3768      PRINT 1 LINE WITH
3769      SHADE(CMSN.SIDE(.CMSN)),   CMSN.SEQ.NR(.CMSN)
3770      THUS
3771      REPORT  FOR  **** CAS MISSION NUM. ***
3772
3773      SKIP 1 LINE
3774      IF .STATUS = 1
3775         WRITE AS B 26, "STATUS: ALL AIRCRAFT ABORTED"
3776      ALWAYS
3777      IF .STATUS = 2
3778         WRITE AS B 26, "STATUS: FLIGHT TIME TOO GREAT"
3779      ALWAYS
3780      IF .STATUS = 3
3781         WRITE AS B 20, "STATUS: ALL AIRCRAFT KILLED ",
3782         "BEFORE REACHING TGT"
3783      ALWAYS
3784      IF .STATUS = 4
3785         WRITE AS B 20, "STATUS: ALL AIRCRAFT KILLED ",
3786         "BEFORE  LEAVING  TGT"
3787      ALWAYS
3788      IF .STATUS = 5
3789         WRITE AS B 24, "STATUS: AIRCRAFT RETURNED TO ",
3790         "AIRFIELD OR KILLED ON THE WAY BACK"
3791      ALWAYS
```

```
3792        SKIP 2 LINES
3793        PRINT 10 LINES WITH
3794        CMSN.SEQ.NR(.CMSN),          CMSN.SIDE(.CMSN),
3795        UNIT.NOS(.TGT),              CMSN.REQUEST.TIME(.CMSN),
3796        .ALPHA,                      CMSN.START.TIME(.CMSN),
3797        EQ.NAME(ACT.EQUIP.ID(CMSN.AC.TYPE(.CMSN))),
3798        CMSN.TAKE.OFF.TIME(.CMSN),
3799        CMSN.NR.AC(.CMSN),           CMSN.FIRST.PASS.TIME(.CMSN),
3800        CMSN.NR.ABORTED(.CMSN),      TIME.V,
3801        CMSN.NR.SURV.AC(.CMSN),      CMSN.FLIGHT.TIME(.CMSN)
3802        THUS
3803        MISSION NR    = ***       SIDE             = **
3804        TGT UNIT NOS = ******       REQUEST TIME   = **.** HOURS
3805        MISSION TYPE = ******     START TIME       = **.** HOURS
3806        AC TYPE       = ******    TAKE OFF TIME  = **.** HOURS
3807        NUM AC        = ***       FIRST PASS       = **.** HOURS
3808        NUM ABORTED  = ***        END MISSION      = **.** HOURS
3809        NUM SURVIVED = ***        1 WAY FLY TIME = **.** HOURS
3810
3811        ------WEAPONS------       ----VICTIMS
3812
3813        IF ANALYSIS(6) = TRUE
3814           USE UNIT 47 FOR OUTPUT
3815           PRINT 1 LINE WITH CMSN.NR.AC(.CMSN),CMSN.NR.ABORTED(.CMSN),
3816           CMSN.SEQ.NR(.CMSN),CMSN.TYPE(.CMSN) THUS
3817           NUM.AC= ***  NUM.ABORTED= ***  MISSION.NR= ***  MISSION.TYPE= ******
3818           USE UNIT 6 FOR OUTPUT
3819        ALWAYS
3820
3821        WRITE AS "ID    NAME    MUNS", B 23
3822        FOR EVERY .LINK IN UN.EQUIP.LIST(.TGT)
3823        WRITE EQ.NAME(UE.ID(.LINK)) AS S 2, T 6 ''                          \TEXT
3824        WRITE AS /, "--  ------  ------", B 23
3825        FOR .I = 1 TO N.UN.EQUIP.LIST(.TGT)
3826        WRITE AS S 2, "------"
3827        FOR EVERY .AC.LINK IN UN.EQUIP.LIST(.AFLD)
3828        WITH UE.ID(.AC.LINK) = ACT.EQUIP.ID(CMSN.AC.TYPE(.CMSN))
3829        FIND THE FIRST CASE
3830        IF NONE
3831           TRACE
3832 <------   STOP
3833        OTHERWISE
3834        LET .I = 0
3835        LOOP FOR EACH .WPN IN UE.WEAPON.SET(.AC.LINK)
3836        DO
3837           ADD 1 TO .I
3838           WRITE WPN.ID(.WPN),
3839           TW.NAME(WPN.ID(.WPN)) AS
3840           /, I 3, S 2, T 6, S 2 ''                                         \TEXT
3841           IF WPN.AC.MUNS(.WPN) GT 0
3842              WRITE AM.NAME(WPN.AC.MUNS(.WPN)) AS T 6 ''                    \TEXT
3843           ALWAYS
3844           WRITE AS B 21
3845           FOR .J = 1 TO N.UN.EQUIP.LIST(.TGT)
3846           WRITE .SCOREBOARD(.I, .J) AS S 4, I 4
3847        ENDLOOP
3848        SKIP 10 LINES
3849
```

```
3850        RELEASE .SCOREBOARD
3851
3852     ALWAYS
3853
3854 <--RETURN
3855 END
```

```
3856 ROUTINE FARRP.CHECK                                                                                    A007
3857    GIVEN
3858    BTL,
3859    SIDE
3860    YIELDING
3861    CLOSEST.FARRP
3862
3863    ADD 1 TO ANAL.CTR(103,1)  ''                                                                        \DYN_ANAL
3864    ''WHEN A BATTLE REQUIRES HELICOPTERS, THIS ROUTINE ATTEMPTS TO
3865    ''LOCATE THE CLOSEST AVAILABLE FARRP FOR A SIDE AT THE START
3866    ''OF BATTLE
3867
3868    NORMALLY MODE IS INTEGER
3869
3870    LET MIN.FLIGHT.TIME = 100000
3871    LOOP FOR EVERY FP IN THE FP.SET(SIDE) WITH FP.BATTLE(FP) = 0
3872    DO THE FOLLOWING
3873       LOOP FOR EVERY LINK IN THE UN.EQUIP.LIST(FP.UNIT(FP))
3874       DO THE FOLLOWING
3875          LOOP FOR I = 1 TO NO.SCOUT.CONFIGURATIONS
3876          DO THE FOLLOWING
3877             IF EQ.NAME(UE.ID(LINK)) = SCT.HC(I)
3878                LET SCT.UE.ID = UE.ID(LINK)
3879          <------EXITLOOP
3880             OTHERWISE
3881          ENDLOOP
3882          LOOP FOR I = 1 TO NO.ATTACK.CONFIGURATIONS
3883          DO THE FOLLOWING
3884             IF EQ.NAME(UE.ID(LINK)) = ATK.HC(I)
3885                LET ATK.UE.ID = UE.ID(LINK)
3886          <------EXITLOOP
3887             OTHERWISE
3888          ENDLOOP
3889       ENDLOOP
3890       CALL HC.COMPUTE.TIMES-------------------------------------------------------------------------------------------->(298)
3891       GIVING
3892       FP,
3893       BTL,
3894       SCT.UE.ID,
3895       ATK.UE.ID
3896       YIELDING
3897       FLIGHT.TIME,
3898       LOITER.TIME
3899       IF FLIGHT.TIME < MIN.FLIGHT.TIME AND LOITER.TIME > 0
3900          LET CLOSEST.FARRP = FP
3901          LET MIN.FLIGHT.TIME = FLIGHT.TIME
3902          IF HC.DEBUG = "YES"
3903             PRINT 2 LINES WITH FP, BTL, FLIGHT.TIME,
3904             LOITER.TIME, CLOSEST.FARRP AS FOLLOWS
3905             ----FARRP.CHECK----  FARRP=******  BATTLE=******
3906             FLIGHT.TIME=******  LOITER.TIME=******  CLOSEST.FARRP=******
3907          ALWAYS
3908       ALWAYS
3909    ENDLOOP
3910
3911 <--EXITROUTINE
3912 ENDROUTINE
```

```
                                                                                                                    A008
3913 ROUTINE HC.COMPUTE.TIMES
3914     GIVEN
3915     FARRP,
3916     BATTLE,
3917     SCT.UE.ID,
3918     ATK.UE.ID
3919     YIELDING
3920     FLIGHT.TIME,
3921     LOITER.TIME
3922                                                                                                         \DYN_ANAL
3923     ADD 1 TO ANAL.CTR(104,1)  ''
3924     ''THIS ROUTINE COMPUTES THE AMOUNT OF TIME REQUIRED FOR A HELICOPTER
3925     ''TEAM TO FLY FROM A FARRP TO A BATTLE AND THE AMOUNT OF TIME THAT
3926     ''THE TEAM CAN REMAIN AT THE BATTLE.
3927
3928     NORMALLY MODE IS INTEGER
3929     DEFINE FARRP, BATTLE AS INTEGER VARIABLES
3930
3931     ''FIND COORDINATES OF CENTER OF SUPPORTED FORCE
3932     LET HT.SIDE = UN.COLOR(FP.UNIT(FARRP))
3933     FOR EVERY FORCE IN THE BTL.FORCE.SET(BATTLE)
3934     WITH FR.SIDE(FORCE) = HT.SIDE
3935     FIND THE FIRST CASE
3936     IF NONE
3937        TRACE
3938 <------STOP
3939     OTHERWISE
3940
3941     LOOP FOR EVERY UNIT IN THE FR.UNIT.SET(FORCE)
3942     DO THE FOLLOWING
3943        ADD UN.X.COORD(UNIT) TO FC.X.COORD
3944        ADD UN.Y.COORD(UNIT) TO FC.Y.COORD
3945     ENDLOOP
3946     LET FC.NO.UNITS = N.FR.UNIT.SET(FORCE)
3947     LET FC.X.COORD = FC.X.COORD / FC.NO.UNITS
3948     LET FC.Y.COORD = FC.Y.COORD / FC.NO.UNITS
3949
3950     ''COORDINATES OF FARRP
3951     LET FP.X.COORD = UN.X.COORD(FP.UNIT(FARRP))
3952     LET FP.Y.COORD = UN.Y.COORD(FP.UNIT(FARRP))
3953
3954     ''COMPUTE DISTANCE FROM FARRP TO FORCE CENTER                                                  \OPTIMIZE
3955     LET DISTANCE = SQRT.F((FC.X.COORD - FP.X.COORD)**2 +     ''
3956     (FC.Y.COORD - FP.Y.COORD)**2)
3957
3958     ''FIND MINIMUM SPEED OF TEAM
3959     LOOP FOR EVERY LINK IN THE UN.EQUIP.LIST(FP.UNIT(FARRP))
3960     WITH UE.ID(LINK) = SCT.UE.ID OR UE.ID(LINK) = ATK.UE.ID
3961     DO THE FOLLOWING
3962        COMPUTE MIN.SPEED AS THE MINIMUM OF EQ.MAX.SPEED(UE.ID(LINK))
3963     ENDLOOP
3964     IF MIN.SPEED LE 0
3965        TRACE
3966 <------STOP
3967     ALWAYS
3968
3969     LET MIN.SPEED = (MIN.SPEED * 1000) / 16
3970
```

```
3971     ''CALCULATE FLIGHT TIME TO BATTLE
3972     LET FLIGHT.TIME = DISTANCE / MIN.SPEED * 60
3973     IF UN.COLOR(FP.UNIT(FARRP)) = BLUE
3974        LET FT.TIME = BL.MAX.FL.TIME
3975     ELSE
3976        LET FT.TIME = RO.MAX.FL.TIME
3977     ALWAYS
3978     LET LOITER.TIME = FT.TIME - (2 * FLIGHT.TIME)
3979     IF LOITER.TIME LE 0
3980        LET LOITER.TIME = 0
3981        PRINT 1 LINE AS FOLLOWS
3982        ====WARNING====  LOITER.TIME FOR HELICOPTER TEAM IS ZERO
3983     ALWAYS
3984
3985     IF HC.DEBUG = "YES"
3986        PRINT 4 LINES WITH DISTANCE,
3987        FC.X.COORD,
3988        FC.Y.COORD,
3989        FP.X.COORD,
3990        FP.Y.COORD,
3991        MIN.SPEED,
3992        FLIGHT.TIME,
3993        LOITER.TIME,
3994        FC.NO.UNITS AS FOLLOWS
3995        ----TRACE FROM HC.COMPUTE.TIMES----
3996        DISTANCE=****  FC.X.COORD=****  FC.Y.COORD=****  FP.X.COORD=****
3997        FP.Y.COORD=****  MIN.SPEED=****  FLIGHT.TIME=****  LOITER.TIME=****
3998        FC.NO.UNITS=****
3999     ALWAYS
4000
4001 <--EXITROUTINE
4002 ENDROUTINE
```

```
4003 ROUTINE HC.DISENGAGE                                                    ''                A009
4004     GIVEN
4005     DIS.FARRP,
4006     DIS.TEAM,
4007     .ENEMY.UNITS
4008
4009     ADD 1 TO ANAL.CTR(105,1)  ''                                                          \DYN_ANAL
4010     ''THIS ROUTINE ELIMINATES HELO.ENGAGEMENTS, HELICOPTER.FIRES,
4011     ''SHOOT.OUTS, ASSESSMENTS, FIRING.TABLES, AND VISIBLE.UNITS ASSOCIATED
4012     ''WITH A HELICOPTER TEAM.
4013
4014     NORMALLY MODE IS INTEGER
4015     DEFINE .ENEMY.UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
4016
4017     LET NO.ENEMY.UNITS = DIM.F(.ENEMY.UNITS(*))
4018
4019     IF HC.DEBUG = "YES"
4020        PRINT 1 LINE WITH DIS.FARRP, DIS.TEAM, .ENEMY.UNITS(*) AS FOLLOWS
4021        ----HC.DISENGAGE----  FARRP=******  TEAM=******  ENEMY.UNITS=******
4022     ALWAYS
4023
4024     ''DISENGAGE CURRENT TEAM
4025     ''EMPTY VISIBLE.UNIT LISTS OF HELICOPTERS AND .ENEMY.UNITS
4026     LOOP FOR EACH .HELICOPTER IN THE HT.MEMBER.LIST(DIS.TEAM)
4027     DO THE FOLLOWING
4028        LOOP FOR EACH VISIBLE.UNIT IN THE HC.UN.LOS.LIST(.HELICOPTER)
4029        DO THE FOLLOWING
4030           REMOVE THE VISIBLE.UNIT FROM THE HC.UN.LOS.LIST(.HELICOPTER)
4031           DESTROY THE VISIBLE.UNIT
4032        ENDLOOP
4033        LOOP FOR K = 1 TO NO.ENEMY.UNITS
4034        DO THE FOLLOWING
4035           LOOP FOR EACH VISIBLE.UNIT IN UN.HC.LOS.LIST(.ENEMY.UNITS(K))
4036           WITH VU.POINTER(VISIBLE.UNIT) = .HELICOPTER
4037           DO THE FOLLOWING
4038              REMOVE THE VISIBLE.UNIT FROM THE UN.HC.LOS.LIST(
4039              .ENEMY.UNITS(K))
4040              DESTROY THE VISIBLE.UNIT
4041           ENDLOOP
4042        ENDLOOP
4043     ENDLOOP
4044
4045     ''EMPTY HT.TARGET.LIST
4046     LOOP FOR EVERY .TARGET.UNIT IN THE HT.TARGET.LIST(DIS.TEAM)
4047     DO THE FOLLOWING
4048        REMOVE THE .TARGET.UNIT FROM THE HT.TARGET.LIST(DIS.TEAM)
4049     ENDLOOP
4050
4051     ''REMOVE FIRING.TABLES FROM HELO UNIT UE.TARGET.LISTS
4052     LOOP FOR EVERY LINK IN THE UN.EQUIP.LIST(FP.UNIT(DIS.FARRP))
4053     DO THE FOLLOWING
4054        LOOP FOR J = 1 TO NO.ENEMY.UNITS
4055        DO THE FOLLOWING
4056           LOOP FOR EACH .FIRING.TABLE IN UE.TARGET.LIST(LINK)
4057           WITH .ENEMY.UNITS(J) = FT.TGT.UNIT(.FIRING.TABLE)
4058           DO THE FOLLOWING
4059              REMOVE THE .FIRING.TABLE FROM THE UE.TARGET.LIST(LINK)
4060              DESTROY THE FIRING.TABLE CALLED .FIRING.TABLE
```

```
4061          ENDLOOP
4062       ENDLOOP
4063    ENDLOOP
4064
4065    ''ELIMINATE HELO.ENGAGEMENTS
4066    LOOP FOR EVERY HELO.ENGAGEMENT IN THE EV.S(I.HELO.ENGAGEMENT)
4067    WITH HCEN.TEAM(HELO.ENGAGEMENT) = DIS.TEAM
4068    DO THE FOLLOWING
4069       CANCEL_THE HELO.ENGAGEMENT------------------------------------------------------------------------------------------>(376)
4070       DESTROY THE HELO.ENGAGEMENT
4071    ENDLOOP
4072
4073    ''ELIMINATE HELICOPTER.FIRES
4074    LOOP FOR EVERY HELICOPTER.FIRE IN EV.S(I.HELICOPTER.FIRE)
4075    WITH HF.TEAM(HELICOPTER.FIRE) = DIS.TEAM
4076    DO THE FOLLOWING
4077       INTERRUPT HELICOPTER.FIRE-------------------------------------------------------------------------------------------->(510)
4078       LET HT.TERMINATOR(HF.TEAM(HELICOPTER.FIRE)) = YES
4079       LET TIME.A(HELICOPTER.FIRE) = -RINF.C
4080       RESUME HELICOPTER.FIRE----------------------------------------------------------------------------------------------->(510)
4081    ENDLOOP
4082
4083    ''ELIMINATE SHOOT.OUTS, ASSESSMENTS AND FIRING.TABLES FROM .ENEMY.UNITS
4084    LOOP FOR I = 1 TO NO.ENEMY.UNITS
4085    DO THE FOLLOWING
4086       LOOP FOR EVERY LINK IN THE UN.EQUIP.LIST(.ENEMY.UNITS(I))
4087       DO THE FOLLOWING
4088          LOOP FOR EVERY HC IN THE HT.MEMBER.LIST(DIS.TEAM)
4089          DO THE FOLLOWING
4090             LOOP FOR EVERY FIRING.TABLE IN THE UE.TARGET.LIST(LINK)
4091             WITH FT.TGT.UNIT(FIRING.TABLE) =
4092             FP.UNIT(DIS.FARRP)
4093             DO THE FOLLOWING
4094                REMOVE THE FIRING.TABLE FROM THE
4095                UE.TARGET.LIST(LINK)
4096                DESTROY THE FIRING.TABLE
4097             ENDLOOP
4098             LOOP FOR EVERY ASSESSMENT IN EV.S(I.ASSESSMENT)
4099             WITH SO.HELICOPTER(AS.SHOOT.OUT(
4100             ASSESSMENT)) = HC
4101             DO THE FOLLOWING
4102                INTERRUPT ASSESSMENT---------------------------------------------------------------------------------------->(487)
4103                LET AS.DESTRUCT.INDIC(ASSESSMENT) = YES
4104                LET TIME.A(ASSESSMENT) = -RINF.C
4105                RESUME ASSESSMENT------------------------------------------------------------------------------------------->(487)
4106             ENDLOOP
4107          ENDLOOP
4108       ENDLOOP
4109    ENDLOOP
4110    LOOP FOR EVERY SHOOT.OUT IN EV.S(I.SHOOT.OUT)
4111    WITH SO.HELICOPTER(SHOOT.OUT) GT 0 AND
4112    HC.BTL.TEAM(SO.HELICOPTER(SHOOT.OUT)) = DIS.TEAM
4113    DO
4114       LET SO.FIRING.TABLE(SHOOT.OUT) = 0
4115    ENDLOOP
4116
4117 <--EXITROUTINE
4118 ENDROUTINE
```

```
4119 ROUTINE HEL.RANGE.COMPUTE                                          ''          A010
4120     GIVEN
4121     .HELICOPTER,
4122     .GROUND.UNIT
4123     YIELDING
4124     .RANGE
4125
4126     ADD 1 TO ANAL.CTR(106,1)  ''                                                \DYN_ANAL
4127     ''THIS ROUTINE COMPUTES THE RANGE BETWEEN A SPECIFIC
4128     ''HELICOPTER AND A GROUND UNIT'S COORDINATES.  THE VALUES
4129     ''ARE HDM.
4130
4131     NORMALLY MODE IS INTEGER
4132
4133     ''LET .X1 = HC.X(.HELICOPTER)
4134     ''LET .Y1 = HC.Y(.HELICOPTER)
4135     ''LET .X2 = UN.X.COORD(.GROUND.UNIT)
4136     ''LET .Y2 = UN.Y.COORD(.GROUND.UNIT)
4137     ''LET .RANGE = SQRT.F((.X1 - .X2)**2 + (.Y1 - .Y2)**2)
4138
4139
4140     ''THE FOLLOWING CODE REPLACES THE ABOVE IN ORDER THAT HELOS WILL
4141     ''MOVE DURING THE BATTLE...THE ACTUAL LOCATION OF THE HELO WILL
4142     ''NOT BE RECORDED, THE ASSUMPTION BEING THAT THEY WILL RETURN TO
4143     ''THE FARRP FROM THE POSITION THEY ASSUMED AT THE BATTLE START
4144
4145
4146     LET .FARRP = HT.FARRP(HC.BTL.TEAM(.HELICOPTER))
4147     IF HC.TYPE(.HELICOPTER) NE ATTACK
4148        FOR EACH .HELO IN HT.MEMBER.LIST(HC.BTL.TEAM(.HELICOPTER))
4149        WITH HC.TYPE(.HELO) = ATTACK    *
4150        FIND THE FIRST CASE
4151        IF NONE
4152           PRINT 1 LINE WITH HC.BTL.TEAM(.HELICOPTER) THUS
4153           === HEL.RNG.COM  NO ATTACK HELOS IN TEAM ******
4154           LET .RANGE = UNIFORM.F(100.,190.,RN.SEED)
4155 <---------RETURN
4156        OTHERWISE
4157     ELSE
4158        LET .HELO = .HELICOPTER
4159     ALWAYS
4160     FOR EACH .UEL IN UN.EQUIP.LIST(FP.UNIT(.FARRP))
4161     WITH UE.ID(.UEL) = HC.UE.ID(.HELO)
4162     FIND THE FIRST CASE
4163     IF NONE
4164        TRACE
4165 <-------STOP
4166     OTHERWISE
4167     LET MAX.WPN.RNG = 0
4168     FOR EACH .WPN IN UE.WEAPON.SET(.UEL)
4169     WITH TW.MAX.RANGE(WPN.ID(.WPN)) > MAX.WPN.RNG
4170     LET MAX.WPN.RNG = TW.MAX.RANGE(WPN.ID(.WPN))
4171     IF UN.COLOR(FP.UNIT(.FARRP)) = BLUE
4172        LET .RANGE = MAX.WPN.RNG * UNIFORM.F(BL.LOW.FRAC.RANGE,
4173        BL.HIGH.FRAC.RANGE,RN.SEED)
4174     ELSE
4175        LET .RANGE = MAX.WPN.RNG * UNIFORM.F(RD.LOW.FRAC.RANGE,
4176        RD.HIGH.FRAC.RANGE,RN.SEED)
```

```
4177     ALWAYS
4178
4179     IF HC.DEBUG = "YES"
4180        PRINT 2 LINES WITH .RANGE*16, .X1*16, .Y1*16, .X2*16, .Y2*16,
4181        .HELICOPTER, UNIT.NOS(.GROUND.UNIT) AS FOLLOWS
4182        ----HEL$RNG$COM----  RANGE=****** X1=****** Y1=****** X2=****** Y2=******
4183        .HELICOPTER=****** .GROUND.UNIT=******
4184     ALWAYS
4185
4186 ENDROUTINE
```

```
4187 ROUTINE REPLACE.HC                                                    ''                                    A011
4188    GIVEN
4189    RH.FARRP,
4190    RH.TEAM
4191
4192    ADD 1 TO ANAL.CTR(107,1)  ''                                                                             \DYN_ANAL
4193    ''THIS ROUTINE TRANSFERS THE HELICOPTERS OF A TEAM BACK TO THE
4194    ''FARRP AND DESTROYS THE TEAM.
4195
4196    NORMALLY MODE IS INTEGER
4197
4198    LOOP FOR EVERY I IN HT.MEMBER.LIST(RH.TEAM)
4199    DO THE FOLLOWING
4200       LOOP FOR EVERY LINK IN THE UN.EQUIP.LIST(FP.UNIT(RH.FARRP))
4201       DO THE FOLLOWING
4202          IF UE.ID(LINK) = HC.UE.ID(I)
4203             ADD 1 TO UE.QUANT(LINK)
4204
4205             IF HC.DEBUG = "YES"
4206                PRINT 1 LINE WITH LINK,
4207                UE.QUANT(LINK), RH.FARRP, RH.TEAM
4208                AS FOLLOWS
4209                ----REPLACE.HC---- LINK=****** UE.QUANT(LINK)=****** FARRP=****** TEAM=******
4210             ALWAYS
4211          ALWAYS
4212       ENDLOOP
4213    ENDLOOP
4214
4215    ''DESTROY THE HELICOPTERS
4216    LOOP UNTIL THE HT.MEMBER.LIST(RH.TEAM) IS EMPTY
4217    DO THE FOLLOWING
4218       REMOVE THE FIRST HELICOPTER FROM THE HT.MEMBER.LIST(RH.TEAM)
4219       DESTROY THE HELICOPTER
4220    ENDLOOP
4221
4222    ''DESTROY THE TEAM
4223    SUBTRACT 1 FROM FP.NO.TEAMS(RH.FARRP)
4224    REMOVE THIS RH.TEAM FROM THE HT.LIST(RH.FARRP)
4225    DESTROY THE HC.TEAM CALLED RH.TEAM
4226
4227    IF HC.DEBUG = "YES"
4228       PRINT 1 LINE WITH RH.FARRP, N.HT.LIST(RH.FARRP) AS FOLLOWS
4229       ----REPLACE.HC----  FARRP=******  N.HT.LIST(FARRP)=******
4230    ALWAYS
4231
4232 <--EXITROUTINE
4233 ENDROUTINE
```

```
4234 ROUTINE UNIT.PRIORITY                                                    A012
4235    GIVEN
4236    FR.COLOR,
4237    BTL
4238    YIELDING
4239    PRIORITY
4240
4241    ADD 1 TO ANAL.CTR(108,1)  ''                                          \DYN_ANAL
4242    ''THIS ROUTINE CALCULATES THE PRIORITY OF A FORCE TO BE SUPPORTED
4243    ''BY HELICOPTERS BASED ON MISSION AND COMPOSITION OF THE FORCE
4244    ''AND THE COMPOSITON OF THE OPPOSING FORCE
4245
4246    NORMALLY MODE IS INTEGER
4247
4248    FOR EVERY FORCE IN THE BTL.FORCE.SET(BTL)
4249    WITH FR.SIDE(FORCE) = FR.COLOR
4250    FIND THE FIRST CASE
4251    IF NONE
4252       TRACE
4253 <------STOP
4254    OTHERWISE
4255
4256    IF HC.DEBUG = "YES"
4257       PRINT 1 LINE WITH SUP.MISSION.PRIORITY(FR.MISSION(FORCE),
4258       FR.SIDE(FORCE)), FR.MISSION(FORCE), FR.SIDE(FORCE)
4259       AS FOLLOWS
4260       ----UNIT.PRIORITY---- SUP.MIS.PRIOR=****** FR.MISSION=**** FR.SIDE=****
4261    ALWAYS
4262
4263    LET P1 = SUP.MISSION.PRIORITY(FR.MISSION(FORCE), FR.SIDE(FORCE))
4264    IF FR.COLOR = BLUE
4265       LET P2 = MAX.F(0, N.FR.UNIT.SET(FORCE) - (MIN.NO.SUP.UNITS - 1))
4266    ELSE
4267       LET P2 = N.FR.UNIT.SET(FORCE)
4268    ALWAYS
4269
4270    LOOP FOR EVERY UNIT IN THE FR.UNIT.SET(FORCE)
4271    DO THE FOLLOWING
4272       ADD TU.SUP.PRIORITY(UN.TYPE.UNIT(UNIT`) TO P3
4273    ENDLOOP
4274
4275    FOR EVERY FORCE IN THE BTL.FORCE.SET(BTL)
4276    WITH FR.SIDE(FORCE) NE FR.COLOR
4277    FIND THE FIRST CASE
4278    IF NONE
4279       TRACE
4280 <------STOP
4281    OTHERWISE
4282
4283    LOOP FOR EVERY UNIT IN THE FR.UNIT.SET(FORCE)
4284    DO THE FOLLOWING
4285       ADD TU.OPP.PRIORITY(UN.TYPE.UNIT(UNIT)) TO P4
4286    ENDLOOP
4287    LET PRIORITY = P1 * P2 * P3 * P4
4288
4289    IF HC.DEBUG = "YES"
4290       PRINT 2 LINE WITH FR.COLOR, P1, P2, P3, P4, PRIORITY AS FOLLOWS
4291       ----TRACE FROM UNIT.PRIORITY----
```

```
4292        FR.COLOR=**  P1=****  P2=****  P3=****  P4=****  PRIORITY=*******
4293     ALWAYS
4294
4295 <--EXITROUTINE
4296 ENDROUTINE
```

```
4297 ROUTINE AD.SHOOT                                                    ''                                                    A013
4298    GIVEN
4299     .RANGE,
4300     .FIRING.TIME,
4301     .AD.UNIT,
4302     .CALLING.PROCESS,
4303     .TYPE.PROCESS
4304
4305    YIELDING
4306     .RESULT,
4307     .NEXT.FIRING
4308
4309    ADD 1 TO ANAL.CTR(109,1)  ''                                                                                  \DYN_ANAL
4310    NORMALLY MODE IS INTEGER
4311    DEFINE .FIRING.TIME, .NEXT.FIRING, .DELTA, .ORDINATE, .PK, .TOF
4312    AS REAL VARIABLES
4313    DEFINE .TYPE.PROCESS AS A TEXT VARIABLE ''                                                                    \TEXT
4314
4315    IF TACAIR.DEBUG  = 1
4316       PRINT 1 LINE WITH TIME.V THUS
4317       = = =AD.SHOOT CALLED AT ***.***
4318       LIST .RANGE, .FIRING.TIME, .AD.UNIT, .CALLING.PROCESS,
4319       .TYPE.PROCESS
4320       IF .TYPE.PROCESS = "CAS"
4321          LIST ATTRIBUTES OF AD.ENGAGEMENT CALLED .CALLING.PROCESS
4322          LIST CMSN.SEQ.NR(AE.CALLING.PROCESS(.CALLING.PROCESS)),
4323          CMSN.SIDE(AE.CALLING.PROCESS(.CALLING.PROCESS))
4324       ELSE
4325          LIST ATTRIBUTES OF AC.ATK.TGT  CALLED .CALLING.PROCESS
4326          LIST CMSN.SEQ.NR(AAT.CMSN(.CALLING.PROCESS)),
4327          CMSN.SIDE(AAT.CMSN(.CALLING.PROCESS))
4328       ALWAYS
4329    ALWAYS
4330    FOR EACH .LINK IN UN.EQUIP.LIST(.AD.UNIT)
4331    WITH EQ.AD.INDICATOR(UE.ID(.LINK)) = 1
4332    FIND THE FIRST CASE
4333    IF NONE
4334       TRACE
4335 <------STOP
4336    OTHERWISE
4337    LET .FIRING.EQ = UE.ID(.LINK)
4338    LET .WEAPON = F.UE.WEAPON.SET(.LINK)
4339    LET .TYPE.WPN = WPN.ID(.WEAPON)
4340    IF .TYPE.PROCESS = "AO"
4341       LET .TARGET.EQ = US.EQ.ID(AO.US.LINK(.CALLING.PROCESS))
4342    ALWAYS
4343    IF .TYPE.PROCESS = "CAS"
4344       LET .TARGET.EQ = ACT.EQUIP.ID(CMSN.AC.TYPE(.CALLING.PROCESS))
4345    ALWAYS
4346    IF .TYPE.PROCESS = "AC.ATK"
4347       LET .TARGET.EQ = ACT.EQUIP.ID(CMSN.AC.TYPE(AAT.CMSN(
4348       .CALLING.PROCESS)))
4349    ALWAYS
4350
4351    LET .PK.PTR = PK.POINTER(TW.PK.PTR(.TYPE.WPN),
4352    EQUIP.PK.PTR(.TARGET.EQ))
4353    IF .PK.PTR = 0
4354       LET .RESULT = NO.PK.PTR
```

```
4355 <-----RETURN
4356    OTHERWISE
4357
4358    IF TW.MIN.RANGE(.TYPE.WPN) GT .RANGE OR
4359       TW.MAX.RANGE(.TYPE.WPN) LT .RANGE
4360       LET .RESULT = NO.SHOT.MADE
4361 <-----RETURN
4362    OTHERWISE
4363
4364    ''INTERPOLATE THE PK BETWEEN GIVEN VALUES.
4365    LET .DELTA = (TW.MAX.RANGE(.TYPE.WPN) - TW.MIN.RANGE(.TYPE.WPN)) / 10.
4366    LET .ORDINATE = (.RANGE - TW.MIN.RANGE(.TYPE.WPN)) / .DELTA
4367    LET .LOW.POINT = TRUNC.F(.ORDINATE)
4368    LET .HIGH.POINT = .LOW.POINT + 1
4369    LET .DELTA = .ORDINATE - .LOW.POINT
4370
4371    IF .DELTA = 0. AND .LOW.POINT GT 0
4372       LET .PK = .02 * PK.PROB(.PK.PTR, .LOW.POINT)
4373    ELSE
4374       FOR EVERY PK.BAND
4375       WITH PK.BAND.RNG(PK.BAND) = .LOW.POINT
4376       FIND THE FIRST CASE
4377       IF NONE
4378          TRACE
4379 <--------STOP
4380       OTHERWISE
4381       LET .LPK = PK.PROB(.PK.PTR, PK.BAND)
4382
4383       FOR EVERY PK.BAND
4384       WITH PK.BAND.RNG(PK.BAND) = .HIGH.POINT
4385       FIND THE FIRST CASE
4386       IF NONE
4387          TRACE
4388 <--------STOP
4389       OTHERWISE
4390       LET .HPK = PK.PROB(.PK.PTR, PK.BAND)
4391
4392       LET .PK = 2. * (.LPK + .DELTA * (.HPK - .LPK)) / 100.
4393       ''THE 100 CONVERTS .PK FROM A PERCENTAGE TO A FRACTION.
4394       ''THE 2 CONVERTS .PK FROM A RANGE OF 0 TO 50 TO A RANGE
4395       ''OF 0 TO 100 (SEE PK.INPUT).
4396    ALWAYS
4397
4398    ''DETERMINE THE NUMBER OF WEAPONS THAT WILL FIRE.
4399    FOR EVERY .ADS IN SD.ADS.SET(UN.COLOR(.AD.UNIT))
4400    WITH ADS.UNIT.PTR(.ADS) = .AD.UNIT
4401    FIND THE FIRST CASE
4402    IF NONE
4403       TRACE
4404 <-----STOP
4405    OTHERWISE
4406
4407    LET .MADS = ADS.MADS.PTR(.ADS)
4408    IF MADS.FCM(.MADS) = RIPPLE
4409       LET .NUM.WPNS = 1
4410    ELSE
4411       LET .NUM.WPNS = UE.QUANT(.LINK) * WPN.QUANTITY(.WEAPON)
4412    ALWAYS
```

```
4413    SUBTRACT .NUM.WPNS FROM WPN.ROUNDS.REMAINING(.WEAPON)
4414
4415    LET .PK = 1. - (1. - .PK)**.NUM.WPNS
4416    IF RANDOM.F(RN.SEED) GT .PK
4417       LET .RESULT = MISS
4418       '' NOTE THAT FOR CATCEM DATA .CALLING.PROCESS SHOULD BE
4419       '' RECORDED BUT FOR TIMELINESS (P90E), THE NUMBER 9999999999 IS
4420       '' BEING SUBSTITUTED.      %12MAY83_%HWJ
4421       IF ANALYSIS(5) = TRUE ''%MAY83_%JAF% CATCEM
4422          WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(.WEAPON)),
4423          EQ.NAME(.FIRING.EQ),
4424          EQ.TE.PTR(.FIRING.EQ),
4425          TW.NAME(WPN.ID(.WEAPON)),
4426          EQ.NAME(.TARGET.EQ),
4427          EQ.TE.PTR(.TARGET.EQ),
4428          .RANGE*16,DEFILADE,
4429          UN.COLOR(.AD.UNIT), .PK AS S 1,"0",S 1,D(7,4),S 1,
4430          I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''                    \TEXT
4431          I 6,S 4,I 1,S 4,"9999999999",S 4,I 1,S 4,D(4,3),
4432          S 4,"@AD1"./
4433          USING UNIT 55
4434       ALWAYS
4435    ELSE
4436       LET .RESULT = HIT
4437       IF ANALYSIS(5) = TRUE ''%MAY83_%JAF% CATCEM
4438          WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(.WEAPON)),
4439          EQ.NAME(.FIRING.EQ),
4440          EQ.TE.PTR(.FIRING.EQ),
4441          TW.NAME(WPN.ID(.WEAPON)),
4442          EQ.NAME(.TARGET.EQ),
4443          EQ.TE.PTR(.TARGET.EQ),
4444          .RANGE*16,DEFILADE,
4445          UN.COLOR(.AD.UNIT), .PK AS S 1,"1",S 1,D(7,4),S 1,
4446          I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''                    \TEXT
4447          I 6,S 4,I 1,S 4,"9999999999",S 4,I 1,S 4,D(4,3),
4448          S 4,"@AD2"./
4449          USING UNIT 55
4450       ALWAYS
4451
4452       ADD 1 TO KV.SCORE(UN.COLOR(.AD.UNIT),
4453       EQ.KV.ID(.FIRING.EQ), EQ.KV.ID(.TARGET.EQ))
4454       ''REDUCE THE NUMBER OF AIRCRAFT.
4455       IF .TYPE.PROCESS = "AO"
4456          LET AO.US.LINK(.CALLING.PROCESS) = 9999
4457          INTERRUPT AIR.OBSERVER CALLED .CALLING.PROCESS------------------------------------------------------------>(428)
4458          REACTIVATE_THE AIR.OBSERVER CALLED .CALLING.PROCESS NOW---------------------------------------------------->(428)
4459       ALWAYS
4460       IF .TYPE.PROCESS = "CAS"
4461          SUBTRACT 1 FROM CMSN.NR.SURV.AC(.CALLING.PROCESS)
4462          IF CMSN.NR.SURV.AC(.CALLING.PROCESS) = 0
4463             INTERRUPT CAS.MISSION CALLED .CALLING.PROCESS------------------------------------------------------->(504)
4464             REACTIVATE_THE CAS.MISSION CALLED .CALLING.PROCESS NOW---------------------------------------------->(504)
4465          ALWAYS
4466       ALWAYS
4467       ''WHEN TYPE IS AC.ATK, THE NUMBER OF AIRCRAFT WILL BE REDUCED                \1
4468       ''IN AC.ATK.TGT.
4469    ALWAYS
4470
```

```
4471     ADD .NUM.WPNS * TW.RND.WEIGHT(.TYPE.WPN) TO
4472     KV.AMMO.CONSUMED(UN.COLOR(.AD.UNIT), EQ.KV.ID(.FIRING.EQ))
4473     ADD .NUM.WPNS TO STW.RND.FIRED(UN.COLOR(.AD.UNIT), .TYPE.WPN)
4474     IF UN.COLOR(.AD.UNIT) = BLUE
4475        ADD .NUM.WPNS TO
4476        STY.BLUE.EXP(.TYPE.WPN, .TARGET.EQ - N.BLUE.TYPE.EQP)
4477     ELSE
4478        ADD .NUM.WPNS TO
4479        STY.RED.EXP(.TYPE.WPN - N.B.WPN.TYPE, .TARGET.EQ)
4480     ALWAYS
4481
4482     ''CREATE A NOISE THAT CAN BE HEARD BY PASSIVE DETECTION BASES.
4483     IF TW.RND.WEIGHT(.TYPE.WPN) GT 1000
4484        CREATE A DF.NOISE
4485        LET DF.TIME(DF.NOISE) = TIME.V
4486        LET DF.UNIT(DF.NOISE) = .AD.UNIT
4487        FILE DF.NOISE IN DF.RATE.LIST
4488     ALWAYS
4489
4490     ''DETERMINE THE TIME WHEN THE NEXT SHOT CAN BE FIRED.
4491     LET .NEXT.FIRING = .FIRING.TIME
4492     IF WPN.ROUNDS.REMAINING(.WEAPON) LT .NUM.WPNS
4493        ADD MADS.WPN.RELOAD.TIME(.MADS) / 60. TO .NEXT.FIRING
4494        LET WPN.ROUNDS.REMAINING(.WEAPON) = UE.QUANT(.LINK)
4495        * WPN.QUANTITY(.WEAPON) * TW.BASIC.LOAD(.TYPE.WPN)
4496     ALWAYS
4497     IF MADS.FCM(.MADS) = RIPPLE OR MADS.FCM(.MADS) = SALVO
4498        ADD MADS.RIPL(.MADS) / 3600 TO .NEXT.FIRING
4499     ELSE
4500        ''LOOK-SHOOT-LOOK. ADD TIME OF FLIGHT PLUS LOOK TIME.
4501        LET .TOF = 16. * .RANGE / (3600. * TW.ROUND.VELOCITY(.TYPE.WPN))
4502        + UNIFORM.F(1., 5., RN.SEED) / 3600.
4503        ADD .TOF TO .NEXT.FIRING
4504     ALWAYS
4505     IF TACAIR.DEBUG = 1
4506        LIST .RESULT, .NEXT.FIRING, .PK, .NUM.WPNS, .TYPE.WPN
4507        LIST .TARGET.EQ, .PK.PTR, .LPK, .HPK, PK.BAND
4508     ALWAYS
4509
4510 <--RETURN
4511 END
```

```
4512 ROUTINE INTER.HELO                                                                                    ''                                                                                
```

```
4570              DESTROY THE VISIBLE.UNIT
4571           ENDLOOP
4572           LOOP FOR .I = 1 TO .NO.ENEMY.UNITS
4573           DO THE FOLLOWING
4574              LOOP FOR EACH VISIBLE.UNIT
4575              IN THE UN.HC.LOS.LIST(.ENEMY.UNITS(.I))
4576              WITH VU.POINTER(VISIBLE.UNIT) = .HC
4577              DO THE FOLLOWING
4578                 REMOVE THE VISIBLE.UNIT FROM
4579                 THE UN.HC.LOS.LIST(.ENEMY.UNITS(.I))
4580                 DESTROY THE VISIBLE.UNIT
4581              ENDLOOP
4582           ENDLOOP
4583        ENDLOOP
4584
4585        ''EMPTY HT.TARGET.LIST
4586        LOOP FOR EVERY .TARGET.UNIT IN THE HT.TARGET.LIST(.TEAM)
4587        DO THE FOLLOWING
4588           REMOVE THE .TARGET.UNIT FROM THE HT.TARGET.LIST(.TEAM)
4589        ENDLOOP
4590
4591        ''REMOVE FIRING.TABLES FROM HELO UNIT UE.TARGET.LISTS
4592        LOOP FOR EVERY .LINK IN THE UN.EQUIP.LIST(FP.UNIT(.FARRP))
4593        DO THE FOLLOWING
4594           LOOP FOR .I = 1 TO .NO.ENEMY.UNITS
4595           DO THE FOLLOWING
4596              LOOP FOR EACH .FIRING.TABLE IN
4597              THE UE.TARGET.LIST(.LINK)
4598              WI'' .ENEMY.UNITS(.I) =
4599              FT  ST.UNIT(.FIRING.TABLE)
4600              DO THE FOLLOWING
4601                 REMOVE THE .FIRING.TABLE FROM
4602                 THE UE.TARGET.LIST(.LINK)
4603                 DESTROY THE FIRING.TABLE CALLED .FIRING.TABLE
4604              ENDLOOP
4605           ENDLOOP
4606        ENDLOOP
4607
4608        ''ELIMINATE HELO.ENGAGEMENTS
4609        LOOP FOR EVERY HELO.ENGAGEMENT IN THE EV.S(I.HELO.ENGAGEMENT)
4610        WITH HCEN.TEAM(HELO.ENGAGEMENT) = .TEAM
4611        DO THE FOLLOWING
4612           CANCEL_THE HELO.ENGAGEMENT------------------------------------------------------------------------------------------>(376)
4613           DESTROY THE HELO.ENGAGEMENT
4614        ENDLOOP
4615
4616        ''ELIMINATE HELICOPTER.FIRES
4617        LOOP FOR EVERY HELICOPTER.FIRE IN EV.S(I.HELICOPTER.FIRE)
4618        WITH HF.TEAM(HELICOPTER.FIRE) = .TEAM
4619        DO THE FOLLOWING
4620           INTERRUPT HELICOPTER.FIRE-------------------------------------------------------------------------------------------->(510)
4621           LET HF.REINFORCE.IND(HELICOPTER.FIRE) = YES
4622           LET TIME.A(HELICOPTER.FIRE) = -RINF.C
4623           RESUME HELICOPTER.FIRE----------------------------------------------------------------------------------------------->(510)
4624        ENDLOOP
4625     ENDLOOP     ''FOR EACH HEL.TARGET.ACQUISITION
4626  ENDLOOP          ''FOR EACH SIDE
4627
```

```
4628 <--EXITROUTINE
4629 ENDROUTINE
```

```
4630 ROUTINE FLIGHT.PATH                                                    ''          A015
4631     GIVEN
4632     .CAS.MISSION
4633
4634     ADD 1 TO ANAL.CTR(111,1)  ''                                                   \DYN_ANAL
4635     NORMALLY MODE IS INTEGER
4636     DEFINE .SIDE TO MEAN CMSN.SIDE(.CAS.MISSION)
4637     DEFINE .AC.TYPE TO MEAN CMSN.AC.TYPE(.CAS.MISSION)
4638     DEFINE .SLOPE, .B, .ROOT AS REAL VARIABLES
4639     DEFINE .MADS.NAME AS TEXT VARIABLE ''                                          \TEXT
4640
4641     ''THIS ROUTINE SETS THE FLIGHT PATH OF A CAS MISSION,
4642     ''MANEUVERING AROUND ANY KNOWN SENSORS, CALLED FLIGHT
4643     ''PATH OBSTACLES.  THE LIST OF OBSTACLES, COL.SET, IS RANKED
4644     ''BY LOW COL.XENTRY.  THE VALUE STORED HERE IS MULTIPLIED
4645     ''BY +1 FOR BLUE AND -1 FOR RED.  THIS MEANS THE
4646     ''OBSTACLES IN THE SET WILL BE RANKED IN THE ORDER
4647     ''THE MISSION FLIES PAST THEM.
4648
4649     LET .ENEMY = UN.COLOR(CMSN.TGT.UNIT(.CAS.MISSION))
4650
4651     ''SET FLIGHT DIRECTION
4652     IF .SIDE = BLUE
4653        LET .SIGN = +1
4654     ELSE
4655        LET .SIGN = -1
4656     ALWAYS
4657     LET .XSTART = UN.X.COORD(SD.AIRFIELD(.SIDE))
4658     LET .YSTART = UN.Y.COORD(SD.AIRFIELD(.SIDE))
4659
4660     ''SET THE FIRST POINT OF THE AIRCRAFTS PASS TRIANGLE.
4661     LET .XOBJ = UN.X.COORD(CMSN.TGT.UNIT(.CAS.MISSION))
4662     +ACT.X1(CMSN.AC.TYPE(.CAS.MISSION))
4663     LET .YOBJ = UN.Y.COORD(CMSN.TGT.UNIT(.CAS.MISSION))
4664     +ACT.Y1(CMSN.AC.TYPE(.CAS.MISSION))
4665     IF TACAIR.DEBUG = 3
4666        LIST .XOBJ, .YOBJ, UN.X.COORD(CMSN.TGT.UNIT(.CAS.MISSION)),
4667        UN.Y.COORD(CMSN.TGT.UNIT(.CAS.MISSION)),
4668        .CAS.MISSION
4669     ALWAYS
4670
4671     ''DETERMINE THE EQUATION OF THE STRAIGHT LINE PATH
4672     ''FROM THE AIRFIELD TO THE OBJECTIVE.
4673     LET .SLOPE = (.YOBJ - .YSTART) /(.XOBJ - .XSTART)
4674     LET .B = .YSTART - .SLOPE * .XSTART
4675
4676     ''SET THE FIRST POINT OF THE FLIGHT PATH.
4677     CREATE A CFP.SEGMENT CALLED .CS
4678     FILE .CS IN THE CFPS.LIST(.CAS.MISSION)
4679     LET CFPS.XSTART(.CS) = .XSTART
4680     LET CFPS.YSTART(.CS) = .YSTART
4681
4682     ''LOOK FOR KNOWN ENEMY AD SENSORS IN THE PATH.
4683     LOOP FOR EVERY .FPO IN SD.FPO.LIST(.ENEMY)
4684     DO
4685        IF TACAIR.DEBUG = 3
4686           LIST ATTRIBUTES OF FP.OBSTACLE CALLED .FPO
4687        ALWAYS
```

```
4688        LET .Y1 = .SLOPE * FPO.XMIN(.FPO) + .B
4689        LET .Y2 = .SLOPE * FPO.XMAX(.FPO) + .B
4690        IF FPO.YMIN(.FPO) LT .Y1 LT FPO.YMAX(.FPO) OR
4691           FPO.YMIN(.FPO) LT .Y2 LT FPO.YMAX(.FPO)
4692           CREATE A CFP.OBS.LINK CALLED .COL
4693           LET COL.OBSTACLE.PTR(.COL) = .FPO
4694           IF .SIDE = BLUE
4695              LET COL.XENTRY(.COL) = .SIGN * FPO.XMIN(.FPO)
4696              LET COL.YENTRY(.COL) = .Y1
4697              LET COL.XEXIT (.COL) = .SIGN * FPO.XMAX(.FPO)
4698              LET COL.YEXIT (.COL) = .Y2
4699           ELSE
4700              LET COL.XENTRY(.COL) = .SIGN * FPO.XMAX(.FPO)
4701              LET COL.YENTRY(.COL) = .Y2
4702              LET COL.XEXIT (.COL) = .SIGN * FPO.XMIN(.FPO)
4703              LET COL.YEXIT (.COL) = .Y1
4704           ALWAYS
4705           FILE .COL IN THE COL.SET
4706        ALWAYS
4707     ENDLOOP
4708
4709     ''CREATE THE FLIGHT PATH. TRY TO MANEUVER AROUND THE OBSTACLES.
4710     LOOP FOR EACH .COL IN THE COL.SET
4711     DO
4712        IF TACAIR.DEBUG = 3
4713           LIST ATTRIBUTES OF CFP.OBS.LINK CALLED .COL
4714        ALWAYS
4715        LET .FPO = COL.OBSTACLE.PTR(.COL)
4716        IF FPO.XMIN(.FPO) LT .XOBJ LT FPO.XMAX(.FPO)
4717           ''OBJECTIVE IS ENCLOSED.
4718           CREATE A CFP.SEGMENT CALLED .CS
4719           FILE .CS IN CFPS.LIST(.CAS.MISSION)
4720           LET CFPS.XSTART(.CS) = .SIGN * COL.XENTRY(.COL)
4721           LET CFPS.YSTART(.CS) = .YOBJ
4722        ELSE
4723           ''FLIGHT PATH PASSES THRU SENSOR.
4724           IF FPO.YMIN(.FPO) LE 0 AND
4725              FPO.YMAX(.FPO) GE FEBA.WIDTH
4726              ''NO WAY AROUND. GO STRAIGHT THRU.
4727              LET .Y1 = COL.YENTRY(.COL)
4728           ELSE
4729              ''GO AROUND THE CLOSEST END.
4730              LET .Y2 = CFPS.YSTART(.CS)
4731              IF ABS.F(.Y2 - FPO.YMIN(.FPO)) LT
4732                 ABS.F(.Y2 - FPO.YMAX(.FPO))
4733                 LET .Y1 = FPO.YMIN(.FPO)
4734                 IF .Y1 LE 0
4735                    LET .Y1 = FPO.YMAX(.FPO)
4736                 ALWAYS
4737              ELSE
4738                 LET .Y1 = FPO.YMAX(.FPO)
4739                 IF .Y1 GE FEBA.WIDTH
4740                    LET .Y1 = FPO.YMIN(.FPO)
4741                 ALWAYS
4742              ALWAYS
4743           ALWAYS
4744           CREATE A CFP.SEGMENT CALLED .CS
4745           FILE .CS IN CFPS.LIST(.CAS.MISSION)
```

```
4746          LET CFPS.XSTART(.CS) = .SIGN * COL.XENTRY(.COL)
4747          LET CFPS.YSTART(.CS) = .Y1
4748          CREATE A CFP.SEGMENT CALLED .CS
4749          FILE .CS IN CFPS.LIST(.CAS.MISSION)
4750          LET CFPS.XSTART(.CS) = .SIGN * COL.XEXIT(.COL)
4751          LET CFPS.YSTART(.CS) = .Y1
4752       ALWAYS
4753       REMOVE .COL FROM THE COL.SET
4754       DESTROY THE CFP.OBS.LINK CALLED .COL
4755    ENDLOOP
4756
4757    ''SET THE END POINTS OF THE SEGMENTS AND THEIR OTHER
4758    ''ATTRIBUTES
4759    LOOP FOR EACH .CS IN CFPS.LIST(.CAS.MISSION)
4760    DO
4761       IF S.CFPS.LIST(.CS) = 0
4762          LET CFPS.XEND(.CS) = .XOBJ
4763          LET CFPS.YEND(.CS) = .YOBJ
4764       ELSE
4765          LET CFPS.XEND(.CS) = CFPS.XSTART(S.CFPS.LIST(.CS))
4766          LET CFPS.YEND(.CS) = CFPS.YSTART(S.CFPS.LIST(.CS))
4767       ALWAYS
4768       LET CFPS.TIME.LENGTH(.CS) = SQRT.F(REAL.F(
4769       (CFPS.XEND(.CS) - CFPS.XSTART(.CS))**2 + ''                                    \OPTIMIZE
4770       (CFPS.YEND(.CS) - CFPS.YSTART(.CS))**2 ))  ''                                  \OPTIMIZE
4771       / (ACT.SPEED(CMSN.AC.TYPE(.CAS.MISSION)) * 3600)
4772       IF TACAIR.DEBUG = 3
4773          LIST ATTRIBUTES OF CFP.SEGMENT CALLED .CS
4774          LIST N.CFPS.LIST(.CAS.MISSION), S.CFPS.LIST(.CS)
4775       ALWAYS
4776    ENDLOOP
4777
4778    ''DETERMINE IF THE FLIGHT PATH WILL CROSS ANY OF THE
4779    ''ENEMY AD SENSORS, KNOWN OR UNKNOWN.
4780    LET .SPEED = ACT.SPEED(.AC.TYPE) * 3600
4781    LOOP FOR EACH .SEG OF CFPS.LIST(.CAS.MISSION)
4782    DO
4783       LET .XMIN = MIN.F(CFPS.XSTART(.SEG), CFPS.XEND(.SEG))
4784       LET .XMAX = MAX.F(CFPS.XSTART(.SEG), CFPS.XEND(.SEG))
4785       LET .SLOPE = (CFPS.YEND(.SEG) - CFPS.YSTART(.SEG))
4786       / (CFPS.XEND(.SEG) -  CFPS.XSTART(.SEG))
4787       LET .B = CFPS.YSTART(.SEG) - .SLOPE * CFPS.XSTART(.SEG)
4788       LOOP FOR EACH .AD OF SD.ADS.SET(.ENEMY)
4789       DO
4790          LET .X0 = UN.X.COORD(ADS.UNIT.PTR(.AD))
4791          LET .Y0 = UN.Y.COORD(ADS.UNIT.PTR(.AD))
4792          LET .MADS.NAME = MADS.NAME(ADS.MADS.PTR(.AD))
4793          LET .UNIT = ADS.UNIT.PTR(.AD)
4794          LET .RANGE = MRH.RANGE(L.MADS.RH.SET(ADS.MADS.PTR(.AD)))
4795          CALL LINE.CIRCLE------------------------------------------------------------------------------------------------------>(645)
4796          GIVEN
4797          .SLOPE,
4798          .B,
4799          .X0,
4800          .Y0,
4801          .RANGE,
4802          .AD
4803          YIELDING
```

```
4804          .X1,
4805          .Y1,
4806          .X2,
4807          .Y2,
4808          .ROOT
4809
4810          IF .ROOT LT 0 OR
4811             .X1 GE .XMAX OR
4812             .X2 LE .XMIN
4813             ''EITHER NO INTERSECTION WITH THIS SENSOR OR
4814             ''THE INTERSECTION IS NOT ON THIS SEGMENT.
4815       <-----CYCLE
4816          OTHERWISE
4817
4818          CREATE A SENSOR.INTERSECT CALLED .SI
4819          FILE .SI IN SI.LIST(.SEG)
4820          ''WHEN THE SENSOR IS KNOWN, THE AIRCRAFT TRIES TO                    \1
4821          ''FLY UNDER THE SENSORS MINIMUM EFFECTIVE ALTITUDE.
4822          FOR EVERY .KAS IN SD.KAS.SET(.SIDE)
4823          WITH KAS.AD.UNIT(.KAS) = ADS.UNIT.PTR(.AD)
4824          FIND THE FIRST CASE
4825          IF FOUND
4826             LET SI.ALTITUDE(.SI) = ACT.MIN.ALT(.AC.TYPE)
4827          ELSE
4828             LET SI.ALTITUDE(.SI) = ACT.NORM.ALT(.AC.TYPE)
4829          ALWAYS
4830          LET SI.AD.UNIT(.SI) = ADS.UNIT.PTR(.AD)
4831          IF .XMIN LE .X1 AND .X2 LE .XMAX
4832             ''THE SEGMENT GOES THRU THE SENSOR.
4833             IF .SIDE = BLUE
4834                LET SI.X.ENTRY(.SI) = .X1
4835                LET SI.Y.ENTRY(.SI) = .Y1
4836                LET SI.X.EXIT (.SI) = .X2
4837                LET SI.Y.EXIT (.SI) = .Y2
4838             ELSE
4839                LET SI.X.ENTRY(.SI) = .X2
4840                LET SI.Y.ENTRY(.SI) = .Y2
4841                IF MADS.DETECT(ADS.MADS.PTR(.AD)) = 1
4842                   IF .X2 LT .X0
4843                      REMOVE .SI FROM SI.LIST(.SEG)
4844                      DESTROY THE SENSOR.INTERSECT
4845                      CALLED .SI
4846       <-------------CYCLE
4847                   OTHERWISE
4848                   IF .X1 LE .X0
4849                      LET .X1 = .X0
4850                      LET .Y1 = .SLOPE*.X1+.B
4851                   ALWAYS
4852                ALWAYS
4853                LET SI.X.EXIT (.SI) = .X1
4854                LET SI.Y.EXIT (.SI) = .Y1
4855             ALWAYS
4856             LET SI.TIME.TIL.INTERSECT(.SI) = SQRT.F(REAL.F(
4857             (SI.X.ENTRY(.SI) - CFPS.XSTART(.SEG)) ** 2 +
4858             (SI.Y.ENTRY(.SI) - CFPS.YSTART(.SEG)) ** 2 ))
4859             / .SPEED
4860          ELSE
4861             ''THE SEGMENT STARTS OR ENDS IN THE SENSOR.
```

```
4862            IF .XMIN LE .X1
4863               IF .SIDE = BLUE
4864                  ''ENDS IN THE SENSOR
4865                  LET SI.X.ENTRY(.SI) = .X1
4866                  LET SI.Y.ENTRY(.SI) = .Y1
4867                  LET SI.X.EXIT (.SI) = CFPS.XEND(.SEG)
4868                  LET SI.Y.EXIT (.SI) = CFPS.YEND(.SEG)
4869                  LET SI.TIME.TIL.INTERSECT(.SI) = SQRT.F(
4870                  REAL.F((.X1 - CFPS.XSTART(.SEG)) ** 2 +
4871                  (.Y1 - CFPS.YSTART(.SEG)) ** 2)) / .SPEED
4872               ELSE
4873                  ''STARTS IN THE SENSOR
4874                  LET SI.X.ENTRY(.SI) = CFPS.XSTART(.SEG)
4875                  LET SI.Y.ENTRY(.SI) = CFPS.YSTART(.SEG)
4876                  IF MADS.DETECT(ADS.MADS.PTR(.AD)) = 1
4877                     IF CFPS.XSTART(.SEG) LT .X0
4878                        REMOVE .SI FROM SI.LIST(.SEG)
4879                        DESTROY THE SENSOR.INTERSECT
4880                        CALLED .SI
4881      <-----------------CYCLE
4882                     OTHERWISE
4883                     IF .X1 LE .X0
4884                        LET .X1 = .X0
4885                        LET .Y1 = .SLOPE * .X1 + .B
4886                     ALWAYS
4887                  ALWAYS
4888                  LET SI.X.EXIT (.SI) = .X1
4889                  LET SI.Y.EXIT (.SI) = .Y1
4890                  LET SI.TIME.TIL.INTERSECT(.SI) = 0.0
4891            ALWAYS
4892         ELSE
4893            IF .SIDE = BLUE
4894               ''STARTS IN THE SENSOR
4895               LET SI.X.ENTRY(.SI) = CFPS.XSTART(.SEG)
4896               LET SI.Y.ENTRY(.SI) = CFPS.YSTART(.SEG)
4897               LET SI.X.EXIT (.SI) = .X2
4898               LET SI.Y.EXIT (.SI) = .Y2
4899               LET SI.TIME.TIL.INTERSECT(.SI) = 0.0
4900            ELSE
4901               ''ENDS IN THE SENSOR
4902               LET SI.X.ENTRY(.SI) = .X2
4903               LET SI.Y.ENTRY(.SI) = .Y2
4904               IF MADS.DETECT(ADS.MADS.PTR(.AD)) = 1
4905                  IF .X2 LT .X0
4906                     REMOVE .SI FROM SI.LIST(.SEG)
4907                     DESTROY THE SENSOR.INTERSECT
4908                     CALLED .SI
4909      <-----------------CYCLE
4910                  OTHERWISE
4911                  IF CFPS.XEND(.SEG) LT .X0
4912                     LET SI.X.EXIT (.SI) = .X0
4913                     LET SI.Y.EXIT (.SI)=.X0*.SLOPE+.B
4914                  ELSE
4915                     LET SI.X.EXIT (.SI) = CFPS.XEND(.SEG)
4916                     LET SI.Y.EXIT (.SI) = CFPS.YEND(.SEG)
4917                  ALWAYS
4918               ELSE
4919                  LET SI.X.EXIT (.SI) = CFPS.XEND(.SEG)
```

```
4920                        LET SI.Y.EXIT (.SI) = CFPS.YEND(.SEG)
4921                     ALWAYS
4922                     LET SI.TIME.TIL.INTERSECT(.SI) = SQRT.F(
4923                     REAL.F((.X2 - CFPS.XSTART(.SEG)) ** 2 +
4924                     (.Y2 - CFPS.YSTART(.SEG)) ** 2)) / .SPEED
4925                  ALWAYS
4926               ALWAYS
4927            ALWAYS
4928            IF .X1 LE 0
4929               LIST ATTRIBUTES OF SENSOR.INTERSECT CALLED .SI
4930            ALWAYS
4931         ENDLOOP
4932      ENDLOOP
4933
4934 <--RETURN
4935 END
```

COMBAT SUPPORT ROUTINES

```
4942 ROUTINE COMPUTE.D                                                    C001
4943    GIVEN
4944    UNIT
4945    YIELDING
4946    D,
4947    POINT
4948
4949    ADD 1 TO ANAL.CTR(112,1)  ''                                      \DYN_ANAL
4950    NORMALLY MODE IS INTEGER
4951
4952    LET COUNT = 1
4953    LOOP FOR EVERY POINT OF UN.PATH(UNIT)
4954    UNTIL COUNT = UN.POSITION.INDEX(UNIT) + 1
4955    DO THE FOLLOWING
4956       ADD 1 TO COUNT
4957    ENDLOOP
4958    LET D.X = UN.X.COORD(UNIT) - P.X(POINT)
4959    LET D.Y = UN.Y.COORD(UNIT) - P.Y(POINT)
4960    LET D = (D.X.**2 + D.Y.**2)**.5          ''                       \OPTIMIZE
4961
4962 <--EXITROUTINE
4963 ENDROUTINE
```

```
                                                                                                  C002
4964 ROUTINE COMPUTE.WD
4965     GIVEN
4966     UNIT
4967     YIELDING
4968     D,
4969     TEMP.POINT
4970                                                                                    \DYN_ANAL
4971     ADD 1 TO ANAL.CTR(113,1)  ''
4972     NORMALLY MODE IS INTEGER
4973
4974     FOR EVERY POINT OF UN.PATH(UNIT)
4975     WITH P.X(POINT) = UN.X.COORD(UNIT) AND
4976     P.Y(POINT) = UN.Y.COORD(UNIT)
4977     FIND THE FIRST CASE
4978     IF FOUND,
4979        LET COUNT = 1
4980     ELSE
4981        LET COUNT = 0
4982     ALWAYS
4983     LOOP FOR EVERY POINT OF UN.PATH(UNIT)
4984     UNTIL COUNT = UN.POSITION.INDEX(UNIT)
4985     DO THE FOLLOWING
4986        LET TEMP.POINT = POINT
4987        ADD 1 TO COUNT
4988     ENDLOOP
4989     IF TEMP.POINT = 0
4990        LET TEMP.POINT = F.UN.PATH(UNIT)
4991     ALWAYS
4992     LET D.X = UN.X.COORD(UNIT) - P.X(TEMP.POINT)
4993     LET D.Y = UN.Y.COORD(UNIT) - P.Y(TEMP.POINT)                                   \OPTIMIZE
4994     LET D = (D.X.**2 + D.Y.**2)**.5      ''
4995
4996 <--EXITROUTINE
4997 ENDROUTINE
```

```
4998 ROUTINE CONTRAST.TO.FREQ                                                    C003
4999     GIVEN
5000     .RANGE
5001
5002     YIELDING
5003     .FREQUENCY
5004
5005     ADD 1 TO ANAL.CTR(114,1)  ''                                            \DYN_ANAL
5006     NORMALLY MODE IS INTEGER
5007     DEFINE .DELTA.CONTRAST, .FREQUENCY, .EXTINCTION.COEF, .SOG
5008     AS REAL VARIABLES
5009
5010     LET .EXTINCTION.COEF = 3.912 / (16 * VISIBILITY / 1000) ''              \OPTIMIZE
5011     LET .SOG = .3333 * (16 * VISIBILITY / 1000) + .6667
5012     LET .DELTA.CONTRAST = .3 / (1.0
5013     + .SOG * (EXP.F(.EXTINCTION.COEF * .RANGE / 1000) - 1.0))
5014     IF NITE.OR.DAY = DAY
5015        IF .DELTA.CONTRAST GT .35
5016           LET .FREQUENCY = 2.8668 * (.DELTA.CONTRAST **0.2251)
5017        ELSE
5018           IF .DELTA.CONTRAST GE 0.025
5019              LET .FREQUENCY = 2.3263 - 0.0515 / .DELTA.CONTRAST
5020           ELSE
5021              LET .FREQUENCY = 0.0
5022           ALWAYS
5023        ALWAYS
5024     ELSE
5025        IF .DELTA.CONTRAST GE .18
5026           LET .FREQUENCY = 0.5687 - 0.0968 / .DELTA.CONTRAST
5027        ELSE
5028           LET .FREQUENCY = 0.0
5029        ALWAYS
5030     ALWAYS
5031
5032 <--RETURN
5033 END
```

```
5034 ROUTINE DECIDE                                                                    ''                                                                
```

```
5078 ROUTINE DEQ.FEBA.SET          ''                                                                                C005
5079    GIVEN UNIT
5080    YIELDING SECTOR '
5081
5082    ADD 1 TO ANAL.CTR(116,1)  ''                                                                              \DYN_ANAL
5083    NORMALLY MODE IS INTEGER
5084    DEFINE SIDE AS AN INTEGER VARIABLE
5085    IF DEBUG=TRUE,
5086       PRINT 1 LINE WITH UNIT.NOS(UNIT) THUS
5087       DEQ.FEBA.SET UNIT=******
5088    ENDIF
5089    CALL LOCATE.SECTOR  GIVEN UN.Y.COORD(UNIT) YIELDING PRESENT.SECTOR------------------------------------------------------------------------>(100)
5090    LET SIDE = UN.COLOR(UNIT)
5091    ''REMOVE THIS UNIT FROM THE SS.SET( SIDE,SECTOR )
5092    ''THIS ROUTINE WAS RECODED %20MAR80 BY %RGR TO REMOVE A UNIT FROM
5093    ''ANY SET THE UNIT MAY HAVE BEEN A MEMBER OF, -IF THE UNIT IS NEAR
5094    ''OR ON A SECTOR BOUNDRY, IT MAY HAVE MOVED ACROSS THE SECTOR LINE
5095    ''AND CALLING THE ROUTINE LOCATE.SECTOR ONLY PRODUCES THE SECTOR
5096    ''THE UNIT PRESENTLY IS LOCATED WITHIN. A CONFLICT WILL ARISE WHEN
5097    ''THE SET OPERATION TO FILE THE UNIT IN A NEW SET  IF
5098    ''THIS ROUTINE IS CALLED WITHIN A LOOP, SUCH AS IN UPDATE.LOCATION.
5099    ''IN THAT SITUATION, THE LAST VALUE FOR SECTOR IS USED.
5100    LOOP FOR EACH SECTOR
5101    DO THIS ''                                                                                                  \1
5102       FOR EACH UN OF SS.SET(SIDE,SECTOR)
5103       WHEN UN = UNIT
5104       FIND THE FIRST CASE
5105       IF FOUND
5106          REMOVE THIS UNIT FROM THE SS.SET(SIDE,SECTOR)
5107          LET OLD.SECTOR = SECTOR
5108    <------EXITLOOP
5109       ENDIF
5110    ENDLOOP
5111    IF OLD.SECTOR = 0    '' IT WAS NOT PREVIOUSLY FILED IN A SS.SET
5112       LET SECTOR = PRESENT.SECTOR
5113    ELSE
5114       LET SECTOR = OLD.SECTOR
5115    ALWAYS
5116
5117 ENDROUTINE
```

```
5118 ROUTINE DESTROY.ORD(UNT.)  ''                                                    C006
5119    ADD 1 TO ANAL.CTR(117,1)  ''                                                  \DYN_ANAL
5120    NORMALLY MODE IS INTEGER
5121    DEFINE ORDER AS A VARIABLE
5122    LOOP UNTIL MU.ORDER.SET(UN.PTR(UNT.)) IS EMPTY,
5123    DO ''                                                                         \1
5124       REMOVE FIRST ORDER FROM MU.ORDER.SET(UN.PTR(UNT.))
5125       IF ORD.TYPE(ORDER)="DEF"
5126          DESTROY THE DEF.ORDER CALLED ORD.ID(ORDER)
5127       ELSEIF ORD.TYPE(ORDER)="ATK"
5128          DESTROY THE ATK.ORDER CALLED ORD.ID(ORDER)
5129       ELSEIF ORD.TYPE(ORDER)="REINF"
5130          DESTROY THE REINF.ORDER CALLED ORD.ID(ORDER)
5131       ELSEIF ORD.TYPE(ORDER)="MOVDIS"
5132          DESTROY THE MOVDIS.ORDER CALLED ORD.ID(ORDER)
5133       ELSEIF ORD.TYPE(ORDER)="MOVCOR"
5134          DESTROY THE MOVCOR.ORDER CALLED ORD.ID(ORDER)
5135       ELSEIF ORD.TYPE(ORDER)="MOVREI"
5136          DESTROY THE MOVREINF.ORD CALLED ORD.ID(ORDER)
5137       ELSE
5138          CALL ERROR.STOP------------------------------------------------------------------------------------------------------>(604)
5139       ENDIF ALWAYS ALWAYS ALWAYS ALWAYS ALWAYS
5140       DESTROY THE ORDER
5141    ENDLOOP
5142 ENDROUTINE
```

```
5143 ROUTINE DQ.CMSN.QUEUE                                                                                    C007
5144    ADD 1 TO ANAL.CTR(118,1)  ''                                                                          \DYN_ANAL
5145    NORMALLY MODE IS INTEGER
5146
5147    ''WEATHER OR LIGHT HAS CHANGED OR A CAS MISSION
5148    ''HAS FINISHED AND REDUCED THE NUMBER OF SORTIES
5149    ''IN THE AIRSPACE, OR A NEW TIME PERIOD FOR CONSTRAINED
5150    ''SORTIES HAS STARTED.  SEE IF ANY OF THE CAS
5151    ''MISSIONS ON HOLD CAN NOW FLY.
5152
5153    IF TACAIR.DEBUG = 1
5154       PRINT 1 LINE WITH TIME.V THUS
5155       = = = = = = DQ.CMSN.QUEUE CALLED AT ***.***
5156    ALWAYS
5157    FOR EACH SIDE
5158    FOR EACH .CAS IN SD.CMSN.QUEUE(SIDE)
5159    CALL CHECK.CAS.CONSTRAINTS------------------------------------------------------------------------------------------------------------>(285)
5160    GIVEN
5161    .CAS
5162
5163 <--RETURN
5164 END
```

```
5165 ROUTINE EMPTY                                                                                                    ''
5166    GIVEN                                                                                                         C008
5167    BATTLE,
5168    WINNER,
5169    END.BATTLE
5170
5171    ADD 1 TO ANAL.CTR(119,1)  ''                                                                                  \DYN_ANAL
5172    ''THIS ROUTINE IS TO EMPTY LISTS
5173    ''AFTER CLOSE COMBAT ENGAGEMENTS ARE CONDUCTED
5174
5175    NORMALLY MODE IS INTEGER
5176    DEFINE UNIT AS AN INTEGER VARIABLE
5177    DEFINE BATTLE.STATUS AS A TEXT VARIABLE   ''                                                                  \TEXT
5178
5179    IF BATTLE = 0
5180       CALL ERROR.STOP------------------------------------------------------------------------------------------->(604)
5181    ALWAYS
5182
5183    LOOP FOR EVERY MOVE OF EV.S(I.MOVE)
5184    WITH UN.BATTLE.INDEX(MV.UNIT(MOVE)) = BATTLE
5185    DO THE FOLLOWING
5186       CANCEL_THE MOVE------------------------------------------------------------------------------------------->(381)
5187       DESTROY THE MOVE
5188    ENDLOOP
5189
5190    LOOP FOR EVERY ENGAGEMENT OF EV.S(I.ENGAGEMENT)
5191    WITH UN.BATTLE.INDEX(DETECTING.UNIT(ENGAGEMENT)) = BATTLE
5192    DO THE FOLLOWING
5193       CANCEL_THE ENGAGEMENT------------------------------------------------------------------------------------->(369)
5194       DESTROY THE ENGAGEMENT
5195    ENDLOOP
5196
5197    LOOP FOR EVERY ASSESSMENT OF EV.S(I.ASSESSMENT)
5198    WITH UN.BATTLE.INDEX(AS.FIRING.UNIT(ASSESSMENT)) = BATTLE AND
5199    AS.TGT.UNIT(ASSESSMENT) NE
5200    SD.AIRFIELD(UN.COLOR(AS.TGT.UNIT(ASSESSMENT)))
5201    DO THE FOLLOWING
5202       INTERRUPT ASSESSMENT-------------------------------------------------------------------------------------->(487)
5203       LET TIME.A(ASSESSMENT) = -RINF.C
5204       LET AS.DESTRUCT.INDIC(ASSESSMENT) = YES
5205       RESUME ASSESSMENT----------------------------------------------------------------------------------------->(487)
5206    ENDLOOP
5207
5208    LOOP FOR EVERY WITH.DRAW OF EV.S(I.WITH.DRAW)
5209    WITH UN.BATTLE.INDEX(WD.UNIT(WITH.DRAW)) = BATTLE
5210    DO THE FOLLOWING
5211       INTERRUPT WITH.DRAW--------------------------------------------------------------------------------------->(478)
5212       LET TIME.A(WITH.DRAW) = -RINF.C
5213       LET WD.DESTRUCT.INDIC(WITH.DRAW) = YES
5214       RESUME WITH.DRAW------------------------------------------------------------------------------------------>(478)
5215    ENDLOOP
5216
5217    LOOP FOR EVERY ARTY.ASSESS OF EV.S(I.ARTY.ASSESS)
5218    WITH UN.BATTLE.INDEX(AA.UNIT(ARTY.ASSESS)) = BATTLE
5219    DO THE FOLLOWING
5220       INTERRUPT ARTY.ASSESS------------------------------------------------------------------------------------->(435)
5221       LET AA.DESTRUCT.INDIC(ARTY.ASSESS) = YES
5222       RESUME ARTY.ASSESS---------------------------------------------------------------------------------------->(435)
```

```
5223    ENDLOOP
5224
5225    LOOP FOR EVERY MINE.ASSESS OF EV.S(I.MINE.ASSESS)
5226    WITH UN.BATTLE.INDEX(MA.UNIT(MINE.ASSESS)) = BATTLE
5227    DO
5228       INTERRUPT MINE.ASSESS------------------------------------------------------------------------------------------>(464)
5229       LET MA.BATTLE.ENDED(MINE.ASSESS) = YES
5230       REACTIVATE_THE MINE.ASSESS NOW---------------------------------------------------------------------------------->(464)
5231    ENDLOOP
5232
5233    LOOP FOR EACH FORCE OF BTL.FORCE.SET(BATTLE)
5234    DO THE FOLLOWING
5235       LOOP FOR EACH UNIT OF FR.UNIT.SET(FORCE)
5236       DO THE FOLLOWING
5237          LET UN.POSITION.INDEX(UNIT) = 0
5238          LET UN.BATTLE.INDEX(UNIT) = 0
5239          LET UN.TIME.LAST.MOVE(UNIT) = 0
5240
5241          LOOP UNTIL UN.LOS.LIST(UNIT) IS EMPTY
5242          DO THE FOLLOWING
5243             REMOVE THE FIRST VISIBLE.UNIT FROM UN.LOS.LIST(UNIT)
5244             DESTROY THE VISIBLE.UNIT
5245          ENDLOOP
5246
5247          LOOP FOR EVERY UE.LINK OF UN.EQUIP.LIST(UNIT)
5248          DO THE FOLLOWING
5249             LOOP FOR EVERY FIRING.TABLE IN UE.TARGET.LIST(UE.LINK)
5250             UNLESS FT.AC.ATK.TGT(FIRING.TABLE) GT 0
5251             DO THE FOLLOWING
5252                REMOVE THE FIRING.TABLE
5253                FROM UE.TARGET.LIST(UE.LINK)
5254                DESTROY THE FIRING.TABLE
5255             ENDLOOP
5256             IF UE.TARGET.LIST(UE.LINK) IS EMPTY
5257                ''THERE ARE NO AIRCRAFT TO BE KILLED.
5258             LOOP UNTIL SO.LIST(UE.LINK) IS EMPTY
5259             DO THE FOLLOWING
5260                REMOVE THE FIRST SHOOT.OUT FROM SO.LIST(UE.LINK)
5261                INTERRUPT SHOOT.OUT------------------------------------------------------------------------------->(493)
5262                LET TIME.A(SHOOT.OUT) = -RINF.C
5263                LET SO.DESTRUCT.INDIC(SHOOT.OUT) = YES
5264                RESUME SHOOT.OUT---------------------------------------------------------------------------------->(493)
5265             ENDLOOP
5266          ALWAYS
5267          ENDLOOP
5268
5269          LOOP UNTIL THE UN.PATH(UNIT) IS EMPTY
5270          DO THE FOLLOWING
5271             REMOVE THE FIRST POINT FROM UN.PATH(UNIT)
5272             DESTROY THE POINT
5273          ENDLOOP
5274
5275          LOOP UNTIL UN.SEGMENT.LIST(UNIT) IS EMPTY
5276          DO THE FOLLOWING
5277             REMOVE THE FIRST SEGMENT FROM UN.SEGMENT.LIST(UNIT)
5278             DESTROY THE SEGMENT
5279          ENDLOOP
5280          REMOVE THE UNIT FROM FR.UNIT.SET(FORCE)
```

```
5281        ENDLOOP
5282        REMOVE THE FORCE FROM BTL.FORCE.SET(BATTLE)
5283        DESTROY THE FORCE
5284     ENDLOOP
5285
5286     IF ANALYSIS(6) > 0,
5287        IF END.BATTLE=YES,
5288           LET BATTLE.STATUS="END"
5289        ELSE
5290           LET BATTLE.STATUS="INTR"
5291        ALWAYS
5292        WRITE BATTLE.STATUS,BATTLE,TIME.V,WINNER
5293        AS T 4,S 2,I 7,S 2,D(7,2),S 1,I 2,/  USING UNIT 60 ''                    \TEXT
5294     ALWAYS
5295
5296     REMOVE THE BATTLE FROM THE BATTLE.SET
5297     DESTROY THE BATTLE
5298
5299 <--EXITROUTINE
5300 ENDROUTINE
```

```
5301 ROUTINE ENQ.FEBA.SET GIVEN      ''                                                                    C009
5302    UNIT, OLD.SECTOR YIELDING SECTOR
5303    '' THIS ROUTINE IS CALLED BY UPDATE LOCATION DURING MOVES
5304    ADD 1 TO ANAL.CTR(120,1)  ''                                                                  \DYN_ANAL
5305    NORMALLY MODE IS INTEGER
5306    DEFINE SIDE AS AN INTEGER VARIABLE
5307    IF DEBUG=TRUE,
5308       PRINT 1 LINE WITH UNIT.NOS(UNIT), OLD.SECTOR THUS
5309       = = =ENQ.FEBA.SET UNIT  •••••• IS IN OLD SECTOR ••  = = =
5310    ENDIF
5311    LET SIDE = UN.COLOR(UNIT)
5312    CALL LOCATE.SECTOR  GIVEN UN.Y.COORD(UNIT) YIELDING SECTOR------------------------------------------------------------------------------------>(100)
5313    IF  UN.MISSION( UNIT ) EQ PATROL  '' %9JAN80_%RGR
5314 <------EXITROUTINE   '' THIS UNIT MAY BE BEHIND ENEMY LINES
5315    OTHERWISE    '' IT HAS ONE OF THE OTHER MISSIONS                                  ELSE CHANGED TO OTHERWISE \1
5316    IF SIDE = RED
5317       LET NEW.UN.FWD.ADV = - UN.X.COORD(UNIT)
5318    ELSE  '' THE UNIT'S SIDE IS BLUE
5319       LET NEW.UN.FWD.ADV = UN.X.COORD(UNIT)
5320    ENDIF
5321    IF NEW.UN.FWD.ADV > SS.REAR( SIDE,SECTOR )
5322       IF M.SS.SET(UNIT) NE 1      ''ALREADY IN A SET  %21MAY80_%JLR
5323          FILE THIS UNIT IN THE SS.SET( SIDE,SECTOR )
5324       ENDIF''%21MAY80_%JLR
5325       IF ( SIDE = BLUE   AND   F.SS.SET( SIDE,SECTOR ) = UNIT )
5326          OR
5327          ( SIDE = RED  AND   L.SS.SET( SIDE,SECTOR ) = UNIT )
5328          LET CANDIDATE.REAR = NEW.UN.FWD.ADV - FRONT.DEPTH
5329          IF SS.REAR( SIDE,SECTOR ) < CANDIDATE.REAR
5330             LET SS.REAR( SIDE,SECTOR ) = CANDIDATE.REAR
5331          ENDIF
5332       ENDIF
5333    ENDIF
5334    IF OLD.SECTOR=0
5335 <------EXITROUTINE   '' THE UNIT MAY NOT HAVE BEEN IN A SECTOR
5336    OTHERWISE    '' CHECK TO SEE IF THE FEBA MUST BE RESET
5337    IF THE SS.SET( SIDE,OLD.SECTOR ) IS EMPTY
5338       CALL RESET.FEBA.SECTOR   GIVEN  SIDE AND OLD.SECTOR------------------------------------------------------------------------------------>(119)
5339    ENDIF
5340 ENDROUTINE
```

```
5341 ROUTINE FDC.TR.DEQ   ''                                                                    C010
5342    GIVEN TARGET AND FDC ''%31JAN79_%GLM
5343    ADD 1 TO ANAL.CTR(121,1)  ''                                                            \DYN_ANAL
5344    ''THIS ROUTINE IS CALLED AFTER A TARGET REPORT IS REMOVED
5345    ''FROM A FD.TR.QUEUE(FDC) - IT WORKS WITH FDC.TR.ENQ TO SEE
5346    ''THAT THE FDC IS ALWAYS WORKING ON THE HIGHEST PRIORITY TARGET
5347    DEFINE TARGET AND FDC AS INTEGER VARIABLES
5348    NORMALLY MODE IS INTEGER
5349    LET FD.CUR.TR(FDC) = F.FD.TR.QUEUE(FDC) ''%31JAN79_%GLM
5350    IF FD.TR.QUEUE(FDC) IS EMPTY,
5351 <------EXITROUTINE
5352    OTHERWISE
5353    LET TR.FDC.STATUS(FD.CUR.TR(FDC)) = ACTIVE
5354    CANCEL_THE TARGET.REPORT CALLED FD.CUR.TR(FDC)-------------------------------------------------------------------------------->(470)
5355    REACTIVATE_THE TARGET.REPORT CALLED FD.CUR.TR(FDC) NOW------------------------------------------------------------------------->(470)
5356 ENDROUTINE
```

```
5357 ROUTINE FDC.TR.ENQ                                                    C011
5358    GIVEN
5359    NEW.TARGET,
5360    FDC
5361
5362    ADD 1 TO ANAL CTR(122,1)  ''                                       \DYN_ANAL
5363    ''THIS ROUTINE IS CALLED AFTER A TARGET REPORT IS
5364    ''ADDED TO A FD.TR.QUEUE - IT WORKS WITH FDC.TR.DEQ TO ENSURE
5365    ''THAT THE FDC IS ALWAYS WORKING ON THE PROPER TARGET
5366
5367    NORMALLY MODE IS INTEGER
5368    DEFINE NEW.TARGET, FDC AS INTEGER VARIABLES
5369
5370    IF FD.CUR.TR(FDC) IS ZERO, ''  THE FDC WAS IDLE
5371       LET FD.CUR.TR(FDC) = NEW.TARGET
5372       LET TR.FDC.STATUS(NEW.TARGET) = ACTIVE
5373    ELSE '' THE FDC ALREADY HAS A TARGET
5374       LET TR.FDC.STATUS(NEW.TARGET) = HOLD
5375    ALWAYS
5376
5377 <--EXITROUTINE
5378 ENDROUTINE
```

```
5379 ROUTINE FINISH.COMPUTATION                                                    ''                                                
```

```
5413 ROUTINE FRAC.COMPUTE GIVEN A YIELDING FRAC ''                          C013
5414    ADD 1 TO ANAL.CTR(124,1)  ''                                       \DYN_ANAL
5415    ''
5416    ''THIS ROUTINE COMPUTES A PORTION OF THE FINAL (AND ESTIMATED)
5417    ''COVERAGE OF AN ARTILLERY MISSION
5418    ''
5419    ''
5420    DEFINE A,X,FRAC,DELTA.X AS REAL VARIABLES
5421    LET FRAC = 1.0 + EXP.F(-(A**2)/2.0)  ''                            \OPTIMIZE
5422    LET X = 0.0
5423    LET DELTA.X = A/NUM.ANG.INCREMENTS
5424    FOR I = 2 TO NUM.ANG.INCREMENTS,
5425    DO  ''                                                             \1
5426       ADD (2.0 * EXP.F(-(X**2)/2.0)) TO FRAC  ''                      \OPTIMIZE
5427       ADD DELTA.X TO X
5428    ENDLOOP ''                                                         \1
5429    LET FRAC = A*FRAC*DELTA.X/2.0
5430    ADD EXP.F(-(A**2)/2.0) TO FRAC  ''                                 \OPTIMIZE
5431    LET FRAC = FRAC/SQRT.F(2.0*PI.C)
5432 <--RETURN
5433 END
```

```
5434 ROUTINE GET.TERRAIN        ''                                          C014
5435    YIELDING TER.TYPE.
5436    ADD 1 TO ANAL.CTR(125,1)  ''                                        \DYN_ANAL
5437
5438
5439    ''UNUSED ARGUMENTS DELETED %10JUL79_%JLR
5440
5441    NORMALLY MODE IS INTEGER
5442    IF TERRAIN.PAR EQ FULDA
5443       LET TER.TYPE = RANDI.F(1,5,RN.SEED)  ''%10JUL79_%JLR  CORRECTION:
5444    ELSE                                 ''     FULDA IS TERRAIN TYPES 1 THRU 5
5445       IF TERRAIN.PAR EQ NO.GER.PLAIN
5446          LET TER.TYPE = RANDI.F(6,11,RN.SEED)
5447       ELSE
5448          IF TERRAIN.PAR EQ MIX.TERRAIN
5449             LET TER.TYPE = RANDI.F(1,11,RN.SEED)
5450          ELSE
5451             LET TER.TYPE = TERRAIN.PAR
5452          ALWAYS
5453       ALWAYS
5454    ALWAYS
5455 <--EXITROUTINE
5456 ENDROUTINE
```

```
5457 ROUTINE HC.EMPTY                                                                                                    C015
5458    GIVEN
5459    HE.FARRP,
5460    HE.SIDE
5461
5462    ADD 1 TO ANAL.CTR(126,1)  ''                                                                                     \DYN_ANAL
5463    ''THIS ROUTINE PERFORMS THE HOUSECLEANING OPERATIONS NECESSARY
5464    ''TO DISENGAGE HELICOPTERS FOR ONE SIDE FROM A BATTLE
5465
5466    NORMALLY MODE IS INTEGER
5467
5468    IF HC.DEBUG = "YES"
5469       PRINT 1 LINE WITH HE.FARRP, HE.SIDE, FP.BATTLE(HE.FARRP)
5470       AS FOLLOWS
5471       ----HC.EMPTY----  FARRP=******  SIDE=***  BATTLE=******
5472    ALWAYS
5473
5474    LOOP FOR EVERY HC.RETURN.FARRP OF EV.S(I.HC.RETURN.FARRP)
5475    WITH RF.FARRP(HC.RETURN.FARRP) = HE.FARRP
5476    DO THE FOLLOWING
5477       LET HT.STATUS(RF.TEAM(HC.RETURN.FARRP)) = RELEASED ''SCHEDULED PROCESS
5478       LET HT.TERMINATOR(RF.TEAM(HC.RETURN.FARRP)) = YES
5479       IF HC.DEBUG = "YES"
5480          PRINT 1 LINE WITH RF.TEAM(HC.RETURN.FARRP)
5481          AS FOLLOWS
5482          ----HC.EMPTY----  RF.TEAM=******
5483       ALWAYS
5484    ENDLOOP
5485    LOOP FOR EVERY SEND.TEAM OF EV.S(I.SEND.TEAM)
5486    WITH ST.FARRP(SEND.TEAM) = HE.FARRP
5487    DO THE FOLLOWING
5488       ACTIVATE_AN HC.RETURN.FARRP CALLED HRF  ''HT.STATUS WILL BE READY------------------------------------------>(450)
5489       GIVEN
5490       ST.FARRP(SEND.TEAM),
5491       ST.TEAM(SEND.TEAM) NOW ''                                                                                     \1
5492       LET HT.TERMINATOR(ST.TEAM(SEND.TEAM)) = YES
5493       IF HC.DEBUG = "YES"
5494          PRINT 1 LINE WITH ST.TEAM(SEND.TEAM) AS FOLLOWS
5495          ----HC.EMPTY----  ST.TEAM=******
5496       ALWAYS
5497       CANCEL_THE SEND.TEAM------------------------------------------------------------------------------------->(391)
5498       DESTROY THE SEND.TEAM
5499    ENDLOOP
5500    LOOP FOR EVERY HC.ARRIVE.BATTLE IN THE EV.S(I.HC.ARRIVE.BATTLE)
5501    WITH AB.FARRP(HC.ARRIVE.BATTLE) = HE.FARRP
5502    DO THE FOLLOWING
5503       LET MOVE.TIME = HT.MOVE.TIME(AB.TEAM(HC.ARRIVE.BATTLE))
5504       LET HT.STATUS(AB.TEAM(HC.ARRIVE.BATTLE)) = RELEASED
5505       LET HT.TERMINATOR(AB.TEAM(HC.ARRIVE.BATTLE)) = YES
5506       ACTIVATE_AN HC.RETURN.FARRP------------------------------------------------------------------------------->(450)
5507       GIVEN
5508       AB.FARRP(HC.ARRIVE.BATTLE),
5509       AB.TEAM(HC.ARRIVE.BATTLE)
5510       IN MOVE.TIME MINUTES
5511       INTERRUPT HC.ARRIVE.BATTLE------------------------------------------------------------------------------->(443)
5512       LET TIME.A(HC.ARRIVE.BATTLE) = -RINF.C
5513       RESUME HC.ARRIVE.BATTLE---------------------------------------------------------------------------------->(443)
5514       IF HC.DEBUG = "YES"
```

```
5515          PRINT 1 LINE WITH AB.TEAM(HC.ARRIVE.BATTLE)
5516          AS FOLLOWS
5517          ----HC.EMPTY----  AB.TEAM=******
5518       ALWAYS
5519    ENDLOOP
5520    LOOP FOR EVERY HEL.TARGET.ACQUISITION IN EV.S(I.HEL.TARGET.ACQUISITION)
5521    WITH HT.FARRP(HTA.TEAM(HEL.TARGET.ACQUISITION)) = HE.FARRP
5522    DO THIS
5523       LET HT.STATUS(HTA.TEAM(HEL.TARGET.ACQUISITION)) = RELEASED
5524       INTERRUPT HEL.TARGET.ACQUISITION------------------------------------------------------------------------------>(454)
5525       LET HT.TERMINATOR(HTA.TEAM(HEL.TARGET.ACQUISITION)) = YES
5526       LET TIME.A(HEL.TARGET.ACQUISITION) = -RINF.C
5527       RESUME HEL.TARGET.ACQUISITION---------------------------------------------------------------------------->(454)
5528
5529       IF HC.DEBUG = "YES"
5530          PRINT 1 LINE WITH HTA.TEAM(HEL.TARGET.ACQUISITION),
5531          ENEMY.UNITS(HEL.TARGET.ACQUISITION) AS FOLLOWS
5532          ----HC.EMPTY----  HTA.TEAM=******  ENEMY.UNITS=******
5533       ALWAYS
5534
5535       CALL HC.DISENGAGE---------------------------------------------------------------------------------------->(300)
5536       GIVING
5537       HE.FARRP,
5538       HTA.TEAM(HEL.TARGET.ACQUISITION),
5539       ENEMY.UNITS(HEL.TARGET.ACQUISITION)
5540
5541 <-----''RETURN TEAM TO FARRP
5542       LET FLIGHT.TIME = HT.MOVE.TIME(HTA.TEAM(HEL.TARGET.ACQUISITION))
5543       ACTIVATE_AN HC.RETURN.FARRP------------------------------------------------------------------------------>(450)
5544       GIVEN
5545       HE.FARRP,
5546       HTA.TEAM(HEL.TARGET.ACQUISITION)
5547       IN FLIGHT.TIME MINUTES
5548       IF HT.TERMINATOR(HTA.TEAM(HEL.TARGET.ACQUISITION)) NE YES
5549          LET HT.STATUS(HTA.TEAM(HEL.TARGET.ACQUISITION)) =
5550          RETURNING.FROM.BATTLE
5551       ALWAYS
5552    ENDLOOP
5553
5554    LOOP FOR EVERY HELO.ENGAGEMENT IN EV.S(I.HELO.ENGAGEMENT)
5555    WITH HT.FARRP(HCEN.TEAM(HELO.ENGAGEMENT)) = HE.FARRP
5556    DO THE FOLLOWING
5557       CANCEL_THE HELO.ENGAGEMENT------------------------------------------------------------------------------>(376)
5558       DESTROY THE HELO.ENGAGEMENT
5559    ENDLOOP
5560
5561    LOOP FOR EVERY HELICOPTER.FIRE IN EV.S(I.HELICOPTER.FIRE)
5562    WITH HT.FARRP(HF.TEAM(HELICOPTER.FIRE)) = HE.FARRP
5563    DO THE FOLLOWING
5564       LET HF.DESTRUCT.INDIC(HELICOPTER.FIRE) = YES
5565    ENDLOOP
5566
5567    LET UN.BATTLE.INDEX(FP.UNIT(HE.FARRP)) = 0
5568    LOOP FOR EVERY HC.DEPART.BATTLE IN THE EV.S(I.HC.DEPART.BATTLE)
5569    WITH DB.FARRP(HC.DEPART.BATTLE) = HE.FARRP
5570    DO THIS
5571       LET HT.STATUS(DB.TEAM(HC.DEPART.BATTLE)) = RELEASED
5572       LET HT.TERMINATOR(DB.TEAM(HC.DEPART.BATTLE)) = YES
```

```
5573        IF HC.DEBUG = "YES"
5574           PRINT 1 LINE WITH DB.TEAM(HC.DEPART.BATTLE),
5575           DB.ENEMY.UNITS(HC.DEPART.BATTLE) AS FOLLOWS
5576           ---- HC.EMPTY ----  DB.TEAM=******  ENEMY.UNITS=*******
5577        ALWAYS
5578        CALL HC.DISENGAGE------------------------------------------------------------------------------------------------------>(300)
5579        GIVING
5580        HE.FARRP,
5581        DB.TEAM(HC.DEPART.BATTLE),
5582        DB.ENEMY.UNITS(HC.DEPART.BATTLE)
5583 <------''RETURN TEAM TO FARRP
5584        LET FLIGHT.TIME = HT.MOVE.TIME(DB.TEAM(HC.DEPART.BATTLE))
5585        ACTIVATE_AN HC.RETURN.FARRP-------------------------------------------------------------------------------------------->(450)
5586        GIVEN
5587        HE.FARRP,
5588        DB.TEAM(HC.DEPART.BATTLE)
5589        IN FLIGHT.TIME MINUTES
5590        IF HT.TERMINATOR(DB.TEAM(HC.DEPART.BATTLE)) NE YES
5591           LET HT.STATUS(DB.TEAM(HC.DEPART.BATTLE)) =
5592           RETURNING.FROM.BATTLE
5593        ALWAYS
5594        CANCEL_THE HC.DEPART.BATTLE-------------------------------------------------------------------------------------------->(375)
5595        DESTROY THE HC.DEPART.BATTLE
5596     ENDLOOP
5597
5598 <--EXITROUTINE
5599 ENDROUTINE
```

```
5600 ROUTINE PROB.INF                                                    ''      C016
5601     GIVEN
5602     .NO.BARS,
5603     .FREQUENCY,
5604     .RANGE,
5605     .FRAC.TARGET.VISIBLE,
5606     .DETECTED.TE
5607
5608     YIELDING
5609     .PROBABILITY.INF
5610
5611     ADD 1 TO ANAL.CTR(127,1)  ''                                        \DYN_ANAL
5612     NORMALLY MODE IS INTEGER
5613     DEFINE .FREQUENCY, .PROBABILITY.INF, .N, .E,
5614     .FRAC.TARGET.VISIBLE, .NO.BARS
5615     AS REAL VARIABLES
5616
5617     IF .RANGE EQ 0
5618        LET .PROBABILITY.INF = 1.0
5619        LET .N = 30.0
5620     ELSE
5621        LET .N = .FREQUENCY * .FRAC.TARGET.VISIBLE
5622        * (TE.HEIGHT(.DETECTED.TE)/10) / (.RANGE/1000)
5623        LET .E = 2.7 + 0.7 * (.N / .NO.BARS)
5624        IF .E GT 20.0 OR .N GT 64.0
5625           LET .PROBABILITY.INF = 1.0
5626        ELSE
5627           LET .PROBABILITY.INF = ((.N / .NO.BARS) **.E)
5628           / (1.0 + (.N / .NO.BARS) **.E)
5629        ALWAYS
5630     ALWAYS
5631
5632 <--RETURN
5633 END
```

```
5634 ROUTINE PROB.TIME                                                    C017
5635    GIVEN
5636    .DETECTING.WPN,
5637    .PROB.INF,
5638    .NUM.SEARCHES
5639
5640    YIELDING
5641    .TIME.TO.DETECT
5642
5643    ADD 1 TO ANAL.CTR(128,1)  ''                                      \DYN_ANAL
5644    NORMALLY MODE IS INTEGER
5645    DEFINE .PROB.INF, .TIME.TO.DETECT, .PROB.TOTAL, .NFOV,
5646    .T.AVERAGE, .TIME.TO.ACQUIRE AS REAL VARIABLES
5647
5648    LET .TIME.TO.DETECT = RINF.C
5649
5650    LET .PROB.TOTAL = RANDOM.F(RN.SEED)
5651    IF .PROB.INF GT .PROB.TOTAL
5652       LET .NFOV = TW.HFOS(.DETECTING.WPN) * TW.VFOS(.DETECTING.WPN)
5653       / (TW.HFOV(.DETECTING.WPN) * TW.VFOV(.DETECTING.WPN) / 100)
5654       LET .T.AVERAGE = 3.4 * .NFOV / (.PROB.INF * .NUM.SEARCHES)
5655       LET .TIME.TO.ACQUIRE = -.T.AVERAGE
5656       * LOG.E.F(1.0 - (.PROB.TOTAL / .PROB.INF))
5657       LET .TIME.TO.DETECT = MAX.F(5.0, .TIME.TO.ACQUIRE)
5658    ALWAYS
5659
5660 <--RETURN
5661 END
```

```
5662 ROUTINE RANGE.COMPUTE                                                    ''                C018
5663     GIVEN UNIT.A AND UNIT.B
5664     YIELDING RANGE
5665
5666     ADD 1 TO ANAL.CTR(129,1)  ''                                                          \DYN_ANAL
5667     DEFINE D.X., D.Y. AS REAL VARIABLES
5668     DEFINE UNIT.A, UNIT.B, AND RANGE AS INTEGER VARIABLES
5669
5670     LET D.X. = UN.X.COORD(UNIT.A)-UN.X.COORD(UNIT.B)
5671     LET D.Y. = UN.Y.COORD(UNIT.A)-UN.Y.COORD(UNIT.B)
5672     LET RANGE = SQRT.F(D.X.*D.X. + D.Y.*D.Y.)   ''                                        \OPTIMIZE
5673
5674 <--RETURN
5675 END
```

```
5676 ROUTINE SEARCH.COVERAGE                                                    C019
5677    GIVEN
5678    UN,
5679    HALF.WIDTH,
5680    DISTANCE,
5681    LEG.SLOPE,
5682    LEG.Y.INTERCEPT, AND TGT.RADIUS
5683    YIELDING FRACT.COVERED
5684    ADD 1 TO ANAL.CTR(130,1)  ''                                          \DYN_ANAL
5685    NORMALLY MODE IS REAL  ''                                             \OPTIMIZE
5686    LET DELTA.INTERCEPT = SQRT.F((LEG.SLOPE*HALF.WIDTH)**2+HALF.WIDTH**2)
5687    LET S1.X.PERPEND = (UN.Y.COORD(UN) + UN.X.COORD(UN)/LEG.SLOPE -
5688    (LEG.Y.INTERCEPT+DELTA.INTERCEPT)) / (LEG.SLOPE + 1./LEG.SLOPE)
5689    LET S2.X.PERPEND = (UN.Y.COORD(UN) + UN.X.COORD(UN)/LEG.SLOPE -
5690    (LEG.Y.INTERCEPT-DELTA.INTERCEPT)) / (LEG.SLOPE + 1./LEG.SLOPE)
5691    LET S1.Y.PERPEND = LEG.SLOPE*S1.X.PERPEND+(LEG.Y.INTERCEPT+DELTA.INTERCEPT)
5692    LET S2.Y.PERPEND = LEG.SLOPE*S2.X.PERPEND+(LEG.Y.INTERCEPT-DELTA.INTERCEPT)
5693    LET S1 = 10. * SQRT.F((S1.X.PERPEND -
5694    UN.X.COORD(UN))**2.+(S1.Y.PERPEND - UN.Y.COORD(UN))**2 )
5695    LET S2 = 10. * SQRT.F((S2.X.PERPEND -
5696    UN.X.COORD(UN))**2.+(S2.Y.PERPEND - UN.Y.COORD(UN))**2 )
5697    IF S1 < S2
5698       LET EDGE1.DIST = S1
5699       LET EDGE2.DIST = S2
5700    ELSE
5701       LET EDGE1.DIST = S2
5702       LET EDGE2.DIST = S1
5703    ENDIF
5704    LET TGT.RAD.SQ = TGT.RADIUS ** 2
5705    LET AREA1=S1*SQRT.F(TGT.RAD.SQ-S1**2)-TGT.RAD.SQ*ARCCOS.F(S1/TGT.RADIUS)
5706    LET AREA2=S2*SQRT.F(TGT.RAD.SQ-S2**2)-TGT.RAD.SQ*ARCCOS.F(S2/TGT.RADIUS)
5707    LET UNIT.AREA = PI.C*TGT.RAD.SQ
5708    IF DISTANCE > HALF.WIDTH  ''CENTER OF UNIT IS OUTSIDE OF SEARCH AREA
5709       LET AREA.COVERED = AREA1
5710       IF TGT.RADIUS > DISTANCE + HALF.WIDTH
5711          SUBTRACT AREA2 FROM AREA.COVERED
5712       ELSE
5713          IF TGT.RADIUS <= DISTANCE - HALF.WIDTH
5714             SUBTRACT AREA1 FROM AREA.COVERED
5715          ENDIF
5716       ENDIF
5717    ELSE   ''CENTER OF UNIT IS INSIDE SEARCH BAND
5718       LET AREA.COVERED = UNIT.AREA
5719       IF TGT.RADIUS > HALF.WIDTH
5720          SUBTRACT AREA1 FROM AREA.COVERED
5721          IF TGT.RADIUS > HALF.WIDTH + DISTANCE
5722             SUBTRACT AREA2 FROM AREA.COVERED
5723          ENDIF
5724       ELSE
5725          IF TGT.RADIUS > HALF.WIDTH - DISTANCE
5726             SUBTRACT AREA1 FROM AREA.COVERED
5727          ENDIF
5728       ENDIF
5729    ENDIF
5730    LET FRACT.COVERED = AREA.COVERED / UNIT.AREA
5731 ENDROUTINE
```

```
5732 ROUTINE TEMPERATURE.ATTENUATION                                    ''
5733     GIVEN                                                                  C020
5734     .DETECTED.TE,
5735     .RANGE
5736
5737     YIELDING
5738     .DELTA.T.PRIME
5739
5740     ADD 1 TO ANAL.CTR(131,1)  ''                                          \DYN_ANAL
5741     NORMALLY MODE IS INTEGER
5742     DEFINE .DELTA.T.PRIME, .SIGMA.55, .EXTINCTION.COEF,
5743     .TOTAL.XMIS, .GAP.XMIS, .ABSORPTION.XMIS
5744     AS REAL VARIABLES
5745
5746     LET .ABSORPTION.XMIS = EXP.F(-.1 * (.RANGE/1000.))  ''                \OPTIMIZE
5747     LET .SIGMA.55 = 3.912 / (16 * VISIBILITY / 1000.)
5748     IF .SIGMA.55 LT .98
5749        LET .EXTINCTION.COEF = 0.0
5750     ELSE
5751        IF .SIGMA.55 GT 7.82
5752           LET .EXTINCTION.COEF = 20.0
5753        ELSE
5754           LET .EXTINCTION.COEF = 0.14 + 10 **(-0.980
5755           + 1.851 * LOG.10.F(.SIGMA.55)
5756           - 0.212 * (LOG.10.F(.SIGMA.55)) **2)
5757        ALWAYS
5758     ALWAYS
5759     LET .GAP.XMIS = EXP.F(-.EXTINCTION.COEF * (.RANGE/1000.))
5760     LET .TOTAL.XMIS = .ABSORPTION.XMIS * .GAP.XMIS
5761     LET .DELTA.T.PRIME = .TOTAL.XMIS * TE.DELTA.T(.DETECTED.TE) / 10.
5762
5763 <--RETURN
5764 END
```

```
5765 ROUTINE TERM CHECK                                                                                    C021
5766    GIVEN
5767    CHECK.UNIT
5768
5769    ADD 1 TO ANAL.CTR(132,1)  ''                                                                       \DYN_ANAL
5770    NORMALLY MODE IS INTEGER
5771
5772    FOR EVERY FORCE OF THE BTL.FORCE.SET(UN.BATTLE.INDEX(CHECK.UNIT))
5773    WITH FR.SIDE(FORCE) = UN.COLOR(CHECK.UNIT)
5774    FIND THE FIRST CASE
5775    IF NONE
5776       CALL ERROR.STOP------------------------------------------------------------------------------>(604)
5777    ALWAYS
5778    IF DECISION.POINT(FORCE) = -1
5779 <------EXITROUTINE
5780    OTHERWISE
5781
5782    FOR EVERY UNIT OF FR.UNIT.SET(FORCE)
5783    WITH UN.POSITION.INDEX(UNIT) NE 1  OR
5784    UN.STATUS(UNIT) NE STATIONARY
5785    FIND THE FIRST CASE
5786    IF FOUND,
5787 <------EXITROUTINE
5788    OTHERWISE
5789
5790    CALL FIN.BATTLE------------------------------------------------------------------------------>(145)
5791    GIVEN
5792    UN.BATTLE.INDEX(CHECK.UNIT),
5793    FORCE
5794
5795 <--EXITROUTINE
5796 ENDROUTINE
```

EVENTS

```
5803 EVENT ACT.REINF                                                                                    E001
5804    GIVEN
5805    R.UNIT,
5806    ORDR.,
5807    EN.UNITS.
5808
5809    ADD 1 TO ANAL.CTR(133,1)  ''                                                                    \DYN_ANAL
5810    ''THIS ROUTINE IS EXECUTED WHEN A UNIT'S CONTINGENCY ORDERS DIRECT
5811    ''     A CALL FOR REINFORCEMENTS.  A SEARCH IS MADE FOR A UNIT OR
5812    ''     TASK FORCE APPROPRIATE TO REINFORCE THE SUBJECT UNIT.  TO BE
5813    ''     APPROPRIATE A UNIT MUST BELONG TO THE SAME PARENT, NOT BE IN
5814    ''     CLOSE PROXIMITY TO AN ENEMY UNIT, NOT BE ENGAGED IN CLOSE
5815    ''     COMBAT,  NOT BE BADLY ATTRITTED, AND MUST HAVE BEEN DESIGNATED
5816    ''     AS AVAILABLE FOR REINFORCEMENT.  IF A UNIT IS FOUND TO REINFORCE
5817    ''     THE REINFORCEMENT ORDER IS INITIATED.  IF NOT, THE SUBJECT UNIT
5818    ''     MUST TAKE ON THE CONTINGENCY ORDER FOR "NO REINFORCEMENTS
5819    ''     AVAILABLE".
5820    ''     R.UNIT IS THE REINFORCED UNIT.
5821
5822    NORMALLY MODE IS INTEGER
5823    DEFINE UNIT AS A VARIABLE
5824    DEFINE BATTLE.NUM AS A VARIABLE
5825    DEFINE EN.UNITS., PROX.UNITS. AS 1-DIMENSIONAL ARRAYS
5826
5827    SKIP 1 LINE
5828    PRINT 1 LINE WITH UNIT.NOS(R.UNIT),ORD.SEQ.NO(ORDR.),TIME.V THUS
5829    REINFORCEMENTS REQUESTED FOR UNIT ****** , ORDER ***** AT ***.* HOURS
5830
5831    LET UNIT.HQ = UN.PARENT(R.UNIT)
5832
5833    IF UNIT.HQ IS GREATER THAN 0
5834       LOOP FOR EACH UNIT IN UN.SUB.LIST(UNIT.HQ)
5835       WITH UNIT.NOS(UNIT) NE UNIT.NOS(R.UNIT) AND
5836       MU.REINF.IND(UN.PTR(UNIT)) EQ YES AND
5837       MU.TF.MEM(UN.PTR(UNIT)) IS GREATER THAN 0 AND
5838       UN.BATTLE.INDEX(UNIT) EQ 0
5839       DO
5840          CALL LOCATE.SECTOR----------------------------------------------------------------------->(100)
5841          GIVEN
5842          UN.Y.COORD(UNIT)
5843          YIELDING
5844          UNIT.SEC.
5845          CALL CHECK.PROX-------------------------------------------------------------------------->(138)
5846          GIVEN
5847          UNIT,
5848          UNIT.SEC.,
5849          REIN.PROX
5850          YIELDING
5851          PROX.TEST.,
5852          PROX.UNITS.(*),
5853          BATTLE.NUM
5854
5855          RELEASE PROX.UNITS.(*)
5856
5857          IF PROX.TEST. = NO
5858             ''THERE IS NO ENEMY UNIT NEAR
5859             CALL CHECK.STREN------------------------------------------------------------------->(141)
5860             GIVEN
```

```
5861              UNIT
5862              YIELDING
5863              PCT.ON.HAND
5864           THEN IF PCT.ON.HAND GE REIN.THRESH
5865
5866              PRINT 1 LINE WITH UNIT.NOS(UNIT), UNIT.NOS(R.UNIT)THUS
5867              UNIT •••••• WILL REINFORCE UNIT ••••
5868              SKIP 1 LINE
5869
5870              CALL INIT.REINF------------------------------------------------------------------------------------------------------>( 95)
5871              GIVEN
5872              R.UNIT,
5873              UNIT,
5874              ORDR.,
5875              EN.UNITS.(•)
5876 <------------EXITEVENT  ''ONLY THE FIRST AVAILABLE UNIT WILL REINF
5877           OTHERWISE
5878        ENDLOOP
5879     ALWAYS
5880
5881     LET NEXT.=UNSUC.REINF.OP(ORD.ID(ORDR.))
5882
5883     PRINT 1 LINE WITH UNIT.NOS(R.UNIT) THUS
5884     NO UNITS AVAILABLE FOR REINFORCEMENT OF UNIT ••••••
5885     SKIP 1 LINE
5886
5887     LET FLAG = 1
5888     SCHEDULE_A GET.NX.ORD------------------------------------------------------------------------------------------------------>(373)
5889     GIVEN
5890     R.UNIT,
5891     NEXT.,
5892     EN.UNITS.(•),
5893     FLAG NOW ''                                                                                                              \1
5894
5895 <--EXITEVENT
5896 ENDEVENT
```

```
5897 EVENT AD.ENGAGEMENT                                                    ''      E002
5898    GIVEN
5899    .INTERSECT,
5900    .CALLING.PROCESS
5901
5902    ADD 1 TO ANAL.CTR(134,1)  ''                                                \DYN_ANAL
5903    NORMALLY MODE IS INTEGER
5904    DEFINE .INTERCEPT, .SLOPE, .PD, .MU, .SIGMA, .TRAVEL.TIME, .A,.B,.C,
5905    .DET.TIME, .XVEL, .YVEL, .FIRING.TIME
5906    AS REAL VARIABLES
5907    DEFINE .TYPE.PROCESS, .MADS.NAME AS TEXT VARIABLES ''                       \TEXT
5908
5909    IF TACAIR.DEBUG = 1
5910       PRINT 1 LINE WITH TIME.V THUS
5911       = = = AD.ENGAGEMENT AT ***.***
5912       LIST ATTRIBUTES OF SENSOR.INTERSECT CALLED .INTERSECT
5913       PRINT 1 LINE WITH .CALLING.PROCESS THUS
5914       CALLING PROCESS IS ************
5915       LIST ATTRIBUTES OF CAS.MISSION CALLED .CALLING.PROCESS
5916    ALWAYS
5917    LET .AD.UNIT = SI.AD.UNIT(.INTERSECT)
5918    LET .X0 = UN.X.COORD(.AD.UNIT)
5919    LET .Y0 = UN.Y.COORD(.AD.UNIT)
5920
5921    ''FIND THE POINT AT WHICH THE AIRCRAFT PATH PASSES CLOSEST TO
5922    ''THE SENSOR AND DETERMINE THE PROBABILITY OF DETECTION THERE.
5923    LET .HIGHX = MAX.F(SI.X.ENTRY(.INTERSECT), SI.X.EXIT(.INTERSECT))
5924    LET .LOWX  = MIN.F(SI.X.ENTRY(.INTERSECT), SI.X.EXIT(.INTERSECT))
5925    LET .HIGHY = MAX.F(SI.Y.ENTRY(.INTERSECT), SI.Y.EXIT(.INTERSECT))
5926    LET .LOWY  = MIN.F(SI.Y.ENTRY(.INTERSECT), SI.Y.EXIT(.INTERSECT))
5927    LET .DELTAX = SI.X.EXIT(.INTERSECT) - SI.X.ENTRY(.INTERSECT)
5928    LET .DELTAY = SI.Y.EXIT(.INTERSECT) - SI.Y.ENTRY(.INTERSECT)
5929    IF .DELTAX = 0
5930       LET .X = SI.X.ENTRY(.INTERSECT)
5931       LET .Y = UN.Y.COORD(.AD.UNIT)
5932    ELSE
5933       LET .SLOPE = .DELTAY / .DELTAX
5934       LET .INTERCEPT = SI.Y.ENTRY(.INTERSECT)
5935       - .SLOPE * SI.X.ENTRY(.INTERSECT)
5936       LET .X = (UN.X.COORD(.AD.UNIT) + .SLOPE * UN.Y.COORD(.AD.UNIT)
5937       - .SLOPE * .INTERCEPT) / (.SLOPE**2 + 1) ''                              \OPTIMIZE
5938       LET .Y = .SLOPE * .X + .INTERCEPT
5939    ALWAYS
5940
5941    ''WHEN THIS POINT IS NOT BETWEEN THE ENTRY AND EXIT, CHOOSE THE                \1
5942    ''SEGMENT END NEAREST THE SENSOR.
5943    IF .LOWX GT .X OR .HIGHX LT .X OR
5944       .LOWY GT .Y OR .HIGHY LT .Y
5945       IF ABS.F(SI.X.ENTRY(.INTERSECT) - UN.X.COORD(.AD.UNIT)) LE
5946          ABS.F(SI.X.EXIT(.INTERSECT) - UN.X.COORD(.AD.UNIT)) AND
5947          ABS.F(SI.Y.ENTRY(.INTERSECT) - UN.Y.COORD(.AD.UNIT)) LE
5948          ABS.F(SI.Y.EXIT(.INTERSECT) - UN.Y.COORD(.AD.UNIT))
5949          LET .X = SI.X.ENTRY(.INTERSECT)
5950          LET .Y = SI.Y.ENTRY(.INTERSECT)
5951       ELSE
5952          LET .X = SI.X.EXIT(.INTERSECT)
5953          LET .Y = SI.Y.EXIT(.INTERSECT)
5954       ALWAYS
```

```
5955     ALWAYS
5956     LET .RANGE = SQRT.F((REAL.F((.X - UN.X.COORD(.AD.UNIT))**2
5957     + (.Y - UN.Y.COORD(.AD.UNIT))**2)))
5958     ''.RANGE = CLOSEST DISTANCE BETWEEN AIRCRAFT AND SENSOR
5959
5960     ''FIND THE MODEL AD SENSOR.
5961     FOR EVERY .ADS IN SD.ADS.SET(UN.COLOR(.AD.UNIT))
5962     WITH ADS.UNIT.PTR(.ADS) = .AD.UNIT
5963     FIND THE FIRST CASE
5964     IF NONE
5965        TRACE
5966 <------STOP
5967     OTHERWISE
5968
5969     IF ADS.NR.SENSORS(.ADS) LE 0
5970        ''ALL OF THE SENSORS HAVE BEEN DESTROYED.
5971 <------RETURN
5972     OTHERWISE
5973
5974     LET .MADS.NAME = MADS.NAME(ADS.MADS.PTR(.ADS))
5975     FOR EVERY .RH IN MADS.RH.SET(ADS.MADS.PTR(.ADS))
5976     WITH MRH.RANGE(.RH) GE .RANGE
5977     FIND THE FIRST CASE
5978     IF NONE
5979        TRACE
5980 <------STOP
5981     OTHERWISE
5982
5983     IF TACAIR.DEBUG = 1
5984        PRINT 1 LINE WITH MRH.RANGE(.RH) AND .RANGE THUS
5985        ######## AD SYSTEM RANGE IS ******, THE RANGE TO THE AIRCRAFT IS ****** ####
5986     ALWAYS
5987     IF SI.ALTITUDE(.INTERSECT) LT MRH.MIN.ALT(.RH)
5988        ''AIRCRAFT TOO LOW TO BE DETECTED.
5989 <------RETURN
5990     OTHERWISE
5991
5992     IF MADS.DETECT(ADS.MADS.PTR(.ADS)) EQ 1 ''CONSIDERS 120 DEGREE
5993        IF RANDOM.F(RN.SEED) GT .667 ''SECTOR FOR PATRIOT
5994 <---------RETURN
5995        ALWAYS
5996        IF TACAIR.DEBUG = 1
5997           PRINT 1 LINE WITH MADS.NAME(ADS.MADS.PTR(.ADS)) THUS
5998           [[[[[[PATRIOT MODELLED AS ****** IS REDUCED TO 120 DEGREES]]]]]]]]]]]]]]
5999        ALWAYS
6000     ALWAYS
6001     LET .PD = (MRH.PD(.RH) / 100) * (MADS.XMIT.PCT(ADS.MADS.PTR(.ADS)) / 100)
6002     IF VISIBILITY LT SD.POOR.FLY.VIS(UN.COLOR(.AD.UNIT))
6003        LET .PD = .PD * (MADS.PW.DEGRADE(ADS.MADS.PTR(.ADS)) / 100)
6004     ALWAYS
6005     LET .PD = 1.0 - (1.0 - .PD)**ADS.NR.SENSORS(.ADS)
6006
6007     ''DETERMINE THE NUMBER OF AIRCRAFT IN THE MISSION.
6008     FOR EACH .PROCESS IN EV.S(I.CAS.MISSION)
6009     WITH .PROCESS = .CALLING.PROCESS
6010     FIND THE FIRST CASE
6011     IF FOUND
6012        ''THIS IS A CAS MISSION.
```

```
6013        LET .NUM.AC = CMSN.NR.SURV.AC(.PROCESS)
6014        LET .VELOCITY = ACT.SPEED(CMSN.AC.TYPE(.PROCESS)) * 3600
6015        LET .TYPE.PROCESS = "CAS"
6016     ELSE
6017        ''THIS IS AN AO MISSION.
6018        LET .NUM.AC = 1
6019        LET .VELOCITY = AO.VELOCITY(.CALLING.PROCESS) * 3600
6020        LET .TYPE.PROCESS = "AO"
6021     ALWAYS
6022
6023     ''DETERMINE THE NUMBER OF AIRCRAFT DETECTED.
6024     LET .NUM.DET = BINOMIAL.F(.NUM.AC, .PD, RN.SEED)
6025     IF TACAIR.DEBUG = 1
6026        PRINT 1 LINE WITH .NUM.DET THUS
6027        ##########THE NUMBER OF AIRCRAFT DETECTED IS **** ################
6028     ALWAYS
6029     IF .NUM.DET = 0
6030        ''NONE OF THE AIRCRAFT ARE DETECTED.
6031 <-------RETURN
6032     OTHERWISE
6033
6034     ''DETERMINE WHEN THE AIRCRAFT WILL BE DETECTED.
6035     LET .TRAVEL.DIST = SQRT.F((REAL.F(
6036     (SI.X.EXIT(.INTERSECT) - SI.X.ENTRY(.INTERSECT))**2
6037     + (SI.Y.EXIT(.INTERSECT) - SI.Y.ENTRY(.INTERSECT))**2)))
6038     LET .TRAVEL.TIME = .TRAVEL.DIST / .VELOCITY
6039     LET .MU = MADS.DELAY.TIME(ADS.MADS.PTR(.ADS))/3600.
6040     LET .SIGMA = .MU / 4
6041     ''UTILIZE NORMAL.F FUNCTION IN NEXT DRAW ------------------------------------------------------------------------------ \1>(641)
6042     LET .DET.TIME = NORMAL.F(.MU, .SIGMA, RN.SEED)
6043     LET .DET.TIME = MAX.F(.DET.TIME, 0.)
6044     LET .DET.TIME = MIN.F(.DET.TIME, .TRAVEL.TIME)
6045     LET .FIRING.TIME = MAX.F((UN.AD.AVAIL(.AD.UNIT)-TIME.V),
6046     .DET.TIME + MADS.DELAY.TIME(ADS.MADS.PTR(.ADS))/3600.)
6047     IF TACAIR.DEBUG = 1
6048        PRINT 1 LINE WITH .TRAVEL.TIME, AND .DET.TIME THUS
6049        #########AIRCRAFT TRANSIT TIME IS *.****, AD DETECT TIME IS *.**** #####
6050     ALWAYS
6051     IF .FIRING.TIME GT .TRAVEL.TIME
6052        IF TACAIR.DEBUG = 1
6053           PRINT 1 LINE AS FOLLOWS
6054           ##########FIRING TIME GREATER THAN TRAVEL TIME, NO SHOT POSSIBLE#######
6055        ALWAYS
6056 <-------RETURN
6057     OTHERWISE
6058
6059     ''COMPUTE THE RANGE AT .FIRING.TIME. THIS RANGE CONSIDERS
6060     ''THE ALTITUDE.
6061     LET .XVEL = .VELOCITY * .DELTAX / .TRAVEL.DIST
6062     LET .YVEL = .VELOCITY * .DELTAY / .TRAVEL.DIST
6063     LET .RANGE = SQRT.F((REAL.F(
6064     (SI.X.ENTRY(.INTERSECT)+.XVEL*.FIRING.TIME
6065     - UN.X.COORD(.AD.UNIT))**2 +
6066     (SI.Y.ENTRY(.INTERSECT)+.YVEL*.FIRING.TIME
6067     -UN.Y.COORD(.AD.UNIT))**2 +
6068     SI.ALTITUDE(.INTERSECT)**2)))
6069
6070     ''WHEN THE AIRCRAFT ARE NOT YET IN RANGE OF THE WEAPON,                                                    \1
```

```
6071     ''DELAY THE FIRING.
6072     FOR EACH .EQ IN UN.EQUIP.LIST(.AD.UNIT)
6073     WITH EQ.AD.INDICATOR(UE.ID(.EQ)) NE 0
6074     FIND THE FIRST CASE
6075     IF NONE
6076        PRINT 1 LINE WITH .AD.UNIT THUS
6077        ####################AIR DEFENSE UNIT ****** HAS NO AIR DEFENSE EQUIPMENT
6078        TRACE
6079  <------STOP
6080     OTHERWISE
6081     LET .MAX.WPN.RANGE = TW.MAX.RANGE(WPN.ID(F.UE.WEAPON.SET(.EQ)))
6082     IF .RANGE GT .MAX.WPN.RANGE
6083        LET .RANGE = SQRT.F((REAL.F(.MAX.WPN.RANGE**2 -
6084        SI.ALTITUDE(.INTERSECT)**2)))
6085        ''FIND THE POINT WHERE THE AIRCRAFT COMES INTO RANGE.
6086        IF .DELTAX = 0
6087           LET .X1 = SI.X.ENTRY(.INTERSECT)
6088           IF ABS.F(.X1 - UN.X.COORD(.AD.UNIT)) GT .RANGE
6089              ''THE AIRCRAFT NEVER COMES INTO RANGE.
6090              IF TACAIR.DEBUG = 1
6091                 PRINT 1 LINE AS FOLLOWS
6092                 ############AIRCRAFT NEVER COME INTO RANGE######################
6093              ALWAYS
6094  <------------RETURN
6095           OTHERWISE
6096
6097           LET .X2 = .X1
6098           LET .A = SQRT.F((REAL.F(.RANGE**2 - (.X -
6099           UN.X.COORD(.AD.UNIT))**2)))
6100           LET .Y1 = UN.Y.COORD(.AD.UNIT) + .A
6101           LET .Y2 = UN.Y.COORD(.AD.UNIT) - .A
6102        ELSE
6103           LET .A = .SLOPE**2 + 1
6104           LET .B = 2 * (.SLOPE *.INTERCEPT - UN.X.COORD(.AD.UNIT)
6105           - .SLOPE * UN.Y.COORD(.AD.UNIT))
6106           LET .C = UN.X.COORD(.AD.UNIT)**2 + .INTERCEPT**2
6107           + UN.Y.COORD(.AD.UNIT)**2 - 2*.INTERCEPT*UN.Y.COORD(.AD.UNIT)
6108           - .RANGE**2
6109           LET .C = .B**2 - 4.*.A*.C
6110           PRINT 1 LINE WITH .C AS FOLLOWS
6111           ######################.C = ***************  ################
6112           IF .C LT 0
6113              ''THE AIRCRAFT WILL NEVER COME IN RANGE.
6114              IF TACAIR.DEBUG = 1
6115                 PRINT 1 LINE AS FOLLOWS
6116                 ############AIRCRAFT NEVER COME INTO RANGE######################
6117              ALWAYS
6118  <------------RETURN
6119           OTHERWISE
6120           PRINT 1 LINE WITH .C AS FOLLOWS
6121           ######################.C = ***************  ################
6122
6123           LET .C = SQRT.F(.C)
6124           LET .X1 = (-.B + .C) / (2. * .A)
6125           LET .X2 = (-.B - .C) / (2. * .A)
6126           LET .Y1 = .SLOPE * .X + .INTERCEPT
6127           LET .Y2 = .SLOPE * .X + .INTERCEPT
6128        ALWAYS
```

```
6129        ''TAKE THE POINT CLOSEST TO THE ENTRY WHICH IS ALSO BETWEEN
6130        ''THE ENTRY AND EXIT.
6131        IF .LOWX GT .X1 OR .HIGHX LT .X1 OR
6132           .LOWY GT .Y1 OR .HIGHY LT .Y1
6133           IF .LOWX LE .X2 LE .HIGHX AND
6134              .LOWY LE .Y2 LE .HIGHY
6135              LET .X = .X2
6136              LET .Y = .Y2
6137           ELSE
6138              ''BOTH POINTS BAD.
6139              IF TACAIR.DEBUG = 1
6140                 PRINT 1 LINE AS FOLLOWS
6141                 ##########AIRCRAFT NEVER COME INTO RANGE#####################
6142              ALWAYS
6143 <------------RETURN
6144           ALWAYS
6145        ELSE
6146           IF .LOWX LE .X2 LE .HIGHX AND
6147              .LOWY LE .Y2 LE .HIGHY AND
6148              ABS.F(.X2 - SI.X.ENTRY(.INTERSECT)) LE
6149              ABS.F(.X1 - SI.X.ENTRY(.INTERSECT)) AND
6150              ABS.F(.Y2 - SI.Y.ENTRY(.INTERSECT)) LE
6151              ABS.F(.Y1 - SI.Y.ENTRY(.INTERSECT))
6152              LET .X = .X2
6153              LET .Y = .Y2
6154           ELSE
6155              LET .X = .X1
6156              LET .Y = .Y1
6157           ALWAYS
6158        ALWAYS
6159        LET .TOTAL.DISTANCE = SQRT.F((REAL.F((.X1-.X2)**2+(.Y1-.Y2)**2)))
6160        LET .DIST = SQRT.F((REAL.F((.X - SI.X.ENTRY(.INTERSECT))**2
6161        + (.Y - SI.Y.ENTRY(.INTERSECT))**2)))
6162        LET .TIME.IN.RANGE = .TOTAL.DISTANCE/.VELOCITY
6163        LET .SHOOT.TIME = .DIST / .VELOCITY
6164        IF .TIME.IN.RANGE LE .FIRING.TIME
6165 <----------RETURN
6166           ''.FIRING.TIME OCCURS AFTER AIRCRAFT IS OUT OF FIRING RANGE.
6167        OTHERWISE      ''WAIT UNTIL AIRCRAFT IS IN RANGE BEFORE FIRING
6168        IF .FIRING.TIME LT .SHOOT.TIME
6169           LET .FIRING.TIME = .SHOOT.TIME
6170        ALWAYS
6171        LET .RANGE = .MAX.WPN.RANGE
6172     ALWAYS
6173
6174     ''SHOOT UNTIL THE AIRCRAFT ARE ALL KILLED, OUT OF RANGE, OR
6175     ''OFF THE SEGMENT.
6176     IF TACAIR.DEBUG = 1
6177        LIST .SHOOT.TIME, .RANGE, .PD, .NUM.DET, .TRAVEL.TIME,
6178        .MAX.WPN.RANGE
6179     ALWAYS
6180     LOOP UNTIL .NUM.DET LE 0 OR
6181     .RANGE GT .MAX.WPN.RANGE OR
6182     .FIRING.TIME GT .TRAVEL.TIME
6183     DO
6184        IF TACAIR.DEBUG = 1
6185           LIST .NUM.DET, .RANGE, .FIRING.TIME, .RESULT
6186        ALWAYS
```

```
6187        ''.FIRING.TIME IS THE TOTAL TIME ELAPSED BETWEEN ENTRY
6188        ''INTO THE SENSOR FAN AND THE SHOT.
6189        CALL AD.SHOOT------------------------------------------------------------------------------------------------------>(307)
6190        GIVEN
6191        .RANGE,
6192        .FIRING.TIME,
6193        .AD.UNIT,
6194        .CALLING.PROCESS,
6195        .TYPE.PROCESS
6196        YIELDING
6197        .RESULT,
6198        .FIRING.TIME
6199        IF .RESULT = NO.PK.PTR
6200 <----------RETURN
6201        OTHERWISE
6202
6203        IF .RESULT = HIT OR .RESULT = MISS
6204           LET .SHOT.MADE = YES
6205        ALWAYS
6206        IF .RESULT = HIT
6207           SUBTRACT 1 FROM .NUM.DET
6208        ALWAYS
6209        LET .RANGE = SQRT.F((REAL.F(
6210        (SI.X.ENTRY(.INTERSECT) + .XVEL*.FIRING.TIME
6211        - UN.X.COORD(.AD.UNIT))**2
6212        + (SI.Y.ENTRY(.INTERSECT) + .YVEL*.FIRING.TIME
6213        - UN.Y.COORD(.AD.UNIT))**2
6214        + SI.ALTITUDE(.INTERSECT)**2)))
6215     ENDLOOP
6216
6217     IF .SHOT.MADE = YES
6218        LET UN.AD.AVAIL(.AD.UNIT) = TIME.V + .FIRING.TIME
6219        CALL FILE.KAD.SENSOR-----------------------------------------------------------------------------------------------------------
```

```
6227 EVENT ARTY.OCCUPATION                ''                                                                    E003
6228    GIVEN .BTRY
6229
6230    ADD 1 TO ANAL.CTR(135,1)  ''                                                                          \DYN_ANAL
6231    NORMALLY MODE IS INTEGER
6232    '' IF DEBUG = TRUE
6233       '' PRINT 1 LINE WITH .BTRY THUS
6234       '' = = = ARTY.OCCUPATION   .BTRY = ******* = = =
6235    '' ENDIF
6236    DEFINE .READY TO MEAN 0
6237    IF ANALYSIS(1) = TRUE
6238       USE UNIT 42 FOR OUTPUT
6239       PRINT 1 LINE WITH .BTRY, BY.STATUS(.BTRY), TIME.V THUS
6240       BTRY *** CHANGES FROM STATUS * TO STATUS 0 AT **.*****
6241       USE UNIT 6 FOR OUTPUT
6242    ALWAYS
6243    LET BY.STATUS(.BTRY) = .READY
6244    LET UNIT = BY.UNIT(.BTRY)     ''  %14FEB80_%RGR
6245    LET UN.TIME.LAST.MOVE(UNIT) = TIME.V     ''%14FEB80_%RGR
6246    '' THIS RECORDS THE TIME OF FINISHING THE MOVE
6247    '' SO THAT IT CANNOT BE MOVED UNTIL THE TIME.BETWEEN.ARTY.MOVE
6248    '' HAS ELAPSED OR IT (THE .BTRY) IS HIT BY OTHER ARTY
6249    IF TB.SHOOT.SCOOT.IND(BY.TYPE(.BTRY)) NE 1
6250       CALL FA.BN.MOVEMENT GIVEN BY.BN(.BTRY), 0------------------------------------------------------------------------------------------>( 89)
6251    ALWAYS
6252 ENDEVENT
```

```
6253 EVENT BTL.ENDED GIVEN WINNER.,BL.UNITS.,RD.UNITS.    ''                                                    E004
6254
6255    ADD 1 TO ANAL.CTR(136,1)  ''                                                                          \DYN_ANAL
6256    ''  THIS EVENT IS EXECUTED WHEN A BATTLE HAS TERMINATED DUE TO
6257    ''  A BREAKPOINT HAVING BEEN EXCEEDED.  IT CALLS CHECK.DEAD
6258    ''  TO SEE IF UNITS ARE DECIMATED, AND CALLS WHAT.NEXT TO DETER-
6259    ''  MINE ORDERS FOR THOSE UNITS THAT ARE NOT DECIMATED.
6260
6261    NORMALLY MODE IS INTEGER
6262    DEFINE BL.UNITS.,RD.UNITS.,NEW.BLUE,NEW.RED
6263    AS 1-DIMENSIONAL ARRAYS
6264    LET BTL = UN.BATTLE.INDEX(BL.UNITS.(1))
6265    LET SEQ = BTL.SEQ.NO(BTL)
6266    LET TOT.TIME = (TIME.V - (BTL.TIME.OF.DAY(BTL)/1000.)) * MINUTES.V
6267    CALL SWITCH.FO GIVEN UN.BATTLE.INDEX(BL.UNITS(1)),2------------------------------------------------>(238)
6268    ''EMPTY HC TEAMS, IF ANY, FROM BATTLE''                          ''14MAY80%CACI
6269    IF HC.SWITCH = 1                                                 ''14MAY80%CACI
6270       LET BTL = UN.BATTLE.INDEX(BL.UNITS(1))                           ''14MAY80%CACI
6271       IF BTL.BL.FARRP(BTL) > 0  ''BLUE TEAMS STILL ACTIVE''            ''14MAY80%CACI
6272          LET FP = BTL.BL.FARRP(BTL)                                       ''14MAY80%CACI
6273          LET SIDE = BLUE                                                  ''14MAY80%CACI
6274          CALL HC.EMPTY                                                    ''14MAY80%CACI------------------>(337)
6275          GIVING                                                           ''14MAY80%CACI
6276          FP,                                                              ''14MAY80%CACI
6277          BLUE                                                             ''14MAY80%CACI
6278          IF HC.DEBUG = "YES"                                              ''01JUL80%CACI
6279             PRINT 1 LINE WITH FP, SIDE, SEQ AS FOLLOWS                       ''01JUL80%CACI
6280             ----BTL.ENDED----  FARRP=****** SIDE=****** BATTLE=******
6281          ALWAYS                                                           ''01JUL80%CACI
6282       ALWAYS                                                           ''14MAY80%CACI
6283       IF BTL.RD.FARRP(BTL) > 0   ''RED TEAMS STILL ACTIVE''            ''14MAY80%CACI
6284          LET FP = BTL.RD.FARRP(BTL)                                       ''14MAY80%CACI
6285          LET SIDE = RED                                                   ''14MAY80%CACI
6286          CALL HC.EMPTY                                                    ''14MAY80%CACI------------------>(337)
6287          GIVING                                                           ''14MAY80%CACI
6288          FP,                                                              ''14MAY80%CACI
6289          SIDE                                                             ''14MAY80%CACI
6290          IF HC.DEBUG = "YES"                                              ''01JUL80%CACI
6291             PRINT 1 LINE WITH FP, SIDE, SEQ AS FOLLOWS                       ''01JUL80%CACI
6292             ----BTL.ENDED----  FARRP=****** SIDE=****** BATTLE=******
6293          ALWAYS                                                           ''01JUL80%CACI
6294       ALWAYS                                                           ''14MAY80%CACI
6295    ALWAYS                                                           ''14MAY80%CACI
6296    CALL EMPTY GIVEN UN.BATTLE.INDEX(BL.UNITS(1)),-------------------------------------------------->(328)
6297    WINNER., YES ''                                                                          CHG\05  \ARG_NO
6298    IF WINNER.=BLUE
6299       LET BL.WIN=YES
6300       LET RD.WIN=NO
6301       SKIP 1 LINE
6302       PRINT 2 LINES WITH SEQ, TIME.V,UNIT.NOS(BL.UNITS.(1)),
6303       UNIT.NOS(RD.UNITS.(1)) THUS
6304       BATTLE ****** ENDED AT **.**** HRS   WITH RED SIDE DISENGAGING
6305       BATTLE INVOLVED BLUE UNIT ****** AND RED UNIT ******
6306    ELSEIF WINNER.=RED
6307       LET BL.WIN=NO
6308       LET RD.WIN=YES
6309       SKIP 1 LINE
6310       PRINT 2 LINES WITH SEQ, TIME.V,UNIT.NOS(BL.UNITS.(1)),
```

```
6311        UNIT.NOS(RD.UNITS.(1)) THUS
6312        BATTLE •••••• ENDED AT ••.•••• HRS   WITH BLUE SIDE DISENGAGING
6313        BATTLE INVOLVED BLUE UNIT •••••• AND RED UNIT ••••••
6314     ELSE
6315        LET BL.WIN=NO
6316        LET RD.WIN=NO
6317        SKIP 1 LINE
6318        PRINT 2 LINES WITH SEQ, TIME.V,UNIT.NOS(BL.UNITS.(1)),
6319        UNIT.NOS(RD.UNITS.(1)) THUS
6320        BATTLE •••••• ENDED AT ••.•••• HRS   WITH BOTH SIDES DISENGAGING
6321        BATTLE INVOLVED BLUE UNIT ••••••• AND RED UNIT •••••••
6322     ENDIF ALWAYS
6323     PRINT 1 LINE WITH TOT.TIME THUS
6324     BATTLE LASTED •••••• MINUTES
6325     SKIP 1 LINE
6326     CALL CHECK.DEAD(BL.UNITS(•))------------------------------------------------------------------------------>(130)
6327     YIELDING BL.STATUS,NEW.BLUE(•)
6328     CALL CHECK.DEAD(RD.UNITS(•))------------------------------------------------------------------------------>(130)
6329     YIELDING RD.STATUS,NEW.RED(•)
6330     '' WHEN THE STATUS VARIABLE = YES, THERE ARE UNITS REMAINING                                              \1
6331     '' ON THE APPROPRIATE SIDE
6332     IF BL.STATUS = YES
6333        CALL WHAT.NEXT(BL.WIN,NEW.BLUE(•))--------------------------------------------------------------------->(125)
6334     ENDIF
6335     IF RD.STATUS = YES
6336        CALL WHAT.NEXT(RD.WIN,NEW.RED(•))---------------------------------------------------------------------->(125)
6337     ENDIF
6338     RELEASE NEW.BLUE(•)
6339     RELEASE NEW.RED(•)
6340  ENDEVENT
```

```
6341 EVENT CFR.ACTIVATION                                                    ''                                                    E005
6342    GIVEN BTRY
6343
6344
6345    ADD 1 TO ANAL.CTR(137,1)  ''                                                                                          \DYN_ANAL
6346    NORMALLY MODE IS INTEGER
6347    DEFINE RADAR.TYPE AS A TEXT VARIABLE ''                                                                               \TEXT
6348    DEFINE BTRY, RANGE AS INTEGER VARIABLES
6349
6350    LET TYPE = BY.TYPE(BTRY)
6351    IF TB.MAX.RANGE(TYPE) < 700    '' IN HEXADECCAMETERS - A MORTAR  %RGR
6352       LET RADAR.TYPE = "CM"
6353    ELSE
6354       LET RADAR.TYPE = "CB"
6355    ENDIF
6356    IF UN.COLOR(BY.UNIT(BTRY)) = BLUE
6357       LET  ENEMY = RED
6358    ELSE
6359       LET ENEMY = BLUE
6360    ENDIF
6361
6362    LOOP FOR EACH CFR IN SIDE.CFR.SET(ENEMY)
6363    DO THIS
6364       LET LINK = CFR.US.LINK(CFR)
6365       LET ST = US.SENSOR.TYPE(LINK)
6366       LET RADAR = US.UNIT(LINK)
6367       ''UTILIZE ACT.RANGE FUNCTION IN NEXT ASSIGNMENT ------------------------------------------------------------------ \1>(628)
6368       LET RANGE = INT.F(ACT.RANGE(BY.UNIT(BTRY),RADAR))
6369       LET MODEL = US.MODEL(LINK)
6370       IF ST.NAME(ST) = RADAR.TYPE
6371          IF US.STATUS(LINK) = ACTIVE
6372             IF RANGE<=CFR.RH.RANGE(L.MCFR.RH.LIST(MODEL))
6373                CALL CFR.DETECTION GIVEN CFR, BTRY, RANGE-------------------------------------------------------------->(166)
6374             ENDIF
6375          ENDIF
6376       ENDIF
6377    ENDLOOP
6378
6379 ENDEVENT
```

```
6380 EVENT CFR.OFF                                                                    E006
6381    GIVEN
6382    SENS.ID,
6383    SENS.LINK
6384
6385    ADD 1 TO ANAL.CTR(138,1)                                                      \DYN_ANAL
6386    NORMALLY MODE IS INTEGER
6387    DEFINE ON.TIME AS A REAL VARIABLE
6388
6389    IF US.STATUS(SENS.LINK) = HOLD
6390 <------EXITEVENT
6391    OTHERWISE
6392
6393    LET MODEL = US.MODEL(SENS.LINK)
6394    LET ON.TIME = (MCFR.MIN.OFF(MODEL)/10.)*(1.+RANDOM.F(RN.SEED))
6395    SCHEDULE_A CFR.ON-------------------------------------------------------------------------------------------->(360)
6396    GIVEN
6397    SENS.ID,
6398    SENS.LINK
6399    IN ON.TIME MINUTES
6400    LET US.STATUS(SENS.LINK) = HOLD
6401    LET CFR.LAST.ON.OR.OFF(SENS.ID) = TIME.V
6402
6403 <--EXITEVENT
6404 ENDEVENT
```

```
6405 EVENT CFR ON                                                                                                    ''  E007
6406    GIVEN
6407    SENS.ID,
6408    SENS.LINK
6409
6410    ADD 1 TO ANAL.CTR(139,1)  ''                                                                                \DYN_ANAL
6411    NORMALLY MODE IS INTEGER
6412    DEFINE SIDE, SECTOR AS INTEGER VARIABLES
6413    DEFINE SET.BACK, THETA, PHI, DELTA, R AS REAL VARIABLES
6414    DEFINE DIRECTION AS A REAL VARIABLE
6415    DEFINE .INCREMENT AS AN INTEGER VARIABLE  ''                                                                 \DEBUG
6416
6417    IF US.STATUS(SENS.LINK) = ACTIVE
6418 <------EXITEVENT
6419    OTHERWISE
6420
6421    LET MODEL = US.MODEL(SENS.LINK)
6422    SCHEDULE_A CFR.OFF-------------------------------------------------------------------------------------->(359)
6423    GIVEN
6424    SENS.ID,
6425    SENS.LINK
6426    IN MCFR.MAX.ON(MODEL) / 10. MINUTES
6427    LET RADAR = US.UNIT(SENS.LINK)
6428    LET SIDE = UN.COLOR(RADAR)
6429    CALL LOCATE.SECTOR-------------------------------------------------------------------------------------->(100)
6430    GIVEN
6431    UN.Y.COORD(RADAR)
6432    YIELDING
6433    SECTOR
6434    ''THE FOLLOWING 10 LINES OF CODE CORRECT THE ZERO FEBA PROBLEM                                        CHG\32 \DEBUG
6435    LET .INCREMENT = -1
6436    UNTIL SS.SET(SIDE,SECTOR) IS NOT EMPTY,
6437    DO
6438       ADD .INCREMENT TO SECTOR
6439       IF SECTOR IS ZERO,
6440          LET .INCREMENT = 1
6441          LET SECTOR = 2 ''ALREADY CHECKED 1
6442       ALWAYS
6443    ENDLOOP ''REPEAT UNTIL SET IS NOT EMPTY
6444
6445    IF SIDE = BLUE
6446       LET DIRECTION = 0.0
6447       LET FEBA = F.SS.SET(SIDE,SECTOR)
6448    ELSE
6449       LET DIRECTION = PI.C
6450       LET FEBA = L.SS.SET(SIDE,SECTOR)
6451    ALWAYS
6452    ''UTILIZE ACT.RANGE FUNCTION IN NEXT ASSIGNMENT ---------------------------------------------------- \1>(628)
6453    LET SET.BACK = ACT.RANGE(RADAR, FEBA)
6454    IF SET.BACK = 0.0
6455       ''IN CASE RADAR UNIT IS ONLY ONE IN SECTOR
6456       LET SET.BACK = 5000./16.
6457    ALWAYS
6458    LET THETA = ARCTAN.F(REAL.F(MCFR.SEARCH.WIDTH(MODEL))/2., SET.BACK)
6459    IF THETA < 0.0
6460       ADD 2.0 * PI.C TO THETA
6461    ALWAYS
6462    LET PHI = MCFR.SWEEP.ANGLE(MODEL)*2.*PI.C/360.
```

```
6463     LET DELTA = 2.*THETA - PHI
6464     IF DELTA < 0.0
6465        LET DELTA = 0.0
6466     ALWAYS
6467     LET R = BETA.F(2.,2.,1)
6468     LET CFR.ORIENTATION(SENS.ID) = ((-DELTA/2.+R*DELTA)+DIRECTION)*100.
6469     IF CFR.ORIENTATION(SENS.ID) < 0
6470        ADD (2.0*PI.C*100.) TO CFR.ORIENTATION(SENS.ID)
6471     ALWAYS
6472     LET US.STATUS(SENS.LINK) = ACTIVE
6473     LET CFR.LAST.ON.OR.OFF(SENS.ID) = TIME.V
6474
6475 <--EXITEVENT
6476 ENDEVENT
```

```
6477 EVENT CFR.OPERATOR                                                                                    ''                                                                                
```

```
6535                BY.PGM.CAP(.BTRY) = SADARM
6536                FIND THE FIRST CASE
6537                IF FOUND
6538                   LET TR.FDC(TR) = .FDC
6539                   LET .FDC = 0
6540                   LET TR.PGM.STATUS(TR) = SADARM
6541             <-----LEAVE
6542                OTHERWISE
6543             ENDLOOP ''                                                                                    \1
6544             IF TR.PGM.STATUS(TR) NE SADARM
6545                LET .FDC = FD.FDC(.FDC)
6546             ALWAYS
6547          ENDLOOP ''                                                                                       \1
6548          IF TR.PGM.STATUS(TR) = SADARM
6549             LET TR.MIL.WORTH(TR) = 1999
6550          ALWAYS
6551          PRINT 1 LINE WITH TR, TR.FDC(TR), TR.PGM.STATUS(TR) THUS
6552          = = = CFR.OPERATOR TGT ****** GOES TO FDC ****** WITH PGM STATUS ****
6553       ALWAYS
6554
6555       LET TR.DET.QUANT(DET.LINK) = N.BY.HOW.SET(BTRY)
6556       LET TR.DET.ELEM.PROB(DET.LINK) = CF.D.PD(THIS.UNIT)
6557       FILE DET.LINK IN TR.DET.LIST(TR)
6558       LET TR.RECVD.TIME(TR) = TIME.V
6559       LET TR.ABORT.TIME(TR) = TR.RECVD.TIME(TR) + .5
6560       DESTROY THE CF.DET.UNIT CALLED THIS.UNIT
6561       CALL CHK.COMP.TR------------------------------------------------------------------------------------------>(168)
6562       GIVEN
6563       TR,
6564       US.FDC(LINK),
6565       CFR
6566       YIELDING
6567       DUPLICATE
6568       IF DUPLICATE = FALSE
6569          CALL CHK.FD.TR------------------------------------------------------------------------------------->(169)
6570          GIVEN
6571          TR,
6572          US.FDC(LINK),
6573          CFR
6574          YIELDING
6575          DUPLICATE
6576       ALWAYS
6577       IF DUPLICATE = TRUE
6578          REMOVE DET.LINK FROM TR.DET.LIST(TR)
6579          DESTROY THE TR.DET.LINK CALLED DET.LINK
6580          DESTROY THE TARGET.REPORT CALLED TR
6581       ELSE
6582          ACTIVATE_THE TARGET.REPORT CALLED TR NOW------------------------------------------------------------>(470)
6583       ALWAYS
6584       IF CF.OP.Q(CFR) IS NOT EMPTY
6585          LET NEXT.UNIT = F.CF.OP.Q(CFR)
6586          SCHEDULE_A CFR.OPERATOR----------------------------------------------------------------------------->(362)
6587          GIVEN
6588          CFR,
6589          NEXT.UNIT NOW ''                                                                                  \1
6590       ALWAYS
6591    ALWAYS
6592
```

```
6593 <--EXITEVENT
6594 ENDEVENT
```

```
6595 EVENT CHANGE.LITE                                                                                    E009
6596
6597    ADD 1 TO ANAL.CTR(141,1)  ''                                                                      \DYN_ANAL
6598    IF NITE.OR.DAY = NITE
6599       LET NITE.OR.DAY = DAY
6600       IF EENT < TIME.V
6601          ADD 24.0 TO EENT
6602       ALWAYS
6603       SCHEDULE_A CHANGE.LITE AT EENT------------------------------------------------------------------------------------------>(365)
6604    ELSE
6605       LET NITE.OR.DAY = NITE
6606       IF BMNT < TIME.V
6607          ADD 24.0 TO BMNT
6608       ALWAYS
6609       SCHEDULE_A CHANGE.LITE AT BMNT------------  ------------------------------------------------------------------------------>(365)
6610    ALWAYS
6611
6612    CALL DQ.CMSN.QUEUE------------------------------------------------------------------  ------------------>(327)
6613
6614 <--EXITEVENT
6615 ENDEVENT
```

```
6616 EVENT CHANGE.WEATHER                                                                                    E010
6617    GIVEN
6618    .VISIBILITY
6619
6620    ADD 1 TO ANAL.CTR(142,1)                                                                        \DYN_ANAL
6621    NORMALLY MODE IS INTEGER
6622
6623    LET VISIBILITY = .VISIBILITY
6624
6625    CALL DQ.CMSN.QUEUE------------------------------------------------------------------------------>(327)
6626
6627    PRINT 2 LINES WITH TIME.V, VISIBILITY,
6628    NITE.OR.DAY, NITE.VIS.PCT THUS
6629    ===CHANGE.WEATHE AT **.****HRS TO VIS= **** HDM,
6630    NITE.OR.DAY= ***, NITE.VIS.PCT= ****          = = =
6631
6632 <--EXITEVENT
6633 ENDEVENT
```

```
6634 EVENT DQ.OLD.SORTIE.QUEUE                    ''                                                    E011
6635    GIVEN
6636    .SIDE
6637
6638    ADD 1 TO ANAL.CTR(143,1)  ''                                                                    \DYN_ANAL
6639    NORMALLY MODE IS INTEGER
6640
6641    ''SD.TP.SORTIE MINUTES HAVE PASSED SINCE THE
6642    ''FIRST OLD.SORTIE IN THE QUEUE WAS ACTIVATED.
6643    ''REMOVE IT AND REDUCE THE SIDES NUMBER OF SORTIES
6644    ''IN THIS TIME PERIOD.  CALL DQ.CMSN.QUEUE TO SEE IF
6645    ''OTHER MISSIONS CAN NOW FLY.
6646    REMOVE THE FIRST OLD.SORTIE
6647    FROM THE SD.OLD.SORTIE.QUEUE(.SIDE)
6648    SUBTRACT OS.QTY(OLD.SORTIE) FROM SD.SORTIES.THIS.TP(.SIDE)
6649    DESTROY THIS OLD.SORTIE
6650    IF TACAIR.DEBUG = 1
6651       PRINT 1 LINE WITH TIME.V THUS
6652       == = = = =   DQ.OLD.SORTIE.QUEUE AT ***.***
6653       LIST .SIDE, N.SD.OLD.SORTIE.QUEUE(.SIDE),
6654       SD.SORTIES.THIS.TP(.SIDE)
6655    ALWAYS
6656
6657    CALL DQ.CMSN.QUEUE------------------------------------------------------------------------------------------------------------------------>(327)
6658
6659 <--RETURN
6660 END
```

```
6661 EVENT END.SIMULATION                                                                                    E012
6662
6663    ADD 1 TO ANAL.CTR(144,1)  ''                                                                        \DYN_ANAL
6664    NORMALLY MODE IS INTEGER
6665
6666    ''PRINT 2 LINES WITH ZTIME.F, USED.DBANK.V, STOP.SIMULATION.TIME  THUS                               \VAX
6667    ''==== END RUN ====  ELAPSED CPU TIME (SEC) = ****.*, EXTRA CORE = *******
6668    ''==== SCHEDULED SIMULATION STOP AT **.*** = = =
6669    PRINT 1 LINE WITH TIME.V THUS
6670    ==== SCHEDULED SIMULATION STOP AT **.*** = = =
6671
6672    CALL KV.SCOREBOARD----------------------------------------------------------------------------->(609)
6673    CALL POSITION.OUT-----------------------------------------------------------------------------><613)
6674    CALL OUTPUT.EXPENDITURES----------------------------------------------------------------------->(624)
6675
6676 <--STOP
6677 ENDEVENT
```

```
6678 EVENT ENGAGEMENT                                                                                    E013
6679    GIVEN
6680    .DETECTING.UNIT,
6681    .DETECTED.UNIT
6682
6683    ADD 1 TO ANAL.CTR(145,1)  ''                                                                     \DYN_ANAL
6684    NORMALLY MODE IS INTEGER
6685
6686    IF DEBUG = TRUE
6687       PRINT 4 LINES WITH
6688       TIME.V,
6689       UN.BATTLE.INDEX(.DETECTED.UNIT),
6690       UNIT.NOS(.DETECTING.UNIT),
6691       UNIT.NOS(.DETECTED.UNIT)
6692       THUS
6693       ==========ENGAGEMENT==========
6694       TIME.V = **.********  BATTLE = ******
6695       DETECTING(SHOOTING) UNIT = ******
6696       DETECTED(TARGET) UNIT = ******
6697    ALWAYS
6698
6699    LOOP FOR EACH ENGAGEMENT OF EV.S(I.ENGAGEMENT,
6700    WITH DETECTING.UNIT(ENGAGEMENT) = .DETECTING.UNIT
6701    DO THE FOLLOWING
6702       CANCEL_THE ENGAGEMENT------------------------------------------------------------------------------------------------------------>(369)
6703       DESTROY THE ENGAGEMENT
6704    ENDLOOP
6705
6706    FOR EVERY VISIBLE.UNIT OF UN.LOS.LIST(.DETECTING.UNIT)
6707    WITH VU.POINTER(VISIBLE.UNIT) = .DETECTED.UNIT
6708    FIND THE FIRST CASE
6709    IF NONE
6710       CALL ERROR.STOP------------------------------------------------------------------------------------------------------------------>(604)
6711    ALWAYS
6712
6713    LET VU.STATUS(VISIBLE.UNIT) = YES
6714    LOOP FOR EVERY .FIRER.EQUIP OF UN.EQUIP.LIST(.DETECTING.UNIT)
6715    WITH UE.QUANT(.FIRER.EQUIP) GT 0 AND
6716    N.UE.WEAPON.SET(.FIRER.EQUIP) GT 0
6717    DO THE FOLLOWING
6718       LOOP FOR EVERY .TARGET.EQUIP OF UN.EQUIP.LIST(.DETECTED.UNIT)
6719       WITH UE.QUANT(.TARGET.EQUIP) GT 0
6720       DO THE FOLLOWING
6721          CREATE A FIRING.TABLE
6722          LET FT.TGT.UNIT(FIRING.TABLE) = .DETECTED.UNIT
6723          LET FT.TARGET.EQUIP(FIRING.TABLE) = .TARGET.EQUIP
6724          FILE THIS FIRING.TABLE IN THE UE.TARGET.LIST(.FIRER.EQUIP)
6725          ''
6726          ''ADD CODE TO CHECK ON THE EQUIPMENT THAT
6727          ''IS WAITING FOR TARGETS TO COME INTO RANGE
6728       ENDLOOP
6729
6730       IF N.SO.LIST(.FIRER.EQUIP) LE 0  '' NOT ENGAGED ELSEWHERE
6731          LOOP FOR I = 1 TO UE.QUANT(.FIRER.EQUIP)
6732          DO THE FOLLOWING
6733             ACTIVATE_A SHOOT.OUT CALLED .KILL.EM NOW------------------------------------------------------------------------------>(493)
6734             LET FIRING.EQUIP(.KILL.EM) = .FIRER.EQUIP
6735             LET FIRER.UNIT(.KILL.EM) = .DETECTING.UNIT
```

```
6736            FILE THIS .KILL EM IN THE SO.LIST(.FIRER.EQUIP)
6737            ADD 1 TO .COUNT
6738         ENDLOOP
6739      ALWAYS
6740
6741      IF DEBUG = TRUE
6742         PRINT 3 LINES WITH .COUNT, UE.QUANT(.FIRER.EQUIP),
6743         UE.ID(.FIRER.EQUIP),
6744         N.UE.TARGET.LIST(.FIRER.EQUIP),
6745         UN.BATTLE.INDEX(.DETECTED.UNIT) THUS
6746         THERE ARE *** OF ***(EQUIP.TYPE ***)
6747         AVAILIBLE TO ENGAGE *** TYPES OF ENEMY
6748         EQUIPMENT IN BATTLE ******
6749      ALWAYS
6750
6751      LET .COUNT=0
6752   ENDLOOP
6753
6754   IF UN.MISSION(.DETECTING.UNIT) LE 3
6755      CALL REQUEST.DEF.FASCAM----------------------------------------------------------------------------------->(215)
6756      GIVEN
6757      .DETECTING.UNIT,
6758      .DETECTED.UNIT
6759   ALWAYS
6760   CALL REQUEST.ILLUM---------------------------------------------------------------------------------------->(218)
6761   GIVEN
6762   .DETECTING.UNIT,
6763   ILLUM.RULE(UN.COLOR(.DETECTING.UNIT),
6764   UN.MISSION(.DETECTING.UNIT)),
6765   0
6766   CALL REQUEST.SMOKE---------------------------------------------------------------------------------------->(222)
6767   GIVEN
6768   .DETECTING.UNIT,
6769   SMK.USE.RULE(UN.COLOR(.DETECTING.UNIT), NITE.OR.DAY,
6770   UN.MISSION(.DETECTING.UNIT)),
6771   0
6772
6773 <--EXITEVENT
6774 ENDEVENT
```

```
6775 EVENT FEBA.SORTIE  ''                                                                                    E014
6776    GIVEN SECTOR,
6777    SIDE,
6778    TIME.INTERVAL
6779
6780
6781    ADD 1 TO ANAL.CTR(146,1)  ''                                                                          \DYN_ANAL
6782    NORMALLY MODE IS INTEGER
6783    DEFINE INTERVAL, AND TIME.INTERVAL AS REAL VARIABLES
6784    LET INTERVAL = 2.*TIME.INTERVAL
6785    '' THIS ROUTINE RESTRUCTURED %14MAR80_%RGR TO ACCOUNT FOR AO'S
6786    '' THAT ARE FLYING THE FIRST TIME AND THOSE BEING RESCHEDULED .
6787
6788    LET FLAG1 = 0 ''%22DEC80_%RWF
6789    LET FLAG2 = 0 ''%22DEC80_%RWF
6790    FOR EACH US.LINK IN AVAIL.AO.LIST(SIDE),
6791    DO THIS ''%22DEC80_%RWF                                                                               \1
6792       IF US.ID(US.LINK) NE 0 ''%22DEC80_%RWF
6793    <------CYCLE ''%22DEC80_%RWF
6794       OTHERWISE ''%22DEC80_%RWF
6795       LET FLAG1 = 1 ''%22DEC80_%RWF
6796       LET AO.Y.COORD = UN.Y.COORD(US.UNIT(US.LINK))  ''%22DEC80_%RWF
6797       CALL LOCATE.SECTOR GIVEN AO.Y.COORD YIELDING AO.SECTOR ''%22DEC80_%RWF----------------------------------------->(100)
6798       IF AO.SECTOR = SECTOR ''%22DEC80_%RWF
6799          LET FLAG2 = 1 ''%22DEC80_%RWF
6800    <------EXITLOOP ''%22DEC80_%RWF
6801       OTHERWISE
6802    ENDLOOP ''%22DEC80_%RWF
6803
6804    IF FLAG1 = 1 AND FLAG2 = 1 ''%22DEC80_%RWF
6805       ACTIVATE_A AIR.OBSERVER CALLED AO NOW  '' %19DEC80_%RWF----------------------------------------------->(428)
6806       '' THOUGH THE  SECTOR IS PASSED AS AN ARGUMENT TO THIS ROUTINE,
6807       ''THE SECTOR OF THE AIRCRAFT MUST MATCH THAT OF THE SORTIE  - %19DEC80_%RWF
6808       LET AO.US.LINK(AO) = US.LINK
6809       LET US.ID(US.LINK) = AO
6810       LET US.STATUS(US.LINK) = ACTIVE    '' %7MAR80_%RGR
6811       LET AO.CURRENT.TR(AO) = 0
6812       IF SIDE = BLUE
6813          LET FEBA = F.SS.SET(SIDE,SECTOR)
6814       ELSE ''SIDE IS RED
6815          LET FEBA = L.SS.SET(SIDE,SECTOR)
6816       ENDIF
6817       LET AO.X.START(AO) = UN X.COORD(FEBA)
6818       LET AO.Y.START(AO) = SE.BNDRY.INT(SECTOR)
6819       LET SECTOR.WIDTH = FEBA.WIDTH/N.SECTOR
6820       CREATE A FLIGHT.LEG CALLED LEG
6821       LET FL.X.START(LEG) = AO.X.START(AO)
6822       LET FL.Y.START(LEG) = AO.Y.START(AO)
6823       LET FL.X.END(LEG) = AO.X.START(AO)
6824       LET FL.Y.END(LEG) = AO.Y.START(AO) + SECTOR.WIDTH
6825       FILE LEG IN AO.FLIGHT.LEG.LIST(AO)
6826       SCHEDULE_A FEBA.SORTIE CALLED SORTIE GIVEN SECTOR, SIDE, AND TIME.INTERVAL------------------------------------>(371)
6827       IN UNIFORM.F(TIME.INTERVAL,INTERVAL,RN.SEED) MINUTES
6828 <------EXITEVENT
6829
6830    OTHERWISE ''                                                                  CHANGED ELSE TO OTHERWISE \1
6831    LET FLAG3 = 0 ''%22DEC80_%RWF
6832    LET FLAG4 = 0 ''%22DEC80_%RWF
```

```
6833    FOR EACH US.LINK IN AVAIL.AO.LIST(SIDE),
6834    DO THIS ''%22DEC80_%RWF                                                                                    \1
6835       IF US.STATUS(US.LINK) = BUSY OR US.ID(US.LINK) = 0 ''%22DEC80_%RWF
6836    <------CYCLE ''%22DEC80_%RWF
6837       OTHERWISE  ''A USED AIRCRAFT IS NOT NOW FLYING OR WAITING TO FLY - %22DEC80
6838       LET FLAG3 = 1 ''%22DEC80_%RWF
6839       LET AO.Y.COORD = UN.Y.COORD(US.UNIT(US.LINK)) ''MOVED HERE %19DEC80_%RWF
6840       CALL LOCATE.SECTOR GIVEN AO.Y.COORD YIELDING AO.SECTOR    ''%22DEC80_%RWF------------------------------------------>(100)
6841       IF AO.SECTOR = SECTOR''%22DEC80_%RWF
6842          LET FLAG4 = 1''%22DEC80_%RWF
6843    <------EXITLOOP''%22DEC80_%RWF
6844       OTHERWISE''%22DEC80_%RWF
6845    ENDLOOP''%22DEC80_%RWF
6846
6847    IF FLAG3 = 1 AND FLAG4 =1''%22DEC80_%RWF
6848       ACTIVATE_A AIR.OBSERVER CALLED AO NOW    '' %19DEC80_%RWF------------------------------------------------>(428)
6849       LET AO.US.LINK(AO) = US.LINK
6850       LET US.ID(US.LINK) = AO
6851       LET US.STATUS(US.LINK) = ACTIVE    '' %7MAR80_%RGR
6852       LET AO.CURRENT.TR(AO) = 0
6853       IF SIDE = BLUE
6854          LET FEBA = F.SS.SET(SIDE,SECTOR)
6855       ELSE ''SIDE IS RED
6856          LET FEBA = L.SS.SET(SIDE,SECTOR)
6857       ENDIF
6858       LET AO.X.START(AO) = UN.X.COORD(FEBA)
6859       LET AO.Y.START(AO) = SE.BNDRY.INT(SECTOR)
6860       LET SECTOR.WIDTH = FEBA.WIDTH/N.SECTOR
6861       CREATE A FLIGHT.LEG CALLED LEG
6862       LET FL.X.START(LEG) = AO.X.START(AO)
6863       LET FL.Y.START(LEG) = AO.Y.START(AO)
6864       LET FL.X.END(LEG) = AO.X.START(AO)
6865       LET FL.Y.END(LEG) = AO.Y.START(AO) + SECTOR.WIDTH
6866       FILE LEG IN AO.FLIGHT.LEG.LIST(AO)
6867       SCHEDULE_A FEBA.SORTIE CALLED SORTIE GIVEN SECTOR,------------------------------------------------------------>(371)
6868       SIDE, AND TIME.INTERVAL    ''%19DEC80_%RWF
6869       IN UNIFORM.F(TIME.INTERVAL,INTERVAL,RN.SEED) MINUTES
6870 <------EXITEVENT
6871    OTHERWISE ''                                                                  CHANGED ELSE TO OTHERWISE \1
6872    SCHEDULE_A FEBA.SORTIE CALLED SORTIE GIVEN SECTOR, SIDE, TIME.INTERVAL------------------------------------------>(371)
6873    IN UNIFORM.F(TIME.INTERVAL,INTERVAL,RN.SEED) MINUTES
6874 ENDEVENT
```

```
6875 EVENT GET.NX.ORD                                                                                                    E015
6876     GIVEN
6877     UNT.,
6878     NEXT.,
6879     EN.UNITS.,
6880     FLAG
6881
6882     ADD 1 TO ANAL.CTR(147,1)  ''                                                                                \DYN_ANAL
6883     ''THIS EVENT LOCATES THE UNITS NEXT ORDER IN HIS LIST OF ORDERS
6884     ''   AND SCHEDULES THE ACTIVATION OF THAT ORDER.
6885
6886     NORMALLY MODE IS INTEGER
6887     DEFINE EN.UNITS. AS AN INTEGER 1-DIMENSIONAL ARRAY
6888
6889     LET T.DELAY = 0
6890     LET MAN.UNT=UN.PTR(UNT.)
6891     FOR EVERY ORDER IN MU.ORDER.SET(MAN.UNT)
6892     WITH ORD.SEQ.NO(ORDER)=MU.CUR.ORDER(MAN.UNT)
6893     FIND THE FIRST CASE
6894     IF FOUND
6895        IF ORD.TYPE(ORDER)="ATK"
6896           LET LAST.ORD.ATK=1
6897        ALWAYS
6898     ELSE
6899        CALL ERROR.STOP------------------------------------------------------------------------------------------>(604)
6900     ALWAYS
6901
6902     PRINT 1 LINE WITH UNIT.NOS(UNT.),ORD.SEQ.NO(ORDER),
6903     ORD.TYPE(ORDER), TIME.V THUS
6904     UNIT ****** LEAVING ORDER #** (******) AT TIME = **.**
6905
6906     FOR EVERY ORDER IN MU.ORDER.SET(MAN.UNT)
6907     WITH ORD.SEQ.NO(ORDER)=NEXT.
6908     FIND THE FIRST CASE
6909     IF FOUND
6910        LET MU.CUR.ORDER(MAN UNT)=NEXT.
6911        IF ORD.TYPE(ORDER)="DEF"
6912           SCHEDULE_AN ACT.DEF-------------------------------------------------------------------------------->(414)
6913           GIVEN
6914           UNT.,
6915           ORDER NOW ''                                                                                          \1
6916           IF FLAG NE 0
6917              RELEASE EN.UNITS.(*)
6918           ALWAYS
6919        ELSEIF ORD.TYPE(ORDER)="ATK"
6920           IF FLAG = 0
6921              TRACE
6922 <-----------STOP
6923           OTHERWISE
6924
6925           IF LAST.ORD.ATK NE 1
6926              SCHEDULE_AN ACT.ATK---------------------------------------------------------------------------->(412)
6927              GIVEN
6928              UNT.,
6929              ORDER,
6930              EN.UNITS.(*) NOW ''                                                                                \1
6931           ELSE
6932              SCHEDULE_AN ACT.ATK---------------------------------------------------------------------------->(412)
```

```
6933              GIVEN
6934              UNT.,
6935              ORDER,
6936              EN.UNITS.(*)
6937              IN ATK.DELAY MINUTES
6938              LET T.DELAY = ATK.DELAY
6939           ALWAYS
6940        ELSEIF ORD.TYPE(ORDER)="REINF"
6941           SCHEDULE_AN ACT.REINF------------------------------------------------------------------------------------------>(347)
6942           GIVEN
6943           UNT.,
6944           ORDER,
6945           EN.UNITS.(*) NOW ''                                                                                            \1
6946        ELSEIF ORD.TYPE(ORDER)="MOVCOR"
6947           IF LAST.ORD.ATK NE 1
6948              SCHEDULE_AN ACT.MOVCOR--------------------------------------------------------------------------------->(415)
6949              GIVEN
6950              UNT.,
6951              ORDER NOW ''                                                                                                \1
6952           ELSE
6953              SCHEDULE_AN ACT.MOVCOR--------------------------------------------------------------------------------->(415)
6954              GIVEN
6955              UNT.,
6956              ORDER
6957              IN ATK.DELAY MINUTES
6958              LET T.DELAY = ATK.DELAY
6959           ALWAYS
6960           IF FLAG NE 0
6961              RELEASE EN.UNITS.(*)
6962           ALWAYS
6963        ELSEIF ORD.TYPE(ORDER)="MOVDIS"
6964           SCHEDULE_AN ACT.MOVDIS--------------------------------------------------------------------------------->(416)
6965           GIVEN
6966           UNT.,
6967           ORDER NOW ''                                                                                                   \1
6968           IF FLAG NE 0
6969              RELEASE EN.UNITS.(*)
6970           ALWAYS
6971        ELSE
6972           CALL ERROR.STOP---------------------------------------------------------------------------------------->(604)
6973        ALWAYS ALWAYS ALWAYS ALWAYS ALWAYS
6974     ELSE
6975        CALL ERROR.STOP---------------------------------------------------------------------------------------->(604)
6976     ALWAYS
6977
6978     IF T.DELAY = 0
6979        PRINT 1 LINE WITH NEXT.,ORD.TYPE(ORDER) THUS
6980        NEXT ORDER IS #** (******)
6981     ELSE
6982        PRINT 1 LINE WITH NEXT.,ORD.TYPE(ORDER),T.DELAY THUS
6983        NEXT ORDER IS #** (******) AFTER ****** MINUTE DELAY
6984     ALWAYS
6985
6986 <--EXITEVENT
6987 ENDEVENT
```

```
6988 EVENT HC.DEPART.BATTLE                                                                                    ''                                                                                
```

```
7030 EVENT HELO.ENGAGEMENT                                                                                    ''                                                                                
```

```
7088
7089        LOOP FOR EVERY TARGET.EQUIPMENT OF UN.EQUIP.LIST(.ENEMY.UNIT)
7090        WITH UE.QUANT(TARGET.EQUIPMENT) > 0
7091        DO THIS
7092           CREATE A FIRING.TABLE
7093           LET FT.TGT.UNIT(FIRING.TABLE) = .ENEMY.UNIT
7094           LET FT.TARGET.EQUIP(FIRING.TABLE) = TARGET.EQUIPMENT
7095           FILE THIS FIRING.TABLE IN UE.TARGET.LIST(.FIRING.HELO)
7096           IF HC.DEBUG = "YES"
7097              PRINT 1 LINE WITH .ENEMY.UNIT,
7098              EQ.NAME(UE.ID(TARGET.EQUIPMENT)),
7099              TARGET.EQUIPMENT AS FOLLOWS
7100              ——HELO$ENGAGEM .ENEMY.UNIT=******   TARGET.EQUIPMENT=****** (******)
7101           ALWAYS
7102        ENDLOOP
7103     ENDLOOP
7104
7105     FOR EACH HELICOPTER.FIRE IN EV.S(I.HELICOPTER.FIRE)
7106     WITH  HF.TEAM(HELICOPTER.FIRE) = ATK.TEAM
7107     FIND THE FIRST CASE
7108     IF NONE
7109        LOOP FOR EVERY .HC IN HT.MEMBER.LIST(ATK.TEAM)
7110        WITH HC.TYPE(.HC) NE SCOUT
7111        DO THE FOLLOWING
7112           IF HC.ALTITUDE(.HC) = MASKED
7113              LOOP FOR EVERY .FIRING.HELO IN
7114              UN.EQUIP.LIST(.ATK.HELO.UNIT)
7115              WITH UE.ID(.FIRING.HELO) = HC.UE.ID(.HC)
7116              DO
7117                 FOR EVERY .WPN IN THE UE.WEAPON.SET(.FIRING.HELO)
7118                 WITH HC.WPN.TYPE(.WPN) = FFATGM
7119                 FIND THE FIRST CASE
7120                 IF NONE
7121                    LET HC.ALTITUDE(.HC) = UNMASKED
7122                    LET HC.STATUS(.HC) = ENGAGING
7123                    CREATE A VISIBLE.UNIT
7124                    LET VU.POINTER(VISIBLE.UNIT) = .HC
7125                    LET VU.STATUS(VISIBLE.UNIT) = NO
7126                    FILE THIS VISIBLE.UNIT IN THE
7127                    UN.HC.LOS.LIST(.ENEMY.UNIT)
7128                    CREATE A VISIBLE.UNIT
7129                    LET VU.POINTER(VISIBLE.UNIT) = .ENEMY.UNIT
7130                    LET VU.STATUS(VISIBLE.UNIT) = NO
7131                    LET VU.PREV.ENG(VISIBLE.UNIT) = YES
7132                    FILE THIS VISIBLE.UNIT IN THE
7133                    HC.UN.LOS.LIST(.HC)
7134                 ALWAYS
7135              ENDLOOP
7136           ALWAYS
7137           ACTIVATE_A HELICOPTER.FIRE CALLED .SNAKE.BITE————————————————————————————————————————————————————>(510)
7138           GIVEN
7139           .HC,
7140           .ENEMY.UNIT NOW ''                                                                                    \!
7141           CALL HEL.RANGE.COMPUTE———————————————————————————————————————————————————————————————————————————>(302)
7142           GIVEN
7143           .ATK.HELICOPTER,
7144           .ENEMY.UNIT
7145           YIELDING
```

```
7146          HF.RANGE(.SNAKE.BITE)
7147          LET HF.TEAM(.SNAKE.BITE) = ATK.TEAM
7148          IF HC.DEBUG = "YES"
7149             PRINT 1 LINE WITH .HC,
7150             .ENEMY.UNIT AS FOLLOWS
7151             ----HELO.EN (2)---- .ATK.HELICOPTER = ******   .ENEMY.UNIT = ******
7152          ALWAYS
7153
7154          FOR EVERY .FIRING.HELO IN UN.EQUIP.LIST(.ATK.HELO.UNIT)
7155          WITH UE.ID(.FIRING.HELO) = HC.UE.ID(.HC)
7156          FIND THE FIRST CASE
7157          IF NONE
7158             TRACE
7159             CALL ERROR.STOP----------------------------------------------------------------------------------------------------->(604)
7160          ALWAYS
7161          FILE THIS .SNAKE.BITE IN THE HF.SO.LIST(.FIRING.HELO)
7162       ENDLOOP
7163    ALWAYS
7164
7165    ''THIS PORTION OF THE PROGRAM IS FOR GROUND UNIT TO HELICOPTER
7166
7167    LOOP FOR EVERY .HC IN HT.MEMBER.LIST(ATK.TEAM)
7168    WITH HC.ALTITUDE(.HC) = UNMASKED
7169    DO THE FOLLOWING
7170       LOOP FOR EVERY VISIBLE.HC IN UN.HC.LOS.LIST(.ENEMY.UNIT)
7171       WITH VU.POINTER(VISIBLE.HC) = .HC
7172       DO THE FOLLOWING
7173          LET VU.STATUS(VISIBLE.HC) = YES
7174       ENDLOOP
7175    ENDLOOP
7176    LOOP FOR EACH .FIRING.EQUIP OF UN.EQUIP.LIST(.ENEMY.UNIT)
7177    WITH UE.QUANT(.FIRING.EQUIP) > 0 AND
7178    N.UE.WEAPON.SET(.FIRING.EQUIP) > 0
7179    DO THIS
7180       LOOP FOR EACH .HELICOPTER IN HT.MEMBER.LIST(ATK.TEAM)
7181       WITH HC.ALTITUDE(.HELICOPTER) = UNMASKED
7182       DO THE FOLLOWING
7183          FOR EACH F.T IN UE.TARGET.LIST(.FIRING.EQUIP)
7184          WITH UE.ID(FT.TARGET.EQUIP(F.T)) =
7185          HC.UE.ID(.HELICOPTER)
7186          FIND THE FIRST CASE
7187          IF NONE
7188             CREATE A FIRING.TABLE
7189             LET FT.TGT.UNIT(FIRING.TABLE) = .ATK.HELO.UNIT
7190             LET UN.STATUS(.ATK.HELO.UNIT) = WITHDRAWING
7191             FOR EACH EQ IN UN.EQUIP.LIST(.ATK.HELO.UNIT)
7192             WITH UE.ID(EQ) = HC.UE.ID(.HELICOPTER)
7193             FIND THE FIRST CASE
7194             IF NONE
7195                TRACE
7196 <-----------STOP
7197             OTHERWISE
7198             LET FT.TARGET.EQUIP(FIRING.TABLE) = EQ
7199             FILE THIS FIRING.TABLE IN UE.TARGET.LIST(.FIRING.EQUIP)
7200
7201             IF HC.DEBUG = "YES"
7202                PRINT 1 LINE WITH .ATK.HELO.UNIT,
7203                EQ.NAME(UE.ID(EQ)) AS FOLLOWS
```

```
7204                ——HELO.EN (4)—— ATK.HELO.UNIT=******   TARGET.HELICOPTER=******
7205             ALWAYS
7206
7207          ALWAYS
7208       ENDLOOP
7209
7210       IF N.SO.LIST(.FIRING.EQUIP) LE 0
7211          LOOP FOR I = 1 TO UE.QUANT(.FIRING.EQUIP)
7212          DO THIS
7213             ACTIVATE_A SHOOT.OUT CALLED SHOOT.HC NOW ''———————————————————————————————————————————— \1>(493)
7214             LET FIRING.EQUIP(SHOOT.HC)=.FIRING.EQUIP
7215             LET FIRER.UNIT(SHOOT.HC)=.ENEMY.UNIT
7216             FILE THIS SHOOT.HC IN SO.LIST(.FIRING.EQUIP)
7217             IF HC.DEBUG = "YES"
7218                PRINT 1 LINE WITH .ATK.HELICOPTER,
7219                .ENEMY.UNIT, .ATK.HELICOPTER
7220                AS FOLLOWS
7221                ——HELO.EN(5)—— ATK.HELICOPTER=****** ENEMY.UNIT=*****  HELICOPTER=******
7222             ALWAYS
7223          ENDLOOP
7224       ALWAYS
7225    ENDLOOP
7226
7227 <—RETURN
7228 ENDEVENT
```

E018

```
7229 EVENT INIT.PREPLAN.CAS                    ''
7230    GIVEN
7231    .TARGET,
7232    .AC,
7233    .NUM.AC
7234                                                                                \DYN_ANAL
7235    ADD 1 TO ANAL.CTR(150,1)  ''
7236    NORMALLY MODE IS INTEGER
7237
7238    CREATE A CAS.MISSION CALLED .CAS
7239    IF UN.COLOR(.TARGET) = BLUE
7240       LET .SIDE = RED
7241    ELSE
7242       LET .SIDE = BLUE
7243    ALWAYS
7244    ADD 1 TO SD.NR.CAS.MISSIONS(.SIDE)
7245    LET CMSN.SEQ.NR(.CAS) = SD.NR.CAS.MISSIONS(.SIDE)
7246    LET CMSN.TYPE(.CAS) = PREPLANNED
7247    LET CMSN.SIDE(.CAS) = .SIDE
7248    LET CMSN.AC.TYPE(.CAS) = .AC
7249    LET CMSN.NR.AC(.CAS) = .NUM.AC
7250    LET CMSN.TGT.UNIT(.CAS) = .TARGET
7251    LET CMSN.REQUEST.TIME(.CAS) = TIME.V
7252    LET CMSN.ASP.STATUS(.CAS) = IDLE
7253    IF TACAIR.DEBUG = 1
7254       PRINT 3 LINES WITH TIME.V, SD.NR.CAS.MISSIONS(.SIDE),                    \1
7255       .SIDE, .TARGET  THUS  ''
7256       AT ***.***, EVENT INIT.PREPLAN.CAS
7257       MISSION NR.  ***  FOR SIDE **
7258       TARGET = ******
7259    ALWAYS
7260
7261    CALL CHECK.CAS.CONSTRAINTS------------------------------------------------------------------------------------------>(285)
7262    GIVEN
7263    .CAS
7264
7265 <--RETURN
7266 END
```

```
7267 EVENT MOVE                                                                                                    E019
7268    GIVEN
7269    UNIT
7270
7271    ADD 1 TO ANAL.CTR(151,1)  ''                                                                       \DYN_ANAL
7272    NORMALLY MODE IS INTEGER
7273
7274    CALL CHANGE.LOC-------------------------------------------------------------------------------->( 83)
7275    GIVEN
7276    UNIT
7277
7278 <--EXITEVENT
7279 ENDEVENT
```

```
7280 EVENT OFF.LINE.ATTRITION  ''                                                    E020
7281    ADD 1 TO ANAL.CTR(152,1)  ''                                                \DYN_ANAL
7282    NORMALLY MODE IS INTEGER
7283    DEFINE KILLER, VICTIM, K.WPN  AS TEXT VARIABLES ''                          \TEXT
7284    READ .UNIT.NOS, VICTIM, QUANT, KILLER, K.WPN, ROUNDS
7285    PRINT 2 LINES WITH TIME.V, .UNIT.NOS, VICTIM, QUANT, KILLER,
7286    K.WPN, ROUNDS THUS
7287    !! OFF.LINE.ATTRITION  TIME.V  **.****, UNIT = ******, VICTIM = ******
7288    !!         QUANT = ******, KILLER = ******, WEAPON = ******, ROUNDS = ******
7289    FOR EACH UNIT CALLED .UNIT WITH UNIT.NOS(.UNIT) = .UNIT.NOS
7290    FIND THE FIRST CASE
7291    IF NONE
7292       PRINT 1 LINE WITH TIME.V, .UNIT.NOS THUS
7293       OFF.LINE.ATTRITION  ERROR AT TIME **.**** UNIT ****** DOES NOT EXIST
7294 <------STOP
7295    OTHERWISE
7296    IF UN.COLOR(.UNIT) = BLUE
7297       LET K.SIDE = RED
7298    ELSE
7299       LET K.SIDE = BLUE
7300    ALWAYS
7301    FOR EACH EQUIPMENT CALLED .VICTIM
7302    WITH EQ.NAME(.VICTIM) = VICTIM
7303    FIND THE FIRST CASE
7304    IF NONE
7305       PRINT 2 LINES WITH TIME.V, VICTIM, .UNIT.NOS THUS
7306       OFF.LINE.ATTRITION  ERROR AT TIME **.****, VICTIM ******, UNIT ******
7307       = VICTIM IS NOT A VALID EQUIPMENT NAME =
7308 <------STOP
7309    OTHERWISE
7310    FOR EACH EQUIPMENT CALLED .KILLER
7311    WITH EQ.NAME(.KILLER) = KILLER
7312    FIND THE FIRST CASE
7313    IF NONE
7314       PRINT 2 LINES WITH TIME.V, KILLER, .UNIT.NOS THUS
7315       OFF.LINE.ATTRITION  ERROR AT TIME **.****, KILLER ******, UNIT ******
7316       = KILLER IS NOT A VALID EQUIPMENT NAME =
7317 <------STOP
7318    OTHERWISE
7319    FOR EACH .EQUIP IN UN.EQUIP.LIST(.UNIT)
7320    WITH UE.ID(.EQUIP) = .VICTIM
7321    FIND THE FIRST CASE
7322    IF NONE
7323       PRINT 2 LINES WITH TIME.V, .UNIT.NOS, VICTIM THUS
7324       OFF.LINE.ATTRITION  ERROR AT TIME **.****, UNIT ******, VICTIM ******
7325       = UNIT IS NOT AUTHORIZED THE VICTIM EQUIPMENT =
7326 <------STOP
7327    OTHERWISE
7328    IF K.WPN NE "NONE"
7329       LOOP FOR EVERY TYPE.WEAPON CALLED TW
7330       WITH TW.NAME(TW) = K.WPN
7331       FIND THE FIRST CASE
7332       IF NONE
7333          PRINT 2 LINES WITH TIME.V, .UNIT.NOS, K.WPN THUS
7334          !!  OFF.LINE.ATTRITION ERROR AT TIME **.**** FOR UNIT ******
7335          !!       WEAPON ****** IS NOT CONTAINED IN THE WEAPON FILE
7336 <---------STOP
7337       OTHERWISE
```

```
7338        IF K.SIDE = BLUE
7339           IF TW GT N.B.WPN.TYPE
7340              PRINT 1 LINE WITH K.WPN THUS
7341              !!          ERROR - WEAPON ****** IS NOT A BLUE WEAPON
7342 <------------STOP
7343           OTHERWISE
7344        ELSE
7345           IF K.SIDE NE RED
7346              TRACE
7347 <------------STOP
7348           OTHERWISE
7349           IF TW LE N.B.WPN.TYPE
7350              PRINT 1 LINE WITH K.WPN THUS
7351              !!          ERROR - WEAPON ****** IS NOT A RED WEAPON
7352 <------------STOP
7353           OTHERWISE
7354        ALWAYS
7355        LET .FIRER = TW
7356     ELSE
7357        LET .FIRER = .KILLER
7358     ALWAYS
7359     IF UE.QUANT(.EQUIP) LE 0
7360        PRINT 3 LINES WITH VICTIM THUS
7361        !!        - UNIT  DOES NOT HAVE ANY ****** REMAINING -
7362        !!          = NO KILLS OR EXPENDITURES RECORDED =
7363        !!
7364 <------EXITROUTINE
7365     ALWAYS
7366     IF UE.QUANT(.EQUIP) LT QUANT
7367        PRINT 1 LINE WITH UE.QUANT(.EQUIP), QUANT THUS
7368        !!   = UNIT ONLY HAS ****** PIECES OF EQUIPMENT ( OF ****** DESIRED) =
7369        LET ROUNDS = INT.F((ROUNDS/QUANT)*UE.QUANT(.EQUIP))
7370        LET QUANT = UE.QUANT(.EQUIP)
7371     ALWAYS
7372     IF .UNIT IS IN A FR.UNIT.SET AND N.SO.LIST(.EQUIP) GT 0
7373        LET DEAD.EQ = 0
7374        LOOP FOR EVERY .SO IN SO.LIST(.EQUIP)
7375        DO  ''                                                                                                    \1
7376           IF DEAD.EQ GE QUANT
7377        <------EXITLOOP
7378           OTHERWISE
7379           IF DROP.DEAD.INDICATOR(.SO) = NO
7380              LET DROP.DEAD.INDICATOR(.SO) = YES
7381              INTERRUPT SHOOT.OUT CALLED .SO------------------------------------------------------------------------>(493)
7382              LET TIME.A(.SO) = 1/60  '' HOUR
7383              RESUME SHOOT.OUT CALLED .SO---------------------------------------------------------------------------->(493)
7384           ELSE
7385              ADD 1 TO DEAD.SO
7386           ALWAYS
7387           ADD 1 TO DEAD.EQ
7388        ENDLOOP
7389        IF DEAD.SO GT 0
7390           PRINT 1 LINE WITH DEAD.SO, QUANT THUS
7391           !!   = **** VICTIMS WERE ALREADY KILLED BY OTHERS (OF ****** DESIRED) =
7392           SUBTRACT DEAD.SO FROM QUANT
7393        ALWAYS
7394        IF DEAD.EQ LT QUANT
7395           PRINT 1 LINE WITH DEAD.EQ THUS
```

```
7396          !!                    ONLY *** VICTIMS HAD SHOOT OUTS
7397       ALWAYS
7398    ALWAYS
7399    SUBTRACT QUANT FROM UE.QUANT(.EQUIP)
7400    IF UN.PTR(.UNIT) GT 0 AND UE.CRITICAL.EQUIP.INDIC(.EQUIP) = YES
7401       SUBTRACT QUANT FROM MU.CRIT.NO(UN.PTR(.UNIT))
7402       IF .UNIT IS IN A FR.UNIT.SET
7403          CALL DECIDE GIVING .UNIT, .EQUIP------------------------------------------------------------------------------------------>(324)
7404       ALWAYS
7405    ALWAYS
7406    IF EQ.KV.ID(.KILLER) GT 0 AND EQ.KV.ID(.VICTIM) GT 0
7407       ADD QUANT TO KV.SCORE(K.SIDE, EQ.KV.ID(.KILLER), EQ.KV.ID(.VICTIM))
7408    ALWAYS
7409    IF K.WPN NE "NONE"
7410       ADD ROUNDS TO STW.RND.FIRED(K.SIDE,TW)
7411       IF K.SIDE = BLUE
7412          ADD ROUNDS TO STY.BLUE.EXP(TW,(.VICTIM-N.BLUE.TYPE.EQP))
7413       ELSE
7414          ADD ROUNDS TO STY.RED.EXP(TW-N.B.WPN.TYPE,.VICTIM)
7415       ALWAYS
7416       PRINT 1 LINE WITH ROUNDS THUS
7417       !!       ****** ROUNDS WERE EXPENDED
7418       IF EQ.KV.ID(.KILLER) GT 0
7419          ADD (ROUNDS * TW.RND.WEIGHT(TW)) TO KV.AMMO.CONSUMED
7420          (K.SIDE, EQ.KV.ID(.KILLER))
7421       ALWAYS
7422    ALWAYS
7423    LET DEAD.SENS = 0
7424    LOOP FOR EACH .LINK IN UN.SENSOR.LIST(.UNIT) WITH
7425    US.EQ.ID(.LINK) = UE.ID(.EQUIP)
7426    DO  ''                                                                                                              \1
7427       IF DEAD.SENS GE QUANT
7428    <------EXITLOOP
7429       OTHERWISE
7430       ADD 1 TO DEAD.SENS
7431       PRINT 1 LINE WITH VICTIM THUS
7432       !!          = SENSOR ****** KILLED =
7433       CALL ATTRIT.SENSOR GIVING .LINK, 1.0--------------------------------------------------------------------------------------->(159)
7434    ENDLOOP
7435    IF CT.GROUP(TU.CAT(UN.TYPE.UNIT(.UNIT))) = ARTILLERY
7436       AND UN.BTRY.INDEX(.UNIT) GT 0
7437       LET TGT.BTRY = UN.BTRY.INDEX(.UNIT)
7438       IF TB.HOW.EQ.ID(BY.TYPE(TGT.BTRY)) = .VICTIM
7439          LET DEAD.HOW = 0
7440          LOOP FOR EACH .HOW IN BY.HOW.SET(TGT.BTRY)
7441          DO  ''                                                                                                        \1
7442             IF DEAD.HOW GE QUANT
7443          <------EXITLOOP
7444             OTHERWISE
7445             REMOVE .HOW FROM BY.HOW.SET(TGT.BTRY)
7446             DESTROY THE HOW CALLED .HOW
7447             ADD 1 TO DEAD.HOW
7448             PRINT 1 LINE WITH VICTIM THUS
7449             !!          = HOWITZER ****** KILLED =
7450          ENDLOOP
7451          IF UN.STATUS(.UNIT) = STATIONARY
7452             LET UN.TIME.LAST.MOVE(.UNIT) = TIME.V - TIME.BETWEEN.ARTY.MOVE
7453             CALL FA.BN.MOVEMENT GIVING BY.BN(TGT.BTRY), 0-------------------------------------------------------------------->( 89)
```

```
7454          ALWAYS
7455       ALWAYS
7456    ALWAYS
7457    IF ANALYSIS(6) GT 0
7458       LOOP FOR I = 1 TO QUANT
7459       DO  ''                                                                    \1
7460          WRITE UN.BATTLE.INDEX(.UNIT), TIME.V, .FIRER, .VICTIM
7461          AS "DIR.  ", I 7, S 2, D(7,2), S 2, I 4, S 2, I 4, / USING UNIT 60
7462       ENDLOOP
7463    ALWAYS
7464    PRINT 1 LINE THUS
7465    !!
7466 ENDROUTINE
```

```
7467 EVENT PDB.ACTIVATION                                                                                    E021
7468    GIVEN BTRY,
7469    ACTIVITY.TYPE
7470
7471
7472    ADD 1 TO ANAL.CTR(153,1)  ''                                                                  \DYN_ANAL
7473    NORMALLY MODE IS INTEGER
7474    DEFINE BTRY AS AN INTEGER VARIABLE
7475    DEFINE ACTIVITY.TYPE, RADAR.TYPE AS TEXT VARIABLES ''                                           \TEXT
7476
7477    LET TYPE = BY.TYPE(BTRY)
7478    IF TB.MAX.RANGE(TYPE) < 700  ''IN HD - IT'S A MORTAR
7479       LET RADAR.TYPE = "CM"
7480    ELSE
7481       LET RADAR.TYPE = "CB"
7482    ENDIF
7483    IF UN.COLOR(BY.UNIT(BTRY)) = BLUE
7484       LET  ENEMY = RED
7485    ELSE
7486       LET ENEMY = BLUE
7487    ENDIF
7488
7489    LOOP FOR EACH PDB IN SIDE.PDB.SET(ENEMY)
7490    DO THIS
7491       LET LINK = PDB.US.LINK(PDB)
7492       LET ST = US.SENSOR.TYPE(LINK)
7493       ''UTILIZE ACT.RANGE FUNCTION IN NEXT ASSIGNMENT ---------------------------------------------- \1>(628)
7494       LET RANGE = INT.F(ACT.RANGE(BY.UNIT(BTRY), US.UNIT(LINK))*16.)
7495       LET MODEL=US.MODEL(LINK)
7496       IF ST.NAME(ST) = ACTIVITY.TYPE
7497          IF RANGE <=PDB.RH.RANGE(L.MPDB.RH.LIST(MODEL))
7498             CALL PDB.DETECTION GIVEN PDB, BTRY, RADAR.TYPE, RANGE-------------------------------------->(206)
7499          ENDIF
7500       ENDIF
7501    ENDLOOP
7502
7503 ENDEVENT
```

```
7504 EVENT PDB.OPERATOR                                                                                    E022
7505    GIVEN
7506    PDB,
7507    THIS.UNIT
7508    ADD 1 TO ANAL.CTR(154,1)                                                                           \DYN_ANAL
7509    NORMALLY MODE IS INTEGER
7510    DEFINE BTRY AS A INTEGER VARIABLE
7511    DEFINE XMIT.TIME AS A REAL VARIABLE
7512    LET LINK = PDB.US.LINK(PDB)
7513    LET ST = US.SENSOR.TYPE(LINK)
7514    IF PD.OPERATOR(PDB) = IDLE
7515       LET PD.OPERATOR(PDB) = BUSY
7516       LET XMIT.TIME = UNIFORM.F(REAL.F(ST.MIN.XMIT(ST)),
7517       REAL.F(ST.MAX.XMIT(ST)),1)/10.
7518       SCHEDULE_A PDB.OPERATOR GIVING PDB, THIS.UNIT IN XMIT.TIME MINUTES------------------------------>(387)
7519    ELSE
7520       LET PD.OPERATOR(PDB) = IDLE
7521       REMOVE THIS.UNIT FROM PDB.OP.Q(PDB)
7522       CREATE A TARGET.REPORT CALLED TR
7523       LET BTRY = PD.D.BTRY(THIS.UNIT)
7524       LET TR.TGT.UNIT(TR) = BY.UNIT(BTRY)
7525       LET TR.SENSOR.TYPE(TR) = ST.NAME(ST)
7526       LET TR.SENSOR.ID(TR) = PDB
7527       LET TR.FDC(TR) = US.FDC(LINK)
7528       LET TR.REP.UNIT(TR) = US.UNIT(LINK)
7529       LET TR.CEP(TR) = PD.D.CEP(THIS.UNIT)
7530       IF TR.CEP(TR) = 0
7531          PRINT 1 LINE WITH TR, TR.REP.UNIT(TR) THUS
7532          = = = PDB.OPERATOR TGT ****** FROM UNIT **** HAS 0 CEP
7533          LET TR.CEP(TR) = 1
7534       ALWAYS
7535       ''UTILIZE NORMAL.F FUNCTION IN NEXT 2 CALCULATIONS ------------------------------------------- \1>(641)
7536       LET TR.EST.X(TR) = UN.X.COORD(BY.UNIT(BTRY))+NORMAL.F(00.0,1.0,1)*
7537       TR.CEP(TR)/(1.1774*16.)
7538       LET TR.EST.Y(TR) = UN.Y.COORD(BY.UNIT(BTRY))+NORMAL.F(00.0,1.0,1)*
7539       TR.CEP(TR)/(1.1774*16.)
7540       LET TR.PGM.STATUS(TR) = FALSE
7541       CREATE A TR.DET.LINK CALLED DET.LINK
7542       LET TR.DET.TE(DET.LINK) = EQ.TE.PTR(TB.HOW.EQ.ID(BY.TYPE(BTRY)))
7543       LET TR.DET.QUANT(DET.LINK) = N.BY.HOW.SET(BTRY)
7544       LET TR.DET.ELEM.PROB(DET.LINK) = PD.D.PD(THIS.UNIT)
7545       FILE DET.LINK IN TR.DET.LIST(TR)
7546       LET TR.RECVD.TIME(TR) = TIME.V
7547       LET TR.ABORT.TIME(TR) = TR.RECVD.TIME(TR) + .5
7548       DESTROY THE PD.DET.UNIT CALLED THIS.UNIT
7549       CALL CHK.COMP.TR GIVEN TR, US.FDC(LINK), PDB------------------------------------------------->(168)
7550       YIELDING DUPLICATE
7551       IF DUPLICATE = FALSE
7552          CALL CHK.FD.TR GIVEN TR, US.FDC(LINK), PDB------------------------------------------------->(169)
7553          YIELDING DUPLICATE
7554       ENDIF
7555       IF DUPLICATE = TRUE
7556          REMOVE DET.LINK FROM TR.DET.LIST(TR)
7557          DESTROY THE TR.DET.LINK CALLED DET.LINK
7558          DESTROY THE TARGET.REPORT CALLED TR
7559       ELSE
7560          ACTIVATE_THE TARGET.REPORT CALLED TR NOW------------------------------------------------->(470)
7561       ENDIF
```

```
7562        IF PDB.OP.Q(PDB) IS NOT EMPTY
7563           LET NEXT.UNIT = F.PDB.OP.Q(PDB)
7564           SCHEDULE_A PDB.OPERATOR GIVING PDB, NEXT.UNIT NOW----------->(387)
7565        ENDIF
7566     ENDIF
7567  ENDEVENT
```

```
7568 EVENT POSITION.REPORT                                                                                    E023
7569
7570    ADD 1 TO ANAL.CTR(155,1)  ''                                                                          \DYN_ANAL
7571    NORMALLY MODE IS INTEGER
7572
7573    CALL POSITION.OUT-------------------------------------------------------------------------------->(613)
7574
7575    SCHEDULE_A POSITION.REPORT AT TIME.V + POS.REP.INT  ''------------------------------------  \OPTIMIZE>(389)
7576
7577 <--EXITEVENT
7578 ENDEVENT
```

```
7579 EVENT SCHEDULE.ARTY.MOVEMENT                                                                    E024
7580
7581    ADD 1 TO ANAL.CTR(156,1)  ''                                                                 \DYN_ANAL
7582    NORMALLY MODE IS INTEGER
7583
7584    SCHEDULE_A SCHEDULE.ARTY.MOVEMENT  ''------------------------------------------------------- \OPTIMIZE>(390)
7585    IN TIME.BETWEEN.ARTY.MOVE MINUTES
7586
7587    LOOP FOR EACH FA.BN CALLED .FA.BN
7588    DO
7589       CALL FA.BN.MOVEMENT-------------------------------------------------------------------------->( 89)
7590       GIVEN
7591       .FA.BN, 0
7592    ENDLOOP
7593
7594 <--EXITEVENT
7595 ENDEVENT
```

```
7596 EVENT SEND.TEAM                                                                                                    E025
7597    GIVEN
7598    .FARRP,
7599    .TEAM
7600
7601    ADD 1 TO ANAL.CTR(157,1)  ''                                                                                    \DYN_ANAL
7602    ''THIS EVENT SENDS THE NEXT AVAILABLE TEAM TO THE BATTLE TO
7603    ''REPLACE THE CURRENT TEAM
7604
7605    NORMALLY MODE IS INTEGER
7606
7607    LET HT.STATUS(.TEAM) = MOVING.TO.BATTLE
7608
7609    IF HC.DEBUG = "YES"
7610       PRINT 1 LINE WITH .TEAM, .FARRP, HT.STATUS(.TEAM)
7611       AS FOLLOWS
7612       ----SEND.TEAM----  TEAM=****** FARRP=****** HT.STATUS=******
7613    ALWAYS
7614
7615    ACTIVATE_AN HC.ARRIVE.BATTLE------------------------------------------------------------------------------------>(443)
7616    GIVEN
7617    .FARRP,
7618    .TEAM
7619    IN HT.MOVE.TIME(.TEAM) MINUTES
7620
7621 <--EXITEVENT
7622 ENDEVENT
```

```
7623 EVENT SET.DEBUG                                                    ''          E026
7624
7625    ADD 1 TO ANAL.CTR(158,1)  ''                                                \DYN_ANAL
7626    NORMALLY MODE IS INTEGER
7627
7628    READ DEBUG
7629
7630    WRITE AS /,/," *****  EVENT SET.DEBUG  ****** ",/,/
7631    LIST  TIME.V, EVENT.V '', ZTIME.F, USED.DBANK.V                             \VAX
7632
7633    IF DEBUG > 0
7634       LET BETWEEN.V = 'BETWEEN.ROUTINE'
7635
7636       IF ANALYSIS(3)= TRUE
7637          WRITE TIME.V AS /,D(8,5),/  USING UNIT 50
7638          LOOP FOR EACH UNIT
7639          DO
7640             WRITE UNIT.NOS(UNIT), TU.LEVEL(UN.TYPE.UNIT(UNIT))
7641             AS I 6,S 1,T 6,S 2 ''                                              \TEXT
7642             USING UNIT 50
7643             FOR EACH EQ IN UN.EQUIP.LIST(UNIT)
7644             WRITE EQ.NAME(UE.ID(EQ)), UE.QUANT(EQ)
7645             AS T 6,S 1,I 3,S 2 USING UNIT 50  ''                               \TEXT
7646             WRITE AS / USING UNIT 50
7647          ENDLOOP
7648       ALWAYS
7649    ELSE
7650       LET BETWEEN.V = 0
7651    ALWAYS
7652
7653 <--EXITEVENT
7654 ENDEVENT
```

```
7655 EVENT START.ARTY.MOVEMENT                                                    ''                                                    E027
7656    GIVEN
7657    BTRY,
7658    MOVE.DIST
7659
7660    ADD 1 TO ANAL.CTR(159,1)  ''                                                                                                \DYN_ANAL
7661    NORMALLY MODE IS INTEGER
7662    DEFINE .MOVE TO MEAN 2
7663    DEFINE MOVE.TIME AS A REAL VARIABLE
7664    DEFINE TAC.FAC AS A REAL VARIABLE
7665    DEFINE VIS.FAC AS A REAL VARIABLE
7666
7667    FOR EACH STAM IN EV.S(I.START.ARTY.MOVEMENT)
7668    WHEN BTRY = STAM.BTRY(STAM)
7669    FIND THE FIRST CASE
7670    IF FOUND
7671       TRACE
7672 <------STOP
7673    OTHERWISE
7674
7675    LET .XEND = UN.X.COORD(BY.UNIT(BTRY)) + MOVE.DIST
7676    CALL CHECK.FOR.MINES------------------------------------------------------------------------------------------------------------>(134)
7677    GIVEN
7678    BY.UNIT(BTRY),
7679    .XEND,
7680    UN.Y.COORD(BY.UNIT(BTRY))
7681
7682    YIELDING
7683
7684    .MINE.FLAG
7685    LET .NMINES   N.MO.LIST(BY.UNIT(BTRY))
7686    LOOP FOR E  RY .MO IN MO.LIST(BY.UNIT(BTRY))
7687    DO
7688       REMC.E .MO FROM THE MO.LIST(BY.UNIT(BTRY))
7689       CALL MINE.EFFECTS---------------------------------------------------------------------------------------------------------->(250)
7690       GIVEN
7691       BY.UNIT(BTRY),
7692       MO.X.INTER(.MO),
7693       MO.Y.INTER(.MO),
7694       1,     ''BARRIER MINE
7695       0,
7696       .MINE.FLAG
7697       YIELDING
7698       .DELAY  ''MINUTES
7699       ADD .DELAY TO .TOTAL.DELAY
7700       DESTROY THE MINE.OBSTACLE CALLED .MO
7701    ENDLOOP
7702
7703    IF MF.DEBUG = TRUE AND .NMINES GT 0
7704       PRINT 2 LINES WITH TIME.V, BY.UNIT(BTRY),
7705       UN.X.COORD(BY.UNIT(BTRY)), UN.Y.COORD(BY.UNIT(BTRY)),
7706       .XEND, .TOTAL.DELAY, .NMINES THUS
7707       ==START.ARTY.M AT **.***HRS UNIT=**** FROM(*****,*****)
7708       TO X=*****, DELAYED ***MIN BY *** MINES
7709    ALWAYS
7710
7711    LET TAC.FAC = 1.00  '' PROG DOES NOT ALLOW FOR ARTY TO MOVE ADMIN
7712    IF NITE.OR.DAY = NITE
```

```
7713        LET VIS.FAC = NITE.MOV.FAC
7714     ELSE
7715        LET VIS.FAC = 1.00
7716     ALWAYS
7717
7718     CALL GET.TERRAIN------------------------------------------------------------------------------------------------------------>(336)
7719     YIELDING
7720     TER.TYPE
7721
7722     IF ANALYSIS(1) = TRUE
7723        USE UNIT 42 FOR OUTPUT
7724        PRINT 1 LINE WITH BTRY, BY.STATUS(BTRY), TIME.V THUS
7725        BTRY *** CHANGES FROM STATUS * TO STATUS 2 AT **.******
7726        USE UNIT 6 FOR OUTPUT
7727     ALWAYS
7728     LET BY.STATUS(BTRY) = .MOVE
7729     LET UN.STATUS(BY.UNIT(BTRY)) = MOVING
7730     LET RATE = TU.MOV.RATE(UN.TYPE.UNIT(BY.UNIT(BTRY))) * TAC.FAC
7731     * VIS.FAC * MOV.FAC(TER.TYPE)
7732     LET MOVE.TIME = ABS.F(MOVE.DIST/RATE) + .TOTAL.DELAY/MINUTES.V
7733     SCHEDULE_A STOP.ARTY.MOVEMENT----------------------------------------------------------------------------------------------->(407)
7734     GIVEN
7735     BTRY,
7736     MOVE.DIST
7737     IN MOVE.TIME HOURS
7738     ''MOVE HALF THE DISTANCE NOW
7739     ADD MOVE.DIST/2 TO UN.X.COORD(BY.UNIT(BTRY))
7740
7741 <--EXITEVENT
7742 ENDEVENT
```

```
7743 EVENT START.BATTLE                                                    ''                                                    E028
7744    GIVEN
7745    TERRAIN,
7746    BLUE.MISSION,
7747    RED.MISSION,
7748    BLUE.UNITS,
7749    RED.UNITS,
7750    ARG.ARRAY
7751
7752    ADD 1 TO ANAL.CTR(160,1)  ''                                                                                    \DYN_ANAL
7753    NORMALLY MODE IS INTEGER
7754    DEFINE THETA,RANGE.RATIO,DELTA.TIME AS REAL VARIABLES
7755    DEFINE .BATTLE.STATUS AS A TEXT VARIABLE    ''                                                                  \TEXT
7756    DEFINE UNIT AS AN INTEGER VARIABLE
7757    DEFINE TYPE.TEAM AND .NEW.SECTOR AS INTEGER VARIABLES
7758    DEFINE BLUE.UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
7759    DEFINE RED.UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
7760    DEFINE ARG.ARRAY AS AN INTEGER, 1-DIMENSIONAL ARRAY
7761
7762    LET NO.BLUE.UNITS = DIM.F(BLUE.UNITS(*))
7763    LET NO.RED.UNITS = DIM.F(RED.UNITS(*))
7764    IF ARG.ARRAY(*) GT 0
7765       LET .NEW.BATTLE = NO
7766    ELSE
7767       LET .NEW.BATTLE = YES
7768    ALWAYS
7769
7770    IF UN.COLOR(BLUE.UNITS(1)) EQ UN.COLOR(RED.UNITS(1))
7771       LIST BLUE.UNITS,
7772       RED.UNITS
7773       WRITE AS /," *** DON'T START BATTLE WITH FRIENDLY FIRE ",/
7774       TRACE
7775       CALL ERROR.STOP------------------------------------------------------------------------------------------------>(604)
7776    ALWAYS
7777
7778    CREATE A BATTLE
7779
7780    IF ANALYSIS(6) GT 0
7781       IF .NEW.BATTLE = YES
7782          LET .BATTLE.STATUS = "BGIN"
7783       ELSE
7784          LET .BATTLE.STATUS = "CONT"
7785       ALWAYS
7786       WRITE .BATTLE.STATUS, BATTLE, TIME.V,
7787       NO.BLUE.UNITS, NO.RED.UNITS
7788       AS T 4, S 2, I 7, S 2, D(7,2), S 2, I 4, ''                                                                  \TEXT
7789       S 2, I 4, /   USING UNIT 60
7790    ALWAYS
7791
7792    IF .NEW.BATTLE = NO
7793       LET BTL.TIME.OF.DAY(BATTLE) = ARG.ARRAY(2)
7794       LET BTL.SEQ.NO(BATTLE) = ARG.ARRAY(1)
7795    ELSE
7796       LET BTL.TIME.OF.DAY(BATTLE) = TIME.V * 1000.
7797       ADD 1 TO CUM.NO.BATTLES
7798       LET BTL.SEQ.NO(BATTLE) = CUM.NO.BATTLES
7799    ALWAYS
7800    LET BTL.TERRAIN.TYPE(BATTLE) = TERRAIN
```

```
7801    LET BTL.BL.UNITS(BATTLE) = BLUE.UNITS(*)
7802    LET BTL.RD.UNITS(BATTLE) = RED.UNITS(*)
7803    FILE THIS BATTLE IN THE BATTLE.SET
7804
7805    ''SET UP THE FORCES
7806    CALL CREATE.FORCE----------------------------------------------------------------------->( 58)
7807    GIVEN
7808    BATTLE,
7809    BLUE,
7810    BLUE.UNITS(*),
7811    BLUE.MISSION
7812    CALL CREATE.FORCE----------------------------------------------------------------------->( 58)
7813    GIVEN
7814    BATTLE,
7815    RED,
7816    RED.UNITS(*),
7817    RED.MISSION
7818
7819    ''COUNT THE NUMBER OF BLUE PLATOONS OF EACH TYPE THAT THE FORCE HAS
7820    ''CALL PLAT.COUNT----------------------------------------------------------------------->(714)
7821    ''      GIVEN
7822    ''          BLUE.UNITS(*)
7823    ''      YIELDING
7824    ''          NO.BLUE.INF.PLAT,
7825    ''          NO.BLUE.MECH.PLAT,
7826    ''          NO.BLUE.ARMOR.PLAT,
7827    ''          NO.BLUE.AH.TEAM,
7828    ''          NO.BLUE.HQ
7829
7830    ''COUNT THE NUMBER OF RED PLATOONS OF EACH TYPE THAT THE FORCE HAS
7831    ''CALL PLAT.COUNT----------------------------------------------------------------------->(714)
7832    ''      GIVEN
7833    ''          RED.UNITS(*)
7834    ''      YIELDING
7835    ''          NO.RED.INF.PLAT,
7836    ''          NO.RED.MECH.PLAT,
7837    ''          NO.RED.ARMOR.PLAT,
7838    ''          NO.RED.AH.TEAM,
7839    ''          NO.RED.HQ
7840
7841    IF .NEW.BATTLE = YES
7842       ''CALCULATE PRIORITIES FOR HELICOPTER SUPPORT OF RED AND BLUE SIDES
7843       CALL UNIT.PRIORITY----------------------------------------------------------------->(305)
7844       GIVING
7845       BLUE,
7846       BATTLE
7847       YIELDING
7848       BL.PRIORITY
7849       LET BLUE.HB.PRIORITY(BATTLE) = BL.PRIORITY
7850       CALL UNIT.PRIORITY----------------------------------------------------------------->(305)
7851       GIVING
7852       RED,
7853       BATTLE
7854       YIELDING
7855       RD.PRIORITY
7856       LET RED.HB.PRIORITY(BATTLE) = RD.PRIORITY
7857    ELSE
7858       LET BLUE.HB.PRIORITY(BATTLE) = ARG.ARRAY(6)
```

```
7858        LET RED.HB.PRIORITY(BATTLE) = ARG.ARRAY(5)
7860    ALWAYS
7861
7862    ''FIND A BATTLE THAT MATCHES IN BOTH UNITS AND MISSIONS
7863    ''FOR EVERY TYPE.BATTLE.FIELD
7864    ''WITH TBF.NO.BL.UNITS(TYPE.BATTLE.FIELD) = NO.BLUE.UNITS
7865    ''AND TBF.NO.RD.UNITS(TYPE.BATTLE.FIELD) = NO.RED.UNITS
7866    ''AND TBF.BL.MISSION(TYPE.BATTLE.FIELD) = BLUE.MISSION
7867    ''AND TBF.RD.MISSION(TYPE.BATTLE.FIELD) = RED.MISSION
7868    ''AND TBF.RD.ARMOR.UNITS(TYPE.BATTLE.FIELD) = NO.RED.ARMOR.PLAT
7869    ''AND TBF.BL.ARMOR.UNITS(TYPE.BATTLE.FIELD) = NO.BLUE.ARMOR.PLAT
7870    ''AND TBF.RD.MECH.UNITS(TYPE.BATTLE.FIELD) = NO.RED.MECH.PLAT
7871    ''AND TBF.BL.MECH.UNITS(TYPE.BATTLE.FIELD) = NO.BLUE.MECH.PLAT
7872    ''AND TBF.RD.INF.UNITS(TYPE.BATTLE.FIELD) = NO.RED.INF.PLAT
7873    ''AND TBF.BL.INF.UNITS(TYPE.BATTLE.FIELD) = NO.BLUE.INF.PLAT
7874    ''AND TBF.RD.HQ.UNITS=NO.RED.HQ
7875    ''AND TBF.BL.HQ.UNITS=NO.BLUE.HQ
7876    ''
7877    ''ADD ATTACK HELICOPTERS TO THE CONSIDERED UNITS
7878    ''
7879    ''FIND THE FIRST CASE,
7880    ''WHEN NONE,                                                                                                    \1
7881
7882    LET TYPE.BATTLE.FIELD = 1
7883    ''LET TBF.NO.BL.UNITS(TYPE.BATTLE.FIELD) = NO.BLUE.UNITS
7884    ''LET TBF.NO.RD.UNITS(TYPE.BATTLE.FIELD) = NO.RED.UNITS
7885    LET TBF.BL.MISSION(TYPE.BATTLE.FIELD) = BLUE.MISSION
7886    LET TBF.RD.MISSION(TYPE.BATTLE.FIELD) = RED.MISSION
7887    ''LET TBF.RD.ARMOR.UNITS(TYPE.BATTLE.FIELD) = NO.RED.ARMOR.PLAT
7888    ''LET TBF.BL.ARMOR.UNITS(TYPE.BATTLE.FIELD) = NO.BLUE.ARMOR.PLAT
7889    ''LET TBF.RD.MECH.UNITS(TYPE.BATTLE.FIELD) = NO.RED.MECH.PLAT
7890    ''LET TBF.BL.MECH.UNITS(TYPE.BATTLE.FIELD) = NO.BLUE.MECH.PLAT
7891    ''LET TBF.RD.INF.UNITS(TYPE.BATTLE.FIELD) = NO.RED.INF.PLAT
7892    ''LET TBF.BL.INF.UNITS(TYPE.BATTLE.FIELD) = NO.BLUE.INF.PLAT
7893    ''LET TBF.RD.HQ.UNITS(TYPE.BATTLE.FIELD) = NO.RED.HQ
7894    ''LET TBF.BL.HQ.UNITS(TYPE.BATTLE.FIELD) = NO.BLUE.HQ
7895
7896    CALL GENERAL.BATTLE------------------------------------------------------------------------------------------>( 67)
7897    GIVEN
7898    BLUE.UNITS(*),
7899    RED.UNITS(*)
7900
7901    ''HAVING FOUND A PRESTORED BATTLE, INITIALIZE
7902    LET BTL.FIELD(BATTLE) = TYPE.BATTLE.FIELD
7903    LET BTL.WIDTH(BATTLE) = TBF.WIDTH(TYPE.BATTLE.FIELD)
7904
7905    ''TRANSLATE THE PRESTORED BATTLE TO THE ACTUAL BATTLE
7906    CALL ORIENTATION---------------------------------------------------------------------------------------------->( 71)
7907    GIVEN
7908    BATTLE
7909    YIELDING
7910    RANGE.RATIO,
7911    X.BLUE,
7912    Y.BLUE,
7913    X.RED,
7914    Y.RED,
7915    X,
7916    Y,
```

```
7917    THETA
7918    LOOP FOR EVERY TYPE.TEAM OF TEAM.TYPES(TYPE.BATTLE.FIELD)
7919    DO THE FOLLOWING
7920       LET TT.IN.USE(TYPE.TEAM) = NO
7921    ENDLOOP
7922    CALL UNIT.ASSIGNMENT------------------------------------------------------------------------------------------------------>( 75)
7923    GIVEN
7924    BLUE,
7925    BLUE.UNITS(*),
7926    TYPE.BATTLE.FIELD,
7927    THETA,
7928    X,
7929    Y
7930    CALL UNIT.ASSIGNMENT------------------------------------------------------------------------------------------------------>( 75)
7931    GIVEN
7932    RED,
7933    RED.UNITS(*),
7934    TYPE.BATTLE.FIELD,
7935    THETA,
7936    X,
7937    Y
7938
7939    ''PUT EACH UNIT ON THE PATH
7940
7941    PRINT 1 LINE WITH BTL.SEQ.NO(BATTLE), TIME.V THUS
7942    ==========UNITS IN BATTLE ******  (TIME.V = **.****)==========
7943
7944    LOOP FOR EACH FORCE OF THE BTL.FORCE.SET(BATTLE)
7945    DO THE FOLLOWING
7946       LOOP FOR EACH UNIT OF THE FR.UNIT.SET(FORCE)
7947       DO THE FOLLOWING
7948          LET UN.ENGAGED.INDEX(UNIT) = 1
7949          IF UN.COLOR(UNIT) = BLUE
7950             LET UN.MISSION(UNIT) = BLUE.MISSION
7951          ELSE
7952             LET UN.MISSION(UNIT) = RED.MISSION
7953          ALWAYS
7954          LET FRACTION = (MU.CRIT.NO(UN.PTR(UNIT)) /
7955          TU.CRIT.NO(UN.TYPE.UNIT(UNIT)))
7956          *100
7957          PRINT 1 LINE WITH
7958          UNIT.NOS.(UNIT),
7959          SHADE(UN.COLOR(UNIT)), MN.NAME(UN.MISSION(UNIT)),
7960          FRACTION THUS
7961          UNIT  ******  ON SIDE ******  WITH MISSION ******  AT ***% STRENGTH
7962
7963          IF DEBUG > 0
7964             LET UN = UNIT
7965             LIST ATTRIBUTES OF UNIT CALLED UN
7966             LOOP FOR EACH UEL IN UN.EQUIP.LIST(UN),
7967             DO
7968                LIST ATTRIBUTES OF UE.LINK CALLED UEL
7969                FOR EACH SO IN SO.LIST(UEL),
7970                LIST ATTRIBUTES OF SHOOT.OUT CALLED SO
7971             ENDLOOP
7972          ALWAYS
7973          IF UN.BATTLE.INDEX(UNIT) <= 0
7974             LET UN.BATTLE.INDEX(UNIT) = BATTLE
```

```
7975          ELSE
7976             CALL ERROR.STOP------------------------------------------------------------------------------------------>(604)
7977          ALWAYS
7978          IF UN.MISSION(UNIT) = DELAY OR
7979             UN.MISSION(UNIT) = DEFEND OR
7980             UN.MISSION(UNIT) = AMBUSH,
7981             LET UN.STATUS(UNIT) = STATIONARY
7982          ELSE
7983             LET UN.STATUS(UNIT) = ADVANCING
7984             LET UN.TIME.LAST.MOVE(UNIT) = TIME.V
7985          ALWAYS
7986          IF UN.MISSION(UNIT) = ATTACK
7987             OR UN.MISSION(UNIT) = PROBE
7988             OR UN.MISSION(UNIT) = PATROL,
7989             FOR EACH POINT OF THE UN.PATH(UNIT)
7990             FIND THE FIRST CASE
7991             IF NONE
7992                TRACE
7993                CALL ERROR.STOP------------------------------------------------------------------------------------->(604)
7994             ALWAYS
7995             LET UN.X.COORD(UNIT) = P.X(POINT)
7996             LET UN.Y.COORD(UNIT) = P.Y(POINT)
7997             LET UN.POSITION.INDEX(UNIT) = 1
7998       <-----CYCLE
7999          OTHERWISE
8000          LET UN.X.COORD(UNIT) = P.X(L.UN.PATH(UNIT))
8001          LET UN.Y.COORD(UNIT) = P.Y(L.UN.PATH(UNIT))
8002          LET UN.POSITION.INDEX(UNIT) = N.UN.PATH(UNIT)
8003       ENDLOOP
8004    ENDLOOP
8005
8006    ''ADJUST THE OPENING RANGES
8007    IF BLUE.MISSION = ATTACK
8008       OR BLUE.MISSION = PATROL
8009       OR BLUE.MISSION = PROBE,
8010       IF RED.MISSION = ATTACK
8011          OR RED.MISSION = PATROL
8012          OR RED.MISSION = PROBE,
8013          IF NO.RED.UNITS GT NO.BLUE.UNITS,
8014             CALL ADJUST------------------------------------------------------------------------------------------>( 77)
8015             GIVEN
8016             RED.UNITS(*),
8017             X.BLUE,
8018             Y.BLUE,
8019             RANGE.RATIO
8020          ELSE
8021             CALL ADJUST------------------------------------------------------------------------------------------>( 77)
8022             GIVEN
8023             BLUE.UNITS(*),
8024             X.RED,
8025             Y.RED,
8026             RANGE.RATIO
8027          ALWAYS
8028       ELSE
8029          CALL ADJUST------------------------------------------------------------------------------------------>( 77)
8030          GIVEN
8031          BLUE.UNITS(*),
8032          X.RED,
```

```
8033          Y.RED,
8034          RANGE.RATIO
8035       ALWAYS
8036    ELSE
8037       IF RED.MISSION = ATTACK
8038          OR RED.MISSION = PATROL
8039          OR RED.MISSION = PROBE,
8040          CALL ADJUST----------->( 77)
8041          GIVEN
8042          RED.UNITS(*),
8043          X.BLUE,
8044          Y.BLUE,
8045          RANGE.RATIO
8046          ''DON'T ADJUST THE RANGE IF BOTH SIDES ARE DEFENDING,                    \1
8047          ''DELAYING, OR AMBUSHING
8048       ALWAYS
8049    ALWAYS
8050
8051    LOOP FOR EACH .FORCE IN BTL.FORCE.SET(BATTLE)
8052    DO
8053       LOOP FOR EACH .UNIT IN FR.UNIT.SET(.FORCE)
8054       DO
8055          CALL DEQ.FEBA.SET----------->(325)
8056          GIVEN
8057          .UNIT
8058          YIELDING
8059          OLD.SECTOR
8060          CALL ENQ.FEBA.SET----------->(331)
8061          GIVEN
8062          .UNIT,
8063          OLD.SECTOR
8064          YIELDING
8065          .NEW.SECTOR
8066       ENDLOOP
8067    ENDLOOP
8068
8069    ''PUT UNITS THAT ARE IN LOS IN THE LOS LIST
8070    CALL LINE.OF.SIGHT---------->( 96)
8071    GIVEN
8072    RED.UNITS(*),
8073    BLUE.UNITS(*),
8074    TERRAIN
8075    CALL INITIAL.DETECT---------->( 93)
8076    GIVEN
8077    BATTLE
8078    CALL INITIAL.MOVE---------->( 94)
8079    GIVEN
8080    BATTLE
8081    CALL SWITCH.FO---------->(238)
8082    GIVEN
8083    BATTLE,
8084    1
8085
8086    IF HC.SWITCH = 0
8087 <------EXITEVENT
8088    OTHERWISE
8089
8090    IF .NEW.BATTLE = YES
```

```
8091        ''ATTEMPT TO ASSIGN HELICOPTERS TO THE BATTLE
8092        IF BLUE.HB.PRIORITY(BATTLE) GT 0
8093           LET SIDE = BLUE
8094           CALL FARRP.CHECK------------------------------------------------------------------------------------------>(297)
8095           GIVING
8096           BATTLE,
8097           SIDE
8098           YIELDING
8099           FARRP
8100
8101           IF HC.DEBUG = "YES"
8102              PRINT 1 LINE WITH BTL.SEQ.NO(BATTLE), FARRP, SIDE AS FOLLOWS
8103              ----START.BATTLE----  BATTLE=****** FARRP=****** SIDE=*****
8104           ALWAYS
8105
8106           IF FARRP > 0
8107              LET BTL.BL.FARRP(BATTLE) = FARRP
8108              LET FP.BATTLE(FARRP) = BATTLE
8109              LET BLUE.HB.PRIORITY(BATTLE) = -1
8110              CALL EMPLOY.HELICOPTERS---------------------------------------------------------------------------------->(289)
8111              GIVING
8112              FARRP,
8113              BATTLE
8114           ALWAYS
8115        ALWAYS
8116        IF RED.HB.PRIORITY(BATTLE) GT 0
8117           LET SIDE = RED
8118           CALL FARRP.CHECK------------------------------------------------------------------------------------------>(297)
8119           GIVING
8120           BATTLE,
8121           SIDE
8122           YIELDING
8123           FARRP
8124
8125           IF HC.DEBUG = "YES"
8126              PRINT 1 LINE WITH BTL.SEQ.NO(BATTLE), FARRP, SIDE AS FOLLOWS
8127              ----START.BATTLE----  BATTLE=****** FARRP=****** SIDE=*****
8128           ALWAYS
8129
8130           IF FARRP > 0
8131              LET BTL.RD.FARRP(BATTLE) = FARRP
8132              LET FP.BATTLE(FARRP) = BATTLE
8133              LET RED.HB.PRIORITY(BATTLE) = -1
8134              CALL EMPLOY.HELICOPTERS---------------------------------------------------------------------------------->(289)
8135              GIVING
8136              FARRP,
8137              BATTLE
8138           ALWAYS
8139        ALWAYS
8140     ELSE
8141        LOOP FOR EACH SIDE
8142        DO THIS
8143           IF SIDE = RED
8144              LET .FARRP = ARG.ARRAY(3)
8145              LET .TEAM = ARG.ARRAY(7)
8146              LET BTL.RD.FARRP(BATTLE) = .FARRP
8147              LET BTL.RD.HC.TEAM(BATTLE) = .TEAM
8148           ELSE
```

```
8149              LET .FARRP = ARG.ARRAY(4)
8150              LET .TEAM = ARG.ARRAY(8)
8151              LET BTL.BL.FARRP(BATTLE) = .FARRP
8152              LET BTL.BL.HC.TEAM(BATTLE) = .TEAM
8153           ALWAYS
8154           IF .FARRP > 0
8155              LET FP.BATTLE(.FARRP) = BATTLE
8156              IF .TEAM > 0
8157                 LET DELTA.TIME =
8158                 (TIME.V - HT.ARR.BTL.TIME(.TEAM)) * MINUTES.V
8159                 FOR EVERY .HAB IN EV.S(I.HC.ARRIVE.BATTLE)
8160                 WITH AB.FARRP(.HAB) = .FARRP AND
8161                 AB.TEAM(.HAB) = .TEAM
8162                 FIND THE FIRST CASE
8163                 IF NONE
8164                    SUBTRACT DELTA.TIME FROM HT.LOITER.TIME(.TEAM)
8165                    ACTIVATE_A HC.ARRIVE.BATTLE------------------------------------------------------------------------------------>(443)
8166                    GIVEN
8167                    .FARRP,
8168                    .TEAM NOW ''                                                                                                  \1
8169                 ALWAYS
8170              ALWAYS
8171           ALWAYS
8172        ENDLOOP
8173     ALWAYS
8174
8175 <--EXITEVENT
8176 ENDEVENT
```

```
8177 EVENT START.MOVE                                                                                    E029
8178    GIVEN
8179    UNT.,
8180    XCOR.,
8181    YCOR.,
8182    TYPE.,
8183    ORDR.
8184
8185    ADD 1 TO ANAL.CTR(161,1)  ''                                                                \DYN_ANAL
8186    ''THIS EVENT STARTS THE MOVEMENT OF A UNIT IN RESPONSE TO
8187    ''  EITHER A MOVDIS OR A MOVCOR ORDER.  IT CHECKS THE TYPE
8188    ''  OF TERRAIN AND DETERMINES THE MOVEMENT RATE OF THE MOVE.
8189    ''  IT SCHEDULES THE NEXT UPDATE OF LOCATION OF THE UNIT IN
8190    ''  THE TIME IT WILL TAKE THE UNIT TO TRAVEL 500 METERS.
8191
8192    NORMALLY MODE IS INTEGER
8193    DEFINE TAC.FAC AND VIS.FAC AS REAL VARIABLES
8194    DEFINE TYPE. AS A TEXT VARIABLE  ''                                                          \TEXT
8195    DEFINE MAN.UNIT AS A VARIABLE
8196    DEFINE ADMIN.MOV.FAC AS A REAL VARIABLE
8197
8198    LET ADMIN.MOV.FAC = 3.00 '' MOVE OUT 3 TIMES FASTER THAN CBT
8199    '' DEPLOYED MOVEMENT SPEED - CONVOY
8200    LET TAC.FAC=1
8201    IF TYPE.="TACTIC" OR TYPE.="REINF"
8202       LET TAC.FAC=TAC.MOV.FAC.
8203    ALWAYS
8204    LET VIS.FAC=1
8205    IF NITE.OR.DAY=NITE
8206       LET VIS.FAC=NITE.MOV.FAC
8207    ALWAYS
8208
8209    IF UN.RADIUS(UNT.) = 333  '' THIS IS A CBT INEFF UNIT
8210       LET RATE.MOVE = TU.MOV.RATE(UN.TYPE.UNIT(UNT.)) * VIS.FAC
8211       * ADMIN.MOV.FAC
8212    ELSE  '' UNIT IS COMBAT EFFECTIVE
8213       CALL GET.TERRAIN------------------------------------------------------------------------------------------------------------>(336)
8214       YIELDING
8215       TER.TYPE
8216       LET RATE.MOVE=TU.MOV.RATE(UN.TYPE.UNIT(UNT.))*TAC.FAC*VIS.FAC
8217       *MOV.FAC(TER.TYPE.)
8218    ALWAYS
8219    IF UN.COLOR(UNT.)=BLUE
8220       IF XCOR.-UN.X.COORD(UNT.) < 0
8221          LET UN.STATUS(UNT.)=WITHDRAWING
8222       ELSE
8223          LET UN.STATUS(UNT.)=ADVANCING
8224       ALWAYS
8225    ELSE
8226       IF XCOR.-UN.X.COORD(UNT.) < 0
8227          LET UN.STATUS(UNT.)=ADVANCING
8228       ELSE
8229          LET UN.STATUS(UNT.)=WITHDRAWING
8230       ALWAYS
8231    ALWAYS
8232
8233    LOOP FOR EVERY MAN.UNIT IN MU.TF.LIST(UN.PTR(UNT.))
8234    DO
```

```
8235        CALL DEQ.FEBA.SET------------------------------------------------------------------------------------------>(325)
8236        GIVEN
8237        MU.UNIT.ID(MAN.UNIT)
8238        YIELDING
8239        OLD.SECTOR
8240        CALL ENQ.FEBA.SET------------------------------------------------------------------------------------------>(331)
8241        GIVEN
8242        MU.UNIT.ID(MAN.UNIT),
8243        OLD.SECTOR
8244        YIELDING
8245        SECT.
8246     ENDLOOP
8247
8248     CALL DEQ.FEBA.SET------------------------------------------------------------------------------------------>(325)
8249     GIVEN
8250     UNT.
8251     YIELDING
8252     OLD.SECTOR
8253     CALL ENQ.FEBA.SET------------------------------------------------------------------------------------------>(331)
8254     GIVEN
8255     UNT.,
8256     OLD.SECTOR
8257     YIELDING
8258     SECT.
8259
8260     CALL CHECK.FOR.MINES---------------------------------------------------------------------------------------->(134)
8261     GIVEN
8262     UNT.,
8263     XCOR.,
8264     YCOR.
8265
8266     YIELDING
8267     .MINE.FLAG
8268
8269     SKIP 1 LINE
8270     PRINT 4 LINES WITH UNIT.NOS(UNT.),TIME.V,TYPE.,UN.X.COORD(UNT.)*1.6,
8271     UN.Y.COORD(UNT.)*1.6, XCOR*1.6,YCOR*1.6, 16 * RATE.MOVE,
8272     N.MO.LIST(UNT.) THUS
8273     MOVEMENT STARTED FOR UNIT ****** AT **.**** HOURS
8274     A ****** MOVE FROM ****,**** TO ****,****
8275     AT THE RATE OF ***** METERS PER HOUR
8276     INTERSECTING *** MINEFIELDS
8277     SKIP 1 LINE
8278
8279     ''ASSESS DAMAGES FOR MINES IN THE NEXT 500 METERS
8280     LOOP FOR EVERY .MO IN THE MO.LIST(UNT.)
8281     DO
8282        LET .DELTAX = UN.X.COORD(UNT.) - MO.X.INTER(.MO)
8283        LET .DELTAY = UN.Y.COORD(UNT.) - MO.Y.INTER(.MO)
8284        IF SQRT.F(.DELTAX**2 + .DELTAY**2)  LE 31.25   ''HDM
8285           ''A MEMBER OF THE TASK FORCE WILL HIT THE FIELD
8286           ''CHOOSE WHICH AND ASSESS DAMAGES
8287           IF N.MU.TF.LIST(UN.PTR(UNT.)) = 0
8288              LET .TGT = UNT.
8289           ELSE
8290              LET .TGT = RANDI.F(1, N.MU.TF.LIST(UN.PTR(UNT.)),
8291              RN.SEED)
8292              LET .COUNTER = 1
```

```
8293            FOR EVERY .MU IN MU.TF.LIST(UN.PTR(UNT.))
8294            UNTIL .COUNTER = .TGT
8295            ADD 1 TO .COUNTER
8296            LET .TGT = MU.UNIT.ID(.MU)
8297         ALWAYS
8298         CALL MINE.EFFECTS------------------------------------------------------------------------------------------>(250)
8299         GIVEN
8300         .TGT,
8301         MO.X.INTER(.MO),
8302         MO.Y.INTER(.MO),
8303         1,    ''BARRIER MINE
8304         0, ''NO FIRE MISSION
8305         .MINE.FLAG
8306         YIELDING
8307         .DELAY
8308         ADD .DELAY TO .TOTAL.DELAY
8309         CREATE AN AWARE.UNIT
8310         LET AU.UNIT.ID(AWARE.UNIT) = UNT.
8311         FILE THIS AWARE.UNIT IN AU.LIST(MO.MINEFIELD(.MO))
8312         REMOVE .MO FROM THE MO.LIST(UNT.)
8313         DESTROY THE MINE.OBSTACLE CALLED .MO
8314      ALWAYS
8315   ENDLOOP
8316
8317   IF MF.DEBUG = TRUE AND .TOTAL.DELAY GT 0
8318      PRINT 1 LINE WITH UNT., .TOTAL.DELAY, .TGT THUS
8319      ===START.MOVE  TF LDR = ****, DELAY = ****MIN, UNIT HIT = ****
8320   ALWAYS
8321
8322   LOOP FOR EACH MAN.UNIT IN MU.TF.LIST(UN.PTR(UNT.))
8323   DO
8324      LET UN.STATUS(MU.UNIT.ID(MAN.UNIT))=UN.STATUS(UNT.)
8325   ENDLOOP
8326   IF ORD.TYPE(ORDR.)="MOVCOR"AND TYPE.="TACTIC"
8327      CALL CHECK.PROX------------------------------------------------------------------------------------------>(138)
8328      GIVEN
8329      UNT.,
8330      SECT.,
8331      ACT.BATTLE.RANGE
8332      YIELDING
8333      PROX.TEST.,
8334      EN.UNITS(*),
8335      BATTLE.NUM
8336      IF PROX.TEST.=YES
8337         IF BATTLE.NUM > 0
8338            CALL INTER.BATTLE------------------------------------------------------------------------------------>(146)
8339            GIVEN
8340            UNT.,
8341            BATTLE.NUM,
8342            ORDR.
8343            RELEASE EN.UNITS.(*)
8344         ELSE
8345            CALL PROX.POS---------------------------------------------------------------------------------------->(117)
8346            GIVEN
8347            UNT.,
8348            ORDR.,
8349            EN.UNITS.(*)
8350         ALWAYS
```

```
8351 <--------EXITEVENT
8352        OTHERWISE
8353        RELEASE EN.UNITS.(*)
8354     ALWAYS
8355
8356     SCHEDULE_AN UPDATE.LOC------------------------------------------------------------------------------------------>(408)
8357     GIVEN
8358     UNT.,
8359     XCOR.,
8360     YCOR.,
8361     RATE.MOVE,
8362     TYPE.,
8363     ORDR.
8364
8365     IN (LOC.UPDATE.FREQ/RATE.MOVE + .TOTAL.DELAY + RANDOM.F(RN.SEED))
8366     MINUTES
8367
8368 <--EXITEVENT
8369 ENDEVENT
```

```
8370 EVENT STOP.ARTY.MOVEMENT                                                                                    E030
8371    GIVEN
8372    BTRY,
8373    MOVE.DIST
8374
8375    ADD 1 TO ANAL.CTR(162,1)  ''                                                                    \DYN_ANAL
8376    NORMALLY MODE IS INTEGER
8377    DEFINE OCCUPATION TO MEAN 1
8378
8379    FOR EACH SPAM IN EV.S(I.STOP.ARTY.MOVEMENT)
8380    WHEN BTRY = SPAM.BTRY(SPAM)
8381    FIND THE FIRST CASE
8382    IF FOUND
8383       TRACE
8384 <------STOP
8385    OTHERWISE
8386
8387    IF ANALYSIS(1) = TRUE
8388       USE UNIT 42 FOR OUTPUT
8389       PRINT 1 LINE WITH BTRY, BY.STATUS(BTRY), TIME.V THUS
8390       BTRY *** CHANGES FROM STATUS * TO STATUS 1 AT **.******
8391       USE UNIT 6 FOR OUTPUT
8392    ALWAYS
8393    LET BY.STATUS(BTRY) = OCCUPATION
8394    LET UN.STATUS(BY.UNIT(BTRY)) = STA.TO.WITH
8395    ''MOVE THE REMAINDER OF THE DISTANCE NOW
8396    ADD MOVE.DIST/2 TO UN.X.COORD(BY.UNIT(BTRY))
8397    SCHEDULE_A ARTY.OCCUPATION------------------------------------------------------------------------------------------------------------>(355)
8398    GIVEN
8399    BTRY
8400    IN TB.OCCUPY(BY.TYPE(BTRY)) MINUTES
8401
8402 <--EXITEVENT
8403 ENDEVENT
```

```
8404 EVENT UPDATE.LOC                                                    ''                                                    E031
8405    GIVEN
8406    UNT.,
8407    XCOR.,
8408    YCOR.,
8409    RATE.MOVE.,
8410    TYP.MOV.,
8411    ORDR.
8412
8413    ADD 1 TO ANAL.CTR(163,1)  ''                                                                                  \DYN_ANAL
8414    ''THIS EVENT UPDATES THE LOCATION OF A MOVING UNIT.  SINCE
8415    ''  IT IS SCHEDULED AT THE COMPLETION OF A 500 METER LINK
8416    ''  OF THE UNITS PATH TO ITS DESTINATION THE NEW LOCATION WILL ALWAYS                                             \1
8417    ''  BE 500 METERS FROM THE PREVIOUS LOCATION UNLESS THE
8418    ''  UNIT'S DESTINATION IS LESS THAN THAT.  IF THE MOVE IS
8419    ''  AN AGGRESSIVE MOVE, PROXIMITY TO ENEMY UNITS IS CHECKED.
8420    ''  WHEN PROXIMITY EXISTS, A ROUTINE IS CALLED TO DETERMINE WHAT                                                 \1
8421    ''  THE UNIT SHOULD DO NEXT.  OTHERWISE THE ROUTINE
8422    ''  SCHEDULES THE NEXT UPDATE OF LOCATION FOR THE UNIT.
8423
8424    NORMALLY MODE IS INTEGER
8425    DEFINE EN.UNITS. AS AN INTEGER 1-DIMENSIONAL ARRAY
8426    DEFINE HYP. AS A REAL VARIABLE
8427    DEFINE RATIO. AS A REAL VARIABLE
8428    DEFINE TYP.MOV. AS A TEXT VARIABLE ''                                                                         \TEXT
8429
8430    IF TYP.MOV. = "REINF"
8431       LET XCOR.=UN.X.COORD(REINFORCED.UN(ORD.ID(ORDR.)))
8432       LET YCOR.=UN.Y.COORD(REINFORCED.UN(ORD.ID(ORDR.)))
8433    ALWAYS
8434
8435    LET XDIF.=XCOR.-UN.X.COORD(UNT.)
8436    LET YDIF.=YCOR.-UN.Y.COORD(UNT.)
8437    LET HYP.=SQRT.F(XDIF.**2+YDIF.**2)
8438    IF HYP. > 31.25          ''31.25 HEXADECAMETERS=500 METERS
8439       LET RATIO.=31.25/HYP.
8440       LET NEWX.=UN.X.COORD(UNT.)+RATIO.*XDIF.
8441       LET NEWY.=UN.Y.COORD(UNT.)+RATIO.*YDIF.
8442    ELSE
8443       LET NEWX.=XCOR.
8444       LET NEWY.=YCOR.
8445       LET CHANGE. = 1
8446    ALWAYS
8447
8448    LET UN.X.COORD(UNT.)=NEWX.
8449    LET UN.Y.COORD(UNT.)=NEWY.
8450
8451    IF DEBUG = TRUE
8452       SKIP 1 LINE
8453       PRINT 2 LINES WITH UNIT.NOS(UNT.),TIME.V,NEWX.,NEWY. THUS
8454       LOCATION OF UNIT ****** UPDATED AT **.*** HOURS
8455       NEW LOCATION IS ****,****
8456       SKIP 1 LINE
8457    ALWAYS
8458
8459    LOOP FOR EVERY MAN.UNIT IN MU.TF.LIST(UN.PTR(UNT.))
8460    DO
8461       LET UN.X.COORD(MU.UNIT.ID(MAN.UNIT))=NEWX.-MU.OFFSET.X(MAN.UNIT)
```

```
8462        LET UN.Y.COORD(MU.UNIT.ID(MAN.UNIT))=NEWY.-MU.OFFSET.Y(MAN.UNIT)
8463        CALL DEQ.FEBA.SET------------------------------------------------------------------------------------------>(325)
8464        GIVEN
8465        MU.UNIT.ID(MAN.UNIT)
8466        YIELDING
8467        OLD.SECTOR
8468        CALL ENQ.FEBA.SET------------------------------------------------------------------------------------------>(331)
8469        GIVEN
8470        MU.UNIT.ID(MAN.UNIT),
8471        OLD.SECTOR
8472        YIELDING
8473        UNT.SECTOR
8474     ENDLOOP
8475
8476     CALL DEQ.FEBA.SET------------------------------------------------------------------------------------------>(325)
8477     GIVEN
8478     UNT.
8479     YIELDING
8480     OLD.SECTOR
8481     CALL ENQ.FEBA.SET------------------------------------------------------------------------------------------>(331)
8482     GIVEN
8483     UNT.,
8484     OLD.SECTOR
8485     YIELDING
8486     UNT.SECTOR.
8487
8488     IF ORD.TYPE(ORDR.)="MOVCOR"AND TYP.MOV.="TACTIC"
8489        CALL CHECK.PROX------------------------------------------------------------------------------------------>(138)
8490        GIVEN
8491        UNT.,
8492        UNT.SECTOR.,
8493        ACT.BATTLE.RANGE
8494        YIELDING
8495        PROX.TEST.,
8496        EN.UNITS.(*),
8497        BATTLE.NUM
8498        IF PROX.TEST.=YES
8499           ''EMPTY LIST OF BARRIER MINES
8500           LOOP FOR EVERY .MO OF MO.LIST(UNT.)
8501           DO
8502              REMOVE THE .MO FROM THE MO.LIST(UNT.)
8503              DESTROY THE MINE.OBSTACLE CALLED .MO
8504           ENDLOOP
8505           IF BATTLE.NUM > 0
8506              CALL INTER.BATTLE------------------------------------------------------------------------------------>(146)
8507              GIVEN
8508              UNT.,
8509              BATTLE.NUM,
8510              ORDR.
8511              RELEASE EN.UNITS.(*)
8512           ELSE
8513              CALL PROX.POS------------------------------------------------------------------------------------->(117)
8514              GIVEN
8515              UNT.,
8516              ORDR.,
8517              EN.UNITS(*))
8518           ALWAYS
8519 <--------EXITEVENT
```

```
8520        OTHERWISE
8521        RELEASE EN.UNITS.(*)
8522     ALWAYS
8523
8524     ''CHECK FOR A MINEFIELD IN THE NEXT 500 METERS
8525     LOOP FOR EVERY .MO IN THE MO.LIST(UNT.)
8526     DO   ''                                                                    \1
8527        LOOP FOR EVERY MINEFIELD CALLED .MF
8528        WITH MF.COLOR(.MF) NE UN.COLOR(UNT.)
8529        DO   ''                                                                 \1
8530           IF MO.MINEFIELD(.MO) = .MF
8531              IF AU.LIST(.MF) IS NOT EMPTY
8532                 LET .MINE.FLAG = 1
8533              ALWAYS
8534           ALWAYS
8535        ENDLOOP
8536        LET .DELTAX = NEWX. - MO.X.INTER(.MO)
8537        LET .DELTAY = NEWY. - MO.Y.INTER(.MO)
8538        LET .DIS = SQRT.F(.DELTAX**2 + .DELTAY**2)  ''                          \OPTIMIZE
8539        IF .DIS LE 31.25  ''HDM
8540           ''A MEMBER OF THE TASK FORCE WILL HIT THE MINEFIELD.
8541           ''CHOOSE WHICH ONE, ASSESS DAMAGES AND DELAYS
8542           IF N.MU.TF.LIST(UN.PTR(UNT.)) = 0
8543              LET .TGT = UNT.
8544           ELSE
8545              LET .TGT = RANDI.F(1, N.MU.TF.LIST(UN.PTR(UNT.)),
8546              RN.SEED)
8547              LET .COUNTER = 1
8548              FOR EVERY .MU IN MU.TF.LIST(UN.PTR(UNT.))
8549              UNTIL .COUNTER = .TGT
8550              ADD 1 TO .COUNTER
8551              LET .TGT = MU.UNIT.ID(.MU)
8552           ALWAYS
8553           CALL MINE.EFFECTS----------------------------------------------------------------------------------------------------->(250)
8554           GIVEN
8555           .TGT,
8556           MO.X.INTER(.MO),
8557           MO.Y.INTER(.MO),
8558           1,     ''BARRIER MINE
8559           0,
8560           .MINE.FLAG
8561           YIELDING
8562           .DELAY
8563           ADD .DELAY TO .TOTAL.DELAY
8564           CREATE AN AWARE.UNIT
8565           LET AU.UNIT.ID(AWARE.UNIT) = UNT.
8566           FILE THIS AWARE.UNIT IN THE AU.LIST(MO.MINEFIELD(.MO))
8567           REMOVE .MO FROM THE MO.LIST(UNT.)
8568           DESTROY THE MINE.OBSTACLE CALLED .MO
8569        ALWAYS
8570     ENDLOOP
8571
8572     IF MF.DEBUG = TRUE AND .TOTAL.DELAY GT 0
8573        PRINT 2 LINES WITH UNT., XCOR., YCOR.,
8574        NEWX., NEWY., .TOTAL.DELAY, .TGT THUS
8575        ===UPDATE.LOC  TF LDR=**** (*****,*****)TO(*****,*****)
8576        ============DEL=************ MIN, TGT=****
8577     ALWAYS
```

```
8578
8579     IF CHANGE.=1
8580        CALL END.MOVE------------------------------------------------------------------------------------------>( 87)
8581        GIVEN
8582        UNT.,
8583        ORDR.
8584     ELSE
8585        SCHEDULE_AN UPDATE.LOC---------------------------------------------------------------------------------->(408)
8586        GIVEN
8587        UNT.,
8588        XCOR.,
8589        YCOR.,
8590        RATE.MOVE.,
8591        TYP.MOV.,
8592        ORDR.
8593        IN (LOC.UPDATE.FREQ/RATE.MOVE.+RANDOM.F(RN.SEED)
8594        + .TOTAL.DELAY) MINUTES
8595     ALWAYS
8596
8597 <--EXITEVENT
8598 ENDEVENT
```

```
8599 EVENT ACT ATK                                                                    E032
8600    GIVEN
8601    UNT.,
8602    ORDR,
8603    EN.UNITS.
8604
8605
8606    ADD 1 TO ANAL.CTR(164,1)  ''                                          \DYN_ANAL
8607    ''THIS ROUTINE IS EXECUTED WHEN ENEMY MANEUVER UNITS COME INTO
8608    ''    PROXIMITY AND THE ORDERS OF THE AGGRESSOR CALL FOR AN ATTACK.
8609    ''    A ROUTINE IS CALLED TO DETERMINE ALL OF THE UNITS TO BE
8610    ''    INVOLVED IN THE BATTLE, AND A START.BATTLE EVENT IS SCHEDULED
8611    ''    IMMEDIATELY.
8612
8613    NORMALLY MODE IS INTEGER
8614    DEFINE MAN.UNIT AS A VARIABLE
8615    DEFINE EN.UNITS. AS AN INTEGER 1-DIMENSIONAL ARRAY
8616    DEFINE RD.UNITS. AS AN INTEGER 1-DIMENSIONAL ARRAY
8617    DEFINE BL.UNITS. AS AN INTEGER 1-DIMENSIONAL ARRAY
8618    DEFINE DUM.UNITS. AS AN INTEGER 1-DIMENSIONAL ARRAY
8619
8620    RESERVE DUM.UNITS. AS 100
8621
8622    ''THE ABOVE ARRAY SIZE WAS CHANGED FROM 15 TO 30 TO AVOID
8623    ''ERROR IN EXECUTION WHEN LARGE FORCES WERE DRAWN INTO A
8624    ''BATTLE - %31MAR80_%RGR - IT IS A POTENTIAL PROBLEM IN
8625    ''THE USE OF COSAGE WITH LARGE FORCES THAT THE USER MUST BE
8626    ''AWARE OF. THE LARGEST KNOWN BATTLE FORCE SIMULATED WAS 16 BLUE UNITS
8627    ''AND 4 RED UNITS (IN THE SAME BATTLE) .
8628
8629    IF MU.TF.MEM(UN.PTR(UNT.))>0
8630       LET UNT. = MU.TF.MEM(UN.PTR(UNT.))
8631    ALWAYS
8632
8633    IF UN.COLOR(UNT.) = RED
8634       LET DUM.UNITS.(1) = UNT.
8635       LET I = 1
8636       LOOP FOR EVERY MAN.UNIT IN MU.TF.LIST(UN.PTR(UNT.))
8637       DO
8638          ADD 1 TO I
8639          LET DUM.UNITS.(I) = MU.UNIT.ID(MAN.UNIT)
8640       ENDLOOP
8641       IF I > 100
8642          TRACE
8643 <--------STOP
8644       OTHERWISE
8645
8646       RESERVE RD.UNITS. AS I
8647
8648       LOOP FOR J = 1 TO I
8649       DO
8650          LET RD.UNITS.(J) = DUM.UNITS.(J)
8651       ENDLOOP
8652
8653       CALL PREPARE.LIST-------------------------------------------------------------------------------------->(112)
8654       GIVEN
8655       EN.UNITS.(*)
8656       YIELDING
```

```
8657        BL.UNITS.(*)
8658     ELSE
8659        LET DUM.UNITS.(1) = UNT.
8660        LET I = 1
8661        LOOP FOR EVERY MAN.UNIT IN MU.TF.LIST(UN.PTR(UNT.))
8662        DO
8663           ADD 1 TO I
8664           LET DUM.UNITS.(I) = MU.UNIT.ID(MAN.UNIT)
8665        ENDLOOP
8666        IF I > 100
8667           TRACE
8668 <----------STOP
8669        OTHERWISE
8670
8671        RESERVE BL.UNITS. AS I
8672
8673        LOOP FOR J = 1 TO I
8674        DO
8675           LET BL.UNITS.(J) = DUM.UNITS.(J)
8676        ENDLOOP
8677
8678        CALL PREPARE.LIST------------------------------------------------------------------------------------------------------------------------>(112)
8679        GIVEN
8680        EN.UNITS.(*)
8681        YIELDING
8682        RD.UNITS.(*)
8683     ALWAYS
8684
8685     LET .NO.UNITS = DIM.F(BL.UNITS(*))
8686     LOOP FOR .I = 1 TO .NO.UNITS
8687     DO
8688        LET UN.DELAY(BL.UNITS(.I)) = 0
8689        LET UN.FASCAM.RECVD(BL.UNITS(.I)) = 0
8690     ENDLOOP
8691     LET .NO.UNITS = DIM.F(RD.UNITS(*))
8692     LOOP FOR .I = 1 TO .NO.UNITS
8693     DO
8694        LET UN.DELAY(RD.UNITS(.I)) = 0
8695        LET UN.FASCAM.RECVD(RD.UNITS(.I)) = 0
8696     ENDLOOP
8697
8698     RELEASE EN.UNITS.(*), DUM.UNITS.(*)
8699
8700     CALL GET.TERRAIN-------------------------------------------------------------------------------------------------------------------------->(336)
8701     YIELDING
8702     TER.TYPE.
8703     SCHEDULE_A START.BATTLE------------------------------------------------------------------------------------------------------------------->(395)
8704     GIVEN
8705     TER.TYPE.,
8706     UN.MISSION(BL.UNITS.(1)),
8707     UN.MISSION(RD.UNITS.(1)),
8708     BL.UNITS.(*),
8709     RD.UNITS.(*),                                                                                                                                \1
8710     0 NOW  ''
8711
8712 <--EXITEVENT
8713 ENDEVENT
```

```
8714 EVENT ACT.DEF                                                    ''          E033
8715    GIVEN
8716    UNT.,
8717    ORDR.
8718
8719    ADD 1 TO ANAL.CTR(165,1)  ''                                          \DYN_ANAL
8720    ''THIS ROUTINE SETS THE MISSION AND STATUS OF A UNIT WHEN IT BEGINS
8721    ''     TO EXECUTE A DEFEND ORDER.
8722
8723    NORMALLY MODE IS INTEGER
8724    DEFINE MAN.UNIT AS A VARIABLE
8725
8726    LET UN.MISSION(UNT.) = ORD.MISSION(ORD.ID(ORDR.))
8727    LET UN.STATUS(UNT.) = STATIONARY
8728    LOOP FOR EVERY MAN.UNIT IN MU.TF.LIST(UN.PTR(UNT.))
8729    DO
8730       LET UN.MISSION(MU.UNIT.ID(MAN.UNIT)) = UN.MISSION(UNT.)
8731       LET UN.STATUS(MU.UNIT.ID(MAN.UNIT)) = STATIONARY
8732    ENDLOOP
8733
8734 <--EXITEVENT
8735 ENDEVENT
```

```
8736 EVENT ACT.MOVCOR                                                                                    ''                                                                                
```

```
8766 EVENT ACT.MOVDIS                                                                                                    E035
8767    GIVEN
8768    UNT.,
8769    ORDR.
8770
8771    ADD 1 TO ANAL.CTR(167,1)  ''                                                                                     \DYN_ANAL
8772    ''THIS ROUTINE IS EXECUTED AT THE BEGINNING OF A MOVDIS ORDER.  IT
8773    ''    DETERMINES THE COORDINATES OF THE UNIT'S DESTINATION AND
8774    ''    SCHEDULES A START.MOVE IMMEDIATELY.
8775
8776    NORMALLY MODE IS INTEGER
8777
8778    IF UN.COLOR(UNT.)=BLUE
8779       IF DIR.OF.MOVE(ORD.ID(ORDR.))="ADVANC"
8780          LET DIR. = 1
8781       ELSE
8782          LET DIR. = -1
8783       ALWAYS
8784    ELSE
8785       IF DIR.OF.MOVE(ORD.ID(ORDR.))="ADVANC"
8786          LET DIR. = -1
8787       ELSE
8788          LET DIR. = 1
8789       ALWAYS
8790    ALWAYS
8791    LET YCOR.=UN.Y.COORD(UNT.)
8792    LET XCOR.=UN.X.COORD(UNT.)+DIR.*DIST.MOVED(ORD.ID(ORDR.))/16
8793    SCHEDULE_A START.MOVE------------------------------------------------------------------------------------------>(403)
8794    GIVEN
8795    UNT.,
8796    XCOR.,
8797    YCOR.,
8798    MOVE.TYPE(ORD.ID(ORDR.)),
8799    ORDR. NOW  ''                                                                                                    \1
8800
8801 <--EXITEVENT
8802 ENDEVENT
```

```
8803 EVENT DYNAMIC.ANALYSIS.REPORT SAVING THE EVENT NOTICE  ''                                                    E036
8804
8805    DEFINE I, J, CELLS, SAVE.OUTPUT.UNIT AS AN INTEGER VARIABLES ''                                          \DYN_ANAL
8806    DEFINE TIME.CPU AS AN INTEGER VARIABLE
8807    DEFINE REAL.TIME.CPU AS A REAL VARIABLE
8808
8809    LET SAVE.OUTPUT.UNIT = WRITE.V
8810    USE UNIT 80 FOR OUTPUT
8811    ''START NEW PAGE
8812    LET LINES.V = INF.C ''TO SUPPRESS FORM-FEEDS
8813
8814    WRITE TIME.V AS "TIME.V= ", D(8,5), /
8815
8816    CALL LIB$STAT_TIMER(2, TIME.CPU)------------------------------------------------------------------------>(650)
8817    LET REAL.TIME.CPU = TIME.CPU / 100
8818
8819    WRITE AS "CPU.TIME= "
8820    WRITE REAL.TIME.CPU AS D(10, 4)
8821    WRITE AS /
8822    FOR I = 1 TO 264
8823    DO
8824       WRITE ANAL.TEXT(I) AS B 2, T *, B 45
8825       LET CELLS = DIM.F(ANAL.CTR(I, *))   ''NUMBER OF ROWS
8826       FOR J = 1 TO CELLS
8827       WRITE ANAL.CTR(I,J) AS I 8, S 2
8828       WRITE AS /
8829       FOR J = 1 TO CELLS
8830       LET ANAL.CTR(I,J) = 0
8831    ENDLOOP
8832
8833    RESCHEDULE_THE DYNAMIC.ANALYSIS.REPORT IN 1 HOURS-------------------------------------------------------->(417)
8834
8835
8836    USE SAVE.OUTPUT.UNIT FOR OUTPUT
8837
8838 <--RETURN
8839
8840 END
```

PROCESSES

```
8847 PROCESS AC.ATK.TGT                                                    ''                                                    P001
8848
8849    ADD 1 TO ANAL.CTR(168,1)  ''                                                                                             \DYN_ANAL
8850    NORMALLY MODE IS INTEGER
8851    DEFINE .AC.ATK.TGT          TO MEAN PROCESS.V ''                                                                         CHG\01
8852    DEFINE .CAS.MISSION         TO MEAN AAT.CMSN(.AC.ATK.TGT)
8853    DEFINE .AD.UNIT             TO MEAN AAT.AD.UNIT(.AC.ATK.TGT)
8854    DEFINE .TGT.RANGE           TO MEAN AAT.RANGE(.AC.ATK.TGT)
8855    DEFINE .VIS.IND             TO MEAN AAT.VIS.IND(.AC.ATK.TGT)
8856    DEFINE .KILLED.IND          TO MEAN AAT.AC.KILLED.IND(.AC.ATK.TGT)
8857    DEFINE .AC.TYPE             TO MEAN CMSN.AC.TYPE(.CAS.MISSION)
8858    DEFINE .TGT                 TO MEAN CMSN.TGT.UNIT(.CAS.MISSION)
8859
8860    DEFINE .AC.DET.TIME  TO MEAN EAAT.AC.DET.TIME(AAT.EX.AATGT(.AC.ATK.TGT))
8861    DEFINE .BLIND        TO MEAN EAAT.BLIND(AAT.EX.AATGT(.AC.ATK.TGT))
8862    DEFINE .DETECT.TIME  TO MEAN EAAT.DETECT.TIME(AAT.EX.AATGT(.AC.ATK.TGT))
8863    DEFINE .DIST.TO.FP   TO MEAN EAAT.DIST.TO.FP(AAT.EX.AATGT(.AC.ATK.TGT))
8864    DEFINE .DIST.TO.P3   TO MEAN EAAT.DIST.TO.P3(AAT.EX.AATGT(.AC.ATK.TGT))
8865    DEFINE .FIRING.RANGE TO MEAN EAAT.FIRING.RANGE(AAT.EX.AATGT(.AC.ATK.TGT))
8866    DEFINE .FIRING.TIME  TO MEAN EAAT.FIRING.TIME(AAT.EX.AATGT(.AC.ATK.TGT))
8867    DEFINE .MODEL.ADS    TO MEAN EAAT.MODEL.ADS(AAT.EX.AATGT(.AC.ATK.TGT))
8868    DEFINE .NUM.FIRES.PER.PASS  TO MEAN EAAT.NUM.FIRES.P(AAT.EX.AATGT(.AC.ATK.TGT))
8869    DEFINE .NUM.PASSES   TO MEAN EAAT.NUM.PASSES(AAT.EX.AATGT(.AC.ATK.TGT))
8870    DEFINE  .P1.TO.P2    TO MEAN EAAT.P1.TO.P2(AAT.EX.AATGT(.AC.ATK.TGT))
8871    DEFINE  .P2.TO.P3    TO MEAN EAAT.P2.TO.P3(AAT.EX.AATGT(.AC.ATK.TGT))
8872    DEFINE  .P3.TO.P1    TO MEAN EAAT.P3.TO.P1(AAT.EX.AATGT(.AC.ATK.TGT))
8873    DEFINE .RANGE        TO MEAN EAAT.RANGE(AAT.EX.AATGT(.AC.ATK.TGT))
8874    DEFINE .RANGE.AT.FP  TO MEAN EAAT.RANGE.AT.FP(AAT.EX.AATGT(.AC.ATK.TGT))
8875    DEFINE .RESULT        TO MEAN EAAT.RESULT(AAT.EX.AATGT(.AC.ATK.TGT))
8876    DEFINE .ROW           TO MEAN EAAT.ROW(AAT.EX.AATGT(.AC.ATK.TGT))
8877    DEFINE .SUM.CRIT.EQ  TO MEAN EAAT.SUM.CRIT.EQ(AAT.EX.AATGT(.AC.ATK.TGT))
8878    DEFINE .TIME.TO.LEAVE TO MEAN EAAT.TIME.TO.LEA(AAT.EX.AATGT(.AC.ATK.TGT))
8879    DEFINE .WAIT.TIME    TO MEAN EAAT.WAIT.TIME(AAT.EX.AATGT(.AC.ATK.TGT))
8880
8881    IF TACAIR.DEBUG = 1
8882       PRINT 1 LINE WITH .AC.ATK.TGT, TIME.V THUS
8883       = = = AC.ATK.TGT ******   AT ***.***
8884       LIST ATTRIBUTES OF AC.ATK.TGT CALLED .AC.ATK.TGT
8885       LIST CMSN.SEQ.NR(.CAS.MISSION), CMSN.SIDE(.CAS.MISSION),
8886       .CAS.MISSION
8887       LOOP FOR EACH .LINK IN UN.EQUIP.LIST(.TGT)
8888       DO
8889          LIST ATTRIBUTES OF UE.LINK CALLED .LINK
8890          FOR EACH .SO IN SO.LIST(.LINK)
8891          LIST ATTRIBUTES OF SHOOT.OUT CALLED .SO
8892          FOR EACH .FT IN UE.TARGET.LIST(.LINK)
8893          LIST ATTRIBUTES OF FIRING.TABLE CALLED .FT
8894       ENDLOOP
8895    ALWAYS
8896
8897    CREATE  EX.AC.ATK.TGT CALLED AAT.EX.AATGT(.AC.ATK.TGT)
8898
8899    ''WHEN .AD.UNIT IS NOT ZERO THE AIRCRAFT WILL BE                                                                         \1
8900    ''FIRED ON BY .AD.UNIT.  FIND THE SENSOR IT WILL USE
8901    ''TO DETECT THE AIRCRAFT.
8902    IF .AD.UNIT IS NOT ZERO
8903       FOR EACH .LINK OF UN.SENSOR.LIST(.AD.UNIT)
8904       WITH ST.NAME(US.SENSOR.TYPE(.LINK)) = "AD"
```

```
8905        FIND THE FIRST CASE
8906        IF NONE
8907           TRACE
8908 <--------STOP
8909        OTHERWISE
8910        LET .MODEL.ADS = US.MODEL(.LINK)
8911     ALWAYS
8912
8913     LET .P1.TO.P2 = ACT.P1.DIST(.AC.TYPE) - ACT.P2.DIST(.AC.TYPE)
8914     LET .P2.TO.P3 = SQRT.F((REAL.F((ACT.X3(.AC.TYPE) - ACT.X2(.AC.TYPE))**2
8915     + (ACT.Y3(.AC.TYPE) - ACT.Y2(.AC.TYPE))**2
8916     + (ACT.Z3(.AC.TYPE) - ACT.Z2(.AC.TYPE))**2)))
8917     LET .P3.TO.P1 = SQRT.F((REAL.F((ACT.X3(.AC.TYPE) - ACT.X1(.AC.TYPE))**2
8918     + (ACT.Y3(.AC.TYPE) - ACT.Y1(.AC.TYPE))**2
8919     + (ACT.Z3(.AC.TYPE) - ACT.Z1(.AC.TYPE))**2)))
8920
8921     ''SET UP THE LINKS USED BY THE AIRCRAFT TO SELECT
8922     ''DIRECT FIRE TARGETS.  PERSONNEL ARE EXCLUDED SINCE
8923     ''THEY CAN ONLY BE KILLED BY BOMBS.  THESE ARE
8924     ''USED IN A MANNER SIMILAR TO FIRING.TABLES.
8925     LOOP FOR EVERY .LINK OF UN.EQUIP.LIST(.TGT)
8926     WITH UE.ID(.LINK) NE 1 AND
8927     UE.ID(.LINK) NE N.BLUE.TYPE.EQP + 1
8928     DO
8929        CREATE AN AAT.TGT
8930        FILE AAT.TGT IN THE AATT.LIST(.AC.ATK.TGT)
8931        LET AATT.UE.LINK(AAT.TGT) = .LINK
8932     ENDLOOP
8933
8934     ''FIND THE MINIMUM TW.AC.DET.TIME IN THE TARGET UNIT.
8935     ''CONSIDER ONLY POSITIVE VALUES.
8936     LET .AC.DET.TIME = INF.C
8937     FOR EACH .LINK OF UN.EQUIP.LIST(.TGT)
8938     FOR EACH .WEAPON OF UE.WEAPON.SET(.LINK)
8939     WITH 0 LT TW.AC.DET.TIME(WPN.ID(.WEAPON)) LT .AC.DET.TIME
8940     LET .AC.DET.TIME = TW.AC.DET.TIME(WPN.ID(.WEAPON))
8941     IF .AC.DET.TIME = INF.C
8942        LET .AC.DET.TIME = 10
8943     ALWAYS
8944
8945     ''FOR EACH WEAPON ON THE AIRCRAFT, COMPUTE THE
8946     ''NUMBER OF PASSES REQUIRED TO EXPEND ALL OF ITS
8947     ''ROUNDS.  MAKE THAT NUMBER OF PASSES. ASSESS
8948     ''LOSSES, THEN GO ON TH THE NEXT WEAPON.
8949     FOR EACH .AC.LINK OF UN.EQUIP.LIST(SD.AIRFIELD(CMSN.SIDE(.CAS.MISSION)))
8950     WITH UE.ID(.AC.LINK) = ACT.EQUIP.ID(.AC.TYPE)
8951     FIND THE FIRST CASE
8952     IF NONE
8953        TRACE
8954 <------STOP
8955     OTHERWISE
8956
8957     ''WHEN THE TARGET IS NOT IN A BATTLE THIS IS A                    \1
8958     ''BAI (BATTLE AIR INTERDICTION) MISSION.  A NUMBER
8959     ''OF BLIND PASSES WILL HAVE TO BE MADE UNTIL THE
8960     ''TARGET IS FOUND.  WHEN THE TARGET IS IN A BATTLE,
8961     ''THE GROUND CONTROLLER DIRECTS THE AIRCRAFT TO THE TARGET.
8962     IF UN.BATTLE.INDEX(.TGT) = 0
```

```
8963        LET .BLIND = 1
8964        + TRUNC.F(ACT.BAI.TA.DELAY(.AC.TYPE) / ACT.PASS.TIME(.AC.TYPE))
8965     ELSE
8966        LET .BLIND = 0
8967     ALWAYS
8968
8969     ''THE AIRCRAFT WILL RETURN TO BASE IF THE TARGET UNIT
8970     ''HAS LESS CRITICAL EQUIPMENT THAN THE BREAKPOINT PERCENT.
8971     LET .SUM.CRIT.EQ = TU.CRIT.NO(UN.TYPE.UNIT(.TGT))
8972
8973     ''THE AIRCRAFT WILL RETURN IF IT DOES NOT HAVE ENOUGH
8974     ''TIME TO MAKE ANOTHER PASS AND RETURN.
8975     LET .TIME.TO.LEAVE = CMSN.TAKE.OFF.TIME(CAS.MISSION)
8976     + (ACT.MAX.ALOFT(.AC.TYPE)/60)
8977     - (CMSN.FLIGHT.TIME(.CAS.MISSION) + ACT.PASS.TIME(.AC.TYPE)/60)
8978
8979     LOOP FOR EVERY .WPN OF UE.WEAPON.SET(.AC.LINK)
8980     UNTIL .KILLED.IND = YES OR
8981     .TIME.TO.LEAVE LT TIME.V OR
8982     .SUM.CRIT.EQ LE TU.CRIT.NO(UN.TYPE.UNIT(.TGT)) *
8983     SD.CAS.BRKPT(CMSN.SIDE(.CAS.MISSION)) / 100
8984     DO
8985        ADD 1 TO .ROW
8986        IF .ROW GT 1
8987           LET .BLIND = 0
8988        ALWAYS
8989        IF WPN.AC.MUNS(.WPN) EQ 0
8990           LET .NUM.FIRES.PER.PASS=TRUNC.F(.P1.TO.P2/(ACT.SPEED(
8991           .AC.TYPE)*60./(1.6 * 7))) ''CONVERTS ACT.SPEED TO DM PER MIN
8992           IF .NUM.FIRES.PER.PASS=0
8993              LET .NUM.FIRES.PER.PASS=1
8994           ALWAYS
8995           IF WPN.QUANTITY(.WPN) = 0
8996              LET .NUM.PASSES = .BLIND
8997           ELSE
8998              LET .NUM.PASSES = .BLIND + TRUNC.F(TW.BASIC.LOAD(WPN.ID(.WPN))*
8999              10/(TW.RATE.OF.FIRE(WPN.ID(.WPN))*WPN.QUANTITY(.WPN)*
9000              .NUM.FIRES.PER.PASS))
9001           ALWAYS
9002           IF TACAIR.DEBUG = 1
9003              PRINT 2 LINES WITH EQ.NAME(ACT.EQUIP.ID(.AC.TYPE)),.TGT,
9004              .NUM.FIRES.PER.PASS,.NUM.PASSES THUS
9005              ////////////FOR AIRCRAFT **** ATTACKING UNIT ****////////////////////
9006              [[[[[[[[[[THE NUMBER OF FIRES PER PASS IS *** FOR *** PASSES[[[[[
9007           ALWAYS
9008        ELSE
9009           LET .NUM.PASSES=.BLIND+TRUNC.F(WPN.QUANTITY(.WPN)*10
9010           /TW.RATE.OF.FIRE(WPN.ID(.WPN)))
9011           IF TACAIR.DEBUG=1
9012              PRINT 1 LINE WITH EQ.NAME(ACT.EQUIP.ID(.AC.TYPE)),.TGT,
9013              .NUM.PASSES THUS
9014              [[[[[[[[[[FOR AIRCRAFT **** ATTACKING UNIT **** PASSES ARE ***
9015           ALWAYS
9016        ALWAYS
9017        LOOP FOR .PASS = 1 TO .NUM.PASSES
9018        DO
9019           IF TACAIR.DEBUG =1
9020              LIST .WPN, .KILLED.IND, .TIME.TO.LEAVE,
```

```
9021            .SUM.CRIT.EQ, .PASS, TIME.V, .WAIT.TIME,
9022            .BLIND, .NUM.PASSES, .RESULT, .RANGE,
9023            .FIRING.TIME, TW.NAME(WPN.ID(.WPN))
9024         ALWAYS
9025         LET .FIRING.RANGE = UNIFORM.F(
9026         REAL.F(TW.MIN.RANGE(WPN.ID(.WPN))),
9027         REAL.F(TW.MAX.RANGE(WPN.ID(.WPN))), RN.SEED)
9028         LET .FIRING.RANGE = MAX.F(.FIRING.RANGE, ACT.P2.DIST(.AC.TYPE))
9029         LET .FIRING.RANGE = MIN.F(.FIRING.RANGE, ACT.P1.DIST(.AC.TYPE))
9030         LET .DIST.TO.FP = ACT.P1.DIST(.AC.TYPE) - .FIRING.RANGE
9031         LET .WAIT.TIME = .DIST.TO.FP/(ACT.SPEED(.AC.TYPE)*3600) ''HRS
9032         IF .AD.UNIT IS NOT ZERO
9033            ''THE AREA AIR DEFENSE UNIT FIRES ON THE AIRCRAFT
9034            ''UNTIL IT REACHES THE FIRING POINT.  IF THE AIRCRAFT
9035            ''IS HIT THE TARGET IS NOT FIRED ON.  AVERAGE THE
9036            ''RANGE FROM P1 TO THE AD UNIT AND FROM THE FIRING POINT
9037            ''TO THE AD UNIT.  BY THE LAW OF COSINES:
9038            LET .RANGE.AT.FP = SQRT.F((REAL.F(CMSN.P1.ADU.RANGE(
9039            .CAS.MISSION)**2 + .DIST.TO.FP**2
9040            - 2 * CMSN.P1.ADU.RANGE(.CAS.MISSION) * .DIST.TO.FP
9041            * COS.F(CMSN.ANGLE(.CAS.MISSION)))))
9042            LET .RANGE = (CMSN.P1.ADU.RANGE(.CAS.MISSION) +
9043            .RANGE.AT.FP) / 2
9044            LET .FIRING.TIME = MADS.DELAY.TIME(.MODEL.ADS) / 3600.
9045            LET .RESULT = 0
9046            LOOP UNTIL .RESULT = HIT OR .RESULT = NO.PK.PTR OR
9047            .RESULT = NO.SHOT.MADE OR .FIRING.TIME GT .WAIT.TIME
9048            DO
9049               CALL AD.SHOOT----------------------------------------------------------------------------------------------------->(307)
9050               GIVEN
9051               .RANGE,
9052               .FIRING.TIME,
9053               .AD.UNIT,
9054               .AC.ATK.TGT,
9055               "AC.ATK"
9056               YIELDING
9057               .RESULT,
9058               .FIRING.TIME
9059               IF .RESULT = HIT
9060                  LET .KILLED.IND  = YES
9061                  LET .WAIT.TIME = .WAIT.TIME/2
9062               ALWAYS
9063            ENDLOOP
9064            IF .FIRING.TIME GT .WAIT.TIME
9065               SUBTRACT .WAIT.TIME FROM .FIRING.TIME
9066            ELSE
9067               LET .FIRING.TIME = 0.
9068            ALWAYS
9069         ALWAYS
9070
9071         ''ALLOW THE TARGET UNIT TO FIRE. IT WILL
9072         ''USE A FIRING RANGE THAT IS THE AVERAGE OF
9073         ''P1 TO TARGET AND FIRING POINT TO TARGET.
9074         ''UTILIZE EXPONENTIAL.F FUNCTION IN NEXT DRAW ---------------------------------------------------------------- \1>(640)
9075         LET .DETECT.TIME = EXPONENTIAL.F(
9076         REAL.F(.AC.DET.TIME),RN.SEED)/3600
9077         LET .TGT.RANGE = (ACT.P1.DIST(.AC.TYPE) + .FIRING.RANGE)/2
9078         IF .DETECT.TIME LE .WAIT.TIME AND .RESULT NE HIT
```

```
9079
9080 <----------WAIT .DETECT.TIME HOURS
9081
9082            IF TACAIR.DEBUG = 1
9083               PRINT 1 LINE WITH .AC.ATK.TGT, TIME.V THUS
9084               = = = = = AC.ATK.TGT ****** DETECTED AT ***.****
9085               LIST CMSN.SEQ.NR(.CAS.MISSION), CMSN.SIDE(.CAS.MISSION)
9086               '' LOOP FOR EACH .LINK IN UN.EQUIP.LIST(.TGT)
9087               ''      DO
9088                  ''      LIST ATTRIBUTES OF UE.LINK CALLED .LINK
9089                  ''      FOR EACH .SO IN SO.LIST(.LINK)
9090                  ''           LIST ATTRIBUTES OF SHOOT.OUT CALLED .SO
9091                  ''      FOR EACH .FT IN UE.TARGET.LIST(.LINK)
9092                  ''           LIST ATTRIBUTES OF FIRING.TABLE CALLED .FT
9093               '' ENDLOOP
9094            ALWAYS
9095            IF .KILLED.IND = YES
9096       <--------LEAVE
9097            OTHERWISE
9098
9099            SUBTRACT .DETECT.TIME FROM .WAIT.TIME
9100            LET .VIS.IND = YES
9101            LET .DETECT.TIME = 0.
9102            LOOP FOR EVERY .LINK IN UN.EQUIP.LIST(.TGT)
9103            FOR EVERY .SO IN SO.LIST(.LINK)
9104            WITH SO.FIRING.TABLE(.SO) = 0 AND
9105            SO.AIR.ATK.INDIC(.SO) = YES
9106            DO
9107               INTERRUPT SHOOT.OUT CALLED .SO------------------------------------------------------------------------------------>(493)
9108               REACTIVATE_THE SHOOT.OUT CALLED .SO NOW---------------------------------------------------------------------------->(493)
9109            ENDLOOP
9110         ALWAYS
9111
9112 <----------WAIT .WAIT.TIME HOURS
9113
9114         IF TACAIR.DEBUG = 1
9115            PRINT 1 LINE WITH .AC.ATK.TGT, TIME.V THUS
9116            = = = = = AC.ATK.TGT ****** AT FIRING POINT AT ***.****
9117            LIST CMSN.SEQ.NR(.CAS.MISSION), CMSN.SIDE(.CAS.MISSION)
9118            ''LOOP FOR EACH .LINK IN UN.EQUIP.LIST(.TGT)
9119            ''      DO
9120               ''      LIST ATTRIBUTES OF UE.LINK CALLED .LINK
9121               ''      FOR EACH .SO IN SO.LIST(.LINK)
9122               ''           LIST ATTRIBUTES OF SHOOT.OUT CALLED .SO
9123               ''      FOR EACH .FT IN UE.TARGET.LIST(.LINK)
9124               ''           LIST ATTRIBUTES OF FIRING.TABLE CALLED .FT
9125            ''ENDLOOP
9126         ALWAYS
9127         IF .KILLED.IND = YES
9128       <------LEAVE
9129         OTHERWISE
9130
9131         SUBTRACT .WAIT.TIME FROM .DETECT.TIME
9132         IF .BLIND = 0
9133            IF WPN.AC.MUNS(.WPN) = 0
9134               CALL AC.DF.EFFECTS------------------------------------------------------------------------------------------------>(280)
9135               GIVEN
9136               .AC.ATK.TGT,
```

```
9137                .WPN,
9138                .ROW,
9139                .NUM.FIRES.PER.PASS
9140             ELSE
9141                CALL AC.BOMB.EFFECTS------------------------------------------------------------------------------------------>(276)
9142                GIVEN
9143                .AC.ATK.TGT,
9144                .WPN,
9145                .ROW
9146             ALWAYS
9147             LET .SUM.CRIT.EQ = 0
9148             FOR EACH .LINK OF UN.EQUIP.LIST(.TGT)
9149             WITH UE.CRITICAL.EQUIP.INDIC(.LINK) = YES
9150             ADD UE.QUANT(.LINK) TO .SUM.CRIT.EQ
9151          ALWAYS
9152
9153          IF .BLIND GT 0
9154             SUBTRACT 1 FROM .BLIND
9155          ALWAYS
9156          ''CONTINUE ON TO P3
9157          LET .DIST.TO.P3 = (.P1.TO.P2 - .DIST.TO.FP) + .P2.TO.P3
9158          LET .WAIT.TIME = .DIST.TO.P3 / (ACT.SPEED(.AC.TYPE)*3600) ''HRS
9159          LET .TGT.RANGE = (.TGT.RANGE + ACT.P3.DIST(.AC.TYPE))/2
9160          IF .AD.UNIT GT 0
9161             LET .RANGE = (CMSN.P3.ADU.RANGE(.CAS.MISSION)
9162             + .RANGE.AT.FP) / 2
9163             LOOP UNTIL .RESULT = HIT OR
9164             .RESULT = NO.SHOT.MADE OR .RESULT = NO.PK.PTR OR
9165             .FIRING.TIME GT .WAIT.TIME
9166             DO
9167                CALL AD.SHOOT------------------------------------------------------------------------------------------------>(307)
9168                GIVEN
9169                .RANGE,
9170                .FIRING.TIME,
9171                .AD.UNIT,
9172                .AC.ATK.TGT,
9173                "AC.ATK"
9174                YIELDING
9175                .RESULT,
9176                .FIRING.TIME
9177                IF .RESULT = HIT
9178                   LET .KILLED.IND  = YES
9179                   LET .WAIT.TIME = .WAIT.TIME/2
9180                ALWAYS
9181             ENDLOOP
9182          ALWAYS
9183          IF .RESULT NE HIT AND
9184             0.0 LT .DETECT.TIME LE .WAIT.TIME
9185             ''THE TARGET HAS NOT YET DETECTED THE AIRCRAFT
9186             ''BUT WILL BEFORE IT COMPLETES ITS PASS.
9187
9188 <----------WAIT .DETECT.TIME HOURS
9189
9190             IF TACAIR.DEBUG = 1
9191                PRINT 1 LINE WITH .AC.ATK.TGT, TIME.V THUS
9192                = = = = = AC.ATK.TGT ****** DETECTED AFTER FIRING AT ***.****
9193                LIST CMSN.SEQ.NR(.CAS.MISSION), CMSN.SIDE(.CAS.MISSION)
9194                ''LOOP FOR EACH .LINK IN UN.EQUIP.LIST(.TGT)
```

```
9195                ''      DO
9196                   ''      LIST ATTRIBUTES OF UE.LINK CALLED .LINK
9197                   ''      FOR EACH .SO IN SO LIST(.LINK)
9198                   ''           LIST ATTRIBUTES OF SHOOT.OUT CALLED .SO
9199                   ''      FOR EACH .FT IN UE.TARGET.LIST(.LINK)
9200                   ''           LIST ATTRIBUTES OF FIRING.TABLE CALLED .FT
9201                ''ENDLOOP
9202             ALWAYS
9203             IF .KILLED.IND = YES
9204          <--------LEAVE
9205             OTHERWISE
9206
9207             SUBTRACT .DETECT.TIME FROM .WAIT.TIME
9208             LET .VIS.IND = YES
9209             LOOP FOR EVERY .LINK IN UN.EQUIP.LIST(.TGT)
9210             FOR EVERY .SO IN SO.LIST(.LINK)
9211             WITH SO.FIRING.TABLE(.SO) = 0 AND
9212             SO.AIR.ATK.INDIC(.SO) = YES
9213             DO
9214                INTERRUPT SHOOT.OUT CALLED .SO------------------------------------------------------------------------------>(493)
9215                REACTIVATE_THE SHOOT.OUT CALLED .SO NOW---------------------------------------------------------------------->(493)
9216             ENDLOOP
9217          ALWAYS
9218
9219 <--------WAIT .WAIT.TIME HOURS
9220
9221          IF TACAIR.DEBUG = 1
9222             PRINT 1 LINE WITH .AC.ATK.TGT, TIME.V THUS
9223             = = = = = AC.ATK.TGT ****** AT P3 AT ***.****
9224             LIST CMSN.SEQ.NR(.CAS.MISSION), CMSN.SIDE(.CAS.MISSION)
9225             ''LOOP FOR EACH .LINK IN UN.EQUIP.LIST(.TGT)
9226             ''      DO
9227                ''      LIST ATTRIBUTES OF UE.LINK CALLED .LINK
9228                ''      FOR EACH .SO IN SO.LIST(.LINK)
9229                ''           LIST ATTRIBUTES OF SHOOT.OUT CALLED .SO
9230                ''      FOR EACH .FT IN UE.TARGET.LIST(.LINK)
9231                ''           LIST ATTRIBUTES OF FIRING.TABLE CALLED .FT
9232             ''ENDLOOP
9233          ALWAYS
9234          IF .KILLED.IND = YES
9235       <------LEAVE
9236          OTHERWISE
9237
9238          ''AT POINT 3.  AIRCRAFT CANNOT BE SEEN UNTIL
9239          ''IT REACHES POINT 1 AGAIN.
9240          LET .VIS.IND = NO
9241          LOOP FOR EVERY .LINK IN UN.EQUIP.LIST(.TGT)
9242          FOR EVERY .SO IN SO.LIST(.LINK)
9243          WITH SO.FIRING.TABLE(.SO) GT 0 AND
9244          FT.AC.ATK.TGT(SO.FIRING.TABLE(.SO)) = .AC.ATK.TGT
9245          DO
9246             LET SO.FIRING.TABLE(.SO)= 0
9247             LOOP FOR EACH .AS IN EV.S(1.ASSESSMENT)
9248             WITH AS.SHOOT.OUT(.AS) = .SO
9249             DO
9250                INTERRUPT ASSESSMENT CALLED .AS------------------------------------------------------------------------------>(487)
9251                LET AS.DESTRUCT.INDIC(.AS) = YES
9252                REACTIVATE_THE ASSESSMENT CALLED .AS NOW---------------------------------------------------------------------->(487)
```

```
9253            ENDLOOP
9254         ENDLOOP
9255         IF .SUM.CRIT.EQ LE TU.CRIT.NO(UN.TYPE.UNIT(.TGT)) *
9256            SD.CAS.BRKPT(CMSN.SIDE(.CAS.MISSION)) / 100
9257      <------LEAVE
9258         OTHERWISE
9259
9260         LET .WAIT.TIME = .P3.TO.P1 / (ACT.SPEED(.AC.TYPE) * 3600)
9261
9262
9263 <---------WAIT .WAIT.TIME HOURS
9264
9265         IF TACAIR.DEBUG = 1
9266            PRINT 1 LINE WITH .AC.ATK.TGT, TIME.V THUS
9267            = = = = = AC.ATK.TGT ****** AT P1 AT ***.****
9268            LIST CMSN.SEQ.NR(.CAS.MISSION), CMSN.SIDE(.CAS.MISSION)
9269            ''LOOP FOR EACH .LINK IN UN.EQUIP.LIST(.TGT)
9270            ''       DO
9271               ''       LIST ATTRIBUTES OF UE.LINK CALLED .LINK
9272               ''       FOR EACH .SO IN SO.LIST(.LINK)
9273               ''            LIST ATTRIBUTES OF SHOOT.OUT CALLED .SO
9274               ''       FOR EACH .FT IN UE.TARGET.LIST(.LINK)
9275               ''            LIST ATTRIBUTES OF FIRING.TABLE CALLED .FT
9276            ''ENDLOOP
9277         ALWAYS
9278         IF .KILLED.IND = YES
9279      <------LEAVE
9280         OTHERWISE
9281      ENDLOOP  ''FOR EACH PASS
9282   ENDLOOP     ''FOR EACH WEAPON
9283
9284   IF TACAIR.DEBUG = 1
9285      LIST ATTRIBUTES OF AC.ATK.TGT CALLED .AC.ATK.TGT
9286      LIST ATTRIBUTES OF CAS.MISSION CALLED .CAS.MISSION
9287      LIST .SUM.CRIT.EQ, .TIME.TO.LEAVE, TIME.V,
9288      TU.CRIT.NO(UN.TYPE.UNIT(.TGT)),
9289      SD.CAS.BRKPT(CMSN.SIDE(.CAS.MISSION))
9290   ALWAYS
9291   ''CLEAN UP AND RETURN TO CAS MISSION
9292
9293   LOOP FOR EACH .SO IN EV.S(I.SHOOT.OUT)
9294   WITH SO.FIRING.TABLE(.SO) GT 0 AND
9295   FT.AC.ATK.TGT(SO.FIRING.TABLE(.SO)) = .AC.ATK.TGT
9296   DO
9297      LET SO.FIRING.TABLE(.SO) = 0
9298      LOOP FOR EACH .ASSESS IN EV.S(I.ASSESSMENT)
9299      WITH AS.SHOOT.OUT(.ASSESS) = .SO
9300      DO
9301         INTERRUPT ASSESSMENT CALLED .ASSESS------------------------------------------------------------------------>(487)
9302         LET AS.DESTRUCT.INDIC(.ASSESS) = YES
9303         REACTIVATE_THE ASSESSMENT CALLED .ASSESS NOW---------------------------------------------------------->(487)
9304      ENDLOOP
9305   ENDLOOP
9306   IF .KILLED.IND = YES
9307      SUBTRACT 1 FROM CMSN.NR.SURV.AC(.CAS.MISSION)
9308   ALWAYS
9309
9310   LET .VIS.IND = NO
```

```
9311     LOOP FOR EVERY .LINK OF AATT.LIST(.AC.ATK.TGT)
9312     DO
9313        REMOVE .LINK FROM AATT.LIST(AC.ATK.TGT)
9314        DESTROY THE AAT.TGT  CALLED .LINK
9315     ENDLOOP
9316
9317     FOR EVERY AC.ATK.TGT IN EV.S(I.AC.ATK.TGT)
9318     WITH AAT.CMSN(AC.ATK.TGT) = .CAS.MISSION
9319     FIND THE FIRST CASE
9320     IF NONE
9321        ''THIS IS THE LAST AIRCRAFT MAKING PASSES.
9322        ''REACTIVATE THE MISSION AND RETURN TO BASE.
9323        INTERRUPT CAS.MISSION CALLED .CAS.MISSION----------------------------------------------------------------------------->(504)
9324        REACTIVATE_THE CAS.MISSION CALLED .CAS.MISSION NOW-------------------------------------------------------------------->(504)
9325     ALWAYS
9326
9327     DESTROY  EX.AC.ATK.TGT CALLED AAT.EX.AATGT(.AC.ATK.TGT)
9328 <--RETURN
9329 END
```

```
9330 PROCESS AIR.OBSERVER                                                    ''                                                    P003
9331
9332
9333    ADD 1 TO ANAL.CTR(169,1)  ''                                                                                           \DYN_ANAL
9334    NORMALLY MODE IS INTEGER
9335    DEFINE AO TO MEAN PROCESS.V ''                                                                                          CHG\01
9336    DEFINE VELOCITY TO MEAN AO.VELOCITY(AO)
9337    DEFINE TRANSMIT TO MEAN WAIT
9338    DEFINE SEARCH TO MEAN WAIT
9339    DEFINE FLY TO MEAN WAIT
9340    DEFINE TURN TO MEAN WAIT
9341    DEFINE PREPARE TO MEAN WAIT
9342    DEFINE GROUPING, SIDE, SENSOR.TYPE AS INTEGER VARIABLES
9343    DEFINE LEG.SLOPE, LEG.Y.INTERCEPT , LEG.LENGTH,
9344    X.PERPENDICULAR, Y.PERPENDICULAR,
9345    PERPENDICULAR.DIST, SEARCH.DIST, OLD.DIST,
9346    RETURN.TIME,APPROACH.TIME,APPROACH.DISTANCE,
9347    LEG.TIME.REMAINING, OBS.TIME.REMAINING,    ''%2JAN81_%RWF
9348    BACK.DIST, FORWARD.DIST AS REAL VARIABLES
9349    DEFINE SEARCH.TIME, XMT.TIME AS REAL VARIABLES  ''%9JAN81_%RWF
9350    DEFINE MAX.RANGE, MAX.DIST, TRANSITION.DIST
9351    AS 1-DIMENSIONAL REAL ARRAYS  ''                                                                                        \VAX
9352    RESERVE MAX.RANGE, MAX.DIST, TRANSITION.DIST AS N.TYPE.EQUIPMENT
9353    DEFINE FLIGHT.RISK AS AN INTEGER VARIABLE
9354    DEFINE RISK AS A REAL VARIABLE
9355    DEFINE FEBA.FLIGHT TO MEAN 1
9356    DEFINE FLT.TIME, FLT.TIME2, FLIGHT.TIME AS REAL VARIABLES '' %RGR
9357    DEFINE END.MINUTE AS A REAL VARIABLE  '' %5MAR80_%RGR
9358    DEFINE RETURN.MINUTE AS A REAL VARIABLE   '' %RGR
9359    LET LINK = AO.US.LINK(AO)
9360    LET AIR.STRIP = US.UNIT(LINK)
9361    LET SIDE = UN.COLOR(US.UNIT(LINK))  ''27SEP79_%RGR
9362    LET MODEL = US.MODEL(LINK)
9363    LET SENSOR.TYPE = US.SENSOR.TYPE(LINK)
9364    LET AO.VEL = MAO.VELOCITY(MODEL)*1000/16
9365    LET AO.VEL2 = INT.F((MAO.VELOCITY(MODEL)*1000/16)/ 2 )
9366    LET VELOCITY = RANDI.F(AO.VEL2,AO.VEL,RN.SEED)  '' %RGR
9367    LET RISK = RANDOM.F(RN.SEED)
9368    IF RISK GE .50
9369       LET FLIGHT.RISK = 1
9370    ELSE '' RISK IS LT .50
9371       LET FLIGHT.RISK = 1   ''%29DEC80_%RWF - ASSUME ALL FLT'S ARE FEBA FLT'S
9372    ALWAYS
9373    LET ALTITUDE = INT.F( (1.00 - RISK) * MAO.ALTITUDE(MODEL) ) '' %RGR
9374    IF SIDE = RED
9375       LET ENEMY = BLUE
9376    ELSE
9377       LET ENEMY = RED
9378    ENDIF
9379    LET FLT.TIME = REAL.F(MAO.MAX.ALOFT.TIME(MODEL))  ''MOVED HERE %16DEC80_%RWF
9380    LET FLT.TIME2 = 0.80 * FLT.TIME    ''%16DEC80_%RWF
9381    LET FLIGHT.TIME = UNIFORM.F ( FLT.TIME2, FLT.TIME, RN.SEED ) ''%16DEC80_%RWF
9382    LET APPROACH.DISTANCE = SQRT.F((UN.X.COORD(AIR.STRIP) - AO.X.START(AO))**2 +  ''                                       \OPTIMIZE
9383    (UN.Y.COORD(AIR.STRIP)-AO.Y.START(AO))**2)  ''%16DEC80_%RWF
9384    LET APPROACH.TIME = APPROACH.DISTANCE/VELOCITY '' %16DEC80_%RWF
9385    LET RETURN.TIME = APPROACH.TIME '' %16DEC80_%RWF
9386    IF (FLIGHT.TIME/60.) < APPROACH.TIME+RETURN.TIME+12./60., ''%31DEC80_%RWF
9387       LET US.STATUS(US.LINK) = IDLE ''MOVED HERE %16DEC80_%RWF
```

```
9388        LET US.ID(US.LINK) = 0   ''%22DEC80_%RWF
9389        LOOP UNTIL AO.FLIGHT.LEG.LIST(AO) IS EMPTY ''%8JAN81_%RWF
9390        DO THIS    ''                                                    \1
9391           REMOVE THE FIRST LEG FROM AO.FLIGHT.LEG.LIST(AO) ''%8JAN81_%RWF
9392           DESTROY THE FLIGHT.LEG CALLED LEG ''%8JAN81_%RWF
9393        ENDLOOP ''%8JAN81_%RWF
9394        RELEASE MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*) ''MOVED HERE %16DEC80_%RWF
9395 <------EXITPROCESS ''MOVED HERE %16DEC80_%RWF
9396     OTHERWISE ''MOVED HERE %16DEC80_%RWF
9397     PREPARE UNIFORM.F(MAO.MIN.PREP(MODEL), MAO.MAX.PREP(MODEL),RN.SEED) MINUTES
9398     IF US.STATUS(AO)=9999          '' THIS AIR OBSERVER HAS BEEN KILLED
9399        LOOP UNTIL AO.FLIGHT.LEG.LIST(AO) IS EMPTY ''%8JAN81_%RWF
9400        DO THIS ''                                                       \1
9401           REMOVE THE FIRST LEG FROM AO.FLIGHT.LEG.LIST(AO) ''%8JAN81_%RWF
9402           DESTROY THE FLIGHT.LEG CALLED LEG ''%8JAN81_%RWF
9403        ENDLOOP ''%8JAN81_%RWF
9404        DESTROY THE US.LINK CALLED LINK
9405        RELEASE MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*)
9406 <------EXITPROCESS
9407     OTHERWISE
9408     ''LINES 98 THRU 108 MOVED HERE %23DEC80_%RWF
9409     LET END.MINUTE = TIME.V * MINUTES.V + FLIGHT.TIME ''MOVED HERE %16DEC80_%RWF
9410     LET RETURN.MINUTE = END.MINUTE - RETURN.TIME*60.   ''%16DEC80_%RWF
9411     LOOP FOR EACH TYPE.EQUIPMENT CALLED EQUIP.TYPE
9412     DO THIS          ''                                                 \1
9413        LET MAX.RANGE(EQUIP.TYPE) = MAO.MAGNIFICATION(MODEL) *
9414        SQRT.F(((TE.PROJECTED.AREA(EQUIP.TYPE) *
9415        1000) ** 2)/5.)   ''%24NOV80_%RWF
9416        LET MAX.DIST(EQUIP.TYPE) = SQRT.F(MAX.RANGE(EQUIP.TYPE) ** 2 +
9417        ALTITUDE ** 2 )
9418        LET TRANSITION.DIST(EQUIP.TYPE) = -15.24 + SQRT.F((
9419        MAX.DIST(EQUIP.TYPE) ** 2)/2. - 762.) ''%21NOV80_%RWF
9420        ''THE FORGOING CALCULATIONS ARE FROM PAGES 19-23
9421        '' BRL RPT # 1923, DTD MAR 68   %21OCT80_%RWF
9422     ENDLOOP
9423     FLY APPROACH.TIME HOURS   ''CHG'D FROM MINUTES %19DEC80_%RWF
9424     IF US.STATUS(AO)=9999          '' THIS AIR OBSERVER HAS BEEN KILLED
9425        DESTROY THE US.LINK CALLED LINK
9426        RELEASE MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*)
9427 <------EXITPROCESS
9428     OTHERWISE
9429     LET LEG.NUMBER = 1
9430     FOR EACH LEG IN AO.FLIGHT.LEG.LIST(AO), FIND THE FIRST CASE ''%5JAN81_%RWF
9431     IF FOUND,                  ''%5JAN81_%RWF
9432        IF MODEL = 1 OR MODEL = 2,    ''%5JAN81_%RWF
9433           ''ONLY ONE LEG FLOWN BACK-AND-FORTH ON BEHIND-THE-FEBA OBS FLIGHT
9434           LET DELTA.Y = ABS.F(FL.Y.END(LEG) - FL.Y.START(LEG)) ''%5JAN81_%RWF
9435           IF FL.X.END(LEG) - FL.X.START(LEG) = 0,   ''MOVED HERE %5JAN81_%RWF
9436              LET DELTA.X = 1   ''%5JAN81_%RWF
9437           ELSE LET DELTA.X = ABS.F(FL.X.END(LEG) - FL.X.START(LEG))   ''%5JAN81_%RWF
9438           ALWAYS    ''%5JAN81_%RWF
9439           LET LEG.LENGTH = SQRT.F(DELTA.X ** 2 + DELTA.Y ** 2)''%5JAN81_%RWF
9440           LET NUMBER.OF.LEGS = TRUNC.F(((RETURN.MINUTE-TIME.V * MINUTES.V)/
9441           (((LEG.LENGTH/VELOCITY)*60)+2))) + 1    ''%29DEC80_%RWF
9442           LET OLD.LEG = LEG
9443           LOOP FOR I = 1 TO (NUMBER.OF.LEGS - 1)
9444           DO THIS
9445              CREATE A FLIGHT.LEG CALLED NEW.LEG
```

```
9446            FILE NEW.LEG IN AO.FLIGHT.LEG.LIST(AO)
9447            LET FL.Y.START(NEW.LEG) = FL.Y.END(OLD.LEG)
9448            LET FL.X.START(NEW.LEG) = FL.X.END(OLD.LEG)
9449            LET FL.Y.END(NEW.LEG) = FL.Y.START(OLD.LEG)
9450            LET FL.X.END(NEW.LEG) = FL.X.START(OLD.LEG)
9451            LET OLD.LEG = NEW.LEG
9452         ENDLOOP
9453      ENDIF     ''END OF CODE MOVED HERE %5JAN81_%RWF
9454   ELSE ''%5JAN81_%RWF                                                    CHANGED OTHERWISE TO ELSE TO PROCESS  \1
9455      CALL ERROR.STOP ''%5JAN81_%RWF-------------------------------------------------------------------------->(604)
9456   ALWAYS      ''%5JAN81_%RWF
9457   LOOP FOR EACH LEG IN AO.FLIGHT.LEG.LIST(AO)
9458   DO THIS  ''                                                                                                \1
9459      IF TIME.V*MINUTES.V >= RETURN.MINUTE   ''%29DEC80_%RWF
9460   <------EXITLOOP     ''%29DEC80_%RWF
9461      ENDIF     ''%29DEC80_%RWF
9462      LET DELTA.Y = ABS.F(FL.Y.END(LEG) - FL.Y.START(LEG)) ''%5JAN81_%RWF
9463      IF FL.X.END(LEG) - FL.X.START(LEG) = 0,
9464         LET DELTA.X = 1
9465      ELSE LET DELTA.X = ABS.F(FL.X.END(LEG) - FL.X.START(LEG))
9466      ALWAYS
9467      LET LEG.SLOPE = DELTA.Y/DELTA.X
9468      LET LEG.Y.INTERCEPT = FL.Y.START(LEG) - LEG.SLOPE * FL.X.START(LEG)
9469      LET LEG.LENGTH = SQRT.F(DELTA.X ** 2 + DELTA.Y ** 2)
9470      IF FLIGHT.RISK = FEBA.FLIGHT
9471         LET FIRST = MANEUVER
9472         LET LAST = SUPPORT
9473      ELSE
9474         LET FIRST = ARTILLERY
9475         LET LAST = SUPPORT
9476      ENDIF
9477      LOOP FOR GROUPING = FIRST TO LAST
9478      DO THIS  ''                                                                                             \1
9479         LOOP FOR EACH UN IN UNIT.SET(ENEMY,GROUPING)
9480         DO THIS  ''                                                                                          \1
9481            LET X.PERPENDICULAR = (UN.Y.COORD(UN) + UN.X.COORD(UN)/LEG.SLOPE -
9482            LEG.Y.INTERCEPT) / (LEG.SLOPE + 1./LEG.SLOPE)
9483            LET Y.PERPENDICULAR = LEG.SLOPE*X.PERPENDICULAR+LEG.Y.INTERCEPT
9484            LET PERPENDICULAR.DIST = SQRT.F((X.PERPENDICULAR -
9485            UN.X.COORD(UN))**2 +(Y.PERPENDICULAR - UN.Y.COORD(UN))**2 )
9486            IF PERPENDICULAR.DIST > ST.MAX.RANGE(SENSOR.TYPE)   '' IN HDM
9487         <------CYCLE
9488            OTHERWISE
9489            LET BACK.DIST = INT.F(SQRT.F((X.PERPENDICULAR-FL.X.START(LEG)) ** 2 + ''%6JAN81
9490            (Y.PERPENDICULAR-FL.Y.START(LEG)) ** 2 ))    ''%6JAN81_%RWF
9491            LET FORWARD.DIST = INT.F(SQRT.F((X.PERPENDICULAR-FL.X.END(LEG)) ** 2 + ''%6JAN81
9492            (Y.PERPENDICULAR-FL.Y.END(LEG)) ** 2 ))    ''%6JAN81_%RWF
9493            IF BACK.DIST+FORWARD.DIST > LEG.LENGTH
9494         <------CYCLE
9495            OTHERWISE
9496            CREATE AN AO.DET.CANDIDATE CALLED CANDIDATE
9497            LET AO.DC.UNIT(CANDIDATE) = UN
9498            LET AO.DC.LEG.DIST(CANDIDATE) = BACK.DIST
9499            LET AO.DC.DIST(CANDIDATE) = PERPENDICULAR.DIST
9500            FILE CANDIDATE IN AO.CAND.DET.LIST(AO)
9501         ENDLOOP
9502      ENDLOOP
9503      LET OLD.DIST = 0.0
```

```
9504        LOOP FOR EACH CANDIDATE IN AO.CAND.DET.LIST(AO)
9505        DO THIS ''                                                                                    \1
9506           REMOVE CANDIDATE FROM AO.CAND.DET.LIST(AO)  '' MOVED HERE %31DEC80_%RWF
9507           IF TIME.V * MINUTES.V >= (RETURN.MINUTE - 2.)     ''%31DEC80_%RWF
9508              DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE  ''%2JAN81_%RWF
9509        <------CYCLE
9510           OTHERWISE
9511           LET SEARCH.DIST = ABS.F(AO.DC.LEG.DIST(CANDIDATE) - OLD.DIST)
9512           '' THE SEARCH TIME MUST BE A POSITIVE VALUE  %18MAR80_%RGR
9513           LET OLD.DIST = AO.DC.LEG.DIST(CANDIDATE)
9514           LET SEARCH.TIME = SEARCH.DIST/VELOCITY  ''%29DEC80_%RWF
9515           IF TIME.V*MINUTES.V + (SEARCH.TIME*60.) + 2. > RETURN.MINUTE ''%31DEC80_%RWF
9516              LET OBS.TIME.REMAINING = ((RETURN.MINUTE-2.)/60.) - TIME.V  ''%RWF
9517              FLY OBS.TIME.REMAINING HOURS  ''%29DEC80_%RWF
9518              DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE   ''%8JAN81_%RWF
9519              IF US.STATUS(AO)=9999 ''%31DEC80_%RWF
9520                 LOOP UNTIL AO.FLIGHT.LEG.LIST(AO) IS EMPTY
9521                 DO THIS ''%31DEC80_%RWF                                                            \1
9522                    REMOVE THE FIRST LEG FROM AO.FLIGHT.LEG.LIST(AO) ''%31DEC80_%RWF
9523                    DESTROY THE FLIGHT.LEG CALLED LEG ''%31DEC80_%RWF
9524                 ENDLOOP ''%31DEC80_%RWF
9525                 LOOP UNTIL AO.CAND.DET.LIST(AO) IS EMPTY
9526                 DO THIS ''%31DEC80_%RWF                                                            \1
9527                    REMOVE THE FIRST CANDIDATE FROM AO.CAND.DET.LIST(AO) ''%31DEC80_%RWF
9528                    DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE ''%31DEC80_%RWF
9529                 ENDLOOP
9530                 DESTROY THE US.LINK CALLED LINK
9531                 RELEASE MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*)
9532 <----------EXITPROCESS
9533              OTHERWISE
9534        <------EXITLOOP   ''%29DEC80_%RWF
9535           OTHERWISE  ''%29DEC80_%RWF
9536           SEARCH SEARCH.TIME HOURS      ''%29DEC80_%RWF
9537           IF US.STATUS(AO)=9999 ''%31DEC_%RWF
9538              LOOP UNTIL AO.FLIGHT.LEG.LIST(AO) IS EMPTY
9539              DO THIS ''31DEC_%RWF                                                                  \1
9540                 REMOVE THE FIRST LEG FROM AO.FLIGHT.LEG.LIST(AO) ''%31DEC_%RWF
9541                 DESTROY THE FLIGHT.LEG CALLED LEG ''%31DEC_%RWF
9542              ENDLOOP ''%31DEC_%RWF
9543              DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE   ''%2JAN81_%RWF
9544              LOOP UNTIL AO.CAND.DET.LIST(AO) IS EMPTY
9545              DO THIS ''%31DEC_%RWF                                                                 \1
9546                 REMOVE THE FIRST CANDIDATE FROM AO.CAND.DET.LIST(AO) ''%31DEC_%RWF
9547                 DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE ''%31DEC_%RWF
9548              ENDLOOP
9549              DESTROY THE US.LINK CALLED LINK
9550              RELEASE MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*)
9551 <----------EXITPROCESS
9552           OTHERWISE
9553           CREATE A TARGET.REPORT CALLED TARGET
9554           LET AO.CURRENT.TR(AO) = TARGET
9555           CALL AO.DETECTION GIVEN AO, VELOCITY, CANDIDATE, MODEL, ALTITUDE,----------------------------------------------------------->(154)
9556           SENSOR.TYPE, MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*), TARGET
9557           IF TR.DET.LIST(TARGET) IS EMPTY
9558              DESTROY THE TARGET.REPORT CALLED TARGET   ''%29DEC80_%RWF
9559              DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE  ''%2JAN81_%RWF
9560        <------CYCLE
9561           OTHERWISE
```

```
9562          LET XMT.TIME = UNIFORM.F(REAL.F(ST.MIN.XMIT(SENSOR.TYPE)) ,
9563          REAL.F(ST.MAX.XMIT(SENSOR.TYPE)) ,RN.SEED)/10. ''%9JAN81_%RWF
9564          IF XMT.TIME < 0.    ''%9JAN81_%RWF
9565             LET XMT.TIME = 0.    ''%9JAN81_%RWF
9566          ENDIF      ''%9JAN81_%RWF
9567          TRANSMIT XMT.TIME MINUTES    ''%9JAN81_%RWF
9568          IF US.STATUS(AO)=9999 ''%31DEC80_%RWF
9569             LOOP UNTIL AO.FLIGHT.LEG.LIST(AO) IS EMPTY
9570             DO THIS ''%31DEC80_%RWF                                                                    \1
9571                REMOVE THE FIRST LEG FROM AO.FLIGHT.LEG.LIST(AO) ''%31DEC80_%RWF
9572                DESTROY THE FLIGHT.LEG CALLED LEG ''%31DEC80_%RWF
9573             ENDLOOP ''%31DEC80_%RWF
9574             DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE ''%2JAN81_%RWF
9575             LOOP UNTIL AO.CAND.DET.LIST(AO) IS EMPTY
9576             DO THIS ''%31DEC80_%RWF                                                                    \1
9577                REMOVE THE FIRST CANDIDATE FROM AO.CAND.DET.LIST(AO) ''%31DEC80_%RWF
9578                DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE ''%31DEC80_%RWF
9579             ENDLOOP
9580             DESTROY THE US.LINK CALLED LINK
9581             DESTROY THE TARGET.REPORT CALLED TARGET
9582             RELEASE MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*)
9583 <-----------EXITPROCESS
9584          OTHERWISE    ''%31DEC80_%RWF
9585          ACTIVATE_THE TARGET.REPORT CALLED TARGET NOW------------------------------------------------------------------------------>(470)
9586          LET TR.RECVD.TIME(TARGET) = TIME.V
9587          LET TR.ABORT.TIME(TARGET) = TIME.V + .5
9588          ''LET AO.CURRENT.TR(AO) = TARGET
9589          IF DEBUG = TRUE, PRINT 1 LINE WITH AO.CURRENT.TR(AO), ENEMY THUS ''%31DEC80
9590             = = = AIR.OBSERVER  AO.CURRENT.TR(AO) = ****** (TARGET) OF SIDE * = = =
9591          ENDIF          ''%31DEC80_%RWF
9592          DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE
9593       ENDLOOP     '' THRU ALL OF THE CANDIDATE UNITS ON THE LIST
9594       IF TIME.V*MINUTES.V < (RETURN.MINUTE-2.)    ''%29DEC80_%RWF
9595          AND OLD.DIST < LEG.LENGTH      ''%29DEC80_%RWF
9596          LET LEG.REMAINING = ABS.F(LEG.LENGTH - OLD.DIST) ''%30DEC80_%RWF
9597          LET LEG.TIME.REMAINING = LEG.REMAINING/VELOCITY  ''%29DEC80_%RWF
9598          IF TIME.V*MINUTES.V + (LEG.TIME.REMAINING*60.)     ''%31DEC80_%RWF
9599             + 2. > RETURN.MINUTE          ''%31DEC80_%RWF
9600             LET OBS.TIME.REMAINING = ((RETURN.MINUTE-2.)/60.) - TIME.V  ''%RWF
9601             FLY ((OBS.TIME.REMAINING*60.) + 2.) MINUTES  ''%6JAN81_%RWF
9602             IF US.STATUS(AO)=9999
9603                LOOP UNTIL AO.FLIGHT.LEG.LIST(AO) IS EMPTY
9604                DO THIS ''%31DEC80_%RWF                                                                 \1
9605                   REMOVE THE FIRST LEG FROM AO.FLIGHT.LEG.LIST(AO) ''%31DEC80_%RWF
9606                   DESTROY THE FLIGHT.LEG CALLED LEG ''%31DEC80_%RWF
9607                ENDLOOP ''%31DEC80_%RWF
9608                LOOP UNTIL AO.CAND.DET.LIST(AO) IS EMPTY
9609                DO THIS ''%31DEC80_%RWF                                                                 \1
9610                   REMOVE THE FIRST CANDIDATE FROM AO.CAND.DET.LIST(AO) ''%31DEC80_%RWF
9611                   DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE ''%31DEC80_%RWF
9612                ENDLOOP ''%31DEC80_%RWF
9613                DESTROY THE US.LINK CALLED LINK
9614                RELEASE MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*)
9615 <--------------EXITPROCESS
9616             OTHERWISE
9617    <--------EXITLOOP    ''%6JAN81_%RWF
9618          OTHERWISE
9619          FLY LEG.TIME.REMAINING HOURS  ''%29DEC80_%RWF
```

```
9620          IF US.STATUS(AO)=9999
9621             LOOP UNTIL AO.FLIGHT.LEG.LIST(AO) IS EMPTY
9622             DO THIS ''%31DEC80_%RWF                                                    \1
9623                REMOVE THE FIRST LEG FROM AO.FLIGHT.LEG.LIST(AO) ''%31DEC80_%RWF
9624                DESTROY THE FLIGHT.LEG CALLED LEG ''%31DEC80_%RWF
9625             ENDLOOP ''%31DEC80_%RWF
9626             LOOP UNTIL AO.CAND.DET.LIST(AO) IS EMPTY
9627             DO THIS ''%31DEC80_%RWF                                                    \1
9628                REMOVE THE FIRST CANDIDATE FROM AO.CAND.DET.LIST(AO) ''%31DEC80_%RWF
9629                DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE ''%31DEC80_%RWF
9630             ENDLOOP ''%31DEC80_%RWF
9631             DESTROY THE US.LINK CALLED LINK
9632             RELEASE MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*)
9633 <-----------EXITPROCESS
9634          OTHERWISE
9635       ENDIF      ''%29DEC80_%RWF
9636       IF FLIGHT.RISK = FEBA.FLIGHT
9637          TURN 2. MINUTES
9638          IF US.STATUS(AO)=9999
9639             LOOP UNTIL AO.FLIGHT.LEG.LIST(AO) IS EMPTY
9640             DO THIS ''%31DEC80_%RWF                                                    \1
9641                REMOVE THE FIRST LEG FROM AO.FLIGHT.LEG.LIST(AO) ''%31DEC80_%RWF
9642                DESTROY THE FLIGHT.LEG CALLED LEG ''%31DEC80_%RWF
9643             ENDLOOP ''%31DEC80_%RWF
9644             LOOP UNTIL AO.CAND.DET.LIST(AO) IS EMPTY
9645             DO THIS ''%31DEC80_%RWF                                                    \1
9646                REMOVE THE FIRST CANDIDATE FROM AO.CAND.DET.LIST(AO) ''%31DEC80_%RWF
9647                DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE ''%31DEC80_%RWF
9648             ENDLOOP ''%31DEC80_%RWF
9649             DESTROY THE US.LINK CALLED LINK
9650             RELEASE MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*)
9651 <-----------EXITPROCESS
9652          OTHERWISE
9653       ENDIF
9654       LET LEG.NUMBER = LEG.NUMBER + 1
9655    ENDLOOP        ''  THRU ALL THE FLIGHT LEGS IN THE AO'S LIST
9656    FLY (RETURN.TIME+60) MINUTES  ''%21NOV80_%RWF
9657    IF US.STATUS(AO)=9999
9658       LOOP UNTIL AO.FLIGHT.LEG.LIST(AO) IS EMPTY
9659       DO THIS ''%31DEC80_%RWF                                                         \1
9660          REMOVE THE FIRST LEG FROM AO.FLIGHT.LEG.LIST(AO) ''%31DEC80_%RWF
9661          DESTROY THE FLIGHT.LEG CALLED LEG ''%31DEC80_%RWF
9662       ENDLOOP ''%31DEC80_%RWF
9663       LOOP UNTIL AO.CAND.DET.LIST(AO) IS EMPTY
9664       DO THIS ''%31DEC80_%RWF                                                         \1
9665          REMOVE THE FIRST CANDIDATE FROM AO.CAND.DET.LIST(AO) ''%31DEC80_%RWF
9666          DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE ''%31DEC80_%RWF
9667       ENDLOOP ''%31DEC80_%RWF
9668       DESTROY THE US.LINK CALLED LINK
9669       RELEASE MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*)
9670 <-----EXITPROCESS
9671    OTHERWISE
9672    RELEASE MAX.RANGE(*), MAX.DIST(*), TRANSITION.DIST(*)  ''%31DEC80_%RWF
9673    LOOP UNTIL AO.FLIGHT.LEG.LIST(AO) IS EMPTY
9674    DO THIS                                ''                                       \1
9675       REMOVE THE FIRST LEG FROM AO.FLIGHT.LEG.LIST(AO)
9676       DESTROY THE FLIGHT.LEG CALLED LEG
9677    ENDLOOP
```

```
9678     LOOP UNTIL AO.CAND.DET.LIST(AO) IS EMPTY
9679     DO THIS                                                                    \1
9680        REMOVE THE FIRST CANDIDATE FROM AO.CAND.DET.LIST(AO)
9681        DESTROY THE AO.DET.CANDIDATE CALLED CANDIDATE
9682     ENDLOOP
9683     LET US.STATUS(LINK) = HOLD
9684 <—RETURN  ''  CONTROL TO THE TIMEING ROUTINE, DESTROY THIS P-NOTICE
9685 ENDPROCESS
```

```
9686 PROCESS ARTY.ASSESS                                                                                    ''                    P004
9687
9688    ADD 1 TO ANAL.CTR(170,1)  ''                                                                                          \DYN_ANAL
9689    NORMALLY MODE IS INTEGER
9690    DEFINE ARTY.ASSESS TO MEAN PROCESS.V ''                                                                               CHG\01
9691    DEFINE UNIT, FIRE.MISSION, KV.SIDE, KILLER, VICTIM,
9692    CNTR, N.KILLED AS INTEGER VARIABLES
9693    DEFINE F AS A REAL VARIABLE
9694    DEFINE DESTRUCT.INDIC TO MEAN AA.DESTRUCT.INDIC(ARTY.ASSESS)
9695
9696    LET FIRE.MISSION = AA.FIRE.MISSION(ARTY.ASSESS)
9697    LET KILLER.TB = BY.TYPE(FM.BTRY(FIRE.MISSION))
9698    LET KILLER = EQ.KV.ID(TB.HOW.EQ.ID(KILLER.TB))
9699    LET KV.SIDE = UN.COLOR(BY.UNIT(FM.BTRY(FIRE.MISSION)))
9700    LET UNIT = AA.UNIT(ARTY.ASSESS)
9701    LET TARGET = FM.TGT(FIRE.MISSION)
9702
9703    LOOP UNTIL AA.SET(ARTY.ASSESS) IS EMPTY
9704    DO THE FOLLOWING
9705
9706       ''
9707       LET EQ.KILLED=0
9708       ''
9709
9710       REMOVE THE FIRST AA.LINK FROM THE AA.SET(ARTY.ASSESS)
9711       LET .AA.LINK = AA.LINK
9712       LET VICTIM = EQ.KV.ID(UE.ID(AA.UE.LINK(.AA.LINK)))
9713       LET ALREADY.DEAD = 0
9714       LET QTY.KILLED = 0
9715       IF DESTRUCT.INDIC = NO AND
9716          N.SO.LIST(AA.UE.LINK(.AA.LINK)) LE 0
9717          LOOP FOR I = 1 TO UE.QUANT(AA.UE.LINK(.AA.LINK))
9718          DO THE FOLLOWING
9719             ACTIVATE_A SHOOT.OUT NOW------------------------------------------------------------------------------------------------------>(493)
9720             LET FIRING.EQUIP(SHOOT.OUT)=AA.UE.LINK(.AA.LINK)
9721             LET FIRER.UNIT(SHOOT.OUT)=UNIT
9722             FILE THIS SHOOT.OUT IN THE SO.LIST(AA.UE.LINK(.AA.LINK))
9723          ENDLOOP
9724
9725 <--------WAIT .1 SECONDS
9726
9727       ALWAYS
9728
9729       IF TR.PGM.STATUS(TARGET) NE TRUE
9730          IF DESTRUCT.INDIC = NO
9731             LOOP FOR EVERY SHOOT.OUT
9732             OF SO.LIST(AA.UE.LINK(.AA.LINK))
9733             DO THE FOLLOWING
9734                IF AA.FRACTION(.AA.LINK) LE RANDOM.F(RN.SEED)
9735             <------CYCLE
9736                OTHERWISE
9737
9738                IF DROP.DEAD.INDICATOR(SHOOT.OUT) = NO
9739                   ADD 1 TO QTY.KILLED
9740                   LET DROP.DEAD.INDICATOR(SHOOT.OUT) = YES
9741                   IF KILLER>0 AND VICTIM>0,
9742                      ADD 1 TO KV.SCORE(KV.SIDE,KILLER,VICTIM)
9743                   ALWAYS
```

```
9744                  SUBTRACT 1 FROM UE.QUANT(AA.UE.LINK(.AA.LINK))
9745
9746                  ''
9747                  ADD 1 TO EQ.KILLED
9748                  ''
9749
9750                  IF UN.PTR(UNIT) > 0 AND
9751                     UE.CRITICAL.EQUIP.INDIC(
9752                     AA.UE.LINK(.AA.LINK)) = TRUE
9753                     SUBTRACT 1 FROM MU.CRIT.NO(UN.PTR(UNIT))
9754                  ALWAYS
9755
9756                  INTERRUPT SHOOT.OUT------------------------------------------------------------------------------------------>(493)
9757                  LET TIME.A(SHOOT.OUT) = 1/60 ''HOUR
9758                  RESUME SHOOT.OUT---------------------------------------------------------------------------------------------->(493)
9759               ELSE
9760                  ADD 1 TO ALREADY.DEAD
9761               ALWAYS
9762            ENDLOOP
9763         ELSE
9764            LET F = AA.FRACTION(.AA.LINK)
9765            IF F GT 0,
9766               LET N = UE.QUANT(AA.UE.LINK(.AA.LINK))
9767               LET QTY.KILLED = BINOMIAL.F(N,F,RN.SEED)
9768            ELSE
9769               LET QTY.KILLED = 0
9770            ALWAYS
9771         ALWAYS
9772      ALWAYS
9773
9774      IF TR.PGM.STATUS(TARGET) EQ TRUE
9775         LET N.KILLED =
9776         AA.FRACTION(.AA.LINK)*UE.QUANT(AA.UE.LINK(.AA.LINK))
9777         LET CNTR=0
9778         IF DESTRUCT.INDIC = NO
9779            LOOP FOR EVERY SHOOT.OUT
9780            IN SO.LIST(AA.UE.LINK(.AA.LINK)),
9781            DO
9782               IF QTY.KILLED GE N.KILLED
9783               <------CYCLE
9784                  OTHERWISE
9785
9786                  IF DROP.DEAD.INDICATOR(SHOOT.OUT)=NO
9787                     ADD 1 TO QTY.KILLED
9788                     SUBTRACT 1 FROM UE.QUANT(AA.UE.LINK(.AA.LINK))
9789
9790                     ''
9791                     ADD 1 TO EQ.KILLED
9792                     ''
9793
9794                     IF UN.PTR(UNIT) > 0 AND
9795                        UE.CRITICAL.EQUIP.INDIC(
9796                        AA.UE.LINK(.AA.LINK)) = TRUE
9797                        SUBTRACT 1 FROM MU.CRIT.NO(UN.PTR(UNIT))
9798                     ALWAYS
9799                     LET DROP.DEAD.INDICATOR(SHOOT.OUT)=YES
9800                     IF KILLER GT 0 AND VICTIM GT 0
9801                        ADD 1 TO KV.SCORE(KV.SIDE,KILLER,VICTIM)
```

```
9802                  ALWAYS
9803
9804                  INTERRUPT SHOOT.OUT------------------------------------------------------------------------------------------>(493)
9805                  LET TIME.A(SHOOT.OUT)=1/60 ''HOUR
9806                  RESUME SHOOT.OUT--------------------------------------------------------------------------------------------->(493)
9807               ELSE
9808                  ADD 1 TO ALREADY.DEAD
9809               ALWAYS
9810            ENDLOOP
9811         ELSE
9812            IF AA.FRACTION(.AA.LINK) GT 0
9813               LET QTY.KILLED=N.KILLED
9814            ELSE
9815               LET QTY.KILLED=0
9816            ALWAYS
9817         ALWAYS
9818
9819         PRINT 1 LINE WITH QTY.KILLED THUS
9820         $$$ARTY.ASSESS - •••••• KILLED WITH PGM
9821
9822         ADD QTY.KILLED TO NUM.KILL.PGM
9823      ALWAYS
9824
9825      IF DESTRUCT.INDIC = YES
9826         SUBTRACT QTY.KILLED FROM UE.QUANT(AA.UE.LINK(.AA.LINK))
9827
9828         ''
9829         ADD QTY.KILLED TO EQ.KILLED
9830         ''
9831
9832         IF UN.PTR(UNIT) > 0 AND
9833            UE.CRITICAL.EQUIP.INDIC(AA.UE.LINK(.AA.LINK)) = TRUE
9834            SUBTRACT QTY.KILLED FROM MU.CRIT.NO(UN.PTR(UNIT))
9835         ALWAYS
9836      ALWAYS
9837
9838      IF ANALYSIS(6) >0 AND EQ.KILLED>0,
9839         CALL OUTPUT.ATTRITION---------------------------------------------------------------------------------------------->(611)
9840         GIVEN
9841         UNIT,
9842         TB.HOW.EQ.ID(KILLER.TB),
9843         UE.ID(AA.UE.LINK(.AA.LINK)),
9844         EQ.KILLED,
9845         "INDIRECT",
9846         FIRE.MISSION
9847      ALWAYS
9848
9849      DESTROY THE AA.LINK CALLED .AA.LINK
9850   ENDLOOP
9851
9852 <--EXITPROCESS
9853 ENDPROCESS
```

```
                                                                                                                                    P005
9854 PROCESS FORWARD.OBSERVER                                                                                                       \DYN_ANAL
9855                                                                                                                                CHG\01
9856    ADD 1 TO ANAL.CTR(171,1)  ''
9857    NORMALLY MODE IS INTEGER
9858    DEFINE FO TO MEAN PROCESS.V ''
9859    DEFINE CANDIDATE       TO MEAN EFO.CANDIDATE(FO.EX.FWD.OBSERVER(FO))
9860    DEFINE ENEMY           TO MEAN EFO.ENEMY(FO.EX.FWD.OBSERVER(FO))
9861    DEFINE FO.SEARCH.TIME  TO MEAN EFO.SEARCH.TIME(FO.EX.FWD.OBSERVER(FO))
9862    DEFINE FO.UNIT         TO MEAN EFO.FO.UNIT(FO.EX.FWD.OBSERVER(FO))
9863    DEFINE FO.X.CORRECT    TO MEAN EFO.X.CORRECT(FO.EX.FWD.OBSERVER(FO))
9864    DEFINE FO.Y.CORRECT    TO MEAN EFO.Y.CORRECT(FO.EX.FWD.OBSERVER(FO))
9865    DEFINE LINK            TO MEAN EFO.LINK(FO.EX.FWD.OBSERVER(FO))
9866    DEFINE MIN             TO MEAN EFO.MIN.XMIT(FO.EX.FWD.OBSERVER(FO))
9867    DEFINE MAX             TO MEAN EFO.MAX.XMIT(FO.EX.FWD.OBSERVER(FO))
9868    DEFINE SEARCH.PERIOD   TO MEAN EFO.PERIOD.OF.SEARCH(FO.EX.FWD.OBSERVER(FO))
9869    DEFINE SEARCH.X.GRID   TO MEAN EFO.X.SEARCH.GRID(FO.EX.FWD.OBSERVER(FO))
9870    DEFINE SEARCH.Y.GRID   TO MEAN EFO.Y.SEARCH.GRID(FO.EX.FWD.OBSERVER(FO))
9871    DEFINE START.TIME      TO MEAN EFO.START.TIME(FO.EX.FWD.OBSERVER(FO))
9872    DEFINE TARGET          TO MEAN EFO.TARGET(FO.EX.FWD.OBSERVER(FO))
9873    DEFINE TARGET.OF.OPPORTUNITY TO MEAN 0
9874    DEFINE SEARCH TO MEAN WAIT
9875    DEFINE TRANSMIT TO MEAN WAIT
9876
9877    CREATE AN EX.FWD.OBSERVER CALLED FO.EX.FWD.OBSERVER(FO)
9878
9879    LET LINK = FO.US.LINK(FO)
9880    LET FO.UNIT = US.UNIT(LINK)
9881    IF UN.COLOR( FO.UNIT ) = RED
9882       LET ENEMY = BLUE
9883    ELSE
9884       LET ENEMY = RED
9885    ALWAYS
9886
9887    LOOP UNTIL TIME.V > STOP.SIMULATION.TIME
9888    DO THIS
9889      IF FO.UNIT IS NOT IN AN FR.UNIT.SET
9890         LET BATTLE = 0
9891         LET US.STATUS(LINK) = TARGET.OF.OPPORTUNITY
9892         LET START.TIME = TIME.V + MINUTES.V
9893         CALL LOCATE.SEARCH.AREA------------------------------------------------------------------------------------------>( 98)
9894         GIVEN
9895         FO
9896         YIELDING
9897         SEARCH.X.GRID,
9898         SEARCH.Y.GRID,
9899         FO.X.CORRECT,
9900         FO.Y.CORRECT
9901        IF DEBUG = YES
9902           PRINT 2 LINES WITH
9903           FO,
9904           SEED.V(RN.SEED),
9905           SEARCH.X.GRID,
9906           SEARCH.Y.GRID,
9907           FO.X.CORRECT,
9908           FO.Y.CORRECT  THUS
9909           FORWARD.OBSE=****** - LOCATE - SEED.V=**************,
9910           SEARCH.X =*******, SEARCH.Y =*******, FO.X =*******, FO.Y =*******
9911        ALWAYS
```

```
9912
9913          LOOP FOR GROUPING = MANEUVER TO SUPPORT
9914          DO THIS
9915             LOOP FOR EACH .UNIT IN UNIT.SET( ENEMY,GROUPING)
9916             DO THIS
9917                IF UN.X.GRID(.UNIT) = SEARCH.X.GRID AND
9918                   UN.Y.GRID(.UNIT) = SEARCH.Y.GRID
9919                   CREATE AN FO.DET.CANDIDATE CALLED CANDIDATE
9920                   LET FO.DC.UNIT( CANDIDATE ) = .UNIT
9921                   ''UTILIZE ACT.RANGE FUNCTION IN NEXT ASSIGNMENT ---------- \1>(628)
9922                   LET FO.DC.RANGE( CANDIDATE ) =
9923                   ACT.RANGE(FO.UNIT, .UNIT)
9924                   FILE CANDIDATE IN FO.CAND.DET.LIST( FO )
9925                ALWAYS
9926             ENDLOOP
9927          ENDLOOP
9928          LET SEARCH.PERIOD = 1024** 2 / MFO.SEARCH.RATE(US.MODEL(LINK))
9929       -ELSE
9930          LET SEARCH.PERIOD = 1500**2 / MFO.SEARCH.RATE(US.MODEL(LINK))
9931          LET BATTLE = US.STATUS(LINK)
9932          LOOP FOR EACH FORCE IN BTL.FORCE.SET(BATTLE),
9933          WHEN FR.SIDE(FORCE) = ENEMY
9934          DO THIS
9935             LOOP FOR EACH .UNIT IN FR.UNIT.SET(FORCE)
9936             DO THIS
9937                CREATE AN FO.DET.CANDIDATE CALLED CANDIDATE
9938                LET FO.DC.UNIT(CANDIDATE) = .UNIT
9939                ''UTILIZE ACT.RANGE FUNCTION IN NEXT ASSIGNMENT ---------- \1>(628)
9940                LET FO.DC.RANGE(CANDIDATE) =
9941                ACT.RANGE(FO.UNIT, .UNIT)
9942                FILE CANDIDATE IN FO.CAND.DET.LIST(FO)
9943             ENDLOOP
9944          ENDLOOP
9945       -ALWAYS
9946       -LET FO.SEARCH.TIME = SEARCH.PERIOD /
9947       -MAX.F( 1 , N.FO.CAND.DET.LIST( FO ) )
9948
9949       -IF DEBUG = YES
9950          PRINT 1 LINE WITH
9951          FO,
9952          TIME.V,
9953          N.FO.CAND.DET.LIST(FO)  THUS
9954          FORWARD.OBSE = ******, BEFORE LOOP, TIME= ***.****, N.FO.CAND.DET.LIST= ****
9955       -ALWAYS
9956
9957       -LOOP UNTIL FO.CAND.DET.LIST( FO ) IS EMPTY
9958       -DO THIS
9959          REMOVE THE FIRST CANDIDATE FROM THE FO.CAND.DET.LIST( FO )
9960          ''DON'T SHOOT AT ANY UNIT INVOLVED IN ANOTHER BATTLE.
9961
9962          IF FO.UNIT IS NOT IN AN FR.UNIT.SET
9963             LET BATTLE = 0
9964          ELSE
9965             LET BATTLE = US.STATUS(LINK)
9966          ALWAYS
9967
9968          IF DEBUG = YES
9969             PRINT 2 LINES WITH
```

```
9970            FO,
9971            BATTLE,
9972            UN.BATTLE.INDEX(FO.DC.UNIT(CANDIDATE)),
9973            N.FO.CAND.DET.LIST(FO),
9974            M.FR.UNIT.SET(FO.DC.UNIT(CANDIDATE))
9975            THUS
9976            FORWARD.OBSE = ******, IN LOOP - BATTLE = ******, UN.BATTLE = ******,
9977            N.FO.CAND.DET.LIST = *****, M.FR.UNIT.SET = **
9978         ALWAYS
9979
9980         IF UN.BATTLE.INDEX(FO.DC.UNIT(CANDIDATE)) EQ BATTLE
9981            OR
9982            BATTLE = 0 AND FO.DC.UNIT(CANDIDATE) NOT IN A FR.UNIT.SET
9983
9984            SEARCH FO.SEARCH.TIME MINUTES
9985
9986            IF DEBUG = TRUE
9987               PRINT 1 LINE WITH
9988               FO,
9989               SEED.V(RN.SEED) THUS
9990               FORWARD.OBSE = ******  -  SEARCH 1 - SEED.V = **************
9991            ALWAYS
9992
9993            IF US.STATUS(LINK)=9999          '' THIS FO HAS BEEN KILLED
9994               DESTROY THE US.LINK CALLED LINK
9995               DESTROY THE FO.DET.CANDIDATE CALLED CANDIDATE
9996               DESTROY THE EX.FWD.OBSERVER
9997               CALLED FO.EX.FWD.OBSERVER(FO)
9998 <-------------EXITPROCESS
9999            OTHERWISE
   0
   1            CREATE A TARGET.REPORT CALLED TARGET
   2            LET FO.CURRENT.TR(FO) = TARGET
   3            CALL FO.DETECTION-------------------------------------------------------------------------------------------->(254)
   4            GIVEN
   5            CANDIDATE,
   6            FO,
   7            TARGET,
   8            FO.X.CORRECT,
   9            FO.Y.CORRECT
  10            IF DEBUG = TRUE
  11               PRINT 2 LINES WITH
  12               FO,
  13               SEED.V(RN.SEED),
  14               CANDIDATE,
  15               TARGET,
  16               FO.X.CORRECT,
  17               FO.Y.CORRECT  THUS
  18               FORWARD.OBSE = ****** - FO.DET - SEED.V =**************, CANDIDATE =*******,
  19               TARGET = ******, FO.X = ******, FO.Y = ******
  20            ALWAYS
  21
  22            IF TR.DET.LIST( TARGET ) IS NOT EMPTY
  23               LET MIN = ST.MIN.XMIT(US.SENSOR.TYPE(LINK))
  24               LET MAX = ST.MAX.XMIT(US.SENSOR.TYPE(LINK))
  25
  26               TRANSMIT UNIFORM.F(REAL.F(MIN),
  27               REAL.F(MAX), 1) / 10. MINUTES
```

```
28
29              IF DEBUG = TRUE
30                 PRINT 2 LINES WITH
31                 FO,
32                 SEED.V(RN.SEED),
33                 SEED.V(1),
34                 MIN,
35                 MAX THUS
36                 FORWARD.OBSE = ******  - TRANSMIT - SEED.V(RN.SEED) = ************
37                 SEED.V(1) = ************, MIN = ****, MAX = ****
38              ALWAYS
39              IF US.STATUS(LINK)=9999   '' THIS FO HAS BEEN KILLED
40                 DESTROY THE US.LINK CALLED LINK
41                 DESTROY THE FO.DET.CANDIDATE CALLED CANDIDATE
42                 DESTROY THE EX.FWD.OBSERVER
43                 CALLED FO.EX.FWD.OBSERVER(FO)
44 <---------------EXITPROCESS
45              OTHERWISE
46              LET TR.RECVD.TIME(TARGET) = TIME.V
47              LET TR.ABORT.TIME(TARGET) = TIME.V + .5
48              ACTIVATE_THE TARGET.REPORT CALLED TARGET NOW-------------------------------------------------------------------->(470)
49              FILE TARGET IN FO.TGT.RPT.LIST(FO)
50              ''DEBUG CHECK FOR NO ASSN OF TR.SENSOR.ID
51              ''PROBABLY FROM FO.DETECTION
52              LET TR.SENSOR.ID(TARGET) = FO  ''                                                                    \DEBUG
53              IF MFO.NAME(US.MODEL(FO.US.LINK(FO)))="RPREP"
54                 USE UNIT 43 FOR OUTPUT
55                 PRINT 1 LINE WITH TARGET, TIME.V,
56                 TR.TGT.UNIT(TARGET),
57                 UNIT.NOS(TR.REP.UNIT(TARGET))
58                 AND UN.X.COORD(TR.REP.UNIT(TARGET) THUS
59                 TR  ******    TIME.V  **.**    TGT  ****    SEN.UNIT  ****    AT  ******
60                 USE 6 FOR OUTPUT
61              ALWAYS                                    •
62           ELSE
63              DESTROY THE TARGET.REPORT CALLED TARGET
64           ALWAYS
65           LET FO.CURRENT.TR(FO) = 0
66        ALWAYS
67        DESTROY THE FO.DET.CANDIDATE CALLED CANDIDATE
68     ENDLOOP
69
70     IF (TIME.V*MINUTES.V) < (START.TIME+SEARCH.PERIOD)
71
72        SEARCH START.TIME + SEARCH.PERIOD - TIME.V*MINUTES.V
73        MINUTES
74
75        IF DEBUG = TRUE
76           PRINT 1 LINE WITH
77           FO,
78           SEED.V(RN.SEED) THUS
79           FORWARD.OBSE = ******  - SEARCH 2 - SEED.V = ************
80        ALWAYS
81        IF US.STATUS(LINK) = 9999
82           DESTROY THE US.LINK CALLED LINK
83           DESTROY THE EX.FWD.OBSERVER
84           CALLED FO.EX.FWD.OBSERVER(FO)
85 <---------EXITPROCESS
```

```
86          OTHERWISE
87        ALWAYS
88     ENDLOOP
89     DESTROY THE EX.FWD.OBSERVER CALLED FO.EX.FWD.OBSERVER(FO)
90
91 <--EXITPROCESS
92 ENDPROCESS
```

```
 93 PROCESS HC.ARRIVE.BATTLE                                                    ''                                                    P006
 94    GIVEN
 95    FARRP,
 96    TEAM
 97
 98    ADD 1 TO ANAL.CTR(172,1)  ''                                                                                                \DYN_ANAL
 99    ''THIS ROUTINE POSITIONS HELICOPTER TEAMS FOR BATTLE.
100    ''HELICOPTERS ARE PLACED ON A LINE PERPENDICULAR TO THE LINE
101    ''PASSING THROUGH THE FORCE CENTERS OF THE OPPOSING FORCES.
102    ''THE LINE IS POSITIONED AT A FRACTION OF THE HELICOPTER'S
103    ''LONGEST RANGE WEAPON FROM THE NEAREST ENEMY UNIT.  HELICOPTERS
104    ''ARE SPACED LATERALLY ALONG THE LINE ACCORDING TO AN INPUT
105    ''VARIABLE.
106
107    NORMALLY MODE IS INTEGER
108    DEFINE FARRP, TEAM, BATTLE AS INTEGER VARIABLES
109    DEFINE THETA1, THETA2, LOW.FRAC.RANGE, HIGH.FRAC.RANGE,
110    HC.DIS AS REAL VARIABLES
111
112    LET BATTLE = FP.BATTLE(FARRP)
113    LET UN.BATTLE.INDEX(FP.UNIT(FARRP)) = BATTLE
114    LET HT.ARR.BTL.TIME(TEAM) = TIME.V
115    LET HT.SIDE = UN.COLOR(FP.UNIT(FARRP))
116
117    IF HC.DEBUG = "YES"
118       PRINT 2 LINES WITH BATTLE, FARRP, TEAM, HT.LOITER.TIME(TEAM),
119       HT.SIDE AS FOLLOWS
120       ---HC.ARRIVE.BA
121       BATTLE=******  FARRP=******  TEAM=******  LOITER.TIME=****  HT.SIDE=***
122    ALWAYS
123
124    ''CHECK FOR BATTLE OVER
125    IF HT.TERMINATOR(TEAM) = YES
126 <------EXITPROCESS
127    OTHERWISE
128
129    ''MAKE SURE THAT PREVIOUS TEAM, IF ANY, IS DISENGAGED
130    FOR EVERY HEL.TARGET.ACQUISITION IN THE EV.S(I.HEL.TARGET.ACQUISITION)
131    WITH HT.FARRP(HTA.TEAM(HEL.TARGET.ACQUISITION)) = FARRP
132    FIND THE FIRST CASE
133    IF FOUND
134       INTERRUPT HEL.TARGET.ACQUISITION------------------------------------------------------------------------------------------>(454)
135       LET TIME.A(HEL.TARGET.ACQUISITION) = -RINF.C
136       RESUME HEL.TARGET.ACQUISITION---------------------------------------------------------------------------------------------->(454)
137       LET HT.STATUS(HTA.TEAM(HEL.TARGET.ACQUISITION)) =
138       RETURNING.FROM.BATTLE
139
140 <------WAIT .01 SECONDS
141
142    ALWAYS
143
144    ''CHECK FOR BATTLE OVER
145    IF HT.TERMINATOR(TEAM) = YES
146 <------EXITPROCESS
147    OTHERWISE
148
149    IF BATTLE IS NOT IN BATTLE.SET
150       PRINT 1 LINE WITH BATTLE, FARRP, TEAM, TIME.V THUS
```

```
151       ---HC$ARRIVE$BA BATTLE=******* FARRP=******,TEAM=****** BTL OVER AT **.****
152       TRACE
153 <-----STOP
154    OTHERWISE
155
156    ''ASSIGN NEW TEAM TO BATTLE
157    IF HT.SIDE = BLUE
158       LET BTL.BL.HC.TEAM(BATTLE) = TEAM
159       LET BTL.BL.FARRP(BATTLE) = FARRP
160    ELSE
161       LET BTL.RD.HC.TEAM(BATTLE) = TEAM
162       LET BTL.RD.FARRP(BATTLE) = FARRP
163    ALWAYS
164
165    ''CALCULATE FORCE CENTERS
166    LOOP FOR EVERY .FORCE IN THE BTL.FORCE.SET(BATTLE)
167    DO THE FOLLOWING
168       LOOP FOR EVERY UNIT IN THE FR.UNIT.SET(.FORCE)
169       DO THE FOLLOWING
170          IF FR.SIDE(.FORCE) = HT.SIDE
171             ADD UN.X.COORD(UNIT) TO SUP.X.COORD
172             ADD UN.Y.COORD(UNIT) TO SUP.Y.COORD
173             ADD 1 TO SUP.NO.UNITS
174          ELSE
175             ADD UN.X.COORD(UNIT) TO OPP.X.COORD
176             ADD UN.Y.COORD(UNIT) TO OPP.Y.COORD
177             ADD 1 TO OPP.NO.UNITS
178          ALWAYS
179       ENDLOOP
180    ENDLOOP
181    LET SUP.X.COORD = SUP.X.COORD / SUP.NO.UNITS
182    LET SUP.Y.COORD = SUP.Y.COORD / SUP.NO.UNITS
183    LET OPP.X.COORD = OPP.X.COORD / OPP.NO.UNITS
184    LET OPP.Y.COORD = OPP.Y.COORD / OPP.NO.UNITS
185
186    IF SUP.X.COORD = OPP.X.COORD AND SUP.Y.COORD = OPP.Y.COORD
187       ''MOVE THE FORCES 1 HDM APART TO ALLOW COMPUTATIONS TO CONTINUE
188       IF HT.SIDE = RED
189          ADD 1 TO SUP.X.COORD
190       ELSE
191          SUBTRACT 1 FROM SUP.X.COORD
192       ALWAYS
193    ALWAYS
194
195    ''FIND CLOSEST OPPOSING UNIT
196    LET MIN.DISTANCE = SQRT.F((SUP.X.COORD - OPP.X.COORD)**2 +  ''                    \OPTIMIZE
197    (SUP.Y.COORD - OPP.Y.COORD)**2)
198    LET MIN.X.COORD = OPP.X.COORD
199    LET MIN.Y.COORD = OPP.Y.COORD
200    LOOP FOR EVERY .FORCE IN THE BTL.FORCE.SET(BATTLE)
201    WITH FR.SIDE(.FORCE) NE HT.SIDE
202    DO THE FOLLOWING
203       LOOP FOR EVERY UNIT IN THE FR.UNIT.SET(.FORCE)
204       DO THE FOLLOWING
205          LET UN.DISTANCE = SQRT.F((SUP.X.COORD - UN.X.COORD(UNIT))**2
206          + (SUP.Y.COORD - UN.Y.COORD(UNIT))**2)
207          IF UN.DISTANCE < MIN.DISTANCE
208             LET MIN.DISTANCE = UN.DISTANCE
```

```
209             LET MIN.X.COORD = UN.X.COORD(UNIT)
210             LET MIN.Y.COORD = UN.Y.COORD(UNIT)
211          ALWAYS
212       ENDLOOP
213    ENDLOOP
214
215    IF SUP.X.COORD = MIN.X.COORD AND SUP.Y.COORD = MIN.Y.COORD
216       ''CLOSEST ENEMY IS CO-LOCATED WITH FORCE CENTER.
217       ''SHIFT THE FORCE CENTER 1 HDM TO ALLOW COMPUTATIONS TO CONTINUE.
218       IF HT.SIDE = RED
219          ADD 1 TO SUP.X.COORD
220       ELSE
221          SUBTRACT 1 FROM SUP.X.COORD
222       ALWAYS
223    ALWAYS
224
225    IF HC.DEBUG = "YES"
226       PRINT 4 LINES WITH BATTLE, FARRP, TEAM,
227       SUP.NO.UNITS, SUP.X.COORD, SUP.Y.COORD,
228       OPP.NO.UNITS, OPP.X.COORD, OPP.Y.COORD, MIN.DISTANCE,
229       MIN.X.COORD, MIN.Y.COORD
230       AS FOLLOWS
231       ---HC.ARRIVE.BA    BATTLE=******, FARRP=******, TEAM=******,
232       SUP.NO.UNITS=****  SUP.X.COORD=******     SUP.Y.COORD=******
233       OPP.NO.UNITS=****  OPP.Y.COORD=******     OPP.Y.COORD=******
234       MIN.DISTANCE=******   MIN.X.COORD=******MIN.Y.COORD=******
235    ALWAYS
236
237    ''COUNT THE ATTACK HELICOPTERS.
238    LOOP FOR EVERY .HELO IN THE HT.MEMBER.LIST(TEAM)
239    WITH HC.TYPE(.HELO) = ATTACK
240    DO
241       ADD 1 TO .NO.ATTACK
242    ENDLOOP
243
244    ''FIND LONGEST RANGE WEAPON
245    FOR EVERY .HELO IN THE HT.MEMBER.LIST(TEAM)
246    WITH HC.TYPE(.HELO) = ATTACK
247    FIND THE FIRST CASE
248    IF NONE
249       LET HT.TERMINATOR(TEAM) = YES
250       LET HT.STATUS(TEAM) = RELEASED
251       ACTIVATE_A HC.RETURN.FARRP------------------------------------------------------------------------------------------------------>(450)
252       GIVEN
253       FARRP,
254       TEAM
255       IN HT.MOVE.TIME(TEAM) MINUTES
256
257       IF HC.DEBUG = "YES"
258          PRINT 1 LINE WITH BATTLE, FARRP, TEAM
259          AS FOLLOWS
260          ---HC.ARRIVE.BA BATTLE=****** FARRP=****** TEAM=****** NO ATTACK IN TEAM
261       ALWAYS
262 <-----EXITPROCESS
263    OTHERWISE
264
265    FOR EVERY UE.LINK IN THE UN.EQUIP.LIST(FP.UNIT(FARRP))
266    WITH UE.ID(UE.LINK) = HC.UE.ID(.HELO)
```

```
267     FIND THE FIRST CASE
268     IF NONE
269        TRACE
270 <-------STOP
271     OTHERWISE
272
273     LET MAX.WPN.RNG = 0
274     LOOP FOR EVERY .WEAPON IN THE UE.WEAPON.SET(UE.LINK)
275     WITH TW.MAX.RANGE(WPN.ID(.WEAPON)) > MAX.WPN.RNG
276     DO
277        LET MAX.WPN.RNG = TW.MAX.RANGE(WPN.ID(.WEAPON))
278     ENDLOOP
279     IF MAX.WPN.RNG LE 0
280        TRACE
281 <-------STOP
282     OTHERWISE
283
284     ''ARM THE TEAM.
285     LOOP FOR EVERY .WEAPON IN UE.WEAPON.SET(UE.LINK)
286     DO
287        LET WPN.ROUNDS.REMAINING(.WEAPON) = .NO.ATTACK
288        * TW.BASIC.LOAD(WPN.ID(.WEAPON))
289        IF HC.DEBUG = "YES"
290           PRINT 1 LINE WITH
291           WPN.ROUNDS.REMAINING(.WEAPON),
292           TW.BASIC.LOAD(WPN.ID(.WEAPON)),
293           .NO.ATTACK    THUS
294           ---HC.ARRIVE.BA  WPN.RDS.REMAIN=**** TW.BASIC.LOAD=**** NO.ATT.HC=**
295        ALWAYS
296     ENDLOOP
297
298     ''DETERMINE ANGLE FROM SUPPORTED FORCE TO OPPOSING FORCE
299     LET X1 = OPP.X.COORD - SUP.X.COORD
300     LET Y1 = OPP.Y.COORD - SUP.Y.COORD
301     IF X1 = 0 AND Y1 = 0    ''SAME COORDINATES
302        TRACE
303 <-------STOP
304     OTHERWISE
305     IF X1 = 0
306        IF Y1 < 0
307           LET THETA1 = (3 * PI.C) / 2.   ''                                    \OPTIMIZE
308        ELSE
309           LET THETA1 = PI.C / 2
310        ALWAYS
311     ELSE
312        IF Y1 = 0
313           IF X1 < 0
314              LET THETA1 = PI.C
315           ELSE
316              LET THETA1 = 0
317           ALWAYS
318        ELSE
319           LET THETA1 = ARCTAN.F(REAL.F(Y1), REAL.F(X1))
320           IF THETA1 LT 0.0
321              ADD 2.*PI.C TO THETA1
322           ALWAYS
323        ALWAYS
324     ALWAYS
```

```
325
326     ''DETERMINE ANGLE FROM SUPPORTED FORCE TO CLOSEST OPPOSING UNIT
327     LET X2 = MIN.X.COORD - SUP.X.COORD
328     LET Y2 = MIN.Y.COORD - SUP.Y.COORD
329     IF X2 = 0 AND Y2 = 0    ''IDENTICAL COORDINATES
330        TRACE
331 <------STOP
332     ALWAYS
333     IF X2 = 0
334        IF Y2 < 0
335           LET THETA2 = (3 * PI.C) / 2.
336        ELSE
337           LET THETA2 = PI.C / 2
338        ALWAYS
339     ELSE
340        IF Y2 = 0
341           IF X1 < 0
342              LET THETA2 = PI.C
343           ELSE
344              LET THETA2 = 0
345           ALWAYS
346        ELSE
347           LET THETA2 = ARCTAN.F(REAL.F(Y2), REAL.F(X2))
348           IF THETA2 LT 0.0
349              ADD 2.*PI.C TO THETA2
350           ALWAYS
351        ALWAYS
352     ALWAYS
353
354     LET SB1 = MIN.DISTANCE * COS.F(THETA1 - THETA2)
355     ''ADJUST RANGE TO A FRACTION OF MAXIMUM WEAPON RANGE
356     IF HT.SIDE = BLUE
357        LET LOW.FRAC.RANGE = BL.LOW.FRAC.RANGE
358        LET HIGH.FRAC.RANGE = BL.HIGH.FRAC.RANGE
359        LET HC.SPACING = BL.HC.SPACING
360     ELSE
361        LET LOW.FRAC.RANGE = RD.LOW.FRAC.RANGE
362        LET HIGH.FRAC.RANGE = RD.HIGH.FRAC.RANGE
363        LET HC.SPACING = RD.HC.SPACING
364     ALWAYS
365     LET FRAC.WPN.RNG = MAX.WPN.RNG
366     * UNIFORM.F(LOW.FRAC.RANGE, HIGH.FRAC.RANGE, RN.SEED)
367     IF SB1 GE FRAC.WPN.RNG
368        LET SB.X.COORD = SUP.X.COORD
369        LET SB.Y.COORD = SUP.Y.COORD
370     ELSE
371        LET SB2 = FRAC.WPN.RNG - SB1
372        LET SB.X.COORD = SUP.X.COORD - (SB2 * COS.F(THETA1))
373        LET SB.Y.COORD = SUP.Y.COORD - (SB2 * SIN.F(THETA1))
374     ALWAYS
375
376     IF HC.DEBUG = "YES"
377        PRINT 4 LINES WITH SB.X.COORD, SB.Y.COORD,
378        SB1, SB2,
379        FRAC.WPN.RNG, N.HT.MEMBER.LIST(TEAM) AS FOLLOWS
380        ----TRACE FROM HC.ARRIVE.BATTLE----
381        SB.X.COORD=******    SB.Y.COORD=******    SB1=*****    SB2=*****
382        FRAC.WPN.RNG=******   NO.HC=****
```

```
383         ----END TRACE----
384      ALWAYS
385
386      ''POSITION HELICOPTERS PERPENDICULAR TO THE LINE PASSING THROUGH THE
387      ''FORCE CENTERS.  SPACE THE HELICOPTERS ACCORDING TO THE VARIABLE
388      ''HC.SPACING
389      LET HC.DIS = .5 * HC.SPACING * (N.HT.MEMBER.LIST(TEAM) - 1)
390      LOOP FOR EVERY .HELO IN THE HT.MEMBER.LIST(TEAM)
391      DO THE FOLLOWING
392         LET HC.X(.HELO) = SB.X.COORD - HC.DIS * COS.F(PI.C / 2 - THETA1)
393         LET HC.Y(.HELO) = SB.Y.COORD + HC.DIS * SIN.F(PI.C / 2 - THETA1)
394         LET HC.DIS = HC.DIS - HC.SPACING
395         IF HC.DEBUG = "YES"
396            PRINT 2 LINES WITH .HELO, HC.X(.HELO), HC.Y(.HELO), HC.DIS
397            AS FOLLOWS
398            ----TRACE FROM HC.ARRIVE.BATTLE----
399            HC=******   HC.X(HC)=*****    HC.Y(HC)=*****    HC.DIS=*****.*
400         ALWAYS
401      ENDLOOP
402
403      ''CHECK IF SUPPORTED SIDE WINNING
404      IF MIN.DISTANCE > MAX.WPN.RNG
405         FOR EVERY FORCE IN THE BTL.FORCE.SET(BATTLE)
406         WITH FR.SIDE(FORCE) NE HT.SIDE
407         FIND THE FIRST CASE
408         IF NONE
409            TRACE
410 <----------STOP
411         ALWAYS
412         FOR EVERY UNIT IN THE FR.UNIT.SET(FORCE)
413         WITH UN.STATUS(UNIT) NE WITHDRAWING AND
414         UN.STATUS(UNIT) NE STA.TO.WITH AND
415         UN.STATUS(UNIT) NE ADV.TO.WITH
416         FIND THE FIRST CASE
417         IF NONE      ''DISENGAGE FROM BATTLE
418            IF HT.SIDE = BLUE
419               LET BLUE.HB.PRIORITY(BATTLE) = 0
420            ELSE
421               LET RED.HB.PRIORITY(BATTLE) = 0
422            ALWAYS
423            CALL HC.EMPTY------------------------------------------------------------------------------------------------------------------------>(337)
424            GIVING
425            FARRP,
426            HT.SIDE
427            LET HT.TERMINATOR(TEAM) = YES
428            LET HT.STATUS(TEAM) = RELEASED
429            ACTIVATE_A HC.RETURN.FARRP------------------------------------------------------------------------------------------------------------>(450)
430            GIVEN
431            FARRP,
432            TEAM
433            IN HT.MOVE.TIME(TEAM) MINUTES
434            IF HC.DEBUG = "YES"
435               PRINT 1 LINE WITH FARRP, TEAM
436               AS FOLLOWS
437               ----HC.ARRIVE----  FARRP=******  TEAM=******
438            ALWAYS
439 <----------EXITPROCESS
440         OTHERWISE
```

```
441     ALWAYS
442
443     ''START HELICOPTER BATTLE
444     IF HT.SIDE = BLUE
445        LET OPP.UNITS = BTL.RD.UNITS(BATTLE)
446     ELSE
447        LET OPP.UNITS = BTL.BL.UNITS(BATTLE)
448     ALWAYS
449     ACTIVATE_A HEL.TARGET.ACQUISITION------------------------------------------------------------------------------------------>(454)
450     GIVEN
451     TEAM,
452     OPP.UNITS NOW ''                                                                                                          \1
453     IF HC.DEBUG = "YES"
454        PRINT 1 LINE WITH TEAM, OPP.UNITS AS FOLLOWS
455        ---- HC.ARRIVE ----  TEAM=******  OPP.UNITS=******
456     ALWAYS
457
458 <--EXITPROCESS
459 ENDPROCESS
```

```
460 PROCESS HC.RETURN.FARRP                                                                                    ''                                                                                
```

```
518
519     IF HC.DEBUG = "YES"
520        PRINT 1 LINE WITH ATK.FAIL.PROB, SCT.FAIL.PROB, RN.SEED
521        AS FOLLOWS
522        ----HC.RETURN----  ATK.FAIL.PROB=**.*** SCT.FAIL.PROB=**.*** RN.SEED=***
523     ALWAYS
524
525     IF NO.ATTACK > 0 AND ATK.FAIL.PROB GT 0.0
526        LET ATK.ATTRITION = BINOMIAL.F(NO.ATTACK,ATK.FAIL.PROB,RN.SEED)
527        LOOP FOR I = 1 TO ATK.ATTRITION
528        DO THE FOLLOWING
529           FOR EVERY HC IN THE HT.MEMBER.LIST(HRF.TEAM)
530           WITH HC.TYPE(HC) = ATTACK
531           FIND THE FIRST CASE
532           IF FOUND
533              REMOVE THE HC FROM THE HT.MEMBER.LIST(HRF.TEAM)
534              DESTROY THE HELICOPTER CALLED HC
535           ALWAYS
536        ENDLOOP
537        LET NO.ATTACK = NO.ATTACK - ATK.ATTRITION
538     ALWAYS
539     IF NO.SCOUT > 0  AND  SCT.FAIL.PROB GT 0.0
540        LET SCT.ATTRITION = BINOMIAL.F(NO.SCOUT, SCT.FAIL.PROB, RN.SEED)
541        LOOP FOR I = 1 TO SCT.ATTRITION
542        DO THE FOLLOWING
543           FOR EVERY HC IN THE HT.MEMBER.LIST(HRF.TEAM)
544           WITH HC.TYPE(HC) = SCOUT
545           FIND THE FIRST CASE
546           IF FOUND
547              REMOVE THE HC FROM THE HT.MEMBER.LIST(HRF.TEAM)
548              DESTROY THE HELICOPTER CALLED HC
549           ALWAYS
550        ENDLOOP
551        LET NO.SCOUT = NO.SCOUT - SCT.ATTRITION
552     ALWAYS
553
554     IF HC.DEBUG = "YES"
555        PRINT 2 LINE WITH ATK.ATTRITION, SCT.ATTRITION AS FOLLOWS
556        ----TRACE FROM HC.RETURN.FARRP----
557        ATK.ATTRITION = ****   SCT.ATTRITION = ****
558     ALWAYS
559
560     ''WHEN STATUS IS READY, TEAM HAS BEEN REARMED/REFUELED                    \1
561     IF HT.STATUS(HRF.TEAM) NE READY
562        LET HT.STATUS(HRF.TEAM) = REARMING.REFUELING
563        LET RR.TIME = MAX.F((NO.ATTACK + NO.SCOUT) / REFUEL.CAP(HRF.FARRP)
564        * REFUEL.TIME(HRF.FARRP), NO.ATTACK / REARM.CAP(HRF.FARRP) *
565        REARM.TIME(HRF.FARRP))
566
567 <----WAIT RR.TIME MINUTES                ''REARM AND REFUEL TEAM
568
569     ALWAYS
570
571     IF HC.DEBUG = "YES"
572        PRINT 2 LINES WITH NO.SCOUT, NO.ATTACK, HRF.TEAM, HRF.FARRP,
573        RR.TIME, HT.STATUS(HRF.TEAM) AS FOLLOWS
574        ----TRACE FROM HC.RETURN.FARRP----  NO.SCOUT=***  NO.ATTACK=***
575        TEAM=******  FARRP=******  TIME=*****  HT.STATUS=***
```

```
576    ALWAYS
577
578    '' CHECK IF BATTLE OVER FOR CURRENT REARMING/REFUELING TEAM
579    IF HT.STATUS(HRF.TEAM) = RELEASED OR    ''BATTLE OVER FOR HELICOPTERS
580       HT.STATUS(HRF.TEAM) = READY OR
581       HT.TERMINATOR(HRF.TEAM) = YES
582       CALL REPLACE.HC                ''RETURN HELICOPTERS TO UNIT------------------------------------------------------------>(304)
583       GIVING
584       HRF.FARRP,
585       HRF.TEAM
586       ''CHECK FOR ALL TEAMS RETURNED
587       IF N.HT.LIST(HRF.FARRP) > 0
588 <--------EXITPROCESS
589       OTHERWISE
590
591       ''COUNT NUMBER OF ATTACK HELICOPTERS IN UNIT
592       LOOP FOR EVERY LINK IN THE UN.EQUIP.LIST(FP.UNIT(HRF.FARRP))
593       DO THE FOLLOWING
594          LOOP FOR I = 1 TO NO.ATTACK.CONFIGURATIONS
595          DO THE FOLLOWING
596             IF EQ.NAME(UE.ID(LINK)) = ATK.HC(I)
597                LET ATK.COUNT = UE.QUANT(LINK)
598                LET ATK.UE.ID = UE.ID(LINK)
599          <-------EXITLOOP
600             OTHERWISE
601          ENDLOOP
602          LOOP FOR I = 1 TO NO.SCOUT.CONFIGURATIONS
603          DO THE FOLLOWING
604             IF EQ.NAME(UE.ID(LINK)) = SCT.HC(I)
605                LET SCT.UE.ID = UE.ID(LINK)
606          <-------EXITLOOP
607             OTHERWISE
608          ENDLOOP
609       ENDLOOP
610       FOR EVERY .BATTLE IN BATTLE.SET
611       WITH .BATTLE = FP.BATTLE(HRF.FARRP)
612       FIND THE FIRST CASE
613       IF FOUND
614          IF TEAM.SIDE = RED
615             IF BTL.RD.FARRP(.BATTLE) = HRF.FARRP
616                LET BTL.RD.HC.TEAM(FP.BATTLE(HRF.FARRP)) = 0
617                LET BTL.RD.FARRP(FP.BATTLE(HRF.FARRP)) = 0
618             ALWAYS
619          ELSE
620             IF BTL.BL.FARRP(.BATTLE) = HRF.FARRP
621                LET BTL.BL.HC.TEAM(FP.BATTLE(HRF.FARRP)) = 0
622                LET BTL.BL.FARRP(FP.BATTLE(HRF.FARRP)) = 0
623             ALWAYS
624          ALWAYS
625       ALWAYS
626       LET FP.BATTLE(HRF.FARRP) = 0
627       IF NO.ATTACK = 0 AND N.HT.LIST(HRF.FARRP) = 0
628          LET FP.BATTLE(HRF.FARRP) = -1
629       ALWAYS
630       IF ATK.COUNT GT 0
631          CALL BTL.CHECK-------------------------------------------------------------------------------------------------------->(128)
632          GIVING
633          HRF.FARRP,
```

```
634          SCT.UE.ID,
635          ATK.UE.ID
636       ALWAYS
637 <-----EXITPROCESS
638    OTHERWISE
639
640    ''RETIRE TEAM IF ALL ATTACK HELICOPTERS LOST
641    IF NO.ATTACK = 0
642       LET HT.STATUS(HRF.TEAM) = READY
643       CALL REPLACE.HC------------------------------------------------------------------------------------------------------------------>(304)
644       GIVING
645       HRF.FARRP,
646       HRF.TEAM
647       IF N.HT.LIST(HRF.FARRP) = 0                    ''NO ACTIVE TEAMS
648          ''WITHDRAW FARRP FROM FUTURE CONSIDERATION
649          LET FP.BATTLE(HRF.FARRP) = -1
650          ''WITHDRAW FARRP FROM BATTLE
651          IF TEAM.SIDE = BLUE
652             LET BTL.BL.FARRP(BTL) = 0
653          ELSE
654             LET BTL.RD.FARRP(BTL) = 0
655          ALWAYS
656          LET UN.BATTLE.INDEX(FP.UNIT(HRF.FARRP)) = 0
657       ALWAYS
658 <-----EXITPROCESS
659    OTHERWISE
660
661    ''DETERMINE WHEN TO SEND THE NEXT TEAM BASED ON THE NUMBER
662    ''OF TEAMS REMAINING
663    IF N.HT.LIST(HRF.FARRP) = 1
664       SCHEDULE_A SEND.TEAM-------------------------------------------------------------------------------------------------------------->(391)
665       GIVING
666       HRF.FARRP,
667       HRF.TEAM NOW ''                                                                                                                  \1
668    ELSE
669       LET FLIGHT.TIME = HT.MOVE.TIME(HRF.TEAM)
670       LET LOITER.TIME = HT.LOITER.TIME(HRF.TEAM)
671       IF N.HT.LIST(HRF.FARRP) = 2
672          LET WAIT.TIME = LOITER.TIME - 2 * FLIGHT.TIME - RR.TIME
673       ELSE
674          ''THREE TEAMS
675          LET WAIT.TIME = 2 * LOITER.TIME - 2 * FLIGHT.TIME - RR.TIME
676       ALWAYS
677       IF WAIT.TIME > 0
678          SCHEDULE_A SEND.TEAM----------------------------------------------------------------------------------------------------------->(391)
679          GIVING
680          HRF.FARRP,
681          HRF.TEAM
682          IN WAIT.TIME MINUTES
683       ELSE
684          SCHEDULE_A SEND.TEAM----------------------------------------------------------------------------------------------------------->(391)
685          GIVING
686          HRF.FARRP,
687          HRF.TEAM NOW ''                                                                                                               \1
688       ALWAYS
689    ALWAYS
690 <--EXITPROCESS
691 ENDPROCESS
```

```
692 PROCESS HEL.TARGET.ACQUISITION                                    ''                    P008
693    GIVEN
694    .HC.TEAM,
695    .ENEMY.UNITS
696
697    ADD 1 TO ANAL.CTR(174,1)  ''                                                        \DYN_ANAL
698    ''THIS PROCESS IS ACTIVATED UPON A HELICOPTER
699    ''FORCES' ARRIVAL IN THE BATTLE AREA.  IT MODELS THE
700    ''ACQUISITION OF GROUND TARGET UNITS BY SCOUT AND ATTACK
701    ''HELICOPTERS AND CONTROLS THE MASK AND UNMASK STATUS OF THE
702    ''HELICOPTER ATTACK TEAM.  WHEN AN ACQUISITION IS MADE A
703    ''HEL.ENGAGEMENT IS SCHEDULED.
704
705    NORMALLY MODE IS INTEGER
706    DEFINE .FLY TO MEAN WORK
707    DEFINE .ENEMY.UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
708    DEFINE .TIME.TO.DETECT, .HAND.OFF.TIME, .ACQUIS.TIME,
709    .RETURN.TIME, .DETECT.TIME
710    AS REAL VARIABLES
711
712    LET .NO.ENEMY.UNITS = DIM.F(.ENEMY.UNITS(*))
713    LET .TERRAIN = BTL.TERRAIN.TYPE(FP.BATTLE(HT.FARRP(.HC.TEAM)))
714    LET HTA.BATTLE(HEL.TARGET.ACQUISITION) = FP.BATTLE(HT.FARRP(.HC.TEAM))
715
716    IF HC.DEBUG = "YES"
717       PRINT 1 LINE WITH FP.BATTLE(HT.FARRP(.HC.TEAM)),
718       HT.FARRP(.HC.TEAM), .HC.TEAM THUS
719       ---HEL.TARGET.A  BTL=******, FARRP=******, TEAM=******
720    ALWAYS
721
722    IF HT.TERMINATOR(.HC.TEAM) = YES  OR
723       HTA.REINFORCE.IND(HEL.TARGET.ACQUISITION) = YES
724 <------EXITPROCESS
725    ALWAYS
726
727    IF UN.COLOR(FP.UNIT(HT.FARRP(.HC.TEAM))) = BLUE
728       LET .HAND.OFF.TIME =
729       UNIFORM.F(BL.MIN.HANDOFF.TIME, BL.MAX.HANDOFF.TIME,
730       RN.SEED)    ''SECONDS
731       LET .ACQUIS.TIME = UNIFORM.F(BL.MIN.UNMASK.TIME,
732       BL.MAX.UNMASK.TIME, RN.SEED)    ''SECONDS
733    ELSE
734       LET .HAND.OFF.TIME =
735       UNIFORM.F(RD.MIN.HANDOFF.TIME, RD.MAX.HANDOFF.TIME,
736       RN.SEED)    ''SECONDS
737       LET .ACQUIS.TIME = UNIFORM.F(RD.MIN.UNMASK.TIME,
738       RD.MAX.UNMASK.TIME, RN.SEED)  ''SECONDS
739    ALWAYS
740
741    LET .RETURN.TIME = (TIME.V * MINUTES.V) + HT.LOITER.TIME(.HC.TEAM)
742
743    LOOP UNTIL (TIME.V * MINUTES.V) GE .RETURN.TIME
744    DO THIS
745
746       LOOP FOR EVERY .HELICOPTER OF HT.MEMBER.LIST(.HC.TEAM)
747       DO THIS
748          LET HC.STATUS(.HELICOPTER) = READY
749          LET HC.ALTITUDE(.HELICOPTER) =  ON.GROUND
```

```
750         LET HC.PAIRED(.HELICOPTER) = NO
751      ENDLOOP
752
753      ''PAIR THE TEAM MEMBERS INTO SCOUT-ATTACK FIRE TEAMS
754
755      LOOP FOR EVERY ..HELICOPTER OF HT.MEMBER.LIST(.HC.TEAM)
756      WITH HC.TYPE(..HELICOPTER) = SCOUT
757      DO
758         FOR EVERY .HELICOPTER OF HT.MEMBER.LIST(.HC.TEAM)
759         WITH HC.TYPE(.HELICOPTER) = ATTACK AND
760         HC.PAIRED(.HELICOPTER) = NO
761         FIND THE FIRST CASE
762         IF FOUND
763            LET HC.PAIRED(..HELICOPTER) = YES
764            LET HC.PAIRED(.HELICOPTER) = YES
765         ALWAYS
766      ENDLOOP
767
768      ''COUNT THE TEAM MEMBERS
769
770      LET .SCOUT.COUNT = 0
771      LET .ATTACK.COUNT = 0
772      LOOP FOR EVERY .HELICOPTER OF THE HT.MEMBER.LIST(.HC.TEAM)
773      DO THIS
774         IF HC.TYPE(.HELICOPTER) = SCOUT
775            ADD 1 TO .SCOUT.COUNT
776            LET HC.ALTITUDE(.HELICOPTER) = UNMASKED
777            LET HC.STATUS(.HELICOPTER) = DETECTING
778         ELSE
779            ADD 1 TO .ATTACK.COUNT
780            IF HC.PAIRED(.HELICOPTER) = NO
781               LET HC.ALTITUDE(.HELICOPTER) = UNMASKED
782               LET HC.STATUS(.HELICOPTER) = DETECTING
783            ALWAYS
784         ALWAYS
785      ENDLOOP
786
787      IF HC.DEBUG = "YES"
788         PRINT 1 LINE WITH FP.BATTLE(HT.FARRP(.HC.TEAM)),
789         HT.FARRP(.HC.TEAM), .HC.TEAM,
790         .ATTACK.COUNT, .SCOUT.COUNT  THUS
791         ---HEL.TARGET.A  BTL=******, FARRP=******, TEAM=******, ATK=**, SCT=**
792      ALWAYS
793
794      IF .ATTACK.COUNT = 0 ''THEY'RE ALL KILLED
795   <------EXITLOOP
796      OTHERWISE
797
798      ''ESTABLISH LISTS OF VISIBLE UNITS AND HELICOPTERS
799
800      LOOP FOR EACH ..HELICOPTER OF HT.MEMBER.LIST(.HC.TEAM)
801      WHEN HC.ALTITUDE(..HELICOPTER) =  UNMASKED
802      DO THIS
803         LOOP FOR .I = 1 TO .NO.ENEMY.UNITS
804         DO
805            CALL HEL.RANGE.COMPUTE------------------------------------------------------------------------------------------------------>(302)
806            GIVEN ..HELICOPTER AND
807             .ENEMY.UNITS(.I)
```

```
808             YIELDING .RANGE
809             IF UN.STATUS( .ENEMY.UNITS(.I)) = STATIONARY
810                LET .RNG = TT.STATIONARY.LOS.BREAK(.TERRAIN)
811             ELSE
812                LET .RNG = TT.M.S.LOS.BREAK(.TERRAIN)
813             ALWAYS
814             FOR EVERY LOS.BAND
815             WITH BAND.RANGE(LOS.BAND) GT .RANGE
816             FIND THE FIRST CASE
817             IF NONE
818                IF HC.DEBUG = "YES"
819                   PRINT 1 LINE WITH .ENEMY.UNITS(.I),
820                   UNIT.NOS(.ENEMY.UNITS(.I)) THUS
821                   ——HEL.TARGET.A  ENEMY.UNIT **** UNIT NOS= ****** TOO FAR FROM HELICOPTER
822                ALWAYS
823             ELSE
824                IF .RANGE LE .RNG  OR
825                   (LOS.PROB(.TERRAIN,LOS.BAND)/50)
826                   GE RANDOM.F(RN.SEED)
827                   FOR EACH VISIBLE.UNIT IN THE
828                   HC.UN.LOS.LIST(..HELICOPTER)
829                   WITH .ENEMY.UNITS(.I) =
830                   VU.POINTER(VISIBLE.UNIT)
831                   FIND THE FIRST CASE
832                   IF NONE
833                      ''THIS UNIT WAS NOT SEEN BEFORE
834                      CREATE A VISIBLE.UNIT
835                      LET VU.POINTER(VISIBLE.UNIT)
836                      = .ENEMY.UNITS(.I)
837                      LET VU.STATUS(VISIBLE.UNIT) = NO
838                      FILE VISIBLE.UNIT IN
839                      THE HC.UN.LOS.LIST(..HELICOPTER)
840                   ALWAYS
841                   FOR EACH VISIBLE.UNIT IN THE
842                   UN.HC.LOS.LIST(.ENEMY.UNITS(.I))
843                   WITH ..HELICOPTER =
844                   VU.POINTER(VISIBLE.UNIT)
845                   FIND THE FIRST CASE
846                   IF NONE
847                      CREATE A VISIBLE.UNIT
848                      LET VU.POINTER(VISIBLE.UNIT)
849                      = ..HELICOPTER
850                      LET VU.STATUS(VISIBLE.UNIT) = NO
851                      FILE VISIBLE.UNIT IN THE
852                      UN.HC.LOS.LIST(.ENEMY.UNITS(.I))
853                   ALWAYS
854                ALWAYS
855             ALWAYS
856          ENDLOOP
857       ENDLOOP
858
859       ''THE HELICOPTERS WILL EVALUATE THE ENEMY UNITS
860
861       LOOP FOR EACH ..HELICOPTER OF HT.MEMBER.LIST(.HC.TEAM)
862       WITH HC.ALTITUDE(..HELICOPTER) =  UNMASKED
863       DO THIS
864          LOOP FOR EACH VISIBLE.UNIT IN HC.UN.LOS.LIST(..HELICOPTER)
865          DO THIS
```

```
866             LET .TARGET.UNIT = VU.POINTER(VISIBLE.UNIT)
867             IF VU.PREV.ENG(VISIBLE.UNIT) = YES
868                SCHEDULE_A HELO.ENGAGEMENT------------------------------------------------------------------------------------>(376)
869                GIVEN
870                .HC.TEAM,
871                HT.FARRP(.HC.TEAM),
872                ..HELICOPTER,
873                .TARGET.UNIT NOW ''                                                                                    \1
874                IF HC.DEBUG = "YES"  ''                                                                          \ERROR_CORRECTION
875                   PRINT 1 LINE WITH .HC.TEAM,
876                   ..HELICOPTER, .TARGET.UNIT, HT.FARRP
877                   (.HC.TEAM) AS FOLLOWS
878                   ----HTA (10.1)---- TEAM=******  HC=****** TGT.UNIT=****** FARRP=******
879                ALWAYS
880             ELSE
881                LET .TIME.TO.DETECT = RINF.C
882                CALL HEL.RANGE.COMPUTE----------------------------------------------------------------------------------->(302)
883                GIVEN
884                ..HELICOPTER,
885                .TARGET.UNIT
886                YIELDING
887                .RANGE
888                IF .RANGE = 0
889                   LET .RANGE = 1 ''HDM
890                   PRINT 2 LINES WITH
891                   FP.BATTLE(HT.FARRP(.HC.TEAM)),
892                   HT.FARRP(.HC.TEAM),
893                   .HC.TEAM,
894                   UNIT.NOS(.TARGET.UNIT) THUS
895                   ----HEL$TARGET$A   WARNING: HELICOPTER CO-LOCATED WITH TARGET UNIT
896                   BTL=******, FARRP=******, TEAM=******, TGT UNIT.NOS= ******
897                ALWAYS
898                FOR EVERY .HELICOPTER IN THE
899                UN.EQUIP.LIST(FP.UNIT(HT.FARRP(.HC.TEAM)))
900                WITH UE.ID(.HELICOPTER) =
901                HC.UE.ID(..HELICOPTER)
902                FIND THE FIRST CASE
903                IF NONE
904                   TRACE
905 <--------------   ----STOP
906                ALWAYS
907                LOOP FOR EVERY .WEAPON OF THE
908                UE.WEAPON.SET(.HELICOPTER)
909                WITH (WPN.QUANTITY(.WEAPON) GT 0 OR
910                HC.TYPE(..HELICOPTER) = SCOUT) AND
911                TW.NO.SENSORS(WPN.ID(.WEAPON)) GT 0 AND
912                TW.MIN.RANGE(WPN.ID(.WEAPON)) LE .RANGE
913                AND
914                .RANGE LE TW.MAX.RANGE(WPN.ID(.WEAPON))
915                DO THIS
916                   ''LOOK AT EACH ITEM OF EQUIP OWNED BY ENEMY
917                   LOOP FOR EACH .UEL
918                   OF UN.EQUIP.LIST(.TARGET.UNIT)
919                   WITH UE.QUANT(.UEL) GT 0
920                   DO
921                      LET .COUNT = WPN.QUANTITY(.WEAPON)
922                      * TW.NO.SENSORS(WPN.ID(.WEAPON))
923                      CALL SEARCH------------------------------------------------------------------------------------------>(120)
```

```
924                  GIVEN
925                  WPN.ID(.WEAPON),
926                  .RANGE,
927                  RANDOM.F(RN.SEED),
928                  EQ.TE.PTR(UE.ID(.UEL)),
929                  .COUNT
930                  YIELDING
931                  .DETECT.TIME
932                  LET .TIME.TO.DETECT = MIN.F(
933                  .DETECT.TIME, .TIME.TO.DETECT)
934               ENDLOOP
935            ENDLOOP
936
937            ''CHECK IF TARGET HAS BEEN FILED BY
938            ''ANOTHER HELICOPTER.
939            IF .TARGET.UNIT IS NOT IN THE HT.TARGET.LIST
940               FILE .TARGET.UNIT IN HT.TARGET.LIST(.HC.TEAM)
941            ALWAYS
942            IF HC.TYPE(.HELICOPTER) =  SCOUT
943               FOR EACH .HELICOPTER IN THE
944               HT.MEMBER.LIST(.HC.TEAM)
945               WITH HC.ALTITUDE(.HELICOPTER)
946               =  MASKED AND
947               HC.TYPE(.HELICOPTER) = ATTACK
948               FIND THE FIRST CASE
949               IF FOUND
950                  SCHEDULE_A HELO.ENGAGEMENT------------------------------------------------------------------------------------------>(376)
951                  GIVEN
952                  .HC.TEAM,
953                  HT.FARRP(.HC.TEAM),
954                  .HELICOPTER,
955                  .TARGET.UNIT
956                  IN (.TIME.TO.DETECT +
957                  .HAND.OFF.TIME) SECONDS
958                  IF HC.DEBUG = "YES"
959                     PRINT 1 LINE WITH .HC.TEAM,
960                     .HELICOPTER, .TARGET.UNIT,
961                     HT.FARRP(.HC.TEAM), .TIME.TO.DETECT
962                     AS FOLLOWS
963                     HTA(09) TEAM=****** HC=****** TGT.UNIT=****** FARRP=****** TTD=****.***SEC
964                  ALWAYS
965               ALWAYS
966            ELSE
967               ''THE HELICOPTER TYPE IS 'ATTACK'
968               SCHEDULE_A HELO.ENGAGEMENT------------------------------------------------------------------------------------------>(376)
969               GIVEN
970               .HC.TEAM,
971               HT.FARRP(.HC.TEAM),
972               ..HELICOPTER,
973               .TARGET.UNIT
974               IN .TIME.TO.DETECT SECONDS
975               IF HC.DEBUG = "YES"
976                  PRINT 1 LINE WITH .HC.TEAM,
977                  ..HELICOPTER, .TARGET.UNIT,
978                  HT.FARRP(.HC.TEAM)
979                  AS FOLLOWS
980                  ----HTA (10)---- TEAM=******  HC=****** TGT.UNIT=****** FARRP=******
981               ALWAYS
```

```
            ALWAYS
         ALWAYS
      ENDLOOP
   ENDLOOP

   ''THE GROUND UNITS WILL NOW ATTEMPT TO DETECT THE HELICOPTERS

   LOOP FOR .J = 1 TO .NO.ENEMY.UNITS
   DO
      LOOP FOR EACH VISIBLE.UNIT IN UN.HC.LOS.LIST(.ENEMY.UNITS(.J))
      DO THIS
         FOR EACH ..HELICOPTER OF HT.MEMBER.LIST(.HC.TEAM)
         WITH VU.POINTER(VISIBLE.UNIT) = ..HELICOPTER
         FIND THE FIRST CASE
         IF NONE
      <------CYCLE
         OTHERWISE
         CALL HEL.RANGE.COMPUTE-------------------------------------------------------------------------------------------->(302)
         GIVEN
         ..HELICOPTER,
         .ENEMY.UNITS(.J)
         YIELDING
         .RANGE
         IF .RANGE = 0
            LET .RANGE = 1 ''HDM
            PRINT 2 LINES WITH
            FP.BATTLE(HT.FARRP(.HC.TEAM)),
            HT.FARRP(.HC.TEAM),
            .HC.TEAM,
            UNIT.NOS(.TARGET.UNIT) THUS
            --HEL$TARGET$A   WARNING: HELICOPTER CO-LOCATED WITH TARGET UNIT
            BTL=******, FARRP=******, TEAM=******, TGT UNIT.NOS= ******
         ALWAYS
         LET .TIME.TO.DETECT = RINF.C
         LOOP FOR EVERY UE.LINK OF
         UN.EQUIP.LIST(.ENEMY.UNITS(.J))
         WITH UE.QUANT(UE.LINK) GT 0
         DO THIS
            LOOP FOR EVERY .WEAPON OF UE.WEAPON.SET(UE.LINK)
            WITH WPN.QUANTITY(.WEAPON) GT 0 AND
            TW.NO.SENSORS(WPN.ID(.WEAPON)) GT 0 AND
            TW.MIN.RANGE(WPN.ID(.WEAPON)) LE .RANGE
            AND
            .RANGE LE TW.MAX.RANGE(WPN.ID(.WEAPON))
            DO
               LET .COUNT = UE.QUANT(UE.LINK)
               * WPN.QUANTITY(.WEAPON)
               * TW.NO.SENSORS(WPN.ID(.WEAPON))
               CALL SEARCH------------------------------------------------------------------------------------------------>(120)
               GIVEN
               WPN.ID(.WEAPON),
               .RANGE,
               RANDOM.F(RN.SEED),
               EQ.TE.PTR(HC.UE.ID(..HELICOPTER)),
               .COUNT
               YIELDING
               .DETECT.TIME
               LET .TIME.TO.DETECT = MIN.F(.TIME.TO.DETECT,
```

```
1040                  .DETECT.TIME)
1041               ENDLOOP
1042            ENDLOOP
1043            SCHEDULE_A HELO.ENGAGEMENT------------------------------------------------------------------------------------------>(376)
1044            GIVEN
1045             HC.TEAM,
1046            HT.FARRP(.HC.TEAM),
1047            ..HELICOPTER,
1048            .ENEMY.UNITS(.J)
1049            IN .TIME.TO.DETECT SECONDS
1050            IF HC.DEBUG = "YES"
1051               PRINT 1 LINE WITH ..HELICOPTER,
1052               .ENEMY.UNITS(.J),
1053               .TIME.TO.DETECT
1054               AS FOLLOWS
1055               ---- HTA (12) - .HELICOPTER=****** .ENEMY.UNIT=****** DETECT=****.***SEC
1056            ALWAYS
1057         ENDLOOP
1058      ENDLOOP
1059
1060 <----WAIT .ACQUIS.TIME SECONDS
1061
1062      IF HT.TERMINATOR(.HC.TEAM) = YES  OR
1063         HTA.REINFORCE.IND(HEL.TARGET.ACQUISITION) = YES
1064         LOOP FOR EVERY .HC IN HT.MEMBER.LIST(.HC.TEAM)
1065         DO
1066            LET HC.STATUS(.HC) = READY
1067         ENDLOOP
1068 <--------EXITPROCESS
1069      OTHERWISE
1070      IF HT.STATUS(.HC.TEAM) = RETURNING.FROM.BATTLE
1071   <----EXITLOOP
1072      OTHERWISE
1073
1074      ''  TERMINATE ALL HELICOPTER.FIRES                                                                              \1
1075      ''  TERMINATE THE SHOOT.OUTS AT HELICOPTERS                                                                     \1
1076      ''  CANCEL HELO.ENGAGEMENTS
1077      ''  CHANGE UN.HC.LOS.LIST AND HC.UN.LOS.LIST
1078      ''  REMOVE FIRING.TABLES FROM SHOOT.OUTS
1079      ''  REMOVE FIRING TABLES FROM THE HELICOPTERS
1080      ''  AFTER WAITING UNMASK                                                                                        \1
1081      LOOP FOR EVERY HELO.ENGAGEMENT IN EV.S(I.HELO.ENGAGEMENT)
1082      WITH HCEN.TEAM(HELO.ENGAGEMENT) = .HC.TEAM
1083      DO
1084         CANCEL_THIS HELO.ENGAGEMENT------------------------------------------------------------------------------------->(376)
1085         DESTROY THIS HELO.ENGAGEMENT
1086      ENDLOOP
1087      LOOP FOR EVERY HELICOPTER.FIRE IN EV.S(I.HELICOPTER.FIRE)
1088      WITH HF.TEAM(HELICOPTER.FIRE) = .HC.TEAM
1089      DO
1090         LET HF.REINFORCE.IND(HELICOPTER.FIRE) = YES
1091      ENDLOOP
1092      LOOP FOR EVERY SHOOT.OUT IN EV.S(I.SHOOT.OUT)
1093      WITH SO.HELICOPTER(SHOOT.OUT) GT 0 AND
1094      HC.BTL.TEAM(SO.HELICOPTER(SHOOT.OUT)) = .HC.TEAM
1095      DO
1096         LET SO.FIRING.TABLE(SHOOT.OUT) = 0
1097      ENDLOOP
```

```
1098        LOOP FOR .I = 1 TO .NO.ENEMY.UNITS
1099        DO  ''                                                                                    \1
1100           LOOP FOR EVERY .UEL OF UN.EQUIP.LIST(.ENEMY.UNITS(.I))
1101           DO  ''                                                                                 \1
1102              LOOP FOR EVERY .FT OF UE.TARGET.LIST(.UEL)
1103              DO  ''                                                                              \1
1104                 LOOP FOR EACH .HELICOPTER IN HT.MEMBER.LIST(.HC.TEAM)
1105                 WITH HC.UE.ID(.HELICOPTER) =
1106                 UE.ID(FT.TARGET.EQUIP(.FT))
1107                 DO
1108                    REMOVE THE .FT FROM UE.TARGET.LIST(.UEL)
1109                    DESTROY THE FIRING.TABLE CALLED .FT
1110                 ENDLOOP
1111              ENDLOOP
1112           ENDLOOP
1113        ENDLOOP
1114        LOOP FOR EVERY .UEL OF UN.EQUIP.LIST(FP.UNIT(HT.FARRP(.HC.TEAM)))
1115        DO
1116           LOOP FOR EVERY .FT OF UE.TARGET.LIST(.UEL)
1117           DO  ''                                                                                 \1
1118              REMOVE THE .FT FROM UE.TARGET.LIST(.UEL)
1119              DESTROY THE FIRING.TABLE CALLED .FT
1120           ENDLOOP
1121        ENDLOOP
1122        LOOP FOR EVERY .HELICOPTER IN HT.MEMBER.LIST(.HC.TEAM)
1123        DO
1124           LET HC.STATUS(.HELICOPTER) = RFADY
1125           LOOP FOR EVERY .VU IN HC.UN.LOS.LIST(.HELICOPTER)
1126           DO  ''                                                                                 \1
1127              LOOP FOR EVERY .VH IN UN.HC.LOS.LIST(VU.POINTER(.VU))
1128              WITH .HELICOPTER = VU.POINTER(.VH)
1129              DO ''                                                                               \1
1130                 REMOVE THE .VH FROM UN.HC.LOS.LIST(VU.POINTER(.VU))
1131                 DESTROY THE VISIBLE.UNIT CALLED .VH
1132              ENDLOOP
1133           ENDLOOP
1134        ENDLOOP
1135        IF UN.COLOR(FP.UNIT(HT.FARRP(.HC.TEAM))) = BLUE
1136  <-------WAIT UNIFORM.F(BL.MIN.MASK.TIME, BL.MAX.MASK.TIME,
1137           RN.SEED) SECONDS
1138        ELSE
1139  <-------WAIT UNIFORM.F(RD.MIN.MASK.TIME, RD.MAX.MASK.TIME,
1140           RN.SEED) SECONDS
1141        ALWAYS
1142        IF HT.TERMINATOR(.HC.TEAM) = YES
1143           OR HTA.REINFORCE.IND(HEL.TARGET.ACQUISITION) = YES
1144  <-------EXITPROCESS
1145        OTHERWISE
1146        IF HT.STATUS(.HC.TEAM) = RETURNING.FROM.BATTLE
1147     <------EXITLOOP
1148        OTHERWISE
1149     ENDLOOP
1150
1151     FOR EVERY .HC IN HT.MEMBER.LIST(.HC.TEAM)
1152     LET HC.STATUS(.HC) = READY
1153
1154     SCHEDULE_AN HC.DEPART.BATTLE------------------------------------------------------------------------------------------------------------>(375)
1155     GIVEN
```

```
1156     HT.FARRP(.HC.TEAM),
1157     .HC.TEAM,
1158     .ENEMY.UNITS(*) NOW  ''                                                    \1
1159
1160     IF HC.DEBUG = "YES"
1161        PRINT 1 LINE WITH HT.FARRP(.HC.TEAM), .HC.TEAM,
1162        .ENEMY.UNITS(*) AS FOLLOWS
1163        ----HTA (15)---- HT.FARRP=****** .HC.TEAM=****** .ENEMY.UNITS=******
1164     ALWAYS
1165
1166 <--EXITPROCESS
1167 ENDPROCESS
```

```
1168 PROCESS HOW.REPAIR                                                                                    ''                                                                                
```

```
1216 PROCESS MINE.ASSESS                                                                                    ''                P010
1217    GIVEN
1218    .UNIT,
1219    .MINE.KV.ID,
1220    .BATTLE.ENDED
1221
1222    ADD 1 TO ANAL.CTR(176,1)  ''                                                                                    \DYN_ANAL
1223    ''THIS PROCESS COMPUTES LOSSES DUE TO MINES WHEN A UNIT IS
1224    ''IN A BATTLE
1225
1226    NORMALLY MODE IS INTEGER
1227    DEFINE .FRAC.CASUALTIES AS A REAL VARIABLE
1228
1229    LET .MA = MINE.ASSESS
1230    IF UN.COLOR(.UNIT) = BLUE
1231       LET .KILLER.COLOR = RED
1232    ELSE
1233       LET .KILLER.COLOR = BLUE
1234    ALWAYS
1235
1236    IF MF.DEBUG = TRUE
1237       PRINT 1 LINE WITH
1238       .MA,
1239       .UNIT,
1240       TIME.V THUS
1241       = = =MINE ASSESS ****** STARTED FOR UNIT= **** AT **.*** HRS
1242    ALWAYS
1243
1244    LOOP
1245    UNTIL MA.SET(.MA) IS EMPTY
1246    DO
1247       REMOVE THE FIRST .LINK FROM THE MA.SET(.MA)
1248       LET .UE.LINK = MA.UE.LINK(.LINK)
1249       LET .CASUALTIES = MA.CASUALTIES(.LINK)
1250       DESTROY THE MA.LINK CALLED .LINK
1251       LET .VICTIM = EQ.KV.ID(UE.ID(.UE.LINK))
1252       LET .CASUALTIES = MIN.F(.CASUALTIES, UE.QUANT(.UE.LINK))
1253       IF UE.QUANT(.UE.LINK) = 0 OR .CASUALTIES = 0
1254    <------CYCLE
1255       OTHERWISE
1256       LET .FRAC.CASUALTIES = .CASUALTIES / UE.QUANT(.UE.LINK)
1257       IF MA.BATTLE.ENDED(.MA) = NO AND N.SO.LIST(.UE.LINK) LE 0
1258          ''START A SHOOT OUT FOR EACH PIECE OF EQUIPMENT
1259          LOOP
1260          FOR .I = 1 TO UE.QUANT(.UE.LINK)
1261          DO
1262             ACTIVATE_A SHOOT.OUT------------------------------------------------------------------------------------------------------------>(493)
1263             CALLED .SO NOW ''                                                                                                                \1
1264             LET FIRING.EQUIP(.SO) = .UE.LINK
1265             LET FIRER.UNIT(.SO) = .UNIT
1266             LET DROP.DEAD.INDICATOR(.SO) = NO
1267             FILE .SO IN THE SO.LIST(.UE.LINK)
1268          ENDLOOP
1269 <--------WAIT .1 SECONDS
1270       ALWAYS
1271
1272       IF MA.BATTLE.ENDED(.MA) = NO
1273          LET .NUM.KILLED = 0
```

```
1274          LET .FLAG = NO
1275          LOOP
1276          UNTIL .NUM.KILLED GE .CASUALTIES OR
1277          UE.QUANT(.UE.LINK) LE 0 OR
1278          .FLAG = YES
1279          DO
1280             LET .FLAG = YES
1281             LOOP
1282             FOR EVERY .SO IN SO.LIST(.UE.LINK)
1283             WITH DROP.DEAD INDICATOR(.SO) = NO
1284             UNTIL .NUM.KILLED GE .CASUALTIES
1285             DO
1286                LET .FLAG = NO
1287                IF .FRAC.CASUALTIES GT RANDOM.F(RN.SEED)
1288                   ADD 1 TO .NUM.KILLED
1289                   SUBTRACT 1 FROM UE.QUANT(.UE.LINK)
1290                   IF UN.PTR(.UNIT) GT 0 AND
1291                      UE.CRITICAL.EQUIP.INDIC(.UE.LINK) = TRUE
1292                      SUBTRACT 1 FROM MU.CRIT.NO(UN.PTR(.UNIT))
1293                   ALWAYS
1294                   IF .MINE.KV.ID GT 0 AND .VICTIM GT 0
1295                      ADD 1 TO
1296                      KV.SCORE(.KILLER.COLOR,
1297                      .MINE.KV.ID, .VICTIM)
1298                   ALWAYS
1299                   INTERRUPT SHOOT.OUT CALLED .SO------------------------------------------------------------>(493)
1300                   LET DROP.DEAD.INDICATOR(.SO) = YES
1301                   LET TIME.A(.SO) = 1/60. ''HOURS
1302                   RESUME SHOOT.OUT CALLED .SO---------------------------------------------------------------->(493)
1303                ALWAYS
1304             ENDLOOP
1305          ENDLOOP
1306       ELSE
1307          ''THE BATTLE HAS ENDED - DESTROY EQUIPMENT HERE
1308          LET .NUM.KILLED = MIN.F(.CASUALTIES, UE.QUANT(.UE.LINK))
1309          SUBTRACT .NUM.KILLED FROM UE.QUANT(.UE.LINK)
1310          IF UN.PTR(.UNIT) GT 0 AND
1311             UE.CRITICAL.EQUIP.INDIC(.UE.LINK) = TRUE
1312             SUBTRACT .NUM.KILLED FROM MU.CRIT.NO(UN.PTR(.UNIT))
1313          ALWAYS
1314          IF .MINE.KV.ID GT 0 AND .VICTIM GT 0
1315             ADD .NUM.KILLED TO
1316             KV.SCORE(.KILLER.COLOR, .MINE.KV.ID, .VICTIM)
1317          ALWAYS
1318       ALWAYS
1319       IF MF.DEBUG = YES
1320          PRINT 1 LINE WITH .MA,
1321          .UNIT,
1322          UE.ID(.UE.LINK),
1323          .NUM.KILLED,
1324          .CASUALTIES,
1325          UE.QUANT(.UE.LINK) THUS
1326          = = =MINE ASSESS ******,UNIT=*****,EQ=****,KILLED=****,EST=****,REM=****
1327       ALWAYS
1328       IF ANALYSIS(6) GT 0 AND .NUM.KILLED GT 0
1329          FOR EVERY EQUIPMENT CALLED .EQ
1330          WITH EQ.KV.ID(.EQ) = .MINE.KV.ID
1331          FIND THE FIRST CASE
```

```
1332          IF NONE
1333             LET .EQ = 0
1334          ALWAYS
1335          CALL OUTPUT.ATTRITION----------------------------------------------------------------------------------------->(611)
1336          GIVEN
1337          .UNIT,
1338          .EQ,
1339          UE.ID(.UE.LINK),
1340          .NUM.KILLED,
1341          "FASCAM",
1342          MA.MSN(.MA)
1343       ALWAYS
1344    ENDLOOP
1345
1346    IF MF.DEBUG = TRUE
1347       PRINT 1 LINE WITH
1348       .MA,
1349       .UNIT,
1350       TIME.V THUS
1351       = = =MINE ASSESS •••••• ENDED FOR UNIT •••• AT ••.••• HRS
1352    ALWAYS
1353
1354 <—EXITPROCESS
1355 ENDROUTINE
```

```
1356 PROCESS REMOTE.PILOT.VEHICLE                                                    ''                                                P012
1357    ADD 1 TO ANAL.CTR(177,1)  ''                                                                                               \DYN_ANAL
1358    NORMALLY MODE IS INTEGER
1359    DEFINE RPV TO MEAN PROCESS.V ''                                                                                            CHG\01
1360    DEFINE TRANSMIT TO MEAN WAIT
1361    DEFINE SEARCH TO MEAN WAIT
1362    DEFINE FLY TO MEAN WAIT
1363    DEFINE TURN TO MEAN WAIT
1364    DEFINE PREPARE TO MEAN WAIT
1365    DEFINE UNIT, GROUPING, SIDE, SENSOR.TYPE AS INTEGER VARIABLES
1366    DEFINE LEG.SLOPE, LEG.Y.INTERCEPT , LEG.LENGTH,
1367    X.PERPENDICULAR, Y.PERPENDICULAR,
1368    PERPENDICULAR.DIST, SEARCH.DIST, OLD.DIST,
1369    RETURN.TIME,APPROACH.TIME,APPROACH.DISTANCE,
1370    BACK.DIST, FORWARD.DIST AS REAL VARIABLES
1371    '' IF DEBUG = TRUE,
1372       '' PRINT 1 LINE WITH RPV THUS
1373       '' = = = REMOTE.PILOT.VEHICLE  RPV = ******  = = =
1374    '' ENDIF
1375    LET LINK = RPV.US.LINK(RPV)
1376    LET UNIT = US.UNIT(LINK)
1377    LET SIDE = UN.COLOR(UNIT)
1378    LET MODEL = US.MODEL(LINK)
1379    LET HALF.WIDTH = MRPV.HALF.COV.WIDTH(MODEL)
1380    LET SENSOR.TYPE = US.SENSOR.TYPE(LINK)
1381    LET VELOCITY = MRPV.VELOCITY(MODEL)
1382    LET AIR.STRIP = UNIT
1383    IF SIDE = RED
1384       LET ENEMY = BLUE
1385    ELSE
1386       LET ENEMY = RED
1387    ENDIF
1388    CREATE A TARGET.REPORT CALLED TARGET
1389    LET RPV.CURRENT.TR(RPV) = TARGET
1390    PREPARE UNIFORM.F(MRPV.MIN.PREP(MODEL), MRPV.MAX.PREP(MODEL),1) MINUTES
1391    LET FL.Y.START = RPV.Y.START(RPV)
1392    LET FL.X.START = RPV.X.START(RPV)
1393    LET APPROACH.DISTANCE = SQRT.F((UN.X.COORD(AIR.STRIP)-FL.X.START)**2 +    ''                                                \OPTIMIZE
1394    (UN.Y.COORD(AIR.STRIP)-FL.Y.START)**2)
1395    LET APPROACH.TIME = APPROACH.DISTANCE/VELOCITY
1396    LET RETURN.TIME = APPROACH.TIME
1397    LET END.MINUTE = TIME.V + MINUTES.V + MRPV.MAX.ALOFT.TIME(MODEL)
1398    LET RETURN.MINUTE = END.MINUTE - RETURN.TIME
1399    FLY APPROACH.TIME MINUTES
1400    LOOP FOR EACH LEG IN RPV.FLIGHT.LEG.LIST(RPV)
1401    DO THIS  ''                                                                                                                \1
1402       LET DELTA.Y = FL.Y.END(LEG) - FL.Y.START
1403       LET DELTA.X = FL.X.END(LEG) - FL.X.START
1404       LET LEG.SLOPE = DELTA.Y/DELTA.X
1405       LET LEG.Y.INTERCEPT = FL.Y.START - LEG.SLOPE * FL.X.START
1406       LET LEG.LENGTH = SQRT.F(DELTA.X ** 2 + DELTA.Y ** 2)
1407       LOOP FOR EACH GROUPING
1408       DO THIS  ''                                                                                                             \1
1409          LOOP FOR EACH UN IN UNIT.SET(ENEMY,GROUPING)
1410          DO THIS  ''                                                                                                          \1
1411             LET X.PERPENDICULAR = (UN.Y.COORD(UN) + UN.X.COORD(UN)/LEG.SLOPE -
1412             LEG.Y.INTERCEPT) / (LEG.SLOPE + 1./LEG.SLOPE)
1413             LET Y.PERPENDICULAR = LEG.SLOPE*X.PERPENDICULAR+LEG.Y.INTERCEPT
```

```
1414            LET PERPENDICULAR.DIST = 10. * SQRT.F((X.PERPENDICULAR -
1415            UN X.COORD(UN))**2.+(Y.PERPENDICULAR - UN.Y.COORD(UN))**2 )
1416            IF PERPENDICULAR.DIST > ST.MAX.RANGE(SENSOR.TYPE)
1417         <------CYCLE
1418            ENDIF
1419            LET BACK.DIST = SQRT.F((X.PERPENDICULAR-FL.X.START) ** 2 +
1420            (Y.PERPENDICULAR-FL.Y.START) ** 2 )
1421            LET FORWARD.DIST = SQRT.F((X.PERPENDICULAR-FL.X.END(LEG)) ** 2 +
1422            (Y.PERPENDICULAR-FL.Y.END(LEG)) ** 2 )
1423            IF BACK.DIST+FORWARD.DIST > LEG.LENGTH
1424         <------CYCLE
1425            OTHERWISE
1426            CALL SEARCH.COVERAGE GIVEN UN, HALF.WIDTH, PERPENDICULAR.DIST, LEG.SLOPE,------------------------------ ->(343)
1427            LEG.Y.INTERCEPT, AND UN.RADIUS(UN)    YIELDING FRACT.COVERED
1428            IF FRACT.COVERED = 0.0
1429         <------CYCLE
1430            OTHERWISE
1431            CREATE AN RPV.DET.CANDIDATE CALLED CANDIDATE
1432            LET RPV.DC.UNIT(CANDIDATE) = UN
1433            LET RPV.DC.LEG.DIST(CANDIDATE) = BACK.DIST
1434            LET RPV.DC.DIST(CANDIDATE) = PERPENDICULAR.DIST
1435            FILE CANDIDATE IN RPV.CAND.DET.LIST(RPV)
1436         ENDLOOP
1437      ENDLOOP
1438      LET OLD.DIST = 0.0
1439      LOOP FOR EACH CANDIDATE IN RPV.CAND.DET.LIST(RPV)
1440      DO THIS ''                                                                                          \1
1441         IF TIME.V * MINUTES.V >= RETURN.MINUTE
1442      <------CYCLE
1443         OTHERWISE
1444         REMOVE CANDIDATE FROM RPV.CAND.DET.LIST(RPV)
1445         LET SEARCH.DIST = RPV.DC.LEG.DIST(CANDIDATE) - OLD.DIST
1446         LET OLD.DIST = RPV.DC.LEG.DIST(CANDIDATE)
1447         SEARCH  SEARCH.DIST/VELOCITY MINUTES
1448         CALL RPV.DETECTION GIVEN RPV, UN, FRACT.COVERED, CANDIDATE, TARGET------------------------------------------>(229)
1449         IF TR.DET.LIST(TARGET) IS EMPTY
1450      <------CYCLE
1451         OTHERWISE
1452         DESTROY THE RPV.DET.CANDIDATE CALLED CANDIDATE
1453         TRANSMIT UNIFORM.F(REAL.F(ST.MIN.XMIT(SENSOR.TYPE)) ,
1454         REAL.F(ST.MAX.XMIT(SENSOR.TYPE)) , 1)/10. MINUTES
1455         LET TR.RECVD.TIME(TARGET) = TIME.V
1456         LET TR.ABORT.TIME(TARGET) = TIME.V + .5
1457         ACTIVATE THE TARGET.REPORT CALLED TARGET NOW------------------------------------------------------------>(470)
1458         IF TR.PGM.STATUS(TARGET) = TRUE
1459 <----------SUSPEND
1460         ENDIF
1461         IF TIME.V * MINUTES.V < RETURN.MINUTE
1462            CREATE A TARGET.REPORT CALLED TARGET
1463            LET RPV.CURRENT.TR(RPV) = TARGET
1464         ENDIF
1465      ENDLOOP
1466      LET FL.Y.START = FL.Y.END(LEG)
1467      LET FL.X.START = FL.X.END(LEG)
1468   ENDLOOP
1469   FLY RETURN.TIME MINUTES
1470   LOOP UNTIL RPV.FLIGHT.LEG.LIST(RPV) IS EMPTY
1471   DO THIS  ''                                                                                           \1
```

```
1472        REMOVE LEG FROM RPV.FLIGHT.LEG.LIST(RPV)
1473        DESTROY THE FLIGHT.LEG CALLED LEG
1474     ENDLOOP
1475     LOOP UNTIL RPV.CAND.DET.LIST(RPV) IS EMPTY
1476     DO THIS   ''                                                    \1
1477        REMOVE CANDIDATE FROM RPV.CAND.DET.LIST(RPV)
1478        DESTROY THE RPV.DET.CANDIDATE CALLED CANDIDATE
1479     ENDLOOP
1480     LET US.STATUS(LINK) = HOLD
1481  ENDPROCESS
```

```
1482 PROCESS TARGET.REPORT                                                    ''                                                    P013
1483
1484    ADD 1 TO ANAL.CTR(178,1)  ''                                                                                              \DYN_ANAL
1485    ''THIS PROCESS REPRESENTS THE PROCESSING OF A TARGET REPORT OR
1486    ''A REQUEST FOR FIRE AT THE FDC
1487
1488    NORMALLY MODE IS INTEGER
1489    DEFINE TARGET      TO MEAN PROCESS.V ''                                                                                   CHG\01
1490    DEFINE FDC         TO MEAN TR.FDC(TARGET)
1491    DEFINE DUPLICATE   TO MEAN ETR.DUPLICATE(TR.EX.TGT.REPORT(TARGET))
1492    DEFINE DUR         TO MEAN ETR.DUR(TR.EX.TGT.REPORT(TARGET))
1493    DEFINE MAX.PREP    TO MEAN ETR.MAX.PREP(TR.EX.TGT.REPORT(TARGET))
1494    DEFINE NEW.STOP    TO MEAN ETR.NEW.STOP(TR.EX.TGT.REPORT(TARGET))
1495    DEFINE RFAF        TO MEAN ETR.RFAF(TR.EX.TGT.REPORT(TARGET))
1496    DEFINE START.TIME  TO MEAN ETR.START.TIME(TR.EX.TGT.REPORT(TARGET))
1497    DEFINE TOT.FOLLOW  TO MEAN ETR.TOT.FOLLOW(TR.EX.TGT.REPORT(TARGET))
1498
1499    CREATE AN EX.TGT.REPORT CALLED TR.EX.TGT.REPORT(TARGET)
1500
1501
1502    IF DEBUG = TRUE AND
1503       TR.PGM.STATUS(TARGET)=TRUE
1504       PRINT 1 LINE WITH TARGET AND TIME.V THUS
1505       $#$#$#$# TARGET REPORT FOR TARGET ********* AT TIME **.*****
1506    ALWAYS
1507
1508    IF PROCESS IS EXTERNAL,
1509       PERFORM TR.INPUT------------------------------------------------------------------------------------------------------>(590)
1510       GIVEN
1511       TARGET
1512    ALWAYS
1513
1514    IF TR.CEP LE 0  ''                                                                                                        CHG\33
1515       IF TARGET IS IN AN FO.TGT.RPT.LIST
1516          REMOVE TARGET FROM FO.TGT.RPT.LIST(TR.SENSOR.ID(TARGET))
1517       ALWAYS
1518       DESTROY THE EX.TGT.REPORT
1519       CALLED TR.EX.TGT.REPORT(TARGET)
1520 <-----EXITPROCESS
1521    OTHERWISE
1522
1523    IF TIME.V > TR.ABORT.TIME(TARGET) ''THIS TARGET IS TOO OLD
1524       IF DEBUG=TRUE,
1525          PRINT 1 LINE THUS
1526          = = = TARGET.REPORT = LINE 23 = TIME.V > ABORT TIME
1527       ALWAYS
1528       CALL FINISH.COMPUTATION----------------------------------------------------------------------------------------------->(334)
1529       GIVEN
1530       TARGET,
1531       FDC
1532       IF TARGET IS IN AN FO.TGT.RPT.LIST
1533          REMOVE TARGET FROM FO.TGT.RPT.LIST(TR.SENSOR.ID(TARGET))
1534       ALWAYS
1535       DESTROY THE EX.TGT.REPORT
1536       CALLED TR.EX.TGT.REPORT(TARGET)
1537 <-----EXITPROCESS
1538    OTHERWISE
1539
```

```
1540    PERFORM TARGET.ANALYSIS ''------------------------------------------------------------------------------------------------ \DEBUG>(239)
1541    GIVEN TARGET, 1  ''                                                                                                          CHG\21
1542     ▸ TR.EST.TU(TARGET) IS ZERO OR TR.SENSOR.ID(TARGET) IS ZERO, ''                                                  CHG\21  \DEBUG
1543       IF TARGET IS IN AN FO.TGT.RPT.LIST
1544          REMOVE TARGET FROM FO.TGT.RPT.LIST(TR.SENSOR.ID(TARGET))
1545       ALWAYS
1546 <-----RETURN
1547    OTHERWISE
1548    IF TR.TGT.UNIT(TARGET) > 0
1549       ''THIS IS NOT AN EXTERNAL TARGET REPORT                                                                                   \1
1550       ''ENQ FOR THE FDC
1551       IF TR.TOT.STATUS(TARGET) NE 2 ''THIS IS NOT A TOT FOLLOW-ON
1552          ''ONLY CHECK HE AND ICM MISSIONS FOR DUPLICATIONS.
1553          IF TR.MISSION.TYPE(TARGET) NE "FASCAM" AND
1554             TR.MISSION.TYPE(TARGET) NE "ILLUM" AND
1555             TR.MISSION.TYPE(TARGET) NE "SMOKE"
1556             LET TR.MISSION.TYPE(TARGET) = "HE.ICM"
1557             IF TR.MIL.WORTH(TARGET) = 0  OR
1558                TR.MIL.WORTH(TARGET) = 2000  OR  ''NEW TR
1559                TR.MIL.WORTH(TARGET) = 1999  OR
1560                TR.MIL.WORTH(TARGET) = 2001  ''PGM
1561                ''WHERE 0 MEANS IT WAS TARGET OF OPPORTUNITY AND
1562                ''2000  MEANS IT WAS GENERATED IN SMALL BATTLE
1563                PERFORM TARGET.ANALYSIS-------------------------------------------------------------------------------------->(239)
1564                GIVEN
1565                TARGET,
1566                0  ''                                                                                                  CHG\21  \DEBUG
1567                IF TR.EST.TU(TARGET) = 0
1568                   ''THE TYPE OF UNIT IS NOT IDENTIFIED
1569                   IF DEBUG=TRUE,
1570                      PRINT 1 LINE THUS
1571                      = = TARGET REPORT = TARGET ANALYSIS CAN'T IDENTIFY TARGET ==
1572                   ALWAYS
1573
1574                   CALL FINISH.COMPUTATION------------------------------------------------------------------------------------->(334)
1575                   GIVEN
1576                   TARGET,
1577                   FDC
1578                   IF TARGET IS IN AN FO.TGT.RPT.LIST
1579                      REMOVE TARGET FROM
1580                      FO.TGT.RPT.LIST(TR.SENSOR.ID(TARGET))
1581                   ALWAYS
1582                   DESTROY THE EX.TGT.REPORT
1583                   CALLED TR.EX.TGT.REPORT(TARGET)
1584 <-----------------EXITPROCESS
1585                OTHERWISE
1586             ALWAYS
1587             CALL CHK.COMP.TR-------------------------------------------------------------------------------------------->(168)
1588             GIVEN
1589             TARGET,
1590             FDC,
1591             0
1592             YIELDING
1593             DUPLICATE
1594             IF DUPLICATE = TRUE
1595                IF DEBUG=TRUE,
1596                   PRINT 1 LINE THUS
1597                   = = = TARGET REPORT = DUPLICATE TR BEING FIRED
```

```
1598                ALWAYS
1599
1600                IF TARGET IS IN AN FO.TGT.RPT.LIST
1601                   REMOVE TARGET FROM
1602                   FO.TGT.RPT.LIST(TR.SENSOR.ID(TARGET))
1603                ALWAYS
1604                ''THIS TARGET REPORT IS A DUPLICATE
1605                DESTROY THE EX.TGT.REPORT
1606                CALLED TR.EX.TGT.REPORT(TARGET)
1607 <--------------EXITPROCESS
1608             OTHERWISE
1609             ''CHECK FOR DUPLICATE TARGET REPORTS
1610             ''AMONG THOSE AWAITING PROCESSING
1611             CALL CHK.FD.TR-------------------------------------------------------------------------------------->(169)
1612             GIVEN
1613             TARGET,
1614             FDC,
1615             0
1616             YIELDING
1617             DUPLICATE
1618             IF DUPLICATE = TRUE
1619                IF DEBUG=TRUE,
1620                   PRINT 1 LINE THUS
1621                   = = = TARGET REPORT = DUPLICATE TR AWAITING FIRE
1622                ALWAYS
1623
1624                IF TARGET IS IN AN FO.TGT.RPT.LIST
1625                   REMOVE TARGET FROM
1626                   FO.TGT.RPT.LIST(TR.SENSOR.ID(TARGET))
1627                ALWAYS
1628                DESTROY THE EX.TGT.REPORT
1629                CALLED TR.EX.TGT.REPORT(TARGET)
1630 <--------------EXITPROCESS
1631             OTHERWISE
1632
1633          ALWAYS
1634          FILE THE TARGET IN THE FD.TR.QUEUE(FDC)
1635          CALL FDC.TR.ENQ------------------------------------------------------------------------------------->(333)
1636          GIVEN
1637          TARGET,
1638          FDC
1639          IF TR.FDC.STATUS(TARGET) = HOLD
1640 <--------WAIT 31 MINUTES
1641          ALWAYS
1642
1643          IF TIME.V >= TR.ABORT.TIME(TARGET) ''TARGET REPORT IS TOO OLD
1644
1645             IF DEBUG = TRUE  AND
1646                TR.PGM.STATUS(TARGET)=TRUE
1647                PRINT 1 LINE WITH TARGET AND TIME.V THUS
1648                $#$#$# TARGET ******* TOO OLD, DESTROYED AT **.*****
1649             ALWAYS
1650
1651             CALL FINISH.COMPUTATION---------------------------------------------------------------------------->(334)
1652             GIVEN
1653             TARGET,
1654             FDC
1655             IF TARGET IS IN AN FO.TGT.RPT.LIST
```

```
1656                REMOVE TARGET FROM
1657                FO.TGT.RPT.LIST(TR.SENSOR.ID(TARGET))
1658             ALWAYS
1659             DESTROY THE EX.TGT.REPORT
1660             CALLED TR.EX.TGT.REPORT(TARGET)
1661 <-----------EXITPROCESS
1662          OTHERWISE
1663
1664          IF TR.TOT.STATUS(TARGET) = TRUE
1665             ''FIND OUT THE EARLIEST TIME  THE TOT CAN BE FIRED AND
1666             ''THE LATEST IT CAN BE STARTED AND STILL FINISH ON TIME
1667             ''(BY TR ABORT TIME)
1668             LOOP FOR EACH TYPE.BTRY,
1669             DO
1670                COMPUTE MAX.PREP AS MAXIMUM
1671                OF TB.MAX.PREP(TYPE.BTRY)/6.   ''HOURS * 100
1672             ENDLOOP
1673             LOOP FOR EACH CATEGORY
1674             FOR EACH TYPE.BTRY
1675             DO
1676                ''UTILIZE FEBA.BAND FUNCTION IN NEXT COMPUTATION ------------------------------------------ \1>(634)
1677                COMPUTE DUR AS MAXIMUM OF
1678                (REAL.F(CDT.MAX.VOLS(CATEGORY,
1679                FEBA.BAND(TARGET), TYPE.BTRY)) /
1680                TB.SUST.FIRE.RATE(TYPE.BTRY))   ''IN MINUTES
1681             ENDLOOP
1682             LET DUR = DUR * 10.0/6.0   ''CONVERTS TO HOURS * 100
1683             CALL FIND.START.TIME-------------------------------------------------------------------------->(194)
1684             GIVEN
1685             INT.F(TIME.V*100)+MAX.PREP,
1686             INT.F(TR.ABORT.TIME(TARGET)*100)-INT.F(DUR),
1687             INT.F(DUR),
1688             FDC
1689             YIELDING
1690             START.TIME
1691             LET START.TIME = START.TIME * 100.
1692          ALWAYS
1693       ALWAYS
1694
1695       IF TR.PGM.STATUS(TARGET) GE TRUE,
1696          ''ATTEMPT CLGP ASSIGNMENT BUT NEVER ON TOT
1697          PERFORM PGM.MSN.ASGN------------------------------------------------------------------------------>(208)
1698          GIVEN
1699          TARGET,
1700          FDC
1701          IF TR.FM.LIST(TARGET) IS EMPTY,
1702             ''NO PGM MISSION WAS ASSIGNED
1703             IF DEBUG = TRUE
1704                PRINT 1 LINE WITH TARGET THUS
1705                $#$#$#$# TARGET ******** CONVERTED TO NON-PGM MSN (155TR)
1706             ALWAYS
1707
1708             LET TR.MIL.WORTH(TARGET) = 0
1709             CALL TARGET.ANALYSIS------------------------------------------------------------------------->(239)
1710             GIVEN
1711             TARGET,
1712             0  ''                                                                    CHG\21  \DEBUG
1713             IF TR.EST.TU(TARGET) = 0
```

```
1714                ''THE TYPE OF UNIT IS NOT IDENTIFIED
1715
1716                IF DEBUG=TRUE,
1717                   PRINT 1 LINE THUS
1718                   = = TARGET REPORT = TARGET ANALYSIS CAN'T IDENTIFY TARGET ==
1719                ALWAYS
1720
1721                CALL FINISH.COMPUTATION------------------------------------------------------------------------------------------>(334)
1722                GIVEN
1723                TARGET,
1724                FDC
1725                IF TARGET IS IN AN FO.TGT.RPT.LIST
1726                   REMOVE TARGET FROM
1727                   FO.TGT.RPT.LIST(TR.SENSOR.ID(TARGET))
1728                ALWAYS
1729                DESTROY THE EX.TGT.REPORT
1730                CALLED TR.EX.TGT.REPORT(TARGET)
1731 <--------------EXITPROCESS
1732             OTHERWISE
1733
1734             IF DEBUG = TRUE
1735                PRINT 1 LINE WITH TR.MIL.WORTH(TARGET) THUS
1736                $#$#  NEW MIL.WORTH = *****
1737             ALWAYS
1738
1739             ''CONVERT TO NON PGM MISSION
1740             LET TR.PGM.STATUS(TARGET) = FALSE
1741             IF TR.SENSOR.TYPE(TARGET) = "RPV"
1742                RESUME REMOTE.PILOT.VEHICLE--------------------------------------------------------------------------------->(467)
1743                CALLED TR.SENSOR.ID(TARGET)
1744             ALWAYS
1745          ELSE
1746             IF TR.PGM.STATUS(TARGET) = TRUE
1747                LET TR.MISSION.TYPE(TARGET) = "PGM"
1748             ELSE
1749                LET TR.MISSION.TYPE(TARGET) = "SDM"
1750             ALWAYS
1751          ALWAYS
1752       ALWAYS
1753       '' THIS IS WHERE AN 'ELSE' IS ADDED AND THE CODE IS PLACED
1754       '' TO HANDLE SCHEDULED MISSIONS ( WHERE TR.TGT.UNIT = 0) AND
1755       '' FOLLOW WITH 'TR.START.TIME > TIME.V'                                                                              \1
1756       '' ALSO AN 'ALWAYS' IS TAGGED ON THE END
1757    ALWAYS
1758
1759    IF TR.PGM.STATUS(TARGET)=FALSE,
1760       ''PROCEED WITH NON PGM MISSION ASSIGNMENT                                                                            \1
1761       IF TR.REM.EFFECTS(TARGET) IS ZERO,
1762          ''THIS IS A NEW TR. DETERMINE EFFECTS REQUIRED.
1763          LET SIDE = UN.COLOR(TR.REP.UNIT(TARGET))
1764          CALL FD.EFFECTS.REQ-------------------------------------------------------------------------------------------->(187)
1765          GIVEN
1766          TARGET,
1767          SIDE
1768          YIELDING
1769          TR.REQ.EFFECTS(TARGET)
1770          LET TR.REM.EFFECTS(TARGET) = TR.REQ.EFFECTS(TARGET)*100
1771
```

```
1772          IF DEBUG=TRUE,
1773             PRINT 1 LINE WITH TARGET,TR.REQ.EFFECTS(TARGET),
1774             TR.REM.EFFECTS(TARGET)   THUS
1775             = TAR.REP CALLED FD.EFF = TARGET = ******* REQ.EFF = ******* REM.EFF = ********
1776          ALWAYS
1777
1778       ALWAYS
1779       IF TIME.V > TR.ABORT.TIME(TARGET) ''THIS TARGET IS TOO OLD
1780          IF DEBUG=TRUE,
1781             PRINT 1 LINE THUS
1782             = = = TARGET.REPORT = LINE 146 = TIME.V > ABORT TIME
1783          ALWAYS
1784
1785          '' AND IF IT WAS A MOVING TARGET IT'S ABORT TIME
1786          '' WAS RE-COMPUTED IN THE EFFECTS REQUIRED ROUTINE
1787
1788          CALL FINISH.COMPUTATION------------------------------------------------------------------------------------------>(334)
1789          GIVEN
1790          TARGET,
1791          FDC
1792          IF TARGET IS IN AN FO.TGT.RPT.LIST
1793             REMOVE TARGET FROM FO.TGT.RPT.LIST(TR.SENSOR.ID(TARGET))
1794          ALWAYS
1795          DESTROY THE EX.TGT.REPORT
1796          CALLED TR.EX.TGT.REPORT(TARGET)
1797 <--------EXITPROCESS
1798       OTHERWISE
1799
1800       LOOP FOR EACH BN.LINK IN FD.BN.LIST(FDC)
1801       UNTIL TR.REM.EFFECTS(TARGET) LE 0,
1802       DO THIS
1803          ''CONTINUE TO ADD BATTALIONS TO TRY TO SATISFY THE
1804          ''REQUIRED EFFECTS (WHEN REMAINING EFFECTS ARE ZERO)
1805          ''COMPUTE PROCESSING TIME TO COMPUTE FIRE MISSIONS FOR
1806          ''ALL THE BATTERIES IN THIS BATTALION
1807
1808 <--------WORK UNIFORM.F( REAL.F(FD.MIN.TIME(FDC)),
1809          REAL.F(FD.MAX.TIME(FDC)), 1) / 10.  MINUTES
1810
1811          IF TIME.V GT TR.ABORT.TIME(TARGET) ''TR HAS BEEN ABORTED
1812             LET TR.REM.EFFECTS(TARGET) = 0
1813          ALWAYS
1814          PERFORM FA.BN.ASGN----------------------------------------------------------------------------------------------->(182)
1815          GIVEN
1816          TARGET,
1817          FB.BN(BN.LINK),
1818          INT.F(DUR)
1819       ENDLOOP
1820
1821       IF TR.TOT.STATUS(TARGET) = FALSE
1822          ''TOT TARGETS ARE NOT PASSED TO HIGHER FD
1823          IF TR.REM.EFFECTS(TARGET) > 0 AND FD.FDC(FDC) IS NOT ZERO,
1824             ''CAN'T MEET REQUIRED EFFECTS WITH THIS FDC'S ASSETS AND
1825             ''THERE EXISTS A CHANNEL TO HIGHER FDC FOR SENDING A RFAF
1826             ''GENERATE A REQUEST FOR ADDITIONAL FIRE
1827             IF DEBUG=TRUE,
1828                PRINT 1 LINE THUS
1829                = = = TARGET REPORT PASSED TO HIGHER FDC
```

```
1830            ALWAYS
1831
1832            ACTIVATE_A TARGET.REPORT CALLED RFAF NOW------------------------------------------------------------------------------------------>(470)
1833            ''EVERYTHING ABOUT THE RFAF IS THE SAME AS THIS TARGET
1834            ''REPORT EXCEPT THE TR FDC AND THE TARGET REPORT NUMBER
1835            PERFORM COPY---------------------------------------------------------------------------------------------------------------->(172)
1836            GIVEN
1837            TARGET,
1838            RFAF
1839            LET TR.FDC(RFAF) = FD.FDC(FDC)
1840            IF TARGET IS IN AN FO.TGT.RPT.LIST
1841               FILE RFAF IN FO.TGT.RPT.LIST(TR.SENSOR.ID(TARGET))
1842               ''THE FO ALSO RESPONSIBLE FOR THE RESULTANT RFAF'S
1843            ALWAYS
1844         ALWAYS
1845      ELSE
1846         IF TR.TOT.STATUS(TARGET) = TRUE
1847            ''ENTER THE TR IN THE FDC SCHEDULED FIRE RECORDS
1848            LET NEW.STOP = START.TIME  + INT.F(DUR)
1849            CALL FILE.FD.SCHD----------------------------------------------------------------------------------------------------------->( 63)
1850            GIVEN
1851            FDC,
1852            START.TIME,
1853            NEW.STOP,
1854            N.TR.FM.LIST(TARGET)
1855            IF TR.REM.EFFECTS(TARGET) > 0
1856               ''GENERATE A TOT FOLLOW-ON TARGET REPORT USING ONLY
1857               ''     THOSE BTRYS ALREADY INVOLVED IN THE TOT
1858               ACTIVATE_A TARGET.REPORT CALLED TOT.FOLLOW------------------------------------------------------------------------------>(470)
1859               AT TR.START.TIME(TARGET)
1860               PERFORM COPY---------------------------------------------------------------------------------------------------------->(172)
1861               GIVEN
1862               TARGET,
1863               TOT.FOLLOW
1864               ''EVERYTHING IS THE SAME IN THE FOLLOW ON EXCEPT
1865               ''THE TARGET REPORT NUMBER AND THE TOT STATUS
1866               LET TR.TOT.STATUS( TOT.FOLLOW ) = 2
1867            ALWAYS
1868         ALWAYS
1869      ALWAYS
1870   ALWAYS
1871
1872   IF ANALYSIS(2) = TRUE
1873      USE UNIT 43 FOR OUTPUT
1874      PRINT 1 LINE WITH TARGET,TIME.V,TR.SENSOR.TYPE(TARGET),
1875      UN.COLOR(TR.TGT.UNIT(TARGET)),
1876      CT.NAME(TU.CAT(UN.TYPE.UNIT(TR.TGT.UNIT(TARGET)))),
1877      TR.RECVD.TIME(TARGET), RFAF THUS
1878      TR ****** TIME **.***** SEN ** SIDE *  CAT ****** REC **.***** RFAF ******
1879      FOR EACH .FM IN THE TR.FM.LIST(TARGET)
1880      PRINT 1 LINE WITH
1881      .FM, FM.BTRY(.FM),BY.TYPE(FM.BTRY(.FM)),FM.N.VOLS(.FM),
1882      FM.TM.CLASS(.FM),FM.RAP.FLAG(.FM)  THUS
1883      FM ****** BY *** TB *** VOLS *** MUN ***** RAP **
1884      USE UNIT 6 FOR OUTPUT
1885   ALWAYS
1886
1887   ADD 1 TO FD.N.PROCESSED(FDC)
```

```
1888    CALL FINISH.COMPUTATION------------------------------------------------------------------------------------------>(334)
1889    GIVEN
1890    TARGET,
1891    FDC
1892    ''CALCULATIONS FOR THIS TARGET REPORT ARE COMPLETED
1893    ''KEEP TRACK OF THE TR'S THAT HAVE BEEN RECENTLY COMPUTED
1894    ''AND ARE STILL BEING FIRED ON
1895    FILE TARGET IN FD.COMPLETE.LIST(FDC)
1896    IF TR.FM.LIST(TARGET) IS NOT EMPTY
1897 <------SUSPEND  ''UNTIL ALL FIRE MISSIONS ARE COMPLETED
1898    ALWAYS
1899
1900    ''THIS HAPPENS WHEN THE LAST FIRE MISSION TO BE COMPLETED REACTIVATES
1901    ''THE TARGET REPORT
1902    REMOVE THE TARGET FROM THE FD.COMPLETE.LIST(FDC)
1903    IF TARGET IS IN AN FO.TGT.RPT.LIST
1904       REMOVE TARGET FROM FO.TGT.RPT.LIST(TR.SENSOR.ID(TARGET))
1905    ALWAYS
1906    IF TARGET IS IN THE FD.TR.QUEUE ''                                                                          \DEBUG
1907       REMOVE TARGET FROM FD.TR.QUEUE
1908    ALWAYS
1909    DESTROY THE EX.TGT.REPORT
1910    CALLED TR.EX.TGT.REPORT(TARGET)
1911
1912 <--EXITPROCESS
1913 ENDPROCESS
```

```
1914 PROCESS WITH.DRAW                                                    ''                                                    P014
1915    GIVEN
1916    CHECK.UNIT
1917
1918    ADD 1 TO ANAL.CTR(179,1)  ''                                                                                    \DYN_ANAL
1919    NORMALLY MODE IS INTEGER
1920    DEFINE WITH.DRAW TO MEAN PROCESS.V ''                                                                             CHG\01
1921    DEFINE SHAPE,SCALE,W AS REAL VARIABLES
1922    DEFINE UNIT AS AN INTEGER VARIABLE
1923
1924 <—WAIT .1 SECONDS
1925
1926    IF WD.DESTRUCT.INDIC(WITH.DRAW) = YES
1927 <———EXITPROCESS
1928    OTHERWISE
1929
1930    LET FRACTION = (MU.CRIT.NO(UN.PTR(CHECK.UNIT))/
1931    TU.CRIT.NO(UN.TYPE.UNIT(CHECK.UNIT)))*100
1932
1933    PRINT 2 LINES WITH
1934    UNIT.NOS(CHECK.UNIT),
1935    TIME.V,
1936    BTL.SEQ.NO(UN.BATTLE.INDEX(CHECK.UNIT)), FRACTION
1937    THUS
1938    UNIT ****** WITHDRAWING==========
1939    AT **.**** HOURS IN BATTLE = ******    AT ***% STRENGTH  '
1940
1941    CALL REQUEST.WD.FASCAM————————————————————————————————————————————————————————————————————————————————————>(226)
1942    GIVEN
1943    CHECK.UNIT
1944    CALL REQUEST.SMOKE————————————————————————————————————————————————————————————————————————————————————>(222)
1945    GIVEN
1946    CHECK.UNIT,
1947    SMK.WD.RULE(UN.COLOR(CHECK.UNIT),
1948    NITE.OR.DAY, UN.MISSION(CHECK.UNIT)),
1949    0
1950
1951    IF UN.STATUS(CHECK.UNIT)  = STA.TO.WITH
1952       FOR EVERY OTHER.FORCE
1953       OF BTL.FORCE.SET(UN.BATTLE.INDEX(CHECK.UNIT))
1954       WITH FR.SIDE(OTHER.FORCE) NE UN.COLOR(CHECK.UNIT)
1955       FIND THE FIRST CASE
1956       IF NONE
1957          CALL ERROR.STOP————————————————————————————————————————————————————————————————————————————————————>(604)
1958       ALWAYS
1959       LOOP FOR EVERY UNIT OF THE FR.UNIT.SET(OTHER.FORCE)
1960       DO THE FOLLOWING
1961          LET .TERRAIN.TYPE = BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT))
1962          CALL RANGE.COMPUTE————————————————————————————————————————————————————————————————————————————————>(342)
1963          GIVEN
1964          UNIT,
1965          CHECK.UNIT
1966          YIELDING
1967          RANGE
1968          FOR EVERY VISIBLE.UNIT OF THE UN.LOS.LIST(CHECK.UNIT)
1969          WITH VU.POINTER(VISIBLE.UNIT) = UNIT
1970          FIND THE FIRST CASE
1971          IF NONE,
```

```
1972              LET LOS = NO
1973              LET SHAPE = TT.NLOS.SHAPE(.TERRAIN.TYPE)
1974              LET SCALE = TT.NLOS.SCALE(.TERRAIN.TYPE)
1975           ELSE
1976              LET LOS = YES
1977              LET SHAPE = TT.LOS.SHAPE(.TERRAIN.TYPE)
1978              LET SCALE = TT.LOS.SCALE(.TERRAIN.TYPE)
1979           ALWAYS
1980           CREATE A SEGMENT
1981           LET SEG.TYPE(SEGMENT) = LOS
1982           LET SEG.UNIT(SEGMENT) = UNIT
1983           ''UTILIZE WEIBULL.F FUNCTION IN NEXT DRAW ---------------------------------------------------------- \1>(642)
1984           LET W = WEIBULL.F(SHAPE,SCALE,RN.SEED)
1985           LET SEG.LENGTH(SEGMENT) = MAX.F(2.,(W*TER.W.INC./16.))
1986           FILE THIS SEGMENT IN THE UN.SEGMENT.LIST(CHECK.UNIT)
1987        ENDLOOP
1988     ELSE ''UN.STATUS(CHECK.UNIT) = ADV.TO.WITH
1989        LOOP FOR EVERY SEGMENT OF THE UN.SEGMENT.LIST(CHECK.UNIT)
1990        DO THE FOLLOWING
1991           LET UNIT = SEG.UNIT(SEGMENT)
1992           LET .TERRAIN.TYPE = BTL.TERRAIN.TYPE(UN.BATTLE.INDEX(UNIT))
1993           CALL RANGE.COMPUTE---------------------------------------------------------------------------------->(342)
1994           GIVEN
1995           SEG.UNIT(SEGMENT),
1996           CHECK.UNIT
1997           YIELDING
1998           RANGE
1999           IF SEG.TYPE(SEGMENT) = YES
2000              LET SHAPE = TT.LOS.SHAPE(.TERRAIN.TYPE)
2001              LET SCALE = TT.LOS.SCALE(.TERRAIN.TYPE)
2002           ELSE
2003              LET SHAPE = TT.NLOS.SHAPE(.TERRAIN.TYPE)
2004              LET SCALE = TT.NLOS.SCALE(.TERRAIN.TYPE)
2005           ALWAYS
2006           ''UTILIZE WEIBULL.F FUNCTION IN NEXT DRAW ---------------------------------------------------------- \1>(642)
2007           LET W = WEIBULL.F(SHAPE,SCALE,RN.SEED)
2008           LET SEG.LENGTH(SEGMENT) = MAX.F(2.,(W*TER.W.INC./16.))
2009        ENDLOOP
2010        FOR EACH MOVE OF EV.S(I.MOVE)
2011        WITH MV.UNIT(MOVE) = CHECK.UNIT
2012        FIND THE FIRST CASE
2013        IF NONE,
2014           CALL ERROR.STOP------------------------------------------------------------------------------------>(604)
2015        ALWAYS
2016        CANCEL_THE MOVE------------------------------------------------------------------------------------->(381)
2017        DESTROY THE MOVE
2018     ALWAYS
2019
2020     LET UN.STATUS(CHECK.UNIT) = WITHDRAWING
2021     CALL MIN.MOVE--------------------------------------------------------------------------------------->(106)
2022     GIVEN
2023     CHECK.UNIT
2024
2025 <--EXITPROCESS
2026 ENDPROCESS
```

```
2027 PROCESS FIRE.MISSION                                                    ''                                                    P015
2028
2029    ADD 1 TO ANAL.CTR(180,1)  ''                                                                                          \DYN_ANAL
2030    ''THIS IS A PROCESS THAT REPRESENTS THE HANDLING OF A FIRE MISSION
2031    ''ONCE IT IS RECEIVED AT A FIRE UNIT UNTIL IT IS COMPLETED OR
2032    ''THROWN OUT - THE FIRE MISSION IS ACTIVATED EITHER DIRECTLY
2033    ''FROM THE FDC FOR PRIORITY PROCESSING, OR FROM THE
2034    ''LIST OF SCHEDULED FIRE MISSIONS - FIRE MISSIONS "SENT IN"
2035    ''BY FORWARD OBSERVERS ARE AFFECTED BY THE FO'S STATUS
2036
2037    NORMALLY MODE IS INTEGER
2038    DEFINE TOT.FOLLOW TO MEAN 2
2039    DEFINE TARGET.ACQUISITION TO MEAN 0
2040    DEFINE PREPARE TO MEAN WAIT  ''                                                                                       \FIX_LATER
2041    DEFINE FIRE TO MEAN WAIT  ''                                                                                          \FIX_LATER
2042    DEFINE FM TO MEAN PROCESS.V ''                                                                                           CHG\01
2043    DEFINE .BTRY TO MEAN FM.BTRY(FM)
2044    DEFINE FO TO MEAN TR.SENSOR.ID(FM.TGT(FM))
2045    DEFINE PERSONNEL TO MEAN 1
2046
2047
2048    DEFINE PREP.TIME TO MEAN EFM.PREP.TIME(FM.EX.FIRE.MISSION(FM))
2049    DEFINE FIRE.RATE TO MEAN EFM.FIRE.RATE(FM.EX.FIRE.MISSION(FM))
2050    DEFINE SUPPRESS  TO MEAN EFM.SUPPRESS(FM.EX.FIRE.MISSION(FM))
2051    DEFINE OLD.CEP TO MEAN EFM.OLD.CEP(FM.EX.FIRE.MISSION(FM))
2052    DEFINE SUPP.TIME TO MEAN EFM.SUPP.TIME(FM.EX.FIRE.MISSION(FM))
2053    DEFINE LAST.ARTY.ENGAGE TO MEAN EFM.LAST.ARTY.ENGAGE(FM.EX.FIRE.MISSION(FM))
2054
2055
2056    CREATE AN EX.FIRE.MISSION CALLED FM.EX.FIRE.MISSION(FM)
2057    LET SIDE = UN.COLOR(BY.UNIT(.BTRY))
2058    FOR EACH TE.LINK IN THE TU.TE.LIST(UN.TYPE.UNIT(BY.UNIT(.BTRY)))
2059    WITH TU.TE.ID(TE.LINK) = PERSONNEL
2060    FIND THE FIRST CASE
2061    IF NONE
2062       PRINT 1 LINE WITH UN.TYPE.UNIT(BY.UNIT(.BTRY)) THUS
2063       = = ERROR = = NO PERSONNEL IN TYPE UNIT *****
2064       TRACE
2065       LIST ATTRIBUTES OF EX.FIRE.MISSION CALLED FM.EX.FIRE.MISSION(FM)
2066 <-----STOP
2067    OTHERWISE
2068    FOR EACH EQ IN THE UN.EQUIP.LIST(BY.UNIT(.BTRY)) WITH
2069    EQ.TE.PTR(UE.ID(EQ)) = PERSONNEL
2070    FIND THE FIRST CASE
2071    IF NONE
2072       PRINT 1 LINE WITH BY.UNIT(.BTRY) THUS
2073       = = ERROR = = NO PERSONNEL IN UNIT ****
2074       TRACE
2075       LIST ATTRIBUTES OF EX.FIRE.MISSION CALLED FM.EX.FIRE.MISSION(FM)
2076 <-----STOP
2077    OTHERWISE
2078    IF UE.QUANT(EQ) LE ARTY.DECIMATE(SIDE)*TU.TE.QUANT(TE.LINK)/100
2079       LET FIRE.RATE = REAL.F((ARTY.DECIMATE(SIDE)/100)
2080       *TB.SUST.FIRE.RATE(BY.TYPE(.BTRY)))
2081       IF DEBUG = TRUE
2082          PRINT 1 LINE WITH SIDE,TB.SUST.FIRE.RATE(BY.TYPE(.BTRY)),FIRE.RATE THUS
2083          &%$# #1 SIDE = *  SUST RATE = ****** MODIFIED RATE = ******.****
2084       ALWAYS
```

```
2085    ELSE IF UE.QUANT(EQ) GT ARTY.DEGRADE(SIDE)*TU.TE.QUANT(TE.LINK)/100
2086       LET FIRE.RATE = REAL.F(TB.SUST.FIRE.RATE(BY.TYPE(.BTRY)))
2087       IF DEBUG = TRUE
2088          PRINT 1 LINE WITH SIDE,TB.SUST.FIRE.RATE(BY.TYPE(.BTRY)),FIRE.RATE THUS
2089          &%$# #2 SIDE = *  SUST RATE = ****** MODIFIED RATE = ******.****
2090       ALWAYS
2091    ELSE
2092       LET FIRE.RATE = (REAL.F((UE.QUANT(EQ)/TU.TE.QUANT(TE.LINK)) -
2093       REAL.F(ARTY.DECIMATE(SIDE)/100))/((ARTY.DEGRADE(SIDE) -
2094       ARTY.DECIMATE(SIDE))/100)
2095       * TB.SUST.FIRE.RATE(BY.TYPE(.BTRY))*(1.0-ARTY.DECIMATE(SIDE)/100.0)
2096       +(ARTY.DECIMATE(SIDE)/100)*TB.SUST.FIRE.RATE(BY.TYPE(.BTRY)))
2097       IF DEBUG = TRUE
2098          PRINT 1 LINE WITH SIDE,TB.SUST.FIRE.RATE(BY.TYPE(.BTRY)),FIRE.RATE THUS
2099          &%$# #3 SIDE = *  SUST RATE = ****** MODIFIED RATE = ******.****
2100       ALWAYS
2101    ALWAYS ALWAYS
2102
2103    IF DEBUG = TRUE AND
2104       TR.PGM.STATUS(FM.TGT(FM)) GE TRUE
2105       PRINT 1 LINE WITH FM.TGT(FM) AND TIME.V THUS
2106       $#$#$# FIRE MISSION STARTED ON TARGET ******* AT TIME **.*****
2107    ALWAYS
2108
2109    IF FM.START.TIME( FM ) > 0.0  ''THEN THIS IS A SCHEDULED MISSION
2110       REMOVE THE FM FROM BY.SCHD.LIST( .BTRY )
2111       DESTROY EX.FIRE.MISSION CALLED FM.EX.FIRE.MISSION(FM)
2112       STORE 0 IN FM.EX.FIRE.MISSION(FM) ''                                  CHG\12  \TO_AVOID_REDUNDANT_DESTROYS
2113       ''NOTE THAT TOT FOLLOW ON FIRE MISSIONS ARE SCHEDULED
2114       ''BUT ARE FIRED IMMEDIATELY AFTER FIRING THE INITIAL
2115       ''TOT VOLLEY
2116    ALWAYS
2117    FILE THE FM IN THE BY.FM.QUEUE(.BTRY)  ''WHERE IT IS PRIORITIZED
2118    CALL BTRY.FM.ENQ------------------------------------------------------------------------------------------------------>(164)
2119    GIVEN
2120    FM,
2121    .BTRY
2122    IF FM.N.VOLS(FM) > 0
2123       UNTIL FM.STATUS(FM)=ACTIVE,        ''FM MUST WAIT ITS TURN
2124 <------SUSPEND
2125    ALWAYS
2126
2127    LET LAST.ARTY.ENGAGE = (TIME.V*MINUTES.V - UN.LAST.ARTY.ENG(BY.UNIT
2128    (.BTRY))) * 10.
2129    LET SUPP.TIME = TB.SUPPRESS.TIME(BY.TYPE(.BTRY)) * 10
2130    IF FM.N.VOLS(FM) GT 0
2131       LOOP UNTIL LAST.ARTY.ENGAGE GE SUPP.TIME
2132       DO   ''                                                               \1
2133          LET SUPPRESS = (SUPP.TIME - LAST.ARTY.ENGAGE)/10.
2134          IF DEBUG = TRUE
2135             PRINT 1 LINE WITH UNIT.NOS(BY.UNIT(.BTRY)), TB.NAME(BY.TYPE(.BTRY)),
2136             FM, SUPPRESS THUS
2137             UNIT ****** ( ****** ) - FM ****** IS DELAYED ***.** MINUTES
2138          ALWAYS
2139 <----------WAIT SUPPRESS MINUTES
2140          IF FM.N.VOLS(FM) = 0
2141       <------EXITLOOP
2142          OTHERWISE
```

```
2143          LET LAST.ARTY.ENGAGE = (TIME.V*MINUTES.V - UN.LAST.ARTY.ENG
2144          (BY.UNIT(.BTRY))) * 10.
2145       ENDLOOP
2146    ALWAYS
2147    IF FM.N.VOLS(FM) > 0
2148       IF TR.PGM.STATUS(FM.TGT(FM)) = FALSE
2149          LET PREP.TIME = UNIFORM.F(REAL.F(TB.MIN.PREP(BY.TYPE(.BTRY))),
2150          REAL.F(TB.MAX.PREP(BY.TYPE(.BTRY))),RN.SEED)/600.
2151          ''FIRE MISSION FOR FIRING
2152          IF TR.TGT.UNIT(FM.TGT(FM))=0  ''PRE-PLANNED FIRES?
2153             LET PREP.TIME=0.0  ''THEN IT WAS PREVIOUSLY COMPUTED
2154
2155             PRINT 1 LINE WITH FM, FM.TGT(FM) THUS
2156             ==FIRE.MISSION ******  TARGET REPORT ******  NO TGT UNIT
2157
2158          ALWAYS
2159       ALWAYS
2160       LET OLD.CEP = TR.CEP(FM.TGT(FM))
2161       IF TR.SENSOR.TYPE(FM.TGT(FM)) = "FO"
2162          IF TR.PGM.STATUS(FM.TGT(FM)) NE FALSE
2163             PREPARE FM.PREP.TIME(FM) SECONDS
2164          ELSE
2165             IF US.STATUS(FO.US.LINK(FO)) > TARGET.ACQUISITION
2166                ''QUICK RESPONSE ON MANEUVER SUPPORT
2167                LET PREP.TIME = PREP.TIME/2.
2168             ALWAYS
2169
2170             PREPARE PREP.TIME HOURS  ''PAUSE TO PREPARE FIRE MISSION
2171
2172          ALWAYS
2173          IF N.FO.CUR.FM.LIST(FO) = 0
2174             ''WHEN THIS FO'S ONLY MISSION AND IS PGM                                                    \1
2175             IF TR.PGM.STATUS(FM.TGT(FM)) = TRUE
2176                ''THE FO STOPS SEARCHING                                                                 \1
2177                INTERRUPT FORWARD.OBSERVER CALLED FO------------------------------------------------------>(438)
2178             ALWAYS
2179             FILE FM IN FO.CUR.FM.LIST(FO)
2180          ELSE
2181             IF N.FO.CUR.FM.LIST(FO) = 1
2182                IF FO IS NOT IN EV.S
2183                   ''FO ALREADY INTERRUPTED - THE FM HE IS
2184                   ''WORKING ON MUST BE A PGM MISSION
2185                   IF TR.PGM.STATUS(FM.TGT(FM)) = TRUE
2186                      FILE FM IN FO.CUR.FM.LIST(FO)
2187                      LET FM.STATUS(FM) = HOLD
2188 <-------------------SUSPEND
2189                      ''UNTIL THE FO IS FINISHED WITH THE
2190                      ''THE CURRENT PGM MISSION AND CAN START
2191                      ''THIS ONE
2192                   ELSE
2193                      ''WILL BE FIRED AS MET+VE
2194                      LET TR.CEP(FM.TGT(FM))=2*TR.CEP(FM.TGT(FM))
2195                      ''SINCE IT CAN'T BE ADJUSTED BECAUSE
2196                      ''THE FO IS TOO BUSY
2197                   ALWAYS
2198                ELSE
2199                   ''THIS MISSION WILL BE ADJUSTED BY FO
2200                   FILE FM IN FO.CUR.FM.LIST(FO)
```

```
2201                ALWAYS
2202             ELSE
2203                IF N.FO.CUR.FM.LIST(FO) = 2
2204                   ''FIRE AS MET+VE AND DON'T TASK THE FO
2205                   LET TR.CEP(FM.TGT(FM)) = 2*TR.CEP(FM.TGT(FM))
2206                ALWAYS
2207             ALWAYS
2208          ALWAYS
2209       ALWAYS
2210    ELSE
2211       IF TR.PGM.STATUS(FM.TGT(FM)) NE FALSE
2212          PREPARE FM.PREP.TIME(FM) SECONDS
2213       ELSE
2214          PREPARE PREP.TIME HOURS
2215       ALWAYS
2216    ALWAYS
2217    FOR EACH LINK IN TB.TM.LIST(BY.TYPE(.BTRY))
2218    WITH TB.TM.CLASS(LINK)=FM.TM.CLASS(FM)
2219    AND  TB.TM(LINK)=FM.TM(FM)
2220    FIND THE FIRST CASE
2221    IF NONE
2222       LET LINK = 0
2223    ALWAYS
2224    ''WHEN THIS IS COUNTER-BATTERY FIRE THE LAST VOLLEY WILL BE                                                    \1
2225    ''FASCAM (IF THE BTRY HAS IT) TO SUPPRESS THE ENEMY BATTERY.
2226    IF UN.STATUS(TR.TGT.UNIT(FM.TGT(FM))) = STATIONARY AND
2227       CT.GROUP(TU.CAT(UN.TYPE.UNIT(TR.TGT.UNIT(FM.TGT(FM))))) =
2228       ARTILLERY
2229       FOR EVERY .F.LINK IN TB.TM.LIST(BY.TYPE(.BTRY))
2230       WITH TB.TM.CLASS(.F.LINK) = "FASCAM" AND
2231       FMM.MAX.RANGE(TB.TM(.F.LINK)) GT FM.RANGE(FM)
2232       FIND THE FIRST CASE
2233       IF NONE
2234          LET .F.LINK = 0
2235       ALWAYS
2236    ALWAYS
2237    LOOP FOR VOL = 1 TO FM.N.VOLS(FM)
2238    DO
2239       ADD 1 TO FM.FIRED.VOLS(FM)
2240       IF FM.FIRED.VOLS(FM) LT FM.N.VOLS(FM) OR
2241          .F.LINK = 0
2242          IF LINK > 0
2243             ADD N.BY.HOW.SET(.BTRY) *
2244             TB.RND.PER.LAUNCH(BY.TYPE(.BTRY))
2245             TO TB.TM.FIRED(LINK)
2246          ALWAYS
2247       ELSE
2248          ADD N.BY.HOW.SET(.BTRY) *
2249          TB.RND.PER.LAUNCH(BY.TYPE(.BTRY))
2250          TO TB.TM.FIRED(.F.LINK)
2251       ALWAYS
2252       CALL VOLLEY------------------------------------------------------------------------------------------------>(246)
2253       GIVEN
2254       .BTRY
2255       IF FM.FIRED.VOLS(FM) LT FM.N.VOLS(FM)
2256          IF TR.PGM.STATUS(FM.TGT(FM)) = TRUE
2257             FIRE 20. SECONDS
2258          ELSE
```

```
2259            FIRE 1./(FIRE.RATE/1000.) MINUTES
2260         ALWAYS
2261      ALWAYS
2262   ENDLOOP
2263   IF FM.FIRED.VOLS(FM) > 0
2264      IF (FM.TM.CLASS(FM) = "FASCAM" OR .F.LINK GT 0)
2265         AND FM.N.VOLS(FM) GT 0
2266         IF TR.TGT.UNIT(FM.TGT(FM)) NE TR.REP.UNIT(FM.TGT(FM))
2267            ''TGT=REP MEANS MISSION WAS TO RESEED BARRIER
2268            ''MINEFIELD - IT WILL HAVE NO EFFECT ON A UNIT
2269            CALL MINE.EFFECTS------------------------------------------------------------------------------------------>(250)
2270            GIVEN
2271            TR.TGT.UNIT(FM.TGT(FM)),
2272            UN.X.COORD(TR.TGT.UNIT(FM.TGT(FM))),
2273            UN.Y.COORD(TR.TGT.UNIT(FM.TGT(FM))),
2274            3,    ''ARTY DELIVERED MINES
2275            FM,
2276            0
2277
2278            YIELDING
2279            .DELAY
2280         ALWAYS
2281         IF CT.GROUP(TU.CAT(UN.TYPE.UNIT(TR.TGT.UNIT(FM.TGT(
2282            FM))))) = ARTILLERY
2283            LET .TYPE.MISSION = 2
2284            LET .ROUNDS = N.BY.HOW.SET(.BTRY)
2285         ELSE
2286            LET .TYPE.MISSION = 1
2287            LET .ROUNDS = N.BY.HOW.SET(.BTRY)*FM.N.VOLS(FM)
2288         ALWAYS
2289         WRITE TIME.V,TR.TGT.UNIT(FM.TGT(FM)),
2290         TB.NAME(BY.TYPE(.BTRY)),.ROUNDS,
2291         .TYPE.MISSION, SIDE AS S 3, D(6,2), S 4, I 6, S 4, T 6, ''                                          \TEXT
2292         S 4, I 4, S 4, I 2, S 4, I 2,/  USING UNIT 47
2293      ELSE
2294         IF FM.TM.CLASS(FM) = "ILLUM"
2295            CALL ILLUM.EFFECTS----------------------------------------------------------------------------------------->(200)
2296            GIVEN
2297            FM
2298         ELSE
2299            IF FM.TM.CLASS(FM) = "SMOKE"
2300               CALL SMOKE.EFFECTS-------------------------------------------------------------------------------------->(234)
2301               GIVEN
2302               FM
2303            ELSE
2304               IF FM.TM.CLASS(FM) = "HE" OR
2305                  FM.TM.CLASS(FM) = "ICM" OR
2306                  FM.TM.CLASS(FM) = "PGM" OR
2307                  FM.TM.CLASS(FM) = "SDM"
2308                  CALL BTRY.EFFECTS------------------------------------------------------------------------------------>(260)
2309                  GIVEN
2310                  FM
2311               ELSE
2312                  TRACE
2313                  LIST ATTRIBUTES OF EX.FIRE.MISSION CALLED FM.EX.FIRE.MISSION(FM)
2314 <----------------STOP
2315               ALWAYS
2316            ALWAYS
```

```
2317          ALWAYS
2318       ALWAYS
2319
2320       IF FM.N.VOLS(FM) GT 0
2321 <-------WAIT 1. / (FIRE.RATE/1000.) MINUTES
2322
2323       ALWAYS
2324    ALWAYS
2325
2326    LET TR.CEP(FM.TGT(FM)) = OLD.CEP
2327    ''DETERMINE IF THERE WERE ANY HOWITZER RAM FAILURES
2328    ''AS A RESULT OF THIS FIRING
2329    IF TR.PGM.STATUS(FM.TGT(FM)) NE TRUE
2330       LOOP FOR EACH HOW IN BY.HOW.SET(.BTRY),
2331       DO
2332          IF TB.SFAIL.MEAN.RNDS(BY.TYPE(.BTRY)) GT 0
2333             SUBTRACT FM.N.VOLS(FM) FROM HW.SFAIL.RNDS(HOW)
2334          ALWAYS
2335          IF TB.LFAIL.MEAN.RNDS(BY.TYPE(.BTRY)) GT 0
2336             SUBTRACT FM.N.VOLS(FM) FROM HW.LFAIL.RNDS(HOW)
2337          ALWAYS
2338          IF HW.SFAIL.RNDS(HOW) IS NOT POSITIVE  OR
2339             HW.LFAIL.RNDS(HOW) IS NOT POSITIVE
2340             ACTIVATE_A HOW.REPAIR GIVING HOW NOW ''BROWN COW----------------------------------------------------------------------->(463)
2341          ALWAYS
2342       ENDLOOP
2343    ALWAYS
2344
2345    IF TR.SENSOR.TYPE(FM.TGT(FM)) = "RPV" AND
2346       TR.PGM.STATUS(FM.TGT(FM)) = TRUE
2347       ''FREE THE ADJUSTING RPV TO CONTINUE ITS SEARCH
2348
2349       LET TIME.A(TR.SENSOR.ID(FM.TGT(FM)))=REAL.F(FM.TOF.TIME(FM))/
2350       3600.
2351
2352       RESUME REMOTE.PILOT.VEHICLE CALLED TR.SENSOR.ID(FM.TGT(FM))----------------------------------------------------------------->(467)
2353    ALWAYS
2354
2355    IF FM IS IN AN FO.CUR.FM.LIST
2356       ''WHEN THE FIRE MISSION WAS FO ADJUSTED, RESTART THE FO                                                                    \1
2357       ''SO THAT HE MAY CONTINUE HIS SEARCH UNLESS HE HAS
2358       ''ANOTHER FIRE MISSION TO ADJUST
2359
2360
2361       REMOVE THE FM FROM FO.CUR.FM.LIST(FO)
2362       DESTROY EX.FIRE.MISSION CALLED FM.EX.FIRE.MISSION(FM)
2363       STORE 0 IN FM.EX.FIRE.MISSION(FM) ''                                                CHG\12  \TO_AVOID_REDUNDANT_DESTROYS
2364       IF FO.CUR.FM.LIST(FO) IS EMPTY
2365          IF TR.PGM.STATUS(FM.TGT(FM)) = TRUE AND FO IS NOT IN EV.S
2366             LET TIME.A(FO) = REAL.F(FM.TOF.TIME(FM))/3600.
2367             RESUME FORWARD.OBSERVER CALLED FO----------------------------------------------------------------------------------->(438)
2368          ALWAYS
2369       ELSE
2370          IF TR.PGM.STATUS(FM.TGT(F.FO.CUR.FM.LIST(FO))) NE TRUE AND
2371             TR.PGM.STATUS(FM.TGT(FM)) = TRUE AND FO IS NOT IN EV.S
2372             RESUME FORWARD.OBSERVER CALLED FO----------------------------------------------------------------------------------->(438)
2373          ELSE
2374             IF TR.PGM.STATUS(FM.TGT(FM)) = TRUE
```

```
2375                AND F.FO.CUR.FM.LIST(FO) IS NOT IN EV.S
2376                RESUME FIRE.MISSION CALLED F.FO.CUR.FM.LIST(FO)------------------------------------------------------------>(480)
2377             ALWAYS
2378          ALWAYS
2379       ALWAYS
2380    ALWAYS
2381
2382    ''AT THE END OF THE FIRE MISSION, CLEAN UP ALL
2383    ''OWNERSHIP RELATIONSHIPS ASSOCIATED WITH THIS
2384    ''FIRE MISSION SO THAT THEY ARE NOT LEFT HANGING
2385    ''AFTER THE PROCESS KILLS ITSELF
2386    REMOVE THE FM FROM THE BY.FM.QUEUE(.BTRY)
2387    ''DESTROY EX.FIRE.MISSION CALLED FM.EX.FIRE.MISSION(FM)                                  CHG\07  \REDUNDANT_DESTROY
2388    CALL BTRY.FM.DEQ------------------------------------------------------------------------------------------------>(163)
2389    GIVEN
2390    FM
2391    REMOVE THE FM FROM THE TR.FM.LIST(FM.TGT(FM))
2392    IF TR.FM.LIST(FM.TGT(FM)) IS EMPTY
2393       IF FM.TGT(FM) IS NOT IN EV.S
2394          REACTIVATE_THE TARGET.REPORT CALLED FM.TGT(FM) NOW------------------------------------------------------------>(470)
2395       ALWAYS
2396    ALWAYS
2397
2398    IF TB.SHOOT.SCOOT.IND(BY.TYPE(.BTRY)) = 1 AND FM.N.VOLS(FM) GT 0
2399       CALL FA.BN.MOVEMENT GIVEN 0 AND .BTRY--------------------------------------------------------------------------->( 89)
2400    ALWAYS
2401
2402    IF TR.PGM.STATUS(FM.TGT(FM)) NE TRUE AND FM.N.VOLS(FM) GT 0
2403 <------WAIT 1.0 MINUTES
2404    ALWAYS
2405    IF FM.EX.FIRE.MISSION(FM) IS NOT ZERO, ''                                                \VAX
2406       DESTROY EX.FIRE.MISSION CALLED FM.EX.FIRE.MISSION(FM)
2407    ALWAYS ''                                                                                \VAX
2408
2409 <--EXITPROCESS
2410 ENDPROCESS
```

```
2411 PROCESS ASSESSMENT                                                                                    P016
2412    GIVEN
2413    PK,
2414    TARGET.EQUIP,
2415    TARGET.UNIT,
2416    FIRING.UNIT,
2417    SHOOT.OUT.NO
2418
2419    ADD 1 TO ANAL.CTR(181,1)  ''                                                                  \DYN_ANAL
2420    NORMALLY MODE IS INTEGER
2421    DEFINE .NO.IN.SET AS AN INTEGER VARIABLE ''                                                CHG\26 \DEBUG
2422    DEFINE CHANCE, PK AS REAL VARIABLES
2423    DEFINE .ASSESSMENT TO MEAN PROCESS.V ''                                                       CHG\01
2424    LET WPN = ESO.WPN(SO.EX.SHOOT.OUT(SHOOT.OUT.NO))
2425    LET RANGE = ESO.RANGE(SO.EX.SHOOT.OUT(SHOOT.OUT.NO))
2426
2427    IF AS.DESTRUCT.INDIC(.ASSESSMENT) = YES
2428 <------EXITPROCESS
2429    OTHERWISE
2430
2431    IF SO.HELICOPTER(SHOOT.OUT.NO) LE 0 AND
2432       TARGET.UNIT NE SD.AIRFIELD(UN.COLOR(TARGET.UNIT))
2433       FOR EVERY VISIBLE.UNIT OF UN.LOS.LIST(TARGET.UNIT)
2434       WITH VU.POINTER(VISIBLE.UNIT) = FIRING.UNIT
2435       FIND THE FIRST CASE
2436       IF NONE
2437          CALL ERROR.STOP------------------------------------------------------------------------------------>(604)
2438       ALWAYS
2439       IF VU.STATUS(VISIBLE.UNIT) = NO
2440          SCHEDULE_AN ENGAGEMENT----------------------------------------------------------------------------->(369)
2441          GIVEN
2442          TARGET.UNIT,
2443          FIRING.UNIT NOW  ''                                                                          \1
2444       ALWAYS
2445
2446       ''THIS WAIT IS TO ALLOW THE RETURN FIRE ENGAGEMENT
2447       ''TO BE STARTED AND HENCE THE SHOOT.OUT TO BE STARTED
2448       ''THE REASON IS THAT IF THEY ARE KILLED, VALUES OF
2449       ''THEIR ATTRIBUTES MUST BE SET
2450
2451 <------WAIT .1 SECONDS
2452
2453    ALWAYS
2454    IF AS.DESTRUCT.INDIC(.ASSESSMENT) = YES
2455 <------EXITPROCESS
2456    OTHERWISE
2457
2458    LET CHANCE = RANDOM.F(RN.SEED)
2459    IF CHANCE GT PK,
2460       LET SO.RESULTS(SHOOT.OUT.NO) = MISS
2461       LET J1 = EQ.KV.ID(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO)))    '' %14FEB79_%JEN
2462       LET J2 = EQ.KV.ID(UE.ID(TARGET.EQUIP)) '' %14FEB79_%JEN
2463       IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
2464          WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(WPN)),
2465          EQ.NAME(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO))),
2466          EQ.TE.PTR(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO))),
2467          TW.NAME(WPN.ID(WPN)),
2468          EQ.NAME(UE.ID(TARGET.EQUIP)),
```

```
2469          EQ.TE.PTR(UE.ID(TARGET.EQUIP)),
2470          RANGE*16,PK.DEFILADE,SHOOT.OUT.NO,
2471          UN.COLOR(FIRING.UNIT), PK AS S 1,"0",S 1,D(7,4),S 1,
2472          I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''                    \TEXT
2473          I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
2474          S 4,"1A1",/
2475          USING UNIT 55
2476       ALWAYS
2477 <-----EXITPROCESS
2478    OTHERWISE
2479
2480    ''GOT A KILL
2481    LET SO.RESULTS(SHOOT.OUT.NO) = HIT
2482    IF SO.HELICOPTER(SHOOT.OUT.NO) > 0 AND
2483       UE.ID(TARGET.EQUIP) = HC.UE.ID(SO.HELICOPTER(SHOOT.OUT.NO))
2484       IF UN.COLOR(TARGET.UNIT) = RED
2485          LET .TEAM = BTL.RD.HC.TEAM(UN.BATTLE.INDEX(TARGET.UNIT))
2486       ELSE
2487          LET .TEAM = BTL.BL.HC.TEAM(UN.BATTLE.INDEX(TARGET.UNIT))
2488       ALWAYS
2489       FOR EVERY .HC IN HT.MEMBER.LIST(.TEAM)
2490       WITH HC.TYPE(.HC) = HC.TYPE(SO.HELICOPTER(SHOOT.OUT.NO))
2491       ADD 1 TO NO.TGTS
2492       LET VICTIM = RANDI.F(1, NO.TGTS, RN.SEED)
2493       LOOP FOR EVERY .HC IN HT.MEMBER.LIST(.TEAM)
2494       WITH HC.TYPE(.HC) = HC.TYPE(SO.HELICOPTER(SHOOT.OUT.NO))
2495       DO
2496          ADD 1 TO NO.VICTIM
2497          IF NO.VICTIM = VICTIM
2498             LET DEAD.CHOPPER = .HC
2499             LET SO.HELICOPTER(SHOOT.OUT.NO) = .HC
2500       <-----EXITLOOP
2501          OTHERWISE
2502       ENDLOOP
2503       IF DEAD.CHOPPER = 0
2504          CALL ERROR.STOP------------------------------------------------------------------------------------------------------>(604)
2505       ALWAYS
2506       LOOP FOR EACH HELICOPTER.FIRE OF EV.S(I.HELICOPTER.FIRE)
2507       WITH ATK.HELICOPTER(HELICOPTER.FIRE) =
2508       SO.HELICOPTER(SHOOT.OUT.NO)
2509       DO THE FOLLOWING
2510          LET HEL.IS.KILLED.INDIC(HELICOPTER.FIRE) = YES
2511          LET HF.DESTRUCT.INDIC(HELICOPTER.FIRE) = YES
2512       ENDLOOP
2513       LOOP FOR EVERY SHOOT.OUT IN THE EV.S(I.SHOOT.OUT)
2514       WITH SO.HELICOPTER(SHOOT.OUT) =
2515       SO.HELICOPTER(SHOOT.OUT.NO)
2516       DO THE FOLLOWING
2517          LET SO.FIRING.TABLE(SHOOT.OUT) = 0
2518          LOOP FOR EVERY ASSESSMENT IN EV.S(I.ASSESSMENT)
2519          WITH AS.SHOOT.OUT(ASSESSMENT) = SHOOT.OUT
2520          DO
2521             INTERRUPT ASSESSMENT---------------------------------------------------------------------------------------->(487)
2522             LET AS.DESTRUCT.INDIC(ASSESSMENT) = YES
2523             REACTIVATE_THE ASSESSMENT NOW------------------------------------------------------------------------------->(487)
2524          ENDLOOP
2525       ENDLOOP
2526       LOOP FOR EVERY VISIBLE.UNIT IN THE
```

```
2527        HC.UN.LOS.LIST(DEAD.CHOPPER)
2528        DO THE FOLLOWING
2529           REMOVE THE VISIBLE.UNIT FROM
2530           THE HC.UN.LOS.LIST(DEAD.CHOPPER)
2531           DESTROY THE VISIBLE.UNIT
2532        ENDLOOP
2533        FOR EACH .FORCE
2534        IN THE BTL.FORCE.SET(UN.BATTLE.INDEX(FIRING.UNIT))
2535        WITH FR.SIDE(.FORCE) = UN.COLOR(FIRING.UNIT)
2536        FIND THE FIRST CASE
2537        IF NONE
2538           CALL ERROR.STOP------------------------------------------------------------------------------------------------------>(604)
2539        ALWAYS
2540        LOOP FOR EACH .UNIT IN FR.UNIT.SET(.FORCE)
2541        DO
2542           FOR EACH VISIBLE.HC IN UN.HC.LOS.LIST(.UNIT)
2543           WITH VU.POINTER(VISIBLE.HC) = DEAD.CHOPPER
2544           FIND THE FIRST CASE
2545           IF FOUND
2546              REMOVE VISIBLE.HC FROM UN.HC.LOS.LIST(.UNIT)
2547              DESTROY THE VISIBLE.UNIT CALLED VISIBLE.HC
2548           ALWAYS
2549        ENDLOOP
2550        LET J1 = EQ.KV.ID(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO)))
2551        LET J2 = EQ.KV.ID(HC.UE.ID(DEAD.CHOPPER))
2552        IF J1 > 0 AND J2 > 0
2553           ADD 1 TO KV.SCORE(UN.COLOR(FIRING.UNIT),J1,J2)
2554        ALWAYS
2555        LET SO.FIRING.TABLE(SHOOT.OUT.NO) = 0
2556        REMOVE DEAD.CHOPPER FROM THE HT.MEMBER.LIST(HC.BTL.TEAM(
2557        DEAD.CHOPPER))
2558        FOR EACH .HC IN HT.MEMBER.LIST(.TEAM)
2559        WITH HC.UE.ID(.HC) = HC.UE.ID(DEAD.CHOPPER) AND
2560        (HC.STATUS(.HC) = DETECTING OR HC.STATUS(.HC) = ENGAGING)
2561        FIND THE FIRST CASE
2562        IF NONE
2563           LOOP FOR EVERY UN IN FR.UNIT.SET(.FORCE)
2564           DO THE FOLLOWING
2565              LOOP FOR EVERY UEL IN UN.EQUIP.LIST(UN)
2566              DO THE FOLLOWING
2567                 LOOP FOR EVERY FT IN UE.TARGET.LIST(UEL)
2568                 WITH FT.TGT.UNIT(FT) = TARGET.UNIT AND
2569                 FT.TARGET.EQUIP(FT) = TARGET.EQUIP
2570                 DO THE FOLLOWING
2571                    REMOVE THE FT FROM THE UE.TARGET.LIST(UEL)
2572                    DESTROY THE FIRING.TABLE CALLED FT
2573                 ENDLOOP
2574              ENDLOOP
2575           ENDLOOP
2576        ALWAYS
2577        IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
2578           WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(WPN)),
2579           EQ.NAME(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO))),
2580           EQ.TE.PTR(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO))),
2581           TW.NAME(WPN.ID(WPN)),
2582           EQ.NAME(UE.ID(TARGET.EQUIP)),
2583           EQ.TE.PTR(UE.ID(TARGET.EQUIP)),
2584           RANGE*16,PK.DEFILADE,SHOOT.OUT.NO,
```

```
2585          UN.COLOR(FIRING.UNIT), PK AS S 1,"1",S 1,D(7,4),S 1,
2586          I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''                                    \TEXT
2587          I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
2588          S 4,"1A2",/
2589          USING UNIT 55
2590       ALWAYS
2591       DESTROY THE HELICOPTER CALLED DEAD.CHOPPER
2592       IF ANALYSIS(6) GT 0
2593          CALL OUTPUT.ATTRITION------------------------------------------------------------------>(611)
2594          GIVEN
2595          TARGET.UNIT,
2596          UE.ID(FIRING.EQUIP(SHOOT.OUT.NO)),
2597          UE.ID(TARGET.EQUIP),
2598          1,
2599          "DIRECT"
2600       ALWAYS
2601 <-----EXITPROCESS
2602    OTHERWISE
2603
2604    IF TARGET.UNIT = SD.AIRFIELD(UN.COLOR(TARGET.UNIT))
2605       ''THIS IS A SHOT AT AN AIRCRAFT.
2606       LET .AAT = FT.AC.ATK.TGT(SO.FIRING.TABLE(SHOOT.OUT.NO))
2607       FOR EACH .AC.ATK.TGT IN EV.S(I.AC.ATK.TGT)
2608       WITH .AC.ATK.TGT = .AAT
2609       FIND THE FIRST CASE
2610       IF FOUND
2611          INTERRUPT AC.ATK.TGT CALLED .AAT------------------------------------------------------->(419)
2612          ADD 1 TO KV.SCORE(UN.COLOR(FIRING.UNIT),
2613          EQ.KV.ID(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO))),
2614          EQ.KV.ID(UE.ID(TARGET.EQUIP)))
2615          IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
2616             WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(WPN)),
2617             EQ.NAME(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO))),
2618             EQ.TE.PTR(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO))),
2619             TW.NAME(WPN.ID(WPN)),
2620             EQ.NAME(UE.ID(TARGET.EQUIP)),
2621             EQ.TE.PTR(UE.ID(TARGET.EQUIP)),
2622             RANGE*16,PK.DEFILADE,SHOOT.OUT.NO,
2623             UN.COLOR(FIRING.UNIT), PK AS S 1,"1",S 1,D(7,4),S 1,
2624             I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''                                 \TEXT
2625             I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
2626             S 4,"1A3",/
2627             USING UNIT 55
2628          ALWAYS
2629          LET AAT.VIS.IND(.AAT) = NO
2630          LET AAT.AC.KILLED.IND(.AAT) = YES
2631          REACTIVATE_THE AC.ATK.TGT CALLED .AAT NOW---------------------------------------------->(419)
2632          LOOP FOR EACH .SO IN EV.S(I.SHOOT.OUT)
2633          WITH SO.FIRING.TABLE(.SO) GT 0 AND
2634          FT.AC.ATK.TGT(SO.FIRING.TABLE(.SO)) = .AAT
2635          DO
2636             LET SO.FIRING.TABLE(.SO) = 0
2637             LOOP FOR EACH .ASSESS IN EV.S(I.ASSESSMENT)
2638             WITH AS.SHOOT.OUT(.ASSESS) = .SO
2639             DO
2640                INTERRUPT ASSESSMENT CALLED .ASSESS------------------------------------------->(487)
2641                LET AS.DESTRUCT.INDIC(.ASSESS) = YES
2642                REACTIVATE_THE ASSESSMENT CALLED .ASSESS NOW---------------------------------->(487)
```

```
2643            ENDLOOP
2644         ENDLOOP
2645         LOOP FOR EVERY UEL IN UN.EQUIP.LIST(FIRING.UNIT)
2646         DO
2647            LOOP FOR EVERY FT IN UE.TARGET.LIST(UEL)
2648            WITH FT.AC.ATK.TGT(FT) = .AAT
2649            DO
2650               REMOVE FT FROM UE.TARGET.LIST(UEL)
2651               DESTROY THE FIRING.TABLE CALLED FT
2652            ENDLOOP
2653         ENDLOOP
2654         IF ANALYSIS(6) GT 0
2655            CALL OUTPUT.ATTRITION------------------------------------------------------------------------------------------------------>(611)
2656            GIVEN
2657            TARGET.UNIT,
2658            UE.ID(FIRING.EQUIP(SHOOT.OUT.NO)),
2659            UE.ID(TARGET.EQUIP),
2660            1,
2661            "DIRECT"
2662         ALWAYS
2663      ALWAYS
2664
2665 <----RETURN
2666   OTHERWISE
2667
2668   IF UE.QUANT(TARGET.EQUIP) LE 0
2669      IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
2670         WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(WPN)),
2671         EQ.NAME(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO))),
2672         EQ.TE.PTR(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO))),
2673         TW.NAME(WPN.ID(WPN)),
2674         EQ.NAME(UE.ID(TARGET.EQUIP)),
2675         EQ.TE.PTR(UE.ID(TARGET.EQUIP)),
2676         RANGE*16,PK.DEFILADE,SHOOT.OUT.NO,
2677         UN.COLOR(FIRING.UNIT), PK AS S 1,"1",S 1,D(7,4),S 1,
2678         I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''                                                    \TEXT
2679         I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
2680         S 4,"1A4",/
2681         USING UNIT 55
2682      ALWAYS
2683 <----EXITPROCESS
2684   OTHERWISE
2685
2686   IF SO.LIST(TARGET.EQUIP) IS NOT EMPTY
2687      LET .NO.IN.SET = N.SO.LIST(TARGET.EQUIP) ''                                  \OPTIMIZE CHG\26 \DEBUG
2688      LET VICTIM = INT.F(RANDI.F(1,.NO.IN.SET,1)) ''                                     CHG\26 \DEBUG
2689      LET N = 1
2690      LOOP FOR EVERY SHOOT.OUT OF SO.LIST(TARGET.EQUIP)
2691      UNTIL N = VICTIM
2692      DO THE FOLLOWING
2693         LET N = N+1
2694      ENDLOOP
2695      IF DROP.DEAD.INDICATOR(SHOOT.OUT) = NO
2696         LET DROP.DEAD.INDICATOR(SHOOT.OUT) = YES
2697         INTERRUPT SHOOT.OUT------------------------------------------------------------------------------------------------------>(493)
2698         LET TIME.A(SHOOT.OUT) = 1/60 ''HOUR                                                                \OPTIMIZE
2699         RESUME SHOOT.OUT---------------------------------------------------------------------------------------------------------->(493)
2700      ELSE
```

```
2701 <--------EXITPROCESS
2702       ALWAYS
2703    ALWAYS
2704
2705    LET J1 = EQ.KV.ID(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO)))
2706    LET J2 = EQ.KV.ID(UE.ID(TARGET.EQUIP))
2707    IF J1 > 0   AND   J2  >  0,
2708       ADD 1 TO KV.SCORE(UN.COLOR(FIRING.UNIT), J1, J2 )
2709    ALWAYS
2710
2711    IF ANALYSIS(6) GT 0
2712       CALL OUTPUT.ATTRITION------------------------------------------------------------------------------------------------------------>(611)
2713       GIVEN
2714       TARGET.UNIT,
2715       UE.ID(FIRING.EQUIP(SHOOT.OUT.NO)),
2716       UE.ID(TARGET.EQUIP),
2717       1,
2718       "DIRECT"
2719    ALWAYS
2720
2721    SUBTRACT 1 FROM UE.QUANT(TARGET.EQUIP)
2722    IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
2723       WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(WPN)),
2724       EQ.NAME(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO))),
2725       EQ.TE.PTR(UE.ID(FIRING.EQUIP(SHOOT.OUT.NO))),
2726       TW.NAME(WPN.ID(WPN)),
2727       EQ.NAME(UE.ID(TARGET.EQUIP)),
2728       EQ.TE.PTR(UE.ID(TARGET.EQUIP)),
2729       RANGE*16,PK.DEFILADE,SHOOT.OUT.NO,
2730       UN.COLOR(FIRING.UNIT), PK AS S 1,"1",S 1,D(7,4),S 1,
2731       I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''                                                                    \TEXT
2732       I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
2733       S 4,"1A5",/
2734       USING UNIT 55
2735    ALWAYS
2736    IF UN.PTR(TARGET.UNIT) GT 0 AND
2737       UE.CRITICAL.EQUIP.INDIC(TARGET.EQUIP) = TRUE
2738       SUBTRACT 1 FROM MU.CRIT.NO(UN.PTR(TARGET.UNIT))
2739    ALWAYS
2740
2741    IF UN.BATTLE.INDEX(TARGET.UNIT) GT 0
2742       CALL CAS.EVAL--------------------------------------------------------------------------------------------------------------->(283)
2743       GIVEN
2744       UN.BATTLE.INDEX(TARGET.UNIT)
2745    ALWAYS
2746
2747 <--EXITPROCESS
2748 ENDPROCESS
```

```
2749 PROCESS SHOOT.OUT
2750
2751    ADD 1 TO ANAL.CTR(182,1)  ''
2752    NORMALLY MODE IS INTEGER
2753    DEFINE SHOOT.OUT TO MEAN PROCESS.V ''
2754    DEFINE CUM1          TO MEAN ESO.CUM1(SO.EX.SHOOT.OUT(SHOOT.OUT))
2755    DEFINE CUM2          TO MEAN ESO.CUM2(SO.EX.SHOOT.OUT(SHOOT.OUT))
2756    DEFINE DUMMY         TO MEAN ESO.DUMMY(SO.EX.SHOOT.OUT(SHOOT.OUT))
2757    DEFINE RANGE         TO MEAN ESO.RANGE(SO.EX.SHOOT.OUT(SHOOT.OUT))
2758    DEFINE SCORE1        TO MEAN ESO.SCR1(SO.EX.SHOOT.OUT(SHOOT.OUT))
2759    DEFINE SCORE2        TO MEAN ESO.SCR2(SO.EX.SHOOT.OUT(SHOOT.OUT))
2760    DEFINE TARGET        TO MEAN ESO.TGT(SO.EX.SHOOT.OUT(SHOOT.OUT))
2761    DEFINE TARGET.EQUIP  TO MEAN ESO.TGT.EQUIP(SO.EX.SHOOT.OUT(SHOOT.OUT))
2762    DEFINE TARGET.UNIT   TO MEAN ESO.TGT.UNIT(SO.EX.SHOOT.OUT(SHOOT.OUT))
2763    DEFINE WEAPON        TO MEAN ESO.WEAPON(SO.EX.SHOOT.OUT(SHOOT.OUT))
2764    DEFINE WPN           TO MEAN ESO.WPN(SO.EX.SHOOT.OUT(SHOOT.OUT))
2765    DEFINE .ACQ.TIME     TO MEAN ESO.ACQ.TIME(SO.EX.SHOOT.OUT(SHOOT.OUT))
2766    DEFINE .QUANT        TO MEAN ESO.QUANT(SO.EX.SHOOT.OUT(SHOOT.OUT))
2767    DEFINE ..TGT         TO MEAN ESO..TGT(SO.EX.SHOOT.OUT(SHOOT.OUT))
2768    DEFINE FIRE.EQUIP    TO MEAN FIRING.EQUIP(SHOOT.OUT)
2769    DEFINE FIRE.UNIT     TO MEAN FIRER.UNIT(SHOOT.OUT)
2770    DEFINE PK            TO MEAN FT.PK(FIRING.TABLE)
2771    DEFINE S.O, FT.ENTRY AS INTEGER VARIABLES
2772
2773    CREATE AN EX.SHOOT.OUT CALLED SO.EX.SHOOT.OUT(SHOOT.OUT)
2774
2775    LET DROP.DEAD.INDICATOR(SHOOT.OUT) = NO
2776    LET SO.FIRING.TABLE(SHOOT.OUT) = 0
2777    IF SO.AIR.ATK.INDIC(SHOOT.OUT) = YES
2778       LET .ACQ.TIME = TW.AC.DET.TIME(WPN.ID(F.UE.WEAPON.SET(FIRE.EQUIP)))
2779    ELSE
2780       LET .ACQ.TIME = .2
2781    ALWAYS
2782
2783 <--WAIT .ACQ.TIME SECONDS
2784
2785    ''THIS WAIT TIME REPRESENTS TARGET ACQUISITION TIME
2786    ''IT SHOULD BE HERE TO MAKE THE MODEL OPERATE BUT
2787    ''MAY BE REDUCED TO .101 SECONDS
2788    ''NEEDS TO BE DETERMINISTIC TO AVOID N.SO.LIST LT UE.QUANT
2789
2790    IF SO.DESTRUCT.INDIC(SHOOT.OUT) = YES
2791       DESTROY THE EX.SHOOT.OUT CALLED SO.EX.SHOOT.OUT(SHOOT.OUT)
2792 <------EXITPROCESS
2793    OTHERWISE
2794
2795    IF DROP.DEAD.INDICATOR(SHOOT.OUT) = YES
2796       IF UN.BATTLE.INDEX(FIRER.UNIT(SHOOT.OUT)) GT 0
2797          CALL DECIDE------------------------------------------------------------------------------------------>(324)
2798          GIVEN
2799          FIRE.UNIT,
2800          FIRE.EQUIP
2801       ALWAYS
2802       REMOVE THE SHOOT.OUT FROM SO.LIST(FIRE.EQUIP)
2803       DESTROY THE EX.SHOOT.OUT CALLED SO.EX.SHOOT.OUT(SHOOT.OUT)
2804 <------EXITPROCESS
2805    OTHERWISE
2806
```

```
2807    LOOP UNTIL UE.TARGET.LIST(FIRE.EQUIP) IS EMPTY
2808    DO THE FOLLOWING
2809       LET SO.RESULTS(SHOOT.OUT) = MISS
2810       LET CUM1 = 0
2811       LET CUM2 = 0
2812       '' THE FOLLOWING IS A SET OF DEBUG CODE                                                    \REMOVE?
2813       FOR EVERY FIRING.TABLE IN UE.TARGET.LIST(FIRE.EQUIP)
2814       WITH CT.NAME(TU.CAT(UN.TYPE.UNIT(FT.TGT.UNIT(FIRING.TABLE))))="CBTAVN"
2815       DO
2816          LIST ATTRIBUTES OF FIRING.TABLE
2817          LIST UNIT.NOS(FT.TGT.UNIT(FIRING.TABLE))
2818          WRITE AS /,/
2819       ENDLOOP
2820       '' THE END OF DEBUG STATEMENTS
2821       LOOP FOR EVERY FIRING.TABLE  OF UE.TARGET.LIST(FIRE.EQUIP)
2822       UNLESS FT.AC.ATK.TGT(FIRING.TABLE) GT 0 AND
2823       AAT.VIS.IND(FT.AC.ATK.TGT(FIRING.TABLE)) = NO
2824       DO
2825          LET SO.HELICOPTER(SHOOT.OUT) = 0
2826          LET TARGET.EQUIP = FT.TARGET.EQUIP(FIRING.TABLE)
2827          LET TARGET.UNIT = FT.TGT.UNIT(FIRING.TABLE)
2828          LET FT.PK(FIRING.TABLE) = 0.0
2829          LET FT.PK.BAR(FIRING.TABLE) = 0.0
2830          LET FT.SCORE1(FIRING.TABLE) = 0
2831          LET FT.SCORE2(FIRING.TABLE) = 0
2832          IF SO.AIR.ATK.INDIC(SHOOT.OUT) = YES AND
2833             (FT.AC.ATK.TGT(FIRING.TABLE) = 0 OR
2834             AAT.VIS.IND(FT.AC.ATK.TGT(FIRING.TABLE)) NE YES)
2835       <------CYCLE
2836          OTHERWISE
2837
2838          FOR EVERY FARRP IN FP.SET(UN.COLOR(TARGET.UNIT))
2839          WITH FP.UNIT(FARRP) = TARGET.UNIT
2840          FIND THE FIRST CASE
2841          IF FOUND
2842             LOOP FOR EVERY HC.TEAM IN HT.LIST(FARRP)
2843             DO THE FOLLOWING
2844                FOR EVERY HELICOPTER IN HT.MEMBER.LIST(HC.TEAM)
2845                WITH HC.UE.ID(HELICOPTER) =
2846                UE ID(TARGET.EQUIP) AND
2847                (HC.STATUS(HELICOPTER) = ENGAGING OR
2848                HC.STATUS(HELICOPTER) = DETECTING)
2849                FIND THE FIRST CASE
2850                IF FOUND
2851                   LET SO.HELICOPTER(SHOOT.OUT) = HELICOPTER
2852             <------LEAVE
2853                OTHERWISE
2854             ENDLOOP
2855             IF SO.HELICOPTER(SHOOT.OUT) = 0
2856                CALL ERROR.STOP------------------------------------------------------------------------------------------------------------------------>(604)
2857             ALWAYS
2858          ALWAYS
2859          IF FT.AC.ATK.TGT(FIRING.TABLE) GT 0
2860             LET .QUANT = 1
2861          ELSE
2862             IF SO.HELICOPTER(SHOOT.OUT) = 0
2863                LET .QUANT = UE.QUANT(FT.TARGET.EQUIP(FIRING.TABLE))
2864             ELSE
```

```
2865                LET .QUANT = N.HT.MEMBER.LIST(
2866                HC.BTL.TEAM(SO.HELICOPTER(SHOOT.OUT)))
2867             ALWAYS
2868          ALWAYS
2869          IF .QUANT LE 0
2870             LOOP FOR EVERY S.O. OF SO.LIST(FIRE.EQUIP)
2871             WITH SO.FIRING.TABLE(S.O.) = FIRING.TABLE
2872             DO THE FOLLOWING
2873                LET SO.FIRING.TABLE(S.O.) = 0
2874             ENDLOOP
2875             REMOVE THE FIRING.TABLE FROM UE.TARGET.LIST(FIRE.EQUIP)
2876             DESTROY THIS FIRING.TABLE
2877             IF UE.TARGET.LIST(FIRE.EQUIP) IS EMPTY
2878                REMOVE THE SHOOT.OUT FROM SO.LIST(FIRE.EQUIP)
2879                DESTROY THE EX.SHOOT.OUT
2880                CALLED SO.EX.SHOOT.OUT(SHOOT.OUT)
2881 <---------------EXITPROCESS
2882             OTHERWISE
2883       <-----CYCLE
2884          OTHERWISE
2885
2886          IF FT.AC.ATK.TGT(FIRING.TABLE) GT 0
2887             LET RANGE = AAT.RANGE(FT.AC.ATK.TGT(FIRING.TABLE))
2888          ELSE
2889             IF SO.HELICOPTER(SHOOT.OUT) > 0 AND
2890                HC.UE.ID(SO.HELICOPTER(SHOOT.OUT)) =
2891                UE.ID(FT.TARGET.EQUIP(FIRING.TABLE))
2892                CALL HEL.RANGE.COMPUTE-------------------------------------------------------------------------------->(302)
2893                GIVEN
2894                SO.HELICOPTER(SHOOT.OUT),
2895                FIRE.UNIT
2896                YIELDING
2897                RANGE
2898             ELSE
2899                CALL RANGE.COMPUTE-----------------------------------------------------------------------------------><(342)
2900                GIVEN
2901                FT.TGT.UNIT(FIRING.TABLE),
2902                FIRE.UNIT
2903                YIELDING
2904                RANGE
2905             ALWAYS
2906             CALL PROX.CHECK--------------------------------------------------------------------------------------><(115)
2907             GIVEN
2908             FIRE.UNIT,
2909             RANGE
2910          ALWAYS
2911          CALL PK.COMPUTE--------------------------------------------------------------------------------------><(149)
2912          GIVEN
2913          FIRE.EQUIP,
2914          FT.TARGET.EQUIP(FIRING.TABLE),
2915          RANGE,
2916          FIRE.UNIT,
2917          FT.TGT.UNIT(FIRING.TABLE)
2918          YIELDING
2919          FT.PK(FIRING.TABLE),
2920          WEAPON
2921          LET ..TGT = FT.TARGET.EQUIP(FIRING.TABLE)
2922          IF SO.HELICOPTER(SHOOT.OUT) GT 0 AND
```

```
2923              HC.TYPE(SO.HELICOPTER(SHOOT.OUT)) = SCOUT
2924              FOR EVERY UE.LINK OF UN.EQUIP.LIST(FP.UNIT(HT.FARRP(
2925              HC.BTL.TEAM(SO.HELICOPTER(SHOOT.OUT)))))
2926              WITH N.UE.WEAPON.SET(UE.LINK) GT 0
2927              FIND THE FIRST CASE
2928              IF NONE
2929                 CALL ERROR.STOP----------------------------------------------------------------------------->(604)
2930              ALWAYS
2931              LET ..TGT = UE.LINK
2932           ALWAYS
2933           CALL PK.COMPUTE----------------------------------------------------------------------------------->(149)
2934           GIVEN
2935           ..TGT,
2936           FIRE.EQUIP,
2937           RANGE,
2938           FT.TGT.UNIT(FIRING.TABLE),
2939           FIRE.UNIT
2940           YIELDING
2941           FT.PK.BAR(FIRING.TABLE),
2942           DUMMY
2943           LET FT.FIRING.WPN(FIRING.TABLE) = WEAPON
2944           LET SCORE1 = FT.PK(FIRING.TABLE) * .QUANT * 100
2945           LET SCORE2 = SCORE1 * FT.PK.BAR(FIRING.TABLE) * 5
2946           LET CUM1 = CUM1 + SCORE1
2947           LET CUM2 = CUM2 + SCORE2
2948           LET FT.SCORE1(FIRING.TABLE) = CUM1
2949           LET FT.SCORE2(FIRING.TABLE) = CUM2
2950        ENDLOOP
2951
2952        IF CUM2 = 0,
2953           IF CUM1 = 0,
2954
2955 <----------WAIT .5 MINUTES
2956
2957              IF SO.DESTRUCT.INDIC(SHOOT.OUT) = YES
2958                 DESTROY THE EX.SHOOT.OUT
2959                 CALLED SO.EX.SHOOT.OUT(SHOOT.OUT)
2960 <------------EXITPROCESS
2961              OTHERWISE
2962              IF DROP.DEAD.INDICATOR(SHOOT.OUT) = YES
2963                 IF UN.BATTLE.INDEX(FIRER.UNIT(SHOOT.OUT)) GT 0
2964                    CALL DECIDE------------------------------------------------------------------------------->(324)
2965                    GIVEN
2966                    FIRE.UNIT,
2967                    FIRE.EQUIP
2968                 ALWAYS
2969                 REMOVE THE SHOOT.OUT FROM SO.LIST(FIRE.EQUIP)
2970                 DESTROY THE EX.SHOOT.OUT
2971                 CALLED SO.EX.SHOOT.OUT(SHOOT.OUT)
2972 <------------EXITPROCESS
2973              OTHERWISE
2974    <---------CYCLE
2975           OTHERWISE
2976           LET TARGET = RANDI.F(1, CUM1, RN.SEED)
2977           FOR EVERY FIRING.TABLE  OF UE.TARGET.LIST(FIRE.EQUIP)
2978           WITH FT.SCORE1(FIRING.TABLE) GE TARGET,
2979           FIND THE FIRST CASE
2980           IF FOUND,
```

```
2981            LET SO.FIRING.TABLE(SHOOT.OUT) = FIRING.TABLE
2982         ELSE
2983            LET SO.FIRING.TABLE(SHOOT.OUT) =
2984            L.UE.TARGET.LIST(FIRE.EQUIP)
2985            PRINT 1 LINE WITH FIRE.EQUIP,
2986            N.UE.TARGET.LIST(FIRE.EQUIP)
2987            THUS
2988            === SHOOT OUT EQUIP •••••• HAS •• FIRING TABLES ===
2989            CALL ERROR.STOP------------------------------------------------------------------------------------------------------>(604)
2990         ALWAYS
2991         ''SHOOT AT THIS TARGET
2992      ELSE
2993         LET TARGET = RANDI.F(1, CUM2, RN.SEED)
2994         FOR EVERY FIRING.TABLE  OF UE.TARGET.LIST(FIRE.EQUIP)
2995         WITH FT.SCORE2(FIRING.TABLE) GE TARGET,
2996         FIND THE FIRST CASE
2997         IF FOUND,
2998            LET SO.FIRING.TABLE(SHOOT.OUT) = FIRING.TABLE
2999         ELSE
3000            LET SO.FIRING.TABLE(SHOOT.OUT) =
3001            L.UE.TARGET.LIST(FIRE.EQUIP)
3002            PRINT 1 LINE WITH FIRE.EQUIP,
3003            N.UE.TARGET.LIST(FIRE.EQUIP)  THUS
3004            === SHOOT OUT  FIRING EQUIP •••••• HAS ••• FIRING TABLES ===
3005            CALL ERROR.STOP------------------------------------------------------------------------------------------------------>(604)
3006         ALWAYS
3007         ''SHOOT AT THIS TARGET
3008      ALWAYS
3009
3010      LET FIRING.TABLE = SO.FIRING.TABLE(SHOOT.OUT)
3011      LET SO.HELICOPTER(SHOOT.OUT) = 0
3012      LET TARGET.EQUIP = FT.TARGET.EQUIP(FIRING.TABLE)
3013      LET TARGET.UNIT = FT.TGT.UNIT(FIRING.TABLE)
3014      FOR EVERY FARRP IN FP.SET(UN.COLOR(TARGET.UNIT))
3015      WITH FP.UNIT(FARRP) = TARGET.UNIT
3016      FIND THE FIRST CASE
3017      IF FOUND
3018         LOOP FOR EVERY HC.TEAM IN HT.LIST(FARRP)
3019         DO THE FOLLOWING
3020            FOR EVERY HELICOPTER IN HT.MEMBER.LIST(HC.TEAM)
3021            WITH HC.UE.ID(HELICOPTER) = UE.ID(TARGET.EQUIP) AND
3022            (HC.STATUS(HELICOPTER) = ENGAGING OR
3023            HC.STATUS(HELICOPTER) = DETECTING)
3024            FIND THE FIRST CASE
3025            IF FOUND
3026               LET SO.HELICOPTER(SHOOT.OUT) = HELICOPTER
3027         <------LEAVE
3028            ALWAYS
3029         ENDLOOP
3030         IF SO.HELICOPTER(SHOOT.OUT) = 0
3031            CALL ERROR.STOP------------------------------------------------------------------------------------------------------>(604)
3032         ALWAYS
3033      ALWAYS
3034
3035      IF FT.AC.ATK.TGT(FIRING.TABLE) = 0
3036         IF SO.HELICOPTER(SHOOT.OUT) > 0 AND
3037            UE.ID(TARGET.EQUIP) = HC.UE.ID(SO.HELICOPTER(SHOOT.OUT))
3038            LET HC.VISIBLE = NO
```

```
3039            LOOP FOR EVERY .HC IN
3040            HT.MEMBER.LIST(HC.BTL.TEAM(SO.HELICOPTER(SHOOT.OUT)))
3041            DO
3042               FOR EACH VISIBLE.UNIT OF UN.HC.LOS.LIST(FIRE.UNIT)
3043               WITH VU.POINTER(VISIBLE.UNIT) = .HC
3044               FIND THE FIRST CASE
3045               IF FOUND
3046                  LET HC.VISIBLE = YES
3047            <-----EXITLOOP
3048               OTHERWISE
3049            ENDLOOP
3050            IF HC.VISIBLE = NO
3051               REMOVE THE FIRING.TABLE FROM
3052               THE UE.TARGET.LIST(FIRE.EQUIP)
3053               DESTROY THE FIRING.TABLE
3054      <----------CYCLE
3055            OTHERWISE
3056         ELSE
3057            FOR EVERY VISIBLE.UNIT OF UN.LOS.LIST(FIRE.UNIT)
3058            WITH VU.POINTER(VISIBLE.UNIT) = TARGET.UNIT
3059            FIND THE FIRST CASE
3060            IF NONE,
3061               REMOVE THE FIRING.TABLE
3062               FROM THE UE.TARGET.LIST(FIRE.EQUIP)
3063               DESTROY THE FIRING.TABLE
3064      <----------CYCLE
3065            OTHERWISE
3066         ALWAYS
3067      ALWAYS
3068
3069      LOOP UNTIL SO.RESULTS(SHOOT.OUT) = HIT
3070      OR UE.TARGET.LIST(FIRE.EQUIP) IS EMPTY
3071      OR SO.FIRING.TABLE(SHOOT.OUT) = 0
3072      OR FT.FIRING.WPN(FIRING.TABLE) = 0
3073      DO THE FOLLOWING
3074         IF FT.AC.ATK.TGT(FIRING.TABLE) GT 0
3075            LET RANGE = AAT.RANGE(FT.AC.ATK.TGT(FIRING.TABLE))
3076         ELSE
3077            IF UE.QUANT(TARGET.EQUIP) LE 0 AND
3078               SO.HELICOPTER(SHOOT.OUT) = 0
3079      <--------EXITLOOP
3080            OTHERWISE
3081            IF SO.HELICOPTER(SHOOT.OUT) > 0 AND
3082               UE.ID(TARGET.EQUIP) = HC.UE.ID(SO.HELICOPTER(SHOOT.OUT))
3083               CALL HEL.RANGE.COMPUTE------------------------------------------------------------------------------------------>(302)
3084               GIVING
3085               SO.HELICOPTER(SHOOT.OUT),
3086               FIRE.UNIT
3087               YIELDING
3088               RANGE
3089            ELSE
3090               CALL RANGE.COMPUTE----------------------------------------------------------------------------------------------->(342)
3091               GIVEN
3092               FT.TGT.UNIT(FIRING.TABLE),
3093               FIRE.UNIT
3094               YIELDING
3095               RANGE
3096            ALWAYS
```

```
3097            CALL PROX.CHECK------------------------------------------------------------------------------------------>(115)
3098            GIVEN
3099            FIRE.UNIT,
3100            RANGE
3101         ALWAYS
3102         LET WPN=FT.FIRING.WPN(FIRING.TABLE)
3103         IF WPN>0
3104            ADD 1 TO STW.RND.FIRED(UN.COLOR(FIRE.UNIT),WPN.ID(WPN))
3105            IF TW.RND.WEIGHT(WPN.ID(WPN))>1000
3106               CREATE A DF.NOISE
3107               LET DF.TIME(DF.NOISE) = TIME.V
3108               LET DF.UNIT(DF.NOISE) = FIRE.UNIT
3109               FILE DF.NOISE IN DF.RATE.LIST
3110            ALWAYS
3111            IF EQ.KV.ID(UE.ID(FIRING.EQUIP(SHOOT.OUT))) GT 0
3112               ADD TW.RND.WEIGHT(WPN.ID(WPN)) TO
3113               KV.AMMO.CONSUMED(UN.COLOR(FIRER.UNIT(SHOOT.OUT)),
3114               EQ.KV.ID(UE.ID(FIRING.EQUIP(SHOOT.OUT))) )
3115            ALWAYS
3116            IF UN.COLOR(FIRE.UNIT) = BLUE
3117               ADD 1 TO STY.BLUE.EXP(WPN.ID(WPN),
3118               UE.ID(TARGET.EQUIP) - N.BLUE.TYPE.EQP)
3119            ELSE
3120               ADD 1 TO STY.RED.EXP(WPN.ID(WPN) - N.B.WPN.TYPE,
3121               UE.ID(TARGET.EQUIP))
3122            ALWAYS
3123
3124            IF ANALYSIS(6) GT 0
3125               WRITE UN.BATTLE.INDEX(FIRER.UNIT(SHOOT.OUT)),
3126               TIME.V,
3127               WPN.ID(WPN),
3128               UE.ID(FT.TARGET.EQUIP(FIRING.TABLE))
3129               AS "DIR.  ", I 7, S 2, D(7,2), S 2,
3130               I 4, S 2, I 4, /   USING UNIT 60
3131            ALWAYS
3132         ELSE
3133            TRACE
3134            CALL ERROR.STOP------------------------------------------------------------------------------------------>(604)
3135         ALWAYS
3136
3137 <-------WAIT ((RANGE * DISTANCE.INCREMENT) /
3138         TW.ROUND.VELOCITY(WPN.ID(FT.FIRING.WPN(FIRING.TABLE))))
3139         SECONDS
3140
3141         IF SO.DESTRUCT.INDIC(SHOOT.OUT) = YES
3142            IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
3143               WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(WPN)),
3144               EQ.NAME(UE.ID(FIRE.EQUIP)),
3145               EQ.TE.PTR(UE.ID(FIRE.EQUIP)),
3146               TW.NAME(WPN.ID(WPN)),
3147               EQ.NAME(UE.ID(TARGET.EQUIP)),
3148               EQ.TE.PTR(UE.ID(TARGET.EQUIP)),
3149               RANGE*16,PK.DEFILADE.SHOOT.OUT,
3150               UN.COLOR(FIRE.UNIT), PK AS S 1,"@",S 1,D(7,4),S 1,
3151               I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3,                                                    \TEXT
3152               I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
3153               S 4,"@S1",/
3154               USING UNIT 55
```

```
3155              ALWAYS
3156              DESTROY THE EX.SHOOT.OUT
3157              CALLED SO.EX.SHOOT.OUT(SHOOT.OUT)
3158 <----------EXITPROCESS
3159           OTHERWISE
3160           LET FIRING.TABLE = SO.FIRING.TABLE(SHOOT.OUT)
3161           IF FIRING.TABLE IS ZERO, ''                                                    CHG\04  \ZERO_SUB
3162        <-----LEAVE
3163           OTHERWISE
3164           LET TARGET.EQUIP = FT.TARGET.EQUIP(FIRING.TABLE)
3165           LET TARGET.UNIT = FT.TGT.UNIT(FIRING.TABLE)
3166
3167           IF DROP.DEAD.INDICATOR(SHOOT.OUT) = YES
3168              IF UN.BATTLE.INDEX(FIRER.UNIT(SHOOT.OUT)) GT 0
3169                 CALL DECIDE-------------------------------------------------------------------------------->(324)
3170                 GIVEN
3171                 FIRE.UNIT,
3172                 FIRE.EQUIP
3173              ALWAYS
3174              REMOVE THE SHOOT.OUT FROM SO.LIST(FIRE.EQUIP)
3175              IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
3176                 WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(WPN)),
3177                 EQ.NAME(UE.ID(FIRE.EQUIP)),
3178                 EQ.TE.PTR(UE.ID(FIRE.EQUIP)),
3179                 TW.NAME(WPN.ID(WPN)),
3180                 EQ.NAME(UE.ID(TARGET.EQUIP)),
3181                 EQ.TE.PTR(UE.ID(TARGET.EQUIP)),
3182                 RANGE*16,PK.DEFILADE,SHOOT.OUT,
3183                 UN.COLOR(FIRE.UNIT), PK AS S 1,"0",S 1,D(7,4),S 1,:
3184                 I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ':                        \TEXT
3185                 I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
3186                 S 4,"0S2",/
3187                 USING UNIT 55
3188              ALWAYS
3189              DESTROY THE EX.SHOOT.OUT
3190              CALLED SO.EX.SHOOT.OUT(SHOOT.OUT)
3191 <----------EXITPROCESS
3192           OTHERWISE
3193           IF FIRING.TABLE = 0
3194              IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
3195                 WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(WPN)),
3196                 EQ.NAME(UE.ID(FIRE.EQUIP)),
3197                 EQ.TE.PTR(UE.ID(FIRE.EQUIP)),
3198                 TW.NAME(WPN.ID(WPN)),
3199                 EQ.NAME(UE.ID(TARGET.EQUIP)),
3200                 EQ.TE.PTR(UE.ID(TARGET.EQUIP)),
3201                 RANGE*16,PK.DEFILADE,SHOOT.OUT,
3202                 UN.COLOR(FIRE.UNIT), PK AS S 1,"0",S 1,D(7,4),S 1,:
3203                 I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ':                        \TEXT
3204                 I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
3205                 S 4,"0S3",/
3206                 USING UNIT 55
3207              ALWAYS
3208        <-----LEAVE
3209           OTHERWISE
3210
3211           IF FT.AC.ATK.TGT(FIRING.TABLE) = 0
3212              IF SO.HELICOPTER(SHOOT.OUT) > 0 AND
```

```
3213                UE.ID(TARGET.EQUIP)=HC.UE_ID(SO.HELICOPTER(SHOOT.OUT))
3214                IF UE.TARGET.LIST(FIRE.EQUIP) IS EMPTY OR
3215                   FT.FIRING.WPN(SO.FIRING.TABLE(SHOOT.OUT)) = 0
3216                   IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
3217                      WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(WPN)),
3218                      EQ.NAME(UE.ID(FIRE.EQUIP)),
3219                      EQ.TE.PTR(UE.ID(FIRE.EQUIP)),
3220                      TW.NAME(WPN.ID(WPN)),
3221                      EQ.NAME(UE.ID(TARGET.EQUIP)),
3222                      EQ.TE.PTR(UE.ID(TARGET.EQUIP)),
3223                      RANGE*16,PK.DEFILADE,SHOOT.OUT,
3224                      UN.COLOR(FIRE.UNIT), PK AS S 1,"0",S 1,D(7,4),S 1,
3225                      I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''                    \TEXT
3226                      I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
3227                      S 4,"0S4",/
3228                      USING UNIT 55
3229                   ALWAYS
3230          <----------EXITLOOP
3231                OTHERWISE
3232             ELSE
3233                IF UE.QUANT(TARGET.EQUIP) LE 0
3234                   OR UE.TARGET.LIST(FIRE.EQUIP) IS EMPTY
3235                   OR FT.FIRING.WPN(SO.FIRING.TABLE(SHOOT.OUT)) = 0
3236          <----------LEAVE
3237                OTHERWISE
3238             ALWAYS
3239          ALWAYS
3240          FOR EVERY FT.ENTRY OF UE.TARGET.LIST(FIRE.EQUIP)
3241          WITH FT.TGT.UNIT(FT.ENTRY) = TARGET.UNIT
3242          FIND THE FIRST CASE
3243          IF NONE,
3244             IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
3245                WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(WPN)),
3246                EQ.NAME(UE.ID(FIRE.EQUIP)),
3247                EQ.TE.PTR(UE.ID(FIRE.EQUIP)),
3248                TW.NAME(WPN.ID(WPN)),
3249                EQ.NAME(UE.ID(TARGET.EQUIP)),
3250                EQ.TE.PTR(UE.ID(TARGET.EQUIP)),
3251                RANGE*16,PK.DEFILADE,SHOOT.OUT,
3252                UN.COLOR(FIRE.UNIT), PK AS S 1,"0",S 1,D(7,4),S 1,
3253                I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''                    \TEXT
3254                I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
3255                S 4,"0S5",/
3256                USING UNIT 55
3257             ALWAYS
3258          <-----LEAVE
3259          OTHERWISE
3260          CALL PK.COMPUTE------------------------------------------------------------------------------------------------------------>(149)
3261          GIVEN
3262          FIRE.EQUIP,
3263          FT.TARGET.EQUIP(FIRING.TABLE),
3264          RANGE,
3265          FIRE.UNIT,
3266          FT.TGT.UNIT(FIRING.TABLE)
3267          YIELDING
3268          FT.PK(FIRING.TABLE),
3269          DUMMY
3270          IF DUMMY = 0,
```

```
3271        <-----LEAVE
3272           OTHERWISE
3273           ACTIVATE_AN ASSESSMENT------------------------------------------------------------------------------------------>(487)
3274           GIVEN
3275           FT.PK(FIRING.TABLE),
3276           TARGET.EQUIP,
3277           TARGET.UNIT,
3278           FIRE.UNIT,
3279           SHOOT.OUT NOW ''                                                                                                \1
3280
3281           IF SO.AIR.ATK.INDIC(SHOOT.OUT) = YES
3282 <---------WAIT 10/TW.ROF.AIR(WPN.ID(FT.FIRING.WPN(FIRING.TABLE)))
3283              MINUTES
3284           ELSE
3285              IF TW.RATE.OF.FIRE(WPN.ID(FT.FIRING.WPN(FIRING.TABLE)))
3286                 LE 0
3287                 PRINT 1 LINE WITH
3288                 TW.NAME(WPN.ID(FT.FIRING.WPN(FIRING.TABLE))) THUS
3289                 = = = SHOOT.OUT ****** HAS RATE OF FIRE LE 0
3290                 CALL ERROR.STOP------------------------------------------------------------------------------------------>(604)
3291              ALWAYS
3292 <---------WAIT 10/TW.RATE.OF.FIRE(WPN.ID(FT.FIRING.WPN(FIRING.TABLE)))
3293              MINUTES
3294           ALWAYS
3295
3296           IF SO.DESTRUCT.INDIC(SHOOT.OUT) = YES
3297              IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
3298                 WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(WPN)),
3299                 EQ.NAME(UE.ID(FIRE.EQUIP)),
3300                 EQ.TE.PTR(UE.ID(FIRE.EQUIP)),
3301                 TW.NAME(WPN.ID(WPN)),
3302                 EQ.NAME(UE.ID(TARGET.EQUIP)),
3303                 EQ.TE.PTR(UE.ID(TARGET.EQUIP)),
3304                 RANGE*16,PK.DEFILADE,SHOOT.OUT,
3305                 UN.COLOR(FIRE.UNIT), PK AS S 1,"0",S 1,D(7,4),S 1,
3306                 I 5,S 1,I 6,S 3,I 1,S 3,I 6,S 3,I 6,S 3,I 1,S 3, ''                                                     \TEXT
3307                 I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
3308                 S 4,"0S6",/
3309                 USING UNIT 55
3310              ALWAYS
3311              DESTROY THE EX.SHOOT.OUT
3312              CALLED SO.EX.SHOOT.OUT(SHOOT.OUT)
3313 <---------EXITPROCESS
3314           OTHERWISE
3315           LET FIRING.TABLE = SO.FIRING.TABLE(SHOOT.OUT)
3316           IF FIRING.TABLE IS ZERO, ''                                                                  CHG\15  \ZERO_SUB
3317        <-----LEAVE ''                                                                                    HUGH\CHECKOUT
3318           OTHERWISE
3319           LET TARGET.EQUIP = FT.TARGET.EQUIP(FIRING.TABLE)
3320           LET TARGET.UNIT = FT.TGT.UNIT(FIRING.TABLE)
3321
3322           IF DROP.DEAD.INDICATOR(SHOOT.OUT) = YES
3323              IF UN.BATTLE.INDEX(FIRER.UNIT(SHOOT.OUT)) GT 0
3324                 CALL DECIDE------------------------------------------------------------------------------------------>(324)
3325                 GIVEN
3326                 FIRE.UNIT,
3327                 FIRE.EQUIP
3328              ALWAYS
```

```
3329              REMOVE THE SHOOT.OUT FROM SO.LIST(FIRE.EQUIP)
3330              DESTROY THE EX.SHOOT.OUT
3331              CALLED SO.EX.SHOOT.OUT(SHOOT.OUT)
3332 <------------EXITPROCESS
3333           OTHERWISE
3334
3335           FOR EVERY FT.ENTRY OF UE.TARGET.LIST(FIRE.EQUIP)
3336           WITH FT.TGT.UNIT(FT.ENTRY) = TARGET.UNIT
3337           FIND THE FIRST CASE
3338           IF NONE,
3339        <------LEAVE
3340           OTHERWISE
3341        ENDLOOP
3342     ENDLOOP
3343
3344     REMOVE THE SHOOT.OUT FROM SO.LIST(FIRE.EQUIP)
3345     DESTROY THE EX.SHOOT.OUT
3346     CALLED SO.EX.SHOOT.OUT(SHOOT.OUT)
3347
3348 <--EXITPROCESS
3349 ENDPROCESS
```

```
3350 PROCESS CAS.MISSION                                                    ''                                                    P018
3351    ADD 1 TO ANAL.CTR(183,1)  ''                                                                                          \DYN_ANAL
3352    NORMALLY MODE IS INTEGER
3353    DEFINE .CAS.MISSION TO MEAN PROCESS.V ''                                                                               CHG\01
3354    DEFINE .SIDE         TO MEAN CMSN.SIDE(.CAS.MISSION)
3355    DEFINE .AC.TYPE      TO MEAN CMSN.AC.TYPE(.CAS.MISSION)
3356    DEFINE .NR.AC        TO MEAN CMSN.NR.AC(.CAS.MISSION)
3357    DEFINE .NR.SURV.AC   TO MEAN CMSN.NR.SURV.AC(.CAS.MISSION)
3358    DEFINE .NR.ABORTED   TO MEAN CMSN.NR.ABORTED(.CAS.MISSION)
3359    DEFINE .ASP.STATUS   TO MEAN CMSN.ASP.STATUS(.CAS.MISSION)
3360    DEFINE .TGT          TO MEAN CMSN.TGT.UNIT(.CAS.MISSION)
3361    DEFINE .FLIGHT.TIME TO MEAN CMSN.FLIGHT.TIME(.CAS.MISSION)
3362    DEFINE .SEG.BEGIN.TIME, .INTERVAL.BW.ATK, .ANGLE.P1.TO.ADU
3363    AS REAL VARIABLES
3364    DEFINE .SCOREBOARD AS A 2-DIM INTEGER ARRAY  ''                                                                        \VAX
3365
3366    IF TACAIR.DEBUG = 1
3367       PRINT 1 LINE WITH TIME.V, .CAS.MISSION THUS
3368       = = =CAS.MISSION CALLED AT ***.***   POINTER = ******
3369       LIST ATTRIBUTES OF CAS.MISSION CALLED .CAS.MISSION
3370       LIST SD.SORTIES.THIS.TP(.SIDE)
3371    ALWAYS
3372    LET CMSN.START.TIME(.CAS.MISSION) = TIME.V
3373
3374    FOR EVERY .AC.LINK IN UN.EQUIP.LIST(SD.AIRFIELD(.SIDE))
3375    WITH UE.ID(.AC.LINK) = ACT.EQUIP.ID(.AC.TYPE)
3376    FIND THE FIRST CASE
3377    IF NONE
3378       TRACE
3379 <------STOP
3380    OTHERWISE
3381
3382    RESERVE .SCOREBOARD AS N.UE.WEAPON.SET(.AC.LINK) BY
3383    N.UN.EQUIP.LIST(.TGT)
3384    LET CMSN.SCOREBOARD(.CAS.MISSION) = .SCOREBOARD(*,*)
3385
3386    ''CREATE AN OLD.SORTIE TO PROVIDE A CHECK ON THE SORTIE RATE.
3387    CREATE AN OLD.SORTIE
3388    FILE OLD.SORTIE IN THE SD.OLD.SORTIE.QUEUE(.SIDE)
3389    LET OS.QTY(OLD.SORTIE) = .NR.AC
3390    SCHEDULE_A DQ.OLD.SORTIE.QUEUE------------------------------------------------------------------------------------------>(367)
3391    GIVEN
3392    .SIDE
3393    IN SD.TP.SORTIE(.SIDE) MINUTES
3394    ADD .NR.AC TO SD.SORTIES.THIS.TP(.SIDE)
3395
3396    ''PREPARE FOR TAKEOFF.
3397    LET .PREP = UNIFORM.F(REAL.F(ACT.MIN.PREP.TIME(.AC.TYPE)),
3398    REAL.F(ACT.MAX.PREP.TIME(.AC.TYPE)), RN.SEED)
3399 <---WAIT .PREP MINUTES
3400
3401    LET .NR.ABORTED = BINOMIAL.F(.NR.AC,
3402    ACT.PROB.SORTIE.ABORT(.AC.TYPE) / 100., RN.SEED)
3403    LET .NR.SURV.AC = .NR.AC - .NR.ABORTED
3404    IF .NR.SURV.AC LE 0
3405       CALL END.CAS.MISSION-------------------------------------------------------------------------------------------------->(293)
3406       GIVEN
3407       .CAS.MISSION,
```

```
3408        1
3409        IF .CAS.MISSION IS IN THE SD.CMSN.QUEUE
3410           REMOVE .CAS.MISSION FROM THE SD.CMSN.QUEUE(.SIDE)
3411        ALWAYS
3412 <------RETURN
3413     OTHERWISE
3414
3415     LET .ASP.STATUS = BUSY
3416     CALL FLIGHT.PATH------------------------------------------------------------------------------------------>(314)
3417     GIVEN
3418     .CAS.MISSION
3419     LET .SEG.BEGIN.TIME = 0.0
3420     IF TACAIR.DEBUG = 1
3421        LIST ATTRIBUTES OF EACH CFP.SEGMENT IN CFPS.LIST(.CAS.MISSION)
3422     ALWAYS
3423     LOOP FOR EACH .SEG OF CFPS.LIST(.CAS.MISSION)
3424     DO
3425        ADD CFPS.TIME.LENGTH(.SEG) TO .FLIGHT.TIME
3426        IF TACAIR.DEBUG = 1
3427           LIST ATTRIBUTES OF EACH SENSOR.INTERSECT IN SI.LIST(.SEG)
3428        ALWAYS
3429        IF TACAIR.DEBUG = 1
3430           LOOP FOR EACH .PROCESS IN EV.S(I.CAS.MISSION),
3431           DO   ''                                                                                    \1
3432              PRINT 1 LINE WITH .PROCESS THUS
3433              $$$$$$$$$$ PROCESS IN EV.S(I.CAS.MISSION) IS ********** $$$$$$$
3434              LIST ATTRIBUTES OF CAS.MISSION CALLED .PROCESS
3435           ENDLOOP
3436        ALWAYS
3437        LOOP FOR EACH .INTERSECT OF SI.LIST(.SEG)
3438        DO
3439           SCHEDULE_AN AD.ENGAGEMENT------------------------------------------------------------------->(349)
3440           GIVEN
3441           .INTERSECT,
3442           .CAS.MISSION
3443           IN .SEG.BEGIN.TIME
3444           + SI.TIME.TIL.INTERSECT(.INTERSECT) HOURS
3445        ENDLOOP
3446        ADD CFPS.TIME.LENGTH(.SEG) TO .SEG.BEGIN.TIME
3447     ENDLOOP
3448     LET CMSN.TAKE.OFF.TIME(.CAS.MISSION) = TIME.V
3449
3450     ''THERE MUST BE ENOUGH TIME TO MAKE AT LEAST ONE SHOOTING PASS.
3451     LET .TOTAL.TIME = 2 * .FLIGHT.TIME * 60 + ACT.PASS.TIME(.AC.TYPE)  ''                          \OPTIMIZE
3452     IF UN.BATTLE.INDEX(.TGT) LE 0
3453        ''THE TARGET IS NOT IN A BATTLE SO THERE IS NO GROUND
3454        ''CONTROLLER. SEVERAL NON-SHOOTING PASSES MUST BE MADE TO
3455        ''FIND THE TARGET.
3456        LET .TOTAL.TIME = .TOTAL.TIME
3457        + ACT.PASS.TIME(.AC.TYPE) * (1 + TRUNC.F(
3458        ACT.BAI.TA.DELAY(.AC.TYPE) / ACT.PASS.TIME(.AC.TYPE)))
3459     ALWAYS
3460     IF .TOTAL.TIME GT ACT.MAX.ALOFT(.AC.TYPE)
3461        CALL END.CAS.MISSION------------------------------------------------------------------------->(293)
3462        GIVEN
3463        .CAS.MISSION,
3464        2
3465        IF .CAS.MISSION IS IN THE SD.CMSN.QUEUE
```

```
3466          REMOVE .CAS.MISSION FROM THE SD.CMSN.QUEUE(.SIDE)
3467       ALWAYS
3468 <------RETURN
3469    OTHERWISE
3470
3471    ''FLY TO THE TARGET.
3472 <--WAIT .FLIGHT.TIME HOURS
3473
3474    IF TACAIR.DEBUG = 1
3475       PRINT 1 LINE WITH TIME.V THUS
3476       = = =CAS.MISSION AT TARGET AT ***.***
3477       LIST ATTRIBUTES  OF CAS.MISSION CALLED .CAS.MISSION
3478    ALWAYS
3479    IF .NR.SURV.AC LE 0
3480       ''ALL THE AIRCRAFT WERE SHOT DOWN BY AREA AIR DEFENSE UNITS.
3481       CALL END.CAS.MISSION------------------------------------------------------------------------------------------>(293)
3482       GIVEN
3483       .CAS.MISSION,
3484       3
3485       IF .CAS.MISSION IS IN THE SD.CMSN.QUEUE
3486          REMOVE .CAS.MISSION FROM THE SD.CMSN.QUEUE(.SIDE)
3487       ALWAYS
3488 <------RETURN
3489    OTHERWISE
3490
3491    ''SET UP FOR THE ATTACK. DETERMINE IF THE TARGET UNIT WILL BE
3492    ''HELPED BY A NEARBY AREA AD UNIT.
3493    LET .MIN.RANGE = INF.C
3494    LOOP FOR EACH .SENSOR OF SD.ADS.SET(UN.COLOR(.TGT))
3495    WITH CT.NAME(TU.CAT(UN.TYPE.UNIT(ADS.UNIT.PTR
3496    (.SENSOR)))) = "AIRDEF" AND
3497    ADS.UNIT.PTR(.SENSOR) NE .TGT
3498    DO
3499       CALL RANGE.COMPUTE--------------------------------------------------------------------------------------------->(342)
3500       GIVEN
3501       .TGT,
3502       ADS.UNIT.PTR(.SENSOR)
3503       YIELDING
3504       .RANGE
3505       IF .RANGE LT .MIN.RANGE AND
3506          .RANGE LT MRH.RANGE(L.MADS.RH.SET(ADS.MADS.PTR(
3507          .SENSOR)))
3508          LET .MIN.RANGE = .RANGE
3509          LET .AD.UNIT = ADS.UNIT.PTR(.SENSOR)
3510       ALWAYS
3511    ENDLOOP
3512    IF .AD.UNIT GT 0
3513       ''SET THE RANGES FROM THE PASS POINTS TO THE AD UNIT.
3514       LET CMSN.P1.ADU.RANGE(.CAS.MISSION) = SQRT.F((REAL.F(
3515       (UN.X.COORD(.TGT)+ACT.X1(.AC.TYPE)
3516       - UN.X.COORD(.AD.UNIT))**2 +  ''                                                    \OPTIMIZE
3517       (UN.Y.COORD(.TGT)+ACT.Y1(.AC.TYPE)
3518       - UN.Y.COORD(.AD.UNIT))**2 +
3519       ACT.Z1(.AC.TYPE)**2)))
3520       LET CMSN.P3.ADU.RANGE(.CAS.MISSION) = SQRT.F((REAL.F(
3521       (UN.X.COORD(.TGT)+ACT.X3(.AC.TYPE)
3522       - UN.X.COORD(.AD.UNIT))**2 +
3523       (UN.Y.COORD(.TGT)+ACT.Y3(.AC.TYPE)
```

```
3524        - UN.Y.COORD(.AD.UNIT))**2 +
3525        ACT.Z3(.AC.TYPE)**2)))
3526        LET .DELTAX = UN.X.COORD(.TGT)+ACT.X1(.AC.TYPE)
3527        - UN.X.COORD(.AD.UNIT)
3528        LET .DELTAY = UN.Y.COORD(.TGT)+ACT.Y1(.AC.TYPE)
3529        - UN.Y.COORD(.AD.UNIT)
3530        IF .DELTAX = 0
3531           IF .DELTAY LT 0
3532              LET .ANGLE.P1.TO.ADU = - PI.C / 2.
3533           ELSE
3534              LET .ANGLE.P1.TO.ADU =   PI.C / 2.
3535           ALWAYS
3536        ELSE
3537           IF .DELTAY = 0
3538              IF .DELTAX  LT 0
3539                 LET .ANGLE.P1.TO.ADU = PI.C
3540              ELSE
3541                 LET .ANGLE.P1.TO.ADU = 0.0
3542              ALWAYS
3543           ELSE
3544              LET .ANGLE.P1.TO.ADU = ARCTAN.F(REAL.F(.DELTAY),
3545              REAL.F(.DELTAX))
3546           ALWAYS
3547        ALWAYS
3548        LET CMSN.ANGLE(.CAS.MISSION) = ACT.ANGLE.P1.P2(.AC.TYPE)
3549        - .ANGLE.P1.TO.ADU
3550     ALWAYS
3551
3552     ''START THE ATTACKS. ALLOW THE TARGET TO FIRE BACK BY
3553     ''INITIATING SHOOT.OUTS.
3554     LET .INTERVAL.BW.ATK = ACT.PASS.TIME(.AC.TYPE) / .NR.SURV.AC
3555     LOOP FOR .I = 1 TO .NR.SURV.AC
3556     DO
3557        ACTIVATE_AN AC.ATK.TGT------------------------------------------------------------------------------------------------------>(419)
3558        GIVEN
3559        .CAS.MISSION,
3560        .AD.UNIT,
3561        0,
3562        NO,
3563        NO
3564        IN (.I-1)*.INTERVAL.BW.ATK MINUTES
3565        LOOP FOR EACH .LINK IN UN.EQUIP.LIST(.TGT)
3566        WITH UE.QUANT(.LINK) GT 0 AND
3567        N.UE.WEAPON.SET(.LINK) GT 0
3568        DO
3569           IF TACAIR.DEBUG = 1
3570              LIST N.SO.LIST(.LINK), UE.QUANT(.LINK)
3571           ALWAYS
3572           CREATE A FIRING.TABLE CALLED .FT
3573           FILE .FT IN UE.TARGET.LIST(.LINK)
3574           LET FT.TGT.UNIT(.FT) = SD.AIRFIELD(.SIDE)
3575           LET FT.TARGET.EQUIP(.FT) = .AC.LINK
3576           LET FT.AC.ATK.TGT(.FT) = AC.ATK.TGT
3577           LET FT.CAS.MISSION(.FT) = .CAS.MISSION
3578           IF N.SO.LIST(.LINK) LT UE.QUANT(.LINK)
3579              LOOP FOR .N = (N.SO.LIST(.LINK) + 1) TO UE.QUANT(.LINK)
3580              DO
3581                 ACTIVATE_A SHOOT.OUT NOW------------------------------------------------------------------------------------------>(493)
```

```
3582                FILE THIS SHOOT.OUT IN SO.LIST(.LINK)
3583                LET FIRING.EQUIP(SHOOT.OUT) = .LINK
3584                LET FIRER.UNIT(SHOOT.OUT) = .TGT
3585             ENDLOOP
3586          ALWAYS
3587          ''WHEN THE FIRING EQUIP IS AD, SET IT TO FIRE AT THE                    \1
3588          ''AIRCRAFT ONLY.
3589          FOR EVERY .SO IN SO.LIST(.LINK)
3590          LET SO.AIR.ATK.INDIC(.SO) =
3591          EQ.AD.INDICATOR(UE.ID(.LINK))
3592       ENDLOOP
3593    ENDLOOP
3594
3595    LET CMSN.FIRST.PASS.TIME(.CAS.MISSION) = TIME.V
3596
3597    ''THE PROCESS WILL BE REACTIVATED BY THE LAST OF ITS AC.ATK.TGTS.
3598 <--WAIT 24 HOURS
3599
3600    IF TACAIR.DEBUG = 1
3601       PRINT 1 LINE WITH TIME.V THUS
3602       = = =CAS.MISSION ATTACK COMPLETE     AT ***.***
3603       LIST ATTRIBUTES OF CAS.MISSION CALLED .CAS.MISSION
3604    ALWAYS
3605    ''FREE THE AIRSPACE.
3606    IF UN.BATTLE.INDEX(.TGT) GT 0
3607       FOR EVERY FORCE IN BTL.FORCE.SET(UN.BATTLE.INDEX(.TGT))
3608       WITH FR.SIDE(FORCE) = UN.COLOR(.TGT)
3609       LET FR.CAS.INDIC(FORCE) = NO
3610    ALWAYS
3611    LET .ASP.STATUS = IDLE
3612    CALL DQ.CMSN.QUEUE----------------------------------------------------------------------------------------------------->(327)
3613
3614    IF .NR.SURV.AC = 0
3615       CALL END.CAS.MISSION-------------------------------------------------------------------------------------------------->(293)
3616       GIVEN
3617       .CAS.MISSION,
3618       4
3619       IF .CAS.MISSION IS IN THE SD.CMSN.QUEUE
3620          REMOVE .CAS.MISSION FROM THE SD.CMSN.QUEUE(.SIDE)
3621       ALWAYS
3622 <-----RETURN
3623    OTHERWISE
3624
3625    ''FLY BACK HOME. AD UNITS MAY AGAIN CAUSE LOSSES.
3626    LET .SEG.BEGIN.TIME = 0.0
3627    LOOP FOR EACH .SEG OF CFPS.LIST(.CAS.MISSION) IN REVERSE ORDER
3628    DO
3629       LOOP FOR EACH .INTERSECT IN SI.LIST(.SEG) IN REVERSE ORDER
3630       DO
3631          SCHEDULE_AN AD.ENGAGEMENT------------------------------------------------------------------------------------------->(349)
3632          GIVEN
3633          .INTERSECT,
3634          .CAS.MISSION
3635          IN .SEG.BEGIN.TIME +
3636          SI.TIME.TIL.INTERSECT(.INTERSECT) HOURS
3637       ENDLOOP
3638       ADD CFPS.TIME.LENGTH(.SEG) TO .SEG.BEGIN.TIME
3639    ENDLOOP
```

```
3640
3641 <—WAIT .FLIGHT.TIME HOURS
3642
3643    IF TACAIR.DEBUG = 1
3644       PRINT 1 LINE WITH TIME.V THUS
3645       = = =CAS.MISSION RETURNS TO AIRFIELD AT ***.***
3646       LIST ATTRIBUTES OF CAS.MISSION CALLED .CAS.MISSION
3647    ALWAYS
3648    CALL END.CAS.MISSION————————————————————————————————————————————————————————————————————————————————————————————>(293)
3649    GIVEN
3650    .CAS.MISSION,
3651    5
3652
3653    IF .CAS.MISSION IS IN THE SD.CMSN.QUEUE
3654       REMOVE .CAS.MISSION FROM THE SD.CMSN.QUEUE(.SIDE)
3655    ALWAYS
3656 <—RETURN
3657 END
```

```
3658 PROCESS HELICOPTER.FIRE                                                    ''                    P019
3659     GIVEN
3660     .ATK.HELICOPTER
3661
3662     ADD 1 TO ANAL.CTR(184,1)  ''                                                                 \DYN_ANAL
3663     ''THIS 'PROCESS' ROUTINE MODELS THE ATTACK HELICOPTER'S SHOTS
3664     ''AT, AND ASSESSMENTS OF ENEMY GROUND UNITS.  THE GROUND UNITS FIRE
3665     ''ON THE ATTACK HELICOPTERS IS MODELED IN THE PROCESS CALLED SHOOT.OUT.
3666
3667     NORMALLY MODE IS INTEGER
3668     DEFINE ATK.TEAM AND .DUMMY.WPN AS INTEGER VARIABLES
3669     DEFINE .PK, .PK.BAR AS REAL VARIABLES
3670     DEFINE HELICOPTER.FIRE TO MEAN PROCESS.V ''                                                  CHG\01
3671     DEFINE FLY TO MEAN WORK
3672     DEFINE .TIME AS A REAL VARIABLE
3673     DEFINE .TGT.SCORE AS A REAL VARIABLE
3674     DEFINE .HIT.CHANCE AS A REAL VARIABLE
3675     DEFINE .NAME AS A TEXT VARIABLE ''                                                           \TEXT
3676
3677     LET ATK.TEAM = HF.TEAM(HELICOPTER.FIRE)
3678     LET HC.UNIT = FP.UNIT(HT.FARRP(HF.TEAM(HELICOPTER.FIRE)))
3679     FOR EACH .HELO.LINK IN THE UN.EQUIP.LIST(HC.UNIT)
3680     WITH UE.ID(.HELO.LINK) = HC.UE.ID(.ATK.HELICOPTER)
3681     FIND THE FIRST CASE
3682     IF NONE
3683        TRACE
3684 <-------STOP
3685     ALWAYS
3686
3687     IF HEL.IS.KILLED.INDIC(HELICOPTER.FIRE) = YES
3688        OR HT.TERMINATOR(ATK.TEAM) = YES
3689        OR HF.DESTRUCT.INDIC(HELICOPTER.FIRE) = YES
3690        OR HF.REINFORCE.IND(HELICOPTER.FIRE) = YES
3691        REMOVE THE HELICOPTER.FIRE FROM THE HF.SO.LIST(.HELO.LINK)
3692 <-------EXITPROCESS
3693     OTHERWISE
3694
3695     IF UN.COLOR(HC.UNIT) = RED
3696        LET .ROUNDS = RD.ROUNDS.PER.POPUP
3697     ELSE ''THE SIDE IS BLUE
3698        LET .ROUNDS = BL.ROUNDS.PER.POPUP
3699     ALWAYS
3700
3701     IF HC.DEBUG = "YES"
3702        PRINT 2 LINES WITH .ATK.HELICOPTER, ATK.TEAM,
3703        HC.ALTITUDE(.ATK.HELICOPTER),
3704        HC.STATUS(.ATK.HELICOPTER) AS FOLLOWS
3705        ----HELO.FIRE(1)---- ATK.HC=****** TEAM=******
3706        HC.ALT=***  HC.STATUS=***
3707     ALWAYS
3708
3709 <--WAIT .001 SECONDS
3710
3711     IF HEL.IS.KILLED.INDIC(HELICOPTER.FIRE) = YES
3712        OR HT.TERMINATOR(ATK.TEAM) = YES
3713        OR HF.DESTRUCT.INDIC(HELICOPTER.FIRE) = YES
3714        OR HF.REINFORCE.IND(HELICOPTER.FIRE) = YES
3715        ''THE BATTLE IS TERMINATED
```

```
3716       REMOVE THE HELICOPTER.FIRE FROM THE HF.SO.LIST(.HELO.LINK)
3717 <-----EXITPROCESS
3718    OTHERWISE
3719
3720    LET HC.ALTITUDE(.ATK.HELICOPTER) = UNMASKED
3721    LET HC.STATUS(.ATK.HELICOPTER) = ENGAGING
3722    LOOP UNTIL UE.TARGET.LIST(.HELO.LINK) IS EMPTY
3723    DO THIS
3724       LET HF.RESULTS(HELICOPTER.FIRE) = MISS
3725       LET .CUM1 = 0
3726       LET .CUM2 = 0
3727       LOOP FOR EACH .FIRE.TAB OF THE UE.TARGET.LIST(.HELO.LINK)
3728       DO THE FOLLOWING
3729          IF UE.QUANT(FT.TARGET.EQUIP(.FIRE.TAB)) LE 0
3730             LOOP FOR EVERY .HEL.FIRE OF THE HF.SO.LIST(.HELO.LINK)
3731             WITH HF.FIRING.TABLE(.HEL.FIRE) = .FIRE.TAB
3732             DO THIS
3733                LET HF.FIRING.TABLE(.HEL.FIRE) = 0
3734             ENDLOOP
3735             REMOVE THIS .FIRE.TAB FROM UE.TARGET.LIST(.HELO.LINK)
3736             DESTROY THIS FIRING.TABLE CALLED .FIRE.TAB
3737             IF UE.TARGET.LIST(.HELO.LINK) IS EMPTY
3738                REMOVE THE HELICOPTER.FIRE FROM
3739                HF.SO.LIST(.HELO.LINK)
3740                IF HC.DEBUG = "YES"
3741                   PRINT 1 LINE WITH
3742                   .ATK.HELICOPTER
3743                   AS FOLLOWS
3744                   ----HELO.FIRE(3)----  ATK.HELICOPTER=******
3745                ALWAYS
3746 <--------------EXITPROCESS
3747             OTHERWISE
3748
3749       <-----CYCLE ''TO NEXT FIRING.TABLE
3750          OTHERWISE
3751
3752          CALL HEL.RANGE.COMPUTE-------------------------------------------------------------------------------------------->(302)
3753          GIVEN
3754          .ATK.HELICOPTER,
3755          FT.TGT.UNIT(.FIRE.TAB)
3756          YIELDING
3757          ..RANGE
3758          IF HC.DEBUG = "YES"
3759             PRINT 5 LINES WITH
3760             EQ.NAME(UE.ID(.HELO.LINK)),
3761             EQ.NAME(UE.ID(FT.TARGET.EQUIP(.FIRE.TAB))),
3762             (..RANGE * 16),
3763             HC.UNIT,
3764             FT.TGT.UNIT(.FIRE.TAB)
3765             THUS
3766             -----HELO$FIRE(3.5)    .HELO.LINK IS ********* ----------
3767             .TGT.EQUIP   IS **********
3768             ..RANGE      IS ****** METERS
3769             HC.UNIT      IS ******
3770             FT.TGT.UNIT  IS ******
3771          ALWAYS
3772
3773          CALL PK.COMPUTE GIVEN--------------------------------------------------------------------------------------------->(149)
```

```
3774          .HELO.LINK,
3775          FT.TARGET.EQUIP(.FIRE.TAB),
3776          ..RANGE,
3777          HC.UNIT,
3778          FT.TGT.UNIT(.FIRE.TAB)
3779          YIELDING
3780          .PK,
3781          ..WEAPON
3782          CALL PK.COMPUTE GIVEN----------------------------------------------------------------------------------------------->(149)
3783          FT.TARGET.EQUIP(.FIRE.TAB),
3784          .HELO.LINK,
3785          ..RANGE,
3786          FT.TGT.UNIT(.FIRE.TAB),
3787          HC.UNIT
3788          YIELDING
3789          .PK.BAR,
3790          .DUMMY.WPN
3791          LET FT.FIRING.WPN(.FIRE.TAB) = ..WEAPON
3792          LET FT.PK(.FIRE.TAB) = .PK
3793          LET FT.PK.BAR(.FIRE.TAB) = .PK.BAR
3794          IF WPN.ROUNDS.REMAINING(..WEAPON) GT 0
3795             LET .SCORE1 = .PK * 100 *
3796             UE.QUANT(FT.TARGET.EQUIP(.FIRE.TAB))
3797          ELSE
3798             LET .SCORE1 = 0
3799          ALWAYS
3800          LET .SCORE2 = .SCORE1 * .PK.BAR * 5
3801          LET .CUM1 = .CUM1 + .SCORE1
3802          LET .CUM2 = .CUM2 + .SCORE2
3803          LET FT.SCORE1(.FIRE.TAB)=.CUM1
3804          LET FT.SCORE2(.FIRE.TAB)=.CUM2
3805          IF HC.DEBUG = "YES"
3806             IF ..WEAPON = 0
3807                LET .NAME = " "
3808             ELSE
3809                LET .NAME = TW.NAME(WPN.ID(..WEAPON))
3810             ALWAYS
3811             PRINT 2 LINES WITH .NAME,
3812             ..WEAPON, .PK, .PK.BAR,
3813             .SCORE1, .SCORE2,
3814             .CUM1, .CUM2
3815             AS FOLLOWS
3816             ---HELIFOPTER$F  ..WEAPON=****** (******) .PK=**.*** .PK.BAR=**.***
3817             .SCORE1=*****  .SCORE2=*****  .CUM1=*****  .CUM2=*****
3818          ALWAYS
3819       ENDLOOP
3820
3821       IF .CUM2 = 0
3822          IF .CUM1 = 0
3823
3824             FLY 30 SECONDS  ''                                                    \FIX_LATER
3825
3826             IF HEL.IS.KILLED.INDIC(HELICOPTER.FIRE) = YES
3827                IF HC.DEBUG = "YES"
3828                   PRINT 1 LINE WITH
3829                   .ATK.HELICOPTER
3830                   AS FOLLOWS
3831                   ----HELO.FIRE(5)----  .ATK.HELICOPTER ****** IS DEAD
```

```
3832                ALWAYS
3833             ALWAYS
3834             IF HEL.IS.KILLED.INDIC(HELICOPTER.FIRE) = YES
3835                OR HT.TERMINATOR(ATK.TEAM) = YES
3836                OR HF.DESTRUCT.INDIC(HELICOPTER.FIRE) = YES
3837                OR HF.REINFORCE.IND(HELICOPTER.FIRE) = YES
3838                REMOVE THE HELICOPTER.FIRE FROM THE
3839                HF.SO.LIST(.HELO.LINK)
3840 <--------------EXITPROCESS
3841             OTHERWISE
3842
3843    <--------CYCLE
3844          OTHERWISE
3845
3846          LET .TGT.SCORE = RANDI.F(1, .CUM1, RN.SEED)
3847          FOR EVERY ..FIRING.TABLE OF UE.TARGET.LIST(.HELO.LINK)
3848          WITH FT.SCORE1(..FIRING.TABLE) GE .TGT.SCORE
3849          FIND THE FIRST CASE
3850          IF NONE
3851             TRACE
3852             CALL ERROR.STOP-------------------------------------------------------------------------------------->(604)
3853          ELSE
3854             LET HF.FIRING.TABLE(HELICOPTER.FIRE) = ..FIRING.TABLE
3855          ALWAYS
3856       ELSE
3857          LET .TGT.SCORE = RANDI.F(1, .CUM2, RN.SEED)
3858          FOR EVERY ..FIRING.TABLE OF UE.TARGET.LIST(.HELO.LINK)
3859          WITH FT.SCORE2(..FIRING.TABLE) GE .TGT.SCORE
3860          FIND THE FIRST CASE
3861          IF NONE
3862             TRACE
3863             CALL ERROR.STOP-------------------------------------------------------------------------------------->(604)
3864          ELSE
3865             LET HF.FIRING.TABLE(HELICOPTER.FIRE) = ..FIRING.TABLE
3866          ALWAYS
3867       ALWAYS
3868       LET .TARGET.UNIT = FT.TGT.UNIT(..FIRING.TABLE)
3869       LET .TARGET.EQUIP = FT.TARGET.EQUIP(..FIRING.TABLE)
3870       LET .WPN = FT.FIRING.WPN(..FIRING.TABLE)
3871       CALL HEL.RANGE.COMPUTE------------------------------------------------------------------------------------->(302)
3872       GIVEN
3873       .ATK.HELICOPTER,
3874       .TARGET.UNIT
3875       YIELDING
3876       .RNG
3877
3878       IF HC.DEBUG = "YES"
3879          PRINT 1 LINE WITH
3880          HC.WPN.TYPE(.WPN),
3881          .ATK.HELICOPTER
3882          AS FOLLOWS
3883          ----HELO.FIRE (2.5)---- HC.WPN.TYPE=*****  ATK.HELO IS ******
3884       ALWAYS
3885
3886       IF HC.WPN.TYPE(.WPN) = ATGM
3887          LET .ROUNDS = 1
3888       ELSE
3889          IF HC.WPN.TYPE(.WPN) = FFATGM
```

```
3890            AND .ROUNDS > 1
3891            LET .ROUNDS = MIN.F(.ROUNDS,WPN.ROUNDS.REMAINING(.WPN))
3892         ELSE
3893            LET .ROUNDS = RANDI.F(2,10,RN.SEED) *
3894            TW.RATE.OF.FIRE(WPN.ID(.WPN)) / 600
3895            IF .ROUNDS LE 0
3896               LET .ROUNDS = 1
3897            ALWAYS
3898         ALWAYS
3899      ALWAYS
3900
3901      LOOP UNTIL WPN.ROUNDS.REMAINING(.WPN) LE 0
3902      OR UE.TARGET.LIST(.HELO.LINK) IS EMPTY
3903      OR UE.QUANT(.TARGET.EQUIP) LE 0
3904      OR HF.RESULTS(HELICOPTER.FIRE) = HIT
3905      DO THIS
3906
3907         IF HC.DEBUG = "YES"
3908            PRINT 2 LINES WITH
3909            .ATK.HELICOPTER, TW.NAME(WPN.ID(.WPN)),
3910            .WPN, .ROUNDS, EQ.NAME(UE.ID(.TARGET.EQUIP)) THUS
3911            ==HELO$FIRE(2.6) HELO ******, WEAPON ****** (******),
3912            HAS ***** ROUNDS TO FIRE AT A ******
3913         ALWAYS
3914
3915         IF .WPN > 0
3916            ADD .ROUNDS TO
3917            STW.RND.FIRED(UN.COLOR(HC.UNIT),
3918            WPN.ID(.WPN))
3919            LET .KV.SIDE = UN.COLOR(HC.UNIT)
3920            LET .KV = EQ.KV.ID(UE.ID
3921            (.HELO.LINK))
3922            IF .KV > 0
3923               ADD (.ROUNDS *
3924               TW.RND.WEIGHT(WPN.ID(.WPN)))
3925               TO KV.AMMO.CONSUMED(.KV.SIDE, .KV)
3926            ALWAYS
3927            SUBTRACT .ROUNDS FROM WPN.ROUNDS.REMAINING(.WPN)
3928            IF UN.COLOR(HC.UNIT) = BLUE
3929               LET I = WPN.ID(.WPN)
3930               LET J = UE.ID(.TARGET.EQUIP) -
3931               N.BLUE.TYPE.EQP
3932               ADD .ROUNDS TO STY.BLUE.EXP(I,J)
3933            ELSE
3934               LET I = WPN.ID(.WPN) - N.B.WPN.TYPE
3935               LET J = UE.ID(.TARGET.EQUIP)
3936               ADD .ROUNDS TO STY.RED.EXP(I,J)
3937            ALWAYS
3938            IF ANALYSIS(6) GT 0
3939               WRITE UN.BATTLE.INDEX(HC.UNIT),
3940               TIME.V, WPN.ID(.WPN),
3941               UE.ID(.TARGET.EQUIP)
3942               AS "DIR. ", I 7, S 2, D(7,2),
3943               S 2, I 4, S 2, I 4, / USING UNIT 60
3944            ALWAYS
3945         ELSE
3946            TRACE
3947            CALL ERROR.STOP----------------------------------------------------------------------------------------------------------->(604)
```

```
3948          ALWAYS
3949          LET .TIME = MAX.F((.RNG*16)/TW.ROUND.VELOCITY(WPN.ID(.WPN)),5)
3950
3951          FLY .TIME SECONDS  ''                                                    \FIX_LATER
3952
3953          IF HEL.IS.KILLED.INDIC(HELICOPTER.FIRE) = YES
3954             IF HC.DEBUG = "YES"
3955                PRINT 1 LINE WITH
3956                .ATK.HELICOPTER
3957                AS FOLLOWS
3958                ----HELO.FIRE(7)----  .ATK.HELICOPTER ****** IS DEAD
3959             ALWAYS
3960          ALWAYS
3961          IF HEL.IS.KILLED.INDIC(HELICOPTER.FIRE) = YES
3962             OR HT.TERMINATOR(ATK.TEAM) = YES
3963             OR HF.DESTRUCT.INDIC(HELICOPTER.FIRE) = YES
3964             OR HF.REINFORCE.IND(HELICOPTER.FIRE) = YES
3965             REMOVE THE HELICOPTER.FIRE FROM THE
3966             HF.SO.LIST(.HELO.LINK)
3967             IF ANALYSIS(6) = TRUE
3968                WRITE UN.COLOR(HC.UNIT),
3969                UN.BATTLE.INDEX(HC.UNIT),
3970                BTL.SEQ.NO(UN.BATTLE.INDEX(HC.UNIT)),
3971                EQ.NAME(HC.UE.ID(.ATK.HELICOPTER)),
3972                TW.NAME(WPN.ID(.WPN)),
3973                .ROUNDS,
3974                EQ.NAME(UE.ID(.TARGET.EQUIP)),
3975                FT.PK(..FIRING.TABLE),
3976                ..RANGE*16,
3977                HT.FARRP(HF.TEAM(HELICOPTER.FIRE)),
3978                FP.UNIT(HT.FARRP(HF.TEAM(HELICOPTER.FIRE))),
3979                N.HT.MEMBER.LIST(HF.TEAM(HELICOPTER.FIRE))
3980                AS I 1,S 2,I 6,S 2,I 4,S 2,T 6,S 2,T 6,S 2, ''                    \TEXT
3981                I 6,S 2,T 6,S 2,D(5,3),S 2,I 6,S 2,I 6,S 2,I 6, ''                \TEXT
3982                S 2,I 6,S 2,"MISS 1",/
3983                USING UNIT 48
3984             ALWAYS
3985 <-----------EXITPROCESS
3986          OTHERWISE
3987
3988          LET .HIT.CHANCE = RANDOM.F(RN.SEED)
3989          IF .HIT.CHANCE LE FT.PK(..FIRING.TABLE)
3990             LET HF.RESULTS(HELICOPTER.FIRE) = HIT
3991             IF UE.QUANT(.TARGET.EQUIP) LE 0
3992                IF ANALYSIS(6) = TRUE
3993                   WRITE UN.COLOR(HC.UNIT),
3994                   UN.BATTLE.INDEX(HC.UNIT),
3995                   BTL.SEQ.NO(UN.BATTLE.INDEX(HC.UNIT)),
3996                   EQ.NAME(HC.UE.ID(.ATK.HELICOPTER)),
3997                   TW.NAME(WPN.ID(.WPN)),
3998                   .ROUNDS,
3999                   EQ.NAME(UE.ID(.TARGET.EQUIP)),
4000                   FT.PK(..FIRING.TABLE),
4001                   ..RANGE*16,
4002                   HT.FARRP(HF.TEAM(HELICOPTER.FIRE)),
4003                   FP.UNIT(HT.FARRP(HF.TEAM(HELICOPTER.FIRE))),
4004                   N.HT.MEMBER.LIST(HF.TEAM(HELICOPTER.FIRE))
4005                   AS I 1,S 2,I 6,S 2,I 4,S 2,T 6,S 2,T 6,S 2,   ''                \TEXT
```

```
4006                I 6,S 2,T 6,S 2,D(5,3),S 2,I 6,S 2,I 6,S 2,I 6, ''                    \TEXT
4007                S 2,I 6,S 2,"MISS 2",/
4008                USING UNIT 48
4009             ALWAYS
4010             IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
4011                WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(.WPN)),
4012                EQ.NAME(UE.ID(.HELO.LINK)),
4013                EQ.TE.PTR(UE.ID(.HELO.LINK)),
4014                TW.NAME(WPN.ID(.WPN)),
4015                EQ.NAME(UE.ID(.TARGET.EQUIP)),
4016                EQ.TE.PTR(UE.ID(.TARGET.EQUIP)),
4017                ..RANGE*16,PK.DEFILADE,.SO,
4018                UN.COLOR(HC.UNIT), FT.PK(..FIRING.TABLE)
4019                AS S 1,"1",S 1,D(7,4),S 1,
4020                I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''                    \TEXT
4021                I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
4022                S 4,"0HF1",/
4023                USING UNIT 55
4024             ALWAYS
4025       <-------EXITLOOP ''TARGET WAS HIT BY SOMEONE
4026          OTHERWISE
4027
4028          IF HC.WPN.TYPE(.WPN) = FFATGM
4029             LET .NO.VICTIMS = MIN.F(.ROUNDS,
4030             UE.QUANT(.TARGET.EQUIP))
4031          ELSE
4032             LET .NO.VICTIMS = 1
4033          ALWAYS
4034          LOOP FOR .I = 1 TO .NO.VICTIMS
4035          UNLESS SO.LIST(.TARGET.EQUIP) IS EMPTY
4036          DO
4037             IF UE.QUANT(.TARGET.EQUIP) LE 0
4038                ''TARGETS ALL DEAD
4039          <------EXITLOOP
4040             OTHERWISE
4041             LET .VICTIM = RANDI.F(1,UE.QUANT(.TARGET.EQUIP),
4042             RN.SEED)
4043             LET .N = 1
4044             LOOP FOR EVERY .SO OF SO.LIST(.TARGET.EQUIP)
4045             UNTIL .N = .VICTIM
4046             DO
4047                ADD 1 TO .N
4048             ENDLOOP
4049             IF DROP.DEAD.INDICATOR(.SO) = NO
4050                LET DROP.DEAD.INDICATOR(.SO) = YES
4051                INTERRUPT SHOOT.OUT CALLED .SO------------------------------------------------------------------------>(493)
4052                REACTIVATE_THE SHOOT.OUT CALLED .SO------------------------------------------------------------------>(493)
4053                IN 1 MINUTES
4054             ALWAYS
4055             SUBTRACT 1 FROM UE.QUANT(.TARGET.EQUIP)
4056             IF UE.CRITICAL.EQUIP.INDIC(.TARGET.EQUIP)
4057                = TRUE
4058                SUBTRACT 1
4059                FROM MU.CRIT.NO(UN.PTR(.TARGET.UNIT))
4060             ALWAYS
4061             LET .THE.KILLER = EQ.KV.ID(UE.ID(.HELO.LINK))
4062             LET .THE.VICTIM = EQ.KV.ID(UE.ID(.TARGET.EQUIP))
4063             IF HC.DEBUG = "YES"
```

```
4064                PRINT 1 LINE WITH .THE.KILLER,
4065                .THE.VICTIM
4066                AS FOLLOWS
4067                ----HELO.FIRE(6)---- KILLER=****** VICTIM=****** NO.VICTIMS=1
4068             ALWAYS
4069             ADD 1 TO KV.SCORE(.KV.SIDE, .THE.KILLER,
4070             .THE.VICTIM)
4071             IF ANALYSIS(6) = TRUE
4072                WRITE UN.COLOR(HC.UNIT),
4073                UN.BATTLE.INDEX(HC.UNIT),
4074                BTL.SEQ.NO(UN.BATTLE.INDEX(HC.UNIT)),
4075                EQ.NAME(HC.UE.ID(.ATK.HELICOPTER)),
4076                TW.NAME(WPN.ID(.WPN)),
4077                .ROUNDS,
4078                EQ.NAME(UE.ID(.TARGET.EQUIP)),
4079                FT.PK(..FIRING.TABLE),
4080                ..RANGE*16,
4081                HT.FARRP(HF.TEAM(HELICOPTER.FIRE)),
4082                FP.UNIT(HT.FARRP(HF.TEAM(HELICOPTER.FIRE))),
4083                N.HT.MEMBER.LIST(HF.TEAM(HELICOPTER.FIRE))
4084                AS I 1,S 2,I 6,S 2,I 4,S 2,T 6,S 2,T 6,S 2, ''                    \TEXT
4085                I 6,S 2,T 6,S 2,D(5,3),S 2,I 6,S 2,I 6,S 2,I 6, ''                \TEXT
4086                S 2,I 6,S 2,"HIT 1",/
4087                USING UNIT 48
4088             ALWAYS
4089             IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
4090                WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(.WPN)),
4091                EQ.NAME(HC.UE.ID(.ATK.HELICOPTER)),
4092                EQ.TE.PTR(HC.UE.ID(.ATK.HELICOPTER)),
4093                TW.NAME(WPN.ID(.WPN)),
4094                EQ.NAME(UE.ID(.TARGET.EQUIP)),
4095                EQ.TE.PTR(UE.ID(.TARGET.EQUIP)),
4096                ..RANGE*16,PK.DEFILADE,.SO,
4097                UN.COLOR(HC.UNIT), FT.PK(..FIRING.TABLE)
4098                AS S 1,"1",S 1,D(7,4),S 1,
4099                I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''               \TEXT
4100                I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
4101                S 4,"0HF2",/
4102                USING UNIT 55
4103             ALWAYS
4104
4105             IF ANALYSIS(6) GT 0
4106                CALL OUTPUT.ATTRITION----------------------------------------------------------------------------->(611)
4107                GIVEN
4108                .TARGET.UNIT,
4109                HC.UE.ID(.ATK.HELICOPTER),
4110                UE.ID(.TARGET.EQUIP),
4111                1,
4112                "DIRECT"
4113             ALWAYS
4114
4115 <-----------WAIT 10./TW.RATE.OF.FIRE(WPN.ID(.WPN))
4116             MINUTES
4117
4118             IF HEL.IS.KILLED.INDIC(HELICOPTER.FIRE) = YES OR
4119                HF.DESTRUCT.INDIC(HELICOPTER.FIRE) = YES OR
4120                HF.REINFORCE.IND(HELICOPTER.FIRE) = YES OR
4121                HT.TERMINATOR(ATK.TEAM) = YES
```

```
4122                  REMOVE THE HELICOPTER.FIRE FROM THE
4123                  HF.SO.LIST(.HELO.LINK)
4124 <----------------EXITPROCESS
4125                OTHERWISE
4126
4127             ENDLOOP
4128          ELSE
4129             LET HF.RESULTS(HELICOPTER.FIRE) = MISS
4130             IF ANALYSIS(6) = TRUE ''%MAY83_%JAF
4131                WRITE UN.COLOR(HC.UNIT),
4132                UN.BATTLE.INDEX(HC.UNIT),
4133                BTL.SEQ.NO(UN.BATTLE.INDEX(HC.UNIT)),
4134                EQ.NAME(HC.UE.ID(.ATK.HELICOPTER)),
4135                TW.NAME(WPN.ID(.WPN)),
4136                .ROUNDS,
4137                EQ.NAME(UE.ID(.TARGET.EQUIP)),
4138                FT.PK(..FIRING.TABLE),
4139                ..RANGE*16,
4140                HT.FARRP(HF.TEAM(HELICOPTER.FIRE)),
4141                FP.UNIT(HT.FARRP(HF.TEAM(HELICOPTER.FIRE))),
4142                N.HT.MEMBER.LIST(HF.TEAM(HELICOPTER.FIRE))
4143                AS I 1,S 2,I 6,S 2,I 4,S 2,T 6,S 2,T 6,S 2, ''                    \TEXT
4144                I 6,S 2,T 6,S 2,D(5,3),S 2,I 6,S 2,I 6,S 2,I 6, ''                \TEXT
4145                S 2,I 6,S 2,"MISS 3",/
4146                USING UNIT 48
4147             ALWAYS
4148             IF ANALYSIS(5) = TRUE ''%MAY83_%JAF CATCEM
4149                WRITE TIME.V, TW.BASIC.LOAD(WPN.ID(.WPN)),
4150                EQ.NAME(HC.UE.ID(.ATK.HELICOPTER)),
4151                EQ.TE.PTR(HC.UE.ID(.ATK.HELICOPTER)),
4152                TW.NAME(WPN.ID(.WPN)),
4153                EQ.NAME(UE.ID(.TARGET.EQUIP)),
4154                EQ.TE.PTR(UE.ID(.TARGET.EQUIP)),
4155                ..RANGE*16,PK.DEFILADE,.SO,
4156                UN.COLOR(HC.UNIT), FT.PK(..FIRING.TABLE)
4157                AS S 1,"1",S 1,D(7,4),S 1,
4158                I 5,S 1,T 6,S 3,I 1,S 3,T 6,S 3,T 6,S 3,I 1,S 3, ''               \TEXT
4159                I 6,S 4,I 1,S 4,I 10,S 4,I 1,S 4,D(4,3),
4160                S 4,"0HF3",/
4161                USING UNIT 55
4162             ALWAYS
4163          ALWAYS
4164
4165       ENDLOOP
4166    ENDLOOP
4167
4168    REMOVE THE HELICOPTER.FIRE FROM THE HF.SO.LIST(.HELO.LINK)
4169
4170    IF HC.DEBUG = "YES"
4171       PRINT 1 LINE WITH .ATK.HELICOPTER THUS
4172       === END HELO$FIRE WITH ATKHELO ****** ===
4173    ALWAYS
4174
4175 <--EXITPROCESS
4176 ENDPROCESS
```

INPUT ROUTINES

```
4183 ROUTINE MAIN2                                                                                                    I001
4184
4185     ADD 1 TO ANAL.CTR(185,1)  ''                                                                                 \DYN_ANAL
4186     NORMALLY MODE IS INTEGER
4187
4188     CALL SYS.INPUT------------------------------------------------------------------------------------------>(523)
4189     IF DEBUG > 0
4190        LET BETWEEN.V = 'BETWEEN.ROUTINE'
4191        ''CALL BETWEEN.ROUTINE BEFORE EXECUTING EACH NEXT EVENT-------------------------------------------->(599)
4192     ALWAYS
4193
4194     ''PRINT 1 LINE WITH ZTIME.F,  USED.DBANK.V THUS                                                           \VAX
4195     ''CPU TIME USED AT INPUT = *****.**  SEC  DBANK AFTER SYS.INPUT = *****
4196     CALL PK.INPUT-------------------------------------------------------------------------------------------->(525)
4197     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4198     ''###  DBANK AFTER PK.INPUT = *****
4199     CALL CAT.TU.INPUT---------------------------------------------------------------------------------------->(527)
4200     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4201     ''###  DBANK AFTER CAT.TU.INPUT = *****
4202     CALL KV.INPUT-------------------------------------------------------------------------------------------->(529)
4203     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4204     ''###  DBANK AFTER KV.INPUT = *****
4205     CALL EQ.TE.INPUT----------------------------------------------------------------------------------------->(530)
4206     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4207     ''###  DBANK AFTER EQ.TE.INPUT = *****
4208     CALL TYPE.WEAPON.INPUT----------------------------------------------------------------------------------->(532)
4209     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4210     ''###  DBANK AFTER TYPE.WEAPON.INPUT = *****
4211
4212     RESERVE STY.RED.EXP(*,*) AS N.R.WPN.TYPE BY N.BLUE.TYPE.EQP
4213     RESERVE STY.BLUE.EXP(*,*) AS N.B.WPN.TYPE BY N.RED.TYPE.EQP
4214
4215     CALL UNIT.INPUT------------------------------------------------------------------------------------------>(533)
4216     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4217     ''###  DBANK AFTER UNIT.INPUT = *****
4218     CALL MFO.INPUT------------------------------------------------------------------------------------------>(538)
4219     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4220     ''###  DBANK AFTER MFO.INPUT = *****
4221     CALL READ.ORDERS----------------------------------------------------------------------------------------->(539)
4222     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4223     ''###  DBANK AFTER READ.ORDERS = *****
4224     CALL P.E.M.INPUT----------------------------------------------------------------------------------------->(547)
4225     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4226     ''###  DBANK AFTER P.E.M.INPUT = *****
4227     CALL TB.INPUT-------------------------------------------------------------------------------------------->(548)
4228     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4229     ''###  DBANK AFTER TB.INPUT = *****
4230     CALL BTRY.INPUT------------------------------------------------------------------------------------------>(550)
4231     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4232     ''###  DBANK AFTER BTRY.INPUT = *****
4233     CALL FBN.FD.INPUT---------------------------------------------------------------------------------------->(553)
4234     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4235     ''###  DBANK AFTER FBN.FD.INPUT = *****
4236     CALL MUNS.INPUT------------------------------------------------------------------------------------------>(555)
4237     ''PRINT 1 LINE WITH USED.DBANK.V THUS
4238     ''###  DBANK AFTER MUNS.INPUT = *****
4239     CALL SUBM.INPUT------------------------------------------------------------------------------------------>(557)
4240     ''PRINT 1 LINE WITH USED.DBANK.V THUS
```

```
4241    ''###  DBANK AFTER SUBM.INPUT = *****
4242    CALL HE.LA.INPUT------------------------------------------------------------>(558)
4243    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4244    ''###  DBANK AFTER HE.LA.INPUT = *****
4245    CALL RUL.EN.INPUT----------------------------------------------------------->(560)
4246    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4247    ''###  DBANK AFTER RUL.EN.INPUT = *****
4248    CALL ST.INPUT--------------------------------------------------------------->(561)
4249    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4250    ''###  DBANK AFTER ST.INPUT = *****
4251    CALL MCFR.INPUT------------------------------------------------------------->(562)
4252    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4253    ''###  DBANK AFTER MCFR.INPUT = *****
4254    CALL MPDB.INPUT------------------------------------------------------------->(563)
4255    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4256    ''###  DBANK AFTER MPDB.INPUT = *****
4257    CALL FEBA.INITIAL----------------------------------------------------------->( 61)
4258    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4259    ''###  DBANK AFTER FEBA.INITIAL = *****
4260    CALL MAO.INPUT    '' %JUL79_%RGR------------------------------------------->(564)
4261    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4262    ''###  DBANK AFTER MAO.INPUT = *****
4263    CALL SENSOR.INPUT----------------------------------------------------------->(565)
4264    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4265    ''###  DBANK AFTER SENSOR.INPUT = *****
4266    CALL TBF.INPUT-------------------------------------------------------------->(569)
4267    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4268    ''###  DBANK AFTER TBF.INPUT = *****
4269    CALL DECISION.INPUT--------------------------------------------------------->(570)
4270    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4271    ''###  DBANK AFTER DECISION.INPUT = *****
4272    CALL TT.FACTORS.INPUT------------------------------------------------------->(571)
4273    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4274    ''###  DBANK AFTER TT.FACTORS.INPUT = *****
4275    CALL FARRP.INPUT------------------------------------------------------------>(572)
4276    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4277    ''###  DBANK AFTER FARRP.INPUT = *****
4278    CALL PGM.INPUT-------------------------------------------------------------->(575)
4279    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4280    ''###  DBANK AFTER PGM.INPUT = *****
4281    CALL ILLUM.INPUT------------------------------------------------------------>(576)
4282    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4283    ''###  DBANK AFTER ILLUM.INPUT = *****
4284    CALL MINE.INPUT------------------------------------------------------------->(577)
4285    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4286    ''###  DBANK AFTER MINE.INPUT = *****
4287    CALL SMOKE.INPUT------------------------------------------------------------>(580)
4288    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4289    ''###  DBANK AFTER SMOKE.INPUT = *****
4290    CALL VIS.INPUT-------------------------------------------------------------->(581)
4291    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4292    ''###  DBANK AFTER VIS.INPUT = *****
4293    CALL TACAIR.INPUT----------------------------------------------------------->(582)
4294    YIELDING
4295    .TACAIR.FLAG
4296    ''PRINT 1 LINE WITH USED.DBANK.V THUS
4297    ''###  DBANK AFTER TACAIR.INPUT = *****
4298    IF .TACAIR.FLAG = 1
```

```
4299        ''CLOSE AIR SUPPORT IS BEING MODELED IN THIS RUN.
4300        CALL MADS.INPUT------------------------------------------------------------------------------------------------------>(587)
4301        ''PRINT 1 LINE WITH USED.DBANK.V THUS
4302        ''###  DBANK AFTER MADS.INPUT = ******
4303        CALL AC.MUNS.INPUT---------------------------------------------------------------------------------------------------->(588)
4304        ''PRINT 1 LINE WITH USED.DBANK.V THUS
4305        ''###  DBANK AFTER AC.MUNS.INPUT = ******
4306        CALL TACAIR.DATA.REPORT----------------------------------------------------------------------------------------------->(621)
4307     ALWAYS
4308
4309     WRITE RRECORD.V AS /," TOTAL RECORDS READ = ",I 7,/
4310
4311     IF ANALYSIS(6) GT 0
4312        CALL ANALYSIS.OUTPUT-------------------------------------------------------------------------------------------------->(595)
4313     ALWAYS
4314
4315 <--EXITROUTINE
4316 ENDROUTINE
```

```
4317 ROUTINE SYS.INPUT                                                    ''                I002
4318    ADD 1 TO ANAL.CTR(186,1)  ''                                                        \DYN_ANAL
4319
4320    NORMALLY MODE IS INTEGER
4321
4322    DEFINE X AS A REAL VARIABLE
4323    READ DEBUG,
4324    RN.SEED,
4325    NUM.POSITION.REPORT,
4326    CALP.ON
4327    RESERVE ANALYSIS(*) AS 6
4328    FOR I = 1 TO 6
4329    READ ANALYSIS(I)
4330
4331    PRINT 1 LINE WITH RN.SEED THUS
4332    ### SYS.INPUT  --  RN.SEED = ***********
4333    IF RN.SEED GT 10
4334       FOR I = 1 TO RN.SEED
4335       LET X = RANDOM.F(1)
4336       LET RN.SEED = 1
4337    ALWAYS
4338
4339
4340    LET POS.REP.INT = 24.0 / NUM.POSITION.REPORT
4341    LOOP FOR EACH SIDE
4342    DO THIS
4343       READ COST.CRITERIA(SIDE),
4344       ARTY.DECIMATE(SIDE),
4345       ARTY.DEGRADE(SIDE),
4346       BREAK.POINT(SIDE),
4347       REQ.EFF.MOVING(SIDE),
4348       REQ.EFF.STA(SIDE)
4349    ENDLOOP
4350
4351    READ STOP.SIMULATION.TIME,
4352    BMNT,
4353    EENT,
4354    N.SECTOR,
4355    FEBA.WIDTH,
4356    FRONT.DEPTH,
4357    INIT.X.FEBA,
4358    INIT.Y.FEBA,
4359    NUM.RAD.INCREMENTS,
4360    NUM.ANG.INCREMENTS,
4361    TERRAIN.PAR,
4362    TIME.BETWEEN.ARTY.MOVE, ''IN MINUTES
4363    LOC.UPDATE.FREQ,
4364    SCAN.RATE
4365    LET FEBA.WIDTH = 10. * FEBA.WIDTH / 16. ''                                          \OPTIM ZE
4366    LET FRONT.DEPTH = 10. * FRONT.DEPTH / 16.
4367    LET INIT.X.FEBA = 10. * INIT.X.FEBA / 16.
4368    LET INIT.Y.FEBA = 10. * INIT.Y.FEBA / 16.
4369    READ TAC.MOV.FAC,
4370    NITE.MOV.FAC,
4371    ACT.BATTLE.RANGE,
4372    REIN.PROX,
4373    REIN.THRESH,
4374    REIN.DELAY,
```

```
4375     ATK.DELAY
4376     LET ACT.BATTLE.RANGE = ACT.BATTLE.RANGE / 16
4377     LET REIN.PROX = REIN.PROX / 16
4378
4379 <--EXITROUTINE
4380 ENDROUTINE
```

```
4381 ROUTINE PK.INPUT                                                          1003
4382    ADD 1 TO ANAL.CTR(187,1)  ''                                          \DYN_ANAL
4383
4384    NORMALLY MODE IS INTEGER
4385    DEFINE X,Y AS REAL VARIABLES
4386
4387    READ N.PK.VECTOR
4388    READ N.PK.BAND
4389    CREATE EVERY PK.BAND AND PK.VECTOR
4390
4391    LOOP FOR EVERY PK.BAND
4392    DO THE FOLLOWING
4393       READ PK.BAND.RNG(PK.BAND)
4394    ENDLOOP
4395
4396    LOOP FOR EVERY PK.VECTOR,
4397    FOR EVERY PK.BAND
4398    DO THE FOLLOWING
4399       READ X
4400       LET X = X/2.
4401       LET PK.PROB(PK.VECTOR,PK.BAND) = X
4402    ENDLOOP
4403
4404    READ FIRERS
4405    READ TARGETS
4406    RESERVE PK.POINTER(*,*) AS FIRERS BY TARGETS
4407
4408    READ PK.POINTER
4409    RESERVE PK.DEF.POINTER AS FIRERS BY TARGETS
4410
4411    READ PK.DEF.POINTER
4412    READ N.PK.MOVE.FACTOR
4413    READ N.PK.MOVE.BAND
4414    CREATE EVERY PK.MOVE.FACTOR AND PK.MOVE.BAND
4415
4416    LOOP FOR EVERY PK.MOVE.BAND
4417    DO THE FOLLOWING
4418       READ Y
4419       LET PK.MOV.RNG(PK.MOVE.BAND) = Y/16
4420    ENDLOOP
4421
4422    LOOP FOR EVERY PK.MOVE.FACTOR
4423    FOR EVERY PK.MOVE.BAND
4424    DO THE FOLLOWING
4425       READ PK.MOV.FAC(PK.MOVE.FACTOR, PK.MOVE.BAND)
4426    ENDLOOP
4427
4428    READ N.PK.F.MOVE.FACTOR
4429    CREATE EVERY PK.F.MOVE.FACTOR
4430    ''THE ARRAY PK.F.MOV.FAC DOES NOT SEEM TO BE ALLOCATED
4431    ''THE FOLLOWING RESERVE STATEMENT ATTEMPTS TO CORRECT THAT                 \VAX
4432    RESERVE PK.F.MOV.FAC(*,*) AS N.PK.F.MOVE.FACTOR BY N.PK.MOVE.BAND
4433
4434    LOOP FOR EVERY PK.F.MOVE.FACTOR
4435    FOR EVERY PK.MOVE.BAND
4436    DO THE FOLLOWING
4437       READ PK.F.MOV.FAC(PK.F.MOVE.FACTOR, PK.MOVE.BAND)
4438    ENDLOOP
```

```
4439
4440     RESERVE TGT.OTM(*,*) AS FIRERS BY TARGETS
4441
4442     READ TGT.OTM
4443
4444     RESERVE FIRE.OTM(*,*) AS FIRERS BY TARGETS
4445
4446     READ FIRE.OTM
4447
4448 <--EXITROUTINE
4449 ENDROUTINE
```

```
4450 ROUTINE CAT.TU.INPUT                                                    ''
4451
4452    ADD 1 TO ANAL.CTR(188,1)  ''
4453    NORMALLY MODE IS INTEGER
4454
4455    IF DEBUG=TRUE, PRINT 1 LINE THUS
4456       = = = CAT.TU.INPUT = = =
4457    ALWAYS
4458
4459    READ N.CATEGORY
4460    CREATE EVERY CATEGORY
4461    LOOP FOR EACH CATEGORY
4462    DO
4463       READ CT.NAME(CATEGORY),
4464       CT.MIN.FEBA(CATEGORY),
4465       CT.GROUP(CATEGORY)
4466       LET CT.MIN.FEBA(CATEGORY) = CT.MIN.FEBA(CATEGORY) * 10. / 16.
4467       FILE CATEGORY IN GP.CAT.SET(CT.GROUP(CATEGORY))
4468    ENDLOOP
4469
4470    READ N.TYPE.UNIT
4471    CREATE EVERY TYPE.UNIT
4472    LOOP FOR EACH TYPE.UNIT
4473    DO
4474       READ TU.SEQ.NO,
4475       TU.LEVEL(TYPE.UNIT),
4476       TU.CAT(TYPE.UNIT),
4477       TU.MIL.WORTH(TYPE.UNIT),
4478       TU.MOV.RATE(TYPE.UNIT),
4479       TU.PRIN.TE(TYPE.UNIT),
4480       TU.RADIUS(TYPE.UNIT),
4481       TU.SIDE(TYPE.UNIT),
4482       TU.SUP.PRIORITY(TYPE.UNIT),
4483       TU.OPP.PRIORITY(TYPE.UNIT),
4484       TU.MF.FACTOR(TYPE.UNIT),
4485       TU.ATKING.AC(TYPE.UNIT),
4486       TU.AC.PER.MSN(TYPE.UNIT)
4487       FILE TYPE.UNIT IN CT.TU.SET(TU.CAT(TYPE.UNIT))
4488
4489       LET FLAG = 0
4490       LOOP UNTIL FLAG = 1
4491       DO THIS
4492          CREATE A TU.TE.LINK
4493          READ TU.TE.ID(TU.TE.LINK)
4494          IF TU.TE.ID(TU.TE.LINK) = 999
4495             DESTROY THE TU.TE.LINK
4496             LET FLAG = 1
4497          ELSE
4498             READ TU.TE.QUANT(TU.TE.LINK),
4499             TU.CRITICAL.EQUIP.INDIC(TU.TE.LINK)
4500             IF TU.CRITICAL.EQUIP.INDIC(TU.TE.LINK)=YES
4501                ADD TU.TE.QUANT(TU.TE.LINK) TO TU.CRIT.NO(TYPE.UNIT)
4502             ALWAYS
4503             FILE TU.TE.LINK IN TU.TE.LIST(TYPE.UNIT)
4504          ALWAYS
4505       ENDLOOP
4506
4507       LET FLAG = 0
```

```
4508        LOOP UNTIL FLAG = 1
4509        DO THIS
4510           CREATE A TU.NTE.LINK
4511           READ TU.NTE.ID(TU.NTE.LINK)
4512           IF TU.NTE.ID(TU.NTE.LINK) = 999
4513              DESTROY THE TU.NTE.LINK
4514              LET FLAG = 1
4515           ELSE
4516              FILE TU.NTE.LINK IN TU.NTE.SET(TYPE.UNIT)
4517           ALWAYS
4518        ENDLOOP
4519     ENDLOOP
4520
4521 <—EXITROUTINE
4522 ENDROUTINE
```

```
4523 ROUTINE KV.INPUT                                                    ''
4524                                                                                I005
4525    ADD 1 TO ANAL.CTR(189,1)  ''                                       \DYN_ANAL
4526    NORMALLY MODE IS INTEGER
4527    DEFINE SIDE, KV AS INTEGER VARIABLES
4528
4529    RESERVE KV.CEM.WPN.NO(*,*),
4530    KV.AMMO.CONSUMED(*,*),
4531    KV.INITIAL.DENSITY(*,*),
4532    KV.EQ.ID(*,*)
4533    AS N.SIDE BY *
4534
4535    ''CREATE RED KILLER VICTIMS
4536    READ RED.N.KV
4537
4538    RESERVE KV.CEM.WPN.NO(RED,*),
4539    KV.AMMO.CONSUMED(RED,*),
4540    KV.INITIAL.DENSITY(RED,*),
4541    KV.EQ.ID(RED,*)
4542    AS RED.N.KV
4543
4544    ''CREATE BLUE KILLER VICTIMS
4545    READ BLUE.N.KV
4546
4547    RESERVE KV.CEM.WPN.NO(BLUE,*),
4548    KV.AMMO.CONSUMED(BLUE,*),
4549    KV.INITIAL.DENSITY(BLUE,*),
4550    KV.EQ.ID(BLUE,*)
4551    AS BLUE.N.KV
4552
4553    ''CREATE SCOREBOARD
4554
4555    RESERVE KV.SCORE(*,*,*)
4556    AS N.SIDE BY *
4557    RESERVE KV.SCORE(RED,*,*)
4558    AS RED.N.KV BY BLUE.N.KV
4559    RESERVE KV.SCORE(BLUE,*,*)
4560    AS BLUE.N.KV BY RED.N.KV
4561
4562    LOOP FOR EACH SIDE,
4563    DO
4564       LET NKV = DIM.F(KV.CEM.WPN.NO(SIDE,*))
4565       LOOP FOR KV = 1 TO NKV,
4566       DO
4567          READ KV.CEM.WPN.NO(SIDE,KV)
4568       ENDLOOP
4569    ENDLOOP
4570
4571 <--EXITROUTINE
4572 ENDROUTINE
```

```
4573 ROUTINE EQ.TE.INPUT                                                                                                    I006
4574
4575     ADD 1 TO ANAL.CTR(190,1)  ''                                                                                       \DYN_ANAL
4576     NORMALLY MODE IS INTEGER
4577     DEFINE .DELTA, .HEIGHT AS REAL VARIABLE
4578
4579     READ N.TYPE.EQUIPMENT
4580     CREATE EVERY TYPE.EQUIPMENT
4581     LOOP FOR EACH TYPE.EQUIPMENT CALLED .TE
4582     DO
4583        READ TE.NAME(.TE),
4584        TE.PGM.INDIC(.TE),
4585        TE.PROJECTED.AREA(.TE),
4586        TE.MIN.MF.LOSS(.TE),
4587        TE.MAX.MF.LOSS(.TE),
4588        .DELTA,                    ''DEGREES C
4589        .HEIGHT                    ''METERS
4590        LET TE.DELTA.T(.TE) = .DELTA * 10
4591        LET TE.HEIGHT(.TE) = .HEIGHT * 10
4592     ENDLOOP
4593
4594     READ N.BLUE.TYPE.EQP,
4595     N.RED.TYPE.EQP
4596     LET N.EQUIPMENT = N.BLUE.TYPE.EQP + N.RED.TYPE.EQP
4597     CREATE EVERY EQUIPMENT
4598
4599     LOOP FOR EACH EQUIPMENT
4600     DO
4601        READ EQ.SEQ.NO,
4602        EQ.NAME(EQUIPMENT),
4603        EQ.TE.PTR(EQUIPMENT),
4604        EQ.MAX.SPEED(EQUIPMENT),
4605        EQ.PERSONNEL.LOAD(EQUIPMENT)
4606        IF CALP.ON = 1
4607           READ EQ.PAX.KILL.RATE(EQUIPMENT)
4608        ALWAYS
4609        READ EQUIP.PK.PTR(EQUIPMENT),
4610        EQ.AD.INDICATOR(EQUIPMENT)
4611        IF EQ.TE.PTR(EQUIPMENT) > 0
4612           FILE EQUIPMENT IN TE.SET(EQ.TE.PTR(EQUIPMENT))
4613        ELSE
4614           CALL ERROR.STOP------------------------------------------------------------------------------------------>(604)
4615        ALWAYS
4616
4617        READ CEM.WPN.NO
4618        IF CEM.WPN.NO IS NOT ZERO,
4619           LET KV = 0
4620           FOR EACH SIDE
4621           FOR KV=1 TO DIM.F(KV.CEM.WPN.NO(SIDE,*))  ''                                                              \OPTIMIZE
4622           WITH CEM.WPN.NO = KV.CEM.WPN.NO(SIDE,KV)
4623           FIND THE FIRST CASE
4624           IF FOUND
4625              LET EQ.KV.ID(EQUIPMENT) = KV
4626              LET KV.EQ.ID(SIDE,KV) = EQUIPMENT
4627           ELSE      ''CEM WEAPON NUMBER NOT FOUND AS KV
4628              PRINT 1 LINE WITH EQUIPMENT, CEM.WPN.NO THUS
4629              KV.INPUT  EQUIPMENT **** CEM.WPN.NO **** NOT FOUND
4630           ALWAYS
```

```
4631
4632        ALWAYS
4633     ENDLOOP
4634
4635
4636 <--EXITROUTINE
4637 ENDROUTINE
```

```
4638 ROUTINE TYPE.WEAPON.INPUT                                          ''
4639
4640    ADD 1 TO ANAL.CTR(191,1)  ''
4641    NORMALLY MODE IS INTEGER
4642    DEFINE MIN.RANGE, MAX.RANGE, TW.RW, .HFOV, .VFOV AS REAL VARIABLES
4643
4644    READ N.B.WPN.TYPE,
4645    N.R.WPN.TYPE
4646    LET N.TYPE.WEAPON = N.B.WPN.TYPE + N.R.WPN.TYPE
4647    CREATE EVERY TYPE.WEAPON
4648
4649    LOOP FOR EVERY TYPE.WEAPON CALLED .TW
4650    DO THE FOLLOWING
4651       READ TW.SEQ.NO,
4652       TW.NAME(.TW),
4653       TW.RATE.OF.FIRE(.TW),
4654       TW.ROF.AIR(.TW),
4655       TW.NO.SENSORS(.TW),
4656       TW.ROUND.VELOCITY(.TW),
4657       TW.RW              ''READ THE WEIGHT IN LBS AND TENTHS
4658       READ MAX.RANGE
4659       READ MIN.RANGE
4660       READ TW.BASIC.LOAD(.TW),
4661       TW.PK.PTR(.TW),
4662       TW.NITE.FAC(.TW),
4663       TW.FIRE.OTM.PTR(.TW),
4664       TW.TYPE.OF.SENSOR(.TW),
4665       TW.SPECTRUM(.TW),
4666       .HFOV,      ''DEGREES
4667       .VFOV,      ''DEGREES
4668       TW.HFOS(.TW),   ''DEGREES
4669       TW.VFOS(.TW),    ''DEGREES
4670       TW.AC.DET.TIME(.TW)
4671       LET TW.RND.WEIGHT(.TW) = INT.F(TW.RW * 100)
4672       LET TW.MAX.RANGE(.TW) = MAX.RANGE/16
4673       LET TW.MIN.RANGE(.TW) = MIN.RANGE/16
4674       LET TW.HFOV(.TW) = .HFOV * 10
4675       LET TW.VFOV(.TW) = .VFOV * 10
4676    ENDLOOP
4677
4678 <--EXITROUTINE
4679 ENDROUTINE
```

```
4680 ROUTINE UNIT.INPUT                                                          ''                                                    1008
4681
4682    ADD 1 TO ANAL.CTR(192,1)  ''                                                                                                  \DYN_ANAL
4683    NORMALLY MODE IS INTEGER
4684    DEFINE NAME, .CTNAME AS TEXT VARIABLES ''                                                                                      \TEXT
4685
4686    READ N.UNIT
4687    CREATE EACH UNIT
4688
4689    LOOP FOR I = 1 TO N.UNIT
4690    DO
4691       READ UNIT.SEQ.NO,
4692       UNIT.NOS(I),
4693       NAME,
4694       UN.X.COORD(I),  ''READ IN DECAMETERS
4695       UN.Y.COORD(I),  ''READ IN DECAMETERS
4696       UN.PARENT(I),
4697       UN.COLOR(I),
4698       UN.RADIUS(I)   '' IN METERS,DO NOT USE "333"
4699       LOOP FOR EACH TYPE.UNIT
4700       DO
4701          IF NAME = TU.LEVEL(TYPE.UNIT)
4702             LET UN.TYPE.UNIT(I) = TYPE.UNIT
4703       <------EXITLOOP
4704          OTHERWISE
4705       ENDLOOP
4706       IF UN.TYPE.UNIT(I) = 0
4707          PRINT 1 LINE WITH NAME THUS
4708          ****** NOT FOUND IN TYPE UNIT FILE
4709          LET DATA.ERROR = TRUE
4710       ALWAYS
4711
4712       IF UNIT.SEQ.NO NE I
4713          PRINT 1 LINE WITH UNIT.NOS(I) THUS
4714          = = =ERROR = UNIT SEQUENCE #'S ARE OUT OF ORDER (CHECK UNIT.NOS ******) ==
4715          LET DATA.ERROR = TRUE
4716       ALWAYS
4717
4718       IF I > 1
4719          LOOP FOR K = 1 TO I-1
4720          DO
4721             IF UNIT.NOS(K) = UNIT.NOS(I)
4722                PRINT 1 LINE WITH UNIT.NOS(I) THUS
4723                UNIT.NOS ****** IS DUPLICATED
4724                LET DATA.ERROR = TRUE
4725             ALWAYS
4726          ENDLOOP
4727       ALWAYS
4728
4729       ADD 1 TO TU.FREQ(UN.TYPE.UNIT(I))
4730       ADD 1 TO SIDE.TU.TOTAL(UN.COLOR(I))
4731       LET UN.X.COORD(I) = 10.*UN.X.COORD(I)/16.    '' CONVERT TO HDM                                                              \OPTIMIZE
4732       LET UN.Y.COORD(I) = 10.*UN.Y.COORD(I)/16.    '' CONVERT TO HDM
4733       LET UN.STATUS(I)=STATIONARY
4734       LET .SIDE = UN.COLOR(I)
4735       LET .GROUPING = CT.GROUP(TU.CAT(UN.TYPE.UNIT(I)))
4736       LET .CTNAME = CT.NAME(TU.CAT(UN.TYPE.UNIT(I)))
4737       IF .GROUPING = MANEUVER
```

```
4738          CREATE A MAN.UNIT
4739          LET UN.PTR(I) = MAN.UNIT
4740          LET MU.UNIT.ID(UN.PTR(I)) = I
4741          READ MU.REINF.IND(UN.PTR(I))
4742          IF MU.REINF.IND(UN.PTR(I)) < 0 OR MU.REINF.IND(UN.PTR(I)) > 2
4743             PRINT 1 LINE WITH UNIT.NOS(I) THUS
4744             MANEUVER UNIT.NOS ****** HAS INCORRECT REINF. INDICATOR
4745             LET DATA.ERROR = TRUE
4746          ALWAYS
4747       ALWAYS
4748
4749       IF M.UNIT.SET(I) = 1
4750          PRINT 1 LINE WITH UNIT.NOS(I), .SIDE, .GROUPING THUS
4751          = = = UNIT ****** IS ALREADY FILED IN SIDE *, GROUP * = = =
4752          LET DATA.ERROR = TRUE
4753       ALWAYS
4754
4755       LET UNIT = I
4756       FILE UNIT IN UNIT.SET(.SIDE,.GROUPING)
4757       LET FLAG = 0
4758       LOOP UNTIL FLAG = 1
4759       DO
4760          CREATE A UE.LINK CALLED EQ
4761          READ NAME
4762          IF NAME = "999"
4763             DESTROY THE UE.LINK CALLED EQ
4764             LET FLAG = 1
4765          ELSE
4766             IF NAME NE "MINEFD" AND NAME NE "MOPMS" AND
4767                NAME NE "FASCAM"
4768                LOOP FOR EACH EQUIPMENT
4769                DO
4770                   IF NAME = EQ.NAME(EQUIPMENT)
4771                      LET UE.ID(EQ) = EQUIPMENT
4772                <------EXITLOOP
4773                   OTHERWISE
4774                ENDLOOP
4775             ELSE
4776                IF UN.COLOR(I) = BLUE
4777                   LET .I1 = 1
4778                   LET .I2 = N.BLUE.TYPE.EQP
4779                ELSE
4780                   LET .I1 = N.BLUE.TYPE.EQP + 1
4781                   LET .I2 = N.EQUIPMENT
4782                ALWAYS
4783                FOR .EQUIP = .I1 TO .I2
4784                WITH EQ.NAME(.EQUIP) = NAME
4785                FIND THE FIRST CASE
4786                IF FOUND
4787                   LET UE.ID(EQ) = .EQUIP
4788                ALWAYS
4789             ALWAYS
4790
4791             IF UE.ID(EQ) = 0
4792                PRINT 1 LINE WITH NAME, UNIT.NOS(I) THUS
4793                ****** NOT FOUND IN EQUIPMENT FILE - FOR UNIT.NOS ******
4794                LET DATA.ERROR = TRUE
4795             ALWAYS
```

```
4796
4797            READ UE.QUANT(EQ),
4798            UE.CRITICAL.EQUIP.INDIC(EQ)
4799            ''
4800            ''
4801            LET KV = EQ.KV.ID(UE.ID(EQ))
4802            IF KV IS NOT ZERO,
4803               ADD UE.QUANT(EQ) TO KV.INITIAL.DENSITY(UN.COLOR,KV)
4804            ALWAYS
4805            IF UE.CRITICAL.EQUIP.INDIC(EQ) = YES AND
4806               .GROUPING = MANEUVER
4807               ADD UE.QUANT(EQ) TO MU.CRIT.NO(UN.PTR(I))
4808            ALWAYS
4809            FILE EQ IN THE UN.EQUIP.LIST(I)
4810            IF .GROUPING=MANEUVER OR .CTNAME = "CBTAVN" OR
4811               .CTNAME = "AIRDEF"
4812               LOOP UNTIL NAME = "999"
4813               DO THE FOLLOWING
4814                  CREATE A WEAPON  CALLED W
4815                  READ NAME
4816                  IF NAME = "999"
4817                     DESTROY THE WEAPON CALLED W
4818               <------LEAVE
4819                  OTHERWISE
4820
4821                  LOOP FOR EACH TYPE.WEAPON
4822                  DO
4823                     IF NAME = TW.NAME(TYPE.WEAPON)
4824                        LET WPN.ID(W) = TYPE.WEAPON
4825                  <------EXITLOOP
4826                     OTHERWISE
4827                  ENDLOOP
4828                  IF WPN.ID(W) = 0
4829                     PRINT 1 LINE WITH NAME, UNIT.NOS(I) THUS
4830                     ****** NOT FOUND IN TYPE WEAPON FILE - UNIT.NOS ******
4831                     LET DATA.ERROR = TRUE
4832                  ALWAYS
4833
4834                  READ .WPN.STATUS
4835                  READ WPN.QUANTITY(W)
4836                  IF .CTNAME= "CBTAVN"
4837                     ''CBTAVN IS EITHER A FARRP OR AN AIRFIELD.
4838                     ''HC.WPN.TYPE IS USED FOR A WEAPON BELONGING
4839                     ''TO A HELICOPTER IN A FARRP.  IT IS
4840                     ''EQUIVALENCED TO WPN.AC.MUNS FOR A WEAPON
4841                     ''BELONGING TO AN AIRCRAFT IN AN AIRFIELD.
4842                     ''WPN.AC.MUNS = 0 MEANS A MISSLE
4843                     ''WPN.AC.MUNS LT 0 MEANS MINES
4844                     ''WPN.AC.MUNS GT 0 IS A POINTER TO AC.MUNS
4845                     ''(BOMBS)
4846                     READ HC.WPN.TYPE(W)
4847                  ALWAYS
4848                  FILE W IN THE UE.WEAPON.SET(EQ)
4849               ENDLOOP  '' LOOP THRU ALL WEAPONS ON THE EQUIP
4850            ALWAYS   '' GROUP IS NOT MANEUVER, OTHER THAN INF,ARM
4851         ALWAYS   '' UNIT EQUIPMENT IS - - -
4852      ENDLOOP  '' LOOP THRU ALL EQUIP ASG TO THE UNIT
4853   ENDLOOP  '' LOOP THRU ALL UNITS ON THE LIST
```

```
4854
4855    LOOP FOR I = 1 TO N.UNIT
4856    DO
4857       LET UNIT = I
4858       IF UN.PARENT(I) > 0
4859          FOR J = 1 TO N.UNIT ,
4860          WITH  UNIT.NOS(J) = UN.PARENT(I)
4861          FIND THE FIRST CASE
4862          IF NONE,
4863             SKIP 1 LINE
4864             PRINT 1 LINE WITH UNIT.NOS(I), UN.PARENT(I) THUS
4865             UNIT.NOS ******'S   PARENT (****), NOT FOUND IN UNIT FILE
4866             LET DATA.ERROR = TRUE
4867          ELSE
4868             LET UN.PARENT(I) = J
4869             FILE UNIT IN UN.SUB.LIST(UN.PARENT(I))
4870          ALWAYS
4871       ALWAYS
4872    ENDLOOP
4873
4874    IF DATA.ERROR = TRUE
4875       PRINT 1 LINE THUS
4876       = = =  ERROR IN UNIT DATA = = =
4877       TRACE   ''                                                                    \1
4878 <-----STOP
4879    OTHERWISE
4880
4881    '' DENSITIES FOR THE THEATER LEVEL MODELS INDIRECT FIRE (ATCAL)
4882    ''%15MAY83_%HWJ
4883
4884 <--RETURN ''                                  LINES BELOW SKIPPED TEMPORARILY FOR VAX TESTING \DEBUG
4885    LOOP FOR EACH EQUIPMENT
4886    DO  ''                                                                        \1
4887       IF EQUIPMENT GT N.BLUE.TYPE.EQP
4888          LET SIDE = 1
4889       ELSE
4890          LET SIDE = 2
4891       ALWAYS
4892       IF EQ.KV.ID(EQUIPMENT) LE 0
4893    <-----CYCLE
4894       ENDIF
4895       WRITE EQ.NAME(EQUIPMENT),
4896       KV.INITIAL.DENSITY(SIDE,EQ.KV.ID(EQUIPMENT)) AS /,
4897       T 6 , S 2 , I 5  USING UNIT 16 ''                                          \TEXT
4898       WRITE  EQ.NAME(EQUIPMENT),
4899       KV.INITIAL.DENSITY(SIDE,EQ.KV.ID(EQUIPMENT)) AS /,
4900       T 6 , S 2 , I 5 USING UNIT 55 ''                                           \TEXT
4901    ENDLOOP  '' ALL EQUIP
4902    ''
4903    WRITE AS / , "UNIT NUMBERS   UNIT COLOR   UNIT RADIUS   TYPE UNIT"
4904    USING UNIT 16
4905    WRITE AS / , "EQUIP NAME     CRIT EQUIP   EQUIP QUANT   PROJ AREA"
4906    USING UNIT 16
4907    WRITE AS / USING UNIT 16
4908    WRITE AS / USING UNIT 55
4909    USE UNIT 16 FOR OUTPUT
4910    LOOP FOR UNIT = 1 TO N.UNIT,
4911    DO   ''                                                                       \1
```

```
4912        WRITE UNIT.NOS(UNIT),UN.COLOR(UNIT),UN.RADIUS(UNIT),
4913        UN.TYPE.UNIT(UNIT) AS
4914        I 5, S 3, I 5, S 3, I 5, S 3, I 5, /
4915        ''
4916        FOR EACH UE.LINK IN UN.EQUIP.LIST(UNIT),
4917        DO   ''                                                                    \1
4918           WRITE EQ.NAME(UE.ID(UE.LINK)),
4919           UE.CRITICAL.EQUIP.INDIC(UE.LINK),
4920           UE.QUANT(UE.LINK),
4921           TE.PROJECTED.AREA(EQ.TE.PTR(UE.ID(UE.LINK)))
4922           AS T 6, S 1, I 1, S 1, I 10, S 1, I 3, S 1, / ''                        \TEXT
4923        ENDLOOP '' EACH UE.LINK
4924     ENDLOOP '' 1 TO N.UNIT
4925     WRITE AS / USING UNIT 16
4926     WRITE AS / USING UNIT 55
4927     ''
4928 <--EXITROUTINE
4929 ENDROUTINE
```

```
4930 ROUTINE MFO.INPUT                                                    ''   I009
4931
4932     ADD 1 TO ANAL.CTR(193,1)  ''                                          \DYN_ANAL
4933     NORMALLY MODE IS INTEGER
4934
4935     LET N.MOVEMENT.STATUS = 2
4936     CREATE EVERY MOVEMENT.STATUS
4937
4938     READ N.MODEL.FO
4939     CREATE EVERY MODEL.FO
4940
4941     READ N.FO.RANGE.BAND
4942     CREATE EVERY FO.RANGE.BAND
4943
4944     LET TOT.RH = 0
4945     LET OLD.TOT.RH = 1
4946     LOOP FOR EACH MODEL.FO CALLED MFO
4947     DO
4948        READ MFO.NAME(MFO),
4949        MFO.EQ.ID(MFO),
4950        MFO.SEARCH.RATE(MFO),
4951        MFO.PGM.CAP(MFO),
4952        NUM.RH
4953        ADD NUM.RH TO TOT.RH
4954        LOOP FOR RH = OLD.TOT.RH TO TOT.RH
4955        DO
4956           READ FO.RB.RANGE(RH)
4957           LET FO.RB.RANGE(RH) = FO.RB.RANGE(RH)/16.0
4958           READ FO.VISIBILITY(RH),
4959           FO.CIR.ERROR(RH)
4960           FILE RH IN MFO.RB.SET(MFO)
4961        ENDLOOP
4962        ADD NUM.RH TO OLD.TOT.RH
4963     ENDLOOP
4964
4965     LET TOT.RH = 0
4966     LET OLD.TOT.RH = 1
4967     LOOP FOR EACH MODEL.FO CALLED MFO
4968     DO
4969        ADD N.MFO.RB.SET(MFO) TO TOT.RH
4970        LOOP FOR EACH TYPE.EQUIPMENT CALLED TE
4971        UNLESS TE.NAME(TE) = "MINES"
4972        FOR EACH NITE.OR.DAY CALLED NOD
4973        FOR EACH MOVEMENT.STATUS CALLED MSTAT
4974        FOR RH = OLD.TOT.RH TO TOT.RH
4975        DO
4976           READ MNMTR.PROB.DETECT(MFO,NOD,MSTAT,TE,RH)
4977        ENDLOOP
4978        ADD N.MFO.RB.SET(MFO) TO OLD.TOT.RH
4979     ENDLOOP
4980
4981 <--EXITROUTINE
4982 ENDROUTINE
```

```
4983 ROUTINE READ.ORDERS                                                                                    I010
4984
4985    ADD 1 TO ANAL.CTR(194,1)  ''                                                                        \DYN_ANAL
4986    ''THIS READS UNIT ORDERS, DETERMINES IF THE UNIT HAS SUBORDINATES,
4987    ''AND IF SO FORMS A TASK FORCE OF UNITS ALL OPERATING UNDER THE SAME SET
4988    ''OF ORDERS.  AFTER DETERMINING THE TYPE OF ORDER, IT TRANSFERS CONTROL
4989    ''TO A SUBROUTINE APPROPRIATE TO THE ORDER TYPE TO READ INFO PERTAINING
4990    ''TO THAT TYPE OF ORDER.  ERROR MESSAGES ARE ISSUED IF SPECIFIED ORDER
4991    ''TYPE IS NOT ONE OF THE LEGAL TYPES OR IF CONFLICTING ORDERS OCCUR.
4992    ''(FOR EXAMPLE, ORDERS ISSUED TO A PLATOON AND ITS PARENT COMPANY
4993    ''PROVIDE A CONFLICT.)
4994
4995    NORMALLY MODE IS INTEGER
4996    DEFINE TYPE.ORDER. AS A TEXT VARIABLE ''                                                            \TEXT
4997
4998    READ UNIT.NO.
4999    LOOP UNTIL UNIT.NO. = 9999,
5000    DO
5001       FOR I = 1 TO N.UNIT,
5002       WITH UNIT.NOS(I) = UNIT.NO.,
5003       FIND THE FIRST CASE
5004       IF NONE,
5005          SKIP 1 LINE
5006          PRINT 1 LINE WITH UNIT.NO. THUS
5007          UNIT **** DOES NOT EXIST
5008          LET ERROR. = 1
5009       ALWAYS
5010
5011       LET REAL.UN.NO. = UNIT.NO.
5012       LET UNIT.NO. = 1
5013       IF CT.GROUP(TU.CAT(UN.TYPE.UNIT(UNIT.NO.))) NE MANEUVER
5014          PRINT 1 LINE WITH REAL.UN.NO. THUS
5015          ****** IS NOT A MANEUVER UNIT
5016          LET ERROR. = 1
5017       ELSE
5018          IF MU.ORDER.SET(UN.PTR(UNIT.NO.)) IS NOT EMPTY
5019             PRINT 1 LINE WITH REAL.UN.NO. THUS
5020             UNIT ****** ALREADY HAS ORDERS
5021             LET ERROR. = 1
5022          ALWAYS
5023       ALWAYS
5024
5025       LET SEQ.=1
5026       IF MU.TF.MEM(UN.PTR(UNIT.NO.))>0
5027          SKIP 1 LINE
5028          PRINT 1 LINE WITH REAL.UN.NO. THUS
5029          A PARENT OF UNIT ****** ALREADY HAS ORDERS
5030          LET ERROR.=1
5031       ALWAYS
5032
5033       IF UN.SUB.LIST(UNIT.NO.) IS NOT EMPTY
5034          CALL FORM.TF.LIST------------------------------------------------------------------------------>( 66)
5035          GIVEN
5036          UNIT.NO,
5037          UNIT.NO,
5038          ERROR
5039       ALWAYS
5040
```

```
5041        READ TYPE.ORDER.
5042        IF SEQ. = 1  AND
5043           TYPE.ORDER NE "DEF"
5044           PRINT 1 LINE WITH REAL.UN.NO. THUS
5045           * * *      FIRST ORDER FOR UNIT **** SHOULD BE DEF
5046           LET ERROR.=1
5047        ALWAYS
5048
5049        LOOP UNTIL TYPE.ORDER.="LAST",
5050        DO
5051           CREATE AN ORDER
5052           LET ORD.TYPE(ORDER)=TYPE.ORDER.
5053           LET ORD.SEQ.NO(ORDER)=SEQ.
5054           IF ORD.TYPE(ORDER)="DEF"
5055              CALL ORD.DEF----------------------------------------------------------->(542)
5056              GIVEN
5057              ORDER
5058              IF SEQ.=1
5059                 SCHEDULE_AN ACT.DEF------------------------------------------------->(414)
5060                 GIVEN
5061                 UNIT.NO,
5062                 ORDER NOW ''                                                        \1
5063              ALWAYS
5064           ELSE
5065              IF ORD.TYPE(ORDER)="ATK"
5066                 CALL ORD.ATK---------------------------------------------------------->(543)
5067                 GIVEN
5068                 ORDER
5069              ELSE
5070                 IF ORD.TYPE(ORDER)="REINF"
5071                    CALL ORD.REINF---------------------------------------------------->(544)
5072                    GIVEN
5073                    ORDER
5074                 ELSE
5075                    IF ORD.TYPE(ORDER)="MOVDIS"
5076                       CALL ORD.MOVDIS------------------------------------------------>(545)
5077                       GIVEN
5078                       ORDER
5079                    ELSE
5080                       IF ORD.TYPE(ORDER)="MOVCOR"
5081                          CALL ORD.MOVCOR---------------------------------------------->(546)
5082                          GIVEN
5083                          ORDER,
5084                          UNIT.NO
5085                       ELSE
5086                          PRINT 1 LINE WITH ORD.TYPE(ORDER),
5087                          UNIT.NOS(I) THUS
5088                          ILLEGAL ORDER TYPE ****** FOR UNIT ******
5089                          LET ERROR.=1
5090                          START NEW CARD
5091                       ALWAYS
5092                    ALWAYS
5093                 ALWAYS
5094              ALWAYS
5095           ALWAYS
5096
5097           ADD 1 TO SEQ.
5098           FILE ORDER IN MU.ORDER.SET(UN.PTR(UNIT.NO.))
```

```
5099            READ TYPE ORDER.
5100         ENDLOOP
5101
5102         LET MU.CUR.ORDER(UN.PTR(UNIT.NO.))=1
5103         READ UNIT.NO.
5104      ENDLOOP
5105
5106      IF ERROR.=1
5107         PRINT 1 LINE THUS
5108         # # # # ERROR IN ORDER INPUT - SIMULATION STOPPED
5109 <-------STOP
5110      ALWAYS
5111
5112 <--EXITROUTINE
5113 ENDROUTINE
```

```
5114 ROUTINE ORD.DEF                                                    ''                    I011
5115     GIVEN
5116     ORDR.
5117
5118     ADD 1 TO ANAL.CTR(195,1)  ''                                                         \DYN_ANAL
5119     ''THIS ROUTINE READS INFORMATION PERTAINING TO A DEFEND ORDER.
5120     ''   INFO REQUIRED IS (1) A THRESHOLD VALUE EXPRESSED AS AN INTEGER PCT
5121     ''   OF CRITICAL EQUIPMENT BELOW WHICH REINFORCEMENTS WILL BE REQUESTED.
5122     ''   (**NOTE - THIS FEATURE HAS NOT BEEN IMPLEMENTED AS OF 24 JUNE 78.)
5123     ''   (2) THIS MISSION OF THE UNIT(4=DELAY,5=DEFEND,6=AMBUSH),
5124     ''   (3)WHICH ORDER IN THE UNITS SEQUENCE OF ORDERS SHOULD BE ASSIGNED IF                \1
5125     ''   THE ATTACKING UNIT BREAKS OFF THE ATTACK (SUCCESSFUL DEFENSE),
5126     ''   (4) WHICH ORDER IF THIS UNIT IS FORCED FROM THE DEFENSIVE POSITION
5127     ''   (UNSUCCESSFUL DEFENSE).
5128
5129     NORMALLY MODE IS INTEGER
5130
5131     CREATE A DEF.ORDER CALLED ORD.ID(ORDR.)
5132     READ REINF.THRESH(ORD.ID(ORDR.)),
5133     ORD.MISSION(ORD.ID(ORDR.)),
5134     EN.DIS.OP(ORD.ID(ORDR.)),
5135     OWN.DIS.OP(ORD.ID(ORDR.))
5136
5137 <--EXITROUTINE
5138 ENDROUTINE
```

```
5139 ROUTINE ORD.ATK                                                    ''                I012
5140     GIVEN
5141     ORDR.
5142
5143     ADD 1 TO ANAL.CTR(196,1)  ''                                                     \DYN_ANAL
5144     '' THIS ROUTINE READS INFORMATION PERTAINING TO AN ATTACK ORDER.  INFO
5145     ''  REQUIRED IS (1) WHICH ORDER IN THE UNITS SEQUENCE OF ORDERS SHOULD
5146     ''  BE ASSIGNED IF THE ENEMY WITHDRAWS FROM THE DEFENSIVE POSITION
5147     ''  (SUCCESSFUL ATTACK), (2) WHICH ORDER IS ADOPTED IF THIS UNIT MUST
5148     ''  BREAK OFF THE ATTACK (UNSUCCESSFUL ATTACK).
5149
5150     NORMALLY MODE IS INTEGER
5151
5152     CREATE AN ATK.ORDER CALLED ORD.ID(ORDR.)
5153     READ ENEMY.DO(ORD.ID(ORDR.)),
5154     OWN.DO(ORD.ID(ORDR.))
5155
5156 <--EXITROUTINE
5157 ENDROUTINE
```

```
5158 ROUTINE ORD.REINF                                                    ''                    I013
5159     GIVEN
5160     ORDR.
5161
5162     ADD 1 TO ANAL.CTR(197,1)  ''                                                          \DYN_ANAL
5163     ''THIS ROUTINE READS INFORMATION PERTAINING TO AN ORDER TO CALL FOR
5164     ''  REINFORCEMENTS.  THIS TYPE OF ORDER IS A CONTINGENCY ORDER
5165     ''  NORMALLY USED AFTER AN UNSUCCESSFUL ATTACK.  INFORMATION
5166     ''  REQUIRED IS (1) WHICH ORDER IN THE UNIT'S SEQUENCE OF ORDERS
5167     ''  SHOULD BE ASSIGNED IF A REINFORCING UNIT IS FOUND AND ARRIVES
5168     ''  AT THIS UNIT'S LOCATION, (2) WHICH ORDER SHOULD BE ADOPTED IF                        \1
5169     ''  NO REINFORCING UNIT IS AVAILABLE.
5170
5171     NORMALLY MODE IS INTEGER
5172
5173     CREATE A REINF.ORDER CALLED ORD.ID(ORDR.)
5174     READ SUC.REINF.OP(ORD.ID(ORDR.)),
5175     UNSUC.REINF.OP(ORD.ID(ORDR.))
5176
5177 <—EXITROUTINE
5178 ENDROUTINE
```

```
5179 ROUTINE ORD.MOVDIS                                                    ''   I014
5180     GIVEN
5181     ORDR.
5182
5183     ADD 1 TO ANAL.CTR(198,1)  ''                                          \DYN_ANAL
5184     ''THIS ROUTINE READS INFORMATION PERTAINING TO AN ORDER TO MOVE
5185     ''  A GIVEN DISTANCE IN A GIVEN DIRECTION.  THIS IS NOT AN AGGTRES-
5186     ''  SIVE MOVE - THAT IS, AN ATTACK CANNOT BE LAUNCHED FROM THIS
5187     ''  TYPE OF MOVE.  INFORMATION REQUIRED IS (1) THE DIRECTION OF THE
5188     ''  MOVE (ADVANC OR WITHDR), (2) THE DISTANCE OF THE MOVE IN METERS,
5189     ''  (3) THE TYPE OF MOVE (ADMIN OR TACTIC) WHICH INFLUENCES THE RATE OF
5190     ''  MOVEMENT, (4) WHICH ORDER IN THE UNIT'S SEQUENCE OF ORDERS SHOULD BE
5191     ''  ASSIGNED WHEN THE MOVE IS COMPLETED.
5192
5193     NORMALLY MODE IS INTEGER
5194
5195     CREATE A MOVDIS.ORDER CALLED ORD.ID(ORDR.)
5196     READ DIR.OF.MOVE(ORD.ID(ORDR.)),
5197     DIST.MOVED(ORD.ID(ORDR.)),
5198     MOVE.TYPE(ORD.ID(ORDR.)),
5199     ORD.NEXT(ORD.ID(ORDR.))
5200
5201 <--EXITROUTINE
5202 ENDROUTINE
```

```
5203 ROUTINE ORD.MOVCOR                                                                                    ''                                                                                
```

```
5255 ROUTINE P.E.M.INPUT                                                    ''          I016
5256
5257    ADD 1 TO ANAL.CTR(200.1)  ''                                                    \DYN_ANAL
5258    ''THIS ROUTINE INITIALIZES THE POSTURE, ENVIRONMENT, AND MISSION
5259    ''DATA SETS.
5260
5261    NORMALLY MODE IS INTEGER
5262
5263    READ N.POSTURE
5264    CREATE EVERY POSTURE
5265
5266    FOR EACH POSTURE
5267    READ PT.NAME(POSTURE)
5268
5269    READ N.ENVIRONMENT
5270    CREATE EVERY ENVIRONMENT
5271
5272    FOR EACH ENVIRONMENT
5273    READ EN.NAME(ENVIRONMENT)
5274
5275    READ N.MISSION
5276    CREATE EVERY MISSION
5277
5278    ''EACH UNIT HAS A PERCENTAGE IN EACH ENVIRONMENT
5279    FOR EACH MISSION
5280    READ MN.NAME(MISSION)
5281
5282    ''EACH CATEGORY HAS AN EXPECTED PERCENTAGE IN EACH ENVIRONMENT
5283    LOOP FOR EACH ENVIRONMENT
5284    DO
5285       LOOP FOR EACH CATEGORY
5286       DO
5287          READ EC.FRACT(ENVIRONMENT, CATEGORY)  ''IN PERCENT
5288       ENDLOOP
5289    ENDLOOP
5290
5291    ''EACH CATEGORY IN EACH MISSION HAS A PERCENTAGE IN EACH POSTURE
5292    ''WHICH VARIES DEPENDING ON WHETHER UNIT HAS BEEN WARNED
5293    LOOP FOR EACH CATEGORY
5294    DO
5295       LOOP FOR EACH MISSION
5296       DO
5297          LOOP FOR EACH POSTURE
5298          DO
5299             READ CPM.WARNED.FRACT(CATEGORY, POSTURE, MISSION),
5300             CPM.UNWARNED.FRACT(CATEGORY, POSTURE, MISSION)
5301          ENDLOOP
5302       ENDLOOP
5303    ENDLOOP
5304
5305 <—EXITROUTINE
5306 ENDROUTINE
```

```
5307 ROUTINE TB.INPUT                                                    ''                                                    I017
5308
5309    ADD 1 TO ANAL.CTR(201,1)  ''                                                                                          \DYN_ANAL
5310    ''THIS ROUTINE CREATES EACH TYPE BATTERY AND THE LIST OF MUNITIONS
5311    ''IT CAN FIRE
5312
5313    NORMALLY MODE IS INTEGER
5314    DEFINE HOW.NAME AS A TEXT VARIABLE ''                                                                                 \TEXT
5315
5316    LET N.TB.N.FM = 5  '' THE NUMBER OF FIRE MISSIONS IN QUEUE
5317    CREATE EVERY TB.N.FM
5318
5319    READ N.TYPE.BTRY
5320    CREATE EVERY TYPE.BTRY
5321
5322    LOOP FOR EACH TYPE.BTRY
5323    DO
5324       READ TB.SEQ.NO,
5325       TB.NAME(TYPE.BTRY),
5326       TB.SHOOT.SCOOT.IND(TYPE.BTRY), ''1=YES, 0=NO
5327       HOW.NAME
5328       FOR EACH EQUIPMENT
5329       WITH EQ.NAME(EQUIPMENT) = HOW.NAME
5330       FIND THE FIRST CASE
5331       IF NONE
5332          PRINT 1 LINE WITH HOW.NAME THUS
5333          = = = NO EQUIPMENT CALLED ****** IN THE FILE = = =
5334 <--------STOP
5335       OTHERWISE
5336       LET TB.HOW.EQ.ID(TYPE.BTRY) = EQUIPMENT
          READ TB.RND.PER.LAUNCH(TYPE.BTRY),
          TB.MIN.HOW(TYPE.BTRY),
5339      TB.SUST.FIRE.RATE(TYPE.BTRY),  ''100.*ROUNDS PER MINUTE
5340      TB.MAX.RANGE(TYPE.BTRY),  ''IN DECAMETERS
5341      TB.MAX.RAP.RANGE(TYPE.BTRY),  ''IN DECAMETERS
5342      TB.SFAIL.MEAN.RNDS(TYPE.BTRY),
5343      TB.LFAIL.MEAN.RNDS(TYPE.BTRY),
5344      TB.SFAIL.REPAIR(TYPE.BTRY),  ''IN HOURS * 10.
5345      TB.LFAIL.REPAIR(TYPE.BTRY),  ''IN HOURS * 10.
5346      TB.SUPPRESS.TIME(TYPE.BTRY),  ''IN MINUTES
5347      TB.MIN.PREP(TYPE.BTRY),  ''IN MINUTES * 10.
5348      TB.MAX.PREP(TYPE.BTRY),  ''IN MINUTES * 10.
5349      TB.MIN.FEBA(TYPE.BTRY), ''IN DECAMETERS
5350      TB.MAX.FEBA(TYPE.BTRY), ''IN DECAMETERS
5351      TB.MARCH.ORDER(TYPE.BTRY), ''IN MINUTES
5352      TB.OCCUPY(TYPE.BTRY),    ''IN MINUTES
5353      TB.MN.FASCAM.SUPP(TYPE.BTRY),  ''MINUTES
5354      TB.MX.FASCAM.SUPP(TYPE.BTRY)   ''MINUTES
5355      LET TB.MAX.RANGE(TYPE.BTRY) = TB.MAX.RANGE(TYPE.BTRY) * 10 /16. ''                                                    \OPTIMIZE
5356      LET TB.MAX.RAP.RANGE(TYPE.BTRY) = TB.MAX.RAP.RANGE(TYPE.BTRY) * 10 /16.
5357      LET TB.MAX.FEBA(TYPE.BTRY) = TB.MAX.FEBA(TYPE.BTRY) * 10 /16.
5358      LET TB.MIN.FEBA(TYPE.BTRY) = TB.MIN.FEBA(TYPE.BTRY) * 10 /16.
5359
5360      LET FLAG = 0
5361      ''READ WHAT MUNITIONS TO HANG ON THIS TYPE BATTERY
5362      LOOP UNTIL FLAG = 1
5363      DO
5364         CREATE A TB.TM.LINK
```

```
5365          READ TB.TM.RAP(TB.TM.LINK),
5366          TB.TM.CLASS(TB.TM.LINK),
5367          TB.TM(TB.TM.LINK)
5368          IF TB.TM.RAP(TB.TM.LINK) = 9
5369             DESTROY THIS TB.TM.LINK
5370             LET FLAG = 1
5371          ELSE
5372             FILE TB.TM.LINK IN TB.TM.LIST(TYPE.BTRY)
5373          ALWAYS
5374       ENDLOOP
5375
5376       ''READ THE NECESSARY FIRE MISSION PRIORITY SCHEME FOR THIS
5377       ''TYPE BATTERY TO CONSIDER, GIVEN A BATTERY OF THIS TYPE ALREADY
5378       ''HAVING TB-N-FM FIRE MISSIONS.
5379       LOOP FOR EACH TB.N.FM
5380       DO
5381          READ TB.MW.THRESHOLD(TYPE.BTRY, TB.N.FM)
5382       ENDLOOP
5383    ENDLOOP
5384
5385 <--EXITROUTINE
5386 ENDROUTINE
```

```
5387 ROUTINE BTRY.INPUT                                                    ''                I018
5388
5389     ADD 1 TO ANAL.CTR(202,1)  ''                                                        \DYN_ANAL
5390     '' THIS ROUTINE CREATES EACH FIRING BATTALION, CREATES EVERY BATTERY
5391     '' AND PLACES IT IN A BATTALION AND FINALLY CREATES EVERY HOWITZER
5392     '' AND PLACES IT IN THE APPROPRIATE BATTERY.
5393
5394     NORMALLY MODE IS INTEGER
5395
5396     LET .ERROR = 0
5397     READ N.FA.BN
5398     CREATE EVERY FA.BN
5399
5400     FOR EACH FA.BN
5401     READ FB.MISSION(FA.BN)
5402
5403     READ N.BTRY
5404     CREATE EVERY BTRY
5405
5406     LET K=1
5407     READ FA.BN.UNIT(K),
5408     DUMMY
5409     LOOP FOR EACH BTRY
5410     DO
5411        'START'
5412        IF DUMMY=999
5413           READ DUMMY
5414           IF DUMMY=999
5415              IF BTRY<N.BTRY
5416                 SKIP 1 LINE
5417                 PRINT 2 LINES WITH N.BTRY,BTRY THUS
5418                 NOT ENOUGH BATTERIES HAVE BEEN LISTED IN BATTERY FILE -
5419                 **** WERE EXPECTED, BUT ONLY **** WERE FOUND.
5420
5421 <---------------STOP
5422              OTHERWISE
5423              GO TO FINISH
5424           ALWAYS
5425           ADD 1 TO K
5426           IF K>N.FA.BN
5427              SKIP 1 LINE
5428              PRINT 2 LINES WITH N.FA.BN THUS
5429              TOO MANY BATTALIONS HAVE BEEN LISTED IN BATTERY FILE -
5430              ONLY **** WERE EXPECTED.
5431
5432
5433 <------------STOP
5434           OTHERWISE
5435           LET FA.BN.UNIT(K)=DUMMY
5436           READ DUMMY
5437           GO TO START
5438        ALWAYS
5439
5440        LET BY.BN(BTRY)=K
5441        LET BY.TYPE(BTRY)=DUMMY
5442        READ REAL.UNIT
5443        READ BY.PGM.CAP(BTRY)
5444        READ DUMMY
```

```
5445        LOOP FOR I = 1 TO N.UNIT,
5446        WITH REAL.UNIT = UNIT.NOS(I),
5447        FIND THE FIRST CASE
5448        IF NONE
5449           SKIP 1 LINE
5450           PRINT 1 LINE WITH BTRY, REAL.UNIT THUS
5451           BY.UNIT(****)******** NOT FOUND IN READING BTRY.INPUT.
5452
5453
5454        ALWAYS
5455        LET BY.UNIT(BTRY) = I
5456        LET UN.BTRY.INDEX(BY.UNIT(BTRY)) = BTRY
5457        LET TB = BY.TYPE(BTRY)
5458        FILE BTRY IN BN.BTRY.SET(BY.BN(BTRY))
5459
5460
5461
5462        FOR EACH EQ IN UN.EQUIP.LIST(BY.UNIT(BTRY))
5463        WHEN UE.ID(EQ) = TB.HOW.EQ.ID(TB)
5464        FIND THE FIRST CASE
5465        IF NONE
5466           IF TB.RND.PER.LAUNCH(TB) > 0
5467              LET .ERROR = 1
5468              PRINT 1 LINE WITH UNIT.NOS(BY.UNIT(BTRY)) THUS
5469              = ERROR = UNIT ****** DOES NOT HAVE CORRECT ARTY EQUIP
5470
5471
5472           ALWAYS
5473        ALWAYS
5474        FOR I = 1 TO BTRY - 1
5475        WITH BY.UNIT(I) = BY.UNIT(BTRY)
5476        FIND THE FIRST CASE
5477        IF FOUND
5478           LET .ERROR = 1
5479           PRINT 1 LINE WITH UNIT.NOS(BY.UNIT(I)) THUS
5480           === ERROR = BTRY UNIT ******* IS LISTED MORE THAN ONCE IN BTRY FILE
5481
5482
5483        ALWAYS
5484
5485        IF UE.QUANT(EQ) > 0 AND TB.RND.PER.LAUNCH(TB) > 0
5486
5487
5488           LOOP UNTIL N.BY.HOW.SET(BTRY) = UE.QUANT(EQ)
5489           DO
5490              CREATE A HOW
5491              FILE HOW IN BY.HOW.SET(BTRY)
5492              LET HW.BTRY(HOW) = BTRY
5493              IF TB.SFAIL.MEAN.RNDS(TB) GT 0
5494                 ''UTILIZE EXPONENTIAL.F FUNCTION IN NEXT DRAW ———————————————————————— \1>(640)
5495                 LET HW.SFAIL.RNDS(HOW) =
5496                 EXPONENTIAL.F(REAL.F(TB.SFAIL.MEAN.RNDS(TB)),
5497                 RN.SEED)
5498              ALWAYS
5499              IF TB.LFAIL.MEAN.RNDS(TB) GT 0
5500                 ''UTILIZE EXPONENTIAL.F FUNCTION IN NEXT DRAW ———————————————————————— \1>(640)
5501                 LET HW.LFAIL.RNDS(HOW) =
5502                 EXPONENTIAL.F(REAL.F(TB.LFAIL.MEAN.RNDS(TB)),
```

```
5503                RN.SEED)
5504              ALWAYS
5505            ENDLOOP
5506         ALWAYS
5507      ENDLOOP
5508
5509
5510      READ DUMMY
5511      IF DUMMY NE 999
5512         SKIP 1 LINE
5513         PRINT 1 LINE THUS
5514         MORE DATA THAN EXPECTED APPEARS IN THE BATTERY FILE.
5515
5516
5517      ALWAYS
5518      'FINISH'
5519
5520
5521      IF K<N.FA.BN
5522         SKIP 1 LINE
5523         PRINT 2 LINES WITH N.FA.BN,K THUS
5524         NOT ENOUGH BATTALIONS HAVE BEEN LISTED IN BATTERY FILE -
5525         **** WERE EXPECTED, BUT ONLY **** WERE FOUND.
5526
5527
5528 <-------STOP
5529      OTHERWISE
5530      IF .ERROR = 1
5531
5532
5533 <-------STOP
5534      OTHERWISE
5535
5536
5537 <--EXITROUTINE
5538 ENDROUTINE
```

```
5539 ROUTINE FBN.FD.INPUT                                                    ''
5540
5541    ADD 1 TO ANAL.CTR(203,1)  ''                                         \DYN_ANAL
5542    '' THIS ROUTINE CREATES EACH FDC AND ESTABLISHES THE CONTROL
5543    '' RELATIONSHIPS OF FDC'S OVER FIRING BATTALIONS.
5544
5545    NORMALLY MODE IS INTEGER
5546    DEFINE FD.PARENT AS A 1-DIMENSIONAL INTEGER ARRAY
5547
5548    READ N.FDC
5549    CREATE EVERY FDC
5550
5551    RESERVE FD.UNIT,FD.PARENT AS N.FDC
5552
5553    LOOP FOR EACH FDC
5554    DO
5555       READ FD.UNIT(FDC),
5556       FD.PARENT(FDC),
5557       FD.MIN.TIME(FDC),
5558       FD.MAX.TIME(FDC),
5559       FD.TGT.THRESHOLD(FDC),
5560       DUMMY
5561       LOOP UNTIL DUMMY=999
5562       DO
5563          CREATE AN FD.BN.LINK
5564          FOR K=1 TO N.FA.BN
5565          WITH FA.BN.UNIT(K) = DUMMY
5566          FIND THE FIRST CASE
5567          IF NONE
5568             SKIP 1 LINE
5569             PRINT 1 LINE WITH DUMMY THUS
5570             BATTALION NUMBER **** REFERENCED IN THE FDC FILE DOES NOT EXIST.
5571 <----------STOP
5572          ALWAYS
5573          LET FB.BN(FD.BN.LINK)=K
5574          FILE FD.BN.LINK IN FD.BN.LIST(FDC)
5575          READ DUMMY
5576       ENDLOOP
5577    ENDLOOP
5578
5579    LOOP FOR EACH FDC
5580    DO
5581       FOR K=1 TO N.FDC
5582       WITH FD.UNIT(K) = FD.PARENT(FDC)
5583       FIND THE FIRST CASE
5584       IF NONE
5585          IF FD.PARENT(FDC) NE 0
5586             SKIP 1 LINE
5587             PRINT 2 LINES WITH FD.PARENT(FDC),FD.UNIT(FDC) THUS
5588             FDC UNIT # **** REFERENCED AS THE PARENT OF FDC UNIT # ****
5589             DOES NOT EXIST.
5590 <----------STOP
5591          ALWAYS
5592          LET K=0
5593       ALWAYS
5594       LET FD.FDC(FDC)=K
5595    ENDLOOP
5596
```

```
5597
5598     RELEASE FD.PARENT(*)
5599
5600 <--EXITROUTINE
5601 ENDROUTINE
```

```
5602 ROUTINE MUNS.INPUT                                                    ''                    I020
5603
5604    ADD 1 TO ANAL.CTR(204,1)  ''                                                             \DYN_ANAL
5605    ''THIS ROUTINE READS IN THE ATTRIBUTES OF ALL ICM AND HE MUNITIONS
5606    ''WITH THE APPROPRIATE FUZES
5607
5608    NORMALLY MODE IS INTEGER
5609
5610    READ N.HE.MUNITION,
5611    N.IC.MUNITION,
5612    N.FUZE
5613
5614    CREATE EVERY HE.MUNITION, IC.MUNITION, FUZE
5615
5616    ''DETERMINE THE TOTAL NUMBER OF HE RANGE HACKS
5617    LOOP FOR EACH TYPE.BTRY
5618    FOR EACH TB.TM.LINK IN  TB.TM.LIST(TYPE.BTRY)
5619    WITH TB.TM.CLASS(TB.TM.LINK) EQ "HE"
5620    DO THIS
5621       ADD 2 TO N.HE.RANGE.HACK
5622    ENDLOOP
5623
5624    ''DETERMINE THE TOTAL NUMBER OF ICM RANGE HACKS
5625    LOOP FOR EACH TYPE.BTRY
5626    FOR EACH TB.TM.LINK IN  TB.TM.LIST(TYPE.BTRY)
5627    WITH TB.TM.CLASS(TB.TM.LINK) EQ "ICM"
5628    DO
5629       ADD 2 TO N.IC.RANGE.HACK
5630    ENDLOOP
5631
5632    CREATE EVERY HE.RANGE.HACK, IC.RANGE.HACK
5633
5634    LET COUNT = 0
5635    FOR EACH FUZE
5636    READ FZ.NAME(FUZE)
5637
5638    LOOP FOR EACH HE.MUNITION
5639    DO THIS
5640       READ HE.ID(HE.MUNITION),
5641       HE.MIN.MARG.EFF(HE.MUNITION),
5642       HE.WEIGHT(HE.MUNITION),
5643       HE.COST(HE.MUNITION),
5644       HE.VOLLEY.RAD(HE.MUNITION),
5645       HE.ROUND.RAD(HE.MUNITION),
5646       .DUST.RAD,                          ''METERS
5647       HE.DUST.DURATION(HE.MUNITION)   ''MINUTES
5648       LET HE.VOL.DUST.RAD(HE.MUNITION) = .DUST.RAD / 16   ''HDM
5649       LOOP FOR EACH FUZE
5650       DO THIS
5651          READ FZ.HE.RELY(FUZE, HE.MUNITION)   ''TIMES 100
5652       ENDLOOP
5653
5654       ''READ ATTRIBUTES OF EACH RANGE HACK (WHICH IS ASSOCIATED WITH
5655       ''A SPECIFIC HE MUNITION FIRED BY A SPECIFIC TYPE BATTERY
5656       ''AT A SPECIFIC RANGE
5657       LOOP FOR EACH TYPE.BTRY
5658       FOR EACH TB.TM.LINK IN TB.TM.LIST(TYPE.BTRY)
5659       WITH TB.TM(TB.TM.LINK) EQ HE.MUNITION AND
```

```
5660        TB.TM.CLASS(TB.TM.LINK) EQ "HE"
5661        DO
5662           LET A = COUNT + 1
5663           LET B = COUNT + 2
5664           LOOP FOR RH = A  TO  B
5665           DO
5666              FILE THE RH IN HE.TB.RH.LIST(HE.MUNITION, TYPE.BTRY)
5667              READ R.RANGE,
5668              HE.RH.TOTAL.CPE(RH),
5669              HE.RH.ROUND.CPE(RH)
5670              LET HE.RH.RANGE(RH) = R.RANGE*10/16.   ''                                    \1
5671           ENDLOOP
5672           LET COUNT = COUNT + 2
5673        ENDLOOP
5674     ENDLOOP
5675
5676     LET COUNT = 0
5677     LOOP FOR EACH IC.MUNITION
5678     DO
5679        READ IC.ID(IC.MUNITION),
5680        IC.MIN.MARG.EFF(IC.MUNITION),
5681        IC.WEIGHT(IC.MUNITION),
5682        IC.COST(IC.MUNITION),
5683        IC.RELIABILITY(IC.MUNITION),  ''TIMES 100.
5684        IC.SUBM.INDEX(IC.MUNITION),
5685        IC.VOLLEY.RAD(IC.MUNITION),
5686        IC.N.SUBM(IC.MUNITION)
5687        LOOP FOR EACH TYPE.BTRY
5688        FOR EACH TB.TM.LINK IN TB.TM.LIST(TYPE.BTRY)
5689        WITH TB.TM(TB.TM.LINK) EQ IC.MUNITION AND
5690        TB.TM.CLASS(TB.TM.LINK) EQ "ICM"
5691        DO THIS
5692           READ IC.TB.SLOPE(IC.MUNITION, TYPE.BTRY),
5693           IC.TB.INTERCEPT(IC.MUNITION, TYPE.BTRY)
5694           LET A = COUNT + 1
5695           LET B = COUNT + 2
5696           ''READ ATTRIBUTES OF EACH RANGE HACK (WHICH IS ASSOCIATED WITH
5697           ''A SPECIFIC ICM MUNITION FIRED BY A SPECIFIC TYPE BATTERY
5698           ''AT A SPECIFIC RANGE
5699           LOOP FOR RH = A  TO  B
5700           DO
5701              FILE THE RH IN IC.TB.RH.LIST(IC.MUNITION, TYPE.BTRY)
5702              READ R.RANGE,
5703              IC.RH.TOTAL.CPE(RH),
5704              IC.RH.ROUND.CPE(RH)
5705              LET IC.RH.RANGE(RH)= R.RANGE*10/.16  ''                                   \OPTIMIZE
5706           ENDLOOP
5707           LET COUNT = COUNT + 2
5708        ENDLOOP
5709     ENDLOOP
5710
5711 <--EXITROUTINE
5712 ENDROUTINE
```

```
5713 ROUTINE SUBM.INPUT                                                    ''  I021
5714
5715    ADD 1 TO ANAL.CTR(205,1)  ''                                           \DYN_ANAL
5716    ''THIS ROUTINE READS IN THE DATA SET FOR EACH SUBMUNITION
5717
5718    NORMALLY MODE IS INTEGER
5719    DEFINE PERSONNEL TO MEAN 1
5720
5721    READ N.SUBMUNITION
5722    CREATE EVERY SUBMUNITION
5723    FOR EACH SUBMUNITION
5724    READ SM.NAME(SUBMUNITION)
5725
5726    LOOP FOR EACH SUBMUNITION
5727    FOR EACH ENVIRONMENT
5728    DO
5729       READ ES.RELY(ENVIRONMENT,SUBMUNITION)  ''TIMES 100
5730    ENDLOOP
5731
5732    LOOP FOR EACH SUBMUNITION
5733    FOR EACH ENVIRONMENT
5734    FOR EACH POSTURE
5735    DO
5736       READ EPS.LA.PERS(ENVIRONMENT,POSTURE,SUBMUNITION) ''IN SQ.METERS
5737    ENDLOOP
5738
5739    LOOP FOR EACH SUBMUNITION
5740    FOR EACH ENVIRONMENT
5741    FOR EACH TYPE.EQUIPMENT
5742    UNLESS TYPE.EQUIPMENT = PERSONNEL OR
5743    TE.NAME(TYPE.EQUIPMENT) = "MINES"
5744    DO
5745       READ TES.LA.EQUIP(TYPE.EQUIPMENT,ENVIRONMENT,SUBMUNITION)''IN SQ.M.*10
5746    ENDLOOP
5747
5748 <--EXITROUTINE
5749 ENDROUTINE
```

```
5750 ROUTINE HE.LA.INPUT                                                    ''
5751
5752     ADD 1 TO ANAL.CTR(206,1)  ''
5753     ''THIS ROUTINE READS IN LETHALITY DATA FOR EACH HE MUNITION
5754
5755     NORMALLY MODE IS INTEGER
5756     DEFINE PERSONNEL TO MEAN 1
5757
5758     LOOP FOR EACH HE.MUNITION
5759     FOR EACH TYPE.BTRY
5760     FOR EACH FUZE
5761     FOR EACH POSTURE
5762     FOR EACH ENVIRONMENT
5763     FOR EACH RH IN HE.TB.RH.LIST(HE.MUNITION, TYPE.BTRY)
5764     DO THIS
5765        LET .REPEAT = 0
5766        IF TYPE.BTRY GT 1
5767           LET I = TYPE.BTRY
5768           FOR J = 1 TO I-1
5769           DO THIS
5770              FOR EACH R.H IN HE.TB.RH.LIST(HE.MUNITION,J)
5771              WITH R.H EQ RH-2
5772              FIND THE FIRST CASE
5773              IF FOUND
5774                 LET .REPEAT  =  1
5775           <------EXITLOOP
5776              OTHERWISE
5777           ENDLOOP
5778        ALWAYS
5779        IF .REPEAT EQ 1
5780           LET REPF.LA.PERS(RH,ENVIRONMENT,POSTURE,FUZE) =
5781           REPF.LA.PERS(RH-2,ENVIRONMENT,POSTURE,FUZE)
5782        ELSE
5783           READ REPF.LA.PERS(RH,ENVIRONMENT,POSTURE,FUZE)
5784        ALWAYS
5785     ENDLOOP
5786
5787     LOOP FOR EACH HE.MUNITION
5788     FOR EACH TYPE.BTRY
5789     FOR EACH FUZE
5790     FOR EACH ENVIRONMENT
5791     FOR EACH TYPE.EQUIPMENT
5792     UNLESS TYPE.EQUIPMENT = PERSONNEL OR
5793     TE.NAME(TYPE.EQUIPMENT) = "MINES"
5794     FOR EACH RH IN HE.TB.RH.LIST(HE.MUNITION, TYPE.BTRY)
5795     DO THIS
5796        LET .REPEAT = 0
5797        IF TYPE.BTRY GT 1
5798           LET I = TYPE.BTRY
5799           FOR J = 1 TO I-1
5800           DO THIS
5801              FOR EACH R.H IN HE.TB.RH.LIST(HE.MUNITION,J)
5802              WITH R.H EQ RH-2
5803              FIND THE FIRST CASE
5804              IF FOUND
5805                 LET .REPEAT  =  1
5806           <------EXITLOOP
5807              OTHERWISE
```

```
5808          ENDLOOP
5809       ALWAYS
5810       IF .REPEAT EQ 1
5811          LET RTEF.LA.EQUIP(RH,TYPE.EQUIPMENT,ENVIRONMENT,FUZE) =
5812          RTEF.LA.EQUIP(RH-2,TYPE.EQUIPMENT,ENVIRONMENT,FUZE)
5813       ELSE
5814          READ RTEF.LA.EQUIP(RH,TYPE.EQUIPMENT,ENVIRONMENT,FUZE)
5815       ALWAYS
5816    ENDLOOP
5817
5818 <--EXITROUTINE
5819 ENDROUTINE
```

```
5820 ROUTINE  RUL.EN.INPUT                                                  ''
5821
5822    ADD 1 TO ANAL.CTR(207,1)  ''
5823    NORMALLY MODE IS INTEGER
5824
5825    READ  N.DIST.FROM.FEBA.BAND
5826    CREATE EVERY DIST.FROM.FEBA.BAND
5827
5828    LOOP FOR EACH DIST.FROM.FEBA.BAND
5829    DO
5830       READ DFFB.MAX.RANGE(DIST.FROM.FEBA.BAND)
5831    ENDLOOP
5832
5833    LOOP FOR EACH CATEGORY CALLED CAT
5834    FOR EACH DIST.FROM.FEBA.BAND CALLED DFFB
5835    FOR EACH IC.MUNITION CALLED ICM
5836    DO
5837       READ CDI.USAGE.INDICATOR(CAT,DFFB,ICM)
5838    ENDLOOP
5839
5840    LOOP FOR EACH CATEGORY CALLED CAT
5841    FOR EACH DIST.FROM.FEBA.BAND CALLED DFFB
5842    FOR EACH TYPE.BTRY CALLED TB
5843    DO
5844       READ CDT.MAX.VOLS(CAT,DFFB,TB)
5845    ENDLOOP
5846
5847    LOOP FOR EACH DIST.FROM.FEBA.BAND CALLED DFFB
5848    FOR EACH TYPE.UNIT CALLED TU
5849    DO
5850       READ DT.MAX.BATS(DFFB,TU)
5851    ENDLOOP
5852
5853 <—EXITROUTINE
5854 ENDROUTINE
```

```
5855 ROUTINE ST.INPUT                                                    ''                                                I024
5856
5857     ''THIS ROUTINE READS IN TYPE SENSOR DATA
5858
5859     ADD 1 TO ANAL.CTR(208,1)  ''                                                                                \DYN_ANAL
5860     NORMALLY MODE IS INTEGER
5861
5862     READ N.SENSOR.TYPE
5863     CREATE EVERY SENSOR.TYPE
5864
5865     LOOP FOR EACH SENSOR.TYPE
5866     DO
5867        READ ST.NAME(SENSOR.TYPE),
5868        ST.MIN.XMIT(SENSOR.TYPE),  ''IN MINUTES * 10
5869        ST.MAX.XMIT(SENSOR.TYPE),  ''IN MINUTES * 10
5870        ST.TE.PTR(SENSOR.TYPE),
5871        ST.MAX.RANGE(SENSOR.TYPE)  ''IN DECAMETERS
5872        ''CONVERT RANGE TO HEXADECAMETERS
5873        LET ST.MAX.RANGE(SENSOR.TYPE) = ST.MAX.RANGE(SENSOR.TYPE)*10/16 ''                                          \OPTIMIZE
5874     ENDLOOP
5875
5876 <--EXITROUTINE
5877 ENDROUTINE
```

```
5878 ROUTINE MCFR.INPUT                                                    ''          1025
5879
5880    ADD 1 TO ANAL.CTR(209,1)  ''                                                   \DYN_ANAL
5881    NORMALLY MODE IS INTEGER
5882
5883    READ N.CFR.RNG.HACK
5884    CREATE EVERY CFR.RNG.HACK
5885
5886    READ N.MODEL.CFR
5887    CREATE EVERY MODEL.CFR
5888
5889    LET TOT.RH = 0
5890    LET OLD.TOT.RH = 1
5891    LOOP FOR EACH MODEL.CFR CALLED MCFR
5892    DO
5893       READ MCFR.NAME(MCFR),
5894       MCFR.EQ.ID(MCFR),
5895       MCFR.MIN.OFF(MCFR),
5896       MCFR.MAX.ON(MCFR),
5897       MCFR.SWEEP.ANGLE(MCFR),
5898       WIDTH    ''IN KILOMETERS
5899       LET MCFR.SEARCH.WIDTH(MCFR) = WIDTH*1000/16.  ''                            \OPTIMIZE
5900       READ NUM.RH
5901       ADD NUM.RH TO TOT.RH
5902       LOOP FOR RH = OLD.TOT.RH TO TOT.RH
5903       DO
5904          FILE RH IN MCFR.RH.LIST(MCFR)
5905          READ RANGE,
5906          CFR.DET.PROB(RH),
5907          CFR.CIR.ERROR(RH)
5908          LET CFR.RH.RANGE(RH) = RANGE/16
5909       ENDLOOP
5910       ADD NUM.RH TO OLD.TOT.RH
5911    ENDLOOP
5912
5913 <--EXITROUTINE
5914 ENDROUTINE
```

```
5915 ROUTINE MPDB.INPUT                                                                                    ''                                                                                
```

```
5953 ROUTINE MAO.INPUT                                                    ''                    I027
5954
5955     ADD 1 TO ANAL.CTR(211,1)  ''                                                          \DYN_ANA:
5956     '' THIS ROUTINE SETS THE ATTRIBUTES OF THE MODEL
5957     '' AERIAL OBSERVER   THE SORTIES RATE PER SECTOR PER DAY
5958     '' IS SPECIFIED IN THE SENSOR.INPUT ROUTINE.
5959
5960     NORMALLY  MODE IS INTEGER
5961
5962     READ  N.MODEL.AO
5963     CREATE EVERY MODEL.AO
5964
5965     LOOP FOR EACH MODEL.AO CALLED MAO,
5966     DO
5967        READ MAO.NAME(MAO),
5968        MAO.EQ.ID(MAO),
5969        MAO.VELOCITY(MAO),          ''IN KM PER HOUR
5970        MAO.ALTITUDE(MAO),          '' MAXIMUM ALTITUDE IN DM
5971        MAO.MAGNIFICATION(MAO),     '' X POWER
5972        MAO.MAX.ALOFT.TIME(MAO),    '' IN MINUTES
5973        MAO.MIN.PREP(MAO),          ''MINUTES TO PREP FOR SORTIE
5974        MAO.MAX.PREP(MAO),          ''MINUTES TO PREP FOR SORTIE
5975        MAO.PGM.CAP(MAO)            ''LASER DESIGNATOR CAPABILITY
5976     ENDLOOP
5977
5978     READ N.AO.ELEVATION.BAND
5979     CREATE EVERY AO.ELEVATION.BAND     '' ELEVATION 'ALTITUDE' RANGES
5980
5981     LOOP FOR EACH AO.ELEVATION.BAND CALLED AEB,
5982     DO
5983        READ AO.EB.ALTITUDE(AEB)
5984        LET AO.EB.ALTITUDE(AEB) = AO.EB.ALTITUDE(AEB)/16
5985        FILE AEB IN AO.EB.SET
5986     ENDLOOP
5987
5988     READ N.AO.RANGE.BAND
5989     CREATE EVERY AO.RANGE.BAND
5990
5991     LOOP FOR EACH AO.RANGE.BAND CALLED ARB
5992     DO
5993        READ AO.RB.RANGE(ARB)  ''  MAX RANGE OF EACH BAND
5994        LET AO.RB.RANGE(ARB) = AO.RB.RANGE(ARB)/16
5995        FILE ARB IN AO.RB.SET
5996     ENDLOOP
5997     LOOP FOR EACH AO.ELEVATION.BAND CALLED AEB,
5998     FOR EACH AO.RANGE.BAND CALLED ARB
5999     DO
6000        READ AO.PROB.LOS(AEB,ARB),
6001        AO.VISIBILITY(AEB,ARB)
6002     ENDLOOP
6003     LOOP FOR EACH MODEL.AO CALLED MAO,
6004     FOR EACH AO.RANGE.BAND CALLED ARB,
6005     DO
6006        READ MAOR.CIR.ERROR(MAO,ARB)  '' TARGET LOCATION ESTIMATION ERROR
6007     ENDLOOP
6008
6009 <--EXITROUTINE
6010 ENDROUTINE
```

```
6011 ROUTINE SENSOR.INPUT                                                    ''  I028
6012
6013    ADD 1 TO ANAL.CTR(212,1)  ''                                             \DYN_ANAL
6014    NORMALLY MODE IS INTEGER
6015    ''DISTANCE NEEDS BE DEFINED AS A REAL VARIABLE SO THAT THE FUNCTION
6016    ''SIGN.F WILL EXECUTE CORRECTLY
6017    DEFINE DISTANCE AS A REAL VARIABLE  ''                                   \VAX_CHG_3
6018    DEFINE INTERVAL AS A REAL VARIABLE
6019
6020    LET LAST.UNIT = FALSE
6021    LOOP UNTIL LAST.UNIT = TRUE
6022    DO
6023       READ SENS.UNIT.NO.
6024       IF SENS.UNIT.NO. GE 99999
6025          LET LAST.UNIT = TRUE
6026    <------CYCLE
6027       OTHERWISE
6028
6029       FOR I = 1 TO N.UNIT,
6030       WITH SENS.UNIT.NO. = UNIT.NOS(I),
6031       FIND THE FIRST CASE
6032       IF NONE
6033          SKIP 1 LINE
6034          PRINT 1 LINE WITH SENS.UNIT.NO. THUS
6035          = = = SENSOR INPUT UNIT ****** IS NOT LISTED IN UNIT DATA = = =
6036          LET ERROR.MSG = 1
6037          LET SENSOR.UNIT = SENS.UNIT.NO.   '' CONTINUE TO READ DATA
6038       ELSE
6039          LET SENSOR.UNIT = I
6040       ALWAYS
6041       LET ASIDE = UN.COLOR(SENSOR.UNIT)
6042       LET LAST.SENSOR = FALSE
6043       LOOP UNTIL LAST.SENSOR = TRUE
6044       DO
6045          READ SENS.TYPE,
6046          SENSOR.MODEL
6047          IF SENS.TYPE GE 99999
6048             LET LAST.SENSOR = TRUE
6049       <------CYCLE
6050          OTHERWISE
6051
6052          READ NUM.SENSOR
6053          IF ST.NAME(SENS.TYPE) = "AD"
6054             ''FDC NOT NEEDED, NO VALUE WILL BE READ.
6055             LET SENSOR.FDC = 0
6056          ELSE
6057             READ DUMMY
6058             FOR K = 1 TO N.FDC,
6059             WITH FD.UNIT(K) = DUMMY
6060             FIND THE FIRST CASE
6061             IF NONE
6062                SKIP 1 LINE
6063                PRINT 2 LINES WITH DUMMY THUS
6064                UNABLE TO FIND A FDC WITH UNIT # **** WHICH WAS
6065                REFERENCED IN THE SENSOR FILE
6066                TRACE
6067 <--------------STOP
6068             OTHERWISE
```

```
6069
6070            LET SENSOR.FDC = K
6071         ALWAYS
6072         LOOP FOR L = 1 TO NUM.SENSOR
6073         DO
6074            CREATE A US.LINK CALLED USL
6075            FILE USL IN UN.SENSOR.LIST( SENSOR.UNIT )
6076            LET US.SENSOR.TYPE( USL ) = SENS.TYPE
6077            LET US.MODEL(USL) = SENSOR.MODEL
6078            LET US.UNIT(USL) = SENSOR.UNIT
6079            LET US.FDC(USL) = SENSOR.FDC
6080            LET SENSOR.TYPE = SENS.TYPE
6081
6082            IF ST.NAME(SENSOR.TYPE) = "AO"
6083               FILE USL IN AVAIL.AO.LIST(ASIDE)
6084               LET US.EQ.ID(USL) = MAO.EQ.ID(SENSOR.MODEL)
6085            ALWAYS
6086
6087            IF ST.NAME(SENSOR.TYPE) = "AD" AND L = 1
6088               CREATE AN AD.SENSOR
6089               FILE AD.SENSOR IN
6090               THE SD.ADS.SET(UN.COLOR(SENSOR.UNIT))
6091               LET ADS.UNIT.PTR(AD.SENSOR) = SENSOR.UNIT
6092               LET ADS.MADS.PTR(AD.SENSOR) = SENSOR.MODEL
6093               LET ADS.NR.SENSORS(AD.SENSOR) = NUM.SENSOR
6094            ALWAYS
6095
6096            IF ST.NAME( SENSOR.TYPE ) = "FO"
6097               LET US.EQ.ID(USL) = MFO.EQ.ID(SENSOR.MODEL)
6098               IF L = 1
6099                  LET RADIUS = UN.RADIUS( SENSOR.UNIT )
6100                  LET DISTANCE = ((NUM.SENSOR-1)/NUM.SENSOR)*RADIUS
6101                  LET INTERVAL = 2*RADIUS/NUM.SENSOR
6102               ALWAYS
6103
6104               ACTIVATE_A FORWARD.OBSERVER CALLED FO NOW------------------------------------------------------------------------------>(438)
6105
6106               READ FO.FST.INDIC(FO)
6107               IF FO.FST.INDIC(FO)=TRUE AND FD.FDC(SENSOR.FDC)=0
6108                  CALL ERROR.STOP-----------------------------------------------------------------------------------
```

```
6127            IF ST.NAME(SENSOR.TYPE) = "SD" OR
6128               ST.NAME(SENSOR.TYPE) = "FL"
6129               LET US.EQ.ID(USL) = MPDB.EQ.ID(SENSOR.MODEL)
6130               CREATE A PASSIVE.DETECTION.BASE CALLED PDB
6131               LET PDB.US.LINK(PDB) = USL
6132               LET US.ID(USL) = PDB
6133               FILE PDB IN SIDE.PDB.SET(ASIDE)
6134               READ NUM.KEYED.SENSORS
6135               LOOP FOR J = 1 TO NUM.KEYED.SENSORS
6136               DO
6137                  CREATE A KEYED.SENSOR CALLED KS
6138                  FILE KS IN PDB.KEYED.LIST(PDB)
6139                  READ KS.TYPE.SENSOR(KS),
6140                  SENS.UN.NO.
6141                  FOR K = 1 TO N.UNIT,
6142                  WITH SENS.UN.NO. = UNIT.NOS(K),
6143                  FIND THE FIRST CASE
6144                  IF NONE,
6145                     SKIP 1 LINE
6146                     PRINT 1 LINE WITH SENS.UN.NO. THUS
6147                     = = = SENSOR INPUT UNIT **** NOT IN UNIT DATA = = =
6148                     LET ERROR.MSG = 1
6149                     LET SENS.UN = SENS.UN.NO.
6150                  ELSE
6151                     LET SENS.UN = K
6152                  ALWAYS
6153                  LET KS.SENSOR.ID(KS) = SENS.UN
6154               ENDLOOP  '' THRU ALL THE KEYED SENSORS IN THE UNIT
6155            ALWAYS
6156         ENDLOOP  '' THRU ALL OF THE SENSORS OF THIS TYPE
6157      ENDLOOP  '' THRU ALL OF THE SENSORS IN THE UNIT
6158   ENDLOOP  '' THRU ALL OF THE UNITS ON THE LIST
6159
6160   ''THIS CODE ALLOWS FOR ONLY ONE KEYED SENSOR OF EACH TYPE IN A UNIT
6161   LOOP FOR EACH SIDE
6162   FOR EACH PDB IN SIDE.PDB.SET(SIDE)
6163   FOR EACH KS IN PDB.KEYED.LIST(PDB)
6164   DO
6165      FOR EACH USL IN UN.SENSOR.LIST(KS.SENSOR.ID(KS))
6166      WHEN US.SENSOR.TYPE(USL) = KS.TYPE.SENSOR(KS)
6167      FIND THE FIRST CASE
6168      IF NONE
6169         LIST KS, USL, US.SENSOR.TYPE(USL), KS.TYPE.SENSOR(KS)
6170         LET ERROR.MSG = 1
6171      ELSE
6172         LET KS.SENSOR.ID(KS) = US.ID(USL)
6173      ALWAYS
6174   ENDLOOP
6175
6176   LOOP FOR EACH SIDE
6177   DO
6178      READ NUM.AO.SORTIE.PER.SECTOR.PER.DAY
6179      IF NUM.AO.SORTIE.PER.SECTOR.PER.DAY = 0
6180   <------CYCLE
6181      OTHERWISE
6182      LET INTERVAL = 24.*60./NUM.AO.SORTIE.PER.SECTOR.PER.DAY
6183      LOOP FOR EACH SECTOR CALLED SECT
6184      DO
```

```
6185          SCHEDULE_A FEBA.SORTIE------------------------------------------------------------------------------------------>(371)
6186          GIVEN
6187          SECT,
6188          SIDE,
6189          INTERVAL
6190          IN UNIFORM.F(0.,2.*INTERVAL, 1) MINUTES
6191       ENDLOOP
6192    ENDLOOP
6193
6194    IF ERROR.MSG = 1
6195       CALL ERROR.STOP------------------------------------------------------------------------------------------------->(604)
6196    ALWAYS
6197
6198    RELEASE FD.UNIT(*)
6199
6200 <--EXITROUTINE
6201 ENDROUTINE
```

```
6202 ROUTINE TBF.INPUT                                                    ''                    1029
6203
6204    ADD 1 TO ANAL.CTR(213,1)  ''                                                          \DYN_ANAL
6205    NORMALLY MODE IS INTEGER
6206    DEFINE TYPBF TO MEAN TYPE.BATTLE.FIELD
6207
6208    READ HQ.SET.BACK,
6209    DIS.WITH.DRAW,
6210    DIS.ATTACK,
6211    R.WIDTH.UNIT,
6212    B.WIDTH.UNIT
6213    LET HQ.SET.BACK = HQ.SET.BACK/16
6214    LET DIS.WITH.DRAW = DIS.WITH.DRAW/16
6215    LET DIS.ATTACK = DIS.ATTACK/16
6216    LET R.WIDTH.UNIT = R.WIDTH.UNIT/16
6217    LET B.WIDTH.UNIT = B.WIDTH.UNIT/16
6218
6219    READ N.TYPE.BATTLE.FIELD
6220    CREATE EVERY TYPE.BATTLE.FIELD
6221
6222    LOOP FOR EACH TYPE.BATTLE.FIELD
6223    DO
6224       READ TBF.NO.BL.UNITS(TYPBF),
6225       TBF.BL.MISSION(TYPBF),
6226       TBF.BL.ARMOR.UNITS(TYPBF),
6227       TBF.BL.MECH.UNITS(TYPBF),
6228       TBF.BL.INF.UNITS(TYPBF),
6229       TBF.BL.HQ.UNITS(TYPBF)
6230       READ TBF.NO.RD.UNITS(TYPBF),
6231       TBF.RD.MISSION(TYPBF),
6232       TBF.RD.ARMOR.UNITS(TYPBF),
6233       TBF.RD.MECH.UNITS(TYPBF),
6234       TBF.RD.INF.UNITS(TYPBF),
6235       TBF.RD.HQ.UNITS(TYPBF)
6236
6237       READ TBF.WIDTH(TYPBF)   ''IN METERS
6238       LET TBF.WIDTH=TBF.WIDTH/16     ''CONVERT TO HEXADECAMETERS
6239
6240       FOR I=1 TO TBF.BL.ARMOR.UNITS(TYPBF)
6241       CALL CREATE.TEAMS(BLUE,BLUE.TK.PLT)------------------------------------------------------>( 60)
6242       FOR I=1 TO TBF.BL.MECH.UNITS(TYPBF)
6243       CALL CREATE.TEAMS(BLUE,BLUE.MECH.PLT)----------------------------------------------------->( 60)
6244       FOR I=1 TO TBF.BL.INF.UNITS(TYPBF)
6245       CALL CREATE.TEAMS(BLUE,BLUE.INF.PLT)------------------------------------------------------>( 60)
6246       FOR I=1 TO TBF.BL.HQ.UNITS(TYPBF)
6247       CALL CREATE.TEAMS(BLUE,BLUE.CO.HQ)-------------------------------------------------------->( 60)
6248       FOR I=1 TO TBF.RD.ARMOR.UNITS(TYPBF)
6249       CALL CREATE.TEAMS(RED,RED.TK.PLT)--------------------------------------------------------->( 60)
6250       FOR I=1 TO TBF.RD.MECH.UNITS(TYPBF)
6251       CALL CREATE.TEAMS(RED,RED.MECH.PLT)------------------------------------------------------->( 60)
6252       FOR I=1 TO TBF.RD.INF.UNITS(TYPBF)
6253       CALL CREATE.TEAMS(RED,RED.INF.PLT)-------------------------------------------------------->( 60)
6254       FOR I=1 TO TBF.RD.HQ.UNITS(TYPBF)
6255       CALL CREATE.TEAMS(RED,RED.CO.HQ)---------------------------------------------------------->( 60)
6256    ENDLOOP
6257
6258 <--EXITROUTINE
6259 ENDROUTINE
```

```
6260 ROUTINE DECISION.INPUT                                    ''
6261
6262     ADD 1 TO ANAL.CTR(214,1)  ''
6263     NORMALLY MODE IS INTEGER
6264
6265     READ N.COLOR
6266     READ N.MISSION
6267     CREATE EVERY COLOR
6268     CREATE EVERY MISSION
6269
6270     LOOP FOR EVERY COLOR
6271     DO THE FOLLOWING
6272        LOOP FOR EVERY MISSION
6273        DO THE FOLLOWING
6274           READ DECISION(MISSION, COLOR),
6275           SUP.MISSION.PRIORITY(MISSION, COLOR),
6276           WD.DIST(MISSION, COLOR)
6277        ENDLOOP
6278     ENDLOOP
6279
6280 <--EXITROUTINE
6281 ENDROUTINE
```

```
6282 ROUTINE TT.FACTORS.INPUT                                                    ''                I031
6283
6284    ADD 1 TO ANAL.CTR(215,1)  ''                                                               \DYN_ANAL
6285    NORMALLY MODE IS INTEGER
6286
6287    READ TER.W.INC
6288    READ N.TERRAIN.TYPE
6289    READ N.LOS.BAND
6290    CREATE EVERY LOS.BAND AND TERRAIN.TYPE
6291
6292    LOOP FOR EVERY LOS.BAND
6293    DO THE FOLLOWING
6294       READ BR
6295       LET BAND.RANGE(LOS.BAND) = BR/16
6296    ENDLOOP
6297
6298    LOOP FOR EVERY TERRAIN.TYPE,
6299    FOR EVERY LOS.BAND,
6300    DO THE FOLLOWING
6301       READ PROB
6302       ''PROB STORED AS [0, 50]
6303       LET LOS.PROB(TERRAIN.TYPE,LOS.BAND) = PROB/2
6304    ENDLOOP
6305
6306    LOOP FOR EVERY TERRAIN.TYPE
6307    DO THE FOLLOWING
6308       READ TT.LOS.SHAPE(TERRAIN.TYPE),
6309       TT.NLOS.SHAPE(TERRAIN.TYPE),
6310       TT.LOS.SCALE(TERRAIN.TYPE),
6311       TT.NLOS.SCALE(TERRAIN.TYPE)
6312       READ X
6313       LET TT.STATIONARY.LOS.BREAK(TERRAIN.TYPE) = X/16
6314       READ X
6315       LET TT.MOVING.LOS.BREAK(TERRAIN.TYPE) = X/16
6316       READ X
6317       LET TT.M.S.LOS.BREAK(TERRAIN.TYPE) = X/16
6318       READ X
6319       LET MOVE.FIRE.DIST(TERRAIN.TYPE) = X/16
6320       READ X
6321       LET DEFILADE.DIST(TERRAIN.TYPE) = X/16
6322       READ MOV.FAC(TERRAIN.TYPE)
6323    ENDLOOP
6324
6325 <--EXITROUTINE
6326 ENDROUTINE
```

```
6327 ROUTINE FARRP.INPUT                                                    ''          I032
6328
6329    ADD 1 TO ANAL.CTR(216,1)  ''                                                    \DYN_ANAL
6330    NORMALLY MODE IS INTEGER
6331
6332    ''CODE FOR DIFFERENT HELICOPTER CONFIGURATIONS
6333    READ HC.SWITCH
6334    IF HC.SWITCH = 0  ''                                        CHG\17  SHUT OFF WITH DATA
6335 <------EXITROUTINE
6336    OTHERWISE
6337
6338    ''READ THE NUMBER AND NAMES OF SCOUT CONFIGURATIONS
6339    READ NO.SCOUT.CONFIGURATIONS
6340    RESERVE SCT.HC AS NO.SCOUT.CONFIGURATIONS
6341    LOOP FOR I = 1 TO NO.SCOUT.CONFIGURATIONS
6342    DO
6343       READ SCT.HC(I)
6344    ENDLOOP
6345
6346    ''READ THE NUMBER AND NAMES OF ATTACK CONFIGURATIONS
6347    READ NO.ATTACK.CONFIGURATIONS
6348    RESERVE ATK.HC AS NO.ATTACK.CONFIGURATIONS
6349    LOOP FOR I = 1 TO NO.ATTACK.CONFIGURATIONS
6350    DO
6351       READ ATK.HC(I)
6352    ENDLOOP
6353
6354    READ PRINT.SWITCH, HC.DEBUG
6355    READ BL.MAX.FL.TIME, RD.MAX.FL.TIME
6356    READ BL.HC.SPACING, RD.HC.SPACING
6357    READ MIN.NO.SUP.UNITS
6358    READ BL.MIN.HANDOFF.TIME, BL.MAX.HANDOFF.TIME,
6359    RD.MIN.HANDOFF.TIME, RD.MAX.HANDOFF.TIME
6360    READ BL.MIN.MASK.TIME, BL.MAX.MASK.TIME,
6361    RD.MIN.MASK.TIME, RD.MAX.MASK.TIME
6362    READ BL.MIN.UNMASK.TIME, BL.MAX.UNMASK.TIME,
6363    RD.MIN.UNMASK.TIME, RD.MAX.UNMASK.TIME
6364    READ BL.LOW.FRAC.RANGE, BL.HIGH.FRAC.RANGE,
6365    RD.LOW.FRAC.RANGE, RD.HIGH.FRAC.RANGE
6366    READ BL.ATK.FAIL.PROB, BL.SCT.FAIL.PROB,
6367    RD.ATK.FAIL.PROB, RD.SCT.FAIL.PROB
6368    READ BL.ROUNDS.PER.POPUP, RD.ROUNDS.PER.POPUP
6369    READ NO.OF.FARRPS
6370
6371    IF PRINT.SWITCH = 1
6372       START NEW PAGE
6373       PRINT 1 LINE THUS
6374       ----FARRP.INPUT
6375
6376       LIST SCT.HC, ATK.HC
6377       SKIP 1 LINE
6378
6379       PRINT 20 LINES WITH
6380       BL.MAX.FL.TIME,       RD.MAX.FL.TIME,
6381       BL.HC.SPACING,        RD.HC.SPACING,
6382       MIN.NO.SUP.UNITS,
6383       BL.MIN.HANDOFF.TIME, RD.MIN.HANDOFF.TIME,
6384       BL.MAX.HANDOFF.TIME, RD.MAX.HANDOFF.TIME,
```

```
6385        BL.MIN.MASK.TIME,     RD.MIN.MASK.TIME,
6386        BL.MAX.MASK.TIME,     RD.MAX.MASK.TIME,
6387        BL.MIN.UNMASK.TIME,   RD.MIN.UNMASK.TIME,
6388        BL.MAX.UNMASK.TIME,   RD.MAX.UNMASK.TIME,
6389        BL.LOW.FRAC.RANGE,    RD.LOW.FRAC.RANGE,
6390        BL.HIGH.FRAC.RANGE,   RD.HIGH.FRAC.RANGE,
6391        BL.ATK.FAIL.PROB,     RD.ATK.FAIL.PROB,
6392        BL.SCT.FAIL.PROB,     RD.SCT.FAIL.PROB,
6393        BL.ROUNDS.PER.POPUP,  RD.ROUNDS.PER.POPUP,
6394        NO.OF.FARRPS    THUS
6395        BLUE          RED
6396        MAX FL TIME           ******       ******
6397        SPACING               ******       ******
6398        MIN NO SUPP UNITS     ******
6399        MIN HANDOFF TIME      ******.**    ******.**
6400        MAX HANDOFF TIME      ******.**    ******.**
6401        MIN MASK    TIME      ******.**    ******.**
6402        MAX MASK    TIME      ******.**    ******.**
6403        MIN UNMASK  TIME      ******.**    ******.**
6404        MAX UNMASK  TIME      ******.**    ******.**
6405        LOW  FRAC RANGE       ******.**    ******.**
6406        HIGH FRAC RANGE       ******.**    ******.**
6407        ATK  FAIL PROB        ******.**    ******.**
6408        SCT  FAIL PROB        ******.**    ******.**
6409        ROUNDS PER POPUP      ******       ******
6410
6411        NO. OF FARRPS =  ****
6412
6413        FARRP     FARRP  UNIT          REFUEL   REARM     REFUEL       REARM
6414        POINTER    UNIT   NOS    SIDE    TIME     TIME    CAPABILITY   CAPABILITY
6415     ALWAYS
6416
6417     LOOP FOR I = 1 TO NO.OF.FARRPS
6418     DO THE FOLLOWING
6419        CREATE A FARRP
6420        READ .UNIT
6421        FOR EVERY UNIT WITH UNIT.NOS(UNIT) = .UNIT
6422        FIND THE FIRST CASE
6423        IF NONE
6424           TRACE
6425 <----------STOP
6426        OTHERWISE
6427        LET FP.UNIT(FARRP) = UNIT
6428        READ REFUEL.TIME(FARRP), REARM.TIME(FARRP),
6429        REFUEL.CAP(FARRP), REARM.CAP(FARRP)
6430        FILE THE FARRP IN THE FP.SET(UN.COLOR(FP.UNIT(FARRP)))
6431
6432        IF PRINT.SWITCH = 1
6433           PRINT 1 LINE WITH I, FARRP, FP.UNIT(FARRP),
6434           UNIT.NOS(FP.UNIT(FARRP)),
6435           UN.COLOR(FP.UNIT(FARRP)), REFUEL.TIME(FARRP),
6436           REARM.TIME(FARRP), REFUEL.CAP(FARRP), REARM.CAP(FARRP)
6437           AS FOLLOWS
6438           ***   ******   ****   ****   **     ****    ****       ****         ****
6439        ALWAYS
6440
6441     ENDLOOP
6442
```

```
6443     ''CONVERT SPACING TO HEXADECAMETERS
6444     LET BL.HC.SPACING = BL.HC.SPACING / 16
6445     LET RD.HC.SPACING = RD.HC.SPACING / 16
6446
6447 <--EXITROUTINE
6448 ENDROUTINE
```

```
6449 ROUTINE PGM.INPUT                                                    ''          I033
6450
6451    ADD 1 TO ANAL.CTR(217,1)  ''                                                  \DYN_ANAL
6452    NORMALLY MODE IS INTEGER
6453    DEFINE NAME AS A TEXT VARIABLE ''                                             \TEXT
6454
6455    RESERVE PGM.HIT(*,*) AS 12 BY 4
6456
6457    LOOP FOR I = 1 TO 12,
6458    DO
6459       LOOP FOR J = 1 TO 4,
6460       DO
6461          READ PGM.HIT(I,J) ''READ PR(HIT/LASE)
6462       ENDLOOP
6463    ENDLOOP
6464
6465    READ N.PGM.TGTS
6466
6467    RESERVE PGM.LINK(*) AS N.PGM.TGTS
6468    RESERVE PGM.KILL(*,*) AS N.PGM.TGTS BY 4
6469
6470    LOOP FOR I = 1 TO N.PGM.TGTS,
6471    DO
6472       READ NAME
6473       FOR EVERY EQUIPMENT
6474       WITH EQ.NAME(EQUIPMENT) = NAME
6475       FIND THE FIRST CASE
6476       IF NONE
6477          TRACE
6478 <--------STOP
6479       OTHERWISE
6480       LET PGM.LINK(I) = EQUIPMENT
6481       LOOP FOR J = 1 TO 4,
6482       DO
6483          READ PGM.KILL(I,J)  ''READ PR(KILL/HIT)
6484       ENDLOOP
6485    ENDLOOP
6486
6487    READ GOOD.WEATHER.PROB,PGM.RELY
6488    READ SADARM.THRESHOLD, SDM.CB, SDM.MNV, SDM.SSPK, SDM.MAX.RANGE,
6489    SDM.TM, SDM.VOLLEY.RADIUS, SDM.PGM.RNG
6490    READ PGM.MARK, SDM.MARK
6491
6492 <--EXITROUTINE
6493 ENDROUTINE
```

```
6494 ROUTINE ILLUM.INPUT                                                    ''                                                    1034
6495
6496    ADD 1 TO ANAL.CTR(218,1)  ''                                                                                          \DYN_ANAL
6497    NORMALLY MODE IS INTEGER
6498
6499    READ ILLUM.SWITCH
6500    IF ILLUM.SWITCH = 0  ''                                                                                 CHG\21 SHUT_OFF_WITH_DATA
6501 <-------EXITROUTINE
6502    OTHERWISE
6503
6504    READ ILLUM.DEBUG,
6505    N.ILLUM.MUNITION
6506
6507    CREATE EACH ILLUM.MUNITION
6508    LOOP
6509    FOR EACH ILLUM.MUNITION CALLED .IM
6510    DO
6511       READ ILLUM.ID(.IM),
6512       ILLUM.RADIUS(.IM),  ''METERS''
6513       .RANGE,             ''DECAMETERS''
6514       ILLUM.DURATION(.IM),
6515       ILLUM.RND.WT(.IM)
6516       LET ILLUM.MAX.RANGE(.IM) = 10 * .RANGE / 16. ''HDM                                                                 \OPTIMIZE
6517    ENDLOOP
6518
6519    LOOP
6520    FOR EACH SIDE
6521    DO
6522       SKIP 1 FIELD
6523       LOOP
6524       FOR EACH MISSION
6525       DO
6526          SKIP 1 FIELD
6527          READ ILLUM.RULE(SIDE, MISSION)
6528       ENDLOOP
6529    ENDLOOP
6530
6531 <--EXITROUTINE
6532 ENDROUTINE
```

```
6533 ROUTINE MINE.INPUT                                                    ''                                                    I035
6534
6535    ADD 1 TO ANAL.CTR(219,1)  ''                                                                                          \DYN_ANAL
6536    ''THIS ROUTINE WILL READ DATA FOR FASCAM MUNITIONS, RULES
6537    ''OF USE, DELAYS, AND BARRIER MINEFIELDS
6538
6539    NORMALLY MODE IS INTEGER
6540    DEFINE .ALPHA, MF.TYPE AS TEXT VARIABLES ''                                                                            \TEXT
6541    DEFINE .XMINEFIELD AS A 1-DIMENSIONAL INTEGER ARRAY
6542    DEFINE .YMINEFIELD AS A 1-DIMENSIONAL INTEGER ARRAY
6543    RESERVE .XMINEFIELD(*) AS 2
6544    RESERVE .YMINEFIELD(*) AS 2
6545
6546    READ MF.SWITCH
6547    IF MF.SWITCH NE 1  ''                                                                  CHG\23  SHUT OFF WITH DATA
6548 <------EXITROUTINE
6549    OTHERWISE
6550
6551    READ MF.DEBUG,
6552    MF.PRINT,
6553    MAX.WD.FASCAM,
6554    MAX.ATT.FASCAM,
6555    MIN.FASCAM.RANGE,    ''DECAMETERS
6556    MAX.FASCAM.RANGE,    ''DECAMETERS
6557    FASCAM.VOLLEYS
6558    LET MIN.FASCAM.RANGE = 10 * MIN.FASCAM.RANGE / 16.  ''HDM                                                             \OPTIMIZE
6559    LET MAX.FASCAM.RANGE = 10 * MAX.FASCAM.RANGE / 16.  ''HDM
6560
6561    READ N.FASCAM.MUNITION
6562    IF N.FASCAM.MUNITION GT 0
6563       CREATE EACH FASCAM.MUNITION
6564       LOOP
6565       FOR EACH FASCAM.MUNITION CALLED .FM
6566       DO
6567          READ FMM.ID(.FM),
6568          .RANGE,          ''DECAMETERS
6569          FMM.RND.WT
6570          LET FMM.MAX.RANGE(.FM) = 10 * .RANGE / 16.  ''HDM
6571       ENDLOOP
6572    ALWAYS
6573
6574    CREATE EACH MF.BAND(3)
6575    LOOP
6576    FOR EACH SIDE
6577    DO
6578       SKIP 1 FIELD
6579       READ MFB.DELAY(SIDE, 1),
6580       .UPPER1,                ''DECAMETERS
6581       MFB.DELAY(SIDE, 2),
6582       .UPPER2,                ''DECAMETERS
6583       MFB.DELAY(SIDE, 3)
6584       LET MFB.UPPER.LIMIT(SIDE, 1) = 10 * .UPPER1 / 16. ''HDM
6585       LET MFB.UPPER.LIMIT(SIDE, 2) = 10 * .UPPER2 / 16. ''HDM
6586
6587       LOOP
6588       FOR EACH MISSION
6589       DO
6590          SKIP 1 FIELD
```

```
6591          READ MINE.USE.RULE(SIDE, MISSION)
6592          IF MISSION GE 4
6593             READ MINE.WD.RULE(SIDE, MISSION)
6594          ALWAYS
6595       ENDLOOP
6596    ENDLOOP
6597
6598    READ N.MINEFIELD
6599    IF N.MINEFIELD GT 0
6600       CREATE EACH MINEFIELD
6601       WRITE N.MINEFIELD AS S 1,I 3, /  USING UNIT 47
6602       LOOP
6603       FOR EACH MINEFIELD
6604       DO
6605          READ MF.ID(MINEFIELD),
6606          .ALPHA,
6607          MF.TYPE
6608          IF .ALPHA = "RED"
6609             LET MF.COLOR(MINEFIELD) = RED
6610          ELSE
6611             LET MF.COLOR(MINEFIELD) = BLUE
6612          ALWAYS
6613
6614          LET .XLOW = INF.C
6615          LET .XHIGH = 0
6616          LET .YLOW = INF.C
6617          LET .YHIGH = 0
6618          LOOP
6619          FOR .I = 1 TO 2
6620          DO
6621             CREATE A MF.POINT
6622             FILE THIS MF.POINT IN MFP.LIST(MINEFIELD)
6623             READ .XMINEFIELD(.I),
6624             .YMINEFIELD(.I)                ''DM
6625             LET MFP.X.COORD(MF.POINT)= 10. *
6626             .XMINEFIELD(.I) / 16.   ''HDM
6627             LET MFP.Y.COORD(MF.POINT) = 10. *
6628             .YMINEFIELD(.I) / 16.   ''HDM
6629             LET .XLOW = MIN.F(.XLOW, MFP.X.COORD(MF.POINT))
6630             LET .XHIGH = MAX.F(.XHIGH, MFP.X.COORD(MF.POINT))
6631             LET .YLOW = MIN.F(.YLOW, MFP.Y.COORD(MF.POINT))
6632             LET .YHIGH = MAX.F(.YHIGH, MFP.Y.COORD(MF.POINT))
6633          ENDLOOP
6634          WRITE MF.COLOR(MINEFIELD), MF.TYPE, .XMINEFIELD(1),
6635          .YMINEFIELD(1), .XMINEFIELD(2), .YMINEFIELD(2)
6636          AS S 12, I 1, S 3, T 1, S 4, I 4, S 2, I 4, S 4, ''                \TEXT
6637          I 4, S 2, I 4, /  USING UNIT 47
6638          LET MF.X.LOW(MINEFIELD) = .XLOW
6639          LET MF.X.HIGH(MINEFIELD) = .XHIGH
6640          LET MF.Y.LOW(MINEFIELD) = .YLOW
6641          LET MF.Y.HIGH(MINEFIELD) = .YHIGH
6642
6643          IF MF.PRINT = TRUE
6644             PRINT 1 LINE WITH
6645             MF.ID(MINEFIELD),
6646             .ALPHA,
6647             .XLOW,
6648             .XHIGH,
```

```
6649              .YLOW,
6650              .YHIGH THUS
6651              ===MINE INPUT-******,SIDE=****,XLOW=*****,XHIGH=*****,YLOW=*****,YHIGH=*****
6652              LOOP
6653              FOR EVERY .POINT IN MFP.LIST(MINEFIELD)
6654              DO
6655                 PRINT 1 LINE WITH
6656                 MFP.X.COORD(.POINT),
6657                 MFP.Y.COORD(.POINT) THUS
6658                 = = =      POINTS (*****,*****)
6659              ENDLOOP
6660           ALWAYS
6661
6662        ENDLOOP
6663     ALWAYS
6664
6665
6666 <--EXITROUTINE
6667 ENDROUTINE
```

```
6668 ROUTINE SMOKE.INPUT                                                    ''                                  I036
6669
6670    ADD 1 TO ANAL.CTR(220,1)  ''                                                                            \DYN_ANAL
6671    NORMALLY MODE IS INTEGER
6672
6673    READ SMK.SWITCH
6674    IF SMK.SWITCH = 0  ''                                                                  CHG\22 SHUT OFF WITH DATA
6675 <------EXITROUTINE
6676    OTHERWISE
6677
6678    READ SMK.DEBUG,
6679    N.SMOKE.MUNITION
6680    CREATE EACH SMOKE.MUNITION
6681    LOOP
6682    FOR EVERY SMOKE.MUNITION CALLED .SM
6683    DO
6684       READ SMK.ID(.SM),
6685       SMK.WIDTH(.SM),      ''METERS
6686       .RANGE,              ''DECAMETERS
6687       SMK.BURN.TIME(.SM),  ''MINUTES
6688       SMK.RND.WT(.SM)
6689       LET SMK.MAX.RANGE(.SM) = 10 * .RANGE / 16.   ''HDM                                                   \OPTIMIZE
6690    ENDLOOP
6691
6692    LOOP
6693    FOR EACH SIDE
6694    DO
6695       SKIP 1 FIELD
6696       LOOP
6697       FOR EACH MISSION
6698       DO
6699          SKIP 1 FIELD
6700          READ SMK.USE.RULE(SIDE, DAY, MISSION),
6701          SMK.USE.RULE(SIDE, NITE, MISSION)
6702          IF MISSION GE 4
6703             READ SMK.WD.RULE(SIDE, DAY, MISSION),
6704             SMK.WD.RULE(SIDE, NITE, MISSION)
6705          ALWAYS
6706       ENDLOOP
6707    ENDLOOP
6708
6709 <--EXITROUTINE
6710 ENDROUTINE
```

```
6711 ROUTINE VIS.INPUT                                                                                    ''  I037
6712
6713    ADD 1 TO ANAL.CTR(221,1)  ''                                                                          \DYN_ANAL
6714    NORMALLY MODE IS INTEGER
6715    DEFINE TIME AS A REAL VARIABLE
6716
6717    READ NITE.VIS.PCT
6718    LOOP UNTIL MODE IS ALPHA ''                                                                           \ALPHA
6719    DO
6720       READ .VIS,
6721       TIME
6722       LET .VIS = .VIS / 16   ''CONVERT TO HDM
6723       IF TIME = 0.0
6724          LET VISIBILITY = .VIS
6725       ELSE
6726          SCHEDULE_A CHANGE.WEATHER----------------------------------------------------------------------------------------------->(366)
6727          GIVEN
6728          .VIS
6729          IN TIME HOURS
6730       ALWAYS
6731    ENDLOOP
6732    SKIP 1 FIELD
6733
6734 <--EXITROUTINE
6735 ENDROUTINE
```

```
6736 ROUTINE TACAIR.INPUT                    ''                                          1038
6737     YIELDING
6738     .TACAIR.FLAG
6739
6740     ADD 1 TO ANAL.CTR(222,1)  ''                                                    \DYN_ANAL
6741     NORMALLY MODE IS INTEGER
6742     DEFINE .ATK.TIME AS A REAL VARIABLE
6743     DEFINE .NAME AS A TEXT VARIABLE ''                                              \TEXT
6744
6745     READ .TACAIR.FLAG
6746     IF .TACAIR.FLAG = 0 ''                                          CHG\17 SHUT OFF WITH DATA
6747 <------RETURN
6748     OTHERWISE
6749
6750     READ CAS.MSN.RPT.FLAG,
6751     TACAIR.DEBUG
6752     LOOP FOR EACH SIDE CALLED .S
6753     DO
6754        SKIP 1 FIELD
6755        READ .AIRFIELD
6756        FOR EVERY UNIT
6757        WITH UNIT.NOS(UNIT) = .AIRFIELD
6758        FIND THE FIRST CASE
6759        IF NONE
6760           TRACE
6761 <----------STOP
6762        OTHERWISE
6763
6764        LET SD.AIRFIELD(.S) = UNIT
6765        READ SD.MAX.SORTIE.TP(.S),
6766        SD.TP.SORTIE(.S),
6767        SD.ASC.MAX.SORTIE(.S),
6768        .RADIUS,                          ''DKM
6769        .NO.FLY,                              ''DKM
6770        .POOR.FLY,                                ''DKM
6771        SD.CAS.BRKPT(.S)
6772        LET SD.ASC.RADIUS(.S) = 10 * .RADIUS / 16. ''                               \OPTIMIZE
6773        LET SD.NO.FLY.VIS(.S) = 10 * .NO.FLY / 16.
6774        LET SD.POOR.FLY.VIS(.S) = 10 * .POOR.FLY / 16.
6775        LOOP FOR EVERY MISSION CALLED .M
6776        DO
6777           READ .NAME
6778           FOR EVERY TYPE.EQUIPMENT
6779           WITH TE.NAME(TYPE.EQUIPMENT) = .NAME
6780           FIND THE FIRST CASE
6781           IF NONE
6782              TRACE
6783 <-------------STOP
6784           OTHERWISE
6785           LET SM.TANK.TE(.S,.M) = TYPE.EQUIPMENT
6786           READ SM.MIN.TANK.RATIO(.S,.M),
6787           SM.MAX.TANK.RATIO(.S,.M),
6788           SM.MIN.CEQ(.S,.M)
6789        ENDLOOP
6790     ENDLOOP
6791
6792     READ N.AC.TYPE
6793     CREATE EVERY AC.TYPE
```

```
6794     LOOP FOR EVERY AC.TYPE CALLED .AC
6795     DO
6796        SKIP 1 FIELD
6797        READ .NAME
6798        FOR EVERY EQUIPMENT
6799        WITH EQ.NAME(EQUIPMENT) = .NAME
6800        FIND THE FIRST CASE
6801        IF NONE
6802           TRACE
6803 <----------STOP
6804        OTHERWISE
6805
6806        LET ACT.EQUIP.ID(.AC) = EQUIPMENT
6807        READ .NAME
6808        IF .NAME = "NONE"
6809           LET ACT.SUBSTITUTE(.AC) = 0
6810        ELSE
6811           FOR EVERY EQUIPMENT
6812           WITH EQ.NAME(EQUIPMENT) = .NAME
6813           FIND THE FIRST CASE
6814           IF NONE
6815              TRACE
6816 <-------------STOP
6817           OTHERWISE
6818           LET ACT.SUBSTITUTE(.AC) = EQUIPMENT
6819        ALWAYS
6820        READ ACT.WEATHER.DEGRADE(.AC),
6821        .MIN.ALT,
6822        .NORM.ALT,
6823        ACT.BAI.TA.DELAY(.AC),
6824        ACT.MAX.ALOFT(.AC),
6825        ACT.PROB.SORTIE.ABORT(.AC),
6826        ACT.MIN.PREP.TIME(.AC),
6827        ACT.MAX.PREP.TIME(.AC),
6828        ACT.NITE.FLY(.AC)
6829        LET ACT.MIN.ALT(.AC) = 10 * .MIN.ALT / 16.
6830        LET ACT.NORM.ALT(.AC) = 10 * .NORM.ALT / 16.
6831        READ .X,
6832        .Y,
6833        .Z
6834        LET ACT.X1(.AC) = 10 * .X / 16.
6835        LET ACT.Y1(.AC) = 10 * .Y / 16.
6836        LET ACT.Z1(.AC) = 10 * .Z / 16.
6837        READ .X,
6838        .Y,
6839        .Z
6840        LET ACT.X2(.AC) = 10 * .X / 16.
6841        LET ACT.Y2(.AC) = 10 * .Y / 16.
6842        LET ACT.Z2(.AC) = 10 * .Z / 16.
6843        READ .X,
6844        .Y,
6845        .Z
6846        LET ACT.X3(.AC) = 10 * .X / 16.
6847        LET ACT.Y3(.AC) = 10 * .Y / 16.
6848        LET ACT.Z3(.AC) = 10 * .Z / 16.
6849
6850        ''SET THE SPEED.  CONVERT FROM KM/HR TO HDM/SEC
6851        LET ACT.SPEED(.AC)  =  EQ.MAX.SPEED(ACT.EQUIP.ID(.AC)) * 7
```

```
6852        * (1000 / 16.) / 3600.
6853
6854        ''SET THE DISTANCE FROM P1, P2 AND P3 TO THE TARGET
6855        LET ACT.P1.DIST(.AC) = SQRT.F(REAL.F((ACT.X1(.AC) ** 2
6856        + ACT.Y1(.AC) ** 2 + ACT.Z1(.AC) ** 2)))
6857        LET ACT.P2.DIST(.AC) = SQRT.F(REAL.F((ACT.X2(.AC) ** 2
6858        + ACT.Y2(.AC) ** 2 + ACT.Z2(.AC) ** 2)))
6859        LET ACT.P3.DIST(.AC) = SQRT.F(REAL.F((ACT.X3(.AC) ** 2
6860        + ACT.Y3(.AC) ** 2 + ACT.Z3(.AC) ** 2)))
6861
6862        ''SET THE RELATIONSHIPS BETWEEN P1 AND P2
6863        LET .DELTAX =  ACT.X2(.AC) - ACT.X1(.AC)
6864        LET .DELTAY =  ACT.Y2(.AC) - ACT.Y1(.AC)
6865        IF .DELTAX = 0
6866           IF .DELTAY LT 0
6867              LET ACT.ANGLE.P1.P2(.AC) = -PI.C/2.
6868           ELSE
6869              LET ACT.ANGLE.P1.P2(.AC) =  PI.C/2.
6870           ALWAYS
6871        ELSE
6872           IF .DELTAY = 0
6873              IF .DELTAX LT 0
6874                 LET ACT.ANGLE.P1.P2(.AC) =  PI.C
6875              ELSE
6876                 LET ACT.ANGLE.P1.P2(.AC) =  0.0
6877              ALWAYS
6878           ELSE
6879              LET ACT.ANGLE.P1.P2 =  ARCTAN.F(REAL.F(.DELTAY),
6880              REAL.F(.DELTAX))
6881           ALWAYS
6882        ALWAYS
6883
6884        ''SET THE TIME REQUIRED TO MAKE A PASS
6885        LET .DISTANCE = SQRT.F((REAL.F((ACT.X2(.AC) - ACT.X1(.AC))**2
6886        + (ACT.Y2(.AC) - ACT.Y1(.AC))**2
6887        + (ACT.Z2(.AC) - ACT.Z1(.AC))**2)))
6888        + SQRT.F((REAL.F((ACT.X3(.AC) - ACT.X2(.AC))**2
6889        +         (ACT.Y3(.AC) - ACT.Y2(.AC))**2
6890        +         (ACT.Z3(.AC) - ACT.Z2(.AC))**2)))
6891        + SQRT.F((REAL.F((ACT.X1(.AC) - ACT.X3(.AC))**2
6892        +         (ACT.Y1(.AC) - ACT.Y3(.AC))**2
6893        +         (ACT.Z1(.AC) - ACT.Z3(.AC))**2)))
6894        LET ACT.PASS.TIME(.AC) = .DISTANCE / (ACT.SPEED(.AC) * 60 / 7)
6895     ENDLOOP
6896
6897     ''ACT.SUBSTITUTE NOW POINTS TO AN EQUIPMENT.
6898     ''SET IT TO POINT TO AN AC.TYPE
6899     LOOP FOR EVERY AC.TYPE CALLED .AC
6900     DO
6901        IF ACT.SUBSTITUTE(.AC) NE 0
6902           FOR EVERY AC.TYPE
6903           WITH ACT.EQUIP.ID(AC.TYPE) = ACT.SUBSTITUTE(.AC)
6904           FIND THE FIRST CASE
6905           IF NONE
6906              TRACE
6907 <----------STOP
6908           OTHERWISE
6909           LET ACT.SUBSTITUTE(.AC) = AC.TYPE
```

```
6910        ALWAYS
6911     ENDLOOP
6912
6913     ''READ AND SCHEDULE THE PREPLANNED CAS MISSIONS.
6914     READ .NUM.MSNS
6915     IF .NUM.MSNS EQ 0
6916 <-------RETURN
6917     OTHERWISE
6918     LOOP FOR .I = 1 TO .NUM.MSNS
6919     DO
6920        SKIP 1 FIELD
6921        READ .ATK.TIME,
6922        .TGT.NOS,
6923        .NAME,
6924        .NUM.AC
6925        FOR EVERY UNIT CALLED .TGT
6926        WITH UNIT.NOS(.TGT) = .TGT.NOS
6927        FIND THE FIRST CASE
6928        IF NONE
6929           TRACE
6930 <----------STOP
6931        OTHERWISE
6932
6933        FOR EVERY EQUIPMENT
6934        WITH EQ.NAME(EQUIPMENT) = .NAME
6935        FIND THE FIRST CASE
6936        IF NONE
6937           TRACE
6938 <----------STOP
6939        OTHERWISE
6940
6941        FOR EVERY AC.TYPE CALLED .AC
6942         WITH ACT.EQUIP.ID(.AC) = EQUIPMENT
6943        FIND THE FIRST CASE
6944        IF NONE
6945           TRACE
6946 <----------STOP
6947        OTHERWISE
6948
6949        ''SUBTRACT THE PREP TIME FROM THE ATTACK TIME.
6950        LET .ATK.TIME = .ATK.TIME
6951        - UNIFORM.F(REAL.F(ACT.MIN.PREP.TIME(.AC)),
6952        REAL.F(ACT.MAX.PREP.TIME(.AC)),
6953        RN.SEED) / 60.
6954
6955        ''SUBTRACT THE FLIGHT TIME FROM THE ATTACK TIME.
6956        IF UN.COLOR(.TGT) = BLUE
6957           LET .AIRFIELD = SD.AIRFIELD(RED)
6958        ELSE
6959           LET .AIRFIELD = SD.AIRFIELD(BLUE)
6960        ALWAYS
6961        LET .DISTANCE = SQRT.F((REAL.F((UN.X.COORD(.AIRFIELD)
6962        - UN.X.COORD(.TGT))**2
6963        +(UN.Y.COORD(.AIRFIELD) - UN.Y.COORD(.TGT))**2)))
6964        SUBTRACT .DISTANCE / (ACT.SPEED(.AC) * 3600)
6965        FROM .ATK.TIME
6966        LET .ATK.TIME = MAX.F(.ATK.TIME, 0.0)
6967
```

```
6968        SCHEDULE_AN INIT.PREPLAN.CAS------------------------------------------------------------------------------>(380)
6969        GIVEN
6970        .TGT,
6971        .AC,
6972        .NUM.AC
6973        AT .ATK.TIME
6974     ENDLOOP
6975
6976 <--RETURN
6977 END
```

```
6978 ROUTINE MADS.INPUT                          ''                                                    I039
6979
6980    ADD 1 TO ANAL.CTR(223,1)  ''                                                                  \DYN_ANAL
6981    NORMALLY MODE IS INTEGER
6982
6983    READ N.MODEL.AD.SENSOR
6984    CREATE EACH MODEL.AD.SENSOR
6985    LOOP FOR EACH MODEL.AD.SENSOR CALLED .MADS
6986    DO
6987       SKIP 1 FIELD
6988       READ MADS.NAME(.MADS),
6989       MADS.DELAY.TIME(.MADS),
6990       MADS.PW.DEGRADE(.MADS),
6991       MADS.FCM(.MADS),
6992       MADS.RIPL(.MADS),
6993       MADS.WPN.RELOAD.TIME(.MADS),
6994       MADS.XMIT.PCT(.MADS)
6995       READ .NUM.RH
6996       LOOP  FOR .I = 1 TO .NUM.RH
6997       DO
6998          CREATE A MADS.RH CALLED .RH
6999          READ .RANGE,      ''DKM
7000          .ALT,             ''DKM
7001          MRH.PD(.RH)
7002          LET MRH.RANGE(.RH) = 10 * .RANGE / 16 ''                                                \OPTIMIZE
7003          LET MRH.MIN.ALT(.RH) = 10 * .ALT / 16
7004          FILE .RH IN THE MADS.RH.SET(.MADS)
7005       ENDLOOP
7006    ENDLOOP
7007
7008
7009 <--RETURN
7010 END
```

```
7011 ROUTINE AC.MUNS.INPUT                                          ''          I040
7012
7013    ADD 1 TO ANAL.CTR(224,1)  ''                                            \DYN_ANAL
7014    NORMALLY MODE IS INTEGER
7015
7016    READ N.AC.MUNS
7017    CREATE EACH AC.MUNS
7018    LOOP FOR EACH AC.MUNS
7019    DO
7020       SKIP 1 FIELD
7021       READ AM.NAME(AC.MUNS),
7022       AM.RELY(AC.MUNS),
7023       AM.RADIUS(AC.MUNS)
7024       FOR EVERY AC.TYPE
7025       READ ATM.DELIV.CEP(AC.TYPE, AC.MUNS)
7026    ENDLOOP
7027
7028    LOOP FOR EACH AC.MUNS
7029    DO
7030       SKIP 1 FIELD
7031       FOR EACH ENVIRONMENT
7032       FOR EACH POSTURE
7033       READ AMEP.LA.PERS(AC.MUNS, ENVIRONMENT, POSTURE)
7034    ENDLOOP
7035
7036    LOOP FOR EACH AC.MUNS
7037    FOR EACH ENVIRONMENT
7038    DO
7039       SKIP 1 FIELD
7040       FOR EACH TYPE.EQUIPMENT
7041       UNLESS TYPE.EQUIPMENT = 1 OR ''PERSONNEL
7042       TE.NAME(TYPE.EQUIPMENT) = "MINES"
7043       READ AMET.LA.EQUIP(AC.MUNS, ENVIRONMENT,
7044       TYPE.EQUIPMENT)
7045    ENDLOOP
7046
7047    IF TACAIR.DEBUG = 1
7048       PRINT 1 LINE WITH N.AC.MUNS, UNIT.NOS(SD.AIRFIELD(BLUE)),
7049       UNIT.NOS(SD.AIRFIELD(RED)) THUS
7050       N.AC.MUNS = **   BLUE AF = ******   RED AF = ******
7051       LOOP FOR EACH SIDE
7052       FOR EACH UE.LINK IN UN.EQUIP.LIST(SD.AIRFIELD(SIDE))
7053       DO
7054          LIST ATTRIBUTES OF UE.LINK
7055          LIST ATTRIBUTES OF EACH WEAPON IN UE.WEAPON.SET(UE.LINK)
7056       ENDLOOP
7057    ALWAYS
7058
7059    ''MAKE SURE THE POINTERS TO THE BOMBS (WPN.AC.MUNS)
7060    ''ARE VALID
7061    FOR EACH SIDE
7062    FOR EACH UE.LINK IN UN.EQUIP.LIST(SD.AIRFIELD(SIDE))
7063    FOR EACH WEAPON IN UE.WEAPON.SET(UE.LINK)
7064    WITH WPN.AC.MUNS(WEAPON) GT N.AC.MUNS
7065    FIND THE FIRST CASE
7066    IF FOUND
7067       TRACE
7068 <------STOP
```

```
7069     OTHERWISE
7070
7071 <--RETURN
7072 END
```

```
7073 ROUTINE TR.INPUT                                                    ''                I041
7074    GIVEN
7075    TARGET
7076
7077    ADD 1 TO ANAL.CTR(225,1)  ''                                                       \DYN_ANAL
7078    ''THIS ROUTINE READS EXTERNALLY GENERATED TARGET REPORTS
7079
7080    NORMALLY MODE IS INTEGER
7081
7082    READ TR.FDC(TARGET),
7083    TR.SENSOR.TYPE(TARGET),
7084    TR.REP.UNIT(TARGET),
7085    TR.TGT.UNIT(TARGET),
7086    TR.PGM.STATUS(TARGET),
7087    TR.MOVE(TARGET),
7088    TR.EST.X(TARGET),
7089    TR.EST.Y(TARGET),
7090    TR.CEP(TARGET)              ''TIMES 100
7091    LET TR.RECVD.TIME(TARGET) = TIME.V
7092    LET TR.ABORT.TIME(TARGET) = TIME.V + .25
7093    LET TR.EST.X(TARGET)=TR.EST.X(TARGET)*10./16.
7094    LET TR.EST.Y(TARGET)=TR.EST.Y(TARGET)*10./16.
7095
7096    LET FLAG=0
7097    LOOP UNTIL FLAG=1
7098    DO THIS
7099       CREATE A TR.DET.LINK CALLED LINK
7100       READ TR.DET.TE(LINK),
7101       TR.DET.ELEM.PROB(LINK),  ''TIMES 1000
7102       TR.DET.QUANT(LINK)
7103       IF TR.DET.TE(LINK) = 999
7104          DESTROY THE TR.DET.LINK CALLED LINK
7105          LET FLAG=1
7106       ELSE
7107          FILE LINK IN TR.DET.LIST(TARGET)
7108       ALWAYS
7109    ENDLOOP
7110
7111 <--EXITROUTINE
7112 ENDROUTINE
```

OUTPUT ROUTINES

```
7119 ROUTINE AMMO.RPT                                                    ''                                                    0001
7120
7121     ADD 1 TO ANAL.CTR(226,1)  ''                                                                                          \DYN_ANAL
7122     ''CALLED BY END.SIMULATION AND FIRE.MISSION
7123
7124     NORMALLY MODE IS INTEGER
7125     DEFINE BTRY AS AN INTEGER VARIABLE
7126     DEFINE .ID AS A TEXT VARIABLE ''                                                                                      \TEXT
7127     DEFINE DENSITY AS A 1-DIMENSIONAL ARRAY
7128
7129     RESERVE DENSITY AS  N.TYPE.BTRY
7130
7131     FOR EACH BTRY
7132     ADD 1 TO DENSITY(BY.TYPE(BTRY))
7133     WRITE AS *,/,/,B 12,"ARTY AMMO CONSUMPTION IN ROUNDS",/,/
7134
7135     LOOP FOR EACH TYPE.BTRY CALLED TB
7136     FOR EACH TML IN TB.TM.LIST(TB)
7137     DO
7138        WRITE TB,TB.NAME(TB),EQ.NAME(TB.HOW.EQ.ID(TB)),DENSITY(TB)
7139        AS I 5, S 3, T 6, S 3, T 6, S 1, I 5 ''                                                                            \TEXT
7140        WRITE TB,TB.NAME(TB),EQ.NAME(TB.HOW.EQ.ID(TB)),DENSITY(TB)
7141        AS I 5, S 3, T 6, S 3, T 6, S 1, I 5 USING UNIT 40 ''                                                              \TEXT
7142        WRITE TB.TM.CLASS(TML) AS S 1, T 6 ''                                                                              \TEXT
7143        WRITE TB.TM.CLASS(TML) AS S 1, T 6 USING UNIT 40 ''                                                                \TEXT
7144        LET EQ = TB.HOW.EQ.ID(TB)
7145        IF EQ GT N.BLUE.TYPE.EQP
7146           LET SIDE = RED
7147        ELSE
7148           LET SIDE = BLUE
7149        ALWAYS
7150        LET .ID = " "
7151        IF TB.TM.CLASS(TML) = "HE"
7152           LET .ID = HE.ID(TB.TM(TML))
7153           LET WEIGHT = HE.WEIGHT(TB.TM(TML))
7154        ALWAYS
7155        IF TB.TM.CLASS(TML) = "ICM"
7156           LET .ID = IC.ID(TB.TM(TML))
7157           LET WEIGHT = IC.WEIGHT(TB.TM(TML))
7158        ALWAYS
7159        IF TB.TM.CLASS(TML) = "FASCAM"
7160           LET .ID = FMM.ID(TB.TM(TML))
7161           LET WEIGHT = FMM.RND.WT(TB.TM(TML))
7162        ALWAYS
7163        IF TB.TM.CLASS(TML) = "ILLUM"
7164           LET .ID = ILLUM.ID(TB.TM(TML))
7165           LET WEIGHT = ILLUM.RND.WT(TB.TM(TML))
7166        ALWAYS
7167        IF TB.TM.CLASS(TML) = "SMOKE"
7168           LET .ID = SMK.ID(TB.TM(TML))
7169           LET WEIGHT = SMK.RND.WT(TB.TM(TML))
7170        ALWAYS
7171        IF .ID = " "
7172           WRITE AS "UNKNOWN"
7173        ELSE
7174           WRITE .ID AS S 1, T 6 ''                                                                                        \TEXT
7175           WRITE .ID AS S 1, T 6   USING UNIT 40  ''                                                                       \TEXT
7176        ALWAYS
```

```
7177        WRITE TB.TM.FIRED(TML) AS I 10
7178        WRITE TB.TM.FIRED(TML) AS I 10 USING UNIT 40
7179        WRITE AS /
7180        WRITE AS / USING UNIT 40
7181        IF EQ.KV.ID(EQ) GT 0
7182           ADD (TB.TM.FIRED(TML) * WEIGHT * 100.) TO KV.AMMO.CONSUMED(SIDE,
7183           EQ.KV.ID(EQ))
7184        ALWAYS
7185        IF TB.NAME(TB) = PGM.MARK AND TB.TM.CLASS(TML) = "ICM"
7186           WRITE TB,TB.NAME(TB),EQ.NAME(TB.HOW.EQ.ID(TB)),DENSITY(TB)
7187           AS I 5, S 3, T 6, S 3, T 6, S 1, I 5, S 1,"PGM    ", ''                    \TEXT
7188           S 1, "XXXX  "
7189           WRITE TB,TB.NAME(TB),EQ.NAME(TB.HOW.EQ.ID(TB)),DENSITY(TB)
7190           AS I 5, S 3, T 6, S 3, T 6, S 1, I 5, S 1,"PGM    ", ''                    \TEXT
7191           S 1, "XXXX  "
7192           USING UNIT 40
7193           WRITE NUM.PGM.FIRED AS I 10
7194           WRITE NUM.PGM.FIRED AS I 10 USING UNIT 40
7195           WRITE AS /
7196           WRITE AS / USING UNIT 40
7197           IF EQ.KV.ID(EQ) GT 0
7198              ADD (NUM.PGM.FIRED * 220 * 100.) TO
7199              KV.AMMO.CONSUMED(SIDE,EQ.KV.ID(EQ))
7200           ALWAYS
7201        ALWAYS
7202        IF TB.NAME(TB) = SDM.MARK AND TB.TM.CLASS(TML) = "ICM"
7203           WRITE TB,TB.NAME(TB),EQ.NAME(TB.HOW.EQ.ID(TB)),DENSITY(TB)
7204           AS I 5, S 3, T 6, S 3, T 6, S 1, I 5, S 1,"SDM    ", ''                    \TEXT
7205           S 1, "YYYY  "
7206           WRITE TB,TB.NAME(TB),EQ.NAME(TB.HOW.EQ.ID(TB)),DENSITY(TB)
7207           AS I 5, S 3, T 6, S 3, T 6, S 1, I 5, S 1,"SDM    ", ''                    \TEXT
7208           S 1, "YYYY  "
7209           USING UNIT 40
7210           WRITE TOT.FIRED.SDM AS I 10
7211           WRITE TOT.FIRED.SDM AS I 10 USING UNIT 40
7212           WRITE AS /
7213           WRITE AS / USING UNIT 40
7214           IF EQ.KV.ID(EQ) GT 0
7215              ADD (TOT.FIRED.SDM * 200 * 100.) TO
7216              KV.AMMO.CONSUMED(SIDE,EQ.KV.ID(EQ))
7217           ALWAYS
7218        ALWAYS
7219     ENDLOOP
7220
7221     PRINT 6 LINES WITH NUM.KILL.PGM AND TOT.KILL.SDM THUS
7222
7223
7224
7225     NUMBER OF PGM KILLS IS *****
7226     NUMBER OF SADARM KILLS IS ********
7227
7228     RELEASE DENSITY(*)
7229
7230     WRITE AS *,/,/,/,B 12,"SUE AMMO BY TYPE WEAPON (ROUNDS)",/,/
7231     LOOP FOR EACH TYPE.WEAPON CALLED TW,
7232     DO
7233        WRITE TW,TW.NAME(TW),STW.RND.FIRED(BLUE,TW),STW.RND.FIRED(RED,TW)
7234        AS I 5, S 3, T 6, S 3, 2 I 10, / ''                                        \TEXT
```

```
7235     ENDLOOP
7236
7237 <--EXITROUTINE
7238 ENDROUTINE
```

```
7239 ROUTINE ANALYSIS.OUTPUT                                                    ''                                                  0002
7240
7241    ADD 1 TO ANAL.CTR(227,1)  ''                                                                                               \DYN_ANAL
7242    NORMALLY MODE IS INTEGER
7243    DEFINE COMBAT.UNITS, WPN.QUANT, HE.N.TUBES, IC.N.TUBES,
7244    .FM.N.TUBES, .IL.N.TUBES, .SM.N.TUBES AS 1-DIM ARRAYS
7245
7246    RESERVE COMBAT.UNITS AS 2,
7247    WPN.QUANT AS N.TYPE.WEAPON,
7248    HE.N.TUBES AS N.HE.MUNITION,
7249    IC.N.TUBES AS N.IC.MUNITION,
7250    .FM.N.TUBES AS N.FASCAM.MUNITION,
7251    .IL.N.TUBES AS N.ILLUM.MUNITION,
7252    .SM.N.TUBES AS N.SMOKE.MUNITION
7253
7254    USE UNIT 60 FOR OUTPUT
7255
7256    FOR EVERY UNIT,
7257    WITH CT.GROUP(TU.CAT(UN.TYPE.UNIT(UNIT))) = 1,
7258    DO
7259       ADD 1 TO COMBAT.UNITS(UN.COLOR(UNIT))
7260    ENDLOOP
7261    WRITE COMBAT.UNITS(1), COMBAT.UNITS(2), N.CATEGORY
7262    AS "STRT  ",I 5,S 2,I 5,/,I 5
7263    FOR EVERY CATEGORY CALLED CAT,
7264    DO
7265       IF MOD.F(CAT-1,10) = 0
7266          START NEW LINE
7267       ALWAYS
7268       WRITE CT.NAME(CAT) AS "'",T 6,"' " ''                                                                                   \TEXT
7269    ENDLOOP
7270
7271    WRITE N.TYPE.UNIT AS /,I 5
7272    FOR EVERY TYPE.UNIT CALLED TU,
7273    DO
7274       IF MOD.F(TU-1,4) = 0
7275          START NEW LINE
7276       ALWAYS
7277       WRITE TU.LEVEL(TU),TU.CAT(TU) AS "'",T 6,"'  ",I 4,S 4 ''                                                               \TEXT
7278    ENDLOOP
7279
7280    WRITE N.UNIT AS /,I 5
7281    FOR EVERY UNIT,
7282    DO
7283       IF MOD.F(UNIT-1,3) = 0
7284          START NEW LINE
7285       ALWAYS
7286       WRITE UNIT.NOS(UNIT),UN.PARENT(UNIT),UN.TYPE.UNIT(UNIT)
7287       AS I 5,S 2,I 5,S 2,I 4,S 4
7288    ENDLOOP
7289
7290    WRITE N.TYPE.EQUIPMENT AS /,I 5
7291    FOR EACH TYPE.EQUIPMENT CALLED TE,
7292    DO
7293       IF MOD.F(TE-1,8) = 0
7294          START NEW LINE
7295       ALWAYS
7296       WRITE TE.NAME(TE) AS "'",T 6,"' " ''                                                                                    \TEXT
```

```
7297    ENDLOOP
7298
7299    WRITE N.BLUE.TYPE.EQP,N.RED.TYPE.EQP AS /,2 I 7
7300    LET I = 2
7301    FOR EACH EQUIPMENT CALLED EQ,
7302    DO
7303       IF EQ>N.BLUE.TYPE.EQP AND I = 2
7304          LET I = 1
7305       ALWAYS
7306       IF MOD.F(EQ-1,3) = 0
7307          START NEW LINE
7308       ALWAYS
7309       WRITE EQ.NAME(EQ),EQ.TE.PTR(EQ),
7310       KV.INITIAL.DENSITY(I,EQ.KV.ID(EQ)) AS
7311       "'",T 6,"'  ",I 3,S 2,I 7,S 4 ''                                    \TEXT
7312    ENDLOOP
7313
7314    FOR EVERY UNIT,
7315    FOR EVERY UEL IN UN.EQUIP.LIST(UNIT),
7316    FOR EVERY WEAPON IN UE.WEAPON.SET(UEL),
7317    ADD WPN.QUANTITY(WEAPON)*UE.QUANT(UEL) TO WPN.QUANT(WPN.ID(WEAPON))
7318
7319    WRITE N.R.WPN.TYPE,N.B.WPN.TYPE AS /,I 5,S 2,I 5
7320    FOR EVERY TYPE.WEAPON CALLED TW,
7321    DO
7322       IF MOD.F(TW-1,4) = 0
7323          START NEW LINE
7324       ALWAYS
7325       WRITE TW.NAME(TW),WPN.QUANT(TW) AS "'",T 6,"'  ",I 5,S 3 ''          \TEXT
7326    ENDLOOP
7327
7328    RELEASE WPN.QUANT(*)
7329
7330    WRITE N.HE.MUNITION, N.IC.MUNITION, N.FASCAM.MUNITION,
7331    N.ILLUM.MUNITION, N.SMOKE.MUNITION, 1, 1 AS
7332    /, "    4", /, "'HE    ' ", I 4, " 'ICM   ' ", I 4,
7333    " 'FASCAM' ",I 4," 'ILLUM ' ",I 4," 'SMOKE ' ",I 4," 'PGM   ' ",
7334    I 4, " 'SDM   ' ", I 4
7335    FOR EVERY BTRY,
7336    DO
7337       LET TB = BY.TYPE(BTRY)
7338       FOR EVERY TML IN TB.TM.LIST(TB), ''                    REMOVED ALSO \1
7339       DO
7340          IF TB.TM.CLASS(TML) = "HE",
7341             ADD TB.N.HOW(TB) TO HE.N.TUBES(TB.TM(TML))
7342          ALWAYS
7343          IF TB.TM.CLASS(TML) = "ICM"
7344             ADD TB.N.HOW(TB) TO IC.N.TUBES(TB.TM(TML))
7345          ALWAYS
7346          IF TB.TM.CLASS(TML) = "FASCAM"
7347             ADD TB.N.HOW(TB) TO .FM.N.TUBES(TB.TM(TML))
7348          ALWAYS
7349          IF TB.TM.CLASS(TML) = "ILLUM"
7350             ADD TB.N.HOW(TB) TO .IL.N.TUBES(TB.TM(TML))
7351          ALWAYS
7352          IF TB.TM.CLASS(TML) = "SMOKE"
7353             ADD TB.N.HOW(TB) TO .SM.N.TUBES(TB.TM(TML))
7354          ALWAYS
```

```
7355        ENDLOOP
7356     ENDLOOP ''                                                    ADD ENDLOOP \1
7357     FOR EVERY HE.MUNITION CALLED HE,
7358     DO
7359        IF MOD.F(HE-1,4) = 0
7360           START NEW LINE
7361        ALWAYS
7362        WRITE HE.ID(HE),HE.N.TUBES(HE) AS "'",T 6,"' ",I 4,S 3 ''          \TEXT
7363     ENDLOOP
7364
7365     FOR EVERY IC.MUNITION CALLED ICM,
7366     DO
7367        IF MOD.F(ICM-1,4) = 0
7368           START NEW LINE
7369        ALWAYS
7370        WRITE IC.ID(ICM),IC.N.TUBES(ICM) AS "'",T 6,"' ",I 4,S 3 ''        \TEXT
7371     ENDLOOP
7372
7373     FOR EVERY FASCAM.MUNITION CALLED .FM
7374     DO
7375        IF MOD.F(.FM-1, 4) = 0
7376           START NEW LINE
7377        ALWAYS
7378        WRITE FMM.ID(.FM), .FM.N.TUBES(.FM) AS "'", T 6, "' ", I 4, S 3 ''  \TEXT
7379     ENDLOOP
7380
7381     FOR EVERY ILLUM.MUNITION CALLED .IM
7382     DO
7383        IF MOD.F(.IM-1, 4) = 0
7384           START NEW LINE
7385        ALWAYS
7386        WRITE ILLUM.ID(.IM), .IL.N.TUBES(.IM) AS "'", T 6, "' ", I 4, S 3 ''  \TEXT
7387     ENDLOOP
7388
7389     FOR EVERY SMOKE.MUNITION CALLED .SM
7390     DO
7391        IF MOD.F(.SM-1, 4) = 0
7392           START NEW LINE
7393        ALWAYS
7394        WRITE SMK.ID(.SM), .SM.N.TUBES(.SM) AS "'", T 6, "' ", I 4, S 3 ''  \TEXT
7395     ENDLOOP
7396
7397     RELEASE HE.N.TUBES, IC.N.TUBES, .FM.N.TUBES, .IL.N.TUBES, .SM.N.TUBES
7398
7399     LET N.PGM.TUBES = 0
7400     LET N.SDM.TUBES = 0
7401     FOR EVERY BTRY,
7402     DO
7403        IF BY.PGM.CAP(BTRY) = 1,
7404           ADD TB.N.HOW(BY.TYPE(BTRY)) TO N.PGM.TUBES
7405        ELSE
7406           IF BY.PGM.CAP(BTRY) = 2,
7407              ADD TB.N.HOW(BY.TYPE(BTRY)) TO N.SDM.TUBES
7408           ALWAYS
7409        ALWAYS
7410     ENDLOOP
7411     WRITE N.PGM.TUBES,N.SDM.TUBES,N.TYPE.BTRY AS
7412     /,"'PGM    ' ",I 5,/,"'SDM    ' ",I 5,/,I 5,/
```

```
7413     FOR EVERY TYPE.BTRY CALLED TB,
7414     DO
7415        WRITE TB.NAME(TB),TB.HOW.EQ.ID(TB),N.TB.TM.LIST(TB) AS
7416        "'",T 6,"'   ",I 5,S 2,I 3,/ ''                                    \TEXT
7417        FOR EVERY TML IN TB.TM.LIST(TB)
7418        DO
7419           WRITE TB.TM.CLASS(TML),TB.TM(TML)
7420           AS "    '",T 6,"'   ",I 4,/ ''                                  \TEXT
7421        ENDLOOP
7422     ENDLOOP
7423
7424     USE UNIT 6 FOR OUTPUT
7425
7426 <--EXITROUTINE
7427 ENDROUTINE
```

```
7428 ROUTINE BETWEEN.ROUTINE                                                    ''                                                  0003
7429
7430    ADD 1 TO ANAL.CTR(228,1)  ''                                                                                                \DYN_ANAL
7431    NORMALLY MODE IS INTEGER
7432
7433    IF DEBUG = 0
7434 <------EXITROUTINE
7435    OTHERWISE
7436
7437    ''WRITE SEED.V(RN.SEED), TIME.V, USED.DBANK.V,ZTIME.F,EVENT.V                                                               \VAX
7438    ''AS "SEED.V = ", I 15, S 3,
7439    ''"TIME.V = ", D (10,7), S 2, "DBANK = ", I 7, S 2,
7440    ''"ZTIME.F = ", D(12,7),S 2,"EVENT ", I 3, S 2
7441    WRITE SEED.V(RN.SEED), TIME.V AND EVENT.V AS
7442    "SEED.V =", I 15, S 3, "TIME.V =", D(10,7),
7443    S 2, "EVENT ", I 3, S 2
7444
7445    GO T(EVENT.V)
7446
7447    'T(1)'
7448    PRINT 1 LINE WITH AC.ATK.TGT. THUS
7449    AC.ATK.TGT. = *****
7450 <--EXITROUTINE
7451    'T(2)'
7452    PRINT 1 LINE WITH AIRBORNE.RADAR THUS
7453    AIRBORNE.RADAR = *****
7454 <--EXITROUTINE
7455    'T(3)'
7456    PRINT 1 LINE WITH AIR.OBSERVER THUS
7457    AIR.OBSERVER = *****
7458 <--EXITROUTINE
7459    'T(4)'
7460    PRINT 1 LINE WITH ARTY.ASSESS THUS
7461    ARTY.ASSESS = *****
7462 <--EXITROUTINE
7463    'T(5)'
7464    PRINT 1 LINE WITH ASSESSMENT THUS
7465    ASSESSMENT = *****
7466 <--EXITROUTINE
7467    'T(6)'
7468    PRINT 1 LINE WITH CAS.MISSION THUS
7469    CAS.MISSION = *****
7470 <--EXITROUTINE
7471    'T(7)'
7472    PRINT 1 LINE WITH FIRE.MISSION THUS
7473    FIRE.MISSION = *****
7474 <--EXITROUTINE
7475    'T(8)'
7476    PRINT 1 LINE WITH FORWARD.OBSERVER THUS
7477    FORWARD.OBSERVER = *****
7478 <--EXITROUTINE
7479    'T(9)'
7480    PRINT 1 LINE WITH HC.ARRIVE.BATTLE THUS
7481    HC.ARRIVE.BATTLE = *****
7482 <--EXITROUTINE
7483    'T(10)'
7484    PRINT 1 LINE WITH HC.RETURN.FARRP THUS
7485    HC.RETURN.FARRP = *****
```

```
7486 <--EXITROUTINE
7487     'T(11)'
7488     PRINT 1 LINE WITH HELICOPTER.FIRE THUS
7489     HELO.FIRE = *****
7490 <--EXITROUTINE
7491     'T(12)'
7492     PRINT 1 LINE WITH HEL.TARGET.ACQUISITION THUS
7493     HEL.TARGET.ACQUISITION=******
7494 <--EXITROUTINE
7495     'T(13)'
7496     PRINT 1 LINE WITH HOW.REPAIR THUS
7497     HOW.REPAIR = *****
7498 <--EXITROUTINE
7499     'T(14)'
7500     PRINT 1 LINE WITH MINE.ASSESS THUS
7501     MINE.ASSESS = *****
7502 <--EXITROUTINE
7503     'T(15)'
7504     PRINT 1 LINE WITH PHOTO.IR.FLIGHT THUS
7505     PHOTO.IR.FLIGHT = *****
7506 <--EXITROUTINE
7507     'T(16)'
7508     PRINT 1 LINE WITH REMOTE.PILOT.VEHICLE THUS
7509     REMOTE.PILOT.VEHICLE=*****
7510 <--EXITROUTINE
7511     'T(17)'
7512     ''PRINT 1 LINE WITH SHOOT.OUT THUS
7513     ''SHOOT.OUT = *****
7514 <--EXITROUTINE
7515     'T(18)'
7516     PRINT 1 LINE WITH TARGET.REPORT THUS
7517     TARGET.REPORT = *****
7518 <--EXITROUTINE
7519     'T(19)'
7520     PRINT 1 LINE WITH WITH.DRAW THUS
7521     WITH.DRAW = *****
7522 <--EXITROUTINE
7523     'T(20)'
7524     PRINT 1 LINE WITH CHANGE.LITE THUS
7525     CHANGE.LITE = *****
7526 <--EXITROUTINE
7527     'T(21)'
7528     PRINT 1 LINE WITH END.SIMULATION THUS
7529     END.SIMULATION = *****
7530 <--EXITROUTINE
7531     'T(22)'
7532     PRINT 1 LINE WITH ACT.ATK THUS
7533     ACT.ATK = *****
7534 <--EXITROUTINE
7535     'T(23)'
7536     PRINT 1 LINE WITH ACT.DEF THUS
7537     ACT.DEF = *****
7538 <--EXITROUTINE
7539     'T(24)'
7540     PRINT 1 LINE WITH ACT.MOVCOR THUS
7541     ACT.MOVCOR = *****
7542 <--EXITROUTINE
7543     'T(25)'
```

```
7544     PRINT 1 LINE WITH ACT.MOVDIS THUS
7545     ACT.MOVDIS = *****
7546 <--EXITROUTINE
7547     'T(26)'
7548     PRINT 1 LINE WITH ACT.REINF THUS
7549     ACT.REINF = *****
7550 <--EXITROUTINE
7551     'T(27)'
7552     PRINT 1 LINE WITH AD.ENGAGEMENT THUS
7553     AD.ENGAGEMENT = *****
7554 <--EXITROUTINE
7555     'T(28)'
7556     PRINT 1 LINE WITH ARTY.OCCUPATION THUS
7557     ARTY.OCCUPATION = *****
7558 <--EXITROUTINE
7559     'T(29)'
7560     PRINT 1 LINE WITH BTL.ENDED THUS
7561     BTL.ENDED = *****
7562 <--EXITROUTINE
7563     'T(30)'
7564     PRINT 1 LINE WITH CFR.ACTIVATION THUS
7565     CFR.ACTIVATION = *****
7566 <--EXITROUTINE
7567     'T(31)'
7568     PRINT 1 LINE WITH CFR.OFF THUS
7569     CFR.OFF = *****
7570 <--EXITROUTINE
7571     'T(32)'
7572     PRINT 1 LINE WITH CFR.ON THUS
7573     CFR.ON = *****
7574 <--EXITROUTINE
7575     'T(33)'
7576     PRINT 1 LINE WITH CFR.OPERATOR THUS
7577     CFR.OPERATOR = ******
7578 <--EXITROUTINE
7579     'T(34)'
7580     PRINT 1 LINE WITH CHANGE.WEATHER THUS
7581     CHANGE.WEATHER = *****
7582 <--EXITROUTINE
7583     'T(35)'
7584     PRINT 1 LINE WITH DQ.OLD.SORTIE.QUEUE THUS
7585     DQ.OLD.SORTIE.QUEUE = *****
7586 <--EXITROUTINE
7587     'T(36)'
7588     PRINT 1 LINE WITH ENGAGEMENT THUS
7589     ENGAGEMENT = *****
7590 <--EXITROUTINE
7591     'T(37)'
7592     PRINT 1 LINE WITH FEBA.SORTIE THUS
7593     FEBA.SORTIE = *****
7594 <--EXITROUTINE
7595     'T(38)'
7596     PRINT 1 LINE WITH GET.NX.ORD THUS
7597     GET.NX.ORD = *****
7598 <--EXITROUTINE
7599     'T(39)'
7600     PRINT 1 LINE WITH HC.DEPART.BATTLE THUS
7601     HC.DEPART.BATTLE = *****
```

```
7602 <--EXITROUTINE
7603     'T(40)'
7604     PRINT 1 LINE WITH HELO.ENGAGEMENT THUS
7605     HELO.ENGAGEMENT = *****
7606 <--EXITROUTINE
7607     'T(41)'
7608     PRINT 1 LINE WITH INIT.PREPLAN.CAS THUS
7609     INIT.PREPLAN.CAS = *****
7610 <--EXITROUTINE
7611     'T(42)'
7612     PRINT 1 LINE WITH MOVE THUS
7613     MOVE = *****
7614 <--EXITROUTINE
7615     'T(43)'
7616     PRINT 1 LINE WITH PDB.ACTIVATION THUS
7617     PDB.ACTIVATION = *****
7618 <--EXITROUTINE
7619     'T(44)'
7620     PRINT 1 LINE WITH PDB.OPERATOR THUS
7621     PDB.OPERATOR = *****
7622 <--EXITROUTINE
7623     'T(45)'
7624     PRINT 1 LINE WITH SCHEDULE.ARTY.MOVEMENT THUS
7625     SCHEDULE.ARTY.MOVEMENT=******
7626 <--EXITROUTINE
7627     'T(46)'
7628     PRINT 1 LINE WITH SEND.TEAM THUS
7629     SEND.TEAM = *****
7630 <--EXITROUTINE
7631     'T(47)'
7632     PRINT 1 LINE WITH START.ARTY.MOVEMENT THUS
7633     START.ARTY.MOVEMENT=******
7634 <--EXITROUTINE
7635     'T(48)'
7636     PRINT 1 LINE WITH START.BATTLE THUS
7637     START.BATTLE = *****
7638 <--EXITROUTINE
7639     'T(49)'
7640     PRINT 1 LINE WITH START.MOVE THUS
7641     START.MOVE = *****
7642 <--EXITROUTINE
7643     'T(50)'
7644     PRINT 1 LINE WITH STOP.ARTY.MOVEMENT THUS
7645     STOP.ARTY.MOVEMENT = *****
7646 <--EXITROUTINE
7647     'T(51)'
7648     PRINT 1 LINE WITH UPDATE.LOC THUS
7649     UPDATE.LOC = *****
7650 <--EXITROUTINE
7651     'T(52)'
7652     PRINT 1 LINE WITH POSITION.REPORT THUS
7653     POSITION.REPORT = *****
7654 <--EXITROUTINE
7655     'T(53)'
7656     PRINT 1 LINE WITH SET.DEBUG THUS
7657     SET.DEBUG = *****
7658
7659 <--EXITROUTINE
```

```
7660     'T(54)'
7661     PRINT 1 LINE WITH OFF.LINE.ATTRITION THUS
7662     OFF.LINE.ATTRITION = *********
7663 <--EXITROUTINE
7664 ENDROUTINE
```

```
7665 ROUTINE ERROR.STOP                                                    ''                    0004
7666
7667    ADD 1 TO ANAL.CTR(229,1)  ''                                                             \DYN_ANAL
7668    NORMALLY MODE IS INTEGER
7669
7670    PRINT 2 LINES WITH TIME.V THUS
7671    ERROR CONDITION EXISTS
7672    SIMULATION ENDED AT ****.***
7673
7674    TRACE
7675    PERFORM SNAP.R------------------------------------------------------------------------------>(618)
7676
7677 <--STOP
7678 ENDROUTINE
```

```
7679 ROUTINE HEADING                                        ''0005
7680
7681    ADD 1 TO ANAL.CTR(230,1)  ''                                    \DYN_ANAL
7682    NORMALLY MODE IS INTEGER
7683
7684    WRITE AS S 70, "COSAGE VERSION 4  = = =  UNCLASSIFIED", /
7685
7686 <--EXITROUTINE
7687 ENDROUTINE
```

```
7688 ROUTINE KV.PRINT GIVEN ''                                                  0006
7689     KSIDE,VSIDE,KILNUM,VICNUM
7690     ADD 1 TO ANAL.CTR(231,1)  ''                                           \DYN_ANAL
7691     ''WHERE KSIDE = KILLER SIDE NUMBER
7692     ''       VSIDE = VICTIM SIDE NUMBER
7693     ''       KILNUM  = NUMBER OF KILLERS
7694     ''       VICNUM= NUMBER OF VICTIMS
7695     NORMALLY MODE IS INTEGER
7696     DEFINE VICTIM,KILLER,KSIDE,VSIDE,VICNUM,
7697     KILNUM AS INTEGER VARIABLES
7698     '' SUM THE TOTAL KILLS FOR EACH VICTIM
7699     DEFINE KILL.TOTAL AS AN INTEGER, 1-DIMENSIONAL ARRAY
7700     RESERVE KILL.TOTAL(*) AS VICNUM ''RELEASE AFTER PRINTING VALUES
7701     LOOP FOR VICTIM = 1 TO VICNUM
7702     DO ''                                                                  \1
7703        LOOP FOR KILLER = 1 TO KILNUM
7704        DO  ''                                                              \1
7705           LET KILL.TOTAL(VICTIM) = KILL.TOTAL(VICTIM)+
7706           KV.SCORE(KSIDE,KILLER,VICTIM)
7707        ENDLOOP ''KILLER LOOP
7708     ENDLOOP ''VICTIM LOOP
7709     ''
7710     ''
7711     ''      THIS SECTION OUTPUTS INTERMEDIATE DATA TO UNITS 9 AND 10 FOR
7712     ''    LATER STATISTICAL ANALYSIS.  TO "CATCH" THESE DATA IN FILES,
7713     ''    THE COMMANDS:
7714     ''
7715     ''      @ASG,OPTIONS  FILENAME1.
7716     ''      @ASG,OPTIONS  FILENAME2.
7717     ''      @USE  SIMU9,FILENAME1.
7718     ''      @USE  SIMU10,FILENAME2.
7719     ''
7720     ''    MUST BE PRESENT IN THE RUNSTREAM.  EITHER OF THE TWO RESULTING FILES
7721     ''    CAN THEN BE USED AS INPUT TO THE  STATKV  PROGRAM BY MEANS OF AN
7722     ''    @ADD  COMMAND.
7723     ''
7724     ''
7725     IF TIME.V >= STOP.SIMULATION.TIME
7726        DEFINE PERCENT.LOSS AS A 1-DIM REAL ARRAY
7727        RESERVE PERCENT.LOSS AS VICNUM
7728        FOR VICTIM=1 TO VICNUM,                                             \1
7729        DO  ''
7730           IF KV.INITIAL.DENSITY(VSIDE,VICTIM)>0
7731              LET PERCENT.LOSS(VICTIM)=100.*KILL.TOTAL(VICTIM)/
7732              KV.INITIAL.DENSITY(VSIDE,VICTIM)
7733           REGARDLESS
7734        ENDLOOP  ''                                   CHANGED REPEAT TO ENDLOOP \1
7735        USE UNIT 8+KSIDE FOR OUTPUT
7736        WRITE KSIDE,VSIDE,KILNUM,VICNUM AS 2 I 2,2 I 4,/,S 19
7737        FOR VICTIM=1 TO VICNUM,                                             \1
7738        DO ''                                                               \1
7739           IF MOD.F(VICTIM,10)=0 ''
7740              WRITE AS /,S 19
7741           REGARDLESS
7742           WRITE KV.INITIAL.DENSITY(VSIDE,VICTIM) AS I 7
7743        ENDLOOP  ''                                   CHANGED REPEAT TO ENDLOOP \1
7744        WRITE AS /,S 19
7745        FOR VICTIM=1 TO VICNUM,
```

```
7746        DO  ''                                                                                    \1
7747           IF MOD.F(VICTIM,10)=0 ''                                                               \1
7748              WRITE AS /,S 19
7749           REGARDLESS
7750           WRITE EQ.NAME(KV.EQ.ID(VSIDE,VICTIM)) AS S 1,T 6 ''                                    \TEXT
7751        ENDLOOP  ''                                                          CHANGED REPEAT TO ENDLOOP \1
7752        WRITE AS /,S 19
7753        FOR VICTIM=1 TO VICNUM,
7754        DO  ''                                                                                    \1
7755           IF MOD.F(VICTIM,10)=0 ''                                                               \1
7756              WRITE AS /,S 19
7757           REGARDLESS
7758           WRITE KV.CEM.WPN.NO(VSIDE,VICTIM) AS I 7
7759        ENDLOOP  ''                                                          CHANGED REPEAT TO ENDLOOP \1
7760        FOR KILLER =1 TO KILNUM,
7761        DO  ''                                                                                    \1
7762           WRITE EQ.NAME(KV.EQ.ID(KSIDE,KILLER)),KV.CEM.WPN.NO(KSIDE,KILLER),
7763           KV.INITIAL.DENSITY(KSIDE,KILLER) AS /,S 1,T 6,I 5,I 7 ''                               \TEXT
7764           FOR VICTIM=1 TO VICNUM,  ''                                          REMOVED ALSO \1
7765           DO  ''                                                                                 \1
7766              IF MOD.F(VICTIM,10)=0 ''                                                            \1
7767                 WRITE AS /,S 19
7768              REGARDLESS
7769              WRITE KV.SCORE(KSIDE,KILLER,VICTIM) AS I 7
7770           ENDLOOP  ''                                                       CHANGED REPEAT TO ENDLOOP \1
7771           WRITE AS /,S 19
7772           FOR VICTIM=1 TO VICNUM,
7773           DO  ''                                                                                 \1
7774              IF MOD.F(VICTIM,10)=0 ''                                                            \1
7775                 WRITE AS /,S 19
7776              REGARDLESS
7777              WRITE KILL.TOTAL(VICTIM) AS I 7
7778           ENDLOOP  ''                                                       CHANGED REPEAT TO ENDLOOP \1
7779           WRITE AS /,S 19
7780           FOR VICTIM=1 TO VICNUM,
7781           DO  ''                                                                                 \1
7782              IF MOD.F(VICTIM,10)=0 ''                                                            \1
7783                 WRITE AS /,S 19
7784              REGARDLESS
7785              WRITE INT.F(PERCENT.LOSS(VICTIM)*10) AS I 7
7786           ENDLOOP  ''                                                       CHANGED REPEAT TO ENDLOOP \1
7787        ENDLOOP  ''                                          SUPPLIED TO MATCH ALSO WHICH WAS REMOVED \1
7788        USE UNIT 6 FOR OUTPUT
7789     REGARDLESS
7790     ''
7791     ''
7792     BEGIN REPORT ON A NEW PAGE
7793     PRINTING FOR VICTIM = 1 TO VICNUM IN GROUPS OF 16 PER PAGE
7794     BEGIN HEADING ''SECTION WHICH TITLES EACH VICTIM ROW
7795     PRINT 1 LINE WITH SHADE(KSIDE),SHADE(VSIDE) THUS
7796     **** KILLER VS **** VICTIM SCOREBOARD
7797     SKIP 1 OUTPUT LINE
7798     PRINT 1 DOUBLE LINE WITH
7799     A GROUP OF KV.INITIAL.DENSITY(VSIDE,VICTIM) FIELDS
7800     THUS
7801     DENSITY****** ****** ****** ****** ****** ****** ****** ****** ******
7802     ****** ****** ****** ****** ****** ******
7803     PRINT 1 DOUBLE LINE WITH
```

```
7804    A GROUP OF EQ.NAME(KV.EQ.ID(VSIDE,VICTIM)) FIELDS
7805    THUS
7806    VICTIM ****** ****** ****** ****** ****** ****** ****** ****** ******
7807    ****** ****** ****** ****** ****** ****** ******
7808    PRINT 1 DOUBLE LINE WITH
7809    A GROUP OF KV.CEM.WPN.NO(VSIDE,VICTIM) FIELDS
7810    THUS
7811    CEM NO  (***)  (***)  (***)  (***)  (***)  (***)  (***)  (***)  (***)
7812    (***)  (***)  (***)  (***)  (***)  (***)  (***)
7813    SKIP 2 OUTPUT LINES
7814    PRINT 1 LINE THUS
7815    SHOOTER/DENSITY
7816    SKIP 1 OUTPUT LINE
7817    ''THE'' END ''OF HEADING SECTION                                                \1
7818    LOOP FOR KILLER=1 TO KILNUM
7819    DO ''                                                                           \1
7820       PRINT 1 DOUBLE LINE WITH
7821       EQ.NAME(KV.EQ.ID(KSIDE,KILLER)),
7822       KV.INITIAL.DENSITY(KSIDE,KILLER),AND
7823       A GROUP OF KV.SCORE(KSIDE,KILLER,VICTIM) FIELDS THUS
7824       ******  ******  ****** ****** ****** ****** ****** ****** ****** ****** ******
7825       ****** ****** ****** ****** ****** ****** ******
7826       PRINT 1 LINE WITH KV.CEM.WPN.NO(KSIDE,KILLER) THUS
7827       (***)
7828    ENDLOOP ''FOR PRINT OF EACH KILLER VICTIM LINE
7829    ''PUT TOTALS PRINT STATEMENT HERE
7830    SKIP 1 OUTPUT LINE
7831    PRINT 1 DOUBLE LINE WITH
7832    A GROUP OF KILL.TOTAL(VICTIM) FIELDS THUS
7833    ==TOTAL KILLS==  ****** ****** ****** ****** ****** ****** ****** ****** ******
7834    ****** ****** ****** ****** ****** ****** ******
7835    SKIP 1 OUTPUT LINE
7836    PRINT 1 DOUBLE LINE WITH A GROUP OF PERCENT.LOSS(VICTIM) FIELDS THUS
7837    ==PERCENT LOSS=  ***.*% ***.*% ***.*% ***.*% ***.*% ***.*% ***.*% ***.*% ***.*%
7838    ***.*% ***.*% ***.*% ***.*% ***.*% ***.*% ***.*% ***.*%
7839    ''THE'' END ''OF REPORT SECTION                                                 \1
7840    RELEASE KILL.TOTAL(*),PERCENT.LOSS(*)
7841 ENDROUTINE
```

```
7842 ROUTINE KV.SCOREBOARD ''                                                                    0007
7843    ADD 1 TO ANAL.CTR(232,1)  ''                                                            \DYN_ANAL
7844    NORMALLY MODE IS INTEGER AND DIMENSION=0
7845    DEFINE BLUE.N.KV, RED.N.KV, KV, KILLER.VICTIM,
7846    SIDE AS INTEGER VARIABLES
7847    LET BLUE.N.KV = DIM.F(KV.CEM.WPN.NO(BLUE,*))
7848    LET RED.N.KV = DIM.F(KV.CEM.WPN.NO(RED,*))
7849    CALL KV.PRINT GIVEN------------------------------------------------------------------------>(606)
7850    BLUE,RED,BLUE.N.KV,RED.N.KV  ''BLUE KILLER BOARD
7851    CALL KV.PRINT GIVEN------------------------------------------------------------------------>(606)
7852    RED,BLUE,RED.N.KV,BLUE.N.KV  ''RED KILLER BOARD
7853
7854    ''AMMO CONSUMPTION
7855
7856    ''CALL AMMO.RPT ------------------------------------------------------- CHG\31 COMMENTED OUT FOR \DEBUG>(592)
7857    WRITE AS /,/,B 12, " KILLER VICTIM AMMO IN SHORT TONS",/,/
7858    WRITE DIM.F(KV.CEM.WPN.NO(2,*)),DIM.F(KV.CEM.WPN.NO(1,*)) AS 2 I 5,
7859    / USING UNIT 11
7860    LOOP FOR SIDE BACK FROM BLUE TO RED,
7861    DO ''                                                                                    \1
7862       LET N.KILLER.VICTIM = DIM.F(KV.CEM.WPN.NO(SIDE,*))
7863       LOOP FOR EACH KILLER.VICTIM CALLED KV,
7864       DO  ''                                                                                \1
7865          WRITE SIDE AS S 2, I 3
7866          WRITE KV.CEM.WPN.NO(SIDE,KV) AS I 5
7867          WRITE EQ.NAME(KV.EQ.ID(SIDE,KV)) AS S 1, T 6 ''                                    \TEXT
7868          WRITE KV.INITIAL.DENSITY(SIDE,KV) AS I 10
7869          WRITE KV.AMMO.CONSUMED(SIDE,KV)/(2000 * 100) AS D(18,5) '' SHORT TONS
7870          WRITE AS /
7871          WRITE SIDE AS S 2, I 3 USING UNIT 11
7872          WRITE KV.CEM.WPN.NO(SIDE,KV) AS I 5 USING UNIT 11
7873          WRITE EQ.NAME(KV.EQ.ID(SIDE,KV)) AS S 1, T 6 USING UNIT 11 ''                      \TEXT
7874          WRITE KV.INITIAL.DENSITY(SIDE,KV) AS I 10 USING UNIT 11
7875          WRITE KV.AMMO.CONSUMED(SIDE,KV)/(2000. * 100.) AS D(18,5) USING UNIT 11
7876          WRITE AS / USING UNIT 11
7877       ENDLOOP
7878    ENDLOOP
7879    ''
7880    ''
7881    ''THE NEXT SECTION OF CODE COUNTS AND REPORTS THE NUMBERS OF
7882    ''EACH EQUIPMENT TYPE THAT ARE ENGAGED IN BATTLE, AND HOW MANY
7883    ''OF THEM ARE LOST
7884    ''
7885    ''
7886    IF ANALYSIS(3) = TRUE
7887       DEFINE KV.COUNTER AS A 2-DIMENSIONAL ARRAY
7888       RESERVE KV.COUNTER(*,*) AS N.EQUIPMENT BY 2
7889       REWIND 50
7890       USE UNIT 50 FOR INPUT
7891       SKIP 1 FIELD
7892       FOR EACH UNIT,
7893       DO  ''                                                                                \1
7894          READ UNIT.NUMBER
7895          IF UNIT.NUMBER NE UNIT.NOS(UNIT)
7896             TRACE
7897 <----------STOP
7898          OTHERWISE
7899          IF UN.ENGAGED.INDEX(UNIT) NE 1
```

```
7900            SKIP 1 INPUT CARD
7901         <------CYCLE
7902            OTHERWISE
7903            SKIP 1 FIELD
7904            FOR EACH EQ IN THE UN.EQUIP.LIST(UNIT),
7905            DO  ''                                                                    \1
7906               SKIP 1 FIELD
7907               READ EQ.QUANT
7908               ADD EQ.QUANT TO KV.COUNTER(UE.ID(EQ),1)
7909               ADD UE.QUANT(EQ) TO KV.COUNTER(UE.ID(EQ),2)
7910            ENDLOOP ''
7911         ENDLOOP ''                                          REPEAT CHANGED TO ENDLOOP \1
7912         USE UNIT 5 FOR INPUT                                REPEAT CHANGED TO ENDLOOP \1
7913         BEGIN REPORT ON A NEW PAGE
7914         BEGIN HEADING
7915         PRINT 5 LINES AS FOLLOWS
7916
7917         ENGAGED EQUIPMENT
7918
7919         EQUIPMENT     STARTING DENSITY   ENDING DENSITY
7920
7921         ''THE'' END ''OF THE HEADING                                                  \1
7922         FOR EACH EQUIPMENT CALLED EQ
7923         PRINT 1 LINE WITH EQ.NAME(EQ), KV.COUNTER(EQ,1),
7924         KV.COUNTER(EQ,2) THUS
7925         ******            ****              ****
7926         ''THE'' END ''OF THE  REPORT
7927         RELEASE KV.COUNTER                                                            \1
7928      ALWAYS
7929   ENDROUTINE
```

```
7930 ROUTINE OUTPUT.ATTRITION GIVEN UNIT,KILLER,VICTIM,N.KILLED,FIRE,MSN  ''                    0008
7931    ADD 1 TO ANAL.CTR(233,1)  ''                                                          \DYN_ANAL
7932    DEFINE FIRE, ROUND.NAME AND SIDE.ALF AND MOVE.STATUS AS TEXT VARIABLES ''              \TEXT
7933    IF TU.SIDE(UN.TYPE.UNIT(UNIT)) = 1
7934       LET SIDE.ALF = "RED"
7935    ELSE LET SIDE.ALF = "BLUE"
7936    ALWAYS
7937    IF N.KILLED NE 0
7938       USE UNIT 60 FOR OUTPUT
7939       WRITE UN.BATTLE.INDEX(UNIT),TIME.V,KILLER,VICTIM,N.KILLED,FIRE
7940       AS "KLLD  ",I 7,S 2,D(7,2),S 2,I 4,S 2,I 4,S 2,I 5,S 2,T 6,/ ''                     \TEXT
7941       FOR EVERY UEL IN UN.EQUIP.LIST(UNIT), WITH UE.ID(UEL)=VICTIM,
7942       FIND THE FIRST CASE
7943       IF FOUND,
7944          FOR EVERY WEAPON IN UE.WEAPON.SET(UEL),
7945          WRITE WPN.ID(WEAPON),WPN.QUANTITY(WEAPON)*N.KILLED
7946          AS"WKLL",S 20,I 5,S 2,I 5,/
7947       REGARDLESS
7948       FOR EVERY BTRY, WITH BY.UNIT(BTRY)=UNIT, FIND THE FIRST CASE
7949       IF FOUND,
7950       THEN IF TB.HOW.EQ.ID(BY.TYPE(BTRY))=VICTIM,
7951          WRITE BY.TYPE(BTRY),N.KILLED AS "HKLL",S 20,I 5,S 2,I 5,/
7952       REGARDLESS
7953    ALWAYS ''                                                                             \1
7954    IF FIRE = "INDIRECT"
7955       USE UNIT 2 FOR OUTPUT
7956       FOR EACH .UEL IN UN.EQUIP.LIST(TR.TGT.UNIT(FM.TGT(MSN))) WITH
7957       UE.ID(.UEL) = VICTIM
7958       FIND THE FIRST CASE
7959       IF NONE
7960          TRACE
7961 <--------STOP ''                                                                         \1
7962       OTHERWISE
7963       IF TR.PGM.STATUS(FM.TGT(MSN)) = 1 LET ROUND.NAME = "XXXXXX"
7964       ELSE IF TR.PGM.STATUS(FM.TGT(MSN)) = 2 LET ROUND.NAME = "YYYYYY"
7965       ELSE IF FM.TM.CLASS(MSN) = "HE" LET ROUND.NAME = HE.ID(FM.TM(MSN))
7966       ELSE IF FM.TM.CLASS(MSN) = "ICM" LET ROUND.NAME = IC.ID(FM.TM(MSN))
7967       ALWAYS ALWAYS ALWAYS ALWAYS
7968       LET FIRED.RNDS = (FM.FIRED.VOLS(MSN)*N.BY.HOW.SET(FM.BTRY(MSN)) *
7969       TB.RND.PER.LAUNCH(BY.TYPE(FM.BTRY(MSN))))
7970       LET .BTRY = FM.BTRY(MSN)
7971       IF EVENT.V = I.ARTY.ASSESS
7972          LET IENG = 1
7973       ELSE
7974          LET IENG = 0
7975       ALWAYS
7976       LET .UNT = TR.TGT.UNIT(FM.TGT(MSN))
7977       IF UN.STATUS(.UNT) = ADVANCING OR UN.STATUS(.UNT) = WITHDRAWING
7978          LET MOVE.STATUS = "M"
7979       ELSE
7980          LET MOVE.STATUS = "S"
7981       ALWAYS
7982       IF FIRED.RNDS > 0
7983          PRINT 1 LINE WITH MSN,FIRED.RNDS,EQ.NAME(KILLER),TB.NAME(BY.TYPE(.BTRY)),
7984          ROUND.NAME, (N.KILLED*UE.QUANT(.UEL)), EQ.NAME(VICTIM), N.KILLED,
7985          SIDE.ALF,TU.LEVEL(UN.TYPE.UNIT(.UNT)),(FM.RANGE(MSN)*10),
7986          MOVE.STATUS, IENG  THUS
7987          ******* **** ****** ****** ****** *** ****** *** **** ****** ***** * *
```

```
7988        REGARDLESS
7989     ALWAYS  ''                                                    \1
7990     USE UNIT 6 FOR OUTPUT
7991 <--RETURN
7992 END
```

```
7993 ROUTINE FOR POSITION.OUT                                                    0009
7994
7995    ADD 1 TO ANAL.CTR(234,1)  ''                                             \DYN_ANAL
7996    NORMALLY MODE IS INTEGER
7997    DEFINE UNIT AS AN INTEGER VARIABLE
7998
7999    USE UNIT 3 FOR OUTPUT
8000
8001    WRITE N.UNIT AND TIME.V AS I 4, D(13,5), /
8002    LOOP FOR EACH UNIT
8003    DO
8004       WRITE UNIT.NOS(UNIT) AND UN.TYPE.UNIT(UNIT) AS 2 I 6
8005       WRITE UN.X.COORD(UNIT) AND UN.Y.COORD(UNIT) AS 2 I 6
8006       IF UN.PARENT(UNIT) = 0
8007          LET PARENT = 0
8008       ELSE
8009          LET PARENT = UNIT.NOS(UN.PARENT(UNIT))
8010       ALWAYS
8011       WRITE PARENT AND UN.COLOR(UNIT) AS I 6, I 2
8012       WRITE TU.CAT(UN.TYPE.UNIT(UNIT)) AS I 10
8013       WRITE AS /
8014    ENDLOOP
8015
8016    USE UNIT 6 FOR OUTPUT
8017
8018 <--EXITROUTINE
8019 ENDROUTINE
```

```
8020 ROUTINE SNAP2                                                    ''                    0010
8021
8022     ADD 1 TO ANAL.CTR(235,1)  ''                                                       \DYN_ANAL
8023     ''SNAP2  CALLED FROM SNAP$R  TO GET THEM TO COMPILE WITH VER 6.2
8024
8025     NORMALLY MODE IS INTEGER
8026
8027     PRINT 56 LINES WITH
8028     TIME.V,
8029     EVENT.V,
8030     I.AC.ATK.TGT,
8031     I.AIRBORNE.RADAR,
8032     I.AIR.OBSERVER,
8033     I.ARTY.ASSESS,
8034     I.ASSESSMENT,
8035     I.CAS.MISSION,
8036     I.FIRE.MISSION,
8037     I.FORWARD.OBSERVER,
8038     I.HC.ARRIVE.BATTLE,
8039     I.HC.RETURN.FARRP,
8040     I.HELICOPTER.FIRE,
8041     I.HEL.TARGET.ACQUISITION,
8042     I.HOW.REPAIR,
8043     I.MINE.ASSESS,
8044     I.PHOTO.IR.FLIGHT,
8045     I.REMOTE.PILOT.VEHICLE,
8046     I.SHOOT.OUT,
8047     I.TARGET.REPORT,
8048     I.WITH.DRAW,
8049     I.CHANGE.LITE,
8050     I.END.SIMULATION,
8051     I.ACT.ATK,
8052     I.ACT.DEF,
8053     I.ACT.MOVCOR,
8054     I.ACT.MOVDIS,
8055     I.ACT.REINF,
8056     I.AD.ENGAGEMENT,
8057     I.ARTY.OCCUPATION,
8058     I.BTL.ENDED,
8059     I.CFR.ACTIVATION,
8060     I.CFR.OFF,
8061     I.CFR.ON,
8062     I.CFR.OPERATOR,
8063     I.CHANGE.WEATHER,
8064     I.DQ.OLD.SORTIE.QUEUE,
8065     I.ENGAGEMENT,
8066     I.FEBA.SORTIE,
8067     I.GET.NX.ORD,
8068     I.HC.DEPART.BATTLE,
8069     I.HELO.ENGAGEMENT,
8070     I.INIT.PREPLAN.CAS,
8071     I.MOVE,
8072     I.PDB.ACTIVATION,
8073     I.PDB.OPERATOR,
8074     I.SCHEDULE.ARTY.MOVEMENT,
8075     I.SEND.TEAM,
8076     I.START.ARTY.MOVEMENT,
8077     I.START.BATTLE,
```

```
8078    I.START.MOVE,
8079    I.STOP.ARTY.MOVEMENT,
8080    I.UPDATE.LOC,
8081    I.POSITION.REPORT,
8082    I.SET.DEBUG,
8083    I.OFF.LINE.ATTRITION
8084    THUS
8085    TIME.V = **.********
8086    EVENT.V = ******
8087    I.AC.ATK.TGT = ******
8088    I.AIRBORNE.RADAR = ******
8089    I.AIR.OBSERVER = ******
8090    I.ARTY.ASSESS = ******
8091    I.ASSESSMENT = ******
8092    I.CAS.MISSION = ******
8093    I.FIRE.MISSION = ******
8094    I.FORWARD.OBSERVER = ******
8095    I.HC.ARRIVE.BATTLE = ******
8096    I.HC.RETURN.FARRP = ******
8097    I.HELICOPTER.FIRE = ******
8098    I.HEL.TARGET.ACQUISITION = ******
8099    I.HOW.REPAIR = ******
8100    I.MINE.ASSESS = ******
8101    I.PHOTO.IR.FLIGHT = ******
8102    I.REMOTE.PILOT.VEHICLE = ******
8103    I.SHOOT.OUT = ******
8104    I.TARGET.REPORT = ******
8105    I.WITH.DRAW = ******
8106    I.CHANGE.LITE = ******
8107    I.END.SIMULATION = ******
8108    I.ACT.ATK = ******
8109    I.ACT.DEF = ******
8110    I.ACT.MOVCOR = ******
8111    I.ACT.MOVDIS = ******
8112    I.ACT.REINF = ******
8113    I.AD.ENGAGEMENT = ******
8114    I.ARTY.OCCUPATION = ******
8115    I.BTL.ENDED = ******
8116    I.CFR.ACTIVATION = ******
8117    I.CFR.OFF = ******
8118    I.CFR.ON = ******
8119    I.CFR.OPERATOR = ******
8120    I.CHANGE.WEATHER = ******
8121    I.DQ.OLD.SORTIE.QUEUE = ******
8122    I.ENGAGEMENT = ******
8123    I.FEBA.SORTIE = ******
8124    I.GET.NX.ORD = ******
8125    I.HC.DEPART.BATTLE = ******
8126    I.HELO.ENGAGEMENT = ******
8127    I.INIT.PREPLAN.CAS = ******
8128    I.MOVE = ******
8129    I.PDB.ACTIVATION = ******
8130    I.PDB.OPERATOR = ******
8131    I.SCHEDULE.ARTY.MOVEMENT = ******
8132    I.SEND.TEAM = ******
8133    I.START.ARTY.MOVEMENT = ******
8134    I.START.BATTLE = ******
8135    I.START.MOVE = ******
```

```
8136    I.STOP.ARTY.MOVEMENT = ******
8137    I.UPDATE.LOC = ******
8138    I.POSITION.REPORT = ******
8139    I.SET.DEBUG = ******
8140    I.OFF.LINE.ATTRITION = ********
8141    LIST ATTRIBUTES OF EACH AC.ATK.TGT IN EV.S(I.AC.ATK.TGT)
8142    ''LIST ATTRIBUTES OF EACH AIRBORNE.RADAR IN EV.S(I.AIRBORNE.RADAR)
8143    LIST ATTRIBUTES OF EACH AIR.OBSERVER IN EV.S(I.AIR.OBSERVER)
8144    LIST ATTRIBUTES OF EACH ARTY.ASSESS IN EV.S(I.ARTY.ASSESS)
8145    LIST ATTRIBUTES OF EACH ASSESSMENT IN EV.S(I.ASSESSMENT)
8146    LIST ATTRIBUTES OF EACH CAS.MISSION IN EV.S(I.CAS.MISSION)
8147    ''LIST ATTRIBUTES OF EACH FIRE.MISSION IN EV.S(I.FIRE.MISSION)
8148    ''LIST ATTRIBUTES OF EACH FORWARD.OBSERVER IN EV.S(I.FORWARD.OBSERVER)
8149    LIST ATTRIBUTES OF EACH HC.ARRIVE.BATTLE IN EV.S(I.HC.ARRIVE.BATTLE)
8150    LIST ATTRIBUTES OF EACH HC.RETURN.FARRP IN EV.S(I.HC.RETURN.FARRP)
8151    LIST ATTRIBUTES OF EACH HELICOPTER.FIRE IN EV.S(I.HELICOPTER.FIRE)
8152    LIST ATTRIBUTES OF EACH HEL.TARGET.ACQUISITION
8153    IN EV.S(I.HEL.TARGET.ACQUISITION)
8154    LIST ATTRIBUTES OF EACH HOW.REPAIR IN EV.S(I.HOW.REPAIR)
8155    ''LIST ATTRIBUTES OF EACH PHOTO.IR.FLIGHT IN EV.S(I.PHOTO.IR.FLIGHT)
8156    ''LIST ATTRIBUTES OF EACH REMOTE.PILOT.VEHICLE
8157    ''      IN EV.S(I.REMOTE.PILOT.VEHICLE)
8158    LIST ATTRIBUTES OF EACH SHOOT.OUT IN EV.S(I.SHOOT.OUT)
8159    ''LIST ATTRIBUTES OF EACH TARGET.REPORT IN EV.S(I.TARGET.REPORT)
8160    LIST ATTRIBUTES OF EACH WITH.DRAW IN EV.S(I.WITH.DRAW)
8161    LIST ATTRIBUTES OF EACH CHANGE.LITE IN EV.S(I.CHANGE.LITE)
8162    LIST ATTRIBUTES OF EACH END.SIMULATION IN EV.S(I.END.SIMULATION)
8163    LIST ATTRIBUTES OF EACH ACT.ATK IN EV.S(I.ACT.ATK)
8164    LIST ATTRIBUTES OF EACH ACT.DEF IN EV.S(I.ACT.DEF)
8165    LIST ATTRIBUTES OF EACH ACT.MOVCOR IN EV.S(I.ACT.MOVCOR)
8166    LIST ATTRIBUTES OF EACH ACT.MOVDIS IN EV.S(I.ACT.MOVDIS)
8167    LIST ATTRIBUTES OF EACH ACT.REINF  IN EV.S(I.ACT.REINF)
8168    LIST ATTRIBUTES OF EACH AD.ENGAGEMENT IN EV.S(I.AD.ENGAGEMENT)
8169    LIST ATTRIBUTES OF EACH ARTY.OCCUPATION IN EV.S(I.ARTY.OCCUPATION)
8170    LIST ATTRIBUTES OF EACH BTL.ENDED IN EV.S(I.BTL.ENDED)
8171    LIST ATTRIBUTES OF EACH CFR.ACTIVATION IN EV.S(I.CFR.ACTIVATION)
8172    LIST ATTRIBUTES OF EACH CFR.OFF IN EV.S(I.CFR.OFF)
8173    LIST ATTRIBUTES OF EACH CFR.ON IN EV.S(I.CFR.ON)
8174    ''LIST ATTRIBUTES OF EACH CFR.OPERATOR IN EV.S(I.CFR.OPERATOR)
8175    LIST ATTRIBUTES OF EACH DQ.OLD.SORTIE.QUEUE IN
8176    EV.S(1.DQ.OLD.SORTIE.QUEUE)
8177    LIST ATTRIBUTES OF EACH ENGAGEMENT IN EV.S(I.ENGAGEMENT)
8178    LIST ATTRIBUTES OF EACH FEBA.SORTIE IN EV.S(I.FEBA.SORTIE)
8179    LIST ATTRIBUTES OF EACH GET.NX.ORD IN EV.S(I.GET.NX.ORD)
8180    LIST ATTRIBUTES OF EACH HC.DEPART.BATTLE IN EV.S(I.HC.DEPART.BATTLE)
8181    LIST ATTRIBUTES OF EACH HELO.ENGAGEMENT IN EV.S(I.HELO.ENGAGEMENT)
8182    LIST ATTRIBUTES OF EACH INIT.PREPLAN.CAS IN EV.S(I.INIT.PREPLAN.CAS)
8183    LIST ATTRIBUTES OF EACH MOVE IN EV.S(I.MOVE)
8184    LIST ATTRIBUTES OF EACH PDB.ACTIVATION IN EV.S(I.PDB.ACTIVATION)
8185    ''LIST ATTRIBUTES OF EACH PDB.OPERATOR IN EV.S(I.PDB.OPERATOR)
8186    LIST ATTRIBUTES OF EACH SCHEDULE.ARTY.MOVEMENT
8187    IN EV.S(I.SCHEDULE.ARTY.MOVEMENT)
8188    LIST ATTRIBUTES OF EACH SEND.TEAM IN EV.S(I.SEND.TEAM)
8189    LIST ATTRIBUTES OF EACH START.ARTY.MOVEMENT
8190    IN EV.S(I.START.ARTY.MOVEMENT)
8191    LIST ATTRIBUTES OF EACH STOP.ARTY.MOVEMENT
8192    IN EV.S(I.STOP.ARTY.MOVEMENT)
8193    LIST ATTRIBUTES OF EACH START.BATTLE IN EV.S(I.START.BATTLE)
```

```
8194     LIST ATTRIBUTES OF EACH START.MOVE IN EV.S(I.START.MOVE)
8195     LIST ATTRIBUTES OF EACH UPDATE.LOC IN EV.S(I.UPDATE.LOC)
8196     LIST ATTRIBUTES OF EACH POSITION.REPORT IN EV.S(I.POSITION.REPORT)
8197     ''LIST ATTRIBUTES OF EACH SET.DEBUG IN EV.S(I.SET.DEBUG)                    \VAX
8198
8199 <--EXITROUTINE
8200 ENDROUTINE
```

```
8201 ROUTINE SNAP.R                                                    ''                                                    0011
8202
8203    ADD 1 TO ANAL.CTR(236,1)  ''                                                                                  \DYN_ANAL
8204    NORMALLY MODE IS INTEGER
8205    USE 20 FOR OUTPUT
8206
8207    LIST  TIME.V'', ZTIME.F, USED.DBANK.V, MAX.DBANK.V                                                            \VAX
8208
8209    CALL SNAP2-------------------------------------------------------------------------------------------------->(614)
8210
8211    ''LIST ATTRIBUTES OF EACH AO.ELEVATION.BAND
8212    ''LIST ATTRIBUTES OF EACH AO.RANGE.BAND
8213    LIST ATTRIBUTES OF EACH AC.TYPE
8214    ''LIST ATTRIBUTES OF EACH BTRY
8215    LIST ATTRIBUTES OF EACH CATEGORY
8216    ''LIST ATTRIBUTES OF EACH CFR.RNG.HACK
8217    ''LIST ATTRIBUTES OF EACH ENVIRONMENT
8218    LIST ATTRIBUTES OF EACH EQUIPMENT
8219    ''LIST ATTRIBUTES OF EACH FASCAM.MUNITION
8220    LIST ATTRIBUTES OF EACH FA.BN
8221    LIST ATTRIBUTES OF EACH FDC
8222    ''LIST ATTRIBUTES OF EACH FO.RANGE.BAND
8223    ''LIST ATTRIBUTES OF EACH FUZE
8224    ''LIST ATTRIBUTES OF EACH GROUPING
8225    ''LIST ATTRIBUTES OF EACH HE.MUNITION
8226    ''LIST ATTRIBUTES OF EACH HE.RANGE.HACK
8227    ''LIST ATTRIBUTES OF EACH IC.MUNITION
8228    ''LIST ATTRIBUTES OF EACH IC.RANGE.HACK
8229    ''LIST ATTRIBUTES OF EACH ILLUM.MUNITION
8230    ''LIST ATTRIBUTES OF EACH MISSION
8231    ''LIST ATTRIBUTES OF EACH MODEL.AO
8232    ''LIST ATTRIBUTES OF EACH MODEL.AR
8233    ''LIST ATTRIBUTES OF EACH MODEL.CFR
8234    ''LIST ATTRIBUTES OF EACH MODEL.FO
8235    ''LIST ATTRIBUTES OF EACH MODEL.PDB
8236    ''LIST ATTRIBUTES OF EACH MODEL.PIR
8237    ''LIST ATTRIBUTES OF EACH MODEL.RPV
8238    ''LIST ATTRIBUTES OF EACH PDB.RNG.HACK
8239    ''LIST ATTRIBUTES OF EACH POSTURE
8240    ''LIST ATTRIBUTES OF EACH SEARCH.POINT
8241    ''LIST ATTRIBUTES OF EACH SECTOR
8242    ''LIST ATTRIBUTES OF EACH SENSOR.TYPE
8243    ''LIST ATTRIBUTES OF EACH SIDE
8244    ''LIST ATTRIBUTES OF EACH SMOKE.MUNITION
8245    ''LIST ATTRIBUTES OF EACH SUBMUNITION
8246    ''LIST ATTRIBUTES OF EACH TERRAIN.TYPE
8247    ''LIST ATTRIBUTES OF EACH TYPE.BATTLE.FIELD
8248    ''LIST ATTRIBUTES OF EACH TYPE.BTRY
8249    LIST ATTRIBUTES OF EACH TYPE.EQUIPMENT
8250    FOR EACH SIDE
8251    LIST ATTRIBUTES OF EACH CAS.MISSION IN SD.CMSN.QUEUE(SIDE)
8252    FOR EACH SIDE
8253    LIST ATTRIBUTES OF EACH OLD.SORTIE IN SD.OLD.SORTIE.QUEUE(SIDE)
8254    LIST ATTRIBUTES OF EACH TYPE.UNIT
8255    LIST ATTRIBUTES OF EACH TYPE.WEAPON
8256    LIST ATTRIBUTES OF EACH UNIT
8257    ''    LOOP FOR EVERY BATTLE OF BATTLE.SET
8258    ''          DO THE FOLLOWING
```

```
8259        ''          LIST ATTRIBUTES OF BATTLE
8260        ''          LOOP FOR EVERY FORCE OF BTL.FORCE.SET(BATTLE)
8261        ''               DO THE FOLLOWING
8262            ''              LIST ATTRIBUTES OF FORCE
8263            ''              LOOP FOR EVERY UNIT OF FR.UNIT.SET(FORCE)
8264            ''                   DO THE FOLLOWING
8265                ''                  LIST ATTRIBUTES OF EACH UE.LINK. IN UN.EQUIP.LIST(UNIT.)
8266                ''                  LIST ATTRIBUTES OF EACH POINT IN UN.PATH(UNIT.)
8267                ''                  LIST ATTRIBUTES OF EACH VISIBLE.UNIT IN UN.LOS.LIST(UNIT.)
8268                ''                  LIST ATTRIBUTES OF EACH SEGMENT IN UN.SEGMENT.LIST(UNIT.)
8269                ''                  LOOP FOR EVERY UE.LINK IN UN.EQUIP.LIST(UNIT.)
8270                ''                       DO THE FOLLOWING
8271                    ''                      LIST ATTRIBUTES OF EACH FIRING.TABLE
8272                    ''                           IN UE.TARGET.LIST(UE.LINK)
8273                    ''                      LIST ATTRIBUTES OF EACH SHOOT.OUT
8274                    ''                           IN SO.LIST(UE.LINK)
8275                ''                       ENDLOOP
8276            ''                   ENDLOOP
8277        ''              ENDLOOP
8278    ''      ENDLOOP
8279    ''      FOR EACH FDC
8280    ''          LIST ATTRIBUTES OF EACH TARGET.REPORT IN FD.TR.QUEUE(FDC)
8281    ''      LOOP FOR EACH UNIT CALLED UN
8282    ''           DO
8283        ''          LOOP FOR EACH US.LINK IN UN.SENSOR.LIST(UN)
8284        ''               DO
8285            ''              LIST ATTRIBUTES OF US.LINK
8286            ''              IF ST.NAME(US.SENSOR.TYPE(US.LINK)) = "FO"
8287                ''                  LIST ATTRIBUTES OF FORWARD.OBSERVER CALLED US.ID(US.LINK)
8288                ''                  LIST ATTRIBUTES OF EACH FO.DET.CANDIDATE
8289                ''                       IN FO.CAND.DET.LIST(US.ID(US.LINK))
8290            ''              ELSE
8291                ''                  IF ST.NAME(US.SENSOR.TYPE(US.LINK)) = "CB" OR
8292                    ''                      ST.NAME(US.SENSOR.TYPE(US.LINK)) = "CM"
8293                    ''                      LIST ATTRIBUTES OF CF.RADAR CALLED US.ID(US.LINK)
8294                ''                  ELSE
8295                    ''                  IF ST.NAME(US.SENSOR.TYPE(US.LINK)) = "SD" OR
8296                        ''                      ST.NAME(US.SENSOR.TYPE(US.LINK)) = "FL"
8297                        ''                      LIST ATTRIBUTES OF PASSIVE.DETECTION.BASE
8298                        ''                           CALLED US.ID(US.LINK)
8299                    ''                  ALWAYS
8300                ''                  ALWAYS
8301            ''              ALWAYS
8302        ''          ENDLOOP
8303    ''      ENDLOOP
8304    ''
8305    ''      LOOP FOR EACH BTRY
8306    ''           DO
8307        ''          LIST ATTRIBUTES OF BTRY
8308        ''          LIST ATTRIBUTES OF EACH FIRE.MISSION IN BY.FM.QUEUE(BTRY)
8309    ''      ENDLOOP
8310
8311    DEFINE I, J, CELLS, SAVE.OUTPUT.UNIT AS AN INTEGER VARIABLES ''                    \DYN_ANAL
8312
8313    LET SAVE.OUTPUT.UNIT = WRITE.V
8314    USE UNIT 80 FOR OUTPUT
8315    ''START NEW PAGE
8316
```

```
8317     WRITE TIME.V AS "TIME.V= ", D(8,5), /
8318     FOR I = 1 TO 264
8319     DO
8320        WRITE ANAL.TEXT(I) AS B 2, T *, B 45
8321        LET CELLS = DIM.F(ANAL.CTR(I, *))    ''NUMBER OF ROWS
8322        FOR J = 1 TO CELLS
8323        WRITE ANAL.CTR(I,J) AS I 8, S 2
8324        WRITE AS /
8325        FOR J = 1 TO CELLS
8326        LET ANAL.CTR(I,J) = 0
8327     ENDLOOP
8328
8329     USE SAVE.OUTPUT.UNIT FOR OUTPUT
8330
8331 <--EXITROUTINE
8332 ENDROUTINE
```

```
8333 ROUTINE TACAIR.DATA.REPORT                    ''                                   0012
8334
8335    ADD 1 TO ANAL.CTR(237,1)  ''                                                    \DYN_ANAL
8336    NORMALLY MODE IS INTEGER
8337
8338
8339    IF TACAIR.DEBUG NE 2
8340 <------RETURN
8341    OTHERWISE
8342
8343    DEFINE .EQ.NAME AS A TEXT VARIABLE ''                                           \TEXT
8344
8345    ''WRITE THE SIDE DEPENDENT DATA
8346    START NEW PAGE
8347    PRINT 14 LINES WITH
8348    SD.AIRFIELD (BLUE),       SD.AIRFIELD(RED),
8349    SD.MAX.SORTIE.TP(BLUE), SD.MAX.SORTIE.TP(RED),
8350    SD.TP.SORTIE(BLUE),    SD.TP.SORTIE(RED),
8351    SD.ASC.MAX.SORTIE(BLUE),    SD.ASC.MAX.SORTIE(RED),
8352    SD.ASC.RADIUS(BLUE)*16,  SD.ASC.RADIUS(RED)*16,
8353    SD.NO.FLY.VIS(BLUE)*16,  SD.NO.FLY.VIS(RED)*16,
8354    SD.POOR.FLY.VIS(BLUE)*16,  SD.POOR.FLY.VIS(RED)*16
8355    THUS
8356    TACAIR DATA REPORT
8357
8358    BLUE     RED
8359    AIRFIELD                 *****   *****
8360    MAX SORTIES
8361    PER TIME PERIOD        *****   *****
8362    TIME PERIOD              *****   *****  MINUTES
8363    AIRSPACE CONSTRAINT
8364    MAX SORTIES            ***   ***
8365    AIRSPACE CONSTRAINT
8366    RADIUS                 *****   *****  METERS
8367    NO   FLY VISIBILITY    *****   *****  METERS
8368    POOR FLY VISIBILITY    *****   *****  METERS
8369
8370    SKIP 4 LINES
8371    PRINT 3 LINES THUS
8372    TANK   TANK  RATIO  MIN
8373    SIDE  MISSION   NAME    MIN   MAX   CEQ
8374    ----  -------  ------  ----------  ---
8375
8376    LOOP FOR  .S BACK FROM BLUE TO RED
8377    DO
8378       IF .S = RED
8379          SKIP 1 LINE
8380       ALWAYS
8381       WRITE SHADE(.S) AS B 16, T 4 ''                                              \TEXT
8382       LOOP FOR EACH MISSION CALLED .M
8383       DO
8384          WRITE MN.NAME(.M),
8385          TE.NAME(SM.TANK.TE(.S,.M)),
8386          SM.MIN.TANK.RATIO(.S,.M),
8387          SM.MAX.TANK.RATIO(.S,.M),
8388          SM.MIN.CEQ(.S,.M) AS
8389          B 22, T 6, B 31, T 6, B 38, 2 D(6,2), ''                                  \TEXT
8390          B 52, I 3, /
```

```
8391        ENDLOOP
8392     ENDLOOP
8393
8394     ''WRITE THE AC TYPE AND PREPLANNED MISSION DATA.
8395     START NEW PAGE
8396     WRITE AS B 54, "AIRCRAFT  TYPES", /, /,
8397     B 35, "RANGES ARE IN METERS, TIMES ARE IN MINUTES", /, /
8398     WRITE AS B 12, "SUBST. WEATHER  MIN  NORM  BAI TA",
8399     "  MAX  PROB   PREP    SPEED  PASS DIST TO TGT FROM   ",
8400     "POINTS IN ATTACK PATH", /
8401     WRITE AS "NUM  NAME    AC   DEGRADE ALTIT ALTIT ",
8402     "DELAY  ALOFT ABORT MIN MAX (M/SEC) TIME",
8403     B 81, "P1", B 87, "P2", B 93, "P3", B 98, "ID", B 104, "X",
8404     B 110, "Y", B 116, "Z"
8405     WRITE AS "--- ------ ------ ------- ----- ----- ",
8406     "------ ---- ---- ------- ------- --- ",
8407     "----- ---- ---- -- ---- ---- ----",/
8408     LOOP FOR EACH AC.TYPE CALLED .AC
8409     DO
8410        IF ACT.SUBSTITUTE(.AC) = 0
8411           LET .EQ.NAME = "NONE"
8412        ELSE
8413           LET .EQ.NAME = EQ.NAME(ACT.EQUIP.ID(ACT.SUBSTITUTE(.AC)))
8414        ALWAYS
8415        WRITE .AC,
8416        EQ.NAME(ACT.EQUIP.ID(.AC)),
8417        .EQ.NAME,
8418        ACT.WEATHER.DEGRADE(.AC),
8419        ACT.MIN.ALT(.AC)*16,
8420        ACT.NORM.ALT(.AC)*16,
8421        ACT.BAI.TA.DELAY(.AC),
8422        ACT.MAX.ALOFT(.AC),
8423        ACT.PROB.SORTIE.ABORT(.AC),
8424        ACT.MIN.PREP.TIME(.AC),
8425        ACT.MAX.PREP.TIME(.AC),
8426        ACT.SPEED(.AC)*16,
8427        ACT.PASS.TIME(.AC),
8428        ACT.P1.DIST(.AC)*16,
8429        ACT.P2.DIST(.AC)*16,
8430        ACT.P3.DIST(.AC)*16
8431        AS I 3, S 1, T 6, S 1, T 6, S 3, I 3, S 2, ''                    \TEXT
8432        2 I 6, I 5, S 3, I 4, S 3, I 3, S 1, 2 I 4,
8433        B 67, I 4, B 74, I 4, 3 I 6
8434        WRITE ACT.X1(.AC)*16, ACT.Y1(.AC)*16, ACT.Z1(.AC)*16,
8435        ACT.X2(.AC)*16, ACT.Y2(.AC)*16, ACT.Z2(.AC)*16,
8436        ACT.X3(.AC)*16, ACT.Y3(.AC)*16, ACT.Z3(.AC)*16
8437        AS B 98, "P1", S 1, 3 I 5, /, B 98, "P2", S 1,
8438        3 I 5, /, B 98, "P3", S 1, 3 I 5, /
8439     ENDLOOP
8440     SKIP 5 LINES
8441     WRITE AS B 52, "PREPLANNED  MISSIONS", /, /,
8442     B 50, "ATTK   TARGET   ACFT   NUM",/,
8443     B 45, "NUM  (HRS)   UNIT    NAME   ACFT", /,
8444     B 45, "---  -----  -----  -----  ----", /
8445     LOOP FOR EACH .MISSION IN EV.S(I.INIT.PREPLAN.CAS)
8446     DO
8447        ADD 1 TO .COUNT
8448        WRITE .COUNT,
```

```
8449        TIME.A(.MISSION),
8450        UNIT.NOS(IPC.TARGET.UNIT(.MISSION)),
8451        EQ.NAME(ACT.EQUIP.ID(IPC.AC.TYPE(.MISSION))),
8452        IPC.NR.AC(.MISSION) AS
8453        B 45, I 2, B 49, D(6, 2), B 58, I 6,
8454        B 65, T 6, B 73, I 3, / ''                                    \TEXT
8455     ENDLOOP
8456
8457     ''WRITE THE MODEL.AD.SENSOR DATA.
8458     START NEW PAGE
8459     WRITE AS B 37, "MODEL  AD  SENSORS",/,
8460     B 75, "———RANGE HACKS———", /
8461     WRITE AS B 24, "DELAY    PW", B 45, "RIPL   RELOAD  ",
8462     "XMIT   RANGE  MIN ALT  PROB", /
8463     WRITE AS B 11, "SEQ  NAME   (SEC)  DEGRADE  FCM  (SEC)  ",
8464     "(MIN)   PCT    (MET)   ",
8465     "(MET)   DET",/
8466     WRITE AS B 11, "—— ———— ———— ——————— —— ———— ",
8467     "———— ———— ———— ———— ———",/
8468     LOOP FOR EVERY MODEL.AD.SENSOR CALLED .MADS
8469     DO
8470        WRITE .MADS,
8471        MADS.NAME(.MADS),
8472        MADS.DELAY.TIME(.MADS),
8473        MADS.PW.DEGRADE(.MADS),
8474        MADS.FCM(.MADS),
8475        MADS.RIPL(.MADS),
8476        MADS.WPN.RELOAD.TIME(.MADS),
8477        MADS.XMIT.PCT(.MADS) AS B 11,
8478        I 2, B 16, T 6, B 24, I 4, B 31, I 5, B 41, I 1, ''           \TEXT
8479        B 45, I 4, B 52, I 4, B 59, I 4
8480        FOR EVERY .RH IN MADS.RH.SET(.MADS)
8481        WRITE MRH.RANGE(.RH) * 16,
8482        MRH.MIN.ALT(.RH) * 16,
8483        MRH.PD(.RH) AS B 66, I 5, B 73, I 5,
8484        B 82, I 4, /
8485     ENDLOOP
8486
8487 <—RETURN
8488 END
```

```
8489 ROUTINE OUTPUT.EXPENDITURES                                                    ''                                                  0013
8490    ADD 1 TO ANAL.CTR(238,1)  ''                                                                                                \DYN_ANAL
8491    DEFINE I.WPN AND I.EQ AS INTEGER VARIABLES
8492    ''
8493    ''
8494    ''      THIS SECTION OUTPUTS INTERMEDIATE DATA TO UNIT 8 FOR LATER
8495    ''    STATISTICAL ANALYSIS.  TO "CATCH" THESE DATA IN A FILE, THE
8496    ''    COMMANDS:
8497    ''
8498    ''      @ASG,OPTIONS  FILENAME.
8499    ''      @USE  SIMU8,FILENAME.
8500    ''
8501    ''    MUST BE PRESENT IN THE RUNSTREAM.  THE RESULTING FILE CAN THEN
8502    ''    BE USED AS INPUT TO THE  STATEXP  PROGRAM BY MEANS OF AN  @ADD
8503    ''    COMMAND.
8504    ''
8505    ''
8506    DEFINE III AS AN INTEGER VARIABLE
8507    DEFINE TOTAL AS A 3-DIM INTEGER ARRAY
8508    RESERVE TOTAL AS 2 BY 2 BY *, TOTAL(1,1,*) AS N.RED.TYPE.EQP+1,
8509    TOTAL(1,2,*) AS N.B.WPN.TYPE, TOTAL(2,1,*) AS N.R.WPN.TYPE,
8510    TOTAL(2,2,*) AS N.BLUE.TYPE.EQP+1
8511    FOR I.WPN=1 TO N.B.WPN.TYPE,
8512    FOR I.EQ=1 TO N.RED.TYPE.EQP,
8513    DO  ''                                                                                                                          \1
8514       ADD STY.BLUE.EXP(I.WPN,I.EQ) TO TOTAL(1,1,I.EQ)
8515       ADD STY.BLUE.EXP(I.WPN,I.EQ) TO TOTAL(1,2,I.WPN)
8516    ENDLOOP ''                                                                                  REPLACED REPEAT WITH ENDLOOP \1
8517    LET III=N.RED.TYPE.EQP+1
8518    FOR I.EQ=1 TO N.RED.TYPE.EQP,
8519    ADD TOTAL(1,1,I.EQ) TO TOTAL(1,1,III)
8520    FOR I.WPN=1 TO N.R.WPN.TYPE,
8521    FOR I.EQ=1 TO N.BLUE.TYPE.EQP,
8522    DO  ''                                                                                                                          \1
8523       ADD STY.RED.EXP(I.WPN,I.EQ) TO TOTAL(2,1,I.WPN)
8524       ADD STY.RED.EXP(I.WPN,I.EQ) TO TOTAL(2,2,I.EQ)
8525    ENDLOOP ''                                                                                  REPLACED REPEAT WITH ENDLOOP \1
8526    LET III=N.BLUE.TYPE.EQP+1
8527    FOR I.EQ=1 TO N.BLUE.TYPE.EQP,
8528    ADD TOTAL(2,2,I.EQ) TO TOTAL(2,2,III)
8529    USE UNIT 8 FOR OUTPUT
8530    WRITE N.R.WPN.TYPE,N.B.WPN.TYPE,N.RED.TYPE.EQP,N.BLUE.TYPE.EQP AS
8531    4 I 3,/,S 9
8532    FOR I.EQ=1 TO N.RED.TYPE.EQP,
8533    DO ''                                                                                                                           \1
8534       IF MOD.F(I.EQ,10)=0 ''                                                                                                       \1
8535          WRITE AS /,S 9
8536       REGARDLESS
8537       WRITE EQ.NAME(N.BLUE.TYPE.EQP+I.EQ) AS A 7
8538    ENDLOOP ''                                                                                    REPLACED REPEAT WITH ENDLOOP
8539    WRITE AS /,/,S 1
8540    FOR I.WPN=1 TO N.B.WPN.TYPE,
8541    DO  ''                                                                                                                          \1
8542       WRITE TW.NAME(I.WPN) AS A 7
8543       FOR I.EQ=1 TO N.RED.TYPE.EQP,
8544       DO  ''                                                                                                                       \1
8545          IF MOD.F(I.EQ,10)=0 ''                                                                                                    \1
8546             WRITE AS /,S 8
```

```
8547          REGARDLESS
8548          WRITE STY.BLUE.EXP(I.WPN,I.EQ) AS I 7
8549       ENDLOOP ''                                          REPLACED REPEAT WITH ENDLOOP
8550       WRITE TOTAL(1,2,I.WPN) AS I 7,/,S 1
8551    ENDLOOP ''                                             REPLACED REPEAT WITH ENDLOOP
8552    WRITE AS /,S 8
8553    LET III=N.RED.TYPE.EQP+1
8554    FOR I.EQ=1 TO III,
8555    DO  ''                                                                              \1
8556       IF MOD.F(I.EQ,10)=0 ''                                                           \1
8557          WRITE AS /,S 8
8558       REGARDLESS
8559       WRITE TOTAL(1,1,I.EQ) AS I 7
8560    ENDLOOP ''                                             REPLACED REPEAT WITH ENDLOOP
8561    WRITE AS /,/,/,S 9
8562    FOR I.EQ=1 TO N.BLUE.TYPE.EQP,
8563    DO  ''                                                                              \1
8564       IF MOD.F(I.EQ,10)=0 ''                                                           \1
8565          WRITE AS /,S 9
8566       REGARDLESS
8567       WRITE EQ.NAME(I.EQ) AS A 7
8568    ENDLOOP ''                                             REPLACED REPEAT WITH ENDLOOP
8569    WRITE AS /,/,S 1
8570    FOR I.WPN=1 TO N.R.WPN.TYPE,
8571    DO ''                                                                               \1
8572       WRITE TW.NAME(N.B.WPN.TYPE+I.WPN) AS A 7
8573       FOR I.EQ=1 TO N.BLUE.TYPE.EQP,
8574       DO ''                                                                            \1
8575          IF MOD.F(I.EQ,10)=0 ''                                                        \1
8576             WRITE AS /,S 8
8577          REGARDLESS
8578          WRITE STY.RED.EXP(I.WPN,I.EQ) AS I 7
8579       ENDLOOP ''                                          REPLACED REPEAT WITH ENDLOOP
8580       WRITE TOTAL(2,1,I.WPN) AS I 7,/,S 1
8581    ENDLOOP ''                                             REPLACED REPEAT WITH ENDLOOP
8582    WRITE AS /,S 8
8583    LET III=N.BLUE.TYPE.EQP+1
8584    FOR I.EQ=1 TO III,
8585    DO  ''                                                                              \1
8586       IF MOD.F(I.EQ,10)=0 ''                                                           \1
8587          WRITE AS /,S 8
8588       REGARDLESS
8589       WRITE TOTAL(2,2,I.EQ) AS I 7
8590    ENDLOOP ''                                             REPLACED REPEAT WITH ENDLOOP
8591    USE UNIT 6 FOR OUTPUT
8592    ''
8593    ''
8594    PRINT 2 LINES AS FOLLOWS
8595    EXPENDITURES FOR BLUE WEAPONS AGAINST RED TARGETS
8596    WPN              TGT          RNDS
8597    LOOP FOR I.WPN = 1 TO N.B.WPN.TYPE,
8598    DO ''                                                                               \1
8599       LOOP FOR I.EQ = 1 TO N.RED.TYPE.EQP,
8600       DO ''                                                                            \1
8601          IF STY.BLUE.EXP(I.WPN,I.EQ) NE 0
8602             PRINT 1 LINE WITH TW.NAME(I.WPN), EQ.NAME(I.EQ+N.BLUE.TYPE.EQP)
8603             AND STY.BLUE.EXP(I.WPN,I.EQ) THUS
8604             **********     **********     *****
```

```
8605          ENDIF
8606       ENDLOOP
8607    ENDLOOP
8608    PRINT 2 LINES AS FOLLOWS
8609    EXPENDITURES FOR RED WEAPONS AGAINST BLUE TARGETS
8610    WPN              TGT          RNDS
8611    LOOP FOR I.WPN = 1 TO N.R.WPN.TYPE,
8612    DO ''
8613       LOOP FOR I.EQ = 1 TO N.BLUE.TYPE.EQP,                                    \1
8614       DO ''                                                                    \1
8615          IF STY.RED.EXP(I.WPN,I.EQ) NE 0
8616             PRINT 1 LINE WITH TW.NAME(I.WPN + N.B.WPN.TYPE),
8617             EQ.NAME(I.EQ), STY.RED.EXP(I.WPN,I.EQ) THUS
8618             **********     **********     *****
8619          ENDIF
8620       ENDLOOP
8621    ENDLOOP
8622 <--EXITROUTINE
8623 ENDROUTINE
```

FUNCTIONS

```
8630 FUNCTION ACT.RANGE                                                    ''          F001
8631    GIVEN UNIT1 AND UNIT2
8632
8633    ADD 1 TO ANAL.CTR(239,1)  ''                                                   \DYN_ANAL
8634    ''THIS FUNCTION COMPUTES THE ACTUAL RANGE BETWEEN TWO UNITS
8635
8636    DEFINE UNIT1, UNIT2, DELTA.X, AND DELTA.Y AS INTEGER VARIABLES
8637    DEFINE RANGE AS A REAL VARIABLE
8638
8639    LET DELTA.X = UN.X.COORD( UNIT1 ) - UN.X.COORD( UNIT2 )
8640    LET DELTA.Y = UN.Y.COORD( UNIT1 ) - UN.Y.COORD( UNIT2 )
8641    LET RANGE = SQRT.F(REAL.F(DELTA.X*DELTA.X+DELTA.Y*DELTA.Y)) ''              \OPTIMIZE
8642
8643 <--RETURN WITH RANGE
8644 END
```

```
8645 FUNCTION BTRY.AVAILABLE(BTRY, TARGET, DURATION)            ''
8646
8647    ADD 1 TO ANAL.CTR(240,1)  ''
8648    ''THIS FUNCTION DETERMINES FIRE UNIT AVAILABILITY FOR A
8649    ''SCHEDULED FIRE MISSION IN TERMS OF OTHER SCHEDULED FIRE
8650    ''MISSIONS ALREADY ASSIGNED TO THIS FIRE UNIT.
8651
8652    NORMALLY MODE IS INTEGER
8653    DEFINE STOP AND START AS REAL VARIABLES
8654
8655    LET START = TR.START.TIME( TARGET )
8656    LET STOP = START + REAL.F(DURATION)/100.
8657    LET TB = BY.TYPE(BTRY)
8658    LOOP FOR EACH MSN IN BY.SCHD.LIST( BTRY )
8659    WHILE FM.START.TIME( MSN ) < STOP
8660    DO THIS
8661       IF FM.START.TIME( MSN ) <= START <=
8662          ( FM.START.TIME( MSN ) +  REAL.F( FM.N.VOLS( MSN ) ) /
8663          TB.SUST.FIRE.RATE( TB ) ) / 6.
8664          OR
8665          FM.START.TIME( MSN ) <= STOP <=
8666          ( FM.START.TIME( MSN ) +  REAL.F( FM.N.VOLS( MSN ) ) /
8667          TB.SUST.FIRE.RATE( TB  ) ) / 6.
8668          LET AVAILABILITY = FALSE     '' THERE IS A CONFLICT
8669    <------EXITLOOP
8670       OTHERWISE
8671       LET AVAILABILITY = TRUE
8672    ENDLOOP
8673    ''IF DEBUG=TRUE,
8674       ''PRINT 1 LINE WITH TARGET, BTRY, AVAILABILITY THUS
8675       ''= = = BTRY.AVAILABLE TARGET=*****, BTRY=***, AVAILABILITY=*  = = =
8676    ''ENDIF
8677 <--RETURN WITH AVAILABILITY
8678 ENDFUNCTION
```

```
8679 FUNCTION COLLISION(.UNIT, .ORDER)                                  ''                                                    F004
8680
8681    ADD 1 TO ANAL.CTR(241,1)  ''                                                                                          \DYN_ANAL
8682    NORMALLY MODE IS INTEGER
8683
8684    FOR EACH UPDATE.LOC IN EV.S(I.UPDATE.LOC)
8685    WITH UL.UNIT(UPDATE.LOC) = .UNIT
8686    FIND THE FIRST CASE
8687    IF FOUND
8688       CANCEL_THE UPDATE.LOC------------------------------------------------------------------------------------------>(408)
8689       DESTROY THE UPDATE.LOC
8690    ELSE
8691 <------EXITFUNCTION WITH NO ''                                                                                           \CLEANUP
8692    ALWAYS
8693
8694    LET MU.CUR.ORDER(UN.PTR(.UNIT)) = NX.ORD.ABOVE(ORD.ID(.ORDER))
8695    FOR EACH .NX.ORDER IN MU.ORDER.SET(UN.PTR(.UNIT))
8696    WITH ORD.SEQ.NO(.NX.ORDER) = MU.CUR.ORDER(UN.PTR(.UNIT))
8697    FIND THE FIRST CASE
8698    IF NONE
8699       CALL ERROR.STOP------------------------------------------------------------------------------------------------>(604)
8700    ALWAYS
8701    IF ORD.TYPE(.NX.ORDER) NE "ATK"
8702       CALL ERROR.STOP------------------------------------------------------------------------------------------------>(604)
8703    ALWAYS
8704
8705    ''MOVEMENT ENDS. EMPTY LIST OF BARRIER MINES.
8706    LOOP FOR EVERY .MO OF MO.LIST(.UNIT)
8707    DO
8708       REMOVE .MO FROM THE MO.LIST(.UNIT)
8709       DESTROY THE MINE.OBSTACLE CALLED .MO
8710    ENDLOOP
8711
8712 <--EXITFUNCTION WITH YES
8713 ENDFUNCTION
```

```
8714 FUNCTION COMBINATIONS( N , M )   ''                                                                    F005
8715
8716
8717    ADD 1 TO ANAL.CTR(242,1)  ''                                                                        \DYN_ANAL
8718    ''CALLED BY SIZE.ESTIMATE
8719    NORMALLY MODE IS INTEGER
8720    IF M > N
8721       CALL ERROR.STOP------------------------------------------------------------------------------------>(604)
8722    ENDIF
8723    IF M < N/2
8724       LET M = N - M
8725    ENDIF
8726    LET NUMERATOR = 1
8727    LOOP FOR I = M + 1  TO  N ,
8728    DO THIS                                                 ''                                          \1
8729       LET NUMERATOR = NUMERATOR * I
8730    ENDLOOP
8731    LET DENOMINATOR = 1
8732    LOOP FOR I = 1  TO  N - M ,
8733    DO THIS                                                  ''                                         \1
8734       LET DENOMINATOR = DENOMINATOR * I
8735    ENDLOOP
8736    IF NUMERATOR <= 0      '' IT GOT TRUNCATED
8737       LET NUMERATOR = INF.C
8738    ALWAYS
8739    IF DENOMINATOR <= 0
8740       LET DENOMINATOR = INF.C
8741    ALWAYS
8742 <--RETURN WITH LOG.E.F(REAL.F(NUMERATOR)) - LOG.E.F(REAL.F(DENOMINATOR))
8743 ENDFUNCTION
```

```
8744 FUNCTION EST.RANGE(TARGET , UNIT)                                  ''                                  F006
8745
8746
8747    ADD 1 TO ANAL.CTR(243,1)  ''                                                                        \DYN_ANAL
8748    ''THIS FUNCTION IS USED TO COMPUTE THE RANGE BETWEEN
8749    ''A GIVEN UNIT AND A TARGET REPORT(ESTIMATED LOCATION)
8750    ''CALLED BY TARGET.ANALYSIS
8751    NORMALLY MODE IS INTEGER
8752    LET DELTA.X = TR.EST.X( TARGET ) - UN.X.COORD(  UNIT )
8753    LET DELTA.Y = TR.EST.Y( TARGET ) - UN.Y.COORD(  UNIT )
8754    LET RANGE = SQRT.F( DELTA.X ** 2  +  DELTA.Y ** 2 ) ''                                              \OPTIMIZE
8755    IF DEBUG=TRUE,
8756       PRINT 1 LINE WITH TARGET, UNIT.NOS(UNIT), RANGE*16 THUS
8757       = = = EST.RANGE TARGET=******(UNIT ******) AT ****** METERS = = =
8758    ENDIF
8759 <--RETURN WITH RANGE
8760 ENDFUNCTION
```

```
8761 FUNCTION EST.TR.RANGE(TR1 , TR2)                                    ''
8762    ADD 1 TO ANAL.CTR(244,1)  ''                                                F007
8763    ''THIS FUNCTION COMPUTES THE RANGE BETWEEN TWO TARGET REPORTS               \DYN_ANAL
8764    NORMALLY MODE IS INTEGER
8765    LET DELTA.X = TR.EST.X( TR1 ) - TR.EST.X( TR2 )
8766    LET DELTA.Y = TR.EST.Y( TR1 ) - TR.EST.Y( TR2 )
8767    LET RANGE = SQRT.F( DELTA.X ** 2  +  DELTA.Y ** 2 ) ''
8768    IF DEBUG=TRUE,                                                           \OPTIMIZE
8769       PRINT 1 LINE WITH TR1,TR2,RANGE THUS
8770       = = = EST.TR.RANGE TR1=******, TR2=******, RANGE=****** = = =
8771    ENDIF
8772 <--RETURN WITH RANGE
8773 ENDFUNCTION
```

```
8774 FUNCTION FEBA.BAND(TARGET)                                                    ''                                    F008
8775     ''                                                                                                          \REDUNDANT_CODE
8776
8777     ADD 1 TO ANAL.CTR(245,1)  ''                                                                                  \DYN_ANAL
8778     ''THIS ROUTINE IS CALLED BY HE.OR.ICM.COMPUTE, ILLUMINATION.COMP-
8779     ''UTE, SMOKE.COMPUTE, OR TARGET REPORT,  IT DETERMINES THE DFFB
8780     ''VALUE EMPLOYED BY THE ARTILLERY RULES OF ENGAGEMENT.
8781
8782     NORMALLY MODE IS INTEGER
8783     DEFINE SIDE,SECTOR AS INTEGER VARIABLES
8784
8785     '' TO GIVE THE TGT SAME SECTOR AND SIDE
8786     LET SIDE = UN.COLOR(TR.TGT.UNIT(TARGET))
8787     IF TR.MISSION.TYPE(TARGET) = "ILLUM" OR
8788        TR.MISSION.TYPE(TARGET) = "FASCAM" OR
8789        TR.MISSION.TYPE(TARGET) = "SMOKE"
8790        ''BTRY FIRES AT A LOCATION BASED ON THE SMOKE RULE AND
8791        ''AND THE LOCATION OF THE TARGETED FORCE, NOT AT A UNIT.
8792        CALL LOCATE.SECTOR------------------------------------------------------------------------------------------>(100)
8793        GIVEN
8794        TR.EST.Y(TARGET)
8795        YIELDING
8796        SECTOR
8797        IF SS.SET(SIDE,SECTOR) IS EMPTY ''                                                                        CHG\30 \DEBUG
8798           LET DFFB = 1
8799 <----------RETURN WITH DFFB
8800        OTHERWISE
8801        IF SIDE=BLUE
8802           LET FEBA=F.SS.SET(SIDE,SECTOR)
8803           LET DIST = UN.X.COORD(FEBA) - TR.EST.X(TARGET)
8804        ELSE
8805           LET FEBA=L.SS.SET(SIDE,SECTOR)
8806           LET DIST = TR.EST.X(TARGET) - UN.X.COORD(FEBA)
8807        ALWAYS
8808     ELSE
8809        ''BTRY FIRES AT A SPECIFIC UNIT.
8810        CALL LOCATE.SECTOR------------------------------------------------------------------------------------------>(100)
8811        GIVEN
8812        UN.Y.COORD(TR.TGT.UNIT(TARGET))
8813        YIELDING
8814        SECTOR
8815        IF SS.SET(SIDE,SECTOR) IS EMPTY ''                                                                        CHG\30 \DEBUG
8816           LET DFFB = 1
8817 <----------RETURN WITH DFFB
8818        OTHERWISE
8819        IF SIDE=BLUE
8820           LET FEBA=F.SS.SET(SIDE,SECTOR)
8821           LET DIST = UN.X.COORD(FEBA) - UN.X.COORD(TR.TGT.UNIT(TARGET))
8822        ELSE
8823           LET FEBA=L.SS.SET(SIDE,SECTOR)
8824           LET DIST = UN.X.COORD(TR.TGT.UNIT(TARGET)) - UN.X.COORD(FEBA)
8825        ALWAYS
8826     ALWAYS
8827     IF FEBA = 0      ''SS.SET IS EMPTY
8828        LET DFFB = 1
8829 <-------RETURN WITH DFFB
8830     OTHERWISE
8831
```

```
8832     IF TR.MISSION.TYPE(TARGET) = "ILLUM" OR
8833        TR.MISSION.TYPE(TARGET) = "FASCAM" OR
8834        TR.MISSION.TYPE(TARGET) = "SMOKE"
8835        ''NEGATIVE VALUES OK SINCE UNIT MAY REQUEST
8836        ''THESE NEAR ITSELF
8837        LET DFFB = 1
8838        LET DIST = 1
8839     ALWAYS
8840     IF DIST < 0
8841        LET DFFB = 33
8842 <------RETURN WITH DFFB
8843     OTHERWISE
8844
8845     FOR EACH DIST.FROM.FEBA.BAND CALLED DFFB
8846     WHEN DFFB.MAX.RANGE(DFFB)>=DIST
8847     FIND THE FIRST CASE
8848     IF NONE
8849        LET DFFB = N.DIST.FROM.FEBA.BAND
8850     ALWAYS
8851
8852     IF DEBUG = TRUE
8853        PRINT 1 LINE WITH TARGET, DFFB THUS
8854        = = = FEBA.BAND  TARGET = *****,  DFFB = ***  = = =
8855     ALWAYS
8856
8857 <--RETURN WITH DFFB
8858 ENDFUNCTION
```

```
8859 FUNCTION HE.WLA(TARGET,BTRY,MUNITION,RANGE,BAT.NUM,FM)  ''
8860    ADD 1 TO ANAL.CTR(246,1)  ''
8861    ''THIS FUNCTION COMPUTES THE EFFECTIVE OR "WEIGHTED" LETHAL AREA OF HE
8862    ''ROUNDS BEING FIRED AT EQUIPMENT FOUND IN VARYING ENVIRONMENTAL STATES
8863    ''AND AT PERSONNEL WITH THE EQUIPMENT FOUND IN A MIX OF POSTURES AND IN
8864    ''A CERTAIN STATE OF WARNING WHEN THE FIRST VOLLEY ARRIVES AT THE TARGET
8865    NORMALLY MODE IS INTEGER
8866    DEFINE OPEN TO MEAN 1
8867    DEFINE PERSONNEL TO MEAN 1
8868    DEFINE FUZE AS AN INTEGER VARIABLE
8869    DEFINE RELY, PI.RAD.2, LA, WLA, FRACT.POST, AND PK.HE AS REAL VARIABLES
8870    '' IF DEBUG=TRUE,
8871       '' PRINT 1 LINE WITH TARGET,BTRY,MUNITION,RANGE,BAT.NUM THUS
8872       '' = = = HE.WLA TARGET=******, BTRY=***, MUNITION=***, RANGE=******, BAT=**** =
8873    '' ENDIF
8874    LET TB = BY.TYPE( BTRY )
8875    LET MSN = UN.MISSION(TR.TGT.UNIT(TARGET))
8876    IF MSN = 0        ''IF A UNIT HAS NO MSN,    %28OCT80_%HWJ
8877       LET MSN = DEFEND  ''GIVE IT A DEFEND MSN
8878    ENDIF
8879    IF RANGE > HE.RH.RANGE(L.HE.TB.RH.LIST( MUNITION, TB ))
8880 <------EXITFUNCTION WITH 0.0
8881    OTHERWISE
8882    FOR EACH RNG.HACK IN HE.TB.RH.LIST(  MUNITION, TB ),
8883    WHEN HE.RH.RANGE( RNG.HACK ) >= RANGE,
8884    FIND THE FIRST CASE
8885    IF NONE
8886       LET WLA = 0.0
8887 <------RETURN WITH WLA
8888    ENDIF
8889    LET TU = TR.EST.TU( TARGET )
8890    IF TU = 0
8891       PRINT 1 LINE WITH TARGET THUS
8892       &%&% WARNING  === HE.WLA HAS TU = 0 FOR TARGET *******
8893       LET WLA = 0.
8894       LIST ATTRIBUTES OF TARGET.REPORT CALLED TARGET
8895 <------RETURN WITH WLA
8896    OTHERWISE
8897    LET PRIN.EQ = TU.PRIN.TE( TU )
8898    LET CAT = TU.CAT(TU)
8899    LET PI.RAD.2 = PI.C*REAL.F( HE.ROUND.RAD( MUNITION ))**2
8900    IF BAT.NUM = 1 OR TR.TOT.STATUS( TARGET ) = TRUE
8901       LET UNWARNED = TRUE
8902    ELSE
8903       LET UNWARNED = FALSE
8904    ENDIF
8905    IF EC.FRACT( OPEN, CAT ) > 50
8906       LET FUZE = VT
8907    ELSE
8908       LET FUZE = PD
8909    ENDIF
8910    LET RELY = REAL.F(FZ.HE.RELY(FUZE, MUNITION)) / 100.
8911    FOR EVERY ENVIRONMENT WITH EN.NAME(ENVIRONMENT) = "TOWN"
8912    FIND THE FIRST CASE ''%10OCT80_%RWF  FIND THE TOWN ENVIRONMENT
8913    IF NONE
8914       CALL ERROR.STOP-------------------------------------------------------------------------------------------------------->(604)
8915    ENDIF
8916    LET ENV.TEST =    ''%10OCT80_%RWF IF THE PROPORTION OF THAT TYPE
```

```
8917    RANDOM.F(RN.SEED)*100.          '' TARGET NORMALLY IN TOWN IS GTE
8918    IF ENV.TEST LE EC.FRACT(ENVIRONMENT,CAT) '' A RANDOM DRAW, LET
8919       LET FM.TM.CLASS(FM) = "HE"              '' THE ENTIRE TARGET BE
8920       LET ENV.FRACT.TOWN = 100                '' CONSIDERED IN TOWN
8921    ENDIF
8922    LOOP FOR EACH ENVIRONMENT CALLED ENVIR
8923    DO THIS ''                                                          \1
8924       LET ENV.PROP = EC.FRACT(ENVIR,CAT)  ''%10OCT80_%RWF SET THE
8925       IF ENV.FRACT.TOWN = 100             ''PROPORTION TO THE INPUT
8926          IF EN.NAME(ENVIR) NE "TOWN"        ''VALUE AND
8927             LET ENV.PROP = 0                  '' ADJUST THE PROPORTIONS
8928          ELSE                              '' FOR A TOWN TARGET
8929             LET ENV.PROP = 100
8930          ALWAYS
8931       ENDIF
8932       IF PRIN.EQ NE PERSONNEL
8933          LET LA = REAL.F(RTEF.LA.EQUIP( RNG.HACK, PRIN.EQ,
8934          ENVIR, FUZE )) / 10.
8935          LET PK.HE = LA * RELY / PI.RAD.2
8936          LET WLA = WLA + PI.RAD.2 * PK.HE * (ENV.PROP/100.)
8937       ELSE
8938          LOOP FOR EACH POSTURE CALLED POST
8939          DO THIS ''                                                    \1
8940             LET LA = REAL.F(REPF.LA.PERS( RNG.HACK,
8941             ENVIR, POST, FUZE))          '' %6MAR80_%RGR
8942             LET PK.HE = LA * RELY / PI.RAD.2
8943             IF UNWARNED = TRUE
8944                LET FRACT.POST = REAL.F(CPM.UNWARNED.FRACT( CAT ,
8945                POST, MSN)) /100.
8946             ELSE
8947                LET FRACT.POST = REAL.F( CPM.WARNED.FRACT( CAT ,
8948                POST, MSN)) /100.
8949             ENDIF
8950             LET WLA = WLA + PI.RAD.2 * PK.HE * (ENV.PROP/100.) * FRACT.POST
8951          ENDLOOP
8952       ENDIF
8953    ENDLOOP
8954    ''LIST WLA
8955 <—RETURN WITH WLA
8956 ENDFUNCTION
```

```
8957 FUNCTION ICM.WLA(TARGET,BTRY,MUNITION,RANGE)  ''                                   F010
8958
8959
8960     ADD 1 TO ANAL.CTR(247,1)  ''                                                  \DYN_ANAL
8961     ''THIS FUNCTION COMPUTES THE EFFECTIVE OR "WEIGHTED" LETHAL AREA OF ICM
8962     ''ROUNDS BEING FIRED AT EQUIPMENT FOUND IN VARYING ENVIRONMENTAL STATES
8963     ''AND AT PERSONNEL WITH THE EQUIPMENT FOUND IN A MIX OF POSTURES AND IN
8964     ''A CERTAIN STATE OF WARNING WHEN THE FIRST VOLLEY ARRIVES AT THE TARGET
8965     NORMALLY MODE IS INTEGER
8966     DEFINE RELY, PATT.RAD.2, LA, X, PK.ICM, WLA AS REAL VARIABLES
8967     DEFINE PERSONNEL TO MEAN 1
8968     LET TB = BY.TYPE( BTRY )
8969     '' IF DEBUG=TRUE,
8970        '' PRINT 1 LINE WITH TARGET,BTRY,MUNITION,RANGE,TB THUS
8971        '' = = = ICM.WLA TARGET=******, BTRY=***, MUNITION=***, RANGE=****, TB=*** = = =
8972     '' ENDIF
8973     IF RANGE > IC.RH.RANGE(L.IC.TB.RH.LIST( MUNITION, TB ))
8974        LET WLA =  0.0
8975 <------RETURN WITH WLA
8976 <------''EXITFUNCTION - IT IS OUT OF RANGE
8977     OTHERWISE
8978     FOR EACH RNG.HACK IN IC.TB.RH.LIST(MUNITION,TB)
8979     WHEN IC.RH.RANGE(RNG.HACK) GE RANGE
8980     FIND THE FIRST CASE
8981     IF NONE
8982        LET WLA = 0.0
8983 <------RETURN WITH WLA
8984 <------''EXITFUNCTION  - ABOVE CODE ADDED %5FEB80_%RGR
8985     OTHERWISE    '' IT IS WITHIN RANGE                          ELSE CHANGED TO OTHERWISE \1
8986     LET PATT.RAD.2=(IC.TB.INTERCEPT(MUNITION,TB)+RANGE*(16./1000.)*
8987     IC.TB.SLOPE(MUNITION, TB)) ** 2   ''                                          \OPTIMIZE
8988     LET QUANT = IC.N.SUBM( MUNITION )
8989     LET ST = IC.SUBM.INDEX( MUNITION )
8990     LET TU = TR.EST.TU( TARGET )
8991     IF TU = 0
8992        LET WLA = 0
8993 <------RETURN WITH WLA
8994     OTHERWISE
8995     LET PRIN.EQ = TU.PRIN.TE( TU )
8996     LET CAT = TU.CAT(TU)
8997     LET UN = TR.TGT.UNIT(TARGET)
8998     LET MSN = UN.MISSION(UN)
8999     IF MSN = 0        ''IF A UNIT HAS NO MSN,    %28OCT80_%HWJ
9000        LET MSN = DEFEND   ''GIVE IT A DEFEND MSN
9001     ENDIF
9002     LOOP FOR EACH ENVIRONMENT CALLED ENVIR
9003     DO THIS   ''                                                                  \1
9004        LET RELY = REAL.F(ES.RELY(ENVIR, ST)*IC.RELIABILITY(MUNITION))/10000.
9005        IF PRIN.EQ NE PERSONNEL
9006           LET LA = REAL.F(TES.LA.EQUIP( PRIN.EQ, ENVIR, ST ))/10.
9007           LET X = QUANT * RELY * LA / (PI.C * PATT.RAD.2)
9008           LET PK.ICM = 1.0 - EXP.F( -X )
9009           LET WLA = WLA + PI.C * PATT.RAD.2 * PK.ICM
9010           * REAL.F( EC.FRACT( ENVIR, CAT ) ) / 100.
9011        ELSE
9012           LOOP FOR EACH POSTURE CALLED POST                                      \1
9013           DO THIS  ''
9014              LET LA = REAL.F( EPS.LA.PERS( ENVIR, POST, ST ) ) / 10.
```

```
9015             LET X = QUANT * RELY * LA / (PI.C * PATT.RAD.2)
9016             LET PK.ICM = 1.0 - EXP.F( -X )
9017             LET WLA = WLA + PI.C * PATT.RAD.2 * PK.ICM
9018             * REAL.F( EC.FRACT( ENVIR, CAT ) ) / 100.
9019             * REAL.F(CPM.WARNED.FRACT(CAT,POST,MSN)) / 100.
9020          ENDLOOP
9021       ENDIF
9022    ENDLOOP
9023    ''LIST WLA
9024 <--RETURN WITH WLA
9025 ENDFUNCTION
```

```
9026 ROUTINE EXPONENTIAL.F(MU,STREAM)  '' (SHOULD BE FUNCTION) \CLEANUP                F012
9027    ADD 1 TO ANAL.CTR(248,1)  ''                                                  \DYN_ANAL
9028    DEFINE STREAM AS AN INTEGER VARIABLE
9029    DEFINE MU, X AS REAL VARIABLES
9030    LOOP UNTIL X > 0.0
9031    DO ''                                                                         \1
9032       LET X = RANDOM.F(STREAM)
9033    ENDLOOP
9034    IF MU < 0.0
9035       LET ERR.F = 132
9036    ALWAYS
9037 <--RETURN WITH -MU*LOG.E.F(X)
9038 ENDROUTINE
```

```
9039 ROUTINE NORMAL.F(MU, SIGMA, STREAM)'' (SHOULD BE FUNCTION) \CLEANUP                F013
9040    ADD 1 TO ANAL.CTR(249,1)  ''                                                   \DYN_ANAL
9041    DEFINE MU AND SIGMA AS REAL VARIABLES
9042    DEFINE STREAM AS AN INTEGER VARIABLE
9043    NORMALLY, MODE IS REAL
9044    IF SIGMA<=0, ''                                                                \1
9045       LET ERR.F=137
9046    ALWAYS
9047    'A' LET X=RANDOM.F(STREAM)  ''                                                 \1
9048    LET Y=2*RANDOM.F(STREAM)-1
9049    LET XX=X*X
9050    LET YY=Y*Y
9051    LET S=XX+YY
9052    IF S > 1, ''                                                                   \1
9053       GO TO A
9054    OTHERWISE  ''                                  ELSE REPLACED WITH OTHERWISE \1
9055    LOOP UNTIL Z > 0.0
9056    DO  ''                                                                         \1
9057       LET Z = RANDOM.F(STREAM)
9058    ENDLOOP
9059    LET R=SQRT.F(-2*LOG.E.F(Z))/S
9060 <--RETURN WITH MU+(XX-YY)*R*SIGMA
9061 END
```

```
9062 ROUTINE WEIBULL.F (SHAPE,SCALE,STREAM) '' (SHOULD BE FUNCTION) \CLEANUP          F014
9063    ADD 1 TO ANAL.CTR(250,1)  ''                                                \DYN_ANAL
9064    NORMALLY MODE IS INTEGER
9065    DEFINE SHAPE, SCALE, X AS REAL VARIABLES
9066    UNTIL X > 0.00
9067    DO ''                                                                       \1
9068       LET X = RANDOM.F(STREAM)
9069    ENDLOOP
9070 <—RETURN WITH   SCALE * ((-LOG.E.F(X))**(1/SHAPE))
9071 END
```

SUPPORT AND COMPUTATION ROUTINES

```
9078 ROUTINE ANGLE.COMPUTE                                                    ''      S001
9079     GIVEN
9080     DX,
9081     DY
9082     YIELDING
9083     THETA
9084
9085     ADD 1 TO ANAL.CTR(251,1)  ''                                                 \DYN_ANAL
9086     NORMALLY MODE IS INTEGER
9087     DEFINE THETA AS A REAL VARIABLE
9088
9089     IF DY = 0,
9090        IF DX = 0,
9091           LET THETA = 0
9092 <--------EXITROUTINE
9093        OTHERWISE
9094
9095        IF DX GT 0,
9096           LET THETA = 0
9097        ELSE
9098           LET THETA = PI.C
9099        ALWAYS
9100 <------EXITROUTINE
9101     OTHERWISE
9102
9103     IF DX = 0
9104        IF DY GT 0,
9105           LET THETA = PI.C/2  ''                                                 \OPTIMIZE
9106        ELSE
9107           LET THETA = PI.C/2 + PI.C
9108        ALWAYS
9109 <------EXITROUTINE
9110     OTHERWISE
9111
9112     LET THETA = ARCTAN.F(REAL.F(DY),REAL.F(DX))
9113
9114 <---EXITROUTINE
9115 ENDROUTINE
```

```
9116 ROUTINE LINE.CIRCLE                                                    ''                                                                  S002
9117     GIVEN
9118     .M,
9119     .YI,
9120     .X0,
9121     .Y0,
9122     .R,
9123     .AD
9124
9125     YIELDING
9126     .X1,
9127     .Y1,
9128     .X2,
9129     .Y2,
9130     .ROOT
9131
9132     ADD 1 TO ANAL.CTR(252,1)  ''                                                                                                    \DYN_ANAL
9133     ''THIS ROUTINE FINDS THE INTERSECTIONS, (X1, Y1) AND
9134     ''(X2, Y2), OF A LINE Y = MX + B, AND A CIRCLE OF
9135     ''RADIUS R CENTERED AT (X0, Y0).
9136
9137     NORMALLY MODE IS INTEGER
9138     DEFINE .A, .B, .C, .M, .ROOT, .YI AS REAL VARIABLES
9139
9140     LET .A = 1 + .M**2 ''                                                                                                           \OPTIMIZE
9141     LET .B = 2. * .M * (.YI - .Y0) - 2. * .X0
9142     LET .C = .Y0**2 + .X0**2  + .YI**2 - .R**2
9143     - 2. * .YI * .Y0
9144     LET .ROOT = .B**2 - 4. * .A * .C
9145     IF .ROOT GE 0.0          ''INTERSECTION
9146        LET .ROOT = SQRT.F(.ROOT)
9147        LET .X1 = (-.B - .ROOT) / (2. * .A)
9148        LET .X2 = (-.B + .ROOT) / (2. * .A)
9149        LET .Y1 = .M * .X1 + .YI
9150        LET .Y2 = .M * .X2 + .YI
9151     ALWAYS
9152
9153 <--RETURN
9154 END
```

```
9155 ROUTINE MRT.TO.FREQ                                                    ''      S003
9156     GIVEN
9157     .DELTA.T.PRIME
9158
9159     YIELDING
9160     .FREQUENCY
9161
9162     ADD 1 TO ANAL.CTR(253,1)  ''                                               \DYN_ANAL
9163     NORMALLY MODE IS INTEGER
9164     DEFINE .DELTA.T.PRIME, .FREQUENCY AS REAL VARIABLES
9165
9166     IF .DELTA.T.PRIME LT 0.0
9167        TRACE
9168 <------STOP
9169     OTHERWISE
9170
9171     IF .DELTA.T.PRIME = 0.0
9172        LET .FREQUENCY = 0.0
9173     ELSE
9174        LET .FREQUENCY = .DELTA.T.PRIME
9175        / (0.0445 + 0.1473 * .DELTA.T.PRIME)
9176     ALWAYS
9177
9178 <--RETURN
9179 END
```

```
9180 ROUTINE OPEN.INPUT.OUTPUT.FILES  ''                                    S004
9181
9182    OPEN UNIT 1 FOR INPUT, FILE NAME IS "MODINPUT.DAT", OLD, ''         CHG\11
9183    RECORDSIZE=132
9184    OPEN UNIT 79 FOR INPUT, FILE NAME IS "ROUNAM.DAT", OLD
9185    OPEN UNIT 80 FOR OUTPUT, FILE NAME IS "DYNANL.LIS", NEW,
9186    RECORDSIZE=132
9187    USE UNIT 1 FOR INPUT  ''                                            CHG\11
9188    WRITE AS B 3, "IN ROUTINE OPEN.INPUT.OUTPUT.FILES", /
9189
9190 <--RETURN
9191 END
```

```
9192 ROUTINE PERFORM.INSTRUMENTATION    ''                                                    \DYN_ANAL  S005
9193
9194    DEFINE I AS AN INTEGER VARIABLE
9195
9196    RESERVE ANAL.CTR(*,*) AS 264 BY *
9197    RESERVE ANAL.TEXT(*) AS 264
9198
9199    FOR I = 1 TO 264
9200    DO
9201       READ ANAL.TEXT(I) USING UNIT 79
9202       ''DON'T READ SPOTS YET
9203       ''RESERVE ANAL.CTR(I,*) AS SPOTS
9204       RESERVE ANAL.CTR(I,*) AS 1
9205    ENDLOOP
9206
9207    SCHEDULE_A DYNAMIC.ANALYSIS.REPORT IN 1 HOURS------------------------------------------------------------------------------------>(417)
9208
9209    ADD 1 TO ANAL.CTR(254,1)    ''                                          FOR OPEN.INPUT.OUTPUT.FILES
9210
9211 <--RETURN
9212
9213 END
```

```
9214 ''PROGRAM LIB$INIT_TIMER
9215
9216    ********************************************************************************************************************************
9217    * LIB$INIT_TIMER INITIALIZES THE CPU TIME COUNTER.                                                                             *
9218    ********************************************************************************************************************************
9219
9220 ENDROUTINE
```

```
9221 ''PROGRAM LIB$STAT_TIMER(FLAG, CPU.TIME)
9222
9223    ********************************************************************************************************************
9224    * LIB$STAT_TIMER RETURNS THE TOTAL CPU USAGE (IF FLAG IS 2)                                                        *
9225    * SINCE THE TIMER WAS INITIALIZED BY SYS$INIT_TIMER IN THE                                                         *
9226    * ARGUMENT CPU.TIME.                                                                                               *
9227    ********************************************************************************************************************
9228
9229 ENDROUTINE
```

UN-USED & DELETION CANDIDATE MODULES

```
9236 ROUTINE GAMMA.F(MEAN, K, STREAM)  '' (SHOULD BE FUNCTION) \CLEANUP \REQUIRED?                U001
9237     ''  COPYRIGHT 1974, CACI, INC.
9238     ADD 1 TO ANAL.CTR(255,1)  ''                                                            \DYN_ANAL
9239     DEFINE MEAN,K,KK AND E AS REAL VARIABLES
9240     DEFINE STREAM AS AN INTEGER VARIABLE
9241     IF MEAN<=0, ''                                                                          \1
9242        LET ERR.F=145
9243     ENDIF
9244     IF K<=0,
9245        LET ERR.F=146
9246     ENDIF
9247     LET E=1
9248     LET KK=TRUNC.F(K)
9249     IF RANDOM.F(STREAM) > K-KK,
9250        LET K=KK
9251        GO TO A
9252     OTHERWISE ''                                          REPLACED ELSE WITH OTHERWISE \1
9253     LET K=KK+1
9254     'A'  FOR I=1 TO K, LET E=E*RANDOM.F(STREAM)
9255     IF E=0
9256        LET E=1
9257        GO TO A
9258     OTHERWISE ''                                          REPLACED ELSE WITH OTHERWISE \1
9259 <--RETURN WITH (MEAN/K)*(-LOG.E.F(E))
9260 ENDROUTINE
```

```
9261 PROCESS AIRBORNE.RADAR                                                    ''                                                    U002
9262    ''                                                                                                                        \UNNECESSARY?
9263
9264    ADD 1 TO ANAL.CTR(256,1)  ''                                                                                              \DYN_ANAL
9265    NORMALLY MODE IS INTEGER
9266    DEFINE AR TO MEAN PROCESS.V ''                                                                                            \2
9267    DEFINE TRANSMIT TO MEAN WAIT ''                                                                                           \FIX_LATER
9268    DEFINE SEARCH TO MEAN WAIT
9269    DEFINE FLY TO MEAN WAIT
9270    DEFINE TURN TO MEAN WAIT
9271    DEFINE PREPARE TO MEAN WAIT
9272    DEFINE UNIT, GROUPING, SIDE, SENSOR.TYPE AS INTEGER VARIABLES
9273    DEFINE  LEG.LENGTH, SEARCH.DIST, OLD.Y,
9274    RETURN.TIME,APPROACH.TIME,APPROACH.DISTANCE
9275    AS REAL VARIABLES
9276    LET SIDE = UN.COLOR(UNIT)
9277    LET MODEL = US.MODEL(LINK)
9278    LET SENSOR.TYPE = US.SENSOR.TYPE(LINK)
9279    LET VELOCITY = MAR.VELOCITY(MODEL)
9280    LET MIN.GRID.SQUARE = INT.F(MAR.MIN.SEARCH.RNG(MODEL)/64.)
9281    LET MAX.GRID.SQUARE = INT.F(MAR.MAX.SEARCH.RNG(MODEL)/64.)
9282    LET AR.X.GRID = AR.X.START(AR)/64.
9283    IF SIDE = RED
9284       LET MIN = AR.X.GRID - MAX.GRID.SQUARE
9285       LET MAX = AR.X.GRID - MIN.GRID.SQUARE
9286       LET ENEMY = BLUE
9287    ELSE
9288       LET MIN = AR.X.GRID + MIN.GRID.SQUARE
9289       LET MAX = AR.X.GRID + MAX.GRID.SQUARE
9290       LET ENEMY = RED
9291    ALWAYS
9292
9293    PREPARE UNIFORM.F(MAR.MIN.PREP(MODEL), MAR.MAX.PREP(MODEL),1) MINUTES
9294
9295    LET APPROACH.DISTANCE = SQRT.F((UN.X.COORD(AIR.STRIP)-FL.X.START)**2 +  ''                                                  \OPTIMIZE
9296    (UN.Y.COORD(AIR.STRIP)-FL.Y.START)**2)
9297    LET APPROACH.TIME = APPROACH.DISTANCE/VELOCITY
9298    LET RETURN.TIME = APPROACH.TIME
9299    LET END.MINUTE = TIME.V * MINUTES.V + MAR.MAX.ALOFT.TIME(MODEL)
9300    LET RETURN.MINUTE = END.MINUTE - RETURN.TIME
9301
9302    FLY APPROACH.TIME MINUTES
9303
9304    LET NUMBER.OF.LEGS = INT.F((RETURN.MINUTE-TIME.V * MINUTES.V)/
9305    ((LEG.LENGTH/VELOCITY)+2))
9306    CREATE A TARGET.REPORT CALLED TARGET
9307    LET DIRECTION = -1
9308    LOOP
9309    FOR I = 1 TO NUMBER.OF.LEGS
9310    DO THIS
9311       LET DIRECTION = -DIRECTION
9312       LOOP
9313       FOR EACH GROUPING
9314       DO THIS
9315          LOOP
9316          FOR EACH UN IN UNIT.SET(ENEMY,GROUPING)
9317          DO THIS
9318             IF UN IS IN AN AR.DET.TGT.LIST
```

```
9319          <------CYCLE
9320             OTHERWISE
9321
9322             IF MIN>=UN.X.GRID(UN) OR MAX<UN.X.GRID(UN)
9323          <------CYCLE
9324             OTHERWISE
9325
9326             ''ONLY LOOKING ON ONE SIDE OF A/C AT A TIME
9327             LET RANGE = ABS.F(UN.X.GRID(UN)-AR.X.GRID)
9328             LET PROB.LOS = .575 - .0079*1.024*RANGE
9329             IF RANDOM.F(1) > PROB.LOS
9330          <------CYCLE
9331             OTHERWISE
9332
9333             CREATE AN AR.DET.CANDIDATE CALLED CANDIDATE
9334             LET AR.DC.UNIT(CANDIDATE) = UN
9335             LET AR.DC.Y.COORD(CANDIDATE) = UN.Y.COORD(UN)
9336             * DIRECTION
9337             FILE CANDIDATE IN AR.CAND.DET.LIST(CANDIDATE)
9338          ENDLOOP
9339       ENDLOOP
9340       LET OLD.Y = 0
9341       LOOP
9342       FOR EACH CANDIDATE IN AR.CAND.DET.LIST(AR)
9343       DO THIS
9344          IF TIME.V * MINUTES.V >= RETURN.MINUTE
9345       <------CYCLE
9346          OTHERWISE
9347
9348          REMOVE CANDIDATE FROM AR.CAND.DET.LIST(AR)
9349          LET SEARCH.DIST = AR.DC.Y.COORD(CANDIDATE) - OLD.Y
9350          LET OLD.Y = AR.DC.Y.COORD(CANDIDATE)
9351
9352          SEARCH  SEARCH.DIST/VELOCITY MINUTES
9353
9354          CALL AR.DETECTION GIVEN AR, CANDIDATE, MODEL, TARGET,  ''------------------------------------ \UNNECESSARY?>(656)
9355          SENSOR.TYPE
9356          LET UN = AR.DC.UNIT(CANDIDATE)
9357          IF TR.DET.LIST(TARGET) IS EMPTY
9358       <------CYCLE
9359          OTHERWISE
9360
9361          DESTROY THE AR.DET.CANDIDATE CALLED CANDIDATE
9362          FILE UN IN AR.DET.TGT.LIST(AR)
9363
9364          TRANSMIT UNIFORM.F(REAL.F(ST.MIN.XMIT(SENSOR.TYPE)),
9365          REAL.F(ST.MAX.XMIT(SENSOR.TYPE)), 1) /10. MINUTES
9366
9367          LET TR.RECVD.TIME(TARGET) = TIME.V
9368          LET TR.ABORT.TIME(TARGET) = TIME.V + .5
9369          IF TIME.V * MINUTES.V < RETURN.MINUTE
9370             CREATE A TARGET.REPORT CALLED TARGET
9371          ALWAYS
9372       ENDLOOP
9373
9374       TURN 2. MINUTES
9375
9376    ENDLOOP
```

```
9377
9378     FLY RETURN.TIME MINUTES
9379
9380     UNTIL AR.DET.TGT.LIST(AR) IS EMPTY
9381     REMOVE THE FIRST UN FROM AR.DET.TGT.LIST(AR)
9382
9383 <--RETURN
9384 ENDPROCESS
```

```
9385 ROUTINE AR.DETECTION GIVEN AR,CANDIDATE,MODEL,TARGET,SENSOR.TYPE  ''                    U003
9386    ADD 1 TO ANAL.CTR(257,1)  ''                                                       \DYN_ANAL
9387    NORMALLY MODE IS INTEGER
9388    DEFINE  ELEM.PROB.DETECT AS A REAL VARIABLE
9389    DEFINE  SIDE, SENSOR.TYPE AS INTEGER VARIABLES
9390    DEFINE NAME AS A TEXT VARIABLE ''                                                  \TEXT
9391    '' IF DEBUG = TRUE,
9392       '' PRINT 1 LINE WITH AR, REC.TARGET THUS
9393       '' = = = AR.DETECTION  AR = ****** ,CANDIDATE = ******  = = =
9394    '' ENDIF
9395    LET UN=AR.DC.UNIT(CANDIDATE)
9396    LET AIR.STRIP = US.UNIT(LINK)
9397    LET MODEL = US.MODEL(LINK)
9398    CALL RANGE.COMPUTE GIVEN UN,AIR.STRIP YIELDING RANGE '' %22FEB79_%JEN----------->(342)
9399    LET SENSOR.TYPE = US.SENSOR.TYPE(LINK)
9400    LET SIDE = UN.COLOR( AIR.STRIP )
9401    LOOP FOR EACH EQUIP IN UN.EQUIP.LIST( UN )
9402    WHEN UE.QUANT( EQUIP ) > 0
9403    DO THIS
9404       LET EQUIP.TYPE = EQ.TE.PTR( UE.ID( EQUIP ) )
9405       ''UTILIZE AR.PROB.DETECT FUNCTION IN NEXT ASSIGNMENT ---------------------------- \1>(657)
9406       LET ELEM.PROB.DETECT = AR.PROB.DETECT(RANGE,MODEL)
9407       IF  ELEM.PROB.DETECT = 0.0
9408    <------CYCLE
9409       OTHERWISE
9410       LET NAME = TE.NAME(EQUIP.TYPE)
9411       IF NAME NE "TANK" AND
9412          NAME NE "APC" AND
9413          NAME NE "ICV" AND
9414          NAME NE "SP HOW"
9415    <------CYCLE
9416       OTHERWISE
9417       ADD UE.QUANT(EQUIP) TO QUANT
9418    ENDLOOP
9419    LET DETECT.QUANT = 0
9420    LET DETECT.QUANT= BINOMIAL.F(QUANT,ELEM.PROB.DETECT,1)  '' %22FEB79_%JEN
9421    IF DETECT.QUANT = 0
9422 <------EXITROUTINE
9423    OTHERWISE
9424    LET TR.TGT.UNIT(TARGET) = UN
9425    LET TR.SENSOR.TYPE(TARGET) = ST.NAME(SENSOR.TYPE)
9426    LET TR.SENSOR.ID(TARGET) = AR
9427    LET TR.FDC(TARGET) = US.FDC(LINK)
9428    LET TR.REP.UNIT(TARGET) = AIR.STRIP
9429    ''UTILIZE NORMAL.F FUNCTION IN NEXT 2 CALCULATIONS ------------------------------- \1>(641)
9430    LET TR.EST.X( TARGET ) = UN.X.COORD( UN ) +
9431    NORMAL.F(0.0,1.0,1) * TR.CEP(TARGET) / (1.1774 * 16.) ''                           \OPTIMIZE
9432    LET TR.EST.Y( TARGET ) = UN.Y.COORD( UN ) +
9433    NORMAL.F(0.0,1.0,1) * TR.CEP(TARGET) / (1.1774 * 16.)
9434    LET TR.PGM.STATUS(TARGET) = FALSE
9435    '' WRITE "///200", TIME.V,AR,SENSOR.TYPE,MODEL,"AR",
9436    '' RANGE, UN,UN.TYPE(UN), TR.FDC(TARGET),
9437    '' TARGET ,TR.MOVE(TARGET) AS BINARY USING UNIT 7
9438    CREATE A TR.DET.LINK CALLED TDL
9439    FILE TDL IN TR.DET.LIST(TARGET)
9440    LET TR.DET.TE(TDL) = 0
9441    LET TR.DET.ELEM.PROB(TDL) = INT.F(100. * ELEM.PROB.DETECT)
9442 ENDROUTINE
```

```
9443 FUNCTION AR.PROB.DETECT(RANGE,MODEL)                                    ''
9444                                                                                U004
9445    ADD 1 TO ANAL.CTR(258,1)  ''                                        \DYN_ANAL
9446    NORMALLY MODE IS INTEGER
9447    DEFINE PROB, R, RANGE, ' AS REAL VARIABLES
9448
9449    LET R = RANGE*.0001
9450    IF MAR.VELOCITY(MODEL) >= 75
9451       LET V=1
9452    ELSE
9453       IF MAR.VELOCITY(MODEL) >= 41
9454          LET V=2
9455       ELSE
9456          IF MAR.VELOCITY(MODEL) >= 21
9457             LET V=3
9458          ELSE
9459             LET V=4
9460          ALWAYS
9461       ALWAYS
9462    ALWAYS
9463    LET S=R
9464    LET PROB = AR.DET.COEFF(V,1)*EXP.F(S*AR.DET.COEFF(V,2))
9465    + AR.DET.COEFF(V,3)
9466    LOOP FOR I = 4 TO 7
9467    DO
9468       LET S=R*S
9469       ADD S*AR.DET.COEFF(V,I) TO PROB
9470    ENDLOOP
9471
9472 <--RETURN WITH PROB
9473 ENDFUNCTION
```

```
9474 PROCESS PHOTO.IR.FLIGHT                                                    ''
9475    ADD 1 TO ANAL.CTR(259,1)  ''                                                \UNNECESSARY? U005
9476    NORMALLY MODE IS INTEGER                                                      \DYN_ANAL
9477    DEFINE PIR TO MEAN PROCESS.V ''
9478    DEFINE TRANSMIT TO MEAN WAIT  ''                                            \2
9479    DEFINE SEARCH TO MEAN WAIT                                                \FIX_LATER
9480    DEFINE ANALYZE TO MEAN WAIT
9481    DEFINE UNIT, GROUPING, SIDE, SENSOR.TYPE AS INTEGER VARIABLES
9482    DEFINE LEG.SLOPE, LEG.Y.INTERCEPT , LEG.LENGTH,
9483    X.PERPENDICULAR, Y.PERPENDICULAR,
9484    PERPENDICULAR.DIST, HALF.WIDTH,
9485    FRACT.COVERED, BACK.DIST, FORWARD.DIST AS REAL VARIABLES
9486    '' IF DEBUG = TRUE,
9487       '' PRINT 1 LINE WITH PIR THUS
9488       '' = = = PHOTO.IR.FLIGHT  PIR = ******  = = =
9489    '' ENDIF
9490    LET LINK = PIR.US.LINK(PIR)
9491    LET UNIT = US.UNIT(LINK)
9492    LET SIDE = UN.COLOR(UNIT)
9493    LET MODEL = US.MODEL(LINK)
9494    LET SENSOR.TYPE = US.SENSOR.TYPE(LINK)
9495    LET VELOCITY = MPIR.VELOCITY(MODEL)
9496    IF SIDE = RED
9497       LET ENEMY = BLUE
9498    ELSE
9499       LET ENEMY = RED
9500    ENDIF
9501    LET FL.Y.START = PIR.Y.START(PIR)
9502    LET FL.X.START = PIR.X.START(PIR)
9503    LET HALF.WIDTH = MPIR.HALF.COV.WIDTH(MODEL)
9504    LOOP FOR EACH LEG IN PIR.FLIGHT.LEG.LIST(PIR)
9505    DO THIS ''                                                                  \1
9506       LOOP UNTIL PIR.FLIGHT.LEG.LIST(PIR) IS EMPTY
9507       DO THIS ''                                                               \1
9508          REMOVE LEG FROM PIR.FLIGHT.LEG.LIST(PIR)
9509          LET DELTA.Y = FL.Y.END(LEG) - FL.Y.START
9510          LET DELTA.X = FL.X.END(LEG) - FL.X.START
9511          LET LEG.SLOPE = DELTA.Y/DELTA.X
9512          LET LEG.Y.INTERCEPT = FL.Y.START - LEG.SLOPE * FL.X.START
9513          LET LEG.LENGTH = SQRT.F(DELTA.X ** 2 + DELTA.Y ** 2)  ''               \OPTIMIZE
9514          SEARCH LEG.LENGTH/VELOCITY MINUTES
9515          LOOP FOR EACH GROUPING
9516          DO THIS ''                                                            \1
9517             LOOP FOR EACH UN IN UNIT.SET(ENEMY,GROUPING),
9518             ''UTILIZE STAY.TIME FUNCTION IN CONTROLLING LOOP EXECUTION ---------------------------- \1>(660)
9519             WHEN STAY.TIME(UN) > MPIR.MAX.PROCESS(MODEL) AND
9520             UN.STATUS(UN) NE STA.TO.WITH AND
9521             UN.STATUS(UN) NE ADV.TO.WITH
9522             DO THIS
9523                CREATE AN PIR.REC.TARGET CALLED REC.TARGET
9524                LET X.PERPENDICULAR = (UN.Y.COORD(UN) + UN.X.COORD(UN)/LEG.SLOPE -
9525                LEG.Y.INTERCEPT) / (LEG.SLOPE + 1./LEG.SLOPE)
9526                LET Y.PERPENDICULAR = LEG.SLOPE*X.PERPENDICULAR+LEG.Y.INTERCEPT
9527                LET PERPENDICULAR.DIST = 10. * SQRT.F((X.PERPENDICULAR -
9528                UN.X.COORD(UN))**2.+(Y.PERPENDICULAR - UN.Y.COORD(UN))**2 )
9529                LET BACK.DIST = SQRT.F((X.PERPENDICULAR-FL.X.START) ** 2 +
9530                (Y.PERPENDICULAR-FL.Y.START) ** 2 )
9531                LET FORWARD.DIST = SQRT.F((X.PERPENDICULAR-FL.X.END(LEG)) ** 2 +
```

```
9532                (Y.PERPENDICULAR-FL.Y.END(LEG)) ** 2 )
9533                IF BACK.DIST+FORWARD.DIST > LEG.LENGTH
9534             <------CYCLE
9535                OTHERWISE
9536                CALL SEARCH.COVERAGE GIVEN UN, HALF.WIDTH, PERPENDICULAR.DIST, LEG.SLOPE,------------------------------>(343)
9537                LEG.Y.INTERCEPT, AND UN.RADIUS(UN)    YIELDING FRACT.COVERED
9538                IF FRACT.COVERED = 0.0
9539             <------CYCLE
9540                OTHERWISE
9541                CALL PIR.DETECTION GIVEN PIR, UN, FRACT.COVERED, AND REC.TARGET------------------------------------------>(212)
9542                IF PIR.RTD.LIST(TARGET) IS EMPTY
9543                   DESTROY THE PIR.REC.TARGET CALLED REC.TARGET
9544             <------CYCLE
9545                OTHERWISE
9546                LET PIR.RT.UNIT(REC.TARGET) = UN
9547                FILE REC.TARGET IN PIR.RECORD.LIST(PIR)
9548             ENDLOOP
9549             LET FL.Y.START = FL.Y.END(LEG)
9550             LET FL.X.START = FL.X.END(LEG)
9551             DESTROY THE FLIGHT.LEG CALLED LEG
9552          ENDLOOP
9553       ENDLOOP
9554       LET PROCESS.TIME=UNIFORM.F(MPIR.MIN.PROCESS(MODEL),MPIR.MAX.PROCESS(MODEL),1)
9555       ANALYZE PROCESS.TIME MINUTES
9556       CREATE A TARGET.REPORT CALLED TARGET
9557       LOOP FOR EACH REC.TARGET IN PIR.RECORD.LIST(PIR)
9558       DO THIS ''                                                                                              \1
9559          REMOVE REC.TARGET FROM PIR.RECORD.LIST(PIR)
9560          LOOP UNTIL PIR.RTD.LIST(REC.TARGET) IS EMPTY
9561          DO THIS
9562             REMOVE THE FIRST DET.LINK FROM PIR.RTD.LIST(REC.TARGET)
9563             FILE DET.LINK IN TR.DET.LIST(TARGET)
9564          ENDLOOP
9565          DESTROY THE PIR.REC.TARGET CALLED REC.TARGET
9566          TRANSMIT UNIFORM.F(REAL.F(ST.MIN.XMIT(SENSOR.TYPE)) ,
9567          REAL.F(ST.MAX.XMIT(SENSOR.TYPE)) , 1)/10. MINUTES
9568          LET TR.RECVD.TIME(TARGET) = TIME.V
9569          LET TR.ABORT.TIME(TARGET) = TIME.V + .5
9570          LET TGT = PIR.RT.UNIT(REC.TARGET)
9571          LET TR.TGT.UNIT(TARGET) = TGT
9572          LET TR.SENSOR.TYPE(TARGET) = ST.NAME(SENSOR.TYPE)
9573          LET TR.SENSOR.ID(TARGET) = PIR
9574          LET TR.FDC(TARGET) = US.FDC(LINK)
9575          LET TR.REP.UNIT(TARGET) = US.UNIT(LINK)
9576          LET TR.CEP(TARGET) = MPIR.CIR.ERROR(MODEL)
9577          ''UTILIZE NORMAL.F FUNCTION IN NEXT 2 CALCULATIONS ------------------------------------------------------ \1>(641)
9578          LET TR.EST.X(TARGET) = UN.X.COORD(TGT)+NORMAL.F(00.0,1.0,1)*
9579          TR.CEP(TGT)/(1.1774*16.)
9580          LET TR.EST.Y(TARGET) = UN.Y.COORD(TGT)+NORMAL.F(00.0,1.0,1)*
9581          TR.CEP(TGT)/(1.1774*16.)
9582          LET TR.PGM.STATUS(TARGET) = FALSE
9583          '' WRITE "///200", TIME.V, PIR, SENSOR.TYPE, MODEL, "PIR", RANGE, UN,
9584          ''  UN.TYPE(UN), TR.FDC(TARGET), TARGET, TR.MOVE(TARGET) AS BINARY USING UNIT 7
9585          ACTIVATE THE TARGET.REPORT CALLED TARGET NOW------------------------------------------------------------>(470)
9586          CREATE A TARGET.REPORT CALLED TARGET
9587       ENDLOOP
9588    ENDLOOP
9589 ENDPROCESS
```

```
9590 FUNCTION STAY.TIME                                                                    U006
9591     GIVEN
9592     UNIT
9593
9594     ADD 1 TO ANAL.CTR(260,1)                                                          \DYN_ANAL
9595     NORMALLY MODE IS INTEGER
9596     DEFINE UNIT, SIDE, SECTOR AS INTEGER VARIABLES
9597     DEFINE FEBA.DIST AS A REAL VARIABLE
9598
9599     LET SIDE = UN.COLOR(UNIT)
9600     CALL LOCATE.SECTORPG                                                  \ROUTINE.NAME?>(    )
9601     GIVEN
9602     UN.Y.COORD(UNIT)
9603     YIELDING
9604     SECTOR
9605     ''UTILIZE ACT.RANGE FUNCTION IN NEXT ASSIGNMENT                                   \1>(628)
9606     LET FEBA.DIST = ABS.F(ACT.RANGE(F.SS.SET(SIDE,SECTOR),UNIT))
9607     ''1KM - 3 MINUTES
9608     ''5KM - 15 MINUTES
9609     ''10KM - 30 MINUTES
9610     ''20KM - 60 MINUTES
9611     ''40KM - 120 MINUTES
9612
9613 <--RETURN WITH FEBA.DIST/1250
9614 ENDFUNCTION
```

```
9615 ROUTINE JOHNSON.CRITERIA                                                    U007
9616     YIELDING
9617     .NO.BARS
9618
9619     ADD 1 TO ANAL.CTR(261,1)  ''                                    \DYN_ANAL
9620     NORMALLY MODE IS REAL
9621
9622     LET .NO.BARS = 1.  ''DISCRIMINATION LEVEL                       \REPLACE?
9623
9624 <—EXITROUTINE
9625 ENDROUTINE
```

```
9626 ROUTINE PROXIMITY.REQ                                    ''          \REQUIRED? U008
9627     GIVEN
9628     TARGET,
9629     TR
9630     YIELDING
9631     REQ.PROXIMITY
9632
9633     ADD 1 TO ANAL.CTR(262,1)  ''                                     \DYN_ANAL
9634     NORMALLY MODE IS INTEGER
9635
9636     LET REQ.PROXIMITY = 5  ''                                        \OPTIMIZE
9637
9638 <--EXITROUTINE
9639 ENDROUTINE
```

```
9640 ROUTINE TIME.REQ                                                    ''      \REQUIRED? U009
9641     GIVEN
9642     TARGET,
9643     TR
9644     YIELDING
9645     REQ.TIME
9646
9647     ADD 1 TO ANAL.CTR(263,1)  ''                                            \DYN_ANAL
9648     NORMALLY MODE IS INTEGER
9649     DEFINE REQ.TIME AS A REAL VARIABLE
9650
9651     LET REQ.TIME = .10
9652
9653 <--EXITROUTINE
9654 ENDROUTINE
```

```
9655 ''PROGRAM  OLDER.VERSION'' PREAMBLE ''                    DATA STRUCTURE FOR COSAGE (ORIGINALLY 2848 LINES) \1  U010
9656
9657
9658    NORMALLY MODE IS INTEGER
9659    DEFINE HELICOPTER TO MEAN HELI.COPTER
9660    DEFINE TR.START.TIME TO MEAN TR.RECVD.TIME
9661
9662    GENERATE LIST ROUTINES
9663
9664    ''*************************************************************
9665    ''*                                                           *
9666    ''*            COSAGE I   PROGRAMMERS                         *
9667    ''*                                                           *
9668    ''*    MR   GLENN    R. STOCKTON    APR 77 - AUG 78           *
9669    ''*    MAJ  ROBERT   J. HILL  III   APR 77 - OCT 78           *
9670    ''*    MAJ  RICHARD     PORTER      APR 77 - JUL 78           *
9671    ''*    MAJ  ROBERT   T. CAMPBELL    APR 77 - DEC 78           *
9672    ''*    MR   GARY     L. MARTIN      APR 77 - JUN 79    GLM    *
9673    ''*    MR   JOE      E. NICHOLS     OCT 78 -           JN & JEN*
9674    ''*    LTC  EARL        DARDEN      SEP 77 - JUL 80    ED     *
9675    ''*    LTC  JOHN     L. RAFFERTY    JUL 78 - OCT 80    JLR    *
9676    ''*    MAJ  RONALD   G. RHOADES     DEC 78 - APR 80    RGR    *
9677    ''*    MS   PAMELA   J. KENNEDY     JUN 79 - AUG 79    PJK    *
9678    ''*    MR   ANDREW   J. CARRAS      OCT 79 - JUN 80    AJK    *
9679    ''*    MR   HUGH     W. JONES       JAN 80 -           HWJ    *
9680    ''*    MAJ  ROBERT   W. FEISZLI     APR 80 -           RWF    *
9681    ''*    CPT  JAMES    S. KRAMER      JUL 80 -           JSK    *
9682    ''*    MR   DANIEL   J. SHEDLOWSKI  SEP 80 -           DJS    *
9683    ''*    CPT  ANDREW   G. LOERCH      OCT 80 -           AGL    *
9684    ''*    MAJ  FRANK    T. PROCTOR JR. OCT 80 -           FTP    *
9685    ''*                                                           *
9686    ''*************************************************************
9687
9688
9689
9690    ''**************************PERMANENT ENTITIES* SECTION ONE ***************
9691
9692
9693    PERMANENT ENTITIES
9694    INCLUDE TB.N.FM,
9695    NITE.OR.DAY,
9696    MOVEMENT.STATUS
9697
9698    EVERY AC.MUNS HAS
9699    AN AM.NAME              IN ARRAY 217,
9700    AN AM.RELY        (1/2) IN ARRAY 218,
9701    AN AM.RADIUS       (2/2) IN ARRAY 218
9702
9703    EVERY AC.MUNS, ENVIRONMENT, POSTURE HAS
9704    AN AMEP.LA.PERS             IN ARRAY 219
9705
9706    EVERY AC.MUNS, ENVIRONMENT, TYPE.EQUIPMENT HAS
9707    AN AMET.LA.EQUIP    IN ARRAY 220
9708
9709    EVERY AC.TYPE HAS
9710    AN ACT.EQUIP.ID                    (1/3) IN ARRAY 221,
9711    AN ACT.SUBSTITUTE                   (2/3) IN ARRAY 221,
9712    AN ACT.WEATHER.DEGRADE              (3/3) IN ARRAY 221,
```

```
9713    AN ACT.MIN.ALT                  (1/3) IN ARRAY 222,
9714    AN ACT.NORM.ALT                 (2/3) IN ARRAY 222,
9715    AN ACT.BAI.TA.DELAY             (5/6) IN ARRAY 222,  ''MINUTES
9716    AN ACT.SPEED                    (1/3) IN ARRAY 223,  ''HDM/SEC
9717    AN ACT.PASS.TIME                (2/3) IN ARRAY 223,  ''MINUTES
9718    AN ACT.MAX.ALOFT                (3/3) IN ARRAY 223,  ''MINUTES
9719    AN ACT.PROB.SORTIE.ABORT        (1/3) IN ARRAY 224,
9720    AN ACT.NITE.FLY                 (4/6) IN ARRAY 224,
9721    AN ACT.MIN.PREP.TIME            (6/6) IN ARRAY 222,  ''MINUTES
9722    AN ACT.MAX.PREP.TIME            (3/6) IN ARRAY 224,  ''MINUTES
9723    AN ACT.P1.DIST                  (1/3) IN ARRAY 225,
9724    AN ACT.P2.DIST                  (3/3) IN ARRAY 224,
9725    AN ACT.P3.DIST                  (2/3) IN ARRAY 225,
9726    AN ACT.ANGLE.P1.P2                    IN ARRAY 226,
9727    AN ACT.X1                       (3/3) IN ARRAY 225,
9728    AN ACT.Y1                       (1/3) IN ARRAY 227,
9729    AN ACT.Z1                       (2/3) IN ARRAY 227,
9730    AN ACT.X2                       (3/3) IN ARRAY 227,
9731    AN ACT.Y2                       (1/3) IN ARRAY 228,
9732    AN ACT.Z2                       (2/3) IN ARRAY 228,
9733    AN ACT.X3                       (3/3) IN ARRAY 228,
9734    AN ACT.Y3                       (1/3) IN ARRAY 229,
9735    AN ACT.Z3                       (2/3) IN ARRAY 229
9736
9737    EVERY AC.TYPE, AC.MUNS HAS
9738    AN ATM.DELIV.CEP     IN ARRAY 230
9739
9740    EVERY AO.ELEVATION.BAND HAS
9741    AN AO.EB.ALTITUDE(2/2) IN ARRAY 20
9742    BELONGS TO
9743    THE AO.EB.SET
9744    HAS
9745    A P.AO.EB.SET(1/6) IN ARRAY 20,
9746    A S.AO.EB.SET(2/6) IN ARRAY 20,
9747    A M.AO.EB.SET(3/6) IN ARRAY 20
9748
9749    EVERY AO.ELEVATION.BAND, AO.RANGE.BAND  HAS
9750    AN AO.PROB.LOS(1/2) IN ARRAY 2,
9751    AN AO.VISIBILITY(2/2) IN ARRAY 2
9752
9753    EVERY AO.RANGE.BAND HAS
9754    AN AO.RB.RANGE(2/2) IN ARRAY 29
9755    BELONGS TO
9756    THE AO.RB.SET
9757    HAS
9758    A P.AO.RB.SET(1/6) IN ARRAY 29,
9759    A S.AO.RB.SET(2/6) IN ARRAY 29,
9760    A M.AO.RB.SET(3/6) IN ARRAY 29
9761
9762    EVERY BTRY HAS
9763    A BY.BN(1-9) IN ARRAY 5,
9764    A BY.STATUS(10-12) IN ARRAY 5,
9765    A BY.TYPE(3/6) IN ARRAY 5,
9766    A BY.PGM.FM(1/2) IN ARRAY 6,
9767    A BY.CUR.FM(2/2) IN ARRAY 6,
9768    A BY.N.ROUNDS(2/2) IN ARRAY 5,
9769    A BY.UNIT(1/3) IN ARRAY 7,
9770    A BY.BN.RANK(2/3) IN ARRAY 7,
```

```
9771    A BY.FIRE.RATE(3/3) IN ARRAY 7,
9772    A BY.PGM.CAP(3/6) IN ARRAY 1,
9773    A BY.STOP.FASCAM.SUPP(1/3) IN ARRAY 1  ''TIME.V + 60
9774    OWNS
9775    A BY.HOW.SET,
9776    A BY.SCHD.LIST,
9777    A BY.FM.QUEUE
9778    BELONGS TO
9779    A BN.BTRY.SET
9780    HAS
9781    A F.BY.HOW.SET(1/2) IN ARRAY 97,
9782    A L.BY.HOW.SET(2/2) IN ARRAY 97,
9783    A F.BY.SCHD.LIST(1/2) IN ARRAY 112,
9784    A L.BY.SCHD.LIST(2/2) IN ARRAY 112,
9785    A F.BY.FM.QUEUE(1/2) IN ARRAY 123,
9786    A L.BY.FM.QUEUE(2/2) IN ARRAY 123,
9787    A P.BN.BTRY.SET(1/2) IN ARRAY 124,
9788    A S.BN.BTRY.SET(2/2) IN ARRAY 124,
9789    A M.BN.BTRY.SET(1/6) IN ARRAY 125,
9790    A N.BY.HOW.SET(2/6) IN ARRAY 125,
9791    A N.BY.SCHD.LIST(3/6) IN ARRAY 125,
9792    A N.BY.FM.QUEUE(4/6) IN ARRAY 125
9793
9794    EVERY CATEGORY HAS
9795    A CT.NAME IN ARRAY 14,
9796    A CT.GROUP(1/2) IN ARRAY 15,
9797    A CT.MIN.FEBA(2/2) IN ARRAY 15
9798    OWNS
9799    A CT.TU.SET
9800    BELONGS TO
9801    A GP.CAT.SET
9802    HAS
9803    A F.CT.TU.SET(1/3) IN ARRAY 126,
9804    A L.CT.TU.SET(2/3) IN ARRAY 126,
9805    A N.CT.TU.SET(3/3) IN ARRAY 126,
9806    A P.GP.CAT.SET(1/3) IN ARRAY 127,
9807    A S.GP.CAT.SET(2/3) IN ARRAY 127,
9808    A M.GP.CAT.SET(6/6) IN ARRAY 127
9809
9810    EVERY CATEGORY,DIST.FROM.FEBA.BAND, IC.MUNITION HAS
9811    A CDI.USAGE.INDICATOR IN ARRAY 194
9812
9813    EVERY CATEGORY, DIST.FROM.FEBA.BAND, TYPE.BTRY HAS
9814    A CDT.MAX.VOLS IN ARRAY 193
9815
9816    EVERY CATEGORY , POSTURE , MISSION   HAS
9817    A CPM.WARNED.FRACT(1/2) IN ARRAY 3,
9818    A CPM.UNWARNED.FRACT(2/2) IN ARRAY 3
9819
9820    EVERY CFR.RNG.HACK HAS
9821    A CFR.RH.RANGE(1/3) IN ARRAY 8,
9822    A CFR.DET.PROB(2/3) IN ARRAY 8,
9823    A CFR.CIR.ERROR(3/3) IN ARRAY 8
9824    BELONGS TO
9825    A MCFR.RH.LIST
9826    HAS
9827    A P.MCFR.RH.LIST(1/6) IN ARRAY 128,
9828    A S.MCFR.RH.LIST(2/6) IN ARRAY 128,
```

```
9829    A M.MCFR.RH.LIST(3/6) IN ARRAY 128
9830
9831    EVERY DIST.FROM.FEBA.BAND HAS
9832    A DFFB.MAX.RANGE IN ARRAY 192
9833
9834    EVERY DIST.FROM.FEBA.BAND, TYPE.UNIT HAS
9835    A DT.MAX.BATS IN ARRAY 195
9836
9837    EVERY ENVIRONMENT HAS
9838    A EN.NAME IN ARRAY 22
9839
9840    EVERY ENVIRONMENT, CATEGORY  HAS
9841    AN EC.FRACT IN ARRAY 23
9842
9843    EVERY ENVIRONMENT, SUBMUNITION HAS
9844    A ES.RELY IN ARRAY 24
9845
9846    EVERY  ENVIRONMENT , POSTURE , SUBMUNITION HAS
9847    A EPS.LA.PERS IN ARRAY 25
9848
9849    EVERY EQUIPMENT HAS
9850    AN EQ.AD.INDICATOR    (2/6) IN ARRAY 10,
9851    A EQ.NAME IN ARRAY 26,
9852    A EQ.KV.ID,
9853    A EQ.TE.PTR(1/6) IN ARRAY 10,
9854    A EQ.MAX.SPEED(2/3) IN ARRAY 10,
9855    A EQ.PERSONNEL.LOAD(3/3) IN ARRAY 10,
9856    A EQUIP.PK.PTR(1/3) IN ARRAY 186
9857    BELONGS TO
9858    A TE.SET
9859    HAS
9860    A P.TE.SET(1/3) IN ARRAY 129,
9861    A S.TE.SET(2/3) IN ARRAY 129,
9862    A M.TE.SET(6/6) IN ARRAY 129
9863
9864    EVERY FASCAM.MUNITION HAS
9865    A FMM.ID              IN ARRAY 85,  ''ALPHA NAME
9866    A FMM.MAX.RANGE(1/3) IN ARRAY 86,  ''HDM
9867    A FMM.RND.WT(2/3)     IN ARRAY 86   ''POUNDS
9868
9869    EVERY FA.BN HAS
9870    A FB.MISSION IN ARRAY 30,
9871    A FA.BN.UNIT IN ARRAY 204
9872    OWNS
9873    A BN.BTRY.SET
9874    HAS
9875    A F.BN.BTRY.SET(1/3) IN ARRAY 130,
9876    A L.BN.BTRY.SET(2/3) IN ARRAY 130,
9877    A N.BN.BTRY.SET(6/6) IN ARRAY 130
9878
9879    EVERY FDC HAS
9880    A FD.FDC(1/3) IN ARRAY 11,
9881    A FD.MIN.TIME(2/3) IN ARRAY 11,
9882    A FD.MAX.TIME(3/3) IN ARRAY 11,
9883    A FD.CUR.TR(1/2) IN ARRAY 12,
9884    A FD.N.PROCESSED(2/2) IN ARRAY 12,
9885    A FD.N.LOST(1/2) IN ARRAY 13,
9886    A FD.TOT.THRESHOLD(2/2) IN ARRAY 13
```

```
9887    OWNS
9888    A FD.TR.QUEUE,
9889    A FD.COMPLETE.LIST,
9890    A FD.SCHD.LIST,
9891    A FD.BN.LIST
9892    HAS
9893    A F.FD.TR.QUEUE(1/2) IN ARRAY 131,
9894    A L.FD.TR.QUEUE(2/2) IN ARRAY 131,
9895    A F.FD.COMPLETE.LIST(1/2) IN ARRAY 132,
9896    A L.FD.COMPLETE.LIST(2/2) IN ARRAY 132,
9897    A F.FD.SCHD.LIST(1/2) IN ARRAY 133,
9898    A L.FD.SCHD.LIST(2/2) IN ARRAY 133,
9899    A F.FD.BN.LIST(1/2) IN ARRAY 134,
9900    A L.FD.BN.LIST(2/2) IN ARRAY 134,
9901    A N.FD.TR.QUEUE(1/6) IN ARRAY 135,
9902    A N.FD.COMPLETE.LIST(2/6) IN ARRAY 135,
9903    A N.FD.SCHD.LIST(3/6) IN ARRAY 135,
9904    A N.FD.BN.LIST(4/6) IN ARRAY 135
9905
9906    EVERY FO.RANGE.BAND HAS
9907    A FO.RB.RANGE(2/2) IN ARRAY 136,
9908    A FO.VISIBILITY (1/2) IN ARRAY 54,
9909    A FO.CIR.ERROR(2/2) IN ARRAY 54
9910    BELONGS TO
9911    A MFO.RB.SET
9912    HAS
9913    A P.MFO.RB.SET(1/6) IN ARRAY 136,
9914    A S.MFO.RB.SET(2/6) IN ARRAY 136,
9915    A M.MFO.RB.SET(3/6) IN ARRAY 136
9916
9917    EVERY FUZE HAS
9918    A FZ.NAME IN ARRAY 39
9919
9920    EVERY FUZE , HE.MUNITION HAS
9921    A FZ.HE.RELY IN ARRAY 16
9922
9923    EVERY GROUPING HAS
9924    A GP.NAME IN ARRAY 40
9925    OWNS
9926    A GP.CAT.SET
9927    HAS
9928    A F.GP.CAT.SET(1/6) IN ARRAY 137,
9929    A L.GP.CAT.SET(2/6) IN ARRAY 137,
9930    A N.GP.CAT.SET(3/6) IN ARRAY 137
9931
9932    EVERY HE.MUNITION HAS
9933    A HE.ID          IN ARRAY 41,
9934    A HE.WEIGHT(1/3) IN ARRAY 17,
9935    A HE.COST(2/3) IN ARRAY 17,
9936    A HE.VOLLEY.RAD(3/3) IN ARRAY 17,
9937    A HE.ROUND.RAD(1/3) IN ARRAY 18,
9938    A HE.VOL.DUST.RAD (2/3) IN ARRAY 18,   ''HDM
9939    A HE.DUST.DURATION (3/3) IN ARRAY 18,  ''MINUTES
9940    A HE.MIN.MARG.EFF IN ARRAY 241  ''MINUTES
9941
9942    EVERY HE.MUNITION, TYPE.BTRY OWNS
9943    A HE.TB.RH.LIST
9944    HAS
```

```
9945    A F.HE.TB.RH.LIST(1/6) IN ARRAY 138,
9946    A L.HE.TB.RH.LIST(2/6) IN ARRAY 138,
9947    A N.HE.TB.RH.LIST(3/6) IN ARRAY 138
9948
9949    EVERY HE.RANGE.HACK HAS
9950    A HE.RH.RANGE(1/3) IN ARRAY 19,
9951    A HE.RH.TOTAL.CPE(2/3) IN ARRAY 19,
9952    A HE.RH.ROUND.CPE(3/3) IN ARRAY 19
9953    BELONGS TO
9954    A HE.TB.RH.LIST
9955    HAS
9956    A P.HE.TB.RH.LIST(1/3) IN ARRAY 139,
9957    A S.HE.TB.RH.LIST(2/3) IN ARRAY 139,
9958    A M.HE.TB.RH.LIST(5/6) IN ARRAY 139
9959
9960    EVERY HE.RANGE.HACK, ENVIRONMENT, POSTURE, FUZE HAS
9961    AN REPF.LA.PERS(*/2) IN ARRAY 50
9962
9963    EVERY HE.RANGE.HACK, TYPE.EQUIPMENT, ENVIRONMENT, FUZE HAS
9964    AN RTEF.LA.EQUIP(*/2) IN ARRAY 51
9965
9966    EVERY IC.MUNITION HAS
9967    A IC.ID IN ARRAY 52,
9968    A IC.WEIGHT(2/2) IN ARRAY 28,
9969    A IC.COST(1/2) IN ARRAY 28,
9970    A IC.RELIABILITY(2/3) IN ARRAY 27,
9971    A IC.SUBM.INDEX(1/3) IN ARRAY 27,
9972    A IC.VOLLEY.RAD(3/3) IN ARRAY 27,
9973    A IC.N.SUBM(1/2) IN ARRAY 200,
9974    A IC.MIN.MARG.EFF(2/2) IN ARRAY 200
9975
9976    EVERY IC.MUNITION, TYPE.BTRY HAS
9977    A IC.TB.SLOPE IN ARRAY 59,
9978    A IC.TB.INTERCEPT IN ARRAY 60
9979    OWNS
9980    A IC.TB.RH.LIST
9981    HAS
9982    A F.IC.TB.RH.LIST(1/6) IN ARRAY 140,
9983    A L.IC.TB.RH.LIST(2/6) IN ARRAY 140,
9984    A N.IC.TB.RH.LIST(3/6) IN ARRAY 140
9985
9986    EVERY IC.RANGE.HACK HAS
9987    A IC.RH.RANGE(1/3) IN ARRAY 21,
9988    A IC.RH.TOTAL.CPE(2/3) IN ARRAY 21,
9989    A IC.RH.ROUND.CPE(3/3) IN ARRAY 21
9990    BELONGS TO
9991    A IC.TB.RH.LIST
9992    HAS
9993    A P.IC.TB.RH.LIST(1/6) IN ARRAY 141,
9994    A S.IC.TB.RH.LIST(2/6) IN ARRAY 141,
9995    A M.IC.TB.RH.LIST(3/6) IN ARRAY 141
9996
9997    EVERY ILLUM.MUNITION HAS
9998    AN ILLUM.ID IN ARRAY 64,
9999    AN ILLUM.RADIUS (1/3) IN ARRAY 32,  ''HDM
   0    AN ILLUM.MAX.RANGE(2/3) IN ARRAY 32, '' HDM
   1    AN ILLUM.DURATION(3/3) IN ARRAY 32, ''MINUTES
   2    AN ILLUM.RND.WT(1/3)   IN ARRAY 31  ''POUNDS
```

```
          'IEFIELD HAS
                                  IN ARRAY 113,
 6    A MF.COLOR          (1/3) IN ARRAY 114,
 7    A MF.X.HIGH         (1/2) IN ARRAY 115,
 8    A MF.X.LOW          (2/2) IN ARRAY 115,
 9    A MF.Y.HIGH         (1/2) IN ARRAY 116,
10    A MF.Y.LOW          (2/2) IN ARRAY 116
11    AND OWNS
12    A  MFP.LIST,
13    AN AU.LIST
14    AND HAS
15    A F.MFP.LIST              IN ARRAY 240,
16    A L.MFP.LIST              IN ARRAY 205,
17    A N.MFP.LIST        (2/3) IN ARRAY 114,
18    A F.AU.LIST               IN ARRAY 206,
19    A L.AU.LIST               IN ARRAY 207,
20    A N.AU.LIST         (3/3) IN ARRAY 114
21
22    EVERY MISSION HAS
23    A MN.NAME IN ARRAY 68
24
25    EVERY MISSION,COLOR HAS
26    A DECISION(1/2) IN ARRAY 69,
27    A SUP.MISSION.PRIORITY(2/2) IN ARRAY 69,
28    A WD.DIST (1/2) IN ARRAY 189
29
30    EVERY MODEL.AD.SENSOR HAS
31    A MADS.NAME                                IN ARRAY 231,
32    A MADS.DELAY.TIME                    (1/3) IN ARRAY 232,  ''SECONDS
33    A MADS.PW.DEGRADE                    (2/3) IN ARRAY 232,
34    A MADS.FCM                           (5/6) IN ARRAY 232,
35    A MADS.RIPL                          (1/3) IN ARRAY 233,  ''SECONDS
36    A MADS.WPN.RELOAD.TIME               (2/3) IN ARRAY 233,  ''MINUTES
37    A MADS.XMIT.PCT                      (3/3) IN ARRAY 233,
38    A MADS.DETECT                        (1/6) IN ARRAY 249,
39    A MADS.RDY.RDS                       (1/2) IN ARRAY 248,
40    A MADS.RESUP.TIME                    (2/2) IN ARRAY 248
41    AND OWNS
42    A MADS.RH.SET
43    AND HAS
44    A F.MADS.RH.SET               (1/2) IN ARRAY 234,
45    A L.MADS.RH.SET               (2/2) IN ARRAY 234,
46    A N.MADS.RH.SET               (6/6) IN ARRAY 232
47
48    EVERY MODEL.AO HAS   ''TIME REFERENCE IS MINUTES
49    A MAO.NAME IN ARRAY 70,
50    A MAO.VELOCITY(1/3) IN ARRAY 34,    '' % HDM PER HOUR
51    A MAO.ALTITUDE(2/3) IN ARRAY 34,    '' % DECCAMETERS
52    A MAO.MAGNIFICATION(3/3) IN ARRAY 34,  '' %     X POWER
53    A MAO.MIN.PREP  IN ARRAY 35,     '' % DECIMAL MINUTES (REAL)
54    A MAO.MAX.PREP  IN ARRAY 36,     '' % DECIMAL MINUTES (REAL)
55    A MAO.MAX.ALOFT.TIME(1/2) IN ARRAY 37,      '' % MINUTES (INTEGER)
56    A MAO.PGM.CAP(4/6) IN ARRAY 37,    '' % BINARY, 0 = NO, 1 = YES
57    A MAO.EQ.ID (3/3) IN ARRAY 37
58
59    EVERY MODEL.AO, AO.RANGE.BAND HAS
60    A MAOR.CIR.ERROR IN ARRAY 38   '' RADIUSIN METERS
```

```
61
62      EVERY MODEL.AR HAS
63      A MAR.NAME IN ARRAY 42,
64      A MAR.VELOCITY(1/2) IN ARRAY 43,
65      A MAR.MIN.PREP(1/3) IN ARRAY 44,
66      A MAR.MAX.PREP(2/3) IN ARRAY 44,
67      A MAR.MAX.ALOFT.TIME(3/3) IN ARRAY 44,
68      A MAR.MAX.SEARCH.RNG(2/2) IN ARRAY 43,
69      A MAR.MIN.SEARCH.RNG(1/2) IN ARRAY 45,
70      A MAR.CEP(2/2) IN ARRAY 45
71
72      EVERY MODEL.CFR HAS
73      A MCFR.MIN.OFF(1/2) IN ARRAY 46, ''IN MINUTES
74      A MCFR.MAX.ON(2/2) IN ARRAY 46, ''IN MINUTES
75      A MCFR.SWEEP.ANGLE(1/2) IN ARRAY 47, ''IN DEGREES
76      A MCFR.SEARCH.WIDTH(2/2) IN ARRAY 47, ''WIDTH OF AREA OF RESPONSIBILITY
77      A MCFR.NAME IN ARRAY 48,
78      A MCFR.EQ.ID (2/2) IN ARRAY 142
79      OWNS
80      A MCFR.RH.LIST
81      HAS
82      A F.MCFR.RH.LIST(1/6) IN ARRAY 142,
83      A L.MCFR.RH.LIST(2/6) IN ARRAY 142,
84      A N.MCFR.RH.LIST(3/6) IN ARRAY 142
85
86      EVERY MODEL.FO HAS
87      A MFO.NAME IN ARRAY 49,
88      A MFO.SEARCH.RATE(1/2) IN ARRAY 53,    ''IN SQ. METERS / MINUTE
89      A MFO.PGM.CAP(2/2) IN ARRAY 53,
90      A MFO.EQ.ID (1/2) IN ARRAY 201
91      OWNS
92      A MFO.RB.SET
93      HAS
94      A F.MFO.RB.SET(4/6) IN ARRAY 201,
95      A L.MFO.RB.SET(5/6) IN ARRAY 201,
96      A N.MFO.RB.SET(6/6) IN ARRAY 201
97
98
99      EVERY MODEL.FO, NITE.OR.DAY, MOVEMENT.STATUS, TYPE.EQUIPMENT, FO.RANGE.BAND HAS
100     A MNMTR.PROB.DETECT IN ARRAY 98
101
102
103     EVERY MODEL.PDB HAS
104     A MPDB.KEY.TIME IN ARRAY 99,
105     A MPDB.NAME IN ARRAY 100,
106     A MPDB.EQ.ID (2/2) IN ARRAY 143
107     OWNS
108     A MPDB.RH.LIST
109     HAS
110     A F.MPDB.RH.LIST(1/6) IN ARRAY 143,
111     A L.MPDB.RH.LIST(2/6) IN ARRAY 143,
112     A N.MPDB.RH.LIST(3/6) IN ARRAY 143
113
114     EVERY MODEL.PIR HAS
115     A MPIR.NAME IN ARRAY 55,
116     A MPIR.VELOCITY(1/2) IN ARRAY 56,
117     A MPIR.MAX.PROCESS(1/3) IN ARRAY 57,
118     A MPIR.MIN.PROCESS(2/3) IN ARRAY 57,
```

```
119     A MPIR.CIR.ERROR(3/3) IN ARRAY 57,
120     A MPIR.HALF.COV.WIDTH(2/2) IN ARRAY 56
121
122     EVERY MODEL.PIR, MOVEMENT.STATUS, ENVIRONMENT, TYPE.EQUIPMENT HAS
123     A MPMET.PROB.DETECT(1/2) IN ARRAY 58,
124     A MPMET.PROB.ACQUIRE(2/2) IN ARRAY 58
125
126     EVERY MODEL.RPV HAS
127     A MRPV.NAME IN ARRAY 61,
128     A MRPV.VELOCITY(1/2) IN ARPAY 62,
129     A MRPV.MAX.PREP(1/3) IN ARRAY 63,
130     A MRPV.MIN.PREP(2/3) IN ARRAY 63,
131     A MRPV.HALF.COV.WIDTH(2/2) IN ARRAY 62,
132     A MRPV.MAX.ALOFT.TIME(1/2) IN ARRAY 65,
133     A MRPV.CIR.ERROR(3/3) IN ARRAY 63,
134     A MRPV.PGM.CAP(2/2) IN ARRAY 65
135
136     EVERY MODEL.RPV, MOVEMENT.STATUS, ENVIRONMENT, TYPE.EQUIPMENT HAS
137     A MRMET.PROB.DETECT(1/2) IN ARRAY 66,
138     A MRMET.PROB.ACQUIRE(2/2) IN ARRAY 66
139
140     EVERY NITE.OR.DAY, PDB.RNG.HACK HAS
141     A MPDB.DET.PROB(1/2) IN ARRAY 71, ''TIMES 1000
142     A MPDB.CIR.ERROR(2/2) IN ARRAY 71 ''IN METERS
143
144     EVERY PDB.RNG.HACK HAS
145     A PDB.RH.RANGE IN ARRAY 72
146     BELONGS TO
147     A MPDB.RH.LIST
148     HAS
149     A P.MPDB.RH.LIST(1/6) IN ARRAY 144,
150     A S.MPDB.RH.LIST(2/6) IN ARRAY 144,
151     A M.MPDB.RH.LIST(3/6) IN ARRAY 144
152
153     EVERY POSTURE HAS
154     A PT.NAME IN ARRAY 73
155
156     EVERY SEARCH.POINT HAS
157     A (SP.X.COORD(1/2),SP.X.GRID(1/3)) IN ARRAY 74,
158     A (SP.Y.COORD(1/2),SP.Y.GRID(1/3)) IN ARRAY 75
159
160     EVERY SECTOR HAS
161     A SE.BNDRY.INT IN ARRAY 76          ''MEASURE OF SECTOR RIGHT BNDRY
162
163     EVERY SENSOR.TYPE HAS
164     A ST.NAME IN ARRAY 77,          ''FO, AO, CB, CM, IR, PH, SD, FL , SO, MT
165     A ST.MIN.XMIT(1/2) IN ARRAY 78,   ''IN MINUTES*10.
166     A ST.MAX.XMIT(2/2) IN ARRAY 78,   ''IN MINUTES*10.
167     A ST.TE.PTR(1/2) IN ARRAY 79,   ''POINTER TO TYPE EQUIP THAT THIS EQUATES TO
168     A ST.MAX.RANGE(2/2) IN ARRAY 79
169
170     EVERY SIDE HAS
171     A COST.CRITERIA(1/3) IN ARRAY 80, ''1 = TRUE(APPLICABLE) FOR ARTY LOGIC, 2 =
172     A FDC.THRESHOLD(2/3) IN ARRAY 80,  ''NOT USED IN LOGIC %26OCT79_%RGR
173     A BREAK.POINT(3/3) IN ARRAY 80, ''PCT ON HAND FOR CBT-INEFF  %RGR
174     A SIDE.TU.TOTAL(1/2) IN ARRAY 111,  ''TOTAL NUM OF TYPE UNITS %RGR
175     A REQ.EFF.MOVING,
176     A REQ.EFF.STA,
```

```
177     A ARTY.DECIMATE(1/2) IN ARRAY 81, ''PCT X 100 ARTY TROOPS FOR CBT-INEFF
178     A ARTY.DEGRADE(2/2) IN ARRAY 81, ''PCT X 100 ARTY TROOPS FOR MAX FIRE RATE
179     A SD.AIRFIELD                    (1/3) IN ARRAY 235,
180     A SD.NR.CAS.MISSIONS             (2/3) IN ARRAY 235,
181     A SD.MAX.SORTIE.TP               (5/6) IN ARRAY 235,
182     A SD.TP.SORTIE                   (1/3) IN ARRAY 236,
183     A SD.ASC.MAX.SORTIE              (6/6) IN ARRAY 235,
184     A SD.ASC.RADIUS                  (2/3) IN ARRAY 236,
185     A SD.SORTIES.THIS.TP             (5/6) IN ARRAY 236,
186     A SD.NO.FLY.VIS                  (1/3) IN ARRAY 237,
187     A SD.POOR.FLY.VIS                (2/3) IN ARRAY 237,
188     A SD.CAS.BRKPT                  (3/3) IN ARRAY 237
189     AND OWNS
190     A SD.ADS.SET,
191     A SD.CMSN.QUEUE,
192     A SD.FPO.LIST,
193     A SD.KAS.SET,
194     A SD.OLD.SORTIE.QUEUE,
195     AN FP.SET,
196     A SIDE.PDB.SET,
197     A SIDE.CFR.SET,
198     AN AVAIL.AO.LIST
199     HAS
200     A F.FP.SET(1/2) IN ARRAY 175,
201     A L.FP.SET(2/2) IN ARRAY 175,
202     A N.FP.SET(2/2) IN ARRAY 111,
203     A F.SIDE.PDB.SET(1/2) IN ARRAY 170,
204     A L.SIDE.PDB.SET(2/2) IN ARRAY 170,
205     A F.SIDE.CFR.SET(1/2) IN ARRAY 146,
206     A L.SIDE.CFR.SET(2/2) IN ARRAY 146,
207     A F.AVAIL.AO.LIST(1/2) IN ARRAY 147,
208     A L.AVAIL.AO.LIST(2/2) IN ARRAY 147,
209     A N.SIDE.PDB.SET(1/3) IN ARRAY 148,
210     A N.SIDE.CFR.SET(2/3) IN ARRAY 148,
211     A N.AVAIL.AO.LIST(3/3) IN ARRAY 148
212
213     EVERY SIDE , GROUPING OWNS
214     A UNIT.SET
215     HAS
216     A F.UNIT.SET(1/3) IN ARRAY 149,
217     A L.UNIT.SET(2/3) IN ARRAY 149,
218     A N.UNIT.SET(3/3) IN ARRAY 149
219
220     EVERY SIDE,KILLER.VICTIM HAS
221     A KV.CEM.WPN.NO,
222     A KV.AMMO.CONSUMED,              ''LBS
223     A KV.INITIAL.DENSITY,
224     A KV.EQ.ID
225
226     EVERY SIDE,KILLER.VICTIM,KILLER.VICTIM  HAS  '' KILLER.SIDE, KILLER.VICTIM %030
227     A KV.SCORE                  ''HITS - INCAPACITATING DAMAGE
228
229     EVERY SIDE, MF.BAND HAS
230     A MFB.UPPER.LIMIT (1/2) IN ARRAY 208,  ''HDM
231     A MFB.DELAY       (2/2) IN ARRAY 208   ''MINUTES
232
233     EVERY SIDE, MISSION HAS
234     A SM.TANK.TE                     (4/6) IN ARRAY 209,
```

```
235   A SM.MIN.TANK.RATIO                    IN ARRAY 238,
236   A SM.MAX.TANK.RATIO                    IN ARRAY 239,
237   A SM.MIN.CEQ                  (3/3) IN ARRAY 209,
238   AN ILLUM.RULE     (1/6) IN ARRAY 209,
239   A  MINE.USE.RULE  (2/6) IN ARRAY 209,
240   A  MINE.WD.RULE   (3/6) IN ARRAY 209
241
242   EVERY SIDE, NITE.OR.DAY, MISSION HAS
243   A SMK.USE.RULE  (1/2) IN ARRAY 210,
244   A SMK.WD.RULE   (2/2) IN ARRAY 210
245
246   EVERY SIDE, SECTOR HAS
247   A SS.REAR IN ARRAY 82
248   OWNS
249   A SS.SET
250   HAS
251   A F.SS.SET(1/3) IN ARRAY 151,
252   A L.SS.SET(2/3) IN ARRAY 151,
253   A N.SS.SET(3/3) IN ARRAY 151
254
255   EVERY SIDE, TYPE.WEAPON HAS
256   A STW.RND.FIRED IN ARRAY 191
257
258   EVERY SMOKE.MUNITION HAS
259   A SMK.ID IN ARRAY 83,
260   A SMK.WIDTH       (1/3) IN ARRAY 84,  ''HDM
261   A SMK.MAX.RANGE   (2/3) IN ARRAY 84,  ''HDM
262   A SMK.BURN.TIME   (3/3) IN ARRAY 84,  ''MINUTES
263   A SMK.RND.WT      (1/3) IN ARRAY 33   ''POUNDS
264
265   EVERY SUBMUNITION HAS
266   A SM.NAME IN ARRAY 87       ''SUBMUNITION NAME
267
268   THE SYSTEM OWNS
269   A COL.SET,
270   A BN.CAN.FM.SET,
271   A BATTLE.SET,
272   AN AO.RB.SET,
273   AN AO.EB.SET,
274   AN IF.RATE.LIST,
275   A DF.RATE.LIST
276
277
278   EVERY TERRAIN.TYPE HAS
279   A TT.LOS.SHAPE,
280   A TT.LOS.SCALE,
281   A TT.NLOS.SHAPE,
282   A TT.NLOS.SCALE,
283   A TT.STATIONARY.LOS.BREAK(1/3) IN ARRAY 145,
284   A TT.MOVING.LOS.BREAK(2/3) IN ARRAY 145,
285   A TT.M.S.LOS.BREAK(3/3) IN ARRAY 145,
286   A MOVE.FIRE.DIST(1/3) IN ARRAY 184,
287   A DEFILADE.DIST(2/3) IN ARRAY 184,
288   A MOV.FAC IN ARRAY 242
289
290   EVERY LOS.BAND HAS
291   A BAND.RANGE(1/2) IN ARRAY 173
292
```

```
293     EVERY TERRAIN.TYPE,LOS.BAND HAS
294     A LOS.PROB(*/6) IN ARRAY 174
295
296     EVERY TYPE.BATTLE.FIELD HAS
297     A TBF.NO.BL.UNITS(1/6) IN ARRAY 88,
298     A TBF.NO.RD.UNITS(2/6) IN ARRAY 88,
299     A TBF.BL.MISSION(3/6) IN ARRAY 88,
300     A TBF.RD.MISSION(4/6) IN ARRAY 88,
301     A TBF.WIDTH(1/2) IN ARRAY 90,
302     A TBF.RD.ARMOR.UNITS(5/6) IN ARRAY 88,
303     A TBF.BL.ARMOR.UNITS(6/6) IN ARRAY 88,
304     A TBF.RD.MECH.UNITS(1/6) IN ARRAY 89,
305     A TBF.BL.MECH.UNITS(2/6) IN ARRAY 89,
306     A TBF.RD.INF.UNITS(3/6) IN ARRAY 89,
307     A TBF.BL.INF.UNITS(4/6) IN ARRAY 89,
308     A TBF.RD.HQ.UNITS(5/6) IN ARRAY 89,
309     A TBF.BL.HQ.UNITS(6/6) IN ARRAY 89
310     AND OWNS
311     A TEAM.TYPES,
312     A TB.SORT.LIST
313     HAS
314     A F.TB.SORT.LIST(1/2) IN ARRAY 180,
315     A L.TB.SORT.LIST(2/2) IN ARRAY 180,
316     A N.TB.SORT.LIST(2/6) IN ARRAY 202,
317     A F.TEAM.TYPES(1/2) IN ARRAY 153,
318     A L.TEAM.TYPES(2/2) IN ARRAY 153,
319     A N.TEAM.TYPES(2/2) IN ARRAY 90
320
321     EVERY TYPE.BTRY HAS
322     A TB.NAME IN ARRAY 91,          ''155, 8IN, ETC
323     A TB.HOW.EQ.ID(1/3) IN ARRAY 92,     ''PTR TO EQUIPMENT
324     A TB.RND.PER.LAUNCH(3/6) IN ARRAY 92,''NUMBER OF TUBES ON LAUNCHER
325     A TB.MIN.HOW(4/6) IN ARRAY 92,       ''MINIMUM HOW TO FIRE MISSION
326     A TB.SUST.FIRE.RATE(1/2) IN ARRAY 93,   ''IN ROUNDS-PER-MINUTE X 100
327     A TB.MAX.RANGE(2/2) IN ARRAY 94,     ''IN DECAMETERS
328     A TB.MAX.RAP.RANGE(1/2) IN ARRAY 94,     ''IN DECAMETERS
329     A TB.SFAIL.MEAN.RNDS(3/3) IN ARRAY 92,  ''IN ROUNDS
330     A TB.LFAIL.MEAN.RNDS(1/3) IN ARRAY 95,  ''IN ROUNDS
331     A TB.SFAIL.REPAIR(2/3) IN ARRAY 95,  ''IN HOURS * 10.
332     A TB.LFAIL.REPAIR(3/3) IN ARRAY 95,  ''IN HOURS * 10.
333     A TB.SUPPRESS.TIME(1/3) IN ARRAY 96,  ''IN MINUTES
334     A TB.MIN.PREP(2/3) IN ARRAY 96,     ''MIN TIME TO PREP A MSN IN MIN X 10
335     A TB.MAX.PREP(3/3) IN ARRAY 96,     ''MAX TIME TO PREP A MSN IN MIN X 10
336     A TB.MIN.FEBA(1/2) IN ARRAY 152,    ''IN HEXADECAMETERS
337     A TB.MAX.FEBA(2/2) IN ARRAY 152,    ''IN HEXADECAMETERS
338     A TB.MARCH.ORDER(1/3) IN ARRAY 171,    ''IN MINUTES
339     A TB.OCCUPY(2/3) IN ARRAY 171,     ''IN MINUTES
340     A TB.SHOOT.SCOOT.IND(4/6) IN ARRAY 93, ''1 = YES, 0 = NO
341     A TB.MN.FASCAM.SUPP (5/6) IN ARRAY 171,  ''MINUTES OF SUPPRESSION
342     A TB.MX.FASCAM.SUPP (6/6) IN ARRAY 171   ''MINUTES
343     OWNS
344     A TB.TM.LIST      ''TYPE MUNITION POINTERS
345     HAS
346     A F.TB.TM.LIST(1/2) IN ARRAY 155,
347     A L.TB.TM.LIST(2/2) IN ARRAY 155,
348     A N.TB.TM.LIST(6/6) IN ARRAY 93
349
350     EVERY TYPE.BTRY, TB.N.FM HAS
```

```
351     A TB.MW.THRESHOLD IN ARRAY 117
352
353     EVERY TYPE.EQUIPMENT HAS
354     A TE.NAME IN ARRAY 118,
355     A TE.PGM.INDIC(1/2) IN ARRAY 119,
356     A TE.PROJECTED.AREA(2/2) IN ARRAY 119, ''IN SQUARE METERS
357     A TE.MIN.MF.LOSS (1/6) IN ARRAY 211,
358     A TE.MAX.MF.LOSS (2/6) IN ARRAY 211,
359     A TE.DELTA.T     (2/3) IN ARRAY 211,  ''DEGREES C * 10
360     A TE.HEIGHT      (3/3) IN ARRAY 211   ''DECIMETERS
361     OWNS A
362     TE.SET
363
364     EVERY TYPE.EQUIPMENT , ENVIRONMENT , SUBMUNITION  HAS
365     A TES.LA.EQUIP IN ARRAY 120    ''LETHAL AREA AGAINST EQUIPMENT - SQ METERS X
366
367     EVERY TYPE.UNIT HAS
368     A TU.ATKING.AC                          (4/6) IN ARRAY 101,
369     A TU.AC.PER.MSN                         (4/6) IN ARRAY 159,
370     A TU.LEVEL IN ARRAY 121,             ''CO, BTRY, BN, ETC
371     A TU.CAT(1/6) IN ARRAY 122,             ''PTR TO CATEGORY
372     A TU.MIL.WORTH(2/3) IN ARRAY 122,
373     A TU.FREQ(3/3) IN ARRAY 122,         ''PERCENT OF UNITS IN FORCE THAT ARE OF
374     A TU.RADIUS(1/3) IN ARRAY 101,  ''IN METERS
375     A TU.PRIN.TE(3/6) IN ARRAY 101,
376     A TU.CRIT.NO(3/3) IN ARRAY 101,          ''NUMBER OF CRITICAL PIECES OF EQUI
377     A TU.MOV.RATE IN ARRAY 9,   ''BASIC MOVEMENT RATE FOR UNIT IN HDM PER HOUR
378     A TU.SIDE(2/6) IN ARRAY 122,
379     A TU.SUP.PRIORITY(1/3) IN ARRAY 172,
380     A TU.OPP.PRIORITY(2/3) IN ARRAY 172,
381     A TU.MF.FACTOR (3/3) IN ARRAY 172  ''MODIFIES MINE DELAY
382     OWNS
383     A TU.TE.LIST,
384     A TU.NTE.SET
385     BELONGS TO
386     A CT.TU.SET
387     HAS
388     A F.TU.TE.LIST(1/2) IN ARRAY 157,
389     A L.TU.TE.LIST(2/2) IN ARRAY 157,
390     A F.TU.NTE.SET(1/2) IN ARRAY 150,
391     A L.TU.NTE.SET(2/2) IN ARRAY 150,
392     A P.CT.TU.SET(1/3) IN ARRAY 158,
393     A S.CT.TU.SET(2/3) IN ARRAY 158,
394     A N.TU.TE.LIST(3/3) IN ARRAY 158,
395     A N.TU.NTE.SET(1/3) IN ARRAY 159,
396     A M.CT.TU.SET(3/6) IN ARRAY 159
397
398     EVERY TYPE.WEAPON HAS
399     A TW.ROF.AIR                        (2/2) IN ARRAY 190,
400     A TW.AC.DET.TIME                    (2/2) IN ARRAY 179,   ''SECONDS
401     A TW.NAME IN ARRAY 102,
402     A TW.RATE.OF.FIRE(1/3) IN ARRAY 103,  ''IN TENTHS OF ROUNDS PER MINUTE
403     A TW.NO.SENSORS(3/6) IN ARRAY 103,
404     A TW.ROUND.VELOCITY(3/3) IN ARRAY 103,  ''IN HDM PER SECOND
405     A TW.RND.WEIGHT(1/2) IN ARRAY 190, ''LBS
406     A TW.MAX.RANGE(2/2) IN ARRAY 104,  ''IN METERS
407     A TW.MIN.RANGE(1/2) IN ARRAY 104,  ''IN METERS
408     A TW.BASIC.LOAD(1/2) IN ARRAY 179,
```

```
409     A TW.PK.PTR(1/3) IN ARRAY 185,
410     A TW.NITE.FAC IN ARRAY 181,
411     A TW.FIRE.OTM.PTR(2/3) IN ARRAY 185,
412     A TW.TYPE.OF.SENSOR  (1/3) IN ARRAY 212,
413     A TW.SPECTRUM        (2/3) IN ARRAY 212,
414     A TW.HFOV            (3/3) IN ARRAY 212,  ''DEGREES * 10
415     A TW.VFOV            (1/3) IN ARRAY 213,  ''DEGREES * 10
416     A TW.HFOS            (2/3) IN ARRAY 213,  ''DEGREES
417     A TW.VFOS            (3/3) IN ARRAY 213   ''DEGREES
418
419     EVERY UNIT HAS
420     A UN.AD.AVAIL                        (6/6) IN ARRAY 214,
421     A UNIT.NOS(2/2) IN ARRAY 109,
422     A UN.ENGAGED.INDEX(5/6) IN ARRAY 105,
423     A UN.TYPE.UNIT(1/3) IN ARRAY 105,      ''TYPE UNIT
424     A (UN.X.COORD(1/2),UN.X.GRID(1/3)) IN ARRAY 106, ''IN HEXADECAMETERS
425     A (UN.Y.COORD(1/2),UN.Y.GRID(1/3)) IN ARRAY 107, ''IN HEXADECAMETERS
426     A UN.RADIUS(2/3) IN ARRAY 105, ''RADIUS IN METERS - DO NOT USE "333" %RGR
427     A UN.LAST.ARTY.ENG(2/2) IN ARRAY 106, ''TIME LAST BTRY VOL(MIN X 100)
428     A UN.TIME.LAST.MOVE IN ARRAY 108,  ''TIME SINCE THE UNIT WAS LAST MOVED
429     A UN.PARENT(2/2) IN ARRAY 107,         ''PARENT UNIT INDEX
430     A UN.MISSION(2/6) IN ARRAY 108,        ''ATTACK, DEFEND, ETC
431     A UN.STATUS(2/6) IN ARRAY 109,         ''ADVANCING, WITHDRAWING, OVERWATCHI
432     A UN.COLOR(3/6) IN ARRAY 109,          ''BLUE OR RED
433     A UN.POSITION.INDEX(1/6) IN ARRAY 109,'' LINK OF PATH IN CLOSE CMBT
434     A UN.BATTLE.INDEX(1/2)  IN ARRAY 110,  '' %5DEC79_%RGR
435     A UN.BTRY.INDEX(2/2) IN ARRAY 110,  '' %5DEC79_%RGR
436     A UN.PTR      IN ARRAY 203,
437     A UN.FASCAM.RECVD  (1/3) IN ARRAY 214,
438     A UN.DELAY         (2/3) IN ARRAY 214
439     AND OWNS
440     A UN.EQUIP.LIST,
441     A UN.SUB.LIST,
442     A UN.PATH,          ''PATH FOR UNIT IN CLOSE COMBAT
443     A UN.SEGMENT.LIST,
444     A UN.LOS.LIST,    ''VISUAL UNITS IN LOS
445     A UN.HC.LOS.LIST,   ''HELICOPTERS IN LOS
446     A UN.SENSOR.LIST,
447     A MO.LIST
448     MAY BELONG TO
449     A SS.SET,
450     A UN.SUB.LIST,
451     A FR.UNIT.SET,
452     AN AO.DET.TGT.LIST,
453     AN AR.DET.TGT.LIST,
454     AN HT.TARGET.LIST,
455     A TB.SORT.LIST
456     BELONGS TO
457     A UNIT.SET
458     HAS
459     A P.TB.SORT.LIST(1/2) IN ARRAY 196,
460     A S.TB.SORT.LIST(2/2) IN ARRAY 196,
461     A M.TB.SORT.LIST(3/6) IN ARRAY 188,
462     A F.UN.EQUIP.LIST(1/2) IN ARRAY 160,
463     A L.UN.EQUIP.LIST(2/2) IN ARRAY 160,
464     A F.UN.SUB.LIST(1/2) IN ARRAY 161,
465     A L.UN.SUB.LIST(2/2) IN ARRAY 161,
466     A F.UN.PATH(1/2) IN ARRAY 162,
```

```
467     A L.UN.PATH(2/2) IN ARRAY 162,
468     A F.UN.SEGMENT.LIST IN ARRAY 163,
469     A L.UN.SEGMENT.LIST IN ARRAY 197,
470     A F.UN.LOS.LIST IN ARRAY 164,
471     A L.UN.LOS.LIST IN ARRAY 198,
472     A F.UN.SENSOR.LIST(1/2) IN ARRAY 165,
473     A L.UN.SENSOR.LIST(2/2) IN ARRAY 165,
474     A P.SS.SET(1/3) IN ARRAY 166,
475     A S.SS.SET(2/3) IN ARRAY 166,
476     A P.UN.SUB.LIST(3/3) IN ARRAY 166,
477     A S.UN.SUB.LIST(1/3) IN ARRAY 167,
478     A P.FR.UNIT.SET(2/3) IN ARRAY 167,
479     A S.FR.UNIT.SET(3/3) IN ARRAY 167,
480     A P.AO.DET.TGT.LIST(1/3) IN ARRAY 168,
481     A S.AO.DET.TGT.LIST(2/3) IN ARRAY 168,
482     A P.AR.DET.TGT.LIST(3/3) IN ARRAY 168,
483     A S.AR.DET.TGT.LIST(1/3) IN ARRAY 169,
484     A P.UNIT.SET(2/3) IN ARRAY 169,
485     A S.UNIT.SET(3/3) IN ARRAY 169,
486     A N.UN.EQUIP.LIST(1/6) IN ARRAY 154,
487     A N.UN.SUB.LIST(2/6) IN ARRAY 154,
488     A N.UN.PATH(3/6) IN ARRAY 154,
489     A N.UN.SEGMENT.LIST(4/6) IN ARRAY 154,
490     A N.UN.LOS.LIST(5/6) IN ARRAY 154,
491     A N.UN.SENSOR.LIST(6/6) IN ARRAY 154,
492     A M.SS.SET(1/6) IN ARRAY 156,
493     A M.UN.SUB.LIST(2/6) IN ARRAY 156,
494     A M.FR.UNIT.SET(3/6) IN ARRAY 156,
495     A M.AO.DET.TGT.LIST(4/6) IN ARRAY 156,
496     A M.AR.DET.TGT.LIST(5/6) IN ARRAY 156,
497     A M.UNIT.SET(6/6) IN ARRAY 156,
498     A P.HT.TARGET.LIST(1/2) IN ARRAY 176,
499     A S.HT.TARGET.LIST(2/2) IN ARRAY 176,
500     A M.HT.TARGET.LIST(1/6) IN ARRAY 188,
501     A F.UN.HC.LOS.LIST IN ARRAY 67,
502     A L.UN.HC.LOS.LIST IN ARRAY 199,
503     A N.UN.HC.LOS.LIST(3/3) IN ARRAY 188,
504     A N.MO.LIST  (5/6) IN ARRAY 214,
505     A F.MO.LIST        IN ARRAY 215,
506     A L.MO.LIST        IN ARRAY 216
507
508     EVERY UNIT, UNIT HAS ''                                          \NEEDED?
509     A ACT.RANGE FUNCTION
510
511     EVERY PK.BAND HAS          ''PK.VECTOR.NO,PK.MOVE.FACTOR.NO REMOVED
512     A PK.BAND.RNG IN ARRAY 182
513
514     EVERY PK.VECTOR,PK.BAND HAS
515     A PK.PROB (*/6) IN ARRAY 183
516
517     EVERY PK.MOVE.BAND HAS
518     A PK.MOV.RNG IN ARRAY 187
519
520     EVERY PK.MOVE.FACTOR,PK.MOVE.BAND HAS
521     A PK.MOV.FAC IN ARRAY 178
522
523     EVERY PK.F.MOVE.FACTOR, PK.MOVE.BAND HAS
524     A PK.F.MOV.FAC IN ARRAY 177
```

```
525
526     ''**************************TEMPORARY  ENTITIES* SECTION TWO ***************
527
528     TEMPORARY ENTITIES
529
530     EVERY AAT.TGT HAS
531     AN AATT.UE.LINK                    (1/2) IN WORD 1,
532     AN AATT.CUM.AREA                   (2/2) IN WORD 1
533     AND BELONGS TO
534     AN AATT.LIST
535     AND HAS
536     A P.AATT.LIST                 (1/2) IN WORD 2,
537     A S.AATT.LIST                 (2/2) IN WORD 2,
538     A M.AATT.LIST                 (3/6) IN WORD 3
539
540     EVERY AD.SENSOR HAS
541     AN ADS.UNIT.PTR                    (1/3) IN WORD 1,
542     AN ADS.MADS.STATUS                 (6/6) IN WORD 1,
543     AN ADS.MADS.PTR                    (3/6) IN WORD 1,
544     AN ADS.NR.SENSORS                  (4/6) IN WORD 1
545     AND BELONGS TO
546     AN SD.ADS.SET
547     AND HAS
548     A P.SD.ADS.SET                (1/2) IN WORD 2,
549     A S.SD.ADS.SET                (2/2) IN WORD 2,
550     A M.SD.ADS.SET                (5/6) IN WORD 1
551
552     EVERY AA.LINK HAS
553     A AA.UE.LINK(1/2) IN WORD 1,
554     A AA.FRACTION IN WORD 2,
555     AND BELONGS TO
556     A AA.SET
557     HAS
558     A P.AA.SET IN WORD 3,
559     A S.AA.SET IN WORD 4,
560     A M.AA.SET(4/6) IN WORD 1
561
562     EVERY AO.DET.CANDIDATE HAS
563     A AO.DC.UNIT(1/3) IN WORD 1,
564     A AO.DC.LEG.DIST(2/3) IN WORD 1,
565     A AO.DC.DIST(3/3) IN WORD 1
566     BELONGS TO
567     AN AO.CAND.DET.LIST
568     HAS
569     A P.AO.CAND.DET.LIST(1/2) IN WORD 2,
570     A S.AO.CAND.DET.LIST(2/2) IN WORD 2,
571     A M.AO.CAND.DET.LIST(1/6) IN WORD 3
572
573     EVERY AR.DET.CANDIDATE HAS
574     A AR.DC.UNIT(1/2) IN WORD 1,
575     A AR.DC.Y.COORD(2/2) IN WORD 1
576     BELONGS TO
577     AN AR.CAND.DET.LIST
578     HAS
579     A P.AR.CAND.DET.LIST(1/2) IN WORD 2,
580     A S.AR.CAND.DET.LIST(2/2) IN WORD 2,
581     A M.AR.CAND.DET.LIST(1/6) IN WORD 3
582
```

```
583   EVERY ATK.ORDER HAS
584   AN ENEMY.DO(1/2) IN WORD 1,
585   AN OWN.DO(2/2) IN WORD 1
586
587   EVERY AWARE.UNIT HAS
588   AN AU.UNIT.ID
589   AND BELONGS TO
590   AN AU.LIST
591
592   EVERY BATTLE HAS
593   A BTL.TIME.OF.DAY(1/2) IN WORD 1,
594   A BTL.SEQ.NO(2/2) IN WORD 1,
595   A BTL.TERRAIN.TYPE(1/6) IN WORD 2,     ''TERRAIN.TYPE IS A POINTER
596   A BTL.FIELD(1/2) IN WORD 3,            ''BTL.FIELD IS THE PRESTORED
597   ''BATTLEFIELD TO BE SELECTED
598   A BTL.BL.UNITS(1/2) IN WORD 4,
599   A BTL.RD.UNITS(2/2) IN WORD 4,
600   A BTL.WIDTH(2/2) IN WORD 3,            ''IN HDM
601   A RED.HB.PRIORITY IN WORD 9,           ''PRIORITY FOR HC SUPPORT
602   A BLUE.HB.PRIORITY IN WORD 10,         ''PRIORITY FOR HC SUPPORT
603   A BTL.BL.HC.TEAM(1/2) IN WORD 7,       ''CURRENT BLUE SUPPORT TEAM
604   A BTL.RD.HC.TEAM(2/2) IN WORD 7,       ''CURRENT RED SUPPORT TEAM
605   A BTL.BL.FARRP(1/2) IN WORD 8,         ''BLUE FARRP
606   A BTL.RD.FARRP(2/2) IN WORD 8,         ''RED FARRP
607   AND BELONGS TO
608   A BATTLE.SET
609   MAY OWN
610   A BTL.FORCE.SET
611   HAS
612   A P.BATTLE.SET(1/2) IN WORD 5,
613   A S.BATTLE.SET(2/2) IN WORD 5,
614   A M.BATTLE.SET(2/6) IN WORD 2,
615   A F.BTL.FORCE.SET(1/2) IN WORD 6,
616   A L.BTL.FORCE.SET(2/2) IN WORD 6,
617   A N.BTL.FORCE.SET(2/2) IN WORD 2
618
619   EVERY CFP.OBS.LINK HAS
620   A COL.OBSTACLE.PTR                (1/2) IN WORD 1,
621   A COL.XENTRY                      (2/2) IN WORD 1,
622   A COL.XEXIT                       (1/2) IN WORD 2,
623   A COL.YENTRY                      (1/2) IN WORD 2,
624   A COL.YEXIT                      (1/2) IN WORD 3
625   AND BELONGS TO
626   A COL.SET
627   AND HAS
628   A P.COL.SET                  (2/2) IN WORD 3,
629   A S.COL.SET                  (1/2) IN WORD 4,
630   A M.COL.SET                  (4/6) IN WORD 4
631
632   EVERY CFP.SEGMENT HAS
633   A CFPS.XSTART                     (1/2) IN WORD 1,
634   A CFPS.YSTART                     (2/2) IN WORD 1,
635   A CFPS.XEND                       (1/2) IN WORD 2,
636   A CFPS.YEND                       (2/2) IN WORD 2,
637   A CFPS.TIME.LENGTH                      IN WORD 3
638   AND OWNS
639   A SI.LIST
640   AND BELONGS TO
```

```
641     A CFPS.LIST
642     AND HAS
643     A F.SI.LIST                     (1/2) IN WORD 4,
644     A L.SI.LIST                     (2/2) IN WORD 4,
645     A N.SI.LIST                     (1/2) IN WORD 5,
646     A P.CFPS.LIST                   (2/2) IN WORD 5,
647     A S.CFPS.LIST                   (1/2) IN WORD 6,
648     A M.CFPS.LIST                   (4/6) IN WORD 6
649
650     EVERY CF.DET.UNIT HAS
651     A CF.D.BTRY(1/3)IN WORD 1,
652     A CF.D.CPE(2/3)IN WORD 1,
653     A CF.D.PD(3/3)IN WORD 1,
654     A CF.D.PRIORITY(3/3)IN WORD 2,
655     BELONGS TO
656     A CF.OP.Q
657     HAS
658     A P.CF.OP.Q(1/2) IN WORD 3,
659     A S.CF.OP.Q(2/2) IN WORD 3,
660     A M.CF.OP.Q(3/6) IN WORD 2
661
662     EVERY CF.RADAR HAS
663     A CFR.LAST.ON.OR.OFF  IN WORD 1,
664     A CFR.ORIENTATION(1/2) IN WORD 2,  ''RADIANS TIME 1000
665     A CFR.US.LINK(2/2) IN WORD 2,
666     A CF.OPERATOR(2/6) IN WORD 4
667     BELONGS TO
668     A SIDE.CFR.SET
669     OWNS
670     A CF.OP.Q
671     HAS
672     A P.SIDE.CFR.SET(1/2) IN WORD 3,
673     A S.SIDE.CFR.SET(2/2) IN WORD 3,
674     A M.SIDE.CFR.SET(1/6) IN WORD 4,
675     A F.CF.OP.Q(1/2) IN WORD 5,
676     A L.CF.OP.Q(2/2) IN WORD 5,
677     A N.CF.OP.Q(2/3) IN WORD 4
678
679
680     EVERY DEF.ORDER HAS
681     A REINF.THRESH(1/2) IN WORD 1,
682     AN EN.DIS.OP(4/6) IN WORD 1,
683     AN OWN.DIS.OP(5/6) IN WORD 1,
684     AN ORD.MISSION(6/6) IN WORD 1
685
686     EVERY DF.NOISE HAS
687     A DF.TIME IN WORD 1,
688     A DF.UNIT(1/2) IN WORD 2
689     BELONGS TO
690     THE DF.RATE.LIST
691     HAS
692     A P.DF.RATE.LIST(2/2) IN WORD 2,
693     A S.DF.RATE.LIST(1/2) IN WORD 3,
694     A M.DF.RATE.LIST(4/6) IN WORD 3
695
696     EVERY EX.AC.ATK.TGT HAS
697     AN EAAT.AC.DET.TIME     (1/2)    IN WORD 1,
698     AN EAAT.BLIND           (2/2)    IN WORD 1,
```

```
699     AN EAAT.DETECT.TIME              IN WORD 2,
700     AN EAAT.DIST.TO.FP       (1/2)   IN WORD 3,
701     AN EAAT.DIST.TO.P3       (2/2)   IN WORD 3,
702     AN EAAT.FIRING.RANGE     (1/2)   IN WORD 4,
703     AN EAAT.FIRING.TIME              IN WORD 5,
704     AN EAAT.MODEL.ADS        (2/2)   IN WORD 4,
705     AN EAAT.NUM.FIRES.P      (1/2)   IN WORD 6,
706     AN EAAT.NUM.PASSES       (2/2)   IN WORD 6,
707     AN EAAT.P1.TO.P2         (1/2)   IN WORD 7,
708     AN EAAT.P2.TO.P3         (2/2)   IN WORD 7,
709     AN EAAT.P3.TO.P1         (1/2)   IN WORD 8,
710     AN EAAT.RANGE            (2/2)   IN WORD 8,
711     AN EAAT.RANGE.AT.FP      (1/2)   IN WORD 9,
712     AN EAAT.RESULT           (2/2)   IN WORD 9,
713     AN EAAT.ROW              (1/2)   IN WORD 10,
714     AN EAAT.SUM.CRIT.EQ      (2/2)   IN WORD 10,
715     AN EAAT.TIME.TO.LEA              IN WORD 11,
716     AN EAAT.WAIT.TIME                IN WORD 12
717
718     EVERY EX.FIRE.MISSION HAS
719     AN EFM.PREP.TIME IN WORD 1,
720     AN EFM.FIRE.RATE IN WORD 2,
721     AN EFM.SUPPRESS IN WORD 3,
722     AN EFM.LAST.ARTY.ENGAGE IN WORD 4,
723     AN EFM.OLD.CEP IN WORD 5,
724     AN EFM.SUPP.TIME IN WORD 6
725
726
727     EVERY EX.FWD.OBSERVER HAS
728     AN EFO.CANDIDATE         (1/2) IN WORD 1,
729     AN EFO.ENEMY             (4/6) IN WORD 1,
730     AN EFO.SEARCH.TIME             IN WORD 2,
731     AN EFO.FO.UNIT           (1/3) IN WORD 3,
732     AN EFO.X.CORRECT         (1/2) IN WORD 4,
733     AN EFO.Y.CORRECT         (2/2) IN WORD 4,
734     AN EFO.LINK              (1/2) IN WORD 5,
735     AN EFO.MIN.XMIT          (1/2) IN WORD 6,
736     AN EFO.MAX.XMIT          (2/2) IN WORD 6,
737     AN EFO.PERIOD.OF.SEARCH        IN WORD 7,
738     AN EFO.X.SEARCH.GRID     (2/3) IN WORD 3,
739     AN EFO.Y.SEARCH.GRID     (3/3) IN WORD 3,
740     AN EFO.START.TIME              IN WORD 8,
741     AN EFO.TARGET            (2/2) IN WORD 5
742
743     EVERY EX.SHOOT.OUT HAS
744     A ESO.CUM1(1/2) IN WORD 1,
745     A ESO.CUM2(2/2) IN WORD 1,
746     A ESO.DUMMY(1/2) IN WORD 2,
747     A ESO.RANGE(2/2) IN WORD 2,
748     A ESO.SCR1(1/2) IN WORD 5,
749     A ESO.SCR2(2/2) IN WORD 5,
750     A ESO.TGT(2/2) IN WORD 4,
751     A ESO.TGT.EQUIP(1/2) IN WORD 8,
752     A ESO.TGT.UNIT(2/2) IN WORD 8,
753     A ESO.WEAPON       IN WORD 7,
754     A ESO.WPN          IN WORD 9,
755     A ESO.ACQ.TIME IN WORD 6,
756     A ESO.QUANT(2/2) IN WORD 3,
```

```
757     A ESO..TGT(1/2) IN WORD 3
758
759     EVERY EX.TGT.REPORT HAS
760     AN ETR.DUPLICATE  (1/2) IN WORD 1,
761     AN ETR.DUR              IN WORD 2,
762     AN ETR.MAX.PREP   (2/2) IN WORD 1,
763     AN ETR.NEW.STOP   (1/2) IN WORD 3,
764     AN ETR.RFAF       (2/2) IN WORD 3,
765     AN ETR.START.TIME (1/2) IN WORD 4,
766     AN ETR.TOT.FOLLOW (2/2) IN WORD 4
767
768     EVERY FARRP HAS
769     AN FP.UNIT(1/2) IN WORD 1,
770     AN FP.BATTLE IN WORD 6,
771     AN FP.NO.TEAMS(1/6) IN WORD 3,
772     AN REFUEL.TIME(1/2) IN WORD 2,
773     A REARM.TIME(2/2) IN WORD 2,
774     A REFUEL.CAP(2/3) IN WORD 3,
775     A REARM.CAP(3/3) IN WORD 3,
776     A P.FP.SET(1/2) IN WORD 4,
777     AN S.FP.SET(2/2) IN WORD 4,
778     AN M.FP.SET(2/6) IN WORD 3,
779     AN F.HT.LIST(1/2) IN WORD 5,
780     AN L.HT.LIST(2/2) IN WORD 5,
781     AN N.HT.LIST(3/4) IN WORD 1
782     AND BELONGS TO
783     AN FP.SET
784     AND OWNS
785     AN HT.LIST
786
787     EVERY FD.BN.LINK HAS
788     A FB.BN(1/2)    IN WORD 1                ''FA.BN INDEX
789     BELONGS TO
790     A FD.BN.LIST
791     HAS
792     A P.FD.BN.LIST(1/2) IN WORD 2,
793     A S.FD.BN.LIST(2/2) IN WORD 2,
794     A M.FD.BN.LIST(4/6) IN WORD 1
795
796     EVERY FD.SCHD.MSN HAS
797     A FS.START IN WORD 1,       ''START TIME OF SCHEDULED BTRY USAGE * 100
798     A FS.STOP IN WORD 2,        ''STOP TIME OF SCHEDULED BTRY USAGE * 100
799     A FS.BATS(1/2) IN WORD 3        ''NUMBER OF BTRYS SCHEDULED
800     MAY BELONG TO
801     A FD.SCHD.LIST
802     HAS
803     A  P.FD.SCHD.LIST(2/2) IN WORD 3,
804     A  S.FD.SCHD.LIST(1/2) IN WORD 4,
805     A  M.FD.SCHD.LIST(4/6) IN WORD 4
806
807     EVERY FIRING.TABLE HAS
808     A FT.AC.ATK.TGT                  (1/2) IN WORD 8,
809     A FT.CAS.MISSION                 (2/2) IN WORD 8,
810     A FT.TGT.UNIT(1/3) IN WORD 1,
811     A FT.TARGET.EQUIP(2/2) IN WORD 1,
812     A FT.FIRING.WPN      IN WORD 2,
813     A FT.PK IN WORD 3,    ''IN FRACTION
814     A FT.PK.BAR IN WORD 4,    ''IN FRACTION
```

```
815     A FT.SCORE1(1/2) IN WORD 5,
816     A FT.SCORE2(2/2) IN WORD 5
817     AND BELONGS TO
818     A UE.TARGET.LIST
819     HAS
820     A P.UE.TARGET.LIST IN WORD 6,
821     A S.UE.TARGET.LIST IN WORD 7,
822     A M.UE.TARGET.LIST(3/6) IN WORD 1
823
824     EVERY FLIGHT.LEG HAS
825     A FL.X.START(1/2) IN WORD 1,
826     A FL.Y.START(2/2) IN WORD 1,
827     A FL.X.END(1/2) IN WORD 2,
828     A FL.Y.END(2/2) IN WORD 2
829     BELONGS TO
830     AN RPV.FLIGHT.LEG.LIST,
831     AN AO.FLIGHT.LEG.LIST,
832     A PIR.FLIGHT.LEG.LIST
833     HAS
834     A P.PIR.FLIGHT.LEG.LIST(1/2) IN WORD 3,
835     A S.PIR.FLIGHT.LEG.LIST(2/2) IN WORD 3,
836     A P.AO.FLIGHT.LEG.LIST(1/2) IN WORD 4,
837     A S.AO.FLIGHT.LEG.LIST(2/2) IN WORD 4,
838     A P.RPV.FLIGHT.LEG.LIST(1/2) IN WORD 5,
839     A S.RPV.FLIGHT.LEG.LIST(2/2) IN WORD 5,
840     A M.RPV.FLIGHT.LEG.LIST(1/6) IN WORD 6,
841     A M.AO.FLIGHT.LEG.LIST(2/6) IN WORD 6,
842     A M.PIR.FLIGHT.LEG.LIST(3/6) IN WORD 6
843
844     EVERY FORCE HAS
845     A DECISION.POINT(1/3) IN WORD 1,
846     A FR.CRIT.NO(2/3) IN WORD 1,
847     A FR.MISSION(3/3) IN WORD 1,
848     A FR.SIDE(1/6) IN WORD 2,
849     A FR.CAS.INDIC                  (6/6) IN WORD 2
850     MAY OWN
851     A FR.UNIT.SET
852     MAY BELONG TO
853     A BTL.FORCE.SET
854     HAS
855     A F.FR.UNIT.SET(1/2) IN WORD 3,
856     A L.FR.UNIT.SET(2/2) IN WORD 3,
857     A N.FR.UNIT.SET(2/3) IN WORD 2,
858     A P.BTL.FORCE.SET(1/2) IN WORD 4,
859     A S.BTL.FORCE.SET(2/2) IN WORD 4,
860     A M.BTL.FORCE.SET(5/6) IN WORD 2
861
862     EVERY FO.DET.CANDIDATE HAS
863     A FO.DC.UNIT(1/2) IN WORD 1,
864     A FO.DC.RANGE(2/2) IN WORD 1
865     MAY BELONG TO
866     AN FO.CAND.DET.LIST
867     HAS
868     A P.FO.CAND.DET.LIST(1/2) IN WORD 2,
869     A S.FO.CAND.DET.LIST(2/2) IN WORD 2,
870     A M.FO.CAND.DET.LIST(1/6) IN WORD 3
871
872     EVERY FP.OBSTACLE HAS
```

```
873   AN FPO.XMIN                          (1/2) IN WORD 1,
874   AN FPO.XMAX                          (2/2) IN WORD 1,
875   AN FPO.YMIN                          (1/2) IN WORD 2,
876   AN FPO.YMAX                          (2/2) IN WORD 2
877   AND BELONGS TO
878   AN SD.FPO.LIST
879   AND HAS
880   A P.SD.FPO.LIST                (1/2) IN WORD 3,
881   A S.SD.FPO.LIST                (2/2) IN WORD 3,
882   A M.SD.FPO.LIST                (1/6) IN WORD 4
883
884   EVERY HC.TEAM HAS
885   A HT.FARRP(1/2) IN WORD 1,
886   A HT.STATUS(1/6) IN WORD 3,
887   A HT.MOVE.TIME IN WORD 6,
888   A HT.LOITER.TIME IN WORD 7,
889   A HT.ARR.BTL.TIME IN WORD 8,
890   A HT.TERMINATOR(4/6) IN WORD 3,
891   AN F.HT.MEMBER.LIST(1/2) IN WORD 2,
892   AN L.HT.MEMBER.LIST(2/2) IN WORD 2,
893   AN N.HT.MEMBER.LIST(3/4) IN WORD 1,
894   AN F.HT.TARGET.LIST(1/2) IN WORD 4,
895   AN L.HT.TARGET.LIST(2/2) IN WORD 4,
896   AN N.HT.TARGET.LIST(3/3) IN WORD 3,
897   A P.HT.LIST(1/2) IN WORD 5,
898   A S.HT.LIST(2/2) IN WORD 5,
899   AN M.HT.LIST(3/6) IN WORD 3
900   AND OWNS
901   AN HT.MEMBER.LIST,  ''MEMBERS OF THE TEAM ''
902   AN HT.TARGET.LIST   ''TEAM TARGET LIST''
903   AND BELONGS TO
904   AN HT.LIST
905
906   EVERY HELICOPTER HAS
907   A HC.TYPE (1/6) IN WORD 1,     ''SCOUT OR ATTACK''
908   A HC.UE.ID(1/2) IN WORD 4,  ''SAME AS UE.ID''
909   A HC.BTL.TEAM(2/2) IN WORD 4,    ''BATTLE TEAM
910   A HC.PAIRED (2/6) IN WORD 1,  ''YES OR NO''
911   A HC.X (1/2) IN WORD 2,        ''X-COORDINATE''
912   A HC.Y (2/2) IN WORD 2,        ''Y-COORDINATE''
913   A HC.ALTITUDE (3/6) IN WORD 1,''MASKED, UNMASKED, ON GROUND''
914   A HC.STATUS (4/6) IN WORD 1,  ''DETECTING, ENGAGING, ETC.''
915   A HC.TIME.ALOFT IN WORD 5,       ''MINUTES''
916   A HC.FIRE.MASK(6/6) IN WORD 1,       ''MINUTES''
917   A P.HT.MEMBER.LIST (1/2) IN WORD 3,
918   A S.HT.MEMBER.LIST (2/2) IN WORD 3,
919   A M.HT.MEMBER.LIST (5/6) IN WORD 1
920   AND BELONGS TO
921   A HT.MEMBER.LIST
922   AND OWNS
923   AN HC.UN.LOS.LIST
924
925   EVERY HOW HAS
926   A HW.BTRY(1/3) IN WORD 1,    ''OWNING BTRY
927   A HW.SFAIL.RNDS(2/3) IN WORD 1, ''ROUNDS TILL SHORT TERM FAILURE
928   A HW.LFAIL.RNDS(3/3) IN WORD 1, ''ROUNDS TILL LONG TERM FAILURE
929   BELONGS TO
930   A BY.HOW.SET ''WHEN NOT FAILED
```

```
931     HAS
932     A P.BY.HOW.SET(1/2) IN WORD 2,
933     A S.BY.HOW.SET(2/2) IN WORD 2,
934     A M.BY.HOW.SET(1/6) IN WORD 3
935
936     EVERY IF.VOLLEY HAS
937     A IF.V.TIME IN WORD 1,
938     A IF.V.BTRY(1/2) IN WORD 2
939     BELONGS TO
940     A IF.RATE.LIST
941     HAS
942     A P.IF.RATE.LIST(1/2) IN WORD 3,
943     A S.IF.RATE.LIST(2/2) IN WORD 3,
944     A M.IF.RATE.LIST(4/6) IN WORD 2
945
946     EVERY KEYED.SENSOR HAS
947     A KS.TYPE.SENSOR(1/6) IN WORD 1,
948     A KS.SENSOR.ID(2/2) IN WORD 1
949     MAY BELONG TO
950     A PDB.KEYED.LIST
951     HAS
952     A P.PDB.KEYED.LIST(1/2) IN WORD 2,
953     A S.PDB.KEYED.LIST(2/2) IN WORD 2,
954     A M.PDB.KEYED.LIST(2/6) IN WORD 1
955
956     EVERY KNOWN.AD.SENSOR HAS
957     A KAS.AD.UNIT                       (1/2) IN WORD 1
958     AND BELONGS TO
959     A SD.KAS.SET
960     AND HAS
961     A P.SD.KAS.SET                (2/2) IN WORD 1,
962     A S.SD.KAS.SET                (1/2) IN WORD 2,
963     A M.SD.KAS.SET                (4/6) IN WORD 2
964
965     EVERY MADS.RH HAS
966     AN MRH.RANGE                        (1/2) IN WORD 1,
967     AN MRH.MIN.ALT                      (2/2) IN WORD 1,
968     AN MRH.PD                           (1/3) IN WORD 2
969     AND BELONGS TO
970     AN MADS.RH.SET
971     AND HAS
972     A P.MADS.RH.SET               (1/2) IN WORD 3,
973     A S.MADS.RH.SET               (2/2) IN WORD 3,
974     A M.MADS.RH.SET               (3/6) IN WORD 2
975
976     EVERY MAN.UNIT HAS
977     A MU.CRIT.NO(1/2) IN WORD 1,
978     A MU.REINF.IND(4/6) IN WORD 1, ''YES OR NO INDICATING AVAILABILITY FOR
979     A MU.CUR.ORDER(5/6) IN WORD 1,
980     A MU.TF.MEM(1/2) IN WORD 3,
981     A MU.OFFSET.X(1/2) IN WORD 2,
982     A MU.OFFSET.Y(2/2) IN WORD 2,
983     A MU.UNIT.ID(2/2) IN WORD 3
984     AND OWNS
985     A MU.TF.LIST,
986     A MU.ORDER.SET
987     AND BELONGS TO
988     A MU.TF.LIST
```

```
    HAS
    A F.MU.TF.LIST       IN WORD 4,
    A L.MU.TF.LIST       IN WORD 9,
    A F.MU.ORDER.SET IN WORD 5,
    A L.MU.ORDER.SET IN WORD 8,
    A N.MU.TF.LIST(1/2) IN WORD 6,
    A N.MU.ORDER.SET(2/2) IN WORD 6,
    A P.MU.TF.LIST       IN WORD 7,
    A S.MU.TF.LIST       IN WORD 10,
    A M.MU.TF.LIST(6/6) IN WORD 1

    EVERY MA.LINK HAS
    A MA.UE.LINK     (1/2) IN WORD 1,
    A MA.CASUALTIES (3/3) IN WORD 1
    AND BELONGS TO
    A MA.SET
    AND HAS
    A P.MA.SET             IN WORD 2,
    A S.MA.SET             IN WORD 3,
    A M.MA.SET       (4/6) IN WORD 1

    EVERY MINE.OBSTACLE HAS
    A MO.MINEFIELD    (1/2) IN WORD 1,
    A MO.X.INTER      (1/2) IN WORD 2,
    A MO.Y.INTER      (2/2) IN WORD 2
    AND BELONGS TO
    A MO.LIST
    AND HAS
    A P.MO.LIST             IN WORD 3,
    A S.MO.LIST             IN WORD 4,
    A M.MO.LIST       (4/6) IN WORD 1

    EVERY MF.POINT HAS
    A MFP.X.COORD     (1/2) IN WORD 1,
    A MFP.Y.COORD     (2/2) IN WORD 1
    AND BELONGS TO
    A MFP.LIST
    AND HAS
    A P.MFP.LIST            IN WORD 2,
    A S.MFP.LIST            IN WORD 3,
    A M.MFP.LIST            IN WORD 4

    EVERY MOVCOR.ORDER HAS
    A DESTIN.X(1/2) IN WORD 1,
    A DESTIN.Y(2/2) IN WORD 1,
    A MOV.MISSION(3/6) IN WORD 2,
    A TYPE.MOVE IN WORD 3,
    A THRESH.REIN(1/3) IN WORD 2,
    A NX.ORDER(4/6) IN WORD 2, ''WHEN MOVE IS COMPLETE
    A NX.ORD.ABOVE(5/6) IN WORD 2, ''WHEN IN PROX TO ENEMY AND ABOVE REINF THR
    A NX.ORD.BELOW(6/6) IN WORD 2   ''WHEN IN PROX TO ENEMY AND BELOW REINF TH

    EVERY MOVDIS.ORDER HAS
    A DIR.OF.MOVE IN WORD 1,        ''ADVANC OR WITHDR
    A DIST.MOVED(1/2) IN WORD 2,
    AN ORD.NEXT(2/2) IN WORD 2,
    A MOVE.TYPE IN WORD 3
```

```
1047    EVERY MOVREINF.ORD HAS
1048    A REINFORCED.UN(1/2) IN WORD 1,
1049    AN ATK.UNITS.PTR(2/2) IN WORD 1
1050
1051    EVERY OLD.SORTIE HAS
1052    AN OS.QTY                          (1/6) IN WORD 1
1053    AND BELONGS TO
1054    A SD.OLD.SORTIE.QUEUE
1055    AND HAS
1056    A P.SD.OLD.SORTIE.QUEUE        (1/2) IN WORD 2,
1057    A S.SD.OLD.SORTIE.QUEUE        (2/2) IN WORD 2,
1058    A M.SD.OLD.SORTIE.QUEUE        (2/6) IN WORD 1
1059
1060    EVERY ORDER HAS
1061    AN ORD.TYPE IN WORD 1,
1062    AN ORD.ID       IN WORD 2,
1063    AN ORD.SEQ.NO(2/2) IN WORD 4
1064    MAY BELONG TO A MU.ORDER.SET
1065    HAS
1066    A P.MU.ORDER.SET IN WORD 3,
1067    A S.MU.ORDER.SET IN WORD 5,
1068    A M.MU.ORDER.SET(1/6) IN WORD 4
1069
1070    EVERY PASSIVE.DETECTION.BASE HAS
1071    A PDB.US.LINK(1/2) IN WORD 1,
1072    A PD.OPERATOR(5/6) IN WORD 1
1073    MAY OWN
1074    A PDB.KEYED.LIST,
1075    A PDB.OP.Q
1076    BELONGS TO
1077    A SIDE.PDB.SET
1078    HAS
1079    A F.PDB.KEYED.LIST(1/2) IN WORD 2,
1080    A L.PDB.KEYED.LIST(2/2) IN WORD 2,
1081    A N.PDB.KEYED.LIST(1/2) IN WORD 3,
1082    A F.PDB.OP.Q(2/2) IN WORD 3,
1083    A L.PDB.OP.Q(1/2) IN WORD 4,
1084    A N.PDB.OP.Q(2/2) IN WORD 4,
1085    A P.SIDE.PDB.SET(1/2) IN WORD 5,
1086    A S.SIDE.PDB.SET(2/2) IN WORD 5,
1087    A M.SIDE.PDB.SET(4/6) IN WORD 1
1088
1089    EVERY PD.DET.UNIT HAS
1090    A PD.D.BTRY(1/3) IN WORD 1,
1091    A PD.D.CEP(2/3) IN WORD 1,
1092    A PD.D.PD(3/3) IN WORD 1,
1093    A PD.D.PRIORITY(1/2) IN WORD 3
1094    BELONGS TO
1095    A PDB.OP.Q
1096    HAS
1097    A P.PDB.OP.Q(1/2) IN WORD 2,
1098    A S.PDB.OP.Q(2/2) IN WORD 2,
1099    A M.PDB.OP.Q(4/6) IN WORD 3
1100
1101    EVERY PATH.POINT HAS
1102    A PP.X.POINT(1/2) IN WORD 1,
1103    AND A PP.Y.POINT(2/2) IN WORD 1,
1104    BELONGS TO
```

```
1105    A PATH.SET
1106    HAS
1107    A P.PATH.SET(1/2) IN WORD 2,
1108    A S.PATH.SET(2/2) IN WORD 2,
1109    A M.PATH.SET(1/6) IN WORD 3
1110
1111    EVERY PIR.REC.TARGET HAS
1112    A PIR.RT.UNIT IN WORD 1
1113    BELONGS TO
1114    A PIR.RECORD.LIST
1115    OWNS
1116    A PIR.RTD.LIST
1117    HAS
1118    A P.PIR.RECORD.LIST(1/2) IN WORD 2,
1119    A S.PIR.RECORD.LIST(2/2) IN WORD 2,
1120    A M.PIR.RECORD.LIST(1/6) IN WORD 3,
1121    A F.PIR.RTD.LIST(2/2) IN WORD 3,
1122    A L.PIR.RTD.LIST(1/2) IN WORD 4,
1123    A N.PIR.RTD.LIST(2/2) IN WORD 4
1124
1125    EVERY PIR.RTD.LINK HAS
1126    A PIR.RTD.TE(1/6) IN WORD 1,       ''TYPE EQUIPMENT PTR
1127    A PIR.RTD.ELEM.PROB(2/3) IN WORD 1,   ''PROB OF A SINGLE ELEMENT BEING DETECT
1128    A PIR.RTD.QUANT(3/3) IN WORD 1    ''QUANT OF EQUIPMENT DETECTED
1129    BELONGS TO
1130    A PIR.RTD.LIST
1131    HAS
1132    A P.PIR.RTD.LIST(1/2) IN WORD 2,
1133    A S.PIR.RTD.LIST(2/2) IN WORD 2,
1134    A M.PIR.RTD.LIST(2/6) IN WORD 1
1135
1136    EVERY POINT HAS
1137    A P.X(1/2) IN WORD 1,
1138    AND A P.Y(2/2) IN WORD 1,
1139    AND BELONGS TO
1140    AN UN.PATH
1141    HAS
1142    A P.UN.PATH(1/2) IN WORD 2,
1143    A S.UN.PATH(2/2) IN WORD 2,
1144    A M.UN.PATH(1/6) IN WORD 3
1145
1146    EVERY REINF.ORDER HAS
1147    A SUC.REINF.OP(1/2) IN WORD 1,
1148    AN UNSUC.REINF.OP(2/2) IN WORD 1
1149
1150    EVERY RPV.DET.CANDIDATE HAS
1151    AN RPV.DC.UNIT(1/3) IN WORD 1,
1152    AN RPV.DC.LEG.DIST(2/3) IN WORD 1,
1153    AN RPV.DC.DIST(3/3) IN WORD 1
1154    AND BELONGS TO
1155    AN RPV.CAND.DET.LIST
1156    HAS
1157    A P.RPV.CAND.DET.LIST(1/2) IN WORD 2,
1158    A S.RPV.CAND.DET.LIST(2/2) IN WORD 2,
1159    A M.RPV.CAND.DET.LIST(1/6) IN WORD 3
1160
1161    EVERY SENSOR.INTERSECT HAS
1162    AN SI.X.ENTRY                    (1/2) IN WORD 1,
```

```
1163    AN SI.Y.ENTRY                    (2/2) IN WORD 1,
1164    AN SI.X.EXIT                     (1/2) IN WORD 2,
1165    AN SI.Y.EXIT                     (2/2) IN WORD 2,
1166    AN SI.TIME.TIL.INTERSECT              IN WORD 3,
1167    AN SI.AD.UNIT                    (1/3) IN WORD 4,
1168    AN SI.ALTITUDE                   (2/3) IN WORD 4
1169    AND BELONGS TO
1170    AN SI.LIST
1171    AND HAS
1172    A P.SI.LIST                 (1/2) IN WORD 5,
1173    A S.SI.LIST                 (2/2) IN WORD 5,
1174    A M.SI.LIST                 (5/6) IN WORD 4
1175
1176    EVERY SEGMENT HAS
1177    A SEG.LENGTH(1/3) IN WORD 1,
1178    A SEG.UNIT(2/3) IN WORD 1,
1179    A SEG.TYPE(3/3) IN WORD 1,
1180    AND BELONGS TO
1181    A UN.SEGMENT.LIST
1182    HAS
1183    A P.UN.SEGMENT.LIST IN WORD 2,
1184    A S.UN.SEGMENT.LIST IN WORD 4,
1185    A M.UN.SEGMENT.LIST(1/6) IN WORD 3
1186
1187    EVERY TB.TM.LINK HAS
1188    A TB.TM.RAP(1/6) IN WORD 1,  ''1=RAP,2=NONRAP
1189    A TB.TM.CLASS IN WORD 2,''HE,ICM,PGM,ILLUM,SMOKE
1190    A TB.TM(2/6) IN WORD 1,   ''INDEX TO TYPE MUNITION
1191    A TB.TM.FIRED(2/2) IN WORD 1
1192    BELONGS TO
1193    A TB.TM.LIST
1194    HAS
1195    A P.TB.TM.LIST(1/2) IN WORD 3,
1196    A S.TB.TM.LIST(2/2) IN WORD 3,
1197    A M.TB.TM.LIST(3/6) IN WORD 1
1198
1199    EVERY TR.DET.LINK HAS
1200    A TR.DET.TE(1/3) IN WORD 1,      ''TYPE EQUIPMENT PTR
1201    A TR.DET.ELEM.PROB(2/3) IN WORD 1,  ''PROB OF A SINGLE ELEMENT BEING DETECTE
1202    A TR.DET.QUANT(3/3) IN WORD 1   ''QUANT OF EQUIPMENT DETECTED
1203    BELONGS TO
1204    A TR.DET.LIST
1205    HAS
1206    A P.TR.DET.LIST(1/2) IN WORD 2,
1207    A S.TR.DET.LIST(2/2) IN WORD 2,
1208    A M.TR.DET.LIST(1/6) IN WORD 3
1209
1210    EVERY TU.NTE.LINK HAS
1211    A TU.NTE.ID IN WORD 1           ''PTR TO TYPE EQUIPMENT
1212    BELONGS TO
1213    A TU.NTE.SET
1214    HAS
1215    A P.TU.NTE.SET(1/2) IN WORD 2,
1216    A S.TU.NTE.SET(2/2) IN WORD 2,
1217    A M.TU.NTE.SET(1/6) IN WORD 3
1218
1219    EVERY TU.TE.LINK HAS
1220    A TU.TE.ID(1/3) IN WORD 1,           ''PTR TO TYPE EQUIPMENT
```

```
1221    A TU.CRITICAL.EQUIP.INDIC(2/3) IN WORD 1,
1222    A TU.TE.QUANT(3/3) IN WORD 1          ''QUANTITY OF EQUIP WITH ID = TU EQUIP
1223    BELONGS TO
1224    A TU.TE.LIST
1225    HAS
1226    A P.TU.TE.LIST(1/2) IN WORD 2,
1227    A S.TU.TE.LIST(2/2) IN WORD 2,
1228    A M.TU.TE.LIST(1/6) IN WORD 3
1229
1230    EVERY TYPE.TEAM HAS
1231    A TT.UNIT(2/2) IN WORD 5,
1232    A TT.IN.USE(1/2) IN WORD 1,
1233    A TT.SIDE(4/6) IN WORD 1,
1234    AND A TT.TYPE IN WORD 2,          ''INF PLAT, MECH PLAT, ETC
1235    AND OWNS
1236    A PATH.SET,
1237    AND BELONGS TO
1238    A TEAM.TYPES
1239    HAS
1240    A F.PATH.SET(1/2) IN WORD 3,
1241    A L.PATH.SET(2/2) IN WORD 3,
1242    A N.PATH.SET(1/2) IN WORD 4,
1243    A P.TEAM.TYPES(2/2) IN WORD 4,
1244    A S.TEAM.TYPES(1/2) IN WORD 5,
1245    A M.TEAM.TYPES(5/6) IN WORD 1
1246
1247    EVERY UE.LINK HAS
1248    A UE.ID(1/3) IN WORD 1,             ''EQUIPMENT INDEX
1249    A UE.CRITICAL.EQUIP.INDIC(3/6) IN WORD 1,
1250    A UE.QUANT(3/3) IN WORD 1,          ''QUANTITY OF EQUIPMENT
1251    BELONGS TO
1252    A UN.EQUIP.LIST
1253    OWNS
1254    A UE.TARGET.LIST,   ''EQUIPMENT IN LOS AND DETECTED
1255    A SO.LIST,
1256    AN UE.WEAPON.SET
1257    MAY OWN
1258    A HF.SO.LIST
1259    HAS
1260    A P.UN.EQUIP.LIST(1/2) IN WORD 2,
1261    A S.UN.EQUIP.LIST(2/2) IN WORD 2,
1262    A M.UN.EQUIP.LIST(4/6) IN WORD 1,
1263    A F.UE.TARGET.LIST IN WORD 3,
1264    A L.UE.TARGET.LIST IN WORD 9,
1265    A N.UE.TARGET.LIST(1/3) IN WORD 4,
1266    A F.SO.LIST(2/2) IN WORD 5,
1267    A L.SO.LIST(1/2) IN WORD 5,
1268    A N.SO.LIST(2/3) IN WORD 4,
1269    A F.UE.WEAPON.SET      IN WORD 6,
1270    A L.UE.WEAPON.SET      IN WORD 7,
1271    A N.UE.WEAPON.SET(5/6) IN WORD 4,
1272    A F.HF.SO.LIST(1/2) IN WORD 8,
1273    A L.HF.SO.LIST(2/2) IN WORD 8,
1274    A N.HF.SO.LIST(6/6) IN WORD 4
1275
1276    EVERY US.LINK HAS
1277    A US.SENSOR.TYPE(1/6) IN WORD 1,
1278    A US.MODEL(2/6) IN WORD 1,
```

```
1279    A US.UNIT(2/2) IN WORD 1,
1280    A US.FDC(3/3) IN WORD 4,
1281    A US.ID(1/2) IN WORD 2,
1282    A US.STATUS(2/2) IN WORD 2,
1283    A US.EQ.ID (2/3) IN WORD 4
1284    BELONGS TO
1285    A UN.SENSOR.LIST,
1286    AN AVAIL.AO.LIST  ''MOVED HERE FROM AO %8DEC80_%RWF
1287    HAS A P.UN.SENSOR.LIST(1/2) IN WORD 3,
1288    A S.UN.SENSOR.LIST(2/2) IN WORD 3,
1289    A M.UN.SENSOR.LIST(1/6) IN WORD 4,
1290    A M.AVAIL.AO.LIST(2/6) IN WORD 4,
1291    A P.AVAIL.AO.LIST(1/2) IN WORD 5,
1292    A S.AVAIL.AO.LIST(2/2) IN WORD 5
1293
1294    EVERY VISIBLE.UNIT HAS
1295    A VU.POINTER(1/2) IN WORD 1,
1296    A VU.STATUS(5/6)  IN WORD 1,
1297    A VU.PREV.ENG (6/6) IN WORD 1
1298    AND BELONGS TO
1299    A UN.LOS.LIST
1300    MAY BELONG TO
1301    A UN.HC.LOS.LIST,
1302    A HC.UN.LOS.LIST
1303    HAS
1304    A P.UN.LOS.LIST IN WORD 2,
1305    A S.UN.LOS.LIST IN WORD 6,
1306    A M.UN.LOS.LIST(1/6) IN WORD 3,
1307    A P.HC.UN.LOS.LIST IN WORD 4,
1308    A S.HC.UN.LOS.LIST IN WORD 7,
1309    A M.HC.UN.LOS.LIST(2/6) IN WORD 3,
1310    A P.UN.HC.LOS.LIST IN WORD 5,
1311    A S.UN.HC.LOS.LIST IN WORD 8,
1312    A M.UN.HC.LOS.LIST(3/6) IN WORD 3
1313
1314    EVERY WEAPON HAS
1315    A WPN.ID(1/2) IN WORD 1,  '' POINTER TO TYPE.WEAPON
1316    A (WPN.AC.MUNS(4/6),
1317    HC.WPN.TYPE (4/6)) IN WORD 1,
1318    A WPN.QUANTITY(1/2) IN WORD 2,
1319    AND SOME WPN.ROUNDS.REMAINING(2/2) IN WORD 2,
1320    AND BELONGS TO
1321    A UE.WEAPON.SET
1322    HAS A P.UE.WEAPON.SET      IN WORD 3,
1323    A S.UE.WEAPON.SET      IN WORD 4,
1324    A M.UE.WEAPON.SET(5/6) IN WORD 1
1325
1326    ''**************************PROCESSES* SECTION THREE **********************
1327
1328    PROCESSES
1329
1330    EVERY AC.ATK.TGT HAS
1331    AN AAT.CMSN                (1/2) IN WORD 6,
1332    AN AAT.AD.UNIT             (3/3) IN WORD 6,
1333    AN AAT.RANGE               (1/3) IN WORD 7,
1334    AN AAT.VIS.IND             (4/6) IN WORD 6,
1335    AN AAT.AC.KILLED.IND       (3/6) IN WORD 7,
1336    AN AAT.EX.AATGT                  IN WORD 9
```

```
1337    AND OWNS
1338    AN AATT.LIST
1339    AND HAS
1340    AN F.AATT.LIST                (2/2) IN WORD 7,
1341    AN L.AATT.LIST                (1/2) IN WORD 8,
1342    AN N.AATT.LIST                (2/2) IN WORD 8
1343
1344    EVERY AIRBORNE.RADAR HAS
1345    AN AR.X.START(1/2) IN WORD 7,
1346    AN AR.Y.START(2/2) IN WORD 7,
1347    AN AR.Y.FINISH(1/2) IN WORD 8,
1348    AN AR.DURATION(2/2) IN WORD 8
1349    OWNS
1350    AN AR.DET.TGT.LIST,
1351    AN AR.CAND.DET.LIST
1352    HAS
1353    A F.AR.DET.TGT.LIST(1/3) IN WORD 9,
1354    A L.AR.DET.TGT.LIST(2/3) IN WORD 9,
1355    A F.AR.CAND.DET.LIST(1/2) IN WORD 6,
1356    A L.AR.CAND.DET.LIST(2/2) IN WORD 6,
1357    A N.AR.DET.TGT.LIST(5/6) IN WORD 9,
1358    A N.AR.CAND.DET.LIST(6/6) IN WORD 9
1359
1360    EVERY AIR.OBSERVER HAS
1361    AN AO.VELOCITY      (2/2) IN WORD 12,
1362    AN AO.CURRENT.TR(1/2) IN WORD 7,
1363    AN AO.US.LINK(2/2) IN WORD 7,
1364    AN AO.X.START(1/2) IN WORD 8,
1365    AN AO.Y.START(2/2) IN WORD 8
1366    OWNS
1367    AN AO.FLIGHT.LEG.LIST,
1368    AN AO.CAND.DET.LIST,
1369    AN AO.DET.TGT.LIST
1370    HAS
1371    A F.AO.FLIGHT.LEG.LIST(1/2) IN WORD 9,
1372    A L.AO.FLIGHT.LEG.LIST(2/2) IN WORD 9,
1373    A F.AO.CAND.DET.LIST(1/2) IN WORD 10,
1374    A L.AO.CAND.DET.LIST(2/2) IN WORD 10,
1375    A F.AO.DET.TGT.LIST(1/2) IN WORD 11,
1376    A L.AO.DET.TGT.LIST(2/2) IN WORD 11,
1377    A N.AO.FLIGHT.LEG.LIST(1/6) IN WORD 12,
1378    A N.AO.CAND.DET.LIST(2/6) IN WORD 12,
1379    A N.AO.DET.TGT.LIST(3/6) IN WORD 12
1380
1381    EVERY ARTY.ASSESS HAS
1382    A AA.DESTRUCT.INDIC IN WORD 7,
1383    A AA.FIRE.MISSION(1/2) IN WORD 8,
1384    A AA.UNIT(2/2) IN WORD 8
1385    AND OWNS
1386    A AA.SET
1387    HAS
1388    A F.AA.SET IN WORD 9,
1389    A L.AA.SET IN WORD 10,
1390    A N.AA.SET(1/2) IN WORD 6
1391
1392    EVERY ASSESSMENT HAS
1393    A AS.PK IN WORD 7,  ''IN PER CENT
1394    A AS.TGT.EQUIP(1/2) IN WORD 8,
```

```
1395    A AS.TGT.UNIT(2/2) IN WORD 8,
1396    A AS.FIRING.UNIT(1/2) IN WORD 9,
1397    A AS.SHOOT.OUT(2/2) IN WORD 9,
1398    A AS.DESTRUCT.INDIC(1/6) IN WORD 6
1399
1400    EVERY CAS.MISSION HAS
1401    A CMSN.SEQ.NR                (1/3) IN WORD 6,
1402    A CMSN.TYPE                  (3/6) IN WORD 6,     ''ON.CALL OR PREPLANNED
1403    A CMSN.Q.FLAG                (4/6) IN WORD 6,
1404    A CMSN.SIDE                  (5/6) IN WORD 6,
1405    A CMSN.AC.TYPE               (6/6) IN WORD 6,
1406    A CMSN.NR.AC                 (1/6) IN WORD 7,
1407    A CMSN.NR.ABORTED            (2/6) IN WORD 7,
1408    A CMSN.NR.SURV.AC            (3/6) IN WORD 7,
1409    A CMSN.TGT.UNIT              (3/3) IN WORD 7,
1410    A CMSN.SCOREBOARD                  IN WORD 18,
1411    A CMSN.REQUEST.TIME                IN WORD 9,     ''HOURS
1412    A CMSN.START.TIME                IN WORD 10,        ''HOURS
1413    A CMSN.TAKE.OFF.TIME               IN WORD 11,        ''HOURS
1414    A CMSN.FLIGHT.TIME               IN WORD 12,        ''HOURS
1415    A CMSN.FIRST.PASS.TIME                IN WORD 13,        ''HOURS
1416    A CMSN.ASP.STATUS              (4/6) IN WORD 7,         ''BUSY OR IDLE
1417    A CMSN.P1.ADU.RANGE          (1/3) IN WORD 14,
1418    A CMSN.P3.ADU.RANGE          (2/3) IN WORD 14,
1419    A CMSN.ANGLE                       IN WORD 15
1420    AND OWNS
1421    A CFPS.LIST
1422    AND BELONGS TO
1423    A SD.CMSN.QUEUE
1424    AND HAS
1425    A F.CFPS.LIST            (2/2) IN WORD 8,
1426    A L.CFPS.LIST            (1/2) IN WORD 16,
1427    A N.CFPS.LIST            (3/3) IN WORD 14,
1428    A P.SD.CMSN.QUEUE        (2/2) IN WORD 16,
1429    A S.SD.CMSN.QUEUE        (1/2) IN WORD 17,
1430    A M.SD.CMSN.QUEUE        (4/6) IN WORD 17
1431
1432    EVERY FIRE.MISSION HAS
1433    A FM.TGT(1/2) IN WORD 7,           ''PNTR TO TARGET.REPORT
1434    A FM.BTRY(1/3) IN WORD 8,          ''BATTERY ASSIGNED MISSION
1435    A FM.STATUS(3/6) IN WORD 8,        ''ACTIVE OR QUEUED
1436    A FM.RAP.FLAG(4/6) IN WORD 8,      ''1=RAPMSN,2=NONRAP
1437    A FM.RANGE(1/3) IN WORD 9,         ''ESTIMATED RANGE IN DECAMETERS
1438    A FM.N.VOLS(5/6) IN WORD 8,        ''NUMBER OF VOLLEYS TO BE FIRED
1439    A FM.Q.SIZE (3/6) IN WORD 9,       ''# OF FM'S IN FM.BTRY.QUEUE
1440    A FM.TM(6/6) IN WORD 8,            ''ROUND TYPE TO BE FIRED
1441    A FM.TM.CLASS IN WORD 10,          '' "HE" OR "ICM"
1442    A FM.PRIORITY(2/2) IN WORD 9,      ''SAME AS TARGET REPORT
1443    A FM.START.TIME IN WORD 11,
1444    A FM.PREP.TIME IN WORD 18,         ''FOR PGM MSNS
1445    A FM.TOF.TIME  IN WORD 17,         ''FOR PGM MSNS
1446    A FM.FIRED.VOLS(2/2) IN WORD 7,    ''VOLLEYS ALREADY FIRED
1447    A FM.EX.FIRE.MISSION      IN WORD 19
1448    MAY BELONG TO
1449    A TR.FM.LIST,
1450    A BN.CAN.FM.SET,                   ''CANDIDATE FIRE MISSION SET
1451    A BY.SCHD.LIST,
1452    A BY.FM.QUEUE,
```

```
1453    A FO.CUR.FM.LIST
1454    HAS
1455    A P.TR.FM.LIST(1/2) IN WORD 12,
1456    A S.TR.FM.LIST(2/2) IN WORD 12,
1457    A P.BN.CAN.FM.SET(1/2) IN WORD 13,
1458    A S.BN.CAN.FM.SET(2/2) IN WORD 13,
1459    A P.BY.SCHD.LIST(1/2) IN WORD 14,
1460    A S.BY.SCHD.LIST(2/2) IN WORD 14,
1461    A P.BY.FM.QUEUE(1/2) IN WORD 15,
1462    A S.BY.FM.QUEUE(2/2) IN WORD 15,
1463    A P.FO.CUR.FM.LIST(1/2) IN WORD 16,
1464    A S.FO.CUR.FM.LIST(2/2) IN WORD 16,
1465    A M.TR.FM.LIST(1/6) IN WORD 6,
1466    A M.BN.CAN.FM.SET(2/6) IN WORD 6,
1467    A M.BY.SCHD.LIST(3/6) IN WORD 6,
1468    A M.BY.FM.QUEUE(4/6) IN WORD 6,
1469    A M.FO.CUR.FM.LIST(5/6) IN WORD 6
1470
1471    EVERY FORWARD.OBSERVER HAS              ''TIME REFERENCE IS MINUTES
1472    AN FO.US.LINK(1/2) IN WORD 7,
1473    AN FO.REL.DIRECTION IN WORD 9,
1474    AN FO.CURRENT.TR(1/2) IN WORD 8,
1475    AN FO.FST.INDIC(2/2) IN WORD 8,
1476    AN FO.EX.FWD.OBSERVER IN WORD 13
1477    OWNS
1478    AN FO.CAND.DET.LIST,
1479    AN FO.TGT.RPT.LIST,
1480    AN FO.CUR.FM.LIST
1481    HAS
1482    A F.FO.CAND.DET.LIST(1/2) IN WORD 10,
1483    A L.FO.CAND.DET.LIST(2/2) IN WORD 10,
1484    A F.FO.TGT.RPT.LIST(1/2) IN WORD 11,
1485    A L.FO.TGT.RPT.LIST(2/2) IN WORD 11,
1486    A F.FO.CUR.FM.LIST(1/2) IN WORD 12,
1487    A L.FO.CUR.FM.LIST(2/2) IN WORD 12,
1488    A N.FO.CAND.DET.LIST(1/3) IN WORD 6,
1489    A N.FO.TGT.RPT.LIST(2/3) IN WORD 6,
1490    A N.FO.CUR.FM.LIST(3/3) IN WORD 6
1491
1492    EVERY HC.ARRIVE.BATTLE HAS
1493    AN AB.FARRP(1/2) IN WORD 7,
1494    AN AB.TEAM(2/2) IN WORD 7
1495
1496    EVERY HC.RETURN.FARRP HAS
1497    AN RF.FARRP(1/2) IN WORD 7,
1498    AN RF.TEAM(2/2) IN WORD 7
1499
1500    EVERY HELICOPTER.FIRE HAS
1501    AN ATK.HELICOPTER(1/2) IN WORD 7,
1502    AN HF.ENEMY.UNIT(2/2) IN WORD 7,
1503    A  HEL.IS.KILLED.INDIC(1/6) IN WORD 8,
1504    AN HF.RESULTS(2/3) IN WORD 8,
1505    AN HF.DESTRUCT.INDIC(2/6) IN WORD 8,
1506    AN HF.FIRING.TABLE IN WORD 12,
1507    AN HF.RANGE(2/2) IN WORD 9,
1508    AN HF.TEAM IN WORD 10,
1509    AN HF.REINFORCE.IND(5/6) IN WORD 8,
1510    A P.HF.SO.LIST(1/2) IN WORD 11,
```

```
1511    A S.HF.SO.LIST(2/2) IN WORD 11,
1512    A M.HF.SO.LIST(6/6) IN WORD 8
1513    AND BELONGS TO
1514    AN HF.SO.LIST
1515
1516    EVERY HEL.TARGET.ACQUISITION HAS
1517    AN HTA.TEAM(1/2) IN WORD 7,
1518    AN ENEMY.UNITS(2/2) IN WORD 7,
1519    AN HTA.BATTLE(1/2) IN WORD 8,
1520    AN HTA.REINFORCE.IND(4/6) IN WORD 8
1521
1522    EVERY HOW.REPAIR HAS
1523    A HR.HOW IN WORD 7
1524
1525    EVERY MINE.ASSESS HAS
1526    A MA.UNIT          (1/2) IN WORD 7,
1527    A MA.MINEFIELD     (2/2) IN WORD 7,
1528    A MA.BATTLE.ENDED (1/2) IN WORD 8,
1529    A MA.MSN (1/2) IN WORD 10
1530    AND OWNS
1531    A MA.SET
1532    AND HAS
1533    A F.MA.SET                IN WORD 9,
1534    A L.MA.SET                IN WORD 11,
1535    A N.MA.SET         (2/2) IN WORD 8
1536
1537    EVERY PHOTO.IR.FLIGHT HAS
1538    A PIR.US.LINK(1/2) IN WORD 7,
1539    A PIR.X.START(2/2) IN WORD 7,
1540    A PIR.Y.START(1/2) IN WORD 8
1541    OWNS
1542    A PIR.FLIGHT.LEG.LIST,
1543    A PIR.RECORD.LIST
1544    HAS
1545    A F.PIR.FLIGHT.LEG.LIST(2/2) IN WORD 8,
1546    A L.PIR.FLIGHT.LEG.LIST(1/2) IN WORD 9,
1547    A F.PIR.RECORD.LIST(2/2) IN WORD 9,
1548    A L.PIR.RECORD.LIST(1/2) IN WORD 6,
1549    A N.PIR.FLIGHT.LEG.LIST(4/6) IN WORD 6,
1550    A N.PIR.RECORD.LIST(3/3) IN WORD 6
1551
1552    EVERY REMOTE.PILOT.VEHICLE HAS
1553    A RPV.CURRENT.TARGET(1/2) IN WORD 7,
1554    A RPV.US.LINK(2/2) IN WORD 7,
1555    A RPV.X.START(1/2) IN WORD 8,
1556    A RPV.Y.START(2/2) IN WORD 8
1557    OWNS
1558    A RPV.FLIGHT.LEG.LIST,
1559    A RPV.CAND.DET.LIST
1560    HAS
1561    A F.RPV.CAND.DET.LIST(1/2) IN WORD 10,
1562    A L.RPV.CAND.DET.LIST(2/2) IN WORD 10,
1563    A N.RPV.CAND.DET.LIST(2/2) IN WORD 11,
1564    A F.RPV.FLIGHT.LEG.LIST(1/2) IN WORD 11,
1565    A L.RPV.FLIGHT.LEG.LIST(1/2) IN WORD 6,
1566    A N.RPV.FLIGHT.LEG.LIST(2/2) IN WORD 6
1567
1568    EVERY SHOOT.OUT HAS
```

```
1569    A SO.AIR.ATK.INDIC     (5/6) IN WORD 7,
1570    A FIRING.EQUIP(1/2) IN WORD 8,
1571    AN FIRER.UNIT(2/2) IN WORD 8,
1572    A DROP.DEAD.INDICATOR(2/3) IN WORD 7,
1573    A SO.RESULTS(6/6) IN WORD 7,
1574    A SO.DESTRUCT.INDIC(1/6) IN WORD 7,
1575    A SO.HELICOPTER(1/2) IN WORD 9,
1576    A SO.FIRING.TABLE IN WORD 10,
1577    A SO.EX.SHOOT.OUT IN WORD 11,
1578    AND BELONGS TO
1579    AN SO.LIST
1580    HAS
1581    A P.SO.LIST(1/2) IN WORD 6,
1582    A S.SO.LIST(2/2) IN WORD 6,
1583    A M.SO.LIST(2/6) IN WORD 7
1584
1585    EVERY TARGET.REPORT HAS
1586    A TR.MISSION.TYPE IN WORD 6,
1587    A TR.FDC(3/3) IN WORD 16,
1588    A TR.FDC.STATUS(2/6) IN WORD 7,
1589    A TR.SENSOR.TYPE  IN WORD 8,
1590    A TR.SENSOR.ID(1/2) IN WORD 9,
1591    A TR.REP.UNIT(1/3) IN WORD 10,    ''UNIT REPORTING TARGET
1592    A TR.TGT.UNIT(2/3) IN WORD 10,     ''ACTUAL ENEMY UNIT
1593    A TR.MIL.WORTH(3/3) IN WORD 10,    ''TARGET PRIORITY
1594    A TR.PGM.STATUS(3/6) IN WORD 7,   ''PRECISION GUIDED MUNITION TARGET
1595    A TR.REM.EFFECTS(2/2) IN WORD 9,     ''TIMES 10000
1596    A TR.REQ.EFFECTS(1/2) IN WORD 11,    ''TIMES 100
1597    A TR.CUM.EFFECTS(2/2) IN WORD 11,    ''TIMES 10000
1598    A TR.MOVE(4/6) IN WORD 7,          ''MOVEMENT STATUS
1599    A TR.EST.X(1/2) IN WORD 12,         ''ESTIMATED X COORD TO NEAREST DECAMETER
1600    A TR.EST.Y(2/2) IN WORD 12,         ''ESTIMATED Y COORD TO NEAREST DECAMETER
1601    A TR.EST.RADIUS(1/2) IN WORD 16,   ''IN METERS
1602    A TR.EST.TU(1/2) IN WORD 13,      ''ESTIMATED TYPE UNIT
1603    A TR.RECVD.TIME IN WORD 14,
1604    A TR.ABORT.TIME IN WORD 15,     ''TIME TR NO LONGER CONSIDERED VALID
1605    A TR.CEP(2/2) IN WORD 13,           ''CIRCULAR ERROR PROBABLE OF TR LOCATION
1606    A TR.TOT.STATUS(5/6) IN WORD 7, ''0 = NONTOT,  1 = TOT, 2 = TOT FOLLOW-ON
1607    A TR.ASGND.BATS(6/6) IN WORD 7,  ''BTRYS ASGD TO FM NOW ED
1608    A TR.EX.TGT.REPORT IN WORD 23
1609    OWNS
1610    A TR.FM.LIST,
1611    A TR.DET.LIST
1612    MAY BELONG TO
1613    A FD.TR.QUEUE,           ''QUEUE OF TRS TO BE COMPUTED AT FDC
1614    A FD.COMPLETE.LIST,    ''LIST OF COMPUTED AND STILL ACTIVE TRS AT FDC
1615    A FO.TGT.RPT.LIST    ''LIST OF TR'S THAT AN FO HAS WORKING
1616    HAS
1617    A F.TR.FM.LIST(1/2) IN WORD 17,
1618    A L.TR.FM.LIST(2/2) IN WORD 17,
1619    A F.TR.DET.LIST(1/2) IN WORD 18,
1620    A L.TR.DET.LIST(2/2) IN WORD 18,
1621    A P.FD.TR.QUEUE(1/2) IN WORD 19,
1622    A S.FD.TR.QUEUE(2/2) IN WORD 19,
1623    A P.FD.COMPLETE.LIST(1/2) IN WORD 22,
1624    A S.FD.COMPLETE.LIST(2/2) IN WORD 22,
1625    A P.FO.TGT.RPT.LIST(1/2) IN WORD 20,
1626    A S.FO.TGT.RPT.LIST(2/2) IN WORD 20,
```

```
1627    A N.TR.FM.LIST(1/6) IN WORD 21,
1628    A N.TR.DET.LIST(3/6) IN WORD 21,
1629    A M.FD.TR.QUEUE(4/6) IN WORD 21,
1630    A M.FD.COMPLETE.LIST(5/6) IN WORD 21,
1631    A M.FO.TGT.RFT.LIST(6/6) IN WORD 21
1632
1633    EVERY WITH.DRAW HAS
1634    A WD.UNIT(1/2) IN WORD 6,
1635    A WD.DESTRUCT.INDIC(4/6) IN WORD 6
1636
1637    ''*********************************EVENTS** SECTION FOUR ************************
1638
1639    ''THE'' EVENT NOTICES INCLUDE CHANGE.LITE, END.SIMULATION
1640
1641    EVERY ACT.ATK HAS
1642    AN ATKUNIT(1/2) IN WORD 5,
1643    AN ATKORDER IN WORD 6,
1644    AN ATKENEMY(2/2) IN WORD 5
1645
1646    EVERY ACT.DEF HAS
1647    A DEFUNIT(1/2) IN WORD 5,
1648    A DEFORDER IN WORD 6
1649
1650    EVERY ACT.MOVCOR HAS
1651    A MCUNIT(1/2) IN WORD 5,
1652    A MCORDER IN WORD 6
1653
1654    EVERY ACT.MOVDIS HAS
1655    A MDUNIT(1/2) IN WORD 5,
1656    A MDORDER IN WORD 6
1657
1658    EVERY ACT.REINF HAS
1659    A REINUNIT(1/2) IN WORD 5,
1660    A REINORDER IN WORD 6,
1661    A EN.UN.PTR(2/2) IN WORD 5
1662
1663    EVERY AD.ENGAGEMENT HAS
1664    AN AE.INTERSECTION    (1/2) IN WORD 4,
1665    AN AE.CALLING.PROCESS (2/2) IN WORD 4
1666
1667    EVERY ARTY.OCCUPATION HAS
1668    AN AOC.BTRY IN WORD 5
1669
1670    EVERY BTL.ENDED HAS
1671    A BTL.WINNER(1/6) IN WORD 6,
1672    A BTL.BU(1/2) IN WORD 5,
1673    A BTL.RU(2/2) IN WORD 5
1674
1675    EVERY CFR.ACTIVATION HAS
1676    A CFR.ACT.BTRY(1/2) IN WORD 5
1677
1678    EVERY CFR.OFF HAS
1679    A CFR.OFF.ID(1/2) IN WORD 5,
1680    A CFR.OFF.LINK(2/2) IN WORD 5
1681
1682    EVERY CFR.ON HAS
1683    A CFR.ON.ID(1/2) IN WORD 5,
1684    A CFR.ON.LINK(2/2) IN WORD 5
```

```
1685
1686    EVERY CFR.OPERATOR HAS
1687    A CFR.SENS.ID(1/2) IN WORD 5,
1688    A CFR.DET.UN(2/2) IN WORD 5
1689
1690    EVERY CHANGE.WEATHER HAS
1691    A CW.VISIBILITY
1692
1693    EVERY DQ.OLD.SORTIE.QUEUE HAS
1694    A DOSQ.SIDE
1695
1696    EVERY ENGAGEMENT HAS
1697    A DETECTING.UNIT(2/2) IN WORD 5,
1698    A DETECTED.UNIT(1/2) IN WORD 5
1699
1700    EVERY FEBA.SORTIE HAS
1701    A FE.SECTOR(1/6) IN WORD 5,
1702    A FE.SIDE(2/6) IN WORD 5,
1703    A FE.TIME.INTERVAL IN WORD 6
1704
1705    EVERY GET.NX.ORD HAS
1706    A GET.UNIT(1/2) IN WORD 6,
1707    A GET.ORDER(2/2) IN WORD 6,
1708    A GET.OTHER(1/2) IN WORD 5,
1709    A GET.FLAG (2/2) IN WORD 5
1710
1711    EVERY HC.DEPART.BATTLE HAS
1712    A DB.FARRP(1/2) IN WORD 5,
1713    A DB.TEAM(2/2) IN WORD 5,
1714    A DB.ENEMY.UNITS IN WORD 6
1715
1716    EVERY HELO.ENGAGEMENT HAS
1717    AN HCEN.TEAM(1/2) IN WORD 5,
1718    AN HCEN.FARRP(2/2) IN WORD 5,
1719    AN HCEN.HELICOPTER(1/2) IN WORD 6,
1720    AN HCEN.ENEMY(2/2) IN WORD 6
1721
1722    EVERY INIT.PREPLAN.CAS HAS
1723    AN IPC.TARGET.UNIT      (1/3) IN WORD 4,
1724    AN IPC.AC.TYPE          (2/3) IN WORD 4,
1725    AN IPC.NR.AC            (3/3) IN WORD 4
1726
1727    EVERY MOVE HAS
1728    A MV.UNIT(1/2) IN WORD 5
1729
1730    EVERY PDB.ACTIVATION HAS
1731    A PDB.ACT.BTRY(1/2) IN WORD 5,
1732    A PDB.ACT.ACTIVITY.TYPE IN WORD 6
1733
1734    EVERY PDB.OPERATOR HAS
1735    A PDB.SENS.ID(1/2) IN WORD 5,
1736    A PDB.DET.UNIT(2/2) IN WORD 5
1737
1738    ''THE'' EVENT NOTICES INCLUDE SCHEDULE.ARTY.MOVEMENT
1739
1740    EVERY SEND.TEAM HAS
1741    A ST.FARRP(1/2) IN WORD 5,
1742    A ST.TEAM(2/2) IN WORD 5
```

```
1743
1744    EVERY START.ARTY.MOVEMENT HAS
1745    A STAM.BTRY(1/2) IN WORD 5,
1746    A STAM.DIRECTION(2/2) IN WORD 5
1747
1748    EVERY START.BATTLE HAS
1749    A TERRAIN(1/6) IN WORD 6,
1750    A BLUE.MISSION(2/6) IN WORD 6,
1751    A RED.MISSION(3/6) IN WORD 6,
1752    A BLUE.UNITS(1/2) IN WORD 5,
1753    A RED.UNITS(2/2) IN WORD 5,
1754    A ARG.ARRAY(2/2) IN WORD 6
1755
1756    EVERY START.MOVE HAS
1757    AN SM.UNIT(1/2) IN WORD 7,
1758    AN SM.XCOR(2/2) IN WORD 7,
1759    AN SM.YCOR(1/2) IN WORD 5,
1760    AN SM.TYP.MOV IN WORD 6,
1761    AN SM.ORDER IN WORD 8
1762
1763    EVERY STOP.ARTY.MOVEMENT HAS
1764    A SPAM.BTRY(1/2) IN WORD 5,
1765    A SPAM.DIRECTION(2/2) IN WORD 5
1766
1767    EVERY UPDATE.LOC HAS
1768    A UL.UNIT(1/2) IN WORD 8,
1769    A UL.XCOR(2/2) IN WORD 8,
1770    A UL.YCOR(1/2) IN WORD 5,
1771    A UL.RATE.MOVE(2/2) IN WORD 5,
1772    A UL.TYP.MOV IN WORD 6,
1773    A UL.ORDER      IN WORD 7
1774
1775    ''THE'' EVENT NOTICES INCLUDE POSITION.REPORT
1776    ''THE'' EXTERNAL EVENTS ARE SET.DEBUG AND OFF.LINE.ATTRITION
1777    SEE SET.DEBUG----------------------------------------------------------------------------------------->(392)
1778    SEE OFF.LINE.ATTRITION-------------------------------------------------------------------------------->(382)
1779    EXTERNAL EVENT UNITS ARE 5 AND 15
1780
1781    ''**********************************DEFINITIONS* SECTION FIVE ********************
1782
1783    DEFINE ''GLOBAL INTEGER VARIABLES
1784    NUM.PGM.FIRED,
1785    NUM.KILL.PGM,
1786    DEBUG,
1787    NUM.RAD.INCREMENTS,
1788    NUM.ANG.INCREMENTS,
1789    CUM.NO.BATTLES,
1790    BREAK.POINT,
1791    DISTANCE.INCREMENT,
1792    LEFT.BNDRY.INT,
1793    FRONT.DEPTH,
1794    FEBA.WIDTH,
1795    INIT.X.FEBA,
1796    INIT.Y.FEBA,
1797    HQ.SET.BACK,
1798    DIS.WITH.DRAW,
1799    DIS.ATTACK,
1800    RN.SEED,
```

```
1801    R.WIDTH.UNIT,
1802    B.WIDTH.UNIT,
1803    PK.BAND.RNG,
1804    PK.PROB,
1805    PK.VECTOR.NR,
1806    MOVE.FACTOR.NR,
1807    DFFB.MAX.RANGE,
1808    DT.MAX.BATS,
1809    CDT.MAX.VOLS,
1810    CDI.USAGE.INDICATOR,
1811    ACT.BATTLE.RANGE,           ''OPENING BATTLE RANGE IN METERS
1812    ATK.DELAY,
1813    REIN.DELAY,
1814    REIN.PROX,                  ''RANGE OF REINFORCING UNITS (METERS)
1815    REIN.THRESH,
1816    NO.ATTACK.CONFIGURATIONS,   ''RED AND BLUE COMBINED
1817    NO.SCOUT.CONFIGURATIONS,    ''RED AND BLUE COMBINED
1818    RD.ROUNDS.PER.POPUP,        ''MAX ROUNDS PER POPUP
1819    BL.ROUNDS.PER.POPUP,        ''MAX ROUNDS PER POPUP
1820    HC.SWITCH,                  ''OFF/ON SWITCH FOR HELICOPTERS
1821    HC.OUTPUT.SWITCH,           ''ON/OFF SWITCH FOR HC OUTPUT
1822    BL.MAX.FL.TIME,             ''MAX TIME ALOFT FOR BLUE TEAM (MIN)
1823    RD.MAX.FL.TIME,             ''MAX TIME ALOFT FOR RED TEAM (MIN)
1824    BL.HC.SPACING,              ''LATERAL DISTANCE BETWEEN BLUE HC (M)
1825    RD.HC.SPACING,              ''LATERAL DISTANCE BETWEEN RED HC (M)
1826    MIN.NO.SUP.UNITS,           ''MIN NO BLUE UNITS WHICH CAN BE SUPPORTED BY HC
1827    SCAN.RATE,
1828    LOC.UPDATE.FREQ,
1829    REQ.EFF.MOVING,
1830    REQ.EFF.STA,
1831    N.BLUE.TYPE.EQP,
1832    N.RED.TYPE.EQP,
1833    N.B.WPN.TYPE,
1834    N.R.WPN.TYPE,
1835    TERRAIN.PAR,
1836    SADARM.THRESHOLD,
1837    SDM.CB,
1838    SDM.PGM.RNG,
1839    SDM.TM,
1840    SDM.MNV,
1841    SDM.SSPK,
1842    SDM.MAX.RANGE,
1843    SDM.VOLLEY.RADIUS,
1844    TOT.KILL.SDM,
1845    TOT.FIRED.SDM,
1846    GOOD.WEATHER.PROB,
1847    FASCAM.VOLLEYS,
1848    ILLUM.DEBUG,
1849    ILLUM.SWITCH,
1850    MAX.FASCAM.RANGE,
1851    MIN.FASCAM.RANGE,
1852    MAX.ATT.FASCAM,
1853    MAX.WD.FASCAM,
1854    MF.DEBUG,
1855    MF.SWITCH,
1856    SMK.DEBUG,
1857    SMK.SWITCH,
1858    VISIBILITY,
```

```
1859    TACAIR.DEBUG,
1860    CAS.MSN.RPT.FLAG,
1861    TER.W.INC,
1862    NITE.VIS.PCT
1863    AS INTEGER VARIABLES
1864
1865    DEFINE ''SIGNED INTEGER VARIABLES  ****************************************
1866    MOVE.FIRE.DIST,
1867    AAT.AD.UNIT,
1868    AAT.RANGE,
1869    ACT.EQUIP.ID,
1870    ACT.SUBSTITURE,
1871    ACT.WEATHER.DEGRADE,
1872    ACT.MIN.ALT,
1873    ACT.NORM.ALT,
1874    ACT.SPEED,
1875    ACT.PASS.TIME,
1876    ACT.MAX.ALOFT,
1877    ACT.PROB.SORTIE.ABORT,
1878    ACT.P1.DIST,
1879    ACT.P2.DIST,
1880    ACT.P3.DIST,
1881    ACT.X1,
1882    ACT.X2,
1883    ACT.X3,
1884    ACT.Y1,
1885    ACT.Y2,
1886    ACT.Y3,
1887    ACT.Z1,
1888    ACT.Z2,
1889    ACT.Z3,
1890    CFPS.XSTART,
1891    CFPS.XEND,
1892    CFPS.YSTART,
1893    CFPS.YEND,
1894    COL.XENTRY,
1895    COL.XEXIT,
1896    COL.YENTRY,
1897    COL.YEXIT,
1898    CMSN.SEQ.NR,
1899    CMSN.TGT.UNIT,
1900    CMSN.P1.ADU.RANGE,
1901    CMSN.P2.ADU.RANGE,
1902    MRH.PD,
1903    SD.AIRFIELD,
1904    SD.NR.CAS.MISSIONS,
1905    SD.TP.SORTIE,
1906    SD.ASC.RADIUS,
1907    SD.NO.FLY.VIS,
1908    SD.POOR.FLY.VIS,
1909    WPN.AC.MUNS,
1910    BY.STOP.FASCAM.SUPP,
1911    DEFILADE.DIST,
1912    MFP.X.COORD,
1913    MFP.Y.COORD,
1914    TW.PK.PTR,
1915    EQUIP.PK.PTR,
1916    PK.MOV.RNG,
```

1917 TW.FIRE.OTM.PTR,
1918 THRESH.REIN,
1919 MU.OFFSET.X,
1920 MU.OFFSET.Y,
1921 DESTIN.X,
1922 DESTIN.Y,
1923 UE.QUANT,
1924 UN.TYPE.UNIT,
1925 UN.X.COORD,
1926 UN.Y.COORD,
1927 UN.X.GRID,
1928 UN.Y.GRID,
1929 UN.RADIUS,
1930 UN.FASCAM.RECVD,
1931 UN.DELAY,
1932 SM.XCOR,
1933 SM.YCOR,
1934 UL.XCOR,
1935 UL.YCOR,
1936 BAND.RANGE,
1937 BY.BN.RANK,
1938 BY.FIRE.RATE,
1939 CFR.CIR.ERROR,
1940 CFR.DET.PROB,
1941 CFR.ORIENTATION,
1942 CF.D.BTRY,
1943 CF.D.CPE,
1944 CF.D.PD,
1945 CF.D.PRIORITY,
1946 EQ.MAX.SPEED,
1947 EQ.PERSONNEL.LOAD,
1948 EFO.FO.UNIT,
1949 EFO.X.CORRECT,
1950 EFO.Y.CORRECT,
1951 EFO.X.SEARCH.GRID,
1952 EFO.Y.SEARCH.GRID,
1953 FD.FDC,
1954 FD.MIN.TIME,
1955 FD.MAX.TIME,
1956 HC.X,
1957 HC.Y,
1958 HE.COST,
1959 HE.DUST.DURATION,
1960 HE.ROUND.RAD,
1961 HE.VOL.DUST.RAD,
1962 HE.VOLLEY.RAD,
1963 HE.WEIGHT,
1964 HE.RH.TOTAL.CPE,
1965 HE.RH.ROUND.CPE,
1966 IC.RELIABILITY,
1967 IC.VOLLEY.RAD,
1968 IC.N.SUBM,
1969 IC.RH.TOTAL.CPE,
1970 IC.RH.ROUND.CPE,
1971 ILLUM.RADIUS,
1972 ILLUM.MAX.RANGE,
1973 ILLUM.DURATION,
1974 MAO.PGM.CAP,

```
1975    MAR.MIN.PREP,
1976    MAR.MAX.PREP,
1977    MA.CASUALTIES,
1978    MF.COLOR,
1979    MF.X.HIGH,
1980    MF.X.LOW,
1981    MF.Y.HIGH,
1982    MF.Y.LOW,
1983    MO.X.INTER,
1984    MO.Y.INTER,
1985    MAR.MAX.ALOFT.TIME,
1986    MPIR.MAX.PROCESS,
1987    MPIR.MIN.PROCESS,
1988    MPIR.CIR.ERROR,
1989    MRPV.MAX.PREP,
1990    MRPV.MIN.PREP,
1991    MRPV.CIR.ERROR,
1992    US.EQ.ID,
1993    SPAM.DIRECTION,
1994    SP.X.COORD,
1995    SP.Y.COORD,
1996    SP.X.GRID,
1997    SP.Y.GRID,
1998    STAM.DIRECTION,
1999    SMK.WIDTH,
2000    SMK.MAX.RANGE,
2001    SMK.BURN.TIME,
2002    TB.HOW.EQ.ID,
2003    TB.SUST.FIRE.RATE,
2004    TB.MAX.RANGE,
2005    TB.SFAIL.MEAN.RNDS,
2006    TB.LFAIL.MEAN.RNDS,
2007    TB.SFAIL.REPAIR,
2008    TB.LFAIL.REPAIR,
2009    TB.SUPPRESS.TIME,
2010    TB.MIN.PREP,
2011    TB.MAX.PREP,
2012    TB.MIN.FEBA,
2013    TB.MAX.FEBA,
2014    TE.DELTA.T,
2015    TE.HEIGHT,
2016    TU.MIL.WORTH,
2017    TU.FREQ,
2018    TU.RADIUS,
2019    TU.CRIT.NO,
2020    TU.MOV.RATE,
2021    TU.OPP.PRIORITY,
2022    TU.SUP.PRIORITY,
2023    TU.MF.FACTOR,
2024    TW.RATE.OF.FIRE,
2025    TW.ROUND.VELOCITY,
2026    TW.MAX.RANGE,
2027    TW.MIN.RANGE,
2028    TW.BASIC.LOAD,
2029    TW.TYPE.OF.SENSOR,
2030    TW.SPECTRUM,
2031    TW.HFOV,
2032    TW.VFOV,
```

```
2033    TW.HFOS,
2034    TW.VFOS,
2035    AO.DC.UNIT,
2036    AO.DC.LEG.DIST,
2037    AO.DC.DIST,
2038    FT.TGT.UNIT,
2039    FT.TARGET.EQUIP,
2040    FT.SCORE1,
2041    FT.SCORE2,
2042    DECISION.POINT,
2043    FR.CRIT.NO,
2044    HW.BTRY,
2045    HW.SFAIL.RNDS,
2046    HW.LFAIL.RNDS,
2047    PP.X.POINT,
2048    PP.Y.POINT,
2049    PIR.RTD.ELEM.PROB,
2050    PIR.RTD.QUANT,
2051    P.X,
2052    P.Y,
2053    RPV.DC.UNIT,
2054    RPV.DC.LEG.DIST,
2055    RPV.DC.DIST,
2056    SEG.LENGTH,
2057    SEG.UNIT,
2058    SEG.TYPE,
2059    TR.DET.TE,
2060    TR.DET.ELEM.PROB,
2061    TR.DET.QUANT,
2062    TU.TE.ID,
2063    TU.CRITICAL.EQUIP.INDIC,
2064    TU.TE.QUANT,
2065    UE.ID,
2066    FM.BTRY,
2067    DROP.DEAD.INDICATOR,
2068    TR.REM.EFFECTS,
2069    TR.REP.UNIT,
2070    TR.TGT.UNIT,
2071    TR.MIL.WORTH,
2072    TR.EST.X,
2073    TR.EST.Y,
2074    SI.X.ENTRY,
2075    SI.Y.ENTRY,
2076    SI.X.EXIT,
2077    SI.Y.EXIT,
2078    TR.EST.RADIUS
2079    AS SIGNED INTEGER VARIABLES
2080
2081
2082    DEFINE ''ALPHA VARIABLES  ************************************************
2083    AM.NAME,
2084    MADS.NAME,
2085    CT.NAME,
2086    DIR.OF.MOVE,
2087    EN.NAME,
2088    EQ.NAME,
2089    FB.MISSION,
2090    FMM.ID,
```

```
2091    FM.TM.CLASS,
2092    FZ.NAME,
2093    GP.NAME,
2094    HE.ID,
2095    IC.ID,
2096    ILLUM.ID,
2097    KV.WPN.NAME,
2098    MAO.NAME,
2099    MAR.NAME,
2100    MCFR.NAME,
2101    MFO.NAME,
2102    MF.ID,
2103    MN.NAME,
2104    MOVE.TYPE,
2105    MPDB.NAME,
2106    MPIR.NAME,
2107    MRPV.NAME,
2108    ORD.TYPE,
2109    PDB.ACT.ACTIVITY.TYPE,
2110    PGM.MARK,
2111    PT.NAME,
2112    SMK.ID,
2113    SDM.MARK,
2114    SM.NAME,
2115    SM.TYP.MOV,
2116    ST.NAME,
2117    TB.NAME,
2118    TB.TM.CLASS,
2119    TB.WPN.NAME,
2120    TE.NAME,
2121    TR.MISSION.TYPE,
2122    TR.SENSOR.TYPE,
2123    TU.LEVEL,
2124    TW.NAME,
2125    TYPE.MOVE,
2126    UL.TYP.MOV,
2127    WD.NAME,
2128    WS.NAME,
2129    HC.DEBUG
2130    AS ALPHA VARIABLES
2131
2132    DEFINE ''REAL VARIABLES   **************************************************
2133    AA.FRACTION,
2134    AS.PK,
2135    ACT.ANGLE.P1.P2,
2136    CFPS.TIME.LENGTH,
2137    CMSN.ANGLE,
2138    CMSN.FIRST.PASS.TIME,
2139    CMSN.FLIGHT.TIME,
2140    CMSN.REQUEST.TIME,
2141    CMSN.START.TIME,
2142    CMSN.TAKE.OFF.TIME,
2143    SI.TIME.TIL.INTERSECT,
2144    SM.MIN.TANK.RATIO,
2145    SM.MAX.TANK.RATIO,
2146    BMNT,
2147    CFR.LAST.ON.OR.OFF,
2148    DF.TIME,
```

```
2149    EFO.SEARCH.TIME,
2150    EFO.PERIOD.OF.SEARCH,
2151    EFO.START.TIME,
2152    EFM.PREP.TIME,
2153    EFM.FIRE.RATE,
2154    EFM.SUPPRESS,
2155    ESO.ACQ.TIME,
2156    EENT,
2157    FE.TIME.INTERVAL,
2158    FM.START.TIME,
2159    FO.REL.DIRECTION,
2160    FT.PK,
2161    FT.PK.BAR,
2162    IC.TB.INTERCEPT,
2163    IC.TB.SLOPE,
2164    IF.V.TIME,
2165    MAO.MAX.PREP,
2166    PGM.RELY,
2167    MAO.MIN.PREP,
2168    MOV.FAC,
2169    NITE.MOV.FAC,
2170    POS.REP.INT,
2171    STOP.SIMULATION.TIME,
2172    TAC.MOV.FAC,
2173    TIME.BETWEEN.ARTY.MOVE,
2174    TR.ABORT.TIME,
2175    TR.RECVD.TIME,
2176    TT.LOS.SHAPE,
2177    TT.LOS.SCALE,
2178    TT.NLOS.SHAPE,
2179    TT.NLOS.SCALE,
2180    UN.TIME.LAST.MOVE,
2181    TW.NITE.FAC,
2182    PK.MOV.FAC,
2183    BL.ATK.FAIL.PROB,       ''ATTACK FAILURE PROBABILITY PER MISSION
2184    BL.SCT.FAIL.PROB,       ''SCOUT FAILURE PROBABILITY PER MISSION
2185    RD.ATK.FAIL.PROB,       ''ATTACK FAILURE PROBABILITY PER MISSION
2186    RD.SCT.FAIL.PROB,       ''SCOUT FAILURE PROBABILITY PER MISSION
2187    PK.F.MOV.FAC,
2188    BL.MIN.HANDOFF.TIME,    ''MIN TIME TO HAND OFF TARGETS
2189    RD.MIN.HANDOFF.TIME,    ''MIN TIME TO HAND OFF TARGETS (SEC)
2190    BL.MAX.HANDOFF.TIME,    ''MAX TIME TO HAND OFF TARGETS (SEC)
2191    RD.MAX.HANDOFF.TIME,    ''MAX TIME TO HAND OFF TARGETS (SEC)
2192    BL.MIN.MASK.TIME,       ''MIN TIME TO REMAIN MASKED (SEC)
2193    RD.MIN.MASK.TIME,       ''MIN TIME TO REMAIN MASKED (SEC)
2194    BL.MAX.MASK.TIME,       ''MAX TIME TO REMAIN MASKED (SEC)
2195    RD.MAX.MASK.TIME,       ''MAX TIME TO REMAIN MASKED (SEC)
2196    BL.MIN.UNMASK.TIME,     ''MIN TIME TO REMAIN UNMASKED (SEC)
2197    RD.MIN.UNMASK.TIME,     ''MIN TIME TO REMAIN UNMASKED (SEC)
2198    BL.MAX.UNMASK.TIME,     ''MAX TIME TO REMAIN UNMASKED (SEC)
2199    RD.MAX.UNMASK.TIME,     ''MAX TIME TO REMAIN UNMASKED (SEC)
2200    BL.LOW.FRAC.RANGE,      ''SMALLEST FRACTION OF HC'S LONGEST RANGE WPN AT
2201    RD.LOW.FRAC.RANGE,      ''WHICH IT WILL BE POSITIONED FROM CLOSEST OPP UNT
2202    BL.HIGH.FRAC.RANGE,     ''LARGEST FRACTION OF HC'S LONGEST RANGE
2203    RD.HIGH.FRAC.RANGE,     ''WEAPON AT WHICH IT WILL BE POSITIONED
2204    HT.ARR.BTL.TIME,
2205    EAAT.DETECT.TIME,
2206    EAAT.FIRING.TIME,
```

```
2207    EAAT.TIME.TO.LEA,
2208    EAAT.WAIT.TIME
2209    AS REAL VARIABLES
2210
2211    ''DEFINE SETS  ***************************************************
2212    DEFINE AO.CAND.DET.LIST AS A SET RANKED BY LOW AO.DC.LEG.DIST
2213    DEFINE AO.DET.TGT.LIST, AR.DET.TGT.LIST  AS FIFO SETS
2214    DEFINE AO.EB.SET AS A SET RANKED BY LOW AO.EB.ALTITUDE
2215    DEFINE AO.FLIGHT.LEG.LIST AS A FIFO SET
2216    DEFINE AO.RB.SET AS A SET RANKED BY LOW AO.RB.RANGE
2217    DEFINE AVAIL.AO.LIST AS A FIFO SET
2218    DEFINE BN.BTRY.SET AS A SET RANKED BY LOW BY.BN.RANK
2219    DEFINE BN.CAN.FM.SET AS A SET RANKED BY  LOW FM.Q.SIZE, THEN BY  ''                \1
2220    LOW FM.RANGE
2221    DEFINE BY.FM.QUEUE AS A SET RANKED BY HIGH FM.PRIORITY
2222    DEFINE BY.HOW.SET AS A LIFO SET
2223    DEFINE BY.SCHD.LIST AS A SET RANKED BY LOW FM.START.TIME
2224    DEFINE CF.OP.Q. AS A SET RANKED BY HIGH CF.D.PRIORITY
2225    DEFINE COL.SET                              AS A SET RANKED BY LOW COL.XENTRY
2226    DEFINE DF.RATE.LIST AS A FIFO SET
2227    DEFINE FD.COMPLETE.LIST AS A LIFO SET
2228    DEFINE FD.SCHD.LIST AS A SET RANKED BY LOW FS.START
2229    DEFINE FD.TR.QUEUE AS A SET RANKED BY HIGH TR.MIL.WORTH
2230    DEFINE FO.CUR.FM.LIST AS A FIFO SET
2231    DEFINE FO.CAND.DET.LIST AS A SET RANKED BY LOW FO.DC.RANGE
2232    DEFINE MFO.RB.SET AS A SET RANKED BY LOW FO.RB.RANGE
2233    DEFINE FO.TGT.RPT.LIST AS A LIFO SET
2234    DEFINE IF.RATE.LIST AS A FIFO SET
2235    DEFINE FP.SET AS A FIFO SET
2236    DEFINE MU.ORDER.SET AS A SET
2237    DEFINE MU.TF.LIST AS A SET
2238    DEFINE MADS.RH.SET                          AS A SET RANKED BY LOW MRH.RANGE
2239    DEFINE PDB.OP.Q AS A SET RANKED BY HIGH PD.D.PRIORITY
2240    DEFINE RPV.CAND.DET.LIST AS A SET
2241    DEFINE RPV.FLIGHT.LEG.LIST AS A FIFO SET
2242    DEFINE SIDE.PDB.SET AS A FIFO SET
2243    DEFINE SIDE.CFR.SET AS A FIFO SET
2244    DEFINE SS.SET AS A SET  RANKED BY HIGH UN.X.COORD
2245    DEFINE FD.BN.LIST AS A FIFO SET
2246    DEFINE TB.SORT.LIST AS A SET RANKED BY LOW UN.Y.COORD
2247    DEFINE TB.TM.LIST AS A FIFO SET
2248    DEFINE TR.DET.LIST AS A LIFO SET
2249    DEFINE TR.FM.LIST AS A LIFO SET
2250    DEFINE UN.EQUIP.LIST, TU.NTE.SET,  AND TU.TE.LIST AS LIFO SETS
2251    DEFINE  UNIT.SET AND UN.SUB.LIST AS FIFO SETS
2252
2253    ''DEFINE ARRAYS   ************************************************
2254    DEFINE AR.DET.COEFF AS A 2-DIMENSIONAL REAL ARRAY
2255    DEFINE SHADE AS A 1-DIMENSIONAL ALPHA ARRAY
2256    DEFINE PK.POINTER      AS AN INTEGER 2-DIMENSIONAL EXTENDED ARRAY
2257    DEFINE PK.DEF.POINTER AS AN INTEGER 2-DIMENSIONAL EXTENDED ARRAY
2258    DEFINE TGT.OTM         AS AN INTEGER 2-DIMENSIONAL EXTENDED ARRAY
2259    DEFINE FIRE.OTM        AS AN INTEGER 2-DIMENSIONAL EXTENDED ARRAY
2260    DEFINE ATK.HC, SCT.HC AS ALPHA, 1-DIM ARRAYS
2261    DEFINE ANALYSIS AS A 1-DIMENSIONAL INTEGER ARRAY
2262    DEFINE STY.BLUE.EXP, STY.RED.EXP AS INTEGER 2-DIMENSIONAL ARRAYS
2263    DEFINE PGM.HIT, PGM.KILL AS REAL 2-DIMENSIONAL ARRAYS
2264    DEFINE PGM.LINK, FD.UNIT AS INTEGER 1-DIMENSIONAL ARRAYS
```

```
2265
2266    ''DEFINE FUNCTIONS   ****************************************************
2267    DEFINE ACT.RANGE AS A REAL FUNCTION WITH 2 ARGUMENTS
2268    DEFINE AR.PROB.DETECT AS A REAL FUNCTION WITH 2 ARGUMENTS
2269    DEFINE BTRY.AVAILABLE AS AN INTEGER FUNCTION WITH 3 ARGUMENTS
2270    DEFINE COLLISION AS AN INTEGER FUNCTION WITH 2 ARGUMENTS
2271    DEFINE COMBINATIONS AS A REAL FUNCTION WITH 2 ARGUMENTS
2272    DEFINE EST.RANGE AS A REAL FUNCTION WITH 2 ARGUMENTS
2273    DEFINE EST.TR.RANGE AS A REAL FUNCTION WITH 2 ARGUMENTS
2274    DEFINE FEBA.BAND AS AN INTEGER FUNCTION WITH 1 ARGUMENT
2275    DEFINE FILE.FD.SCHD AS A ROUTINE WITH 4 ARGUMENTS
2276    DEFINE HE.WLA AS A REAL FUNCTION WITH 6 ARGUMENTS
2277    DEFINE ICM.WLA AS A REAL FUNCTION WITH 4 ARGUMENTS
2278    DEFINE STAY.TIME AS A REAL FUNCTION WITH 1 ARGUMENT
2279
2280    ''*****************************SUBSTITUTIONS* SECTION SIX *********************
2281
2282    ''SDDL SUBSTITUTIONS
2283    DEFINE ENDPREAMBLE     TO MEAN END
2284    DEFINE ENDMAIN         TO MEAN END
2285    DEFINE EXITMAIN        TO MEAN STOP
2286    DEFINE ENDROUTINE      TO MEAN END
2287    DEFINE EXITROUTINE     TO MEAN RETURN
2288    DEFINE ENDFUNCTION     TO MEAN END
2289    DEFINE EXITFUNCTION    TO MEAN RETURN
2290    DEFINE LEFTROUTINE     TO MEAN LEFT ROUTINE
2291    DEFINE ENDLEFTROUTINE  TO MEAN END
2292    DEFINE EXITLEFTROUTINE TO MEAN RETURN
2293    DEFINE ENDEVENT        TO MEAN END
2294    DEFINE EXITEVENT       TO MEAN RETURN
2295    DEFINE ENDPROCESS      TO MEAN END
2296    DEFINE EXITPROCESS     TO MEAN RETURN
2297    DEFINE ELSEIF          TO MEAN ELSE IF
2298    DEFINE ENDIF           TO MEAN ALWAYS
2299    DEFINE ENDLOOP         TO MEAN REPEAT
2300    DEFINE EXITLOOP        TO MEAN LEAVE
2301    DEFINE LOOP            TO MEAN RESUME SUBSTITUTION
2302
2303    DEFINE HOURS   TO MEAN UNITS
2304    DEFINE MINUTES TO MEAN / 60.0 UNITS
2305    DEFINE SECONDS TO MEAN / 3600.0 UNITS
2306
2307    DEFINE NORTH TO MEAN PI.C/2  ''                                    \OPTIMIZE
2308    DEFINE SOUTH TO MEAN 3.*PI.C/2 ''                                  \OPTIMIZE
2309    DEFINE EAST  TO MEAN 0.0
2310    DEFINE WEST  TO MEAN PI.C
2311
2312    ''CT.GROUP DEFINITIONS
2313    DEFINE MANEUVER  TO MEAN 1
2314    DEFINE ARTILLERY TO MEAN 2
2315    DEFINE SUPPORT   TO MEAN 3
2316
2317    DEFINE FEBA.FLIGHT TO MEAN 1
2318
2319    DEFINE ACTIVE TO MEAN 1
2320    DEFINE HOLD   TO MEAN 0
2321
2322    ''PD.OPERATOR VALUES
```

```
2323    DEFINE BUSY   TO MEAN 1
2324    DEFINE IDLE   TO MEAN 0
2325
2326    DEFINE TRUE   TO MEAN 1
2327    DEFINE FALSE  TO MEAN 0
2328    DEFINE SADARM TO MEAN 2
2329
2330    DEFINE DAY    TO MEAN 1
2331    DEFINE NITE   TO MEAN 2
2332
2333    ''FUZE TYPES
2334    DEFINE PD     TO MEAN 1
2335    DEFINE VT     TO MEAN 2
2336
2337    ''UN.STATUS
2338    DEFINE ADVANCING   TO MEAN 1
2339    DEFINE WITHDRAWING TO MEAN 2
2340    DEFINE STATIONARY  TO MEAN 3
2341    DEFINE MOVING      TO MEAN 4
2342    DEFINE ADV.TO.WITH TO MEAN 5
2343    DEFINE STA.TO.WITH TO MEAN 6
2344
2345    DEFINE YES TO MEAN 1
2346    DEFINE NO  TO MEAN 2
2347
2348    DEFINE RED.TK.PLT    TO MEAN 1
2349    DEFINE RED.MECH.PLT  TO MEAN 2
2350    DEFINE BLUE.TK.PLT   TO MEAN 3
2351    DEFINE RED.CO.HQ     TO MEAN 4
2352    DEFINE BLUE.MECH.PLT TO MEAN 5
2353    DEFINE BLUE.CO.HQ    TO MEAN 6
2354    DEFINE RED.INF.PLT   TO MEAN 7
2355    DEFINE BLUE.INF.PLT  TO MEAN 8
2356    DEFINE RED.AH.TEAM   TO MEAN 9
2357    DEFINE BLUE.AH.TEAM  TO MEAN 10
2358
2359    ''MN.NAME AND UN.MISSION DEFINITIONS
2360    DEFINE PATROL TO MEAN 1
2361    DEFINE PROBE  TO MEAN 2
2362    DEFINE ATTACK TO MEAN 3
2363    DEFINE DELAY  TO MEAN 4
2364    DEFINE DEFEND TO MEAN 5
2365    DEFINE AMBUSH TO MEAN 6
2366
2367    DEFINE BLUE TO MEAN 2
2368    DEFINE RED  TO MEAN 1
2369
2370    DEFINE HIT  TO MEAN 1
2371    DEFINE MISS TO MEAN 2
2372
2373    ''TERRAIN.PAR SUBSTITUTIONS
2374    DEFINE FULDA        TO MEAN 21
2375    DEFINE NO.GER.PLAIN TO MEAN 22
2376    DEFINE MIX.TERRAIN  TO MEAN 23
2377
2378    ''HC.TYPE
2379    DEFINE SCOUT TO MEAN 1
2380    ''ATTACK DEFINED ABOVE TO MEAN 3
```

```
2381
2382    ''HC.ALTITUDE
2383    DEFINE UNMASKED  TO MEAN 1
2384    DEFINE MASKED    TO MEAN 2
2385    DEFINE ON.GROUND TO MEAN 3
2386
2387    ''HT.STATUS
2388    DEFINE READY                    TO MEAN 1
2389    DEFINE MOVING.TO.BATTLE         TO MEAN 2
2390    DEFINE RETURNING.FROM.BATTLE    TO MEAN 3
2391    DEFINE DETECTING                TO MEAN 4
2392    DEFINE ENGAGING                 TO MEAN 5
2393    DEFINE REARMING.REFUELING       TO MEAN 6
2394    DEFINE RELEASED                 TO MEAN 7
2395
2396    ''CMSN.Q.FLAG
2397    DEFINE NO.FLY.VIS TO MEAN 1
2398    ''NITE DEFINED ABOVE TO MEAN 2
2399    DEFINE SORTIE.RATE TO MEAN 3
2400    DEFINE AIR.SPACE TO MEAN 4
2401
2402    ''CMSN.TYPE
2403    DEFINE PREPLANNED TO MEAN 1
2404    DEFINE ON.CALL TO MEAN 2
2405
2406    ''MADS.FCM
2407    DEFINE LOOK.SHOOT.LOOK                  TO MEAN 1
2408    DEFINE RIPPLE                           TO MEAN 2
2409    DEFINE SALVO                            TO MEAN 3
2410
2411    DEFINE NO.SHOT.MADE                     TO MEAN 3
2412    DEFINE NO.PK.PTR                        TO MEAN 4
2413    DEFINE ATGM   TO MEAN 1 ''ANTI-TANK GUIDED MISSILE''
2414    DEFINE FFATGM TO MEAN 2  ''FIRE & FORGET ATGM''
2415
2416
2417    ''************SECTION 7 - EXTENDED MEMORY ***********
2418
2419
2420    EXTENDED ENTITIES INCLUDE
2421
2422    ''PERMANENT ENTITIES
2423
2424    AC.MUNS,
2425    AC.TYPE,
2426    AO.ELEVATION.BAND,
2427    AO.RANGE.BAND,
2428    BTRY,
2429    CATEGORY,
2430    CFR.RNG.HACK,
2431    DIST.FROM.FEBA.BAND,
2432    ENVIRONMENT,
2433    EQUIPMENT,
2434    FASCAM.MUNITION,
2435    FA.BN,
2436    FDC,
2437    KILLER.VICTIM,
2438    SIDE,
```

```
2439    FO.RANGE.BAND,
2440    FUZE,
2441    GROUPING,
2442    HE.MUNITION,
2443    HE.RANGE.HACK,
2444    IC.MUNITION,
2445    IC.RANGE.HACK,
2446    ILLUM.MUNITION,
2447    MINEFIELD,
2448    MISSION,
2449    MODEL.AD.SENSOR,
2450    MODEL.AO,
2451    MODEL.AR,
2452    MODEL.CFR,
2453    MODEL.FO,
2454    MODEL.PDB,
2455    MODEL.PIR,
2456    MODEL.RPV,
2457    PDB.RNG.HACK,
2458    PK.BAND,
2459    PK.F.MOVE.FACTOR,
2460    PK.MOVE.BAND,
2461    PK.MOVE.FACTOR,
2462    PK.VECTOR,
2463    POSTURE,
2464    SEARCH.POINT,
2465    SECTOR,
2466    SENSOR.TYPE,
2467    SIDE,
2468    SMOKE.MUNITION,
2469    SUBMUNITION,
2470    TERRAIN.TYPE,
2471    LOS.BAND,
2472    TYPE.BATTLE.FIELD,
2473    TYPE.BTRY,
2474    TYPE.EQUIPMENT,
2475    TYPE.UNIT,
2476    TYPE.WEAPON,
2477    UNIT,
2478
2479    ''TEMPORARY ENTITIES
2480
2481    AA.LINK,
2482    ATK.ORDER,
2483    AWARE.UNIT,
2484    DEF.ORDER,
2485    EX.AC.ATK.TGT,
2486    EX.FWD.OBSERVER,
2487    EX.FIRE.MISSION,
2488    EX.SHOOT.OUT,
2489    EX.TGT.REPORT,
2490    FIRING.TABLE,
2491    MAN.UNIT,
2492    MA.LINK,
2493    MF.POINT,
2494    MINE.OBSTACLE,
2495    MOVCOR.ORDER,
2496    MOVDIS.ORDER,
```

```
2497     MOVREINF.ORD,
2498     ORDER,
2499     REINF.ORDER,
2500     SEGMENT,
2501     VISIBLE.UNIT,
2502     WEAPON
2503
2504 ENDPREAMBLE
```

```
2505 ROUTINE PLAT.COUNT                                                    \UNNECESSARY? U011
2506     GIVEN
2507     UNITS
2508     YIELDING
2509     NO.INF.PLAT,
2510     NO.MECH.PLAT,
2511     NO.ARMOR.PLAT,
2512     NO.AH.TEAM,
2513     NO.HQ
2514
2515     ADD 1 TO ANAL.CTR(264,1)                                          \DYN_ANAL
2516     NORMALLY MODE IS INTEGER
2517     DEFINE UNITS AS AN INTEGER, 1-DIMENSIONAL ARRAY
2518     LET NO.UNITS = DIM.F(UNITS(*))
2519     LOOP
2520     FOR I = 1 TO NO.UNITS
2521     DO THE FOLLOWING
2522        IF UN.TYPE.UNIT(UNITS(I)) = RED.INF.PLT
2523           OR UN.TYPE.UNIT(UNITS(I)) = BLUE.INF.PLT,
2524           ADD 1 TO NO.INF.PLAT
2525        ELSE
2526           IF UN.TYPE.UNIT(UNITS(I)) = RED.MECH.PLT
2527              OR UN.TYPE.UNIT(UNITS(I)) = BLUE.MECH.PLT,
2528              ADD 1 TO NO.MECH.PLAT
2529           ELSE
2530              IF UN.TYPE.UNIT(UNITS(I)) = RED.TK.PLT
2531                 OR UN.TYPE.UNIT(UNITS(I)) = BLUE.TK.PLT
2532                 ADD 1 TO NO.ARMOR.PLAT
2533              ELSE
2534                 IF UN.TYPE.UNIT(UNITS(I)) = RED.AH.TEAM
2535                    OR UN.TYPE.UNIT(UNITS(I)) = BLUE.AH.TEAM,
2536                    ADD 1 TO NO.AH.TEAM
2537                 ELSE
2538                    IF UN.TYPE.UNIT(UNITS(I)) = RED.CO.HQ
2539                       OR UN.TYPE.UNIT(UNITS(I)) = BLUE.CO.HQ,
2540                       ADD 1 TO NO.HQ
2541                    ALWAYS
2542                 ALWAYS
2543              ALWAYS
2544           ALWAYS
2545        ALWAYS
2546     ENDLOOP
2547 <--EXITROUTINE
2548 ENDROUTINE
```

SCIENCE APPLICATIONS, INC. Hugh Jones

FINAL DESIGN REPORT
FOR THE
STUDY ENTITLED
"COSAGE ANALYSIS AND DESIGN REPORT"
VOLUME II
(VAX COSAGE SOURCE CODE PROCESSED WITH SAI-SDDL)
(continued)

Contract expiration date:
April 29, 1984

Prepared for:
U.S. Army - Concepts Analysis Agency
Bethesda MD 20014
Mr. Hugh Jones

Prepared by:
Science Applications, Inc.
La Jolla, CA 92038
Mr. Donald A. Heimburger
Ms. Marcia A. Metcalfe
Ms. Suellen S. Worrells
Ms. Diane K. Graham



VAX COSAGE SOURCE CODE PROCESSED WITH SAI-SDDL

This volume contains the COSAGE SIMSCRIPT source code as it is implemented on the VAX computer. This source code has been processed with SAI-SDDL . This tool provides the formatting and document summary information automatically.

```
  1    1  CROSS_REFERENCING

  2    2  REVISIONS

  3    5  PREAMBLE

  4    5  PERMANENT_ENTITIES

  5   19  TEMPORARY_ENTITIES

  6   33  PROCESSES

  7   39  EVENTS
  8  392  . SET.DEBUG
  9  382  . OFF.LINE.ATTRITION
 10  493  . : SHOOT.OUT
 11  324  . : . DECIDE
 12  114  . : . : PREP.WITHDRAW
 13  604  . : . : . ERROR.STOP
 14  618  . : . : . : SNAP.R
 15  614  . : . : . : . SNAP2
 16  381  . : . : . MOVE
 17   83  . : . : . : CHANGE.LOC
 18  321  . : . : . : . COMPUTE.D
 19  122  . : . : . : . SEGMENT.ADJUST
 20  107  . : . : . : . : NEW.SEGMENT
 21  342  . : . : . : . : . RANGE.COMPUTE
 22  369  . : . : . : . : . ENGAGEMENT
 23  369  . : . : . : . : . : ENGAGEMENT
 24       . : . : . : . : . : ** RECURSIVE INVOCATION -- LINE 22
 25  604  . : . : . : . : . : ERROR.STOP
 26       . : . : . : . : . : ** SEE LINE 13
 27  493  . : . : . : . : . : SHOOT.OUT
 28       . : . : . : . : . : ** RECURSIVE INVOCATION -- LINE 10
 29  215  . : . : . : . : . : REQUEST.DEF.FASCAM
 30  342  . : . : . : . : . : . RANGE.COMPUTE
 31  216  . : . : . : . : . : . REQUEST.FASCAM
 32  604  . : . : . : . : . : . : ERROR.STOP
 33       . : . : . : . : . : . : ** SEE LINE 13
 34  470  . : . : . : . : . : . : TARGET.REPORT
 35  590  . : . : . : . : . : . : . TR.INPUT
 36  334  . : . : . : . : . : . : . FINISH.COMPUTATION
 37  332  . : . : . : . : . : . : . : FDC.TR.DEQ
 38  470  . : . : . : . : . : . : . : . TARGET.REPORT
 39       . : . : . : . : . : . : . : . ** RECURSIVE INVOCATION -- LINE 34
 40  239  . : . : . : . : . : . : . TARGET.ANALYSIS
 41  100  . : . : . : . : . : . : . : LOCATE.SECTOR
 42  632  . : . : . : . : . : . : . : EST.RANGE
 43  231  . : . : . : . : . : . : . : SIZE.ESTIMATE
 44  604  . : . : . : . : . : . : . : . ERROR.STOP
 45       . : . : . : . : . : . : . : . ** SEE LINE 13
 46  631  . : . : . : . : . : . : . : . COMBINATIONS
 47  604  . : . : . : . : . : . : . : . : ERROR.STOP
 48       . : . : . : . : . : . : . : . : ** SEE LINE 13
 49  180  . : . : . : . : . : . : . : EST.MIL.WORTH
 50  168  . : . : . : . : . : . : . CHK.COMP.TR
 51  171  . : . : . : . : . : . : . : COMPARE.TRS
 52  663  . : . : . : . : . : . : . : . TIME.REQ
```

```
 53  662 . : . : . : . : . : . : . : . : . : . : . : . : . PROXIMITY.REQ
 54  633 . : . : . : . : . : . : . : . : . : . : . : . : . EST.TR.RANGE
 55  169 . : . : . : . : . : . : . : . : . : . : . : . CHK.FD.TR
 56  171 . : . : . : . : . : . : . : . : . : . : . : . : COMPARE.TRS
 57      . : . : . : . : . : . : . : . : . : . : . : . : ** SEE LINE 51
 58  170 . : . : . : . : . : . : . : . : . : . : . : . : COMBINE.TRS
 59  333 . : . : . : . : . : . : . : . : . : . : . : . FDC.TR.ENQ
 60  634 . : . : . : . : . : . : . : . : . : . : . : . FEBA.BAND
 61  100 . : . : . : . : . : . : . : . : . : . : . : . : LOCATE.SECTOR
 62  194 . : . : . : . : . : . : . : . : . : . : . : . FIND.START.TIME
 63  208 . : . : . : . : . : . : . : . : . : . : . : . PGM.MSN.ASGN
 64  632 . : . : . : . : . : . : . : . : . : . : . : . : EST.RANGE
 65  336 . : . : . : . : . : . : . : . : . : . : . : . : GET.TERRAIN
 66  642 . : . : . : . : . : . : . : . : . : . : . : . : WEIBULL.F
 67  274 . : . : . : . : . : . : . : . : . : . : . : . : CLEAN.UP.FIRE.MISSIONS
 68  480 . : . : . : . : . : . : . : . : . : . : . : . : FIRE.MISSION
 69  164 . : . : . : . : . : . : . : . : . : . : . : . : . BTRY.FM.ENQ
 70  438 . : . : . : . : . : . : . : . : . : . : . : . : . FORWARD.OBSERVER
 71   98 . : . : . : . : . : . : . : . : . : . : . : . : . : LOCATE.SEARCH.AREA
 72  628 . : . : . : . : . : . : . : . : . : . : . : . : . : ACT.RANGE
 73  254 . : . : . : . : . : . : . : . : . : . : . : . : . : FO.DETECTION
 74  111 . : . : . : . : . : . : . : . : . : . : . : . : . : . PRED.POS
 75  644 . : . : . : . : . : . : . : . : . : . : . : . : . : . : ANGLE.COMPUTE
 76  641 . : . : . : . : . : . : . : . : . : . : . : . : . : . NORMAL.F
 77  470 . : . : . : . : . : . : . : . : . : . : . : . : . : TARGET.REPORT
 78      . : . : . : . : . : . : . : . : . : . : . : . : . : ** RECURSIVE INVOCATION — LINE 34
 79  246 . : . : . : . : . : . : . : . : . : . : . : . : . VOLLEY
 80  100 . : . : . : . : . : . : . : . : . : . : . : . : . : LOCATE.SECTOR
 81  628 . : . : . : . : . : . : . : . : . : . : . : . : . : ACT.RANGE
 82  386 . : . : . : . : . : . : . : . : . : . : . : . : . : PDB.ACTIVATION
 83  628 . : . : . : . : . : . : . : . : . : . : . : . : . : . ACT.RANGE
 84  206 . : . : . : . : . : . : . : . : . : . : . : . : . : . PDB.DETECTION
 85  205 . : . : . : . : . : . : . : . : . : . : . : . : . : . : NOISE.DEGRADE
 86  628 . : . : . : . : . : . : . : . : . : . : . : . : . : . : . ACT.RANGE
 87  387 . : . : . : . : . : . : . : . : . : . : . : . : . : . : PDB.OPERATOR
 88  387 . : . : . : . : . : . : . : . : . : . : . : . : . : . : . PDB.OPERATOR
 89      . : . : . : . : . : . : . : . : . : . : . : . : . : . : . ** RECURSIVE INVOCATION — LINE 87
 90  641 . : . : . : . : . : . : . : . : . : . : . : . : . : . : . NORMAL.F
 91  168 . : . : . : . : . : . : . : . : . : . : . : . : . : . : . CHK.COMP.TR
 92      . : . : . : . : . : . : . : . : . : . : . : . : . : . : . ** SEE LINE 50
 93  169 . : . : . : . : . : . : . : . : . : . : . : . : . : . : . CHK.FD.TR
 94      . : . : . : . : . : . : . : . : . : . : . : . : . : . : . ** SEE LINE 55
 95  470 . : . : . : . : . : . : . : . : . : . : . : . : . : . : . TARGET.REPORT
 96      . : . : . : . : . : . : . : . : . : . : . : . : . : . : . ** RECURSIVE INVOCATION — LINE 34
 97  360 . : . : . : . : . : . : . : . : . : . : . : . : . : . : CFR.ON
 98  359 . : . : . : . : . : . : . : . : . : . : . : . : . : . : . CFR.OFF
 99  360 . : . : . : . : . : . : . : . : . : . : . : . : . : . : . : CFR.ON
100      . : . : . : . : . : . : . : . : . : . : . : . : . : . : . : ** RECURSIVE INVOCATION — LINE 97
101  100 . : . : . : . : . : . : . : . : . : . : . : . : . : . : . LOCATE.SECTOR
102  628 . : . : . : . : . : . : . : . : . : . : . : . : . : . : . ACT.RANGE
103  358 . : . : . : . : . : . : . : . : . : . : . : . : . : CFR.ACTIVATION
104  628 . : . : . : . : . : . : . : . : . : . : . : . : . : . ACT.RANGE
105  166 . : . : . : . : . : . : . : . : . : . : . : . : . : CFR.DETECTION
106  165 . : . : . : . : . : . : . : . : . : . : . : . : . : . CFR.DEGRADE
107  362 . : . : . : . : . : . : . : . : . : . : . : . : . : . CFR.OPERATOR
108  362 . : . : . : . : . : . : . : . : . : . : . : . : . : . . CFR.OPERATOR
109      . : . : . : . : . : . : . : . : . : . : . : . : . : . : . ** RECURSIVE INVOCATION — LINE 107
110  641 . : . : . : . : . : . : . : . : . : . : . : . : . : . : . NORMAL.F
```

```
111   168                                                    CHK.COMP.TR
112                                                          ** SEE LINE 50
113   169                                                    CHK.FD.TR
114                                                          ** SEE LINE 55
115   470                                                    TARGET.REPORT
116                                                          ** RECURSIVE INVOCATION — LINE 34
117   250                                          MINE.EFFECTS
118   103                                          MINE.DELAY
119   216                                          REQUEST.FASCAM
120                                                ** RECURSIVE INVOCATION — LINE 31
121   604                                          ERROR.STOP
122                                                ** SEE LINE 13
123   464                                          MINE.ASSESS
124   493                                          SHOOT.OUT
125                                                ** RECURSIVE INVOCATION — LINE 10
126   611                                          OUTPUT.ATTRITION
127    89                                          FA.BN.MOVEMENT
128   100                                          LOCATE.SECTOR
129   393                                          START.ARTY.MOVEMENT
130   134                                            CHECK.FOR.MINES
131   250                                            MINE.EFFECTS
132                                                  ** RECURSIVE INVOCATION — LINE 117
133   336                                            GET.TERRAIN
134   407                                            STOP.ARTY.MOVEMENT
135   355                                              ARTY.OCCUPATION
136    89                                                FA.BN.MOVEMENT
137                                                      ** RECURSIVE INVOCATION — LINE 127
138   611                                          OUTPUT.ATTRITION
139   159                                          ATTRIT.SENSOR
140   428                                          AIR.OBSERVER
141   604                                            ERROR.STOP
142                                                  ** SEE LINE 13
143   154                                            AO.DETECTION
144   242                                              UNIT.ENVIR
145   641                                                NORMAL.F
146   336                                                GET.TERRAIN
147   641                                              NORMAL.F
148   470                                            TARGET.REPORT
149                                                  ** RECURSIVE INVOCATION — LINE 34
150   438                                          FORWARD.OBSERVER
151                                                ** SEE LINE 70
152   387                                          PDB.OPERATOR
153                                                ** SEE LINE 87
154   360                                          CFR.ON
155                                                ** SEE LINE 97
156   359                                          CFR.OFF
157                                                ** SEE LINE 98
158   362                                          CFR.OPERATOR
159                                                ** SEE LINE 107
160   200                                          ILLUM.EFFECTS
161   218                                          REQUEST.ILLUM
162   604                                          ERROR.STOP
163                                                ** SEE LINE 13
164   342                                          RANGE.COMPUTE
165   470                                          TARGET.REPORT
166                                                ** RECURSIVE INVOCATION — LINE 34
167   604                                          ERROR.STOP
168                                                ** SEE LINE 13
```

```
169  123  TIME.TO.DETECT
170  342  . RANGE.COMPUTE
171  120  . SEARCH
172  323  . . CONTRAST.TO.FREQ
173  344  . . TEMPERATURE.ATTENUATION
174  646  . . MRT.TO.FREQ
175  661  . . JOHNSON.CRITERIA
176  340  . . PROB.INF
177  341  . . PROB.TIME
178  369  . ENGAGEMENT
179       . ** RECURSIVE INVOCATION — LINE 22
180  234  SMOKE.EFFECTS
181   *   IF
182  222  REQUEST.SMOKE
183  604  . ERROR.STOP
184       . ** SEE LINE 13
185  342  . RANGE.COMPUTE
186  470  . TARGET.REPORT
187       . ** RECURSIVE INVOCATION — LINE 34
188  604  ERROR.STOP
189       ** SEE LINE 13
190  381  MOVE
191       ** RECURSIVE INVOCATION — LINE 16
192   83  CHANGE.LOC
193       ** RECURSIVE INVOCATION — LINE 17
194   80  BLOCK.LOS
195  369  . ENGAGEMENT
196       . ** RECURSIVE INVOCATION — LINE 22
197  487  . ASSESSMENT
198  604  . . ERROR.STOP
199       . . ** SEE LINE 13
200  369  . . ENGAGEMENT
201       . . ** RECURSIVE INVOCATION — LINE 22
202  487  . . ASSESSMENT
203       . . ** RECURSIVE INVOCATION — LINE 197
204  611  . . OUTPUT.ATTRITION
205  419  . . AC.ATK.TGT
206  307  . . . AD.SHOOT
207  428  . . . . AIR.OBSERVER
208       . . . . ** SEE LINE 140
209  504  . . . . CAS.MISSION
210  367  . . . . . DQ.OLD.SORTIE.QUEUE
211  327  . . . . . . DQ.CMSN.QUEUE
212  285  . . . . . . . CHECK.CAS.CONSTRAINTS
213   *   . . . . . . . . IF
214  504  . . . . . . . . CAS.MISSION
215       . . . . . . . . ** RECURSIVE INVOCATION — LINE 209
216  293  . . . . . END.CAS.MISSION
217  349  . . . . . . AD.ENGAGEMENT
218  641  . . . . . . . NORMAL.F
219  307  . . . . . . . AD.SHOOT
220       . . . . . . . ** RECURSIVE INVOCATION — LINE 206
221   65  . . . . . . . FILE.KAD.SENSOR
222  493  . . . . . . SHOOT.OUT
223       . . . . . . ** RECURSIVE INVOCATION — LINE 10
224  487  . . . . . . ASSESSMENT
225       . . . . . . ** RECURSIVE INVOCATION — LINE 197
226  314  . . . . . FLIGHT.PATH
```

```
 227  645                                                
```

```
 285                                                    ** SEE LINE 13
 286   381                                              MOVE
 287                                                    ** RECURSIVE INVOCATION -- LINE 16
 288    83                                              CHANGE.LOC
 289                                                    ** RECURSIVE INVOCATION -- LINE 17
 290    80                                              BLOCK.LOS
 291                                                    ** SEE LINE 194
 292   106                                              MIN.MOVE
 293                                                    ** SEE LINE 266
 294   611                                            OUTPUT.ATTRITION
 295    89                                            FA.BN.MOVEMENT
 296                                                  ** SEE LINE 127
 297   159                                            ATTRIT.SENSOR
 298                                                  ** SEE LINE 139
 299   435                                            ARTY.ASSESS
 300   493                                              SHOOT.OUT
 301                                                    ** RECURSIVE INVOCATION -- LINE 10
 302   611                                              OUTPUT.ATTRITION
 303   463                                          HOW.REPAIR
 304   640                                            EXPONENTIAL.F
 305   604                                            ERROR.STOP
 306                                                  ** SEE LINE 13
 307   467                                          REMOTE.PILOT.VEHICLE
 308   343                                            SEARCH.COVERAGE
 309   229                                            RPV.DETECTION
 310   242                                              UNIT.ENVIR
 311                                                    ** SEE LINE 144
 312   641                                              NORMAL.F
 313   470                                            TARGET.REPORT
 314                                                  ** RECURSIVE INVOCATION -- LINE 34
 315   480                                          FIRE.MISSION
 316                                                ** RECURSIVE INVOCATION -- LINE 68
 317   163                                          BTRY.FM.DEQ
 318   480                                            FIRE.MISSION
 319                                                  ** RECURSIVE INVOCATION -- LINE 68
 320   470                                          TARGET.REPORT
 321                                                ** RECURSIVE INVOCATION -- LINE 34
 322    89                                          FA.BN.MOVEMENT
 323                                                ** SEE LINE 127
 324   467                                      REMOTE.PILOT.VEHICLE
 325                                            ** SEE LINE 307
 326   187                                      FD.EFFECTS.REQ
 327   182                                      FA.BN.ASGN
 328   634                                        FEBA.BAND
 329   100                                          LOCATE.SECTOR
 330   629                                        BTRY.AVAILABLE
 331   632                                        EST.RANGE
 332   274                                        CLEAN.UP.FIRE.MISSIONS
 333   199                                        ILLUM.COMPUTATION
 334   634                                          FEBA.BAND
 335   100                                            LOCATE.SECTOR
 336   233                                        SMOKE.COMPUTATION
 337   634                                          FEBA.BAND
 338   100                                            LOCATE.SECTOR
 339   181                                        FASCAM.COMPUTATION
 340   634                                          FEBA.BAND
 341   100                                            LOCATE.SECTOR
 342   196                                        HE.OR.ICM.COMPUTATION
```

```
 343   634                                                  FEBA.BAND
 344   100                                                  : LOCATE.SECTOR
 345   638                                                  ICM.WLA
 346   636                                                  HE.WLA
 347   604                                                  : ERROR.STOP
 348                                                        : ** SEE LINE 13
 349   248                                                  WEIGHTED.VOLLEYS
 350   176                                                  : EST.COVERAGE
 351   636                                                  : . HE.WLA
 352                                                        : . ** SEE LINE 346
 353   335                                                  : . FRAC.COMPUTE
 354   203                                                  MARGINAL.EFFECTS.ADJ
 355   636                                                  : HE.WLA
 356                                                        : ** SEE LINE 346
 357   480                                                FIRE.MISSION
 358                                                      ** SEE LINE 68
 359   214                                                REM.EFFECTS.COMPUTATION
 360   470                                              TARGET.REPORT
 361                                                    ** RECURSIVE INVOCATION — LINE 34
 362   172                                              COPY
 363    63                                              FILE.FD.SCHD
 364    63                                              : FILE.FD.SCHD
 365                                                    : ** RECURSIVE INVOCATION — LINE 363
 366   218                                          REQUEST.ILLUM
 367                                                : ** SEE LINE 161
 368   222                                          REQUEST.SMOKE
 369                                                : ** SEE LINE 182
 370   487                                        ASSESSMENT
 371                                              : ** SEE LINE 197
 372   642                                        WEIBULL.F
 373   325                                    DEQ.FEBA.SET
 374   100                                    : LOCATE.SECTOR
 375   331                                    ENQ.FEBA.SET
 376   100                                    : LOCATE.SECTOR
 377   119                                    : RESET.FEBA.SECTOR
 378   100                                    : . LOCATE.SECTOR
 379   101                                    LOS.CHECK
 380   604                                    : ERROR.STOP
 381                                          : ** SEE LINE 13
 382   342                                    : RANGE.COMPUTE
 383   132                                    : CHECK.ENGAGEMENT
 384   123                                    : . TIME.TO.DETECT
 385                                          : . ** SEE LINE 169
 386   110                                    POSITION
 387   122                                    : SEGMENT.ADJUST
 388                                          : ** SEE LINE 19
 389   325                                    : DEQ.FEBA.SET
 390   100                                    : . LOCATE.SECTOR
 391   331                                    : ENQ.FEBA.SET
 392                                          : ** SEE LINE 375
 393   322                                    COMPUTE.WD
 394   345                                    TERM.CHECK
 395   604                                    : ERROR.STOP
 396                                          : ** SEE LINE 13
 397   145                                    : FIN.BATTLE
 398   356                                    : . BTL.ENDED
 399   238                                    : . : SWITCH.FO
 400   438                                    : . : . FORWARD.OBSERVER
```

```
401          ** SEE LINE 70
402  337   HC.EMPTY
403  450     HC.RETURN.FARRP
404  298       HC.COMPUTE.TIMES
405  304       REPLACE.HC
406  128       BTL.CHECK
407  298         HC.COMPUTE.TIMES
408  289         EMPLOY.HELICOPTERS
409  298           HC.COMPUTE.TIMES
410  391           SEND.TEAM
411  443             HC.ARRIVE.BATTLE
412  454               HEL.TARGET.ACQUISITION
413  302                 HEL.RANGE.COMPUTE
414  376                 HELO.ENGAGEMENT
415  376                   HELO.ENGAGEMENT
416                        ** RECURSIVE INVOCATION — LINE 414
417  510                   HELICOPTER.FIRE
418  302                     HEL.RANGE.COMPUTE
419  149                     PK.COMPUTE
420  604                       ERROR.STOP
421                            ** SEE LINE 13
422  604                     ERROR.STOP
423                          ** SEE LINE 13
424  493                     SHOOT.OUT
425                          ** RECURSIVE INVOCATION — LINE 10
426  611                     OUTPUT.ATTRITION
427  302                   HEL.RANGE.COMPUTE
428  604                   ERROR.STOP
429                        ** SEE LINE 13
430  493                   SHOOT.OUT
431                        ** RECURSIVE INVOCATION — LINE 10
432  120                 SEARCH
433                      ** SEE LINE 171
434  375                 HC.DEPART.BATTLE
435  300                   HC.DISENGAGE
436  376                     HELO.ENGAGEMENT
437                          ** SEE LINE 414
438  510                     HELICOPTER.FIRE
439                          ** SEE LINE 417
440  487                     ASSESSMENT
441                          ** SEE LINE 197
442  450                   HC.RETURN.FARRP
443                        ** RECURSIVE INVOCATION — LINE 403
444  450               HC.RETURN.FARRP
445                    ** RECURSIVE INVOCATION — LINE 403
446  337               HC.EMPTY
447                    ** RECURSIVE INVOCATION — LINE 402
448  391         SEND.TEAM
449                ** SEE LINE 410
450  391       SEND.TEAM
451          ** SEE LINE 410
452  443       HC.ARRIVE.BATTLE
453          ** SEE LINE 411
454  454       HEL.TARGET.ACQUISITION
455          ** SEE LINE 412
456  300       HC.DISENGAGE
457          ** SEE LINE 435
458  376       HELO.ENGAGEMENT
```

```
 459          .. SEE LINE 414
 460  375     HC.DEPART.BATTLE
 461          .. SEE LINE 434
 462  328   EMPTY
 463  604     ERROR.STOP
 464          .. SEE LINE 13
 465  381     MOVE
 466          .. RECURSIVE INVOCATION -- LINE 16
 467  369     ENGAGEMENT
 468          .. SEE LINE 22
 469  487     ASSESSMENT
 470          .. SEE LINE 197
 471  478     WITH.DRAW
 472  226       REQUEST.WD.FASCAM
 473  604         ERROR.STOP
 474          .     .. SEE LINE 13
 475  342         RANGE.COMPUTE
 476  216         REQUEST.FASCAM
 477          .     .. SEE LINE 31
 478  250         MINE.EFFECTS
 479          .     .. SEE LINE 117
 480  222       REQUEST.SMOKE
 481          .     .. SEE LINE 182
 482  604       ERROR.STOP
 483          .     .. SEE LINE 13
 484  342       RANGE.COMPUTE
 485  642       WEIBULL.F
 486  381       MOVE
 487          .     .. RECURSIVE INVOCATION -- LINE 16
 488  106       MIN.MOVE
 489          .     .. SEE LINE 266
 490  435     ARTY.ASSESS
 491          .. SEE LINE 299
 492  464     MINE.ASSESS
 493          .. SEE LINE 123
 494  493     SHOOT.OUT
 495          .. RECURSIVE INVOCATION -- LINE 10
 496  130   CHECK.DEAD
 497  142     DEAD.UNIT
 498  604       ERROR.STOP
 499          .     .. SEE LINE 13
 500  326       DESTROY.ORD
 501  604         ERROR.STOP
 502          .     .. SEE LINE 13
 503  416       ACT.MOVDIS
 504  403         START.MOVE
 505  336           GET.TERRAIN
 506  325           DEQ.FEBA.SET
 507  100             LOCATE.SECTOR
 508  331           ENQ.FEBA.SET
 509          .     .     .. SEE LINE 375
 510  134           CHECK.FOR.MINES
 511  250           MINE.EFFECTS
 512          .     .     .. SEE LINE 117
 513  138           CHECK.PROX
 514  628             ACT.RANGE
 515  146           INTER.BATTLE
 516  141             CHECK.STREN
```

```
 517  373  GET.NX.ORD
 518  604    ERROR.STOP
 519         ** SEE LINE 13
 520  414    ACT.DEF
 521  412    ACT.ATK
 522  112      PREPARE.LIST
 523  137        CHECK.LIST
 524  408        UPDATE.LOC
 525  325          DEQ.FEBA.SET
 526  100            LOCATE.SECTOR
 527  331          ENQ.FEBA.SET
 528               ** SEE LINE 375
 529  138          CHECK.PROX
 530  628            ACT.RANGE
 531  146          INTER.BATTLE
 532               ** RECURSIVE INVOCATION — LINE 515
 533  117          PROX.POS
 534  141            CHECK.STREN
 535  373            GET.NX.ORD
 536               ** RECURSIVE INVOCATION — LINE 517
 537  250          MINE.EFFECTS
 538               ** SEE LINE 117
 539   87          END.MOVE
 540  325            DEQ.FEBA.SET
 541  100              LOCATE.SECTOR
 542  331            ENQ.FEBA.SET
 543               ** SEE LINE 375
 544  118            REIN.ARRIVE
 545  146              INTER.BATTLE
 546                 ** RECURSIVE INVOCATION — LINE 515
 547  373              GET.NX.ORD
 548                   ** RECURSIVE INVOCATION — LINE 517
 549  604              ERROR.STOP
 550                   ** SEE LINE 13
 551  373            GET.NX.ORD
 552               ** RECURSIVE INVOCATION — LINE 517
 553  408          UPDATE.LOC
 554               ** RECURSIVE INVOCATION — LINE 524
 555  403        START.MOVE
 556             ** RECURSIVE INVOCATION — LINE 504
 557  630        COLLISION
 558  408          UPDATE.LOC
 559               ** SEE LINE 524
 560  604          ERROR.STOP
 561               ** SEE LINE 13
 562  604        ERROR.STOP
 563             ** SEE LINE 13
 564  336      GET.TERRAIN
 565  395      START.BATTLE
 566  604        ERROR.STOP
 567             ** SEE LINE 13
 568   58        CREATE.FORCE
 569  415          ACT.MOVCOR
 570  403            START.MOVE
 571                 ** RECURSIVE INVOCATION — LINE 504
 572  412          ACT.ATK
 573               ** RECURSIVE INVOCATION — LINE 521
 574  714        PLAT.COUNT
```

```
 575   305  UNIT.PRIORITY
 576    67  GENERAL.BATTLE
 577    71  ORIENTATION
 578   644  . ANGLE.COMPUTE
 579    75  UNIT.ASSIGNMENT
 580   604  . ERROR.STOP
 581        . ** SEE LINE 13
 582   644  . ANGLE.COMPUTE
 583    77  ADJUST
 584   325  DEQ.FEBA.SET
 585   100  . LOCATE.SECTOR
 586   331  ENQ.FEBA.SET
 587        ** SEE LINE 375
 588    96  LINE.OF.SIGHT
 589   342  . RANGE.COMPUTE
 590   642  . WEIBULL.F
 591    93  INITIAL.DETECT
 592   123  . TIME.TO.DETECT
 593        . ** SEE LINE 169
 594    94  INITIAL.MOVE
 595   106  . MIN.MOVE
 596        . ** SEE LINE 266
 597   613  . POSITION.OUT
 598   238  SWITCH.FO
 599        ** SEE LINE 399
 600   297  FARRP.CHECK
 601   298  . HC.COMPUTE.TIMES
 602   289  EMPLOY.HELICOPTERS
 603        ** SEE LINE 408
 604   443  HC.ARRIVE.BATTLE
 605        : ** SEE LINE 411
 606   347  ACT.REINF
 607   100  . LOCATE.SECTOR
 608   138  . CHECK.PROX
 609   628  . : ACT.RANGE
 610   141  . CHECK.STREN
 611    95  . INIT.REINF
 612   403  . : START.MOVE
 613        . : ** RECURSIVE INVOCATION — LINE 504
 614   373  . GET.NX.ORD
 615        . ** RECURSIVE INVOCATION — LINE 517
 616   415  ACT.MOVCOR
 617        ** SEE LINE 569
 618   416  ACT.MOVDIS
 619        ** RECURSIVE INVOCATION — LINE 503
 620   604  ERROR.STOP
 621        ** SEE LINE 13
 622   311  INTER.HELO
 623   454  : HEL.TARGET.ACQUISITION
 624        : ** SEE LINE 412
 625   376  : HELO.ENGAGEMENT
 626        : ** SEE LINE 414
 627   510  : HELICOPTER.FIRE
 628        : ** SEE LINE 417
 629   238  SWITCH.FO
 630        ** SEE LINE 399
 631   328  EMPTY
 632        ** SEE LINE 462
```

```
633   130  . : . : . :
```

```
 688    56   .  MAIN1
 689   605   .  :  HEADING
 690   520   .  MAIN2
 691   523   .  :  SYS.INPUT
 692   599   .  :  BETWEEN.ROUTINE
 693   525   .  :  PK.INPUT
 694   527   .  :  CAT.TU.INPUT
 695   529   .  :  KV.INPUT
 696   530   .  :  EQ.TE.INPUT
 697   604   .  :  .  ERROR.STOP
 698         .  :  .  ** SEE LINE 13
 699   532   .  :  TYPE.WEAPON.INPUT
 700   533   .  :  UNIT.INPUT
 701   538   .  :  MFO.INPUT
 702   539   .  :  READ.ORDERS
 703    66   .  :  .  FORM.TF.LIST
 704    66   .  :  .  :  FORM.TF.LIST
 705         .  :  .  :  ** RECURSIVE INVOCATION -- LINE 703
 706   542   .  :  .  ORD.DEF
 707   414   .  :  .  ACT.DEF
 708   543   .  :  .  ORD.ATK
 709   544   .  :  .  ORD.REINF
 710   545   .  :  .  ORD.MOVDIS
 711   546   .  :  .  ORD.MOVCOR
 712   415   .  :  .  :  ACT.MOVCOR
 713         .  :  .  :  ** SEE LINE 569
 714   547   .  :  P.E.M.INPUT
 715   548   .  :  TB.INPUT
 716   550   .  :  BTRY.INPUT
 717   640   .  :  .  EXPONENTIAL.F
 718   553   .  :  FBN.FD.INPUT
 719   555   .  :  MUNS.INPUT
 720   557   .  :  SUBM.INPUT
 721   558   .  :  HE.LA.INPUT
 722   560   .  :  RUL.EN.INPUT
 723   561   .  :  ST.INPUT
 724   562   .  :  MCFR.INPUT
 725   563   .  :  MPDB.INPUT
 726   604   .  :  .  ERROR.STOP
 727         .  :  .  ** SEE LINE 13
 728    61   .  :  FEBA.INITIAL
 729   100   .  :  .  LOCATE.SECTOR
 730   119   .  :  .  RESET.FEBA.SECTOR
 731   100   .  :  .  :  LOCATE.SECTOR
 732   564   .  :  MAO.INPUT
 733   565   .  :  SENSOR.INPUT
 734   438   .  :  .  FORWARD.OBSERVER
 735         .  :  .  ** SEE LINE 70
 736   604   .  :  .  ERROR.STOP
 737         .  :  .  ** SEE LINE 13
 738   371   .  :  .  FEBA.SORTIE
 739   100   .  :  .  :  LOCATE.SECTOR
 740   428   .  :  .  :  AIR.OBSERVER
 741         .  :  .  :  ** SEE LINE 140
 742   371   .  :  .  :  FEBA.SORTIE
 743         .  :  .  :  ** RECURSIVE INVOCATION -- LINE 738
 744   569   .  :  TBF.INPUT
 745    60   .  :  .  CREATE.TEAMS
```

```
 746   570   .  :  DECISION.INPUT
 747   571   .  :  TT.FACTORS.INPUT
 748   572   .  :  FARRP.INPUT
 749   575   .  :  PGM.INPUT
 750   576   .  :  ILLUM.INPUT
 751   577   .  :  MINE.INPUT
 752   580   .  :  SMOKE.INPUT
 753   581   .  :  VIS.INPUT
 754   366   .  :  .  CHANGE.WEATHER
 755   327   .  :  .  :  DQ.CMSN.QUEUE
 756         .  :  .  :  ** SEE LINE 211
 757   582   .  :  TACAIR.INPUT
 758   380   .  :  .  INIT.PREPLAN.CAS
 759   285   .  :  .  :  CHECK.CAS.CONSTRAINTS
 760         .  :  .  :  ** SEE LINE 212
 761   587   .  :  MADS.INPUT
 762   588   .  :  AC.MUNS.INPUT
 763   621   .  :  TACAIR.DATA.REPORT
 764   595   .  :  ANALYSIS.OUTPUT
 765    57   .  MAIN3
 766   390   .  :  SCHEDULE.ARTY.MOVEMENT
 767   390   .  :  .  SCHEDULE.ARTY.MOVEMENT
 768         .  :  .  ** RECURSIVE INVOCATION -- LINE 766
 769    89   .  :  .  FA.BN.MOVEMENT
 770         .  :  .  ** SEE LINE 127
 771   368   .  :  END.SIMULATION
 772   609   .  :  .  KV.SCOREBOARD
 773   606   .  :  .  :  KV.PRINT
 774   592   .  :  .  :  AMMO.RPT
 775   613   .  :  .  POSITION.OUT
 776   624   .  :  .  OUTPUT.EXPENDITURES
 777   365   .  :  CHANGE.LITE
 778   365   .  :  .  CHANGE.LITE
 779         .  :  .  ** RECURSIVE INVOCATION -- LINE 777
 780   327   .  :  .  DQ.CMSN.QUEUE
 781         .  :  .  ** SEE LINE 211
 782   389   .  :  POSITION.REPORT
 783   613   .  :  .  POSITION.OUT
 784   389   .  :  .  POSITION.REPORT
 785         .  :  .  ** RECURSIVE INVOCATION -- LINE 782

 786   652   GAMMA.F

 787   653   AIRBORNE.RADAR
 788   656   .  AR.DETECTION
 789   342   .  :  RANGE.COMPUTE
 790   657   .  :  AR.PROB.DETECT
 791   641   .  :  NORMAL.F

 792   658   PHOTO.IR.FLIGHT
 793   660   .  STAY.TIME
 794     *   .  :  LOCATE.SECTORPG
 795   628   .  :  ACT.RANGE
 796   343   .  SEARCH.COVERAGE
 797   212   .  PIR.DETECTION
 798   242   .  :  UNIT.ENVIR
 799         .  :  ** SEE LINE 144
 800   641   .  NORMAL.F
```

```
 801   470   .  TARGET.REPORT
 802         .  ** SEE LINE 34

 803   664  OLDER.VERSION
 804   392   .  SET.DEBUG
 805   382   .  OFF.LINE.ATTRITION
 806         .  ** SEE LINE 9
```

```
ACT.ATK
    PAGE  39  ''SECTION FOR EVENTS                    2204
    PAGE  59  ROUTINE CREATE.FORCE                    3150 3151
    PAGE 373  EVENT GET.NX.ORD                        6926 6932
    PAGE 412  EVENT ACT.ATK                           8599
    PAGE 600  ROUTINE BETWEEN.ROUTINE                 7532 7533
    PAGE 616  ROUTINE SNAP2                           8163
    PAGE 698  ''PROGRAM  OLDER.VERSION                1641
ACT.DEF
    PAGE  39  ''SECTION FOR EVENTS                    2209
    PAGE 373  EVENT GET.NX.ORD                        6912
    PAGE 414  EVENT ACT.DEF                           8714
    PAGE 540  ROUTINE READ.ORDERS                     5059
    PAGE 600  ROUTINE BETWEEN.ROUTINE                 7536 7537
    PAGE 616  ROUTINE SNAP2                           8164
    PAGE 698  ''PROGRAM  OLDER.VERSION                1646
ACT.MOVCOR
    PAGE  39  ''SECTION FOR EVENTS                    2213
    PAGE  58  ROUTINE CREATE.FORCE                    3144 3145
    PAGE 374  EVENT GET.NX.ORD                        6948 6953
    PAGE 415  EVENT ACT.MOVCOR                        8736
    PAGE 546  ROUTINE ORD.MOVCOR                      5246
    PAGE 600  ROUTINE BETWEEN.ROUTINE                 7540 7541
    PAGE 616  ROUTINE SNAP2                           8165
    PAGE 698  ''PROGRAM  OLDER.VERSION                1650
ACT.MOVDIS
    PAGE  39  ''SECTION FOR EVENTS                    2217
    PAGE 144  ROUTINE DEAD.UNIT                       6686
    PAGE 374  EVENT GET.NX.ORD                        6964
    PAGE 416  EVENT ACT.MOVDIS                        8766
    PAGE 601  ROUTINE BETWEEN.ROUTINE                 7544 7545
    PAGE 616  ROUTINE SNAP2                           8166
    PAGE 698  ''PROGRAM  OLDER.VERSION                1654
ACT.RANGE
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES        1067
    PAGE  50  ''SECTION FOR DEFINITIONS               2848
    PAGE 138  ROUTINE CHECK.PROX                      6425 6426
    PAGE 139                                          6442 6443
    PAGE 205  ROUTINE NOISE.DEGRADE                   9398 9399 9410 9411
    PAGE 246  ROUTINE VOLLEY                          1341 1342
    PAGE 358  EVENT CFR.ACTIVATION                    6367 6368
    PAGE 360  EVENT CFR.ON                            6452 6453
    PAGE 386  EVENT PDB.ACTIVATION                    7493 7494
    PAGE 439  PROCESS FORWARD.OBSERVER                9921 9923 9939 9941
    PAGE 628  FUNCTION ACT.RANGE                      8630
    PAGE 660  FUNCTION STAY.TIME                      9605 9606
    PAGE 678  ''PROGRAM  OLDER.VERSION                 509
    PAGE 709                                          2267
ACT.REINF
    PAGE  39  ''SECTION FOR EVENTS                    2221
    PAGE 347  EVENT ACT.REINF                         5803
    PAGE 374  EVENT GET.NX.ORD                        6941
    PAGE 601  ROUTINE BETWEEN.POUTINE                 7548 7549
    PAGE 616  ROUTINE SNAP2                           8167
    PAGE 698  ''PROGRAM  OLDER.VERSION                1658
```

```
AC.ATK.TGT
    PAGE  33  ''SECTION FOR PROCESSES              1892
    PAGE 307  ROUTINE AD.SHOOT                     4325
    PAGE 419  PROCESS AC.ATK.TGT                   8847 8883 8884
    PAGE 423                                       9084 9116
    PAGE 424                                       9192
    PAGE 425                                       9223
    PAGE 426                                       9267 9285
    PAGE 427                                       9313 9317 9318
    PAGE 490  PROCESS ASSESSMENT                   2611 2631
    PAGE 507  PROCESS CAS.MISSION                  3557 3576
    PAGE 616  ROUTINE SNAP2                        8141
    PAGE 692  ''PROGRAM  OLDER.VERSION             1330
AC.BOMB.EFFECTS
    PAGE 276  ROUTINE AC.BOMB.EFFECTS              2787 2796
    PAGE 424  PROCESS AC.ATK.TGT                   9141
AC.DF.EFFECTS
    PAGE 280  ROUTINE AC.DF.EFFECTS                2996 3007
    PAGE 423  PROCESS AC.ATK.TGT                   9134
AC.MUNS.INPUT
    PAGE 522  ROUTINE MAIN2                        4303 4305
    PAGE 588  ROUTINE AC.MUNS.INPUT                7011
ADJUST
    PAGE  77  ROUTINE ADJUST                       3879
    PAGE 104  ROUTINE MINE.DELAY                   5027
    PAGE 185  ROUTINE FA.BN.ASGN                   8509
    PAGE 399  EVENT START.BATTLE                   8006 8014 8021 8029
    PAGE 400                                       8040
    PAGE 447  PROCESS HC.ARRIVE.BATTLE              355
    PAGE 485  PROCESS FIRE.MISSION                 2358
    PAGE 637  FUNCTION HE.WLA                      8927
AD.ENGAGEMENT
    PAGE  39  ''SECTION FOR EVENTS                 2226
    PAGE 293  ROUTINE END.CAS.MISSION              3700 3701 3702
    PAGE 307  ROUTINE AD.SHOOT                     4321
    PAGE 349  EVENT AD.ENGAGEMENT                  5897 5911
    PAGE 505  PROCESS CAS.MISSION                  3439
    PAGE 508                                       3631
    PAGE 601  ROUTINE BETWEEN.ROUTINE              7552 7553
    PAGE 616  ROUTINE SNAP2                        8168
    PAGE 698  ''PROGRAM  OLDER.VERSION             1663
AD.SHOOT
    PAGE 307  ROUTINE AD.SHOOT                     4297 4317
    PAGE 354  EVENT AD.ENGAGEMENT                  6189
    PAGE 422  PROCESS AC.ATK.TGT                   9049
    PAGE 424                                       9167
AIRBORNE.RADAR
    PAGE  34  ''SECTION FOR PROCESSES              1906
    PAGE 599  ROUTINE BETWEEN.ROUTINE              7452 7453
    PAGE 616  ROUTINE SNAP2                        8142
    PAGE 653  PROCESS AIRBORNE.RADAR               9261
    PAGE 693  ''PROGRAM  OLDER.VERSION             1344
AIR.OBSERVER
    PAGE  34  ''SECTION FOR PROCESSES              1922
    PAGE 159  ROUTINE ATTRIT.SENSOR                7356 7367
    PAGE 309  ROUTINE AD.SHOOT                     4457 4458
```

```
    PAGE 371  EVENT FEBA.SORTIE                    6805
    PAGE 372                                       6848
    PAGE 428  PROCESS AIR.OBSERVER                 9330
    PAGE 432                                       9590
    PAGE 599  ROUTINE BETWEEN.ROUTINE              7456 7457
    PAGE 616  ROUTINE SNAP2                        8143
    PAGE 693  ''PROGRAM  OLDER.VERSION             1360
AMMO.RPT
    PAGE   4  PROGRAM REVISIONS                     189  189
    PAGE 592  ROUTINE AMMO.RPT                     7119
    PAGE 609  ROUTINE KV.SCOREBOARD                7856
ANALYSIS.OUTPUT
    PAGE 522  ROUTINE MAIN2                        4312
    PAGE 595  ROUTINE ANALYSIS.OUTPUT              7239
ANGLE.COMPUTE
    PAGE  72  ROUTINE ORIENTATION                  3739 3745
    PAGE  73                                       3784 3801
    PAGE  74                                       3811
    PAGE  75  ROUTINE UNIT.ASSIGNMENT              3857
    PAGE 111  ROUTINE PRED.POS                     5335
    PAGE 644  ROUTINE ANGLE.COMPUTE                9078
AO.DETECTION
    PAGE 154  ROUTINE AO.DETECTION                 7074 7115
    PAGE 242  ROUTINE UNIT.ENVIR                   1127
    PAGE 431  PROCESS AIR.OBSERVER                 9555
ARTY.ASSESS
    PAGE  34  ''SECTION FOR PROCESSES              1943
    PAGE 261  ROUTINE BTRY.EFFECTS                 2079
    PAGE 272                                       2700
    PAGE 328  ROUTINE EMPTY                        5217 5218 5220 5221 5222
    PAGE 435  PROCESS ARTY.ASSESS                  9686 9690 9694 9696 9700 9703 9710
    PAGE 599  ROUTINE BETWEEN.ROUTINE              7460 7461
    PAGE 616  ROUTINE SNAP2                        8144
    PAGE 693  ''PROGRAM  OLDER.VERSION             1381
ARTY.OCCUPATION
    PAGE  39  ''SECTION FOR EVENTS                 2230
    PAGE 355  EVENT ARTY.OCCUPATION                6227 6234
    PAGE 407  EVENT STOP.ARTY.MOVEMENT             8397
    PAGE 601  ROUTINE BETWEEN.ROUTINE              7556 7557
    PAGE 616  ROUTINE SNAP2                        8169
    PAGE 698  ''PROGRAM  OLDER.VERSION             1667
AR.DETECTION
    PAGE 654  PROCESS AIRBORNE.RADAR               9354
    PAGE 656  ROUTINE AR.DETECTION                 9385 9393
AR.PROB.DETECT
    PAGE  50  ''SECTION FOR DEFINITIONS            2849
    PAGE 656  ROUTINE AR.DETECTION                 9405 9406
    PAGE 657  FUNCTION AR.PROB.DETECT              9443
    PAGE 709  ''PROGRAM  OLDER.VERSION             2268
ASSESSMENT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING          50
    PAGE   3  PROGRAM REVISIONS                     168
    PAGE  34  ''SECTION FOR PROCESSES              1954
    PAGE  81  ROUTINE BLOCK.LOS                    4087 4089
    PAGE 108  ROUTINE NEW.SEGMENT                  5172 5173 5174 5175 5176 5186 5187 5188 5189 5190
    PAGE 250  ROUTINE MINE.EFFECTS                 1472
```

```
    PAGE 294  ROUTINE END.CAS.MISSION              3748 3749 3751 3752 3753
    PAGE 301  ROUTINE HC.DISENGAGE                 4098 4100 4102 4103 4104 4105
    PAGE 328  ROUTINE EMPTY                        5197 5198 5199 5200 5202 5203 5204 5205
    PAGE 425  PROCESS AC.ATK.TGT                   9250 9252
    PAGE 426                                       9301 9303
    PAGE 487  PROCESS ASSESSMENT                   2411
    PAGE 488                                       2518 2519 2521 2522 2523
    PAGE 490                                       2640 2642
    PAGE 502  PROCESS SHOOT.OUT                    3273
    PAGE 599  ROUTINE BETWEEN.ROUTINE              7464 7465
    PAGE 616  ROUTINE SNAP2                        8145
    PAGE 693  ''PROGRAM  OLDER.VERSION             1392
ATTRIT.SENSOR
    PAGE   3  PROGRAM REVISIONS                     175
    PAGE 159  ROUTINE ATTRIT.SENSOR                7311 7344
    PAGE 253  ROUTINE MINE.EFFECTS                 1655
    PAGE 271  ROUTINE BTRY.EFFECTS                 2629
    PAGE 279  ROUTINE AC.BOMB.EFFECTS              2974
    PAGE 384  EVENT OFF.LINE.ATTRITION             7433
BETWEEN.ROUTINE
    PAGE 520  ROUTINE MAIN2                        4191
    PAGE 599  ROUTINE BETWEEN.ROUTINE              7428
BLOCK.LOS
    PAGE  80  ROUTINE BLOCK.LOS                    4005
    PAGE 174  ROUTINE DUST.EFFECTS                 7955
    PAGE 235  ROUTINE SMOKE.EFFECTS                 843
    PAGE 236                                        871
BTL.CHECK
    PAGE 128  ROUTINE BTL.CHECK                    5984 6002
    PAGE 129                                       6061
    PAGE 452  PROCESS HC.RETURN.FARRP               631
BTL.ENDED
    PAGE   2  PROGRAM REVISIONS                      87
    PAGE  39  ''SECTION FOR EVENTS                 2233
    PAGE 130  ROUTINE CHECK.DEAD                   6074
    PAGE 145  ROUTINE FIN.BATTLE                   6736
    PAGE 356  EVENT BTL.ENDED                      6253 6280 6292
    PAGE 601  ROUTINE BETWEEN.ROUTINE              7560 7561
    PAGE 616  ROUTINE SNAP2                        8170
    PAGE 698  ''PROGRAM  OLDER.VERSION             1670
BTRY.AVAILABLE
    PAGE  50  ''SECTION FOR DEFINITIONS            2850
    PAGE 183  ROUTINE FA.BN.ASGN                   8402
    PAGE 184                                       8403
    PAGE 629  FUNCTION BTRY.AVAILABLE              8645 8675
    PAGE 709  ''PROGRAM  OLDER.VERSION             2269
BTRY.EFFECTS
    PAGE   2  PROGRAM REVISIONS                     109
    PAGE  83  ROUTINE CHANGE.LOC                   4129
    PAGE 242  ROUTINE UNIT.ENVIR                   1127
    PAGE 260  ROUTINE BTRY.EFFECTS                 1964
    PAGE 484  PROCESS FIRE.MISSION                 2308
BTRY.FM.DEQ
    PAGE 163  ROUTINE BTRY.FM.DEQ                  7513 7526
    PAGE 486  PROCESS FIRE.MISSION                 2388
```

```
BTRY.FM.ENQ
    PAGE 164  ROUTINE BTRY.FM.ENQ                    7543 7552 7570
    PAGE 481  PROCESS FIRE.MISSION                   2118
BTRY.INPUT
    PAGE 520  ROUTINE MAIN2                          4230 4232
    PAGE 550  ROUTINE BTRY.INPUT                     5387
CAS.EVAL
    PAGE 283  ROUTINE CAS.EVAL                       3167
    PAGE 492  PROCESS ASSESSMENT                     2742
CAS.MISSION
    PAGE  35  ''SECTION FOR PROCESSES                1962
    PAGE 284  ROUTINE CAS.EVAL                       3259
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS          3398
    PAGE 288                                         3460 3467
    PAGE 309  ROUTINE AD.SHOOT                       4463 4464
    PAGE 349  EVENT AD.ENGAGEMENT                    5915
    PAGE 380  EVENT INIT.PREPLAN.CAS                 7238
    PAGE 421  PROCESS AC.ATK.TGT                     8975
    PAGE 426                                         9286
    PAGE 427                                         9323 9324
    PAGE 504  PROCESS CAS.MISSION                    3350 3368 3369
    PAGE 505                                         3434
    PAGE 506                                         3476 3477
    PAGE 508                                         3602 3603
    PAGE 509                                         3645 3646
    PAGE 599  ROUTINE BETWEEN.ROUTINE                7468 7469
    PAGE 616  ROUTINE SNAP2                          8146
    PAGE 618  ROUTINE SNAP.R                         8251
    PAGE 694  ''PROGRAM  OLDER.VERSION               1400
CAT.TU.INPUT
    PAGE 520  ROUTINE MAIN2                          4199 4201
    PAGE 527  ROUTINE CAT.TU.INPUT                   4450 4456
CFR.ACTIVATION
    PAGE  39  ''SECTION FOR EVENTS                   2238
    PAGE 246  ROUTINE VOLLEY                         1359
    PAGE 358  EVENT CFR.ACTIVATION                   6341
    PAGE 601  ROUTINE BETWEEN.ROUTINE                7564 7565
    PAGE 616  ROUTINE SNAP2                          8171
    PAGE 698  ''PROGRAM  OLDER.VERSION               1675
CFR.DEGRADE
    PAGE 165  ROUTINE CFR.DEGRADE                    7573
    PAGE 167  ROUTINE CFR.DETECTION                  7683
CFR.DETECTION
    PAGE 166  ROUTINE CFR.DETECTION                  7617
    PAGE 358  EVENT CFR.ACTIVATION                   6373
CFR.OFF
    PAGE  39  ''SECTION FOR EVENTS                   2241
    PAGE 161  ROUTINE ATTRIT.SENSOR                  7462 7463
    PAGE 359  EVENT CFR.OFF                          6380
    PAGE 360  EVENT CFR.ON                           6422
    PAGE 601  ROUTINE BETWEEN.ROUTINE                7568 7569
    PAGE 616  ROUTINE SNAP2                          8172
    PAGE 698  ''PROGRAM  OLDER.VERSION               1678
CFR.ON
    PAGE   4  PROGRAM REVISIONS                       197
    PAGE  39  ''SECTION FOR EVENTS                   2245
```

```
    PAGE 161  ROUTINE ATTRIT.SENSOR                7454 7455
    PAGE 206  ROUTINE PDB.DETECTION                9478 9479 9482
    PAGE 359  EVENT CFR.OFF                        6395
    PAGE 360  EVENT CFR.ON                         6405
    PAGE 601  ROUTINE BETWEEN.ROUTINE              7572 7573
    PAGE 616  ROUTINE SNAP2                        8173
    PAGE 698  ''PROGRAM  OLDER.VERSION             1682
CFR.OPERATOR
    PAGE  39  ''SECTION FOR EVENTS                 2249
    PAGE 161  ROUTINE ATTRIT.SENSOR                7469 7470
    PAGE 167  ROUTINE CFR.DETECTION                7695
    PAGE 362  EVENT CFR.OPERATOR                   6477 6494 6514 6529
    PAGE 363                                       6552 6586
    PAGE 601  ROUTINE BETWEEN.ROUTINE              7576 7577
    PAGE 616  ROUTINE SNAP2                        8174
    PAGE 699  ''PROGRAM  OLDER.VERSION             1686
CHANGE.LITE
    PAGE  39  ''SECTION FOR EVENTS                 2202
    PAGE  57  ROUTINE MAIN3                        3079
    PAGE 365  EVENT CHANGE.LITE                    6595 6603 6609
    PAGE 600  ROUTINE BETWEEN.ROUTINE              7524 7525
    PAGE 616  ROUTINE SNAP2                        8161
    PAGE 698  ''PROGRAM  OLDER.VERSION             1639
CHANGE.LOC
    PAGE  83  ROUTINE CHANGE.LOC                   4124
    PAGE 106  ROUTINE MIN.MOVE                     5065
    PAGE 173  ROUTINE DUST.EFFECTS                 7917
    PAGE 235  ROUTINE SMOKE.EFFECTS                 805
    PAGE 261  ROUTINE BTRY.EFFECTS                 2075
    PAGE 381  EVENT MOVE                           7274
CHANGE.WEATHER
    PAGE  40  ''SECTION FOR EVENTS                 2253
    PAGE 366  EVENT CHANGE.WEATHER                 6616
    PAGE 581  ROUTINE VIS.INPUT                    6726
    PAGE 601  ROUTINE BETWEEN.ROUTINE              7580 7581
    PAGE 699  ''PROGRAM  OLDER.VERSION             1690
CHECK.CAS.CONSTRAINTS
    PAGE 284  ROUTINE CAS.EVAL                     3272
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS        3280 3290
    PAGE 327  ROUTINE DQ.CMSN.QUEUE                5159
    PAGE 380  EVENT INIT.PREPLAN.CAS               7261
CHECK.DEAD
    PAGE 130  ROUTINE CHECK.DEAD                   6067 6107
    PAGE 147  ROUTINE INTER.BATTLE                 6841 6846
    PAGE 148                                       6878 6883
    PAGE 356  EVENT BTL.ENDED                      6257
    PAGE 357                                       6326 6328
CHECK.ENGAGEMENT
    PAGE 101  ROUTINE LOS.CHECK                    4923 4927
    PAGE 132  ROUTINE CHECK.ENGAGEMENT             6131
CHECK.FORCE
    PAGE 116  ROUTINE PROX.CHECK                   5556
    PAGE 133  ROUTINE CHECK.FORCE                  6153
    PAGE 324  ROUTINE DECIDE                       5072
CHECK.FOR.MINES
    PAGE 134  ROUTINE CHECK.FOR.MINES              6211
```

```
    PAGE 393  EVENT START.ARTY.MOVEMENT                7676
    PAGE 404  EVENT START.MOVE                         8260
CHECK.LIST
    PAGE 112  ROUTINE PREPARE.LIST                     5378
    PAGE 137  ROUTINE CHECK.LIST                       6332 6361
CHECK.PROX
    PAGE 138  ROUTINE CHECK.PROX                       6373
    PAGE 347  EVENT ACT.REINF                          5845
    PAGE 405  EVENT START.MOVE                         8327
    PAGE 409  EVENT UPDATE.LOC                         8489
CHECK.STREN
    PAGE 117  ROUTINE PROX.POS                         5577
    PAGE 141  ROUTINE CHECK.STREN                      6503
    PAGE 146  ROUTINE INTER.BATTLE                     6758
    PAGE 347  EVENT ACT.REINF                          5859
CHK.COMP.TR
    PAGE 168  ROUTINE CHK.COMP.TR                      7699 7706
    PAGE 363  EVENT CFR.OPERATOR                       6561
    PAGE 387  EVENT PDB.OPERATOR                       7549
    PAGE 471  PROCESS TARGET.REPORT                    1587
CHK.FD.TR
    PAGE 169  ROUTINE CHK.FD.TR                        7722
    PAGE 363  EVENT CFR.OPERATOR                       6569
    PAGE 387  EVENT PDB.OPERATOR                       7552
    PAGE 472  PROCESS TARGET.REPORT                    1611
CLEAN.UP.FIRE.MISSIONS
    PAGE   3  PROGRAM REVISIONS                         143
    PAGE 184  ROUTINE FA.BN.ASGN                       8436 8457
    PAGE 185                                           8489
    PAGE 210  ROUTINE PGM.MSN.ASGN                     9658
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS           2733
COLLISION
    PAGE   1  ROUTINE FOR CROSS_REFERENCING             55
    PAGE  50  ''SECTION FOR DEFINITIONS                2851
    PAGE 113  ROUTINE PREPARE.LIST                     5412 5414
    PAGE 630  FUNCTION COLLISION                       8679
    PAGE 709  ''PROGRAM  OLDER.VERSION                 2270
COMBINATIONS
    PAGE   1  ROUTINE FOR CROSS_REFERENCING             55
    PAGE  50  ''SECTION FOR DEFINITIONS                2852
    PAGE 231  ROUTINE SIZE.ESTIMATE                     665  666
    PAGE 631  FUNCTION COMBINATIONS                    8714
    PAGE 709  ''PROGRAM  OLDER.VERSION                 2271
COMBINE.TRS
    PAGE   4  PROGRAM REVISIONS                         207
    PAGE 169  ROUTINE CHK.FD.TR                        7752
    PAGE 170  ROUTINE COMBINE.TRS                      7764
COMPARE.TRS
    PAGE 168  ROUTINE CHK.COMP.TR                      7711
    PAGE 169  ROUTINE CHK.FD.TR                        7742
    PAGE 171  ROUTINE COMPARE.TRS                      7797
COMPUTE.D
    PAGE  83  ROUTINE CHANGE.LOC                       4160 4180
    PAGE 321  ROUTINE COMPUTE.D                        4942
COMPUTE.WD
    PAGE  84  ROUTINE CHANGE.LOC                       4224
```

```
    PAGE  85                                    4244
    PAGE 322  ROUTINE COMPUTE.WD                4964
CONTRAST.TO.FREQ
    PAGE 120  ROUTINE SEARCH                    5728
    PAGE 323  ROUTINE CONTRAST.TO.FREQ          4998
COPY
    PAGE 172  ROUTINE COPY                      7843 7849
    PAGE 476  PROCESS TARGET.REPORT             1835 1860
CREATE.FORCE
    PAGE  58  ROUTINE CREATE.FORCE              3089
    PAGE 396  EVENT START.BATTLE                7806 7812
CREATE.TEAMS
    PAGE  60  ROUTINE CREATE.TEAMS              3157
    PAGE 569  ROUTINE TBF.INPUT                 6241 6243 6245 6247 6249 6251 6253 6255
CROSS_REFERENCING
    PAGE   1  ROUTINE FOR CROSS_REFERENCING       37
DEAD.UNIT
    PAGE 130  ROUTINE CHECK.DEAD                6073 6091 6104 6107
    PAGE 142  ROUTINE DEAD.UNIT                 6521
DECIDE
    PAGE 324  ROUTINE DECIDE                    5034
    PAGE 384  EVENT OFF.LINE.ATTRITION          7403
    PAGE 493  PROCESS SHOOT.OUT                 2797
    PAGE 496                                    2964
    PAGE 500                                    3169
    PAGE 502                                    3324
DECISION.INPUT
    PAGE 521  ROUTINE MAIN2                     4269 4271
    PAGE 570  ROUTINE DECISION.INPUT            6260
DEFINITIONS
    PAGE  41  ''SECTION FOR DEFINITIONS         2350
    PAGE  51  ''SECTION FOR SUBSTITUTIONS       2904
    PAGE  52                                    2952
    PAGE 700  ''PROGRAM  OLDER.VERSION          1781
    PAGE 709                                    2312
    PAGE 710                                    2359
DEQ.FEBA.SET
    PAGE  84  ROUTINE CHANGE.LOC                4187
    PAGE  85                                    4251
    PAGE  87  ROUTINE END.MOVE                  4323 4342
    PAGE 110  ROUTINE POSITION                  5285
    PAGE 325  ROUTINE DEQ.FEBA.SET              5078 5087
    PAGE 400  EVENT START.BATTLE                8055
    PAGE 403  EVENT START.MOVE                  8235
    PAGE 404                                    8248
    PAGE 409  EVENT UPDATE.LOC                  8463 8476
DESTROY.ORD
    PAGE 143  ROUTINE DEAD.UNIT                 6615
    PAGE 326  ROUTINE DESTROY.ORD               5118
DQ.CMSN.QUEUE
    PAGE 327  ROUTINE DQ.CMSN.QUEUE             5143 5155
    PAGE 365  EVENT CHANGE.LITE                 6612
    PAGE 366  EVENT CHANGE.WEATHER              6625
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE         6644 6657
    PAGE 508  PROCESS CAS.MISSION               3612
```

```
DQ.OLD.SORTIE.QUEUE
     PAGE  40  ''SECTION FOR EVENTS                    2256
     PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE               6634 6652
     PAGE 504  PROCESS CAS.MISSION                     3390
     PAGE 601  ROUTINE BETWEEN.ROUTINE                 7584 7585
     PAGE 616  ROUTINE SNAP2                           8175
     PAGE 699  ''PROGRAM  OLDER.VERSION                1693
DUST.EFFECTS
     PAGE   3  PROGRAM REVISIONS                        161
     PAGE 173  ROUTINE DUST.EFFECTS                    7882
     PAGE 193  ROUTINE FINAL.COVERAGE                  8881
DYNAMIC.ANALYSIS.REPORT
     PAGE  41  ''SECTION FOR EVENTS                    2338
     PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT           8803 8833
     PAGE 648  ROUTINE PERFORM.INSTRUMENTATION         9207
EMPLOY.HELICOPTERS
     PAGE 129  ROUTINE BTL.CHECK                       6054
     PAGE 289  ROUTINE EMPLOY.HELICOPTERS              3473 3519
     PAGE 401  EVENT START.BATTLE                      8110 8134
EMPTY
     PAGE 147  ROUTINE INTER.BATTLE                    6816
     PAGE 163  ROUTINE BTRY.FM.DEQ                     7529
     PAGE 184  ROUTINE FA.BN.ASGN                      8449
     PAGE 239  ROUTINE TARGET.ANALYSIS                 1009 1016
     PAGE 246  ROUTINE VOLLEY                          1327 1332
     PAGE 252  ROUTINE MINE.EFFECTS                    1613
     PAGE 256  ROUTINE FO.DETECTION                    1827 1830
     PAGE 278  ROUTINE AC BOMB.EFFECTS                 2927
     PAGE 282  ROUTINE AC DF.EFFECTS                   3113 3140
     PAGE 293  ROUTINE END.CAS.MISSION                 3688
     PAGE 300  ROUTINE HC.DISENGAGE                    4025 4045
     PAGE 304  ROUTINE REPLACE HC                      4216
     PAGE 311  ROUTINE INTER.HELO                      4564
     PAGE 312                                          4585
     PAGE 326  ROUTINE DESTROY.ORD                     5122
     PAGE 328  ROUTINE EMPTY                           5165 5172
     PAGE 329                                          5241 5256 5258 5269 5275
     PAGE 331  ROUTINE ENQ.FEBA.SET                    5337
     PAGE 332  ROUTINE FDC.TR.DEQ                      5350
     PAGE 356  EVENT BTL.ENDED                         6268 6296
     PAGE 360  EVENT CFR.ON                            6436 6443
     PAGE 363  EVENT CFR.OPERATOR                      6584
     PAGE 388  EVENT PDB.OPERATOR                      7562
     PAGE 409  EVENT UPDATE.LOC                        8499
     PAGE 410                                          8531
     PAGE 429  PROCESS AIR.OBSERVER                    9389 9399
     PAGE 431                                          9520 9525 9538 9544 9557
     PAGE 432                                          9569 9575 9603 9608
     PAGE 433                                          9621 9626 9639 9644 9658 9663 9673
     PAGE 434                                          9678
     PAGE 435  PROCESS ARTY.ASSESS                     9703
     PAGE 439  PROCESS FORWARD.OBSERVER                9957
     PAGE 440                                            22
     PAGE 464  PROCESS MINE.ASSESS                     1245
     PAGE 468  PROCESS REMOTE.PILOT.VEHICLE            1449 1470
     PAGE 469                                          1475
```

```
    PAGE 473  PROCESS TARGET.REPORT                    1701
    PAGE 477                                           1896
    PAGE 485  PROCESS FIRE.MISSION                     2364
    PAGE 486                                           2392
    PAGE 491  PROCESS ASSESSMENT                       2686
    PAGE 494  PROCESS SHOOT.OUT                        2807
    PAGE 495                                           2877
    PAGE 498                                           3070
    PAGE 501                                           3214 3234
    PAGE 511  PROCESS HELICOPTER.FIRE                  3722 3737
    PAGE 514                                           3902
    PAGE 516                                           4035
    PAGE 539  ROUTINE READ.ORDERS                      5018 5033
    PAGE 630  FUNCTION COLLISION                       8705
    PAGE 634  FUNCTION FEBA.BAND                       8797 8815 8827
    PAGE 654  PROCESS AIRBORNE.RADAR                   9357
    PAGE 655                                           9380
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                  9506
    PAGE 659                                           9542 9560
END.CAS.MISSION
    PAGE 293  ROUTINE END.CAS.MISSION                  3676
    PAGE 504  PROCESS CAS.MISSION                      3405
    PAGE 505                                           3461
    PAGE 506                                           3481
    PAGE 508                                           3615
    PAGE 509                                           3648
END.MOVE
    PAGE  87  ROUTINE END.MOVE                         4307
    PAGE 411  EVENT UPDATE.LOC                         8580
END.SIMULATION
    PAGE  39  ''SECTION FOR EVENTS                     2202
    PAGE  57  ROUTINE MAIN3                            3078
    PAGE 368  EVENT END.SIMULATION                     6661
    PAGE 592  ROUTINE AMMO.RPT                         7122
    PAGE 600  ROUTINE BETWEEN.ROUTINE                  7528 7529
    PAGE 616  ROUTINE SNAP2                            8162
    PAGE 698  ''PROGRAM  OLDER.VERSION                 1639
ENGAGEMENT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING              53
    PAGE  40  ''SECTION FOR EVENTS                     2259
    PAGE  81  ROUTINE BLOCK.LOS                        4076 4077
    PAGE 107  ROUTINE NEW.SEGMENT                      5160
    PAGE 108                                           5161 5168 5169
    PAGE 124  ROUTINE TIME.TO.DETECT                   5852 5853 5854 5857 5858 5864
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM               9833
    PAGE 218  ROUTINE REQUEST.ILLUM                    9987
    PAGE 222  ROUTINE REQUEST.SMOKE                     178
    PAGE 328  ROUTINE EMPTY                            5190 5191 5193 5194
    PAGE 369  EVENT ENGAGEMENT                         6678 6693 6699 6700 6702 6703
    PAGE 376  EVENT HELO.ENGAGEMENT                    7054
    PAGE 487  PROCESS ASSESSMENT                       2440 2446
    PAGE 601  ROUTINE BETWEEN.ROUTINE                  7588 7589
    PAGE 616  ROUTINE SNAP2                            8177
    PAGE 699  ''PROGRAM  OLDER.VERSION                 1696
ENQ.FEBA.SET
    PAGE  84  ROUTINE CHANGE.LOC                       4192
```

```
    PAGE  85                                4256
    PAGE  87  ROUTINE END.MOVE              4333 4348
    PAGE 110  ROUTINE POSITION              5290
    PAGE 331  ROUTINE ENQ.FEBA.SET          5301 5309
    PAGE 400  EVENT START.BATTLE            8060
    PAGE 404  EVENT START.MOVE              8240 8253
    PAGE 409  EVENT UPDATE.LOC              8468 8481
EQ.TE.INPUT
    PAGE 520  ROUTINE MAIN2                 4205 4207
    PAGE 530  ROUTINE EQ.TE.INPUT           4573
ERROR.STOP
    PAGE  75  ROUTINE UNIT.ASSIGNMENT       3849
    PAGE 101  ROUTINE LOS.CHECK             4881
    PAGE 104  ROUTINE MINE.DELAY            5034
    PAGE 113  ROUTINE PREPARE.LIST          5434 5438
    PAGE 114  ROUTINE PREP.WITHDRAW         5472 5485
    PAGE 118  ROUTINE REIN.ARRIVE           5638 5641
    PAGE 125  ROUTINE WHAT.NEXT             5901 5910
    PAGE 126                                5958 5971
    PAGE 133  ROUTINE CHECK.FORCE           6163 6204
    PAGE 142  ROUTINE DEAD.UNIT             6539 6550
    PAGE 143                                6583
    PAGE 145  ROUTINE FIN.BATTLE            6730
    PAGE 146  ROUTINE INTER.BATTLE          6779
    PAGE 147                                6801 6844 6849
    PAGE 148                                6881 6886
    PAGE 150  ROUTINE PK.COMPUTE            6964 6989 6996
    PAGE 173  ROUTINE DUST.EFFECTS          7913 7927 7933
    PAGE 174                                7977 7992
    PAGE 200  ROUTINE ILLUM.EFFECTS         9227 9234
    PAGE 216  ROUTINE REQUEST.FASCAM        9936
    PAGE 217                                9951
    PAGE 218  ROUTINE REQUEST.ILLUM            3   10
    PAGE 219                                   66   81
    PAGE 222  ROUTINE REQUEST.SMOKE           192  198
    PAGE 223                                  239  254
    PAGE 226  ROUTINE REQUEST.WD.FASCAM       410  416
    PAGE 231  ROUTINE SIZE.ESTIMATE           663  669
    PAGE 235  ROUTINE SMOKE.EFFECTS           801  817  824
    PAGE 236                                  901
    PAGE 261  ROUTINE BTRY.EFFECTS           2071
    PAGE 263                                 2170 2187
    PAGE 268                                 2449
    PAGE 326  ROUTINE DESTROY.ORD            5138
    PAGE 328  ROUTINE EMPTY                  5180
    PAGE 345  ROUTINE TERM.CHECK             5776
    PAGE 369  EVENT ENGAGEMENT               6710
    PAGE 373  EVENT GET.NX.ORD               6899
    PAGE 374                                 6972 6975
    PAGE 378  EVENT HELO.ENGAGEMENT          7159
    PAGE 395  EVENT START.BATTLE             7775
    PAGE 399                                 7976 7993
    PAGE 430  PROCESS AIR.OBSERVER           9455
    PAGE 463  PROCESS HOW.REPAIR             1200
    PAGE 478  PROCESS WITH.DRAW              1957
    PAGE 479                                 2014
```

```
    PAGE 487  PROCESS ASSESSMENT                    2437
    PAGE 488                                        2504
    PAGE 489                                        2538
    PAGE 494  PROCESS SHOOT.OUT                     2856
    PAGE 495                                        2929
    PAGE 497                                        2989 3005 3031
    PAGE 499                                        3134
    PAGE 502                                        3290
    PAGE 513  PROCESS HELICOPTER.FIRE               3852 3853
    PAGE 514                                        3947
    PAGE 530  ROUTINE EQ.TE.INPUT                   4614
    PAGE 563  ROUTINE MPDB.INPUT                    5936
    PAGE 566  ROUTINE SENSOR.INPUT                  6108
    PAGE 568                                        6195
    PAGE 604  ROUTINE ERROR.STOP                    7665
    PAGE 630  FUNCTION COLLISION                    8699 8702
    PAGE 631  FUNCTION COMBINATIONS                 8721
    PAGE 636  FUNCTION HE.WLA                       8914
EST.COVERAGE
    PAGE   2  PROGRAM REVISIONS                      102
    PAGE   3                                         172
    PAGE 176  ROUTINE EST.COVERAGE                  8004 8024
    PAGE 179                                        8186
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS              1399
EST.MIL.WORTH
    PAGE 180  ROUTINE EST.MIL.WORTH                 8219 8239
    PAGE 241  ROUTINE TARGET.ANALYSIS               1100
EST.RANGE
    PAGE  50  ''SECTION FOR DEFINITIONS             2853
    PAGE 184  ROUTINE FA.BN.ASGN                    8423 8424
    PAGE 208  ROUTINE PGM.MSN.ASGN                  9526 9529
    PAGE 209                                        9556 9558 9603 9604
    PAGE 240  ROUTINE TARGET.ANALYSIS               1030 1031
    PAGE 632  FUNCTION EST.RANGE                    8744 8757
    PAGE 709  ''PROGRAM  OLDER.VERSION              2272
EST.TR.RANGE
    PAGE  50  ''SECTION FOR DEFINITIONS             2854
    PAGE 171  ROUTINE COMPARE.TRS                   7834 7835
    PAGE 633  FUNCTION EST.TR.RANGE                 8761 8770
    PAGE 709  ''PROGRAM  OLDER.VERSION              2273
EVENTS
    PAGE  39  ''SECTION FOR EVENTS                  2199
    PAGE  41                                        2341 2342 2347
    PAGE 698  ''PROGRAM  OLDER.VERSION              1637
    PAGE 700                                        1776
EXPONENTIAL.F
    PAGE 422  PROCESS AC.ATK.TGT                    9074 9075
    PAGE 463  PROCESS HOW.REPAIR                    1192 1193 1207 1208
    PAGE 551  ROUTINE BTRY.INPUT                    5494 5496 5500 5502
    PAGE 640  ROUTINE EXPONENTIAL.F                 9026
FARRP.CHECK
    PAGE 297  ROUTINE FARRP.CHECK                   3856 3905
    PAGE 401  EVENT START.BATTLE                    8094 8118
FARRP.INPUT
    PAGE 521  ROUTINE MAIN2                         4275 4277
    PAGE 572  ROUTINE FARRP.INPUT                   6327 6374
```

```
FASCAM.COMPUTATION
    PAGE 181  ROUTINE FASCAM.COMPUTATION              8242
    PAGE 185  ROUTINE FA.BN.ASGN                      8474
FA.BN.ASGN
    PAGE   2  PROGRAM REVISIONS                        90
    PAGE   3                                          142
    PAGE 181  ROUTINE FASCAM.COMPUTATION              8246
    PAGE 182  ROUTINE FA.BN.ASGN                      8286 8330 8337
    PAGE 183                                          8348 8355
    PAGE 184                                          8417 8441
    PAGE 475  PROCESS TARGET.REPORT                   1814
FA.BN.MOVEMENT
    PAGE   3  PROGRAM REVISIONS                        130
    PAGE  89  ROUTINE FA.BN.MOVEMENT                  4395
    PAGE 252  ROUTINE MINE.EFFECTS                    1629
    PAGE 271  ROUTINE BTRY.EFFECTS                    2603
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                 2967
    PAGE 282  ROUTINE AC.DF.EFFECTS                   3147
    PAGE 355  EVENT ARTY.OCCUPATION                   6250
    PAGE 384  EVENT OFF.LINE.ATTRITION                7453
    PAGE 390  EVENT SCHEDULE.ARTY.MOVEMENT            7589
    PAGE 486  PROCESS FIRE.MISSION                    2399
FBN.FD.INPUT
    PAGE 520  ROUTINE MAIN2                           4233 4235
    PAGE 553  ROUTINE FBN.FD.INPUT                    5539
FDC.TR.DEQ
    PAGE 332  ROUTINE FDC.TR.DEQ                      5341
    PAGE 333  ROUTINE FDC.TR.ENQ                      5364
    PAGE 334  ROUTINE FINISH.COMPUTATION              5404
FDC.TR.ENQ
    PAGE 332  ROUTINE FDC.TR.DEQ                      5345
    PAGE 333  ROUTINE FDC.TR.ENQ                      5357
    PAGE 472  PROCESS TARGET.REPORT                   1635
FD.EFFECTS.REQ
    PAGE 187  ROUTINE FD.EFFECTS.REQ                  8521
    PAGE 474  PROCESS TARGET.REPORT                   1764
FEBA.BAND
    PAGE   4  PROGRAM REVISIONS                        185
    PAGE  50  ''SECTION FOR DEFINITIONS               2855
    PAGE 181  ROUTINE FASCAM.COMPUTATION              8254 8255
    PAGE 182  ROUTINE FA.BN.ASGN                      8310 8311
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION           9012 9013
    PAGE 199  ROUTINE ILLUM.COMPUTATION               9145 9146
    PAGE 233  ROUTINE SMOKE.COMPUTATION                699  700
    PAGE 473  PROCESS TARGET.REPORT                   1676 1679
    PAGE 634  FUNCTION FEBA.BAND                      8774
    PAGE 635                                          8854
    PAGE 709  ''PROGRAM  OLDER.VERSION                2274
FEBA.INITIAL
    PAGE  61  ROUTINE FEBA.INITIAL                    3184
    PAGE 521  ROUTINE MAIN2                           4257 4259
FEBA.SORTIE
    PAGE  40  ''SECTION FOR EVENTS                    2263
    PAGE 371  EVENT FEBA.SORTIE                       6775 6826
    PAGE 372                                          6867 6872
    PAGE 507  ROUTINE SENSOR.INPUT                    6185
```

```
    PAGE 601  ROUTINE BETWEEN.ROUTINE                7592 7593
    PAGE 616  ROUTINE SNAP2                          8178
    PAGE 699  ''PROGRAM  OLDER.VERSION               1700
FILE.FD.SCHD
    PAGE  50  ''SECTION FOR DEFINITIONS              2856
    PAGE  63  ROUTINE FILE.FD.SCHD                   3260 3266
    PAGE  64                                         3323 3353
    PAGE 476  PROCESS TARGET.REPORT                  1849
    PAGE 709  ''PROGRAM  OLDER.VERSION               2275
FILE.KAD.SENSOR
    PAGE  65  ROUTINE FILE.KAD.SENSOR                3358
    PAGE 354  EVENT AD.ENGAGEMENT                    6219
FINAL.COVERAGE
    PAGE   2  PROGRAM REVISIONS                       105
    PAGE 188  ROUTINE FINAL.COVERAGE                 8547
    PAGE 265  ROUTINE BTRY.EFFECTS                   2275
    PAGE 266                                         2345 2353 2361
    PAGE 269                                         2494
FIND.START.TIME
    PAGE 194  ROUTINE FIND.START.TIME                8891 8903
    PAGE 473  PROCESS TARGET.REPORT                  1683
FINISH.COMPUTATION
    PAGE 334  ROUTINE FINISH.COMPUTATION             5379 5396
    PAGE 470  PROCESS TARGET.REPORT                  1528
    PAGE 471                                         1574
    PAGE 472                                         1651
    PAGE 474                                         1721
    PAGE 475                                         1788
    PAGE 476                                         1888
FIN.BATTLE
    PAGE 145  ROUTINE FIN.BATTLE                     6692
    PAGE 345  ROUTINE TERM.CHECK                     5790
FIRE.MISSION
    PAGE   2  PROGRAM REVISIONS                        94  113
    PAGE  35  ''SECTION FOR PROCESSES                1994
    PAGE 163  ROUTINE BTRY.FM.DEQ                    7541
    PAGE 184  ROUTINE FA.BN.ASGN                     8414 8438 8459
    PAGE 185                                         8491 8498 8504
    PAGE 209  ROUTINE PGM.MSN.ASGN                   9593
    PAGE 210                                         9660
    PAGE 211                                         9692
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS         2754 2768
    PAGE 435  PROCESS ARTY.ASSESS                    9691 9696 9697 9699 9701
    PAGE 437                                         9846
    PAGE 480  PROCESS FIRE.MISSION                   2027
    PAGE 482                                         2156
    PAGE 486                                         2376
    PAGE 592  ROUTINE AMMO.RPT                       7122
    PAGE 599  ROUTINE BETWEEN.ROUTINE                7472 7473
    PAGE 616  ROUTINE SNAP2                          8147
    PAGE 619  ROUTINE SNAP.R                         8308
    PAGE 694  ''PROGRAM  OLDER.VERSION               1432
FLIGHT.PATH
    PAGE 314  ROUTINE FLIGHT.PATH                    4630
    PAGE 505  PROCESS CAS.MISSION                    3416
```

```
FORM.TF.LIST
    PAGE  66  ROUTINE FORM.TF.LIST                3416 3443
    PAGE 539  ROUTINE READ.ORDERS                 5034
FORWARD.OBSERVER
    PAGE  36  ''SECTION FOR PROCESSES             2033
    PAGE 160  ROUTINE ATTRIT.SENSOR               7405 7411
    PAGE 238  ROUTINE SWITCH.FO                    937  947  951  953
    PAGE 438  PROCESS FORWARD.OBSERVER            9854
    PAGE 482  PROCESS FIRE.MISSION                2177
    PAGE 485                                      2367 2372
    PAGE 566  ROUTINE SENSOR.INPUT                6104
    PAGE 599  ROUTINE BETWEEN.ROUTINE             7476 7477
    PAGE 616  ROUTINE SNAP2                       8148
    PAGE 619  ROUTINE SNAP.R                      8287
    PAGE 695  ''PROGRAM  OLDER.VERSION            1471
FO.DETECTION
    PAGE   2  PROGRAM REVISIONS                     98
    PAGE   3                                       165
    PAGE 254  ROUTINE FO.DETECTION                1671 1693
    PAGE 255                                      1747
    PAGE 258                                      1904 1928
    PAGE 440  PROCESS FORWARD.OBSERVER               3
    PAGE 441                                        51
FRAC.COMPUTE
    PAGE 179  ROUTINE EST.COVERAGE                8192 8195 8199 8202
    PAGE 192  ROUTINE FINAL.COVERAGE              8834 8836
    PAGE 193                                      8844 8846
    PAGE 335  ROUTINE FRAC.COMPUTE                5413
GAMMA.F
    PAGE 652  ROUTINE GAMMA.F                     9236
GENERAL.BATTLE
    PAGE  67  ROUTINE GENERAL.BATTLE              3448
    PAGE 397  EVENT START.BATTLE                  7396
GET.NX.ORD
    PAGE  40  ''SECTION FOR EVENTS                2268
    PAGE  88  ROUTINE END.MOVE                    4381
    PAGE 117  ROUTINE PROX.POS                    5588
    PAGE 118  ROUTINE REIN.ARRIVE                 5631
    PAGE 125  ROUTINE WHAT.NEXT                   5881
    PAGE 126                                      5961
    PAGE 146  ROUTINE INTER.BATTLE                6769 6782
    PAGE 348  EVENT ACT.REINF                     5888
    PAGE 373  EVENT GET.NX.ORD                    6875
    PAGE 601  ROUTINE BETWEEN.ROUTINE             7596 7597
    PAGE 616  ROUTINE SNAP2                       8179
    PAGE 699  ''PROGRAM  OLDER.VERSION            1705
GET.TERRAIN
    PAGE 210  ROUTINE PGM.MSN.ASGN                9613
    PAGE 243  ROUTINE UNIT.ENVIR                  1170
    PAGE 261  ROUTINE BTRY.EFFECTS                2036
    PAGE 336  ROUTINE GET.TERRAIN                 5434
    PAGE 394  EVENT START.ARTY.MOVEMENT           7718
    PAGE 403  EVENT START.MOVE                    8213
    PAGE 413  EVENT ACT.ATK                       8700
HC.ARRIVE.BATTLE
    PAGE  36  ''SECTION FOR PROCESSES             2054
```

```
    PAGE 337  ROUTINE HC.EMPTY                    5500 5501 5503 5504 5505 5508 5509 5511 5512 5513
    PAGE 338                                      5515
    PAGE 391  EVENT SEND.TEAM                     7615
    PAGE 402  EVENT START.BATTLE                  8165
    PAGE 443  PROCESS HC.ARRIVE.BATTLE              93
    PAGE 447                                       380
    PAGE 448                                       398
    PAGE 599  ROUTINE BETWEEN.ROUTINE             7480 7481
    PAGE 616  ROUTINE SNAP2                       8149
    PAGE 695  ''PROGRAM  OLDER.VERSION            1492
HC.COMPUTE.TIMES
    PAGE 128  ROUTINE BTL.CHECK                   6009 6028
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS          3527
    PAGE 297  ROUTINE FARRP.CHECK                 3890
    PAGE 298  ROUTINE HC.COMPUTE.TIMES            3913
    PAGE 299                                      3995
    PAGE 450  PROCESS HC.RETURN.FARRP              491
HC.DEPART.BATTLE
    PAGE  40  ''SECTION FOR EVENTS                2274
    PAGE 338  ROUTINE HC.EMPTY                    5568 5569 5571 5572
    PAGE 339                                      5574 5575 5581 5582 5584 5588 5590 5591 5594 5595
    PAGE 375  EVENT HC.DEPART.BATTLE              6988
    PAGE 461  PROCESS HEL.TARGET.ACQUISITION      1154
    PAGE 601  ROUTINE BETWEEN.ROUTINE             7600 7601
    PAGE 616  ROUTINE SNAP2                       8180
    PAGE 699  ''PROGRAM  OLDER.VERSION            1711
HC.DISENGAGE
    PAGE 300  ROUTINE HC.DISENGAGE                4003 4021
    PAGE 338  ROUTINE HC.EMPTY                    5535
    PAGE 339                                      5578
    PAGE 375  EVENT HC.DEPART.BATTLE              7011
HC.EMPTY
    PAGE 337  ROUTINE HC.EMPTY                    5457 5471 5482 5495
    PAGE 338                                      5517 5532
    PAGE 339                                      5576
    PAGE 356  EVENT BTL.ENDED                     6274 6286
    PAGE 448  PROCESS HC.ARRIVE.BATTLE             423
HC.RETURN.FARRP
    PAGE  36  ''SECTION FOR PROCESSES             2058
    PAGE 337  ROUTINE HC.EMPTY                    5474 5475 5477 5478 5480 5488 5506
    PAGE 338                                      5543
    PAGE 339                                      5585
    PAGE 375  EVENT HC.DEPART.BATTLE              7019
    PAGE 445  PROCESS HC.ARRIVE.BATTLE             251
    PAGE 448                                       429
    PAGE 450  PROCESS HC.RETURN.FARRP              460
    PAGE 451                                       556  574
    PAGE 599  ROUTINE BETWEEN.ROUTINE             7484 7485
    PAGE 616  ROUTINE SNAP2                       8150
    PAGE 695  ''PROGRAM  OLDER.VERSION            1496
HEADING
    PAGE  56  ROUTINE MAIN1                       3045
    PAGE 605  ROUTINE HEADING                     7679
    PAGE 607  ROUTINE KV.PRINT                    7794
    PAGE 608                                      7817
    PAGE 610  ROUTINE KV.SCOREBOARD               7914 7921
```

```
HELICOPTER.FIRE
    PAGE  36  ''SECTION FOR PROCESSES                2062
    PAGE 301  ROUTINE HC.DISENGAGE                   4074 4075 4077 4078 4079 4080
    PAGE 312  ROUTINE INTER.HELO                     4617 4618 4620 4621 4622 4623
    PAGE 338  ROUTINE HC.EMPTY                       5561 5562 5564
    PAGE 376  EVENT HELO.ENGAGEMENT                  7042
    PAGE 377                                         7105 7106 7137
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION         1087 1088 1090
    PAGE 488  PROCESS ASSESSMENT                     2506 2507 2510 2511
    PAGE 510  PROCESS HELICOPTER.FIRE                3658 3670 3677 3678 3687 3689 3690 3691 3711 3713 3714
    PAGE 511                                         3716 3724 3738
    PAGE 512                                         3826
    PAGE 513                                         3834 3836 3837 3838 3854 3865
    PAGE 514                                         3904
    PAGE 515                                         3953 3961 3963 3964 3965 3977 3978 3979 3990 4002 4003 4004
    PAGE 517                                         4081 4082 4083 4118 4119 4120
    PAGE 518                                         4122 4129 4140 4141 4142 4168
    PAGE 600  ROUTINE BETWEEN.ROUTINE                7488
    PAGE 616  ROUTINE SNAP2                          8151
    PAGE 695  ''PROGRAM  OLDER.VERSION               1500
HELO.ENGAGEMENT
    PAGE  40  ''SECTION FOR EVENTS                   2279
    PAGE 301  ROUTINE HC.DISENGAGE                   4066 4067 4069 4070
    PAGE 312  ROUTINE INTER.HELO                     4609 4610 4612 4613
    PAGE 338  ROUTINE HC.EMPTY                       5554 5555 5557 5558
    PAGE 376  EVENT HELO.ENGAGEMENT                  7030 7058 7059 7060 7062 7063
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION          868
    PAGE 458                                          950  968
    PAGE 460                                         1043 1081 1082 1084 1085
    PAGE 602  ROUTINE BETWEEN.ROUTINE                7604 7605
    PAGE 616  ROUTINE SNAP2                          8181
    PAGE 699  ''PROGRAM  OLDER.VERSION               1716
HEL.RANGE.COMPUTE
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4119
    PAGE 377  EVENT HELO.ENGAGEMENT                  7141
    PAGE 455  PROCESS HEL.TARGET.ACQUISITION          805
    PAGE 457                                          882
    PAGE 459                                          999
    PAGE 495  PROCESS SHOOT.OUT                      2892
    PAGE 498                                         3083
    PAGE 511  PROCESS HELICOPTER.FIRE                3752
    PAGE 513                                         3871
HEL.TARGET.ACQUISITION
    PAGE  37  ''SECTION FOR PROCESSES                2078
    PAGE 311  ROUTINE INTER.HELO                     4545 4547 4549 4550 4551 4552 4553 4555
    PAGE 312                                         4625
    PAGE 338  ROUTINE HC.EMPTY                       5520 5521 5523 5524 5525 5526 5527 5530 5531 5538 5539 5542 5546 5548 5549
    PAGE 376  EVENT HELO.ENGAGEMENT                  7039
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                130  131  134  135  136  137
    PAGE 449                                          449
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          692  714  723
    PAGE 460                                         1063
    PAGE 461                                         1143
    PAGE 600  ROUTINE BETWEEN.ROUTINE                7492 7493
    PAGE 616  ROUTINE SNAP2                          8152
    PAGE 696  ''PROGRAM  OLDER.VERSION               1516
```

```
HE.LA.INPUT
     PAGE 521  ROUTINE MAIN2                          4242 4244
     PAGE 558  ROUTINE HE.LA.INPUT                    5750
HE.OR.ICM.COMPUTATION
     PAGE   2  PROGRAM REVISIONS                       117
     PAGE 185  ROUTINE FA.BN.ASGN                     8478
     PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          8989 9023
HE.WLA
     PAGE  50  ''SECTION FOR DEFINITIONS              2857
     PAGE 178  ROUTINE EST.COVERAGE                   8161 8162
     PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9042 9045
     PAGE 197                                         9098
     PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9353 9354
     PAGE 636  FUNCTION HE.WLA                        8859 8872 8892
     PAGE 709  ''PROGRAM  OLDER.VERSION               2276
HOW.REPAIR
     PAGE  37  ''SECTION FOR PROCESSES                2084
     PAGE 463  PROCESS HOW.REPAIR                     1168
     PAGE 485  PROCESS FIRE.MISSION                   2340
     PAGE 600  ROUTINE BETWEEN.ROUTINE                7496 7497
     PAGE 616  ROUTINE SNAP2                          8154
     PAGE 696  ''PROGRAM  OLDER.VERSION               1522
ICM.WLA
     PAGE  50  ''SECTION FOR DEFINITIONS              2858
     PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9037 9040
     PAGE 638  FUNCTION ICM.WLA                       8957 8971
     PAGE 709  ''PROGRAM  OLDER.VERSION               2277
IF
     PAGE 234  ROUTINE SMOKE.EFFECTS                   764
     PAGE 242  ROUTINE UNIT.ENVIR                     1125
     PAGE 252  ROUTINE MINE.EFFECTS                   1608 1610 1625
     PAGE 270  ROUTINE BTRY.EFFECTS                   2586 2598
     PAGE 271                                         2604 2618
     PAGE 278  ROUTINE AC.BOMB.EFFECTS                2955
     PAGE 282  ROUTINE AC.DF.EFFECTS                  3137
     PAGE 286  ROUTINE CHECK.CAS.CONSTRAINTS          3344
     PAGE 316  ROUTINE FLIGHT.PATH                    4778
     PAGE 325  ROUTINE DEQ.FEBA.SET                   5093 5097
     PAGE 327  ROUTINE DQ.CMSN.QUEUE                  5150
     PAGE 331  ROUTINE ENQ.FEBA.SET                   5336
     PAGE 334  ROUTINE FINISH.COMPUTATION             5400
     PAGE 347  EVENT ACT.REINF                        5816 5817
     PAGE 348                                         5864
     PAGE 356  EVENT BTL.ENDED                        6258 6268
     PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE              6644
     PAGE 421  PROCESS AC.ATK.TGT                     8969 8973
     PAGE 422                                         9034
     PAGE 443  PROCESS HC.ARRIVE.BATTLE                129
     PAGE 448                                          403
     PAGE 450  PROCESS HC.RETURN.FARRP                 467
     PAGE 452                                          578
     PAGE 453                                          640
     PAGE 458  PROCESS HEL.TARGET.ACQUISITION          937
     PAGE 475  PROCESS TARGET.REPORT                  1785
     PAGE 481  PROCESS FIRE.MISSION                   2085
     PAGE 483                                         2225
```

```
    PAGE 485                                    2327
    PAGE 487  PROCESS ASSESSMENT                2448
    PAGE 506  PROCESS CAS.MISSION               3491
    PAGE 539  ROUTINE READ.ORDERS               4986 4987 4990 4991
    PAGE 542  ROUTINE ORD.DEF                   5124 5126
    PAGE 543  ROUTINE ORD.ATK                   5146 5147
    PAGE 544  ROUTINE ORD.REINF                 5167
    PAGE 546  ROUTINE ORD.MOVCOR                5222 5224 5227
    PAGE 611  ROUTINE OUTPUT.ATTRITION          7950 7964 7965 7966
    PAGE 636  FUNCTION HE.WLA                   8876 8916
    PAGE 638  FUNCTION ICM.WLA                  8999
    PAGE 709  ''PROGRAM  OLDER.VERSION          2297
ILLUM.COMPUTATION
    PAGE 185  ROUTINE FA.BN.ASGN                8464
    PAGE 199  ROUTINE ILLUM.COMPUTATION         9133
ILLUM.EFFECTS
    PAGE 200  ROUTINE ILLUM.EFFECTS             9181
    PAGE 484  PROCESS FIRE.MISSION              2295
ILLUM.INPUT
    PAGE 521  ROUTINE MAIN2                     4281 4283
    PAGE 576  ROUTINE ILLUM.INPUT               6494
INITIAL.DETECT
    PAGE  93  ROUTINE INITIAL.DETECT            4587
    PAGE 400  EVENT START.BATTLE                8075
INITIAL.MOVE
    PAGE  94  ROUTINE INITIAL.MOVE              4613
    PAGE 400  EVENT START.BATTLE                8078
INIT.PREPLAN.CAS
    PAGE  40  ''SECTION FOR EVENTS              2285
    PAGE 380  EVENT INIT.PREPLAN.CAS            7229 7256
    PAGE 585  ROUTINE TACAIR.INPUT              6968
    PAGE 602  ROUTINE BETWEEN.ROUTINE           7608 7609
    PAGE 616  ROUTINE SNAP2                     8182
    PAGE 699  ''PROGRAM  OLDER.VERSION          1722
INIT.REINF
    PAGE  95  ROUTINE INIT.REINF                4637
    PAGE 348  EVENT ACT.REINF                   5870
INTER.BATTLE
    PAGE 118  ROUTINE REIN.ARRIVE               5618
    PAGE 146  ROUTINE INTER.BATTLE              6739
    PAGE 405  EVENT START.MOVE                  8338
    PAGE 409  EVENT UPDATE.LOC                  8506
INTER.HELO
    PAGE 146  ROUTINE INTER.BATTLE              6785
    PAGE 311  ROUTINE INTER.HELO                4512
JOHNSON.CRITERIA
    PAGE 120  ROUTINE SEARCH                    5749
    PAGE 661  ROUTINE JOHNSON.CRITERIA          9615
KV.INPUT
    PAGE 520  ROUTINE MAIN2                     4202 4204
    PAGE 529  ROUTINE KV.INPUT                  4523
    PAGE 530  ROUTINE EQ.TE.INPUT               4629
KV.PRINT
    PAGE 606  ROUTINE KV.PRINT                  7688
    PAGE 609  ROUTINE KV.SCOREBOARD             7849 7851
```

```
KV.SCOREBOARD
    PAGE   4  PROGRAM REVISIONS                    190
    PAGE 368  EVENT END.SIMULATION                6672
    PAGE 609  ROUTINE KV.SCOREBOARD               7842
LIB$INIT_TIMER
    PAGE  55  ''PROGRAM'' MAIN                    3030
    PAGE 649  ''PROGRAM LIB$INIT_TIMER            9214
LIB$STAT_TIMER
    PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT       8816
    PAGE 650  ''PROGRAM LIB$STAT_TIMER            9221
LINE.CIRCLE
    PAGE 316  ROUTINE FLIGHT.PATH                 4795
    PAGE 645  ROUTINE LINE.CIRCLE                 9116
LINE.OF.SIGHT
    PAGE  96  ROUTINE LINE.OF.SIGHT               4663
    PAGE 400  EVENT START.BATTLE                  8070
LOCATE.SEARCH.AREA
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA          4770
    PAGE 438  PROCESS FORWARD.OBSERVER            9893
LOCATE.SECTOR
    PAGE   4  PROGRAM REVISIONS                    187  199
    PAGE  61  ROUTINE FEBA.INITIAL                3225
    PAGE  89  ROUTINE FA.BN.MOVEMENT              4436
    PAGE  91                                      4516
    PAGE 100  ROUTINE LOCATE.SECTOR               4831 4867
    PAGE 119  ROUTINE RESET.FEBA.SECTOR           5681
    PAGE 239  ROUTINE TARGET.ANALYSIS             1000
    PAGE 246  ROUTINE VOLLEY                      1317
    PAGE 260  ROUTINE BTRY.EFFECTS                2020
    PAGE 325  ROUTINE DEQ.FEBA.SET                5089 5095
    PAGE 331  ROUTINE ENQ.FEBA.SET                5312
    PAGE 347  EVENT ACT.REINF                     5840
    PAGE 360  EVENT CFR.ON                        6429
    PAGE 371  EVENT FEBA.SORTIE                   6797
    PAGE 372                                      6840
    PAGE 634  FUNCTION FEBA.BAND                  8792 8810
LOCATE.SECTORPG
    PAGE 660  FUNCTION STAY.TIME                  9600
LOS.CHECK
    PAGE  84  ROUTINE CHANGE.LOC                  4205
    PAGE  85                                      4265 4298
    PAGE 101  ROUTINE LOS.CHECK                   4871
MADS.INPUT
    PAGE 522  ROUTINE MAIN2                       4300 4302
    PAGE 587  ROUTINE MADS.INPUT                  6978
MAIN
    PAGE  55  ''PROGRAM'' MAIN                    3023
    PAGE 115  ROUTINE PROX.CHECK                  5511
MAIN1
    PAGE  55  ''PROGRAM'' MAIN                    3031
    PAGE  56  ROUTINE MAIN1                       3036
MAIN2
    PAGE  55  ''PROGRAM'' MAIN                    3032
    PAGE 520  ROUTINE MAIN2                       4183
MAIN3
    PAGE  55  ''PROGRAM'' MAIN                    3033
```

```
    PAGE  57  ROUTINE MAIN3                          3069
MAO.INPUT
    PAGE 521  ROUTINE MAIN2                          4260 4262
    PAGE 564  ROUTINE MAO.INPUT                      5953
MARGINAL.EFFECTS.ADJ
    PAGE 198  ROUTINE HE.OR.ICM.COMPUTATION          9122
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9300 9316
MCFR.INPUT
    PAGE 521  ROUTINE MAIN2                          4251 4253
    PAGE 562  ROUTINE MCFR.INPUT                     5878
MFO.INPUT
    PAGE 520  ROUTINE MAIN2                          4218 4220
    PAGE 538  ROUTINE MFO.INPUT                      4930
MINE.ASSESS
    PAGE  37  ''SECTION FOR PROCESSES                2087
    PAGE 251  ROUTINE MINE.EFFECTS                   1531
    PAGE 329  ROUTINE EMPTY                          5225 5226 5228 5229 5230
    PAGE 464  PROCESS MINE.ASSESS                    1216 1229
    PAGE 600  ROUTINE BETWEEN.ROUTINE                7500 7501
    PAGE 696  ''PROGRAM  OLDER.VERSION               1525
MINE.DELAY
    PAGE 103  ROUTINE MINE.DELAY                      932 4962
    PAGE 104                                         5015
    PAGE 251  ROUTINE MINE.EFFECTS                   1519
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                2984
MINE.EFFECTS
    PAGE 227  ROUTINE REQUEST.WD.FASCAM               462
    PAGE 250  ROUTINE MINE.EFFECTS                   1459 1515
    PAGE 251                                         1541 1551
    PAGE 253                                         1648 1666
    PAGE 393  EVENT START.ARTY.MOVEMENT              7689
    PAGE 405  EVENT START.MOVE                       8298
    PAGE 410  EVENT UPDATE.LOC                       8553
    PAGE 484  PROCESS FIRE.MISSION                   2269
MINE.INPUT
    PAGE 521  ROUTINE MAIN2                          4284 4286
    PAGE 577  ROUTINE MINE.INPUT                     6533
MIN.MOVE
    PAGE  86  ROUTINE CHANGE.LOC                     4301
    PAGE  94  ROUTINE INITIAL.MOVE                   4628
    PAGE 106  ROUTINE MIN.MOVE                       5060
    PAGE 174  ROUTINE DUST.EFFECTS                   7981 7996
    PAGE 236  ROUTINE SMOKE.EFFECTS                   905
    PAGE 479  PROCESS WITH.DRAW                      2021
MOVE
    PAGE   1  ROUTINE FOR CROSS_REFERENCING            53
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES       1598
    PAGE  40  ''SECTION FOR EVENTS                   2290
    PAGE  83  ROUTINE CHANGE.LOC                     4129
    PAGE  85                                         4297
    PAGE  89  ROUTINE FA.BN.MOVEMENT                 4408
    PAGE  90                                         4501
    PAGE  95  ROUTINE INIT.REINF                     4641
    PAGE 106  ROUTINE MIN.MOVE                       5079 5095
    PAGE 114  ROUTINE PREP.WITHDRAW                  5481 5482 5488 5489
    PAGE 134  ROUTINE CHECK.FOR.MINES                6222
```

```
    PAGE 142  ROUTINE DEAD.UNIT                    6531
    PAGE 163  ROUTINE BTRY.FM.DEQ                  7518
    PAGE 173  ROUTINE DUST.EFFECTS                 7915 7916
    PAGE 174                                       7979 7980 7994 7995
    PAGE 210  ROUTINE PGM.MSN.ASGN                 9655
    PAGE 235  ROUTINE SMOKE.EFFECTS                 803  804
    PAGE 236                                        903  904
    PAGE 244  ROUTINE UNIT.ENVIR                   1262
    PAGE 261  ROUTINE BTRY.EFFECTS                 2067 2068 2073 2074
    PAGE 268                                       2439
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE            4141
    PAGE 328  ROUTINE EMPTY                        5183 5184 5186 5187
    PAGE 355  EVENT ARTY.OCCUPATION                6246
    PAGE 381  EVENT MOVE                           7267
    PAGE 393  EVENT START.ARTY.MOVEMENT            7711
    PAGE 394                                       7738
    PAGE 403  EVENT START.MOVE                     8198
    PAGE 404                                       8274
    PAGE 407  EVENT STOP.ARTY.MOVEMENT             8395
    PAGE 408  EVENT UPDATE.LOC                     8419
    PAGE 444  PROCESS HC.ARRIVE.BATTLE              187
    PAGE 479  PROCESS WITH.DRAW                    2010 2011 2016 2017
    PAGE 545  ROUTINE ORD.MOVDIS                   5184 5186 5188 5188 5189 5191
    PAGE 546  ROUTINE ORD.MOVCOR                   5209 5211 5213 5217 5222
    PAGE 602  ROUTINE BETWEEN.ROUTINE              7612 7613
    PAGE 616  ROUTINE SNAP2                        8183
    PAGE 687  ''PROGRAM  OLDER.VERSION             1037
    PAGE 699                                       1727
MPDB.INPUT
    PAGE 521  ROUTINE MAIN2                        4254 4256
    PAGE 563  ROUTINE MPDB.INPUT                   5915
MRT.TO.FREQ
    PAGE 120  ROUTINE SEARCH                       5742
    PAGE 646  ROUTINE MRT.TO.FREQ                  9155
MUNS.INPUT
    PAGE 520  ROUTINE MAIN2                        4236 4238
    PAGE 555  ROUTINE MUNS.INPUT                   5602
NEW.SEGMENT
    PAGE 107  ROUTINE NEW.SEGMENT                  5102
    PAGE 122  ROUTINE SEGMENT.ADJUST               5785
NOISE.DEGRADE
    PAGE 205  ROUTINE NOISE.DEGRADE                9385
    PAGE 206  ROUTINE PDB.DETECTION                9451
NORMAL.F
    PAGE 157  ROUTINE AO.DETECTION                 7273 7275 7277
    PAGE 193  ROUTINE FINAL.COVERAGE               8878 8879 8880
    PAGE 230  ROUTINE RPV.DETECTION                 596  598  600
    PAGE 242  ROUTINE UNIT.ENVIR                   1154 1155
    PAGE 243                                       1213 1214
    PAGE 244                                       1229 1230 1240 1241 1263 1264 1272 1273
    PAGE 257  ROUTINE FO.DETECTION                 1874 1875 1876
    PAGE 276  ROUTINE AC.BOMB.EFFECTS              2842 2843 2844
    PAGE 351  EVENT AD.ENGAGEMENT                  6041 6042
    PAGE 362  EVENT CFR.OPERATOR                   6517 6519 6521
    PAGE 387  EVENT PDB.OPERATOR                   7535 7536 7538
    PAGE 641  ROUTINE NORMAL.F                     9039
```

```
    PAGE 656  ROUTINE AR.DETECTION                    9429 9431 9433
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                 9577 9578 9580
OFF.LINE.ATTRITION
    PAGE  41  ''SECTION FOR EVENTS                    2342 2344
    PAGE 382  EVENT OFF.LINE.ATTRITION                7280 7287 7293 7306 7315 7324 7334
    PAGE 603  ROUTINE BETWEEN.ROUTINE                 7661 7662
    PAGE 700  ''PROGRAM  OLDER.VERSION                1776 1778
OLDER.VERSION
    PAGE 664  ''PROGRAM  OLDER.VERSION                9655
OPEN.INPUT.OUTPUT.FILES
    PAGE  55  ''PROGRAM'' MAIN                        3027
    PAGE 647  ROUTINE OPEN.INPUT.OUTPUT.FILES         9180 9188
    PAGE 648  ROUTINE PERFORM.INSTRUMENTATION         9209
ORD.ATK
    PAGE 540  ROUTINE READ.ORDERS                     5066
    PAGE 543  ROUTINE ORD.ATK                         5139
ORD.DEF
    PAGE 540  ROUTINE READ.ORDERS                     5055
    PAGE 542  ROUTINE ORD.DEF                         5114
ORD.MOVCOR
    PAGE 540  ROUTINE READ.ORDERS                     5081
    PAGE 546  ROUTINE ORD.MOVCOR                      5203
ORD.MOVDIS
    PAGE 540  ROUTINE READ.ORDERS                     5076
    PAGE 545  ROUTINE ORD.MOVDIS                      5179
ORD.REINF
    PAGE 540  ROUTINE READ.ORDERS                     5071
    PAGE 544  ROUTINE ORD.REINF                       5158
ORIENTATION
    PAGE  71  ROUTINE ORIENTATION                     3634 3646
    PAGE 397  EVENT START.BATTLE                      7906
OUTPUT.ATTRITION
    PAGE 253  ROUTINE MINE.EFFECTS                    1634
    PAGE 270  ROUTINE BTRY.EFFECTS                    2559
    PAGE 437  PROCESS ARTY.ASSESS                     9839
    PAGE 466  PROCESS MINE.ASSESS                     1335
    PAGE 490  PROCESS ASSESSMENT                      2593
    PAGE 491                                          2655
    PAGE 492                                          2712
    PAGE 517  PROCESS HELICOPTER.FIRE                 4106
    PAGE 611  ROUTINE OUTPUT.ATTRITION                7930
OUTPUT.EXPENDITURES
    PAGE 368  EVENT END.SIMULATION                    6674
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES             8489
PDB.ACTIVATION
    PAGE  40  ''SECTION FOR EVENTS                    2293
    PAGE 246  ROUTINE VOLLEY                          1345 1350
    PAGE 386  EVENT PDB.ACTIVATION                    7467
    PAGE 602  ROUTINE BETWEEN.ROUTINE                 7616 7617
    PAGE 616  ROUTINE SNAP2                           8184
    PAGE 699  ''PROGRAM  OLDER.VERSION                1730
PDB.DETECTION
    PAGE 206  ROUTINE PDB.DETECTION                   9428
    PAGE 386  EVENT PDB.ACTIVATION                    7498
PDB.OPERATOR
    PAGE  40  ''SECTION FOR EVENTS                    2297
```

```
    PAGE 160  ROUTINE ATTRIT.SENSOR                    7427
    PAGE 161                                           7428
    PAGE 206  ROUTINE PDB.DETECTION                    9461
    PAGE 387  EVENT PDB.OPERATOR                       7504 7518 7532
    PAGE 388                                           7564
    PAGE 602  ROUTINE BETWEEN.ROUTINE                  7620 7621
    PAGE 616  ROUTINE SNAP2                            8185
    PAGE 699  ''PROGRAM  OLDER.VERSION                 1734
PERFORM.INSTRUMENTATION
    PAGE  55  ''PROGRAM'' MAIN                         3028
    PAGE 648  ROUTINE PERFORM.INSTRUMENTATION          9192
PERMANENT_ENTITIES
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES          250
PGM.INPUT
    PAGE 521  ROUTINE MAIN2                            4278 4280
    PAGE 575  ROUTINE PGM.INPUT                        6449
PGM.MSN.ASGN
    PAGE   3  PROGRAM REVISIONS                         141
    PAGE 208  ROUTINE PGM.MSN.ASGN                     9489
    PAGE 473  PROCESS TARGET.REPORT                    1697
PHOTO.IR.FLIGHT
    PAGE  37  ''SECTION FOR PROCESSES                  2099
    PAGE 600  ROUTINE BETWEEN.ROUTINE                  7504 7505
    PAGE 616  ROUTINE SNAP2                            8155
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                  9474 9488
    PAGE 696  ''PROGRAM  OLDER.VERSION                 1537
PIR.DETECTION
    PAGE 212  ROUTINE PIR.DETECTION                    9701 9715
    PAGE 242  ROUTINE UNIT.ENVIR                       1127
    PAGE 659  PROCESS PHOTO IR.FLIGHT                  9541
PK.COMPUTE
    PAGE 149  ROUTINE PK.COMPUTE                       6893
    PAGE 495  PROCESS SHOOT.OUT                        2911
    PAGE 496                                           2933
    PAGE 501                                           3260
    PAGE 511  PROCESS HELICOPTER.FIRE                  3773
    PAGE 512                                           3782
PK.INPUT
    PAGE 281  ROUTINE AC.DF.EFFECTS                    3101
    PAGE 308  ROUTINE AD.SHOOT                         4395
    PAGE 520  ROUTINE MAIN2                            4196 4198
    PAGE 525  ROUTINE PK.INPUT                         4381
PLAT.COUNT
    PAGE 396  EVENT START.BATTLE                       7820 7831
    PAGE 714  ROUTINE PLAT.COUNT                       2505
POSITION
    PAGE  84  ROUTINE CHANGE.LOC                       4216
    PAGE  85                                           4287
    PAGE 110  ROUTINE POSITION                         5264
    PAGE 182  ROUTINE FA.BN.ASGN                       8327
    PAGE 208  ROUTINE PGM.MSN.ASGN                     9516
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE                4143
    PAGE 448  PROCESS HC.ARRIVE.BATTLE                  386
    PAGE 542  ROUTINE ORD.DEF                          5126
    PAGE 543  ROUTINE ORD.ATK                          5146
```

```
POSITION.OUT
    PAGE  94  ROUTINE INITIAL.MOVE                4630
    PAGE 368  EVENT END.SIMULATION                6673
    PAGE 389  EVENT POSITION.REPORT               7573
    PAGE 613  ROUTINE FOR POSITION.OUT            7993
POSITION REPORT
    PAGE  41  ''SECTION FOR EVENTS                2338
    PAGE  57  ROUTINE MAIN3                       3080
    PAGE 389  EVENT POSITION.REPORT               7568 7575
    PAGE 602  ROUTINE BETWEEN.ROUTINE             7652 7653
    PAGE 617  ROUTINE SNAP2                       8196
    PAGE 700  ''PROGRAM  OLDER.VERSION            1775
PREAMBLE
    PAGE   5  ''PROGRAM'' PREAMBLE                 218
    PAGE 664  ''PROGRAM  OLDER.VERSION            9655
PRED.POS
    PAGE 111  ROUTINE PRED.POS                    5299
    PAGE 257  ROUTINE FO.DETECTION                1850
PREPARE.LIST
    PAGE 112  ROUTINE PREPARE.LIST                5346
    PAGE 113                                      5406
    PAGE 412  EVENT ACT.ATK                       8653
    PAGE 413                                      8678
PREP.WITHDRAW
    PAGE 114  ROUTINE PREP.WITHDRAW               5462
    PAGE 115  ROUTINE PROX.CHECK                  5550
    PAGE 133  ROUTINE CHECK.FORCE                 6192
    PAGE 324  ROUTINE DECIDE                      5058
PROB.INF
    PAGE 120  ROUTINE SEARCH                      5753
    PAGE 340  ROUTINE PROB.INF                    5600
PROB.TIME
    PAGE 120  ROUTINE SEARCH                      5763
    PAGE 341  ROUTINE PROB.TIME                   5634
PROCESSES
    PAGE  33  ''SECTION FOR PROCESSES             1887 1890
    PAGE 142  ROUTINE DEAD.UNIT                   6526
    PAGE 692  ''PROGRAM  OLDER.VERSION            1326 1328
PROXIMITY REQ
    PAGE 171  ROUTINE COMPARE.TRS                 7828
    PAGE 662  ROUTINE PROXIMITY.REQ               9626
PROX.CHECK
    PAGE 115  ROUTINE PROX.CHECK                  5496
    PAGE 495  PROCESS SHOOT.OUT                   2906
    PAGE 498                                      3097
PROX.POS
    PAGE 117  ROUTINE PROX.POS                    5563
    PAGE 405  EVENT START.MOVE                    8345
    PAGE 409  EVENT UPDATE.LOC                    8513
P.E.M.INPUT
    PAGE 242  ROUTINE UNIT.ENVIR                  1121
    PAGE 520  ROUTINE MAIN2                       4224 4226
    PAGE 547  ROUTINE P.E.M.INPUT                 5255
RANGE COMPUTE
    PAGE  96  ROUTINE LINE.OF.SIGHT               4685
    PAGE 101  ROUTINE LOS.CHECK                   4906
```

```
    PAGE 107  ROUTINE NEW.SEGMENT                5113
    PAGE 123  ROUTINE TIME.TO.DETECT             5803
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM         9865
    PAGE 218  ROUTINE REQUEST.ILLUM               24
    PAGE 222  ROUTINE REQUEST.SMOKE              212
    PAGE 226  ROUTINE REQUEST.WD.FASCAM          432
    PAGE 227                                     483
    PAGE 342  ROUTINE RANGE.COMPUTE              5662
    PAGE 478  PROCESS WITH.DRAW                  1962
    PAGE 479                                     1993
    PAGE 495  PROCESS SHOOT.OUT                  2899
    PAGE 498                                     3090
    PAGE 506  PROCESS CAS.MISSION                3499
    PAGE 656  ROUTINE AR.DETECTION               9398
READ.ORDERS
    PAGE 520  ROUTINE MAIN2                      4221 4223
    PAGE 539  ROUTINE READ.ORDERS                4983
REIN.ARRIVE
    PAGE  88  ROUTINE END.MOVE                   4370
    PAGE 118  ROUTINE REIN.ARRIVE                5597
REMOTE.PILOT.VEHICLE
    PAGE  37  ''SECTION FOR PROCESSES            2114
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE       1356 1373
    PAGE 474  PROCESS TARGET.REPORT              1742
    PAGE 485  PROCESS FIRE.MISSION               2352
    PAGE 600  ROUTINE BETWEEN.ROUTINE            7508 7509
    PAGE 616  ROUTINE SNAP2                      8156
    PAGE 696  ''PROGRAM  OLDER.VERSION           1552
REM.EFFECTS.COMPUTATION
    PAGE 185  ROUTINE FA.BN.ASGN                 8510
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION    9791 9806
REPLACE.HC
    PAGE 304  ROUTINE REPLACE.HC                 4187 4209 4229
    PAGE 452  PROCESS HC.RETURN.FARRP             582
    PAGE 453                                      643
REQUEST.DEF.FASCAM
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM         9828
    PAGE 370  EVENT ENGAGEMENT                   6755
REQUEST.FASCAM
    PAGE   3  PROGRAM REVISIONS                   157
    PAGE 103  ROUTINE MINE.DELAY                 4986
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM         9873
    PAGE 216  ROUTINE REQUEST.FASCAM             9885
    PAGE 226  ROUTINE REQUEST.WD.FASCAM           446
    PAGE 227                                      501
REQUEST.ILLUM
    PAGE   3  PROGRAM REVISIONS                   124  137  145
    PAGE 200  ROUTINE ILLUM.EFFECTS              9215
    PAGE 218  ROUTINE REQUEST.ILLUM              9981
    PAGE 370  EVENT ENGAGEMENT                   6760
REQUEST.SMOKE
    PAGE   3  PROGRAM REVISIONS                   153
    PAGE 222  ROUTINE REQUEST.SMOKE               172
    PAGE 234  ROUTINE SMOKE.EFFECTS               779
    PAGE 370  EVENT ENGAGEMENT                   6766
    PAGE 478  PROCESS WITH.DRAW                  1944
```

```
REQUEST.WD.FASCAM
    PAGE 226  ROUTINE REQUEST.WD.FASCAM              391
    PAGE 478  PROCESS WITH.DRAW                     1941
RESET.FEBA.SECTOR
    PAGE  62  ROUTINE FEBA.INITIAL                  3247
    PAGE 119  ROUTINE RESET.FEBA.SECTOR             5647 5658
    PAGE 331  ROUTINE ENQ.FEBA.SET                  5338
REVISIONS
    PAGE   2  PROGRAM REVISIONS                       62
RPV.DETECTION
    PAGE 229  ROUTINE RPV.DETECTION                  512  527
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1448
RUL.EN.INPUT
    PAGE 521  ROUTINE MAIN2                         4245 4247
    PAGE 560  ROUTINE  RUL.EN.INPUT                 5820
SCHEDULE.ARTY.MOVEMENT
    PAGE  40  ''SECTION FOR EVENTS                  2301
    PAGE  57  ROUTINE MAIN3                         3076
    PAGE 390  EVENT SCHEDULE.ARTY.MOVEMENT          7579 7584
    PAGE 602  ROUTINE BETWEEN.ROUTINE               7624 7625
    PAGE 616  ROUTINE SNAP2                         8186
    PAGE 699  ''PROGRAM  OLDER.VERSION              1738
SEARCH
    PAGE 120  ROUTINE SEARCH                        5705
    PAGE 123  ROUTINE TIME.TO.DETECT                5831
    PAGE 343  ROUTINE SEARCH.COVERAGE               5708 5717
    PAGE 347  EVENT ACT.REINF                       5811
    PAGE 428  PROCESS AIR.OBSERVER                  9338
    PAGE 431                                        9512 9536
    PAGE 438  PROCESS FORWARD.OBSERVER              9874
    PAGE 440                                        9984 9990
    PAGE 441                                          72   79
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION         923
    PAGE 459                                        1030
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1361
    PAGE 468                                        1447
    PAGE 485  PROCESS FIRE.MISSION                  2347 2357
    PAGE 653  PROCESS AIRBORNE.RADAR                9268
    PAGE 654                                        9352
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9479 9514
SEARCH.COVERAGE
    PAGE 343  ROUTINE SEARCH.COVERAGE               5676
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1426
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9536
SEGMENT.ADJUST
    PAGE  83  ROUTINE CHANGE.LOC                    4173
    PAGE  84                                        4240
    PAGE 110  ROUTINE POSITION                      5281
    PAGE 122  ROUTINE SEGMENT.ADJUST                5773
SEND.TEAM
    PAGE  40  ''SECTION FOR EVENTS                  2303
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS            3562
    PAGE 291                                        3605
    PAGE 292                                        3660
    PAGE 337  ROUTINE HC.EMPTY                      5485 5486 5490 5491 5492 5494 5497 5498
    PAGE 391  EVENT SEND.TEAM                       7596 7612
```

```
    PAGE 453  PROCESS HC.RETURN.FARRP                    664  678  684
    PAGE 602  ROUTINE BETWEEN.ROUTINE                   7628 7629
    PAGE 616  ROUTINE SNAP2                             8188
    PAGE 699  ''PROGRAM  OLDER.VERSION                  1740
SENSOR.INPUT
    PAGE 521  ROUTINE MAIN2                             4263 4265
    PAGE 564  ROUTINE MAO.INPUT                         5958
    PAGE 565  ROUTINE SENSOR.INPUT                      6011
SET.DEBUG
    PAGE  41  ''SECTION FOR EVENTS                      2342 2343
    PAGE 392  EVENT SET.DEBUG                           7623 7630
    PAGE 602  ROUTINE BETWEEN.ROUTINE                   7656 7657
    PAGE 617  ROUTINE SNAP2                             8197
    PAGE 700  ''PROGRAM  OLDER.VERSION                  1776 1777
SHOOT.OUT
    PAGE   2  PROGRAM REVISIONS                          84
    PAGE   3                                            127  133
    PAGE  37  ''SECTION FOR PROCESSES                   2130
    PAGE 107  ROUTINE NEW.SEGMENT                       5111
    PAGE 108                                            5215 5216
    PAGE 109                                            5218 5232 5233 5235
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                   2939 2940
    PAGE 282  ROUTINE AC.DF.EFFECTS                     3123 3124
    PAGE 294  ROUTINE END.CAS.MISSION                   3741 3744
    PAGE 301  ROUTINE HC.DISENGAGE                      4110 4111 4112 4114
    PAGE 329  ROUTINE EMPTY                             5260 5261 5262 5263 5264
    PAGE 369  EVENT ENGAGEMENT                          6733
    PAGE 379  EVENT HELO.ENGAGEMENT                     7213
    PAGE 383  EVENT OFF.LINE.ATTRITION                  7381 7383
    PAGE 398  EVENT START.BATTLE                        7970
    PAGE 419  PROCESS AC.ATK.TGT                        8891
    PAGE 423                                            9090 9107 9108 9122
    PAGE 425                                            9198 9214 9215 9229
    PAGE 426                                            9273
    PAGE 435  PROCESS ARTY.ASSESS                       9719 9720 9721 9722 9731 9738 9740
    PAGE 436                                            9756 9757 9758 9779 9786 9799
    PAGE 437                                            9804 9805 9806
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION            1092 1093 1094 1096
    PAGE 464  PROCESS MINE.ASSESS                       1262
    PAGE 465                                            1299 1302
    PAGE 487  PROCESS ASSESSMENT                        2447
    PAGE 488                                            2513 2514 2517 2519
    PAGE 491                                            2690 2695 2696 2697 2698 2699
    PAGE 493  PROCESS SHOOT.OUT                         2749 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766
                                                        2767 2768 2769 2773 2775 2776 2777 2790 2791 2795 2796 2802 2803
    PAGE 494                                            2809 2825 2832 2851 2855 2862
    PAGE 495                                            2866 2878 2880 2889 2890 2894 2922
    PAGE 496                                            2923 2925 2957 2959 2962 2963 2969 2971
    PAGE 497                                            2981 2983 2998 3000 3010 3011 3026 3030 3036 3037
    PAGE 498                                            3040 3069 3071 3078 3081 3082 3085
    PAGE 499                                            3111 3113 3114 3125 3141 3149
    PAGE 500                                            3157 3160 3167 3168 3174 3182 3190 3201 3212
    PAGE 501                                            3213 3215 3223 3235 3251
    PAGE 502                                            3279 3281 3289 3296 3304 3312 3315 3322 3323
    PAGE 503                                            3329 3331 3344 3346
    PAGE 507  PROCESS CAS.MISSION                       3581
```

```
    PAGE 508                                3582 3583 3584
    PAGE 516  PROCESS HELICOPTER.FIRE       4051 4052
    PAGE 600  ROUTINE BETWEEN.ROUTINE       7512 7513
    PAGE 616  ROUTINE SNAP2                 8158
    PAGE 619  ROUTINE SNAP.R                8273
    PAGE 696  ''PROGRAM  OLDER.VERSION      1568
SIZE.ESTIMATE
    PAGE 231  ROUTINE SIZE.ESTIMATE          623
    PAGE 240  ROUTINE TARGET.ANALYSIS       1063
    PAGE 631  FUNCTION COMBINATIONS         8718
SMOKE.COMPUTATION
    PAGE 185  ROUTINE FA.BN.ASGN            8469
    PAGE 233  ROUTINE SMOKE.COMPUTATION      687
SMOKE.EFFECTS
    PAGE 222  ROUTINE REQUEST.SMOKE          179
    PAGE 234  ROUTINE SMOKE.EFFECTS          738
    PAGE 484  PROCESS FIRE.MISSION          2300
SMOKE.INPUT
    PAGE 521  ROUTINE MAIN2                 4287 4289
    PAGE 580  ROUTINE SMOKE.INPUT           6668
SNAP2
    PAGE 614  ROUTINE SNAP2                 8020 8023
    PAGE 618  ROUTINE SNAP.R                8209
SNAP.R
    PAGE 604  ROUTINE ERROR.STOP            7675
    PAGE 618  ROUTINE SNAP.R                8201
START.ARTY.MOVEMENT
    PAGE  40  ''SECTION FOR EVENTS          2307
    PAGE  90  ROUTINE FA.BN.MOVEMENT        4465 4485
    PAGE  91                                4544 4558
    PAGE  92                                4575
    PAGE 393  EVENT START.ARTY.MOVEMENT     7655
    PAGE 602  ROUTINE BETWEEN.ROUTINE       7632 7633
    PAGE 616  ROUTINE SNAP2                 8189
    PAGE 700  ''PROGRAM  OLDER.VERSION      1744
START.BATTLE
    PAGE  41  ''SECTION FOR EVENTS          2311
    PAGE 147  ROUTINE INTER.BATTLE          6851
    PAGE 148                                6888
    PAGE 395  EVENT START.BATTLE            7743
    PAGE 401                                8103 8127
    PAGE 412  EVENT ACT.ATK                 8610
    PAGE 413                                8703
    PAGE 602  ROUTINE BETWEEN.ROUTINE       7636 7637
    PAGE 616  ROUTINE SNAP2                 8193
    PAGE 700  ''PROGRAM  OLDER.VERSION      1748
START.MOVE
    PAGE  41  ''SECTION FOR EVENTS          2319
    PAGE  95  ROUTINE INIT.REINF            4643 4660
    PAGE 112  ROUTINE PREPARE.LIST          5396 5397
    PAGE 403  EVENT START.MOVE              8177
    PAGE 405                                8319
    PAGE 415  EVENT ACT.MOVCOR              8756
    PAGE 416  EVENT ACT.MOVDIS              8774 8793
    PAGE 602  ROUTINE BETWEEN.ROUTINE       7640 7641
    PAGE 617  ROUTINE SNAP2                 8194
```

```
    PAGE 700  ''PROGRAM  OLDER.VERSION                1756
STAY.TIME
    PAGE  50  ''SECTION FOR DEFINITIONS               2859
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                 9518 9519
    PAGE 660  FUNCTION STAY.TIME                      9590
    PAGE 709  ''PROGRAM  OLDER.VERSION                2278
STOP.ARTY.MOVEMENT
    PAGE  41  ''SECTION FOR EVENTS                    2326
    PAGE 394  EVENT START.ARTY.MOVEMENT               7733
    PAGE 407  EVENT STOP.ARTY.MOVEMENT                8370
    PAGE 602  ROUTINE BETWEEN.ROUTINE                 7644 7645
    PAGE 616  ROUTINE SNAP2                           8191
    PAGE 700  ''PROGRAM  OLDER.VERSION                1763
ST.INPUT
    PAGE 521  ROUTINE MAIN2                           4248 4250
    PAGE 561  ROUTINE ST.INPUT                        5855
SUBM.INPUT
    PAGE 520  ROUTINE MAIN2                           4239
    PAGE 521                                          4241
    PAGE 557  ROUTINE SUBM.INPUT                      5713
SUBSTITUTIONS
    PAGE  50  ''SECTION FOR SUBSTITUTIONS             2861 2864
    PAGE  52                                          2966
    PAGE 709  ''PROGRAM  OLDER.VERSION                2280 2282
    PAGE 710                                          2373
SWITCH.FO
    PAGE   2  PROGRAM REVISIONS                         80
    PAGE 146  ROUTINE INTER.BATTLE                    6790
    PAGE 238  ROUTINE SWITCH.FO                        913
    PAGE 356  EVENT BTL.ENDED                         6267
    PAGE 400  EVENT START.BATTLE                      8081
SYS.INPUT
    PAGE 520  ROUTINE MAIN2                           4188 4195
    PAGE 523  ROUTINE SYS.INPUT                       4317 4332
TACAIR.DATA.REPORT
    PAGE 522  ROUTINE MAIN2                           4306
    PAGE 621  ROUTINE TACAIR.DATA.REPORT              8333
TACAIR.INPUT
    PAGE   3  PROGRAM REVISIONS                        134
    PAGE 521  ROUTINE MAIN2                           4293 4297
    PAGE 582  ROUTINE TACAIR.INPUT                    6736
TARGET.ANALYSIS
    PAGE   3  PROGRAM REVISIONS                        150  151
    PAGE   4                                           181
    PAGE 180  ROUTINE EST.MIL.WORTH                   8223
    PAGE 231  ROUTINE SIZE.ESTIMATE                    629
    PAGE 239  ROUTINE TARGET.ANALYSIS                  971  997
    PAGE 470  PROCESS TARGET.REPORT                   1540
    PAGE 471                                          1563
    PAGE 473                                          1709
    PAGE 632  FUNCTION EST.RANGE                      8750
TARGET.REPORT
    PAGE   3  PROGRAM REVISIONS                        149
    PAGE   4                                           208
    PAGE  35  ''SECTION FOR PROCESSES                 1995
    PAGE  38                                          2147
```

```
    PAGE 160  ROUTINE ATTRIT.SENSOR                7386
    PAGE 182  ROUTINE FA.BN.ASGN                   8319
    PAGE 217  ROUTINE REQUEST.FASCAM               9957
    PAGE 220  ROUTINE REQUEST.ILLUM                 136
    PAGE 225  ROUTINE REQUEST.SMOKE                 354
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS       2769
    PAGE 332  ROUTINE FDC.TR.DEQ                   5354 5355
    PAGE 334  ROUTINE FINISH.COMPUTATION           5384
    PAGE 362  EVENT CFR.OPERATOR                   6503
    PAGE 363                                       6580 6582
    PAGE 387  EVENT PDB.OPERATOR                   7522 7558 7560
    PAGE 431  PROCESS AIR.OBSERVER                 9553 9558
    PAGE 432                                       9581 9585
    PAGE 440  PROCESS FORWARD.OBSERVER                1
    PAGE 441                                         48   63
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE         1388
    PAGE 468                                       1457 1462
    PAGE 470  PROCESS TARGET.REPORT                1482 1526
    PAGE 475                                       1782
    PAGE 476                                       1832 1858
    PAGE 486  PROCESS FIRE.MISSION                 2394
    PAGE 600  ROUTINE BETWEEN.ROUTINE              7516 7517
    PAGE 616  ROUTINE SNAP2                        8159
    PAGE 619  ROUTINE SNAP.R                       8280
    PAGE 636  FUNCTION HE.WLA                      8894
    PAGE 653  PROCESS AIRBORNE.RADAR               9306
    PAGE 654                                       9370
    PAGE 659  PROCESS PHOTO.IR.FLIGHT              9556 9585 9586
    PAGE 694  ''PROGRAM  OLDER.VERSION             1433
    PAGE 697                                       1585
TBF.INPUT
    PAGE 521  ROUTINE MAIN2                        4266 4268
    PAGE 569  ROUTINE TBF.INPUT                    6202
TB.INPUT
    PAGE 520  ROUTINE MAIN2                        4227 4229
    PAGE 548  ROUTINE TB.INPUT                     5307
TEMPERATURE.ATTENUATION
    PAGE 120  ROUTINE SEARCH                       5735
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION      5732
TEMPORARY_ENTITIES
    PAGE  19  ''SECTION FOR TEMPORARY_ENTITIES     1084
TERM.CHECK
    PAGE  85  ROUTINE CHANGE.LOC                   4269
    PAGE 345  ROUTINE TERM.CHECK                   5765
TIME.REQ
    PAGE 171  ROUTINE COMPARE.TRS                  7816
    PAGE 663  ROUTINE TIME.REQ                     9640
TIME.TO.DETECT
    PAGE  93  ROUTINE INITIAL.DETECT               4605
    PAGE 123  ROUTINE TIME.TO.DETECT               5794
    PAGE 132  ROUTINE CHECK.ENGAGEMENT             6146
    PAGE 201  ROUTINE ILLUM.EFFECTS                9282
TR.INPUT
    PAGE 470  PROCESS TARGET.REPORT                1509
    PAGE 590  ROUTINE TR.INPUT                     7073
```

```
TT.FACTORS.INPUT
    PAGE 521  ROUTINE MAIN2                          4272 4274
    PAGE 571  ROUTINE TT.FACTORS.INPUT               6282
TYPE.WEAPON.INPUT
    PAGE 520  ROUTINE MAIN2                          4208 4210
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT              4638
UNIT.ASSIGNMENT
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                3823
    PAGE 398  EVENT START.BATTLE                     7922 7930
UNIT.ENVIR
    PAGE 154  ROUTINE AO.DETECTION                   7118
    PAGE 212  ROUTINE PIR.DETECTION                  9725 9728
    PAGE 229  ROUTINE RPV.DETECTION                   538  541
    PAGE 242  ROUTINE UNIT.ENVIR                     1111
    PAGE 261  ROUTINE BTRY.EFFECTS                   2040
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2822
UNIT.INPUT
    PAGE 520  ROUTINE MAIN2                          4215 4217
    PAGE 533  ROUTINE UNIT.INPUT                     4680
UNIT.PRIORITY
    PAGE 305  ROUTINE UNIT.PRIORITY                  4234 4260 4291
    PAGE 396  EVENT START.BATTLE                     7843 7850
UPDATE.LOC
    PAGE  41  ''SECTION FOR EVENTS                   2330
    PAGE 112  ROUTINE PREPARE.LIST                   5389 5390
    PAGE 406  EVENT START.MOVE                       8356
    PAGE 408  EVENT UPDATE.LOC                       8404
    PAGE 410                                         8575
    PAGE 411                                         8585
    PAGE 602  ROUTINE BETWEEN.ROUTINE                7648 7649
    PAGE 617  ROUTINE SNAP2                          8195
    PAGE 630  FUNCTION COLLISION                     8684 8685 8688 8689
    PAGE 700  ''PROGRAM  OLDER.VERSION               1767
VIS.INPUT
    PAGE 521  ROUTINE MAIN2                          4290 4292
    PAGE 581  ROUTINE VIS.INPUT                      6711
VOLLEY
    PAGE 246  ROUTINE VOLLEY                         1307
    PAGE 263  ROUTINE BTRY.EFFECTS                   2150 2151
    PAGE 267                                         2385
    PAGE 272                                         2715
    PAGE 481  PROCESS FIRE.MISSION                   2115
    PAGE 483                                         2224 2252
    PAGE 636  FUNCTION HE.WLA                        8864
    PAGE 638  FUNCTION ICM.WLA                       8964
WEIBULL.F
    PAGE  97  ROUTINE LINE.OF.SIGHT                  4738 4739 4757 4758
    PAGE 109  ROUTINE NEW.SEGMENT                    5256 5257
    PAGE 210  ROUTINE PGM.MSN.ASGN                   9618 9619
    PAGE 479  PROCESS WITH.DRAW                      1983 1984 2006 2007
    PAGE 642  ROUTINE WEIBULL.F                      9062
WEIGHTED.VOLLEYS
    PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION          9061 9081
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS               1366
WHAT.NEXT
    PAGE 125  ROUTINE WHAT.NEXT                      5873
```

```
    PAGE 356  EVENT BTL.ENDED                    6258
    PAGE 357                                     6333 6336
WITH.DRAW
    PAGE  39  ''SECTION FOR PROCESSES            2195
    PAGE 115  ROUTINE PROX.CHECK                 5553
    PAGE 133  ROUTINE CHECK.FORCE                6195
    PAGE 324  ROUTINE DECIDE                     5061
    PAGE 328  ROUTINE EMPTY                      5208 5209 5211 5212 5213 5214
    PAGE 478  PROCESS WITH.DRAW                  1914 1920 1926
    PAGE 600  ROUTINE BETWEEN.ROUTINE            7520 7521
    PAGE 616  ROUTINE SNAP2                      8160
    PAGE 698  ''PROGRAM  OLDER.VERSION           1633
```

```
AATT.CUM.AREA
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1091
    PAGE 280  ROUTINE AC.DF.EFFECTS                     3023 3035
    PAGE 679  ''PROGRAM  OLDER.VERSION                   532
AATT.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1093
    PAGE  33  ''SECTION FOR PROCESSES                   1900
    PAGE 280  ROUTINE AC.DF.EFFECTS                     3018 3034
    PAGE 420  PROCESS AC.ATK.TGT                        8930
    PAGE 427                                            9311 9313
    PAGE 679  ''PROGRAM  OLDER.VERSION                   534
    PAGE 693                                            1338
AATT.UE.LINK
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1090
    PAGE 280  ROUTINE AC.DF.EFFECTS                     3020 3021 3045
    PAGE 420  PROCESS AC.ATK.TGT                        8931
    PAGE 679  ''PROGRAM  OLDER.VERSION                   531
AAT.AC.KILLED.IND
    PAGE  33  ''SECTION FOR PROCESSES                   1897
    PAGE 419  PROCESS AC.ATK.TGT                        8856
    PAGE 490  PROCESS ASSESSMENT                        2630
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1335
AAT.AD.UNIT
    PAGE  33  ''SECTION FOR PROCESSES                   1894
    PAGE  43  ''SECTION FOR DEFINITIONS                 2447
    PAGE 419  PROCESS AC.ATK.TGT                        8853
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1332
    PAGE 702                                            1867
AAT.CMSN
    PAGE  33  ''SECTION FOR PROCESSES                   1893
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                   2807 2810
    PAGE 280  ROUTINE AC.DF.EFFECTS                     3044 3047 3048
    PAGE 281                                            3106
    PAGE 282                                            3155
    PAGE 307  ROUTINE AD.SHOOT                          4326 4327 4347
    PAGE 419  PROCESS AC.ATK.TGT                        8852
    PAGE 427                                            9318
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1331
AAT.EX.AATGT
    PAGE  33  ''SECTION FOR PROCESSES                   1898
    PAGE 419  PROCESS AC.ATK.TGT                        8860 8861 8862 8863 8864 8865 8866 8867 8868 8869 8870 8871 8872 8873 8874
                                                        8875 8876 8877 8878 8879 8897
    PAGE 427                                            9327
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1336
AAT.RANGE
    PAGE  33  ''SECTION FOR PROCESSES                   1895
    PAGE  43  ''SECTION FOR DEFINITIONS                 2448
    PAGE 281  ROUTINE AC.DF.EFFECTS                     3064 3065 3071
    PAGE 419  PROCESS AC.ATK.TGT                        8854
    PAGE 495  PROCESS SHOOT.OUT                         2887
    PAGE 498                                            3075
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1333
    PAGE 702                                            1868
AAT.TGT
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1089
    PAGE 420  PROCESS AC.ATK.TGT                        8929 8930 8931
```

```
    PAGE 427                                    9314
    PAGE 679  ''PROGRAM  OLDER.VERSION           530
AAT.VIS.IND
    PAGE  33  ''SECTION FOR PROCESSES           1896
    PAGE 419  PROCESS AC.ATK.TGT                8855
    PAGE 490  PROCESS ASSESSMENT                2629
    PAGE 494  PROCESS SHOOT.OUT                 2823 2834
    PAGE 692  ''PROGRAM  OLDER.VERSION          1334
AA.DESTRUCT.INDIC
    PAGE  34  ''SECTION FOR PROCESSES           1944
    PAGE 262  ROUTINE BTRY.EFFECTS              2080
    PAGE 328  ROUTINE EMPTY                     5221
    PAGE 435  PROCESS ARTY.ASSESS               9694
    PAGE 693  ''PROGRAM  OLDER.VERSION          1382
AA.FIRE.MISSION
    PAGE  34  ''SECTION FOR PROCESSES           1945
    PAGE 262  ROUTINE BTRY.EFFECTS              2082
    PAGE 435  PROCESS ARTY.ASSESS               9696
    PAGE 693  ''PROGRAM  OLDER.VERSION          1383
AA.FRACTION
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES  1115
    PAGE  47  ''SECTION FOR DEFINITIONS         2710
    PAGE 269  ROUTINE BTRY.EFFECTS              2535
    PAGE 435  PROCESS ARTY.ASSESS               9734
    PAGE 436                                    9764 9776
    PAGE 437                                    9812
    PAGE 679  ''PROGRAM  OLDER.VERSION           554
    PAGE 706                                    2133
AA.LINK
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES  1113
    PAGE 269  ROUTINE BTRY.EFFECTS              2533
    PAGE 435  PROCESS ARTY.ASSESS               9710 9711
    PAGE 437                                    9849
    PAGE 679  ''PROGRAM  OLDER.VERSION           552
    PAGE 712                                    2481
AA.SET
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES  1117
    PAGE  34  ''SECTION FOR PROCESSES           1948
    PAGE 269  ROUTINE BTRY.EFFECTS              2536
    PAGE 435  PROCESS ARTY.ASSESS               9703 9710
    PAGE 679  ''PROGRAM  OLDER.VERSION           556
    PAGE 693                                    1386
AA.UE.LINK
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES  1114
    PAGE 269  ROUTINE BTRY.EFFECTS              2534
    PAGE 435  PROCESS ARTY.ASSESS               9712 9716 9717 9720 9722 9732
    PAGE 436                                    9744 9752 9766 9776 9780 9788 9796
    PAGE 437                                    9826 9833 9843
    PAGE 679  ''PROGRAM  OLDER.VERSION           553
AA.UNIT
    PAGE  34  ''SECTION FOR PROCESSES           1946
    PAGE 262  ROUTINE BTRY.EFFECTS              2081
    PAGE 328  ROUTINE EMPTY                     5218
    PAGE 435  PROCESS ARTY.ASSESS               9700
    PAGE 693  ''PROGRAM  OLDER.VERSION          1384
```

```
ABS.F
    PAGE 104  ROUTINE MINE.DELAY                    4994
    PAGE 106  ROUTINE MIN.MOVE                      5077
    PAGE 150  ROUTINE PK.COMPUTE                    6982
    PAGE 154  ROUTINE AO.DETECTION                  7106
    PAGE 155                                        7133
    PAGE 174  ROUTINE DUST.EFFECTS                  7949 7952
    PAGE 179  ROUTINE EST.COVERAGE                  8194 8201
    PAGE 191  ROUTINE FINAL.COVERAGE                8759 8760
    PAGE 192                                        8833
    PAGE 193                                        8843
    PAGE 315  ROUTINE FLIGHT.PATH                   4731 4732
    PAGE 349  EVENT AD.ENGAGEMENT                   5945 5946 5947 5948
    PAGE 352                                        6088
    PAGE 353                                        6148 6149 6150 6151
    PAGE 394  EVENT START.ARTY.MOVEMENT             7732
    PAGE 429  PROCESS AIR.OBSERVER                  9434 9437
    PAGE 430                                        9462 9465
    PAGE 431                                        9511
    PAGE 432                                        9596
    PAGE 566  ROUTINE SENSOR.INPUT                  6111
    PAGE 654  PROCESS AIRBORNE.RADAR                9327
    PAGE 660  FUNCTION STAY.TIME                    9606
AB.FARRP
    PAGE  36  ''SECTION FOR PROCESSES               2055
    PAGE 337  ROUTINE HC.EMPTY                      5501 5508
    PAGE 402  EVENT START.BATTLE                    8160
    PAGE 695  ''PROGRAM  OLDER.VERSION              1493
AB.TEAM
    PAGE  36  ''SECTION FOR PROCESSES               2056
    PAGE 337  ROUTINE HC.EMPTY                      5503 5504 5505 5509
    PAGE 338                                        5515 5517
    PAGE 402  EVENT START.BATTLE                    8161
    PAGE 695  ''PROGRAM  OLDER.VERSION              1494
ACQUIRED.QUANT
    PAGE 213  ROUTINE PIR.DETECTION                 9766 9770 9773 9786
    PAGE 230  ROUTINE RPV.DETECTION                  579  583  586  617
ACTIVATED.
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE             6642
ACTIVITY.TYPE
    PAGE 386  EVENT PDB.ACTIVATION                  7469 7475 7496
ACT.ANGLE.P1.P2
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       284
    PAGE  47  ''SECTION FOR DEFINITIONS             2713
    PAGE 507  PROCESS CAS.MISSION                   3548
    PAGE 584  ROUTINE TACAIR.INPUT                  6867 6869 6874 6876 6879
    PAGE 665  ''PROGRAM  OLDER.VERSION              9726
    PAGE 706                                        2135
ACT.ATK
    PAGE  39  ''SECTION FOR EVENTS                  2204
    PAGE  59  ROUTINE CREATE.FORCE                  3150 3151
    PAGE 373  EVENT GET.NX.ORD                      6926 6932
    PAGE 412  EVENT ACT.ATK                         8599
    PAGE 600  ROUTINE BETWEEN.ROUTINE               7532 7533
    PAGE 616  ROUTINE SNAP2                         8163
    PAGE 698  ''PROGRAM  OLDER.VERSION              1641
```

```
ACT.BAI.TA.DELAY
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES          273
    PAGE 421  PROCESS AC.ATK.TGT                       8964
    PAGE 505  PROCESS CAS.MISSION                      3458
    PAGE 583  ROUTINE TACAIR.INPUT                     6823
    PAGE 622  ROUTINE TACAIR.DATA.REPORT               8421
    PAGE 665  ''PROGRAM  OLDER.VERSION                 9715
ACT.BATTLE.RANGE
    PAGE  42  ''SECTION FOR DEFINITIONS                2381
    PAGE 405  EVENT START.MOVE                         8331
    PAGE 409  EVENT UPDATE.LOC                         8493
    PAGE 523  ROUTINE SYS.INPUT                        4371
    PAGE 524                                           4376 4376
    PAGE 701  ''PROGRAM  OLDER.VERSION                 1811
ACT.DEF
    PAGE  39  ''SECTION FOR EVENTS                     2209
    PAGE 373  EVENT GET.NX.ORD                         6912
    PAGE 414  EVENT ACT.DEF                            8714
    PAGE 540  ROUTINE READ.ORDERS                      5059
    PAGE 600  ROUTINE BETWEEN.ROUTINE                  7536 7537
    PAGE 616  ROUTINE SNAP2                            8164
    PAGE 698  ''PROGRAM  OLDER.VERSION                 1646
ACT.EQUIP.ID
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES          268
    PAGE  43  ''SECTION FOR DEFINITIONS                2449
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                  2811
    PAGE 280  ROUTINE AC.DF.EFFECTS                    3047
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS            3298 3310 3320 3328 3335
    PAGE 286                                           3349 3357 3368 3385
    PAGE 287                                           3404
    PAGE 295  ROUTINE END.CAS.MISSION                  3797 3828
    PAGE 307  ROUTINE AD.SHOOT                         4344 4347
    PAGE 420  PROCESS AC.ATK.TGT                       8950
    PAGE 421                                           9003 9012
    PAGE 504  PROCESS CAS.MISSION                      3375
    PAGE 583  ROUTINE TACAIR.INPUT                     6806 6851
    PAGE 584                                           6903
    PAGE 585                                           6942
    PAGE 622  ROUTINE TACAIR.DATA.REPORT               8413 8416
    PAGE 623                                           8451
    PAGE 664  ''PROGRAM  OLDER.VERSION                 9710
    PAGE 702                                           1869
ACT.MAX.ALOFT
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES          276
    PAGE  43  ''SECTION FOR DEFINITIONS                2456
    PAGE 421  PROCESS AC.ATK.TGT                       8976
    PAGE 505  PROCESS CAS.MISSION                      3460
    PAGE 583  ROUTINE TACAIR.INPUT                     6824
    PAGE 622  ROUTINE TACAIR.DATA.REPORT               8422
    PAGE 665  ''PROGRAM  OLDER.VERSION                 9718
    PAGE 702                                           1876
ACT.MAX.PREP.TIME
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES          280
    PAGE 504  PROCESS CAS.MISSION                      3398
    PAGE 583  ROUTINE TACAIR.INPUT                     6827
    PAGE 585                                           6952
```

```
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8425
    PAGE 665  ''PROGRAM  OLDER.VERSION                9722
ACT.MIN.ALT
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES         271
    PAGE  43  ''SECTION FOR DEFINITIONS               2452
    PAGE 317  ROUTINE FLIGHT.PATH                     4826
    PAGE 583  ROUTINE TACAIR.INPUT                    6829
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8419
    PAGE 665  ''PROGRAM  OLDER.VERSION                9713
    PAGE 702                                          1872
ACT.MIN.PREP.TIME
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         279
    PAGE 504  PROCESS CAS.MISSION                     3397
    PAGE 583  ROUTINE TACAIR.INPUT                    6826
    PAGE 585                                          6951
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8424
    PAGE 665  ''PROGRAM  OLDER.VERSION                9721
ACT.MOVCOR
    PAGE  39  ''SECTION FOR EVENTS                    2213
    PAGE  58  ROUTINE CREATE.FORCE                    3144 3145
    PAGE 374  EVENT GET.NX.ORD                        6948 6953
    PAGE 415  EVENT ACT.MOVCOR                        8736
    PAGE 546  ROUTINE ORD.MOVCOR                      5246
    PAGE 600  ROUTINE BETWEEN.ROUTINE                 7540 7541
    PAGE 616  ROUTINE SNAP2                           8165
    PAGE 698  ''PROGRAM  OLDER.VERSION                1650
ACT.MOVDIS
    PAGE  39  ''SECTION FOR EVENTS                    2217
    PAGE 144  ROUTINE DEAD.UNIT                       6686
    PAGE 374  EVENT GET.NX.ORD                        6964
    PAGE 416  EVENT ACT.MOVDIS                        8766
    PAGE 601  ROUTINE BETWEEN.ROUTINE                 7544 7545
    PAGE 616  ROUTINE SNAP2                           8166
    PAGE 698  ''PROGRAM  OLDER.VERSION                1654
ACT.NITE.FLY
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         278
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS           3421
    PAGE 583  ROUTINE TACAIR.INPUT                    6828
    PAGE 665  ''PROGRAM  OLDER.VERSION                9720
ACT.NORM.ALT
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES         272
    PAGE  43  ''SECTION FOR DEFINITIONS               2453
    PAGE 317  ROUTINE FLIGHT.PATH                     4828
    PAGE 583  ROUTINE TACAIR.INPUT                    6830
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8420
    PAGE 665  ''PROGRAM  OLDER.VERSION                9714
    PAGE 702                                          1873
ACT.PASS.TIME
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES         275
    PAGE  43  ''SECTION FOR DEFINITIONS               2455
    PAGE 421  PROCESS AC.ATK.TGT                      8964 8977
    PAGE 505  PROCESS CAS.MISSION                     3451 3457 3458
    PAGE 507                                          3554
    PAGE 584  ROUTINE TACAIR.INPUT                    6894
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8427
    PAGE 665  ''PROGRAM  OLDER.VERSION                9717
```

```
    PAGE 702                                        1875
ACT.PROB.SORTIE.ABORT
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       277
    PAGE  43  ''SECTION FOR DEFINITIONS             2457
    PAGE 504  PROCESS CAS.MISSION                   3402
    PAGE 583  ROUTINE TACAIR.INPUT                  6825
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8423
    PAGE 665  ''PROGRAM  OLDER.VERSION              9719
    PAGE 702                                        1877
ACT.P1.DIST
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       281
    PAGE  43  ''SECTION FOR DEFINITIONS             2458
    PAGE 420  PROCESS AC.ATK.TGT                    8913
    PAGE 422                                        9029 9030 9077
    PAGE 584  ROUTINE TACAIR.INPUT                  6855
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8428
    PAGE 665  ''PROGRAM  OLDER.VERSION              9723
    PAGE 702                                        1878
ACT.P2.DIST
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       282
    PAGE  43  ''SECTION FOR DEFINITIONS             2459
    PAGE 420  PROCESS AC.ATK.TGT                    8913
    PAGE 422                                        9028
    PAGE 584  ROUTINE TACAIR.INPUT                  6857
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8429
    PAGE 665  ''PROGRAM  OLDER.VERSION              9724
    PAGE 702                                        1879
ACT.P3.DIST
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       283
    PAGE  43  ''SECTION FOR DEFINITIONS             2460
    PAGE 424  PROCESS AC.ATK.TGT                    9159
    PAGE 584  ROUTINE TACAIR.INPUT                  6859
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8430
    PAGE 665  ''PROGRAM  OLDER.VERSION              9725
    PAGE 702                                        1880
ACT.RANGE
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1067
    PAGE  50  ''SECTION FOR DEFINITIONS             2848
    PAGE 138  ROUTINE CHECK.PROX                    6425 6426
    PAGE 139                                        6442 6443
    PAGE 205  ROUTINE NOISE.DEGRADE                 9398 9399 9410 9411
    PAGE 246  ROUTINE VOLLEY                        1341 1342
    PAGE 358  EVENT CFR.ACTIVATION                  6367 6368
    PAGE 360  EVENT CFR.ON                          6452 6453
    PAGE 386  EVENT PDB.ACTIVATION                  7493 7494
    PAGE 439  PROCESS FORWARD.OBSERVER              9921 9923 9939 9941
    PAGE 628  FUNCTION ACT.RANGE                    8630
    PAGE 660  FUNCTION STAY.TIME                    9605 9606
    PAGE 678  ''PROGRAM  OLDER.VERSION               509
    PAGE 709                                        2267
ACT.REINF
    PAGE  39  ''SECTION FOR EVENTS                  2221
    PAGE 347  EVENT ACT.REINF                       5803
    PAGE 374  EVENT GET.NX.ORD                      6941
    PAGE 601  ROUTINE BETWEEN.ROUTINE               7548 7549
    PAGE 616  ROUTINE SNAP2                         8167
```

```
    PAGE 698  ''PROGRAM  OLDER.VERSION                1658
ACT.SPEED
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES         274
    PAGE  43  ''SECTION FOR DEFINITIONS               2454
    PAGE 316  ROUTINE FLIGHT.PATH                     4771 4780
    PAGE 351  EVENT AD.ENGAGEMENT                     6014
    PAGE 421  PROCESS AC.ATK.TGT                      8990 8991
    PAGE 422                                          9031
    PAGE 424                                          9158
    PAGE 426                                          9260
    PAGE 583  ROUTINE TACAIR.INPUT                    6851
    PAGE 584                                          6894
    PAGE 585                                          6964
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8426
    PAGE 665  ''PROGRAM  OLDER.VERSION                9716
    PAGE 702                                          1874
ACT.SUBSTITURE
    PAGE  43  ''SECTION FOR DEFINITIONS               2450
    PAGE 702  ''PROGRAM  OLDER.VERSION                1870
ACT.SUBSTITUTE
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES         269
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS           3315
    PAGE 286                                          3345
    PAGE 583  ROUTINE TACAIR.INPUT                    6809 6818
    PAGE 584                                          6897 6901 6903 6909
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8410 8413
    PAGE 664  ''PROGRAM  OLDER.VERSION                9711
ACT.WEATHER.DEGRADE
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES         270
    PAGE  43  ''SECTION FOR DEFINITIONS               2451
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                 2840
    PAGE 281  ROUTINE AC.DF.EFFECTS                   3105
    PAGE 583  ROUTINE TACAIR.INPUT                    6820
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8418
    PAGE 664  ''PROGRAM  OLDER.VERSION                9712
    PAGE 702                                          1871
ACT.X1
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         285
    PAGE  43  ''SECTION FOR DEFINITIONS               2461
    PAGE 314  ROUTINE FLIGHT.PATH                     4662
    PAGE 420  PROCESS AC.ATK.TGT                      8917
    PAGE 506  PROCESS CAS.MISSION                     3515
    PAGE 507                                          3526
    PAGE 583  ROUTINE TACAIR.INPUT                    6834
    PAGE 584                                          6855 6863 6885 6891
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8434
    PAGE 665  ''PROGRAM  OLDER.VERSION                9727
    PAGE 702                                          1881
ACT.X2
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         288
    PAGE  43  ''SECTION FOR DEFINITIONS               2462
    PAGE 420  PROCESS AC.ATK.TGT                      8914
    PAGE 583  ROUTINE TACAIR.INPUT                    6840
    PAGE 584                                          6857 6863 6885 6888
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8435
    PAGE 665  ''PROGRAM  OLDER.VERSION                9730
```

```
    PAGE 702                                        1882
ACT.X3
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       291
    PAGE  43  ''SECTION FOR DEFINITIONS             2463
    PAGE 420  PROCESS AC.ATK.TGT                    8914 8917
    PAGE 506  PROCESS CAS.MISSION                   3521
    PAGE 583  ROUTINE TACAIR.INPUT                  6846
    PAGE 584                                        6859 6888 6891
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8436
    PAGE 665  ''PROGRAM  OLDER.VERSION              9733
    PAGE 702                                        1883
ACT.Y1
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       286
    PAGE  43  ''SECTION FOR DEFINITIONS             2464
    PAGE 314  ROUTINE FLIGHT.PATH                   4664
    PAGE 420  PROCESS AC.ATK.TGT                    8918
    PAGE 506  PROCESS CAS.MISSION                   3517
    PAGE 507                                        3528
    PAGE 583  ROUTINE TACAIR.INPUT                  6835
    PAGE 584                                        6856 6864 6886 6892
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8434
    PAGE 665  ''PROGRAM  OLDER.VERSION              9728
    PAGE 702                                        1884
ACT.Y2
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       289
    PAGE  43  ''SECTION FOR DEFINITIONS             2465
    PAGE 420  PROCESS AC.ATK.TGT                    8915
    PAGE 583  ROUTINE TACAIR.INPUT                  6841
    PAGE 584                                        6858 6864 6886 6889
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8435
    PAGE 665  ''PROGRAM  OLDER.VERSION              9731
    PAGE 702                                        1885
ACT.Y3
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       292
    PAGE  43  ''SECTION FOR DEFINITIONS             2466
    PAGE 420  PROCESS AC.ATK.TGT                    8915 8918
    PAGE 506  PROCESS CAS.MISSION                   3523
    PAGE 583  ROUTINE TACAIR.INPUT                  6847
    PAGE 584                                        6860 6889 6892
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8436
    PAGE 665  ''PROGRAM  OLDER.VERSION              9734
    PAGE 702                                        1886
ACT.Z1
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       287
    PAGE  43  ''SECTION FOR DEFINITIONS             2467
    PAGE 420  PROCESS AC.ATK.TGT                    8919
    PAGE 506  PROCESS CAS.MISSION                   3519
    PAGE 583  ROUTINE TACAIR.INPUT                  6836
    PAGE 584                                        6856 6887 6893
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8434
    PAGE 665  ''PROGRAM  OLDER.VERSION              9729
    PAGE 702                                        1887
ACT.Z2
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       290
    PAGE  43  ''SECTION FOR DEFINITIONS             2468
    PAGE 420  PROCESS AC.ATK.TGT                    8916
```

```
    PAGE 583  ROUTINE TACAIR.INPUT                    6842
    PAGE 584                                          6858 6887 6890
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8435
    PAGE 665  ''PROGRAM  OLDER.VERSION                9732
    PAGE 702                                          1888
ACT.Z3
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         293
    PAGE  43  ''SECTION FOR DEFINITIONS               2469
    PAGE 420  PROCESS AC.ATK.TGT                      8916 8919
    PAGE 507  PROCESS CAS.MISSION                     3525
    PAGE 583  ROUTINE TACAIR.INPUT                    6848
    PAGE 584                                          6860 6890 6893
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8436
    PAGE 665  ''PROGRAM  OLDER.VERSION                9735
    PAGE 702                                          1889
AC.ATK
    PAGE 307  ROUTINE AD.SHOOT                        4346
    PAGE 309                                          4467
    PAGE 422  PROCESS AC.ATK.TGT                      9055
    PAGE 424                                          9173
AC.ATK.TGT
    PAGE  33  ''SECTION FOR PROCESSES                 1892
    PAGE 307  ROUTINE AD.SHOOT                        4325
    PAGE 419  PROCESS AC.ATK.TGT                      8847 8883 8884
    PAGE 423                                          9084 9116
    PAGE 424                                          9192
    PAGE 425                                          9223
    PAGE 426                                          9267 9285
    PAGE 427                                          9313 9317 9318
    PAGE 490  PROCESS ASSESSMENT                      2611 2631
    PAGE 507  PROCESS CAS.MISSION                     3557 3576
    PAGE 616  ROUTINE SNAP2                           8141
    PAGE 692  ''PROGRAM  OLDER.VERSION                1330
AC.ATK.TGTS.
    PAGE 508  PROCESS CAS.MISSION                     3597
AC.ATK.TGT.
    PAGE 309  ROUTINE AD.SHOOT                        4468
    PAGE 599  ROUTINE BETWEEN.ROUTINE                 7448 7449
AC.BOMB.EFFECTS
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                 2787 2796
    PAGE 424  PROCESS AC.ATK.TGT                      9141
AC.DF.EFFECTS
    PAGE 280  ROUTINE AC.DF.EFFECTS                   2996 3007
    PAGE 423  PROCESS AC.ATK.TGT                      9134
AC.MUNS
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES         256  261  264
    PAGE   6                                          295
    PAGE 535  ROUTINE UNIT.INPUT                      4844
    PAGE 588  ROUTINE AC.MUNS.INPUT                   7017 7018 7021 7022 7023 7025 7028 7033 7036 7043
    PAGE 664  ''PROGRAM  OLDER.VERSION                9698 9703 9706
    PAGE 665                                          9737
    PAGE 711                                          2424
AC.MUNS.INPUT
    PAGE 522  ROUTINE MAIN2                           4303 4305
    PAGE 588  ROUTINE AC.MUNS.INPUT                   7011
```

```
AC.TYPE
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES          267
    PAGE   6                                            295
    PAGE 582  ROUTINE TACAIR.INPUT                     6793
    PAGE 583                                           6794
    PAGE 584                                           6898 6899 6902 6903 6909
    PAGE 585                                           6941
    PAGE 588  ROUTINE AC.MUNS.INPUT                    7024 7025
    PAGE 618  ROUTINE SNAP.R                           8213
    PAGE 622  ROUTINE TACAIR.DATA.REPORT               8394 8408
    PAGE 664  ''PROGRAM  OLDER.VERSION                 9709
    PAGE 665                                           9737
    PAGE 711                                           2425
ADMIN.MOV.FAC
    PAGE 403  EVENT START.MOVE                         8196 8198 8211
ADS.MADS.PTR
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES         1102
    PAGE 308  ROUTINE AD.SHOOT                         4407
    PAGE 316  ROUTINE FLIGHT.PATH                      4792 4794
    PAGE 317                                           4841
    PAGE 318                                           4876 4904
    PAGE 350  EVENT AD.ENGAGEMENT                      5974 5975 5992 5997 6001 6003
    PAGE 351                                           6039 6046
    PAGE 354                                           6222
    PAGE 506  PROCESS CAS.MISSION                      3506
    PAGE 566  ROUTINE SENSOR.INPUT                     6092
    PAGE 679  ''PROGRAM  OLDER.VERSION                  543
ADS.MADS.STATUS
    PAGE 679  ''PROGRAM  OLDER.VERSION                  542
ADS.NR.SENSORS
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES         1103
    PAGE 162  ROUTINE ATTRIT.SENSOR                    7495
    PAGE 350  EVENT AD.ENGAGEMENT                      5969 6005
    PAGE 566  ROUTINE SENSOR.INPUT                     6093
    PAGE 679  ''PROGRAM  OLDER.VERSION                  544
ADS.RDS.REMAINING
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES         1105
ADS.TIME.DOWN
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES         1104
    PAGE  47  ''SECTION FOR DEFINITIONS                2711
ADS.UNIT.PTR
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES         1100
    PAGE 162  ROUTINE ATTRIT.SENSOR                    7489
    PAGE 308  ROUTINE AD.SHOOT                         4400
    PAGE 316  ROUTINE FLIGHT.PATH                      4790 4791 4793
    PAGE 317                                           4823 4830
    PAGE 350  EVENT AD.ENGAGEMENT                      5962
    PAGE 506  PROCESS CAS.MISSION                      3495 3497 3502 3509
    PAGE 566  ROUTINE SENSOR.INPUT                     6091
    PAGE 679  ''PROGRAM  OLDER.VERSION                  541
ADS.UNIT.STATUS
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES         1101
    PAGE  53  ''SECTION FOR SUBSTITUTIONS              3009
ADV.TO.WITH
    PAGE  51  ''SECTION FOR SUBSTITUTIONS              2935
    PAGE  83  ROUTINE CHANGE.LOC                       4159
```

```
    PAGE 111  ROUTINE PRED.POS                    5319
    PAGE 114  ROUTINE PREP.WITHDRAW               5470 5480
    PAGE 115  ROUTINE PROX.CHECK                  5519
    PAGE 173  ROUTINE DUST.EFFECTS                7907
    PAGE 174                                      7971 7987
    PAGE 234  ROUTINE SMOKE.EFFECTS                795
    PAGE 236                                       895
    PAGE 256  ROUTINE FO.DETECTION                1844
    PAGE 267  ROUTINE BTRY.EFFECTS                2410
    PAGE 324  ROUTINE DECIDE                      5047
    PAGE 448  PROCESS HC.ARRIVE.BATTLE             415
    PAGE 479  PROCESS WITH.DRAW                   1988
    PAGE 658  PROCESS PHOTO.IR.FLIGHT             9521
    PAGE 710  ''PROGRAM  OLDER.VERSION            2342
AD.ENGAGEMENT
    PAGE  39  ''SECTION FOR EVENTS                2226
    PAGE 293  ROUTINE END.CAS.MISSION             3700 3701 3702
    PAGE 307  ROUTINE AD.SHOOT                    4321
    PAGE 349  EVENT AD.ENGAGEMENT                 5897 5911
    PAGE 505  PROCESS CAS.MISSION                 3439
    PAGE 508                                      3631
    PAGE 601  ROUTINE BETWEEN.ROUTINE             7552 7553
    PAGE 616  ROUTINE SNAP2                       8168
    PAGE 698  ''PROGRAM  OLDER.VERSION            1663
AD.SENSOR
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES    1099
    PAGE 566  ROUTINE SENSOR.INPUT                6088 6089 6091 6092 6093
    PAGE 679  ''PROGRAM  OLDER.VERSION             540
AD.SHOOT
    PAGE 307  ROUTINE AD.SHOOT                    4297 4317
    PAGE 354  EVENT AD.ENGAGEMENT                 6189
    PAGE 422  PROCESS AC.ATK.TGT                  9049
    PAGE 424                                      9167
AE.CALLING.PROCESS
    PAGE  39  ''SECTION FOR EVENTS                2228
    PAGE 293  ROUTINE END.CAS.MISSION             3701
    PAGE 307  ROUTINE AD.SHOOT                    4322 4323
    PAGE 698  ''PROGRAM  OLDER.VERSION            1665
AE.INTERSECTION
    PAGE  39  ''SECTION FOR EVENTS                2227
    PAGE 698  ''PROGRAM  OLDER.VERSION            1664
AGAIN.
    PAGE 425  PROCESS AC.ATK.TGT                  9239
AIRBORNE.RADAR
    PAGE  34  ''SECTION FOR PROCESSES             1906
    PAGE 599  ROUTINE BETWEEN.ROUTINE             7452 7453
    PAGE 616  ROUTINE SNAP2                       8142
    PAGE 653  PROCESS AIRBORNE.RADAR              9261
    PAGE 693  ''PROGRAM  OLDER.VERSION            1344
AIRCRAFT.
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS       3306 3314
    PAGE 309  ROUTINE AD.SHOOT                    4454
    PAGE 419  PROCESS AC.ATK.TGT                  8901
    PAGE 490  PROCESS ASSESSMENT                  2605
AIRFIELD.
    PAGE 535  ROUTINE UNIT.INPUT                  4837 4841
```

```
AIRSPACE.
    PAGE 508  PROCESS CAS.MISSION                    3605
AIR.OBSERVER
    PAGE  34  ''SECTION FOR PROCESSES                1922
    PAGE 159  ROUTINE ATTRIT.SENSOR                  7356 7367
    PAGE 309  ROUTINE AD.SHOOT                       4457 4458
    PAGE 371  EVENT FEBA.SORTIE                      6805
    PAGE 372                                         6848
    PAGE 428  PROCESS AIR.OBSERVER                   9330
    PAGE 432                                         9590
    PAGE 599  ROUTINE BETWEEN.ROUTINE                7456 7457
    PAGE 616  ROUTINE SNAP2                          8143
    PAGE 693  ''PROGRAM  OLDER.VERSION               1360
AIR.SPACE
    PAGE  52  ''SECTION FOR SUBSTITUTIONS            2993
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS          3451
    PAGE 711  ''PROGRAM  OLDER.VERSION               2400
AIR.STRIP
    PAGE 154  ROUTINE AO.DETECTION                   7109
    PAGE 157                                         7271
    PAGE 428  PROCESS AIR.OBSERVER                   9360 9382 9383
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE           1382 1393 1394
    PAGE 653  PROCESS AIRBORNE.RADAR                 9295 9296
    PAGE 656  ROUTINE AR.DETECTION                   9396 9398 9400 9428
ALREADY.DEAD
    PAGE 435  PROCESS ARTY.ASSESS                    9713
    PAGE 436                                         9760
    PAGE 437                                         9806
ALTITUDE.
    PAGE 317  ROUTINE FLIGHT.PATH                    4821
    PAGE 351  EVENT AD.ENGAGEMENT                    6060
AMEP.LA.PERS
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES        262
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                2900
    PAGE 588  ROUTINE AC.MUNS.INPUT                  7033
    PAGE 664    'PROGRAM  OLDER.VERSION              9704
AMET.LA.EO''
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES        265
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                2910
    PAGE 588  ROUTINE AC.MUNS.INPUT                  7043
    PAGE 664  ''PROGRAM  OLDER.VERSION               9707
AM.NAME
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES        257
    PAGE  47  ''SECTION FOR DEFINITIONS              2659
    PAGE 295  ROUTINE END.CAS.MISSION                3842
    PAGE 588  ROUTINE AC.MUNS.INPUT                  7021
    PAGE 664  ''PROGRAM  OLDER.VERSION               9699
    PAGE 705                                         2083
AM.RADIUS
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES        259
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2831 2832
    PAGE 277                                         2877 2879
    PAGE 588  ROUTINE AC.MUNS.INPUT                  7023
    PAGE 664  ''PROGRAM  OLDER.VERSION               9701
AM.RELY
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES        258
```

```
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                2901
    PAGE 278                                         2911
    PAGE 588  ROUTINE AC.MUNS.INPUT                  7022
    PAGE 664  ''PROGRAM  OLDER.VERSION               9700
ANALYSIS
    PAGE   1  ROUTINE FOR CROSS_REFERENCING            55
    PAGE  50  ''SECTION FOR DEFINITIONS              2840
    PAGE  90  ROUTINE FA.BN.MOVEMENT                 4467 4468 4487
    PAGE  91                                         4546 4560
    PAGE  92                                         4577
    PAGE 163  ROUTINE BTRY.FM.DEQ                    7531
    PAGE 164  ROUTINE BTRY.FM.ENQ                    7556
    PAGE 240  ROUTINE TARGET.ANALYSIS                1084
    PAGE 252  ROUTINE MINE.EFFECTS                   1581
    PAGE 253                                         1633
    PAGE 260  ROUTINE BTRY.EFFECTS                   2005
    PAGE 262                                         2121
    PAGE 264                                         2242 2252
    PAGE 265                                         2257 2263 2284 2292 2305
    PAGE 266                                         2314 2323
    PAGE 267                                         2391
    PAGE 295  ROUTINE END.CAS.MISSION                3813
    PAGE 309  ROUTINE AD.SHOOT                       4421 4437
    PAGE 330  ROUTINE EMPTY                          5286
    PAGE 355  EVENT ARTY.OCCUPATION                  6237
    PAGE 385  EVENT OFF.LINE.ATTRITION               7457
    PAGE 392  EVENT SET.DEBUG                        7636
    PAGE 394  EVENT START.ARTY.MOVEMENT              7722
    PAGE 395  EVENT START.BATTLE                     7780
    PAGE 407  EVENT STOP.ARTY.MOVEMENT               8387
    PAGE 437  PROCESS ARTY.ASSESS                    9838
    PAGE 465  PROCESS MINE.ASSESS                    1328
    PAGE 471  PROCESS TARGET.REPORT                  1571
    PAGE 474                                         1718
    PAGE 476                                         1872
    PAGE 487  PROCESS ASSESSMENT                     2463
    PAGE 489                                         2577
    PAGE 490                                         2592 2615
    PAGE 491                                         2654 2669
    PAGE 492                                         2711 2722
    PAGE 499  PROCESS SHOOT.OUT                      3124 3142
    PAGE 500                                         3175 3194
    PAGE 501                                         3216 3244
    PAGE 502                                         3297
    PAGE 514  PROCESS HELICOPTER.FIRE                3938
    PAGE 515                                         3967 3992
    PAGE 516                                         4010
    PAGE 517                                         4071 4089 4105
    PAGE 518                                         4130 4148
    PAGE 522  ROUTINE MAIN2                          4311
    PAGE 523  ROUTINE SYS.INPUT                      4327 4329
    PAGE 609  ROUTINE KV.SCOREBOARD                  7886
    PAGE 708  ''PROGRAM  OLDER.VERSION               2261
ANALYSIS.
    PAGE 606  ROUTINE KV.PRINT                       7712
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES            8495
```

ANAL.CTR

PAGE 50	''SECTION FOR DEFINITIONS	2844
PAGE 55	''PROGRAM'' MAIN	3029
PAGE 56	ROUTINE MAIN1	3037
PAGE 57	ROUTINE MAIN3	3070
PAGE 58	ROUTINE CREATE.FORCE	3095
PAGE 60	ROUTINE CREATE.TEAMS	3161
PAGE 61	ROUTINE FEBA.INITIAL	3185
PAGE 63	ROUTINE FILE.FD.SCHD	3261
PAGE 65	ROUTINE FILE.KAD.SENSOR	3362
PAGE 66	ROUTINE FORM.TF.LIST	3417
PAGE 67	ROUTINE GENERAL.BATTLE	3452
PAGE 71	ROUTINE ORIENTATION	3645
PAGE 75	ROUTINE UNIT.ASSIGNMENT	3831
PAGE 77	ROUTINE ADJUST	3885
PAGE 80	ROUTINE BLOCK.LOS	4010
PAGE 83	ROUTINE CHANGE.LOC	4127
PAGE 87	ROUTINE END.MOVE	4311
PAGE 89	ROUTINE FA.BN.MOVEMENT	4398
PAGE 93	ROUTINE INITIAL.DETECT	4590
PAGE 94	ROUTINE INITIAL.MOVE	4616
PAGE 95	ROUTINE INIT.REINF	4639
PAGE 96	ROUTINE LINE.OF.SIGHT	4668
PAGE 98	ROUTINE LOCATE.SEARCH.AREA	4776
PAGE 100	ROUTINE LOCATE.SECTOR	4835
PAGE 101	ROUTINE LOS.CHECK	4875
PAGE 103	ROUTINE MINE.DELAY	4942
PAGE 106	ROUTINE MIN.MOVE	5064
PAGE 107	ROUTINE NEW.SEGMENT	5108
PAGE 110	ROUTINE POSITION	5271
PAGE 111	ROUTINE PRED.POS	5307
PAGE 112	ROUTINE PREPARE.LIST	5352
PAGE 114	ROUTINE PREP.WITHDRAW	5468
PAGE 115	ROUTINE PROX.CHECK	5501
PAGE 117	ROUTINE PROX.POS	5569
PAGE 118	ROUTINE REIN.ARRIVE	5602
PAGE 119	ROUTINE RESET.FEBA.SECTOR	5652
PAGE 120	ROUTINE SEARCH	5715
PAGE 122	ROUTINE SEGMENT.ADJUST	5778
PAGE 123	ROUTINE TIME.TO.DETECT	5799
PAGE 125	ROUTINE WHAT.NEXT	5878
PAGE 128	ROUTINE BTL.CHECK	5990
PAGE 130	ROUTINE CHECK.DEAD	6070
PAGE 132	ROUTINE CHECK.ENGAGEMENT	6135
PAGE 133	ROUTINE CHECK.FORCE	6157
PAGE 134	ROUTINE CHECK.FOR.MINES	6221
PAGE 137	ROUTINE CHECK.LIST	6334
PAGE 138	ROUTINE CHECK.PROX	6376
PAGE 141	ROUTINE CHECK.STREN	6505
PAGE 142	ROUTINE DEAD.UNIT	6525
PAGE 145	ROUTINE FIN.BATTLE	6695
PAGE 146	ROUTINE INTER.BATTLE	6741
PAGE 149	ROUTINE PK.COMPUTE	6904
PAGE 154	ROUTINE AO.DETECTION	7087
PAGE 159	ROUTINE ATTRIT.SENSOR	7316
PAGE 163	ROUTINE BTRY.FM.DEQ	7514

PAGE 164 ROUTINE BTRY.FM.ENQ 7545
PAGE 165 ROUTINE CFR.DEGRADE 7579
PAGE 166 ROUTINE CFR.DETECTION 7622
PAGE 168 ROUTINE CHK.COMP.TR 7702
PAGE 169 ROUTINE CHK.FD.TR 7730
PAGE 170 ROUTINE COMBINE.TRS 7769
PAGE 171 ROUTINE COMPARE.TRS 7805
PAGE 172 ROUTINE COPY 7848
PAGE 173 ROUTINE DUST.EFFECTS 7890
PAGE 176 ROUTINE EST.COVERAGE 8012
PAGE 180 ROUTINE EST.MIL.WORTH 8224
PAGE 181 ROUTINE FASCAM.COMPUTATION 8249
PAGE 182 ROUTINE FA.BN.ASGN 8292
PAGE 187 ROUTINE FD.EFFECTS.REQ 8528
PAGE 188 ROUTINE FINAL.COVERAGE 8551
PAGE 194 ROUTINE FIND.START.TIME 8898
PAGE 196 ROUTINE HE.OR.ICM.COMPUTATION 8996
PAGE 199 ROUTINE ILLUM.COMPUTATION 9137
PAGE 200 ROUTINE ILLUM.EFFECTS 9188
PAGE 203 ROUTINE MARGINAL.EFFECTS.ADJ 9308
PAGE 205 ROUTINE NOISE.DEGRADE 9386
PAGE 206 ROUTINE PDB.DETECTION 9434
PAGE 208 ROUTINE PGM.MSN.ASGN 9494
PAGE 212 ROUTINE PIR.DETECTION 9705
PAGE 214 ROUTINE REM.EFFECTS.COMPUTATION 9799
PAGE 215 ROUTINE REQUEST.DEF.FASCAM 9836
PAGE 216 ROUTINE REQUEST.FASCAM 9898
PAGE 218 ROUTINE REQUEST.ILLUM 9990
PAGE 222 ROUTINE REQUEST.SMOKE 181
PAGE 226 ROUTINE REQUEST.WD.FASCAM 399
PAGE 229 ROUTINE RPV.DETECTION 517
PAGE 231 ROUTINE SIZE.ESTIMATE 628
PAGE 233 ROUTINE SMOKE.COMPUTATION 691
PAGE 234 ROUTINE SMOKE.EFFECTS 742
PAGE 238 ROUTINE SWITCH.FO 918
PAGE 239 ROUTINE TARGET.ANALYSIS 976
PAGE 242 ROUTINE UNIT.ENVIR 1119
PAGE 246 ROUTINE VOLLEY 1311
PAGE 248 ROUTINE WEIGHTED.VOLLEYS 1378
PAGE 250 ROUTINE MINE.EFFECTS 1471
PAGE 254 ROUTINE FO.DETECTION 1679
PAGE 260 ROUTINE BTRY.EFFECTS 1968
PAGE 276 ROUTINE AC.BOMB.EFFECTS 2793
PAGE 280 ROUTINE AC.DF.EFFECTS 3004
PAGE 283 ROUTINE CAS.EVAL 3171
PAGE 285 ROUTINE CHECK.CAS.CONSTRAINTS 3284
PAGE 289 ROUTINE EMPLOY.HELICOPTERS 3478
PAGE 293 ROUTINE END.CAS.MISSION 3681
PAGE 297 ROUTINE FARRP.CHECK 3863
PAGE 298 ROUTINE HC.COMPUTE.TIMES 3923
PAGE 300 ROUTINE HC.DISENGAGE 4009
PAGE 302 ROUTINE HEL.RANGE.COMPUTE 4126
PAGE 304 ROUTINE REPLACE.HC 4192
PAGE 305 ROUTINE UNIT.PRIORITY 4241
PAGE 307 ROUTINE AD.SHOOT 4309
PAGE 311 ROUTINE INTER.HELO 4518

PAGE 314	ROUTINE FLIGHT.PATH	4634
PAGE 321	ROUTINE COMPUTE.D	4949
PAGE 322	ROUTINE COMPUTE.WD	4971
PAGE 323	ROUTINE CONTRAST.TO.FREQ	5005
PAGE 324	ROUTINE DECIDE	5039
PAGE 325	ROUTINE DEQ.FEBA.SET	5082
PAGE 326	ROUTINE DESTROY.ORD	5119
PAGE 327	ROUTINE DQ.CMSN.QUEUE	5144
PAGE 328	ROUTINE EMPTY	5171
PAGE 331	ROUTINE ENQ.FEBA.SET	5304
PAGE 332	ROUTINE FDC.TR.DEQ	5343
PAGE 333	ROUTINE FDC.TR.ENQ	5362
PAGE 334	ROUTINE FINISH.COMPUTATION	5383
PAGE 335	ROUTINE FRAC.COMPUTE	5414
PAGE 336	ROUTINE GET.TERRAIN	5436
PAGE 337	ROUTINE HC.EMPTY	5462
PAGE 340	ROUTINE PROB.INF	5611
PAGE 341	ROUTINE PROB.TIME	5643
PAGE 342	ROUTINE RANGE.COMPUTE	5666
PAGE 343	ROUTINE SEARCH.COVERAGE	5684
PAGE 344	ROUTINE TEMPERATURE.ATTENUATION	5740
PAGE 345	ROUTINE TERM.CHECK	5769
PAGE 347	EVENT ACT.REINF	5809
PAGE 349	EVENT AD.ENGAGEMENT	5902
PAGE 355	EVENT ARTY.OCCUPATION	6230
PAGE 356	EVENT BTL.ENDED	6255
PAGE 358	EVENT CFR.ACTIVATION	6345
PAGE 359	EVENT CFR.OFF	6385
PAGE 360	EVENT CFR.ON	6410
PAGE 362	EVENT CFR.OPERATOR	6482
PAGE 365	EVENT CHANGE.LITE	6597
PAGE 366	EVENT CHANGE.WEATHER	6620
PAGE 367	EVENT DQ.OLD.SORTIE.QUEUE	6638
PAGE 368	EVENT END.SIMULATION	6663
PAGE 369	EVENT ENGAGEMENT	6683
PAGE 371	EVENT FEBA.SORTIE	6781
PAGE 373	EVENT GET.NX.ORD	6882
PAGE 375	EVENT HC.DEPART.BATTLE	6994
PAGE 376	EVENT HELO.ENGAGEMENT	7037
PAGE 380	EVENT INIT.PREPLAN.CAS	7235
PAGE 381	EVENT MOVE	7271
PAGE 382	EVENT OFF.LINE.ATTRITION	7281
PAGE 386	EVENT PDB.ACTIVATION	7472
PAGE 387	EVENT PDB.OPERATOR	7508
PAGE 389	EVENT POSITION.REPORT	7570
PAGE 390	EVENT SCHEDULE.ARTY.MOVEMENT	7581
PAGE 391	EVENT SEND.TEAM	7601
PAGE 392	EVENT SET.DEBUG	7625
PAGE 393	EVENT START.ARTY.MOVEMENT	7660
PAGE 395	EVENT START.BATTLE	7752
PAGE 403	EVENT START.MOVE	8185
PAGE 407	EVENT STOP.ARTY.MOVEMENT	8375
PAGE 408	EVENT UPDATE.LOC	8413
PAGE 412	EVENT ACT.ATK	8606
PAGE 414	EVENT ACT.DEF	8719
PAGE 415	EVENT ACT.MOVCOR	8741

PAGE 416	EVENT ACT.MOVDIS	8771
PAGE 417	EVENT DYNAMIC.ANALYSIS.REPORT	8825 8827 8830
PAGE 419	PROCESS AC.ATK.TGT	8849
PAGE 428	PROCESS AIR.OBSERVER	9333
PAGE 435	PROCESS ARTY.ASSESS	9688
PAGE 438	PROCESS FORWARD.OBSERVER	9856
PAGE 443	PROCESS HC.ARRIVE.BATTLE	98
PAGE 450	PROCESS HC.RETURN.FARRP	465
PAGE 454	PROCESS HEL.TARGET.ACQUISITION	697
PAGE 463	PROCESS HOW.REPAIR	1172
PAGE 464	PROCESS MINE.ASSESS	1222
PAGE 467	PROCESS REMOTE.PILOT.VEHICLE	1357
PAGE 470	PROCESS TARGET.REPORT	1484
PAGE 478	PROCESS WITH.DRAW	1918
PAGE 480	PROCESS FIRE.MISSION	2029
PAGE 487	PROCESS ASSESSMENT	2419
PAGE 493	PROCESS SHOOT.OUT	2751
PAGE 504	PROCESS CAS.MISSION	3351
PAGE 510	PROCESS HELICOPTER.FIRE	3662
PAGE 520	ROUTINE MAIN2	4185
PAGE 523	ROUTINE SYS.INPUT	4318
PAGE 525	ROUTINE PK.INPUT	4382
PAGE 527	ROUTINE CAT.TU.INPUT	4452
PAGE 529	ROUTINE KV.INPUT	4525
PAGE 530	ROUTINE EQ.TE.INPUT	4575
PAGE 532	ROUTINE TYPE.WEAPON.INPUT	4640
PAGE 533	ROUTINE UNIT.INPUT	4682
PAGE 538	ROUTINE MFO.INPUT	4932
PAGE 539	ROUTINE READ.ORDERS	4985
PAGE 542	ROUTINE ORD.DEF	5118
PAGE 543	ROUTINE ORD.ATK	5143
PAGE 544	ROUTINE ORD.REINF	5162
PAGE 545	ROUTINE ORD.MOVDIS	5183
PAGE 546	ROUTINE ORD.MOVCOR	5208
PAGE 547	ROUTINE P.E.M.INPUT	5257
PAGE 548	ROUTINE TB.INPUT	5309
PAGE 550	ROUTINE BTRY.INPUT	5389
PAGE 553	ROUTINE FBN.FD.INPUT	5541
PAGE 555	ROUTINE MUNS.INPUT	5604
PAGE 557	ROUTINE SUBM.INPUT	5715
PAGE 558	ROUTINE HE.LA.INPUT	5752
PAGE 560	ROUTINE RUL.EN.INPUT	5822
PAGE 561	ROUTINE ST.INPUT	5859
PAGE 562	ROUTINE MCFR.INPUT	5880
PAGE 563	ROUTINE MPDB.INPUT	5917
PAGE 564	ROUTINE MAO.INPUT	5955
PAGE 565	ROUTINE SENSOR.INPUT	6013
PAGE 569	ROUTINE TBF.INPUT	6204
PAGE 570	ROUTINE DECISION.INPUT	6262
PAGE 571	ROUTINE TT.FACTORS.INPUT	6284
PAGE 572	ROUTINE FARRP.INPUT	6329
PAGE 575	ROUTINE PGM.INPUT	6451
PAGE 576	ROUTINE ILLUM.INPUT	6496
PAGE 577	ROUTINE MINE.INPUT	6535
PAGE 580	ROUTINE SMOKE.INPUT	6670
PAGE 581	ROUTINE VIS.INPUT	6713

```
    PAGE 582  ROUTINE TACAIR.INPUT                    6740
    PAGE 587  ROUTINE MADS.INPUT                      6980
    PAGE 588  ROUTINE AC.MUNS.INPUT                   7013
    PAGE 590  ROUTINE TR.INPUT                        7077
    PAGE 592  ROUTINE AMMO.RPT                        7121
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                 7241
    PAGE 599  ROUTINE BETWEEN.ROUTINE                 7430
    PAGE 604  ROUTINE ERROR.STOP                      7667
    PAGE 605  ROUTINE HEADING                         7681
    PAGE 606  ROUTINE KV.PRINT                        7690
    PAGE 609  ROUTINE KV.SCOREBOARD                   7843
    PAGE 611  ROUTINE OUTPUT.ATTRITION                7931
    PAGE 613  ROUTINE FOR POSITION.OUT                7995
    PAGE 614  ROUTINE SNAP2                           8022
    PAGE 618  ROUTINE SNAP.R                          8203
    PAGE 620                                          8321 8323 8326
    PAGE 621  ROUTINE TACAIR.DATA.REPORT              8335
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES             8490
    PAGE 628  FUNCTION ACT.RANGE                      8633
    PAGE 629  FUNCTION BTRY.AVAILABLE                 8647
    PAGE 630  FUNCTION COLLISION                      8681
    PAGE 631  FUNCTION COMBINATIONS                   8717
    PAGE 632  FUNCTION EST.RANGE                      8747
    PAGE 633  FUNCTION EST.TR.RANGE                   8762
    PAGE 634  FUNCTION FEBA.BAND                      8777
    PAGE 636  FUNCTION HE.WLA                         8860
    PAGE 638  FUNCTION ICM.WLA                        8960
    PAGE 640  ROUTINE EXPONENTIAL.F                   9027
    PAGE 641  ROUTINE NORMAL.F                        9040
    PAGE 642  ROUTINE WEIBULL.F                       9063
    PAGE 644  ROUTINE ANGLE.COMPUTE                   9085
    PAGE 645  ROUTINE LINE.CIRCLE                     9132
    PAGE 646  ROUTINE MRT.TO.FREQ                     9162
    PAGE 648  ROUTINE PERFORM.INSTRUMENTATION         9196 9203 9204 9209
    PAGE 652  ROUTINE GAMMA.F                         9238
    PAGE 653  PROCESS AIRBORNE.RADAR                  9264
    PAGE 656  ROUTINE AR.DETECTION                    9386
    PAGE 657  FUNCTION AR.PROB.DETECT                 9445
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                 9475
    PAGE 660  FUNCTION STAY.TIME                      9594
    PAGE 661  ROUTINE JOHNSON.CRITERIA                9619
    PAGE 662  ROUTINE PROXIMITY.REQ                   9633
    PAGE 663  ROUTINE TIME.REQ                        9647
    PAGE 714  ROUTINE PLAT.COUNT                      2515
ANAL.TEXT
    PAGE  50  ''SECTION FOR DEFINITIONS               2845
    PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT           8824
    PAGE 620  ROUTINE SNAP.R                          8320
    PAGE 648  ROUTINE PERFORM.INSTRUMENTATION         9197 9201
AOC.BTRY
    PAGE  39  ''SECTION FOR EVENTS                    2231
    PAGE 698  ''PROGRAM  OLDER.VERSION                1668
AO.CAND.DET.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES        1128
    PAGE  34  ''SECTION FOR PROCESSES                 1930
    PAGE  49  ''SECTION FOR DEFINITIONS               2791
```

```
    PAGE 430  PROCESS AIR.OBSERVER                    9500
    PAGE 431                                          9504 9506 9525 9527 9544 9546
    PAGE 432                                          9575 9577 9608 9610
    PAGE 433                                          9626 9628 9644 9646 9663 9665
    PAGE 434                                          9678 9680
    PAGE 679  ''PROGRAM  OLDER.VERSION                 567
    PAGE 693                                          1368
    PAGE 708                                          2212
AO.CURRENT.TR
    PAGE  34  ''SECTION FOR PROCESSES                 1924
    PAGE 371  EVENT FEBA.SORTIE                       6811
    PAGE 372                                          6852
    PAGE 431  PROCESS AIR.OBSERVER                    9554
    PAGE 432                                          9588 9589 9590
    PAGE 693  ''PROGRAM  OLDER.VERSION                1362
AO.DC.DIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES        1126
    PAGE  46  ''SECTION FOR DEFINITIONS               2614
    PAGE 154  ROUTINE AO.DETECTION                    7106
    PAGE 430  PROCESS AIR.OBSERVER                    9499
    PAGE 679  ''PROGRAM  OLDER.VERSION                 565
    PAGE 705                                          2037
AO.DC.LEG.DIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES        1125
    PAGE  46  ''SECTION FOR DEFINITIONS               2613
    PAGE  49                                          2791
    PAGE 430  PROCESS AIR.OBSERVER                    9498
    PAGE 431                                          9511 9513
    PAGE 679  ''PROGRAM  OLDER.VERSION                 564
    PAGE 705                                          2036
    PAGE 708                                          2212
AO.DC.UNIT
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES        1124
    PAGE  46  ''SECTION FOR DEFINITIONS               2612
    PAGE 154  ROUTINE AO.DETECTION                    7107
    PAGE 430  PROCESS AIR.OBSERVER                    9497
    PAGE 679  ''PROGRAM  OLDER.VERSION                 563
    PAGE 705                                          2035
AO.DETECTION
    PAGE 154  ROUTINE AO.DETECTION                    7074 7115
    PAGE 242  ROUTINE UNIT.ENVIR                      1127
    PAGE 431  PROCESS AIR.OBSERVER                    9555
AO.DET.CANDIDATE
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES        1123
    PAGE 430  PROCESS AIR.OBSERVER                    9496
    PAGE 431                                          9508 9518 9528 9543 9547 9559
    PAGE 432                                          9574 9578 9592 9611
    PAGE 433                                          9629 9647 9666
    PAGE 434                                          9681
    PAGE 679  ''PROGRAM  OLDER.VERSION                 562
AO.DET.TGT.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES        1010
    PAGE  34  ''SECTION FOR PROCESSES                 1931
    PAGE  49  ''SECTION FOR DEFINITIONS               2792
    PAGE 677  ''PROGRAM  OLDER.VERSION                 452
    PAGE 693                                          1369
```

```
    PAGE 708                                        2213
AO.EB.ALTITUDE
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       299
    PAGE  49  ''SECTION FOR DEFINITIONS             2793
    PAGE 155  ROUTINE AO.DETECTION                  7156
    PAGE 156                                        7190 7191 7191 7223 7224 7224
    PAGE 564  ROUTINE MAO.INPUT                     5983 5984 5984
    PAGE 665  ''PROGRAM  OLDER.VERSION              9741
    PAGE 708                                        2214
AO.EB.SET
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       301
    PAGE  15                                         831
    PAGE  49  ''SECTION FOR DEFINITIONS             2793
    PAGE 155  ROUTINE AO.DETECTION                  7155
    PAGE 564  ROUTINE MAO.INPUT                     5985
    PAGE 665  ''PROGRAM  OLDER.VERSION              9743
    PAGE 674                                         273
    PAGE 708                                        2214
AO.ELEVATION.BAND
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       298  307
    PAGE 564  ROUTINE MAO.INPUT                     5979 5981 5997
    PAGE 618  ROUTINE SNAP.R                        8211
    PAGE 665  ''PROGRAM  OLDER.VERSION              9740 9749
    PAGE 711                                        2426
AO.ELEV.BD
    PAGE 155  ROUTINE AO.DETECTION                  7155 7156 7170 7171 7172 7174 7176 7180 7181 7182 7183
    PAGE 156                                        7190 7191 7191 7203 7204 7205 7207 7209 7213 7214 7215 7216 7223 7224 7224
AO.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES      1392
    PAGE  34  ''SECTION FOR PROCESSES               1929
    PAGE  49  ''SECTION FOR DEFINITIONS             2794
    PAGE 371  EVENT FEBA.SORTIE                     6825
    PAGE 372                                        6866
    PAGE 429  PROCESS AIR.OBSERVER                  9389 9391 9399 9401 9430
    PAGE 430                                        9446 9457
    PAGE 431                                        9520 9522 9538 9540
    PAGE 432                                        9569 9571 9603 9605
    PAGE 433                                        9621 9623 9639 9641 9658 9660 9673 9675
    PAGE 684  ''PROGRAM  OLDER.VERSION               831
    PAGE 693                                        1367
    PAGE 708                                        2215
AO.PROB.LOS
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       308
    PAGE 155  ROUTINE AO.DETECTION                  7170 7171 7172 7174 7180 7181 7182 7183
    PAGE 156                                        7203 7204 7205 7207 7213 7214 7215 7216
    PAGE 564  ROUTINE MAO.INPUT                     6000
    PAGE 665  ''PROGRAM  OLDER.VERSION              9750
AO.RANGE.BAND
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       307  311
    PAGE  11                                         618
    PAGE 564  ROUTINE MAO.INPUT                     5989 5991 5998 6004
    PAGE 618  ROUTINE SNAP.R                        8212
    PAGE 665  ''PROGRAM  OLDER.VERSION              9749 9753
    PAGE 670                                          59
    PAGE 711                                        2427
```

```
AO.RB.RANGE
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES          312
    PAGE  49  ''SECTION FOR DEFINITIONS                2795
    PAGE 155  ROUTINE AO.DETECTION                     7162 7168 7169 7169 7178 7179 7179
    PAGE 156                                           7195 7201 7202 7202 7211 7212 7212
    PAGE 564  ROUTINE MAO.INPUT                        5993 5994 5994
    PAGE 665  ''PROGRAM  OLDER.VERSION                 9754
    PAGE 708                                           2216
AO.RB.SET
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES          314
    PAGE  15                                            830
    PAGE  49  ''SECTION FOR DEFINITIONS                2795
    PAGE 155  ROUTINE AO.DETECTION                     7161
    PAGE 156                                           7194
    PAGE 564  ROUTINE MAO.INPUT                        5995
    PAGE 665  ''PROGRAM  OLDER.VERSION                 9756
    PAGE 674                                            272
    PAGE 708                                           2216
AO.RNG.BD
    PAGE 155  ROUTINE AO.DETECTION                     7161 7162 7167 7168 7169 7169 7170 7171 7172 7174 7177 7178 7179 7179 7180
                                                       7181 7182 7183
    PAGE 156                                           7194 7195 7200 7201 7202 7202 7203 7204 7205 7207 7210 7211 7212 7212 7213
                                                       7214 7215 7216
    PAGE 157                                           7272
AO.SECTOR
    PAGE 371  EVENT FEBA.SORTIE                        6797 6798
    PAGE 372                                           6840 6841
AO.US.LINK
    PAGE  34  ''SECTION FOR PROCESSES                  1925
    PAGE 154  ROUTINE AO.DETECTION                     7108
    PAGE 307  ROUTINE AD.SHOOT                         4341
    PAGE 309                                           4456
    PAGE 371  EVENT FEBA.SORTIE                        6808
    PAGE 372                                           6849
    PAGE 428  PROCESS AIR.OBSERVER                     9359
    PAGE 693  ''PROGRAM  OLDER.VERSION                 1363
AO.VEL
    PAGE 428  PROCESS AIR.OBSERVER                     9364 9366
AO.VELOCITY
    PAGE  34  ''SECTION FOR PROCESSES                  1923
    PAGE 351  EVENT AD.ENGAGEMENT                      6019
    PAGE 428  PROCESS AIR.OBSERVER                     9336
    PAGE 693  ''PROGRAM  OLDER.VERSION                 1361
AO.VEL2
    PAGE 428  PROCESS AIR.OBSERVER                     9365 9366
AO.VISIBILITY
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES          309
    PAGE 564  ROUTINE MAO.INPUT                        6001
    PAGE 665  ''PROGRAM  OLDER.VERSION                 9751
AO.X.START
    PAGE  34  ''SECTION FOR PROCESSES                  1926
    PAGE 371  EVENT FEBA.SORTIE                        6817 6821 6823
    PAGE 372                                           6858 6862 6864
    PAGE 428  PROCESS AIR.OBSERVER                     9382
    PAGE 693  ''PROGRAM  OLDER.VERSION                 1364
```

```
AO.Y.COORD
    PAGE 371  EVENT FEBA.SORTIE                    6796 6797
    PAGE 372                                       6839 6840
AO.Y.START
    PAGE  34  ''SECTION FOR PROCESSES              1927
    PAGE 371  EVENT FEBA.SORTIE                    6818 6822 6824
    PAGE 372                                       6859 6863 6865
    PAGE 428  PROCESS AIR.OBSERVER                 9383
    PAGE 693  ''PROGRAM  OLDER.VERSION             1365
APPROACH.DISTANCE
    PAGE 428  PROCESS AIR.OBSERVER                 9346 9382 9384
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE         1369 1393 1395
    PAGE 653  PROCESS AIRBORNE.RADAR               9274 9295 9297
APPROACH.TIME
    PAGE 428  PROCESS AIR.OBSERVER                 9346 9384 9385 9386
    PAGE 429                                       9423
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE         1369 1395 1396 1399
    PAGE 653  PROCESS AIRBORNE.RADAR               9274 9297 9298 9302
ARCCOS.F
    PAGE 277  ROUTINE AC.BOMB.EFFECTS              2855 2857
    PAGE 343  ROUTINE SEARCH.COVERAGE              5705 5706
ARCSIN.F
    PAGE 566  ROUTINE SENSOR.INPUT                 6111
ARCTAN.F
    PAGE 166  ROUTINE CFR.DETECTION                7652
    PAGE 360  EVENT CFR.ON                         6458
    PAGE 446  PROCESS HC.ARRIVE.BATTLE              319
    PAGE 447                                        347
    PAGE 507  PROCESS CAS.MISSION                  3544
    PAGE 584  ROUTINE TACAIR.INPUT                 6879
    PAGE 644  ROUTINE ANGLE.COMPUTE                9112
AREA.
    PAGE 280  ROUTINE AC.DF.EFFECTS                3016
AREA.ADJ.PAT
    PAGE 178  ROUTINE EST.COVERAGE                 8168
    PAGE 179                                       8178
    PAGE 191  ROUTINE FINAL.COVERAGE               8736
    PAGE 192                                       8810
AREA.COVERED
    PAGE 343  ROUTINE SEARCH.COVERAGE              5709 5711 5714 5718 5720 5722 5726 5730
AREA.DAM
    PAGE 191  ROUTINE FINAL.COVERAGE               8741 8747
    PAGE 192                                       8810 8820 8824 8827
    PAGE 193                                       8855 8859
ARG.ARRAY
    PAGE  41  ''SECTION FOR EVENTS                 2317
    PAGE 146  ROUTINE INTER.BATTLE                 6750 6789
    PAGE 147                                       6852
    PAGE 148                                       6889
    PAGE 311  ROUTINE INTER.HELO                   4516 4520 4523 4525 4526 4527 4528 4529 4530 4531 4532
    PAGE 395  EVENT START.BATTLE                   7750 7760 7764 7793 7794
    PAGE 396                                       7858
    PAGE 397                                       7859
    PAGE 401                                       8144 8145
    PAGE 402                                       8149 8150
    PAGE 700  ''PROGRAM  OLDER.VERSION             1754
```

```
AROUND.
    PAGE 315  ROUTINE FLIGHT.PATH                    4726
ARTY.ASS
    PAGE 261  ROUTINE BTRY.EFFECTS                   2079
    PAGE 262                                         2080 2081 2082 2091
    PAGE 269                                         2532 2536
    PAGE 270                                         2550
    PAGE 272                                         2699 2700
ARTY.ASSESS
    PAGE  34  ''SECTION FOR PROCESSES                1943
    PAGE 261  ROUTINE BTRY.EFFECTS                   2079
    PAGE 272                                         2700
    PAGE 328  ROUTINE EMPTY                          5217 5218 5220 5221 5222
    PAGE 435  PROCESS ARTY.ASSESS                    9686 9690 9694 9696 9700 9703 9710
    PAGE 599  ROUTINE BETWEEN.ROUTINE                7460 7461
    PAGE 616  ROUTINE SNAP2                          8144
    PAGE 693  ''PROGRAM  OLDER.VERSION               1381
ARTY.DECIMATE
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES        734
    PAGE 183  ROUTINE FA.BN.ASGN                     8385
    PAGE 209  ROUTINE PGM.MSN.ASGN                   9553
    PAGE 480  PROCESS FIRE.MISSION                   2078 2079
    PAGE 481                                         2093 2094 2095 2096
    PAGE 523  ROUTINE SYS.INPUT                      4344
    PAGE 673  ''PROGRAM  OLDER.VERSION                177
ARTY.DEGRADE
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES        735
    PAGE 481  PROCESS FIRE.MISSION                   2085 2093
    PAGE 523  ROUTINE SYS.INPUT                      4345
    PAGE 673  ''PROGRAM  OLDER.VERSION                178
ARTY.MINE
    PAGE 250  ROUTINE MINE.EFFECTS                   1480 1493
ARTY.OCCUPATION
    PAGE  39  ''SECTION FOR EVENTS                   2230
    PAGE 355  EVENT ARTY.OCCUPATION                  6227 6234
    PAGE 407  EVENT STOP.ARTY.MOVEMENT               8397
    PAGE 601  ROUTINE BETWEEN.ROUTINE                7556 7557
    PAGE 616  ROUTINE SNAP2                          8169
    PAGE 698  ''PROGRAM  OLDER.VERSION               1667
AR.CAND.DET.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES       1138
    PAGE  34  ''SECTION FOR PROCESSES                1913
    PAGE 654  PROCESS AIRBORNE.RADAR                 9337 9342 9348
    PAGE 679  ''PROGRAM  OLDER.VERSION                577
    PAGE 693                                         1351
AR.DC.UNIT
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES       1135
    PAGE 654  PROCESS AIRBORNE.RADAR                 9334 9356
    PAGE 656  ROUTINE AR.DETECTION                   9395
    PAGE 679  ''PROGRAM  OLDER.VERSION                574
AR.DC.Y.COORD
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES       1136
    PAGE 654  PROCESS AIRBORNE.RADAR                 9335 9349 9350
    PAGE 679  ''PROGRAM  OLDER.VERSION                575
AR.DETECTION
    PAGE 654  PROCESS AIRBORNE.RADAR                 9354
```

```
    PAGE 656  ROUTINE AR.DETECTION                    9385 9393
AR.DET.CANDIDATE
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES        1134
    PAGE 654  PROCESS AIRBORNE.RADAR                  9333 9361
    PAGE 679  ''PROGRAM  OLDER.VERSION                 573
AR.DET.COEFF
    PAGE  50  ''SECTION FOR DEFINITIONS               2833
    PAGE 657  FUNCTION AR.PROB.DETECT                 9464 9464 9465 9469
    PAGE 708  ''PROGRAM  OLDER.VERSION                2254
AR.DET.TGT.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES        1011
    PAGE  34  ''SECTION FOR PROCESSES                 1912
    PAGE  49  ''SECTION FOR DEFINITIONS               2792
    PAGE 653  PROCESS AIRBORNE.RADAR                  9318
    PAGE 654                                          9362
    PAGE 655                                          9380 9381
    PAGE 677  ''PROGRAM  OLDER.VERSION                 453
    PAGE 693                                          1350
    PAGE 708                                          2213
AR.DURATION
    PAGE  34  ''SECTION FOR PROCESSES                 1910
    PAGE 693  ''PROGRAM  OLDER.VERSION                1348
AR.PROB.DETECT
    PAGE  50  ''SECTION FOR DEFINITIONS               2849
    PAGE 656  ROUTINE AR.DETECTION                    9405 9406
    PAGE 657  FUNCTION AR.PROB.DETECT                 9443
    PAGE 709  ''PROGRAM  OLDER.VERSION                2268
AR.X.GRID
    PAGE 653  PROCESS AIRBORNE.RADAR                  9282 9284 9285 9288 9289
    PAGE 654                                          9327
AR.X.START
    PAGE  34  ''SECTION FOR PROCESSES                 1907
    PAGE 653  PROCESS AIRBORNE.RADAR                  9282
    PAGE 693  ''PROGRAM  OLDER.VERSION                1345
AR.Y.FINISH
    PAGE  34  ''SECTION FOR PROCESSES                 1909
    PAGE 693  ''PROGRAM  OLDER.VERSION                1347
AR.Y.START
    PAGE  34  ''SECTION FOR PROCESSES                 1908
    PAGE 693  ''PROGRAM  OLDER.VERSION                1346
AS.DESTRUCT.INDIC
    PAGE  35  ''SECTION FOR PROCESSES                 1960
    PAGE  81  ROUTINE BLOCK.LOS                       4086 4088
    PAGE 188  ROUTINE NEW.SEGMENT                     5186 5189
    PAGE 294  ROUTINE END.CAS.MISSION                 3752
    PAGE 301  ROUTINE HC.DISENGAGE                    4103
    PAGE 328  ROUTINE EMPTY                           5204
    PAGE 425  PROCESS AC.ATK.TGT                      9251
    PAGE 426                                          9302
    PAGE 487  PROCESS ASSESSMENT                      2427 2454
    PAGE 488                                          2522
    PAGE 490                                          2641
    PAGE 694  ''PROGRAM  OLDER.VERSION                1398
AS.FIRING.UNIT
    PAGE  34  ''SECTION FOR PROCESSES                 1958
    PAGE  81  ROUTINE BLOCK.LOS                       4082 4084
```

```
    PAGE 108  ROUTINE NEW.SEGMENT                    5174 5176
    PAGE 328  ROUTINE EMPTY                          5198
    PAGE 694  ''PROGRAM  OLDER.VERSION               1396
AS.PK
    PAGE  34  ''SECTION FOR PROCESSES                1955
    PAGE  47  ''SECTION FOR DEFINITIONS              2712
    PAGE 693  ''PROGRAM  OLDER.VERSION               1393
    PAGE 706                                         2134
AS.SHOOT.OUT
    PAGE  34  ''SECTION FOR PROCESSES                1959
    PAGE 294  ROUTINE END.CAS.MISSION                3749
    PAGE 301  ROUTINE HC.DISENGAGE                   4099
    PAGE 425  PROCESS AC.ATK.TGT                     9248
    PAGE 426                                         9299
    PAGE 488  PROCESS ASSESSMENT                     2519
    PAGE 490                                         2638
    PAGE 694  ''PROGRAM  OLDER.VERSION               1397
AS.TGT.EQUIP
    PAGE  34  ''SECTION FOR PROCESSES                1956
    PAGE 693  ''PROGRAM  OLDER.VERSION               1394
AS.TGT.UNIT
    PAGE  34  ''SECTION FOR PROCESSES                1957
    PAGE  81  ROUTINE BLOCK.LOS                      4081 4083
    PAGE 108  ROUTINE NEW.SEGMENT                    5173 5175
    PAGE 328  ROUTINE EMPTY                          5199 5200
    PAGE 694  ''PROGRAM  OLDER.VERSION               1395
ATKENEMY
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           51
    PAGE  39  ''SECTION FOR EVENTS                   2207
    PAGE 698  ''PROGRAM  OLDER.VERSION               1644
ATKORDER
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           50
    PAGE  39  ''SECTION FOR EVENTS                   2206
    PAGE 698  ''PROGRAM  OLDER.VERSION               1643
ATKUNIT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           50
    PAGE  39  ''SECTION FOR EVENTS                   2205
    PAGE  59  ROUTINE CREATE.FORCE                   3148
    PAGE 698  ''PROGRAM  OLDER.VERSION               1642
ATK.ATTRITION
    PAGE 451  PROCESS HC.RETURN.FARRP                 526  527  537  555  557
ATK.COUNT
    PAGE 452  PROCESS HC.RETURN.FARRP                 597  630
ATK.DELAY
    PAGE  42  ''SECTION FOR DEFINITIONS              2382
    PAGE 374  EVENT GET.NX.ORD                       6937 6938 6957 6958
    PAGE 524  ROUTINE SYS.INPUT                      4375
    PAGE 701  ''PROGRAM  OLDER.VERSION               1812
ATK.FAIL.PROB
    PAGE 450  PROCESS HC.RETURN.FARRP                 472  512  515
    PAGE 451                                          520  522  525  526
ATK.FARRP
    PAGE 376  EVENT HELO.ENGAGEMENT                  7033 7048
ATK.HC
    PAGE  50  ''SECTION FOR DEFINITIONS              2839
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS             3508
```

```
    PAGE 297  ROUTINE FARRP.CHECK                    3884
    PAGE 452  PROCESS HC.RETURN.FARRP                 596
    PAGE 510  PROCESS HELICOPTER.FIRE                3705
    PAGE 572  ROUTINE FARRP.INPUT                    6348 6351 6376
    PAGE 708  ''PROGRAM  OLDER.VERSION               2260
ATK.HELICOPTER
    PAGE  36  ''SECTION FOR PROCESSES                2063
    PAGE 379  EVENT HELO.ENGAGEMENT                  7221
    PAGE 488  PROCESS ASSESSMENT                     2507
    PAGE 511  PROCESS HELICOPTER.FIRE                3744
    PAGE 695  ''PROGRAM  OLDER.VERSION               1501
ATK.HELO
    PAGE 513  PROCESS HELICOPTER.FIRE                3883
ATK.HELO.UNIT
    PAGE 379  EVENT HELO.ENGAGEMENT                  7204
ATK.ORDER
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES       1144
    PAGE 326  ROUTINE DESTROY.ORD                    5128
    PAGE 543  ROUTINE ORD.ATK                        5152
    PAGE 680  ''PROGRAM  OLDER.VERSION                583
    PAGE 712                                         2482
ATK.TEAM
    PAGE 376  EVENT HELO.ENGAGEMENT                  7032 7059 7066 7078
    PAGE 377                                         7106 7109
    PAGE 378                                         7147 7167 7180
    PAGE 510  PROCESS HELICOPTER.FIRE                3668 3677 3688 3702 3712
    PAGE 513                                         3835
    PAGE 515                                         3962
    PAGE 517                                         4121
ATK.UE.ID
    PAGE 128  ROUTINE BTL.CHECK                      5988 6014 6033
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS             3511
    PAGE 290                                         3532 3543 3580
    PAGE 291                                         3637
    PAGE 297  ROUTINE FARRP.CHECK                    3885 3895
    PAGE 298  ROUTINE HC.COMPUTE.TIMES               3918 3960
    PAGE 450  PROCESS HC.RETURN.FARRP                 481  496
    PAGE 452                                          598
    PAGE 453                                          635
ATK.UNITS.PTR
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES       1610
    PAGE  95  ROUTINE INIT.REINF                     4653
    PAGE 118  ROUTINE REIN.ARRIVE                    5615
    PAGE 688  ''PROGRAM  OLDER.VERSION               1049
ATM.DELIV.CEP
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES        296
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2837
    PAGE 588  ROUTINE AC.MUNS.INPUT                  7025
    PAGE 665  ''PROGRAM  OLDER.VERSION               9738
ATTACKS.
    PAGE 507  PROCESS CAS.MISSION                    3552
ATTACK.
    PAGE 412  EVENT ACT.ATK                          8608
    PAGE 506  PROCESS CAS.MISSION                    3491
    PAGE 544  ROUTINE ORD.REINF                      5165
```

```
ATTRIT.SENSOR
    PAGE   3  PROGRAM REVISIONS                    175
    PAGE 159  ROUTINE ATTRIT.SENSOR                7311 7344
    PAGE 253  ROUTINE MINE.EFFECTS                 1655
    PAGE 271  ROUTINE BTRY.EFFECTS                 2629
    PAGE 279  ROUTINE AC.BOMB.EFFECTS              2974
    PAGE 384  EVENT OFF.LINE.ATTRITION             7433
AU.LIST
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES      572
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES     1151
    PAGE 134  ROUTINE CHECK.FOR.MINES              6248 6254
    PAGE 143  ROUTINE DEAD.UNIT                    6605 6608
    PAGE 405  EVENT START.MOVE                     8311
    PAGE 410  EVENT UPDATE.LOC                     8531 8566
    PAGE 670  ''PROGRAM  OLDER.VERSION               13
    PAGE 680                                        590
AU.UNIT.ID
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES     1149
    PAGE 134  ROUTINE CHECK.FOR.MINES              6249
    PAGE 143  ROUTINE DEAD.UNIT                    6609 6611
    PAGE 405  EVENT START.MOVE                     8310
    PAGE 410  EVENT UPDATE.LOC                     8565
    PAGE 680  ''PROGRAM  OLDER.VERSION              588
AVAILABLE.
    PAGE 544  ROUTINE ORD.REINF                    5169
AVAIL.AO.LIST
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES      756
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES     1847
    PAGE  49  ''SECTION FOR DEFINITIONS            2796
    PAGE 159  ROUTINE ATTRIT.SENSOR                7358
    PAGE 160                                       7369
    PAGE 371  EVENT FEBA.SORTIE                    6790
    PAGE 372                                       6833
    PAGE 566  ROUTINE SENSOR.INPUT                 6083
    PAGE 673  ''PROGRAM  OLDER.VERSION              198
    PAGE 692                                       1286
    PAGE 708                                       2217
AVER.APPAR.SIZE
    PAGE 154  ROUTINE AO.DETECTION                 7092
    PAGE 156                                       7239 7240 7245
AVER.PROB.LOS
    PAGE 154  ROUTINE AO.DETECTION                 7091
    PAGE 156                                       7227 7228 7243
AVG.APP.SIZE
    PAGE 156  ROUTINE AO.DETECTION                 7246
AV.PROJ
    PAGE 269  ROUTINE BTRY.EFFECTS                 2498 2505
AV.TGTS
    PAGE 269  ROUTINE BTRY.EFFECTS                 2496 2498 2499 2503 2508 2512
AWARE.UNIT
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES     1148
    PAGE 405  EVENT START.MOVE                     8309 8310 8311
    PAGE 410  EVENT UPDATE.LOC                     8564 8565 8566
    PAGE 680  ''PROGRAM  OLDER.VERSION              587
    PAGE 712                                       2483
```

```
A.FLASH
    PAGE 246  ROUTINE VOLLEY                          1314 1344 1348
A.MIN.SEG
    PAGE 106  ROUTINE MIN.MOVE                        5076 5077 5078
A.SOUND
    PAGE 246  ROUTINE VOLLEY                          1314 1349 1353
A.TF.LDR
    PAGE 112  ROUTINE PREPARE.LIST                    5374 5375
A.TF.MEM
    PAGE 137  ROUTINE CHECK.LIST                      6358 6360 6363 6368
A.UNPCT.OPEN
    PAGE 245  ROUTINE UNIT.ENVIR                      1291 1297
A.UNPCT.TOWN
    PAGE 245  ROUTINE UNIT.ENVIR                      1293 1297
A.UNPCT.WOOD
    PAGE 245  ROUTINE UNIT.ENVIR                      1292 1297
BACK.DIST
    PAGE 428  PROCESS AIR.OBSERVER                    9348
    PAGE 430                                          9489 9493 9498
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE            1370
    PAGE 468                                          1419 1423 1433
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                 9485 9529
    PAGE 659                                          9533
BAD.
    PAGE 353  EVENT AD.ENGAGEMENT                     6138
BAND.RANGE
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES         849
    PAGE  44  ''SECTION FOR DEFINITIONS               2515
    PAGE  96  ROUTINE LINE.OF.SIGHT                   4705
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION           815
    PAGE 571  ROUTINE TT.FACTORS.INPUT                6295
    PAGE 674  ''PROGRAM  OLDER.VERSION                 291
    PAGE 703                                          1936
BASES.
    PAGE 310  ROUTINE AD.SHOOT                        4482
BASE.
    PAGE 427  PROCESS AC.ATK.TGT                      9322
BATTERY.
    PAGE 483  PROCESS FIRE.MISSION                    2225
    PAGE 550  ROUTINE BTRY.INPUT                      5392
BATTLE
    PAGE   1  ROUTINE FOR CROSS_REFERENCING             48
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES        1153
    PAGE  26                                          1470
    PAGE  42  ''SECTION FOR DEFINITIONS               2381
    PAGE  58  ROUTINE CREATE.FORCE                    3091 3110 3139
    PAGE  71  ROUTINE ORIENTATION                     3651 3653 3653 3655 3655 3691
    PAGE  72                                          3725
    PAGE  85  ROUTINE CHANGE.LOC                      4268
    PAGE  93  ROUTINE INITIAL.DETECT                  4589 4597
    PAGE  94  ROUTINE INITIAL.MOVE                    4615 4620 4622
    PAGE  96  ROUTINE LINE.OF.SIGHT                   4671
    PAGE 104  ROUTINE MINE.DELAY                      5026
    PAGE 107  ROUTINE NEW.SEGMENT                     5151
    PAGE 108                                          5184
    PAGE 113  ROUTINE PREPARE.LIST                    5444
```

```
PAGE 115  ROUTINE PROX.CHECK                    5504 5548
PAGE 118  ROUTINE REIN.ARRIVE                   5608
PAGE 128  ROUTINE BTL.CHECK                     5992
PAGE 133  ROUTINE CHECK.FORCE                   6185
PAGE 146  ROUTINE INTER.BATTLE                  6739 6742 6743 6744 6745 6752 6753 6756 6787 6790 6793 6794 6795 6796
PAGE 147                                        6816
PAGE 200  ROUTINE ILLUM.EFFECTS                 9196 9201
PAGE 234  ROUTINE SMOKE.EFFECTS                  760
PAGE 238  ROUTINE SWITCH.FO                      915  923  925  944  960
PAGE 251  ROUTINE MINE.EFFECTS                  1551
PAGE 255  ROUTINE FO.DETECTION                  1735
PAGE 289  ROUTINE EMPLOY.HELICOPTERS            3476 3483 3490 3491 3530
PAGE 297  ROUTINE FARRP.CHECK                   3864 3866 3905
PAGE 298  ROUTINE HC.COMPUTE.TIMES              3916 3925 3929 3933
PAGE 299                                        3971
PAGE 302  ROUTINE HEL.RANGE.COMPUTE             4143
PAGE 324  ROUTINE DECIDE                        5068
PAGE 328  ROUTINE EMPTY                         5167 5179 5184 5191 5198 5209 5218
PAGE 329                                        5226 5233
PAGE 330                                        5282 5292 5296 5297
PAGE 337  ROUTINE HC.EMPTY                      5464 5471
PAGE 356  EVENT BTL.ENDED                       6256 6268 6280 6292 6304 6305
PAGE 357                                        6312 6313 6320 6321 6324
PAGE 369  EVENT ENGAGEMENT                      6694
PAGE 370                                        6748
PAGE 391  EVENT SEND.TEAM                       7602
PAGE 395  EVENT START.BATTLE                    7778 7786 7793 7794 7796 7798 7800
PAGE 396                                        7801 7802 7803 7808 7814 7846 7849 7853 7856 7858
PAGE 397                                        7859 7862 7901 7902 7903 7905 7905 7908
PAGE 398                                        7941 7942 7944 7974
PAGE 400                                        8051 8077 8080 8083
PAGE 401                                        8091 8092 8096 8102 8103 8107 8108 8109 8113 8116 8120 8126 8127 8131 8132
                                                8133 8137 8146 8147
PAGE 402                                        8151 8152 8155
PAGE 412  EVENT ACT.ATK                         8610 8624 8626 8627
PAGE 420  PROCESS AC.ATK.TGT                    8957 8958 8960
PAGE 438  PROCESS FORWARD.OBSERVER              9890
PAGE 439                                        9931 9932 9963 9965
PAGE 440                                        9971 9976 9980 9982
PAGE 443  PROCESS HC.ARRIVE.BATTLE               108  112  113  118  121  124  144  149  150
PAGE 444                                         151  156  158  159  161  162  166  200
PAGE 445                                         226  231  258  260
PAGE 448                                         405  417  419  421
PAGE 449                                         443  445  447
PAGE 450  PROCESS HC.RETURN.FARRP                467  506
PAGE 452                                         578  579
PAGE 453                                         650
PAGE 464  PROCESS MINE.ASSESS                   1224
PAGE 465                                        1307
PAGE 471  PROCESS TARGET.REPORT                 1562
PAGE 478  PROCESS WITH.DRAW                     1939
PAGE 505  PROCESS CAS.MISSION                   3453
PAGE 510  PROCESS HELICOPTER.FIRE               3715
PAGE 609  ROUTINE KV.SCOREBOARD                 7882
PAGE 618  ROUTINE SNAP.R                        8257
PAGE 619                                        8259 8260
```

```
    PAGE 680  ''PROGRAM  OLDER.VERSION                592
    PAGE 685                                          909
    PAGE 701                                         1811
BATTLE.
    PAGE  71  ROUTINE ORIENTATION                    3635 3685
    PAGE  72                                         3695 3698 3699 3700 3708 3709 3710
    PAGE  73                                         3753 3754 3755 3756 3757 3758 3770 3771 3772
    PAGE  77  ROUTINE ADJUST                         3887
    PAGE 112  ROUTINE PREPARE.LIST                   5354
    PAGE 125  ROUTINE WHAT.NEXT                      5880
    PAGE 137  ROUTINE CHECK.LIST                     6336
    PAGE 146  ROUTINE INTER.BATTLE                   6743
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS             3480
    PAGE 298  ROUTINE HC.COMPUTE.TIMES               3926
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                 99
    PAGE 450  PROCESS HC.RETURN.FARRP                 468
BATTLE.NUM
    PAGE 138  ROUTINE CHECK.PROX                     6374
    PAGE 139                                         6470 6472 6475
    PAGE 347  EVENT ACT.REINF                        5824 5853
    PAGE 405  EVENT START.MOVE                       8335 8337 8341
    PAGE 409  EVENT UPDATE.LOC                       8497 8505 8509
BATTLE.SET
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        829
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES       1169
    PAGE 128  ROUTINE BTL.CHECK                      6006 6025
    PAGE 330  ROUTINE EMPTY                          5296
    PAGE 396  EVENT START.BATTLE                     7803
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                 149
    PAGE 452  PROCESS HC.RETURN.FARRP                  610
    PAGE 618  ROUTINE SNAP.R                         8257
    PAGE 674  ''PROGRAM  OLDER.VERSION                 271
    PAGE 680                                           608
BATTLE.SPECIFICALLY
    PAGE  71  ROUTINE ORIENTATION                    3651
BATTLE.STATUS
    PAGE 328  ROUTINE EMPTY                          5177
    PAGE 330                                         5288 5290 5292
BATTLE...THE
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4141
BAT.NUM
    PAGE 184  ROUTINE FA.BN.ASGN                     8448 8455 8455
    PAGE 185                                         8480 8503
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          8991 9022 9045
    PAGE 198                                         9127
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9303 9315 9352
    PAGE 636  FUNCTION HE.WLA                        8859 8871 8900
BAT.TYPE
    PAGE   2  PROGRAM REVISIONS                       117
BC.FARRP
    PAGE 128  ROUTINE BTL.CHECK                      5986 5996 6001 6011 6030
    PAGE 129                                         6046 6049 6052 6056 6060
BC.SIDE
    PAGE 128  ROUTINE BTL.CHECK                      5996 6005
    PAGE 129                                         6047
```

```
BEST.HE
     PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9043
     PAGE 197                                         9084
     PAGE 198                                         9107
BEST.ICM
     PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9038
     PAGE 197                                         9064 9099
BEST.LETH.AREA
     PAGE 248  ROUTINE WEIGHTED.VOLLEYS               1371 1388 1394
     PAGE 249                                         1447
BEST.MUNITION
     PAGE 248  ROUTINE WEIGHTED.VOLLEYS               1369 1394 1403
     PAGE 249                                         1425 1427 1429 1432 1434 1436
BEST.TIME
     PAGE 195  ROUTINE FIND.START.TIME                8955 8983 8987
BEST.USAGE
     PAGE 195  ROUTINE FIND.START.TIME                8954 8956 8982 8984
BEST.WLA
     PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9002
     PAGE 197                                         9102
     PAGE 198                                         9109 9126
     PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9302 9311 9315 9336 9354 9357
     PAGE 204                                         9367
BEST.WLA.HE
     PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9001 9044
     PAGE 197                                         9050 9080 9086
     PAGE 198                                         9109
BEST.WLA.ICM
     PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9001 9039
     PAGE 197                                         9050 9056 9066 9102
BETA.F
     PAGE  98  ROUTINE LOCATE.SEARCH.AREA             4791 4802
     PAGE 104  ROUTINE MINE.DELAY                     5004
     PAGE 361  EVENT CFR.ON                           6467
BETWEEN.V
     PAGE 392  EVENT SET.DEBUG                        7634 7650
     PAGE 520  ROUTINE MAIN2                          4190
BINOMIAL.F
     PAGE 157  ROUTINE AO.DETECTION                   7257
     PAGE 269  ROUTINE BTRY.EFFECTS                   2541
     PAGE 278  ROUTINE AC.BOMB.EFFECTS                2929 2957
     PAGE 351  EVENT AD.ENGAGEMENT                    6024
     PAGE 436  PROCESS ARTY.ASSESS                    9767
     PAGE 451  PROCESS HC.RETURN.FARRP                 526  540
     PAGE 504  PROCESS CAS.MISSION                    3401
     PAGE 656  ROUTINE AR.DETECTION                   9420
BLUE.AH.TEAM
     PAGE  52  ''SECTION FOR SUBSTITUTIONS            2950
     PAGE 710  ''PROGRAM  OLDER.VERSION               2357
     PAGE 714  ROUTINE PLAT.COUNT                     2535
BLUE.CO.HQ
     PAGE  51  ''SECTION FOR SUBSTITUTIONS            2946
     PAGE 569  ROUTINE TBF.INPUT                      6247
     PAGE 710  ''PROGRAM  OLDER.VERSION               2353
     PAGE 714  ROUTINE PLAT.COUNT                     2539
```

```
BLUE.HB.PRIORITY
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES          1163
    PAGE 128  ROUTINE BTL.CHECK                         6007 6018 6020
    PAGE 129                                            6048
    PAGE 311  ROUTINE INTER.HELO                        4530
    PAGE 396  EVENT START.BATTLE                        7849 7858
    PAGE 401                                            8092 8109
    PAGE 448  PROCESS HC.ARRIVE.BATTLE                   419
    PAGE 680  ''PROGRAM  OLDER.VERSION                   602
BLUE.INF.PLT
    PAGE  51  ''SECTION FOR SUBSTITUTIONS               2948
    PAGE 569  ROUTINE TBF.INPUT                         6245
    PAGE 710  ''PROGRAM  OLDER.VERSION                  2355
    PAGE 714  ROUTINE PLAT.COUNT                        2523
BLUE.MECH.PLT
    PAGE  51  ''SECTION FOR SUBSTITUTIONS               2945
    PAGE 569  ROUTINE TBF.INPUT                         6243
    PAGE 710  ''PROGRAM  OLDER.VERSION                  2352
    PAGE 714  ROUTINE PLAT.COUNT                        2527
BLUE.MISSION
    PAGE  41  ''SECTION FOR EVENTS                      2313
    PAGE 395  EVENT START.BATTLE                        7746
    PAGE 396                                            7811
    PAGE 397                                            7866 7885
    PAGE 398                                            7950
    PAGE 399                                            8007 8008 8009
    PAGE 700  ''PROGRAM  OLDER.VERSION                  1750
BLUE.N.KV
    PAGE 529  ROUTINE KV.INPUT                          4545 4551 4558 4560
    PAGE 609  ROUTINE KV.SCOREBOARD                     7845 7847 7850 7852
BLUE.TK.PLT
    PAGE  51  ''SECTION FOR SUBSTITUTIONS               2943
    PAGE 569  ROUTINE TBF.INPUT                         6241
    PAGE 710  ''PROGRAM  OLDER.VERSION                  2350
    PAGE 714  ROUTINE PLAT.COUNT                        2531
BLUE.UNITS
    PAGE  41  ''SECTION FOR EVENTS                      2315
    PAGE  67  ROUTINE GENERAL.BATTLE                    3450 3455 3458 3485 3486
    PAGE  68                                            3520 3526 3530 3531 3534 3541
    PAGE  96  ROUTINE LINE.OF.SIGHT                     4666 4676 4679 4687 4692 4698 4718 4720
    PAGE  97                                            4727 4744 4749
    PAGE 395  EVENT START.BATTLE                        7748 7758 7762 7770 7771
    PAGE 396                                            7801 7810 7822
    PAGE 397                                            7898
    PAGE 398                                            7925
    PAGE 399                                            8023 8031
    PAGE 400                                            8073
    PAGE 700  ''PROGRAM  OLDER.VERSION                  1752
BL.ATK.FAIL.PROB
    PAGE  48  ''SECTION FOR DEFINITIONS                 2761
    PAGE 450  PROCESS HC.RETURN.FARRP                    512
    PAGE 572  ROUTINE FARRP.INPUT                       6366
    PAGE 573                                            6391
    PAGE 707  ''PROGRAM  OLDER.VERSION                  2183
BL.HC.SPACING
    PAGE  42  ''SECTION FOR DEFINITIONS                 2394
```

```
    PAGE 447  PROCESS HC.ARRIVE.BATTLE                359
    PAGE 572  ROUTINE FARRP.INPUT                     6356 6381
    PAGE 574                                          6444 6444
    PAGE 701  ''PROGRAM  OLDER.VERSION                1824
BL.HIGH.FRAC.RANGE
    PAGE  49  ''SECTION FOR DEFINITIONS               2780
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE               4173
    PAGE 447  PROCESS HC.ARRIVE.BATTLE                358
    PAGE 572  ROUTINE FARRP.INPUT                     6364
    PAGE 573                                          6390
    PAGE 707  ''PROGRAM  OLDER.VERSION                2202
BL.LOW.FRAC.RANGE
    PAGE  49  ''SECTION FOR DEFINITIONS               2778
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE               4172
    PAGE 447  PROCESS HC.ARRIVE.BATTLE                357
    PAGE 572  ROUTINE FARRP.INPUT                     6364
    PAGE 573                                          6389
    PAGE 707  ''PROGRAM  OLDER.VERSION                2200
BL.MAX.FL.TIME
    PAGE  42  ''SECTION FOR DEFINITIONS               2392
    PAGE 299  ROUTINE HC.COMPUTE.TIMES                3974
    PAGE 572  ROUTINE FARRP.INPUT                     6355 6380
    PAGE 701  ''PROGRAM  OLDER.VERSION                1822
BL.MAX.HANDOFF.TIME
    PAGE  48  ''SECTION FOR DEFINITIONS               2768
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION           729
    PAGE 572  ROUTINE FARRP.INPUT                     6358 6384
    PAGE 707  ''PROGRAM  OLDER.VERSION                2190
BL.MAX.MASK.TIME
    PAGE  48  ''SECTION FOR DEFINITIONS               2772
    PAGE 461  PROCESS HEL.TARGET.ACQUISITION          1136
    PAGE 572  ROUTINE FARRP.INPUT                     6360
    PAGE 573                                          6386
    PAGE 707  ''PROGRAM  OLDER.VERSION                2194
BL.MAX.UNMASK.TIME
    PAGE  49  ''SECTION FOR DEFINITIONS               2776
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION           732
    PAGE 572  ROUTINE FARRP.INPUT                     6362
    PAGE 573                                          6388
    PAGE 707  ''PROGRAM  OLDER.VERSION                2198
BL.MIN.HANDOFF.TIME
    PAGE  48  ''SECTION FOR DEFINITIONS               2766
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION           729
    PAGE 572  ROUTINE FARRP.INPUT                     6358 6383
    PAGE 707  ''PROGRAM  OLDER.VERSION                2188
BL.MIN.MASK.TIME
    PAGE  48  ''SECTION FOR DEFINITIONS               2770
    PAGE 461  PROCESS HEL.TARGET.ACQUISITION          1136
    PAGE 572  ROUTINE FARRP.INPUT                     6360
    PAGE 573                                          6385
    PAGE 707  ''PROGRAM  OLDER.VERSION                2192
BL.MIN.UNMASK.TIME
    PAGE  49  ''SECTION FOR DEFINITIONS               2774
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION           731
    PAGE 572  ROUTINE FARRP.INPUT                     6362
    PAGE 573                                          6387
```

```
    PAGE 707  ''PROGRAM  OLDER.VERSION                2196
BL.PRIORITY
    PAGE 396  EVENT START.BATTLE                      7848 7849
BL.ROUNDS.PER.POPUP
    PAGE  42  ''SECTION FOR DEFINITIONS               2389
    PAGE 510  PROCESS HELICOPTER.FIRE                 3698
    PAGE 572  ROUTINE FARRP.INPUT                     6368
    PAGE 573                                          6393
    PAGE 701  ''PROGRAM  OLDER.VERSION                1819
BL.SCT.FAIL.PROB
    PAGE  48  ''SECTION FOR DEFINITIONS               2762
    PAGE 450  PROCESS HC.RETURN.FARRP                  513
    PAGE 572  ROUTINE FARRP.INPUT                     6366
    PAGE 573                                          6392
    PAGE 707  ''PROGRAM  OLDER.VERSION                2184
BL.STATUS
    PAGE 147  ROUTINE INTER.BATTLE                    6846 6847
    PAGE 148                                          6883 6884
    PAGE 357  EVENT BTL.ENDED                         6327 6332
BL.UNITS
    PAGE 146  ROUTINE INTER.BATTLE                    6748 6753
    PAGE 147                                          6818 6822 6823 6825 6826 6853
    PAGE 148                                          6883
    PAGE 356  EVENT BTL.ENDED                         6267 6270 6296
    PAGE 357                                          6326
    PAGE 413  EVENT ACT.ATK                           8685 8688 8689
BL.UNITS.
    PAGE 356  EVENT BTL.ENDED                         6253 6262 6264 6302 6310
    PAGE 357                                          6318
    PAGE 412  EVENT ACT.ATK                           8617
    PAGE 413                                          8657 8671 8675 8706 8708
BL.WIN
    PAGE 356  EVENT BTL.ENDED                         6299 6307
    PAGE 357                                          6315 6333
BMNT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING            54
    PAGE  48  ''SECTION FOR DEFINITIONS               2724
    PAGE  57  ROUTINE MAIN3                           3079
    PAGE  80  ROUTINE BLOCK.LOS                       4024
    PAGE  83  ROUTINE CHANGE.LOC                      4150
    PAGE 106  ROUTINE MIN.MOVE                        5085
    PAGE 150  ROUTINE PK.COMPUTE                      7000
    PAGE 365  EVENT CHANGE.LITE                       6606 6607 6609
    PAGE 523  ROUTINE SYS.INPUT                       4352
    PAGE 706  ''PROGRAM  OLDER.VERSION                2146
BN.BTRY.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES         337
    PAGE   8                                           432
    PAGE  49  ''SECTION FOR DEFINITIONS               2797
    PAGE  89  ROUTINE FA.BN.MOVEMENT                  4422
    PAGE 182  ROUTINE FA.BN.ASGN                      8325
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9514
    PAGE 258  ROUTINE FO.DETECTION                    1909
    PAGE 362  EVENT CFR.OPERATOR                      6534
    PAGE 551  ROUTINE BTRY.INPUT                      5458
    PAGE 666  ''PROGRAM  OLDER.VERSION                9779
```

```
    PAGE 667                                          9873
    PAGE 708                                          2218
BN.CAN.FM.SET
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES         828
    PAGE  35  ''SECTION FOR PROCESSES                 2012
    PAGE  49  ''SECTION FOR DEFINITIONS               2798
    PAGE 184  ROUTINE FA.BN.ASGN                      8427 8434 8449 8454
    PAGE 674  ''PROGRAM  OLDER.VERSION                 270
    PAGE 694                                          1450
    PAGE 708                                          2219
BN.LINK
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9510 9513
    PAGE 475  PROCESS TARGET.REPORT                   1800 1817
BN.REM.EFFECTS
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS                1389 1416 1417 1418 1419 1420
    PAGE 249                                          1446 1449 1452
BOMBS.
    PAGE 420  PROCESS AC.ATK.TGT                      8923
BREAK.OFF.RANGE
    PAGE 115  ROUTINE PROX.CHECK                      5532 5535
BREAK.POINT
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         730
    PAGE  41  ''SECTION FOR DEFINITIONS               2360
    PAGE 130  ROUTINE CHECK.DEAD                      6090
    PAGE 133  ROUTINE CHECK.FORCE                     6173
    PAGE 138  ROUTINE CHECK.PROX                      6430
    PAGE 139                                          6447
    PAGE 145  ROUTINE FIN.BATTLE                      6717
    PAGE 324  ROUTINE DECIDE                          5057
    PAGE 523  ROUTINE SYS.INPUT                       4346
    PAGE 672  ''PROGRAM  OLDER.VERSION                 173
    PAGE 700                                          1790
BTL.
    PAGE 149  ROUTINE PK.COMPUTE                      6909 6911 6912 6917 6923 6932
BTL.BL.FARRP
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES        1166
    PAGE 129  ROUTINE BTL.CHECK                       6049
    PAGE 311  ROUTINE INTER.HELO                      4528 4539
    PAGE 356  EVENT BTL.ENDED                         6271 6272
    PAGE 401  EVENT START.BATTLE                      8107
    PAGE 402                                          8151
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                 159
    PAGE 452  PROCESS HC.RETURN.FARRP                  620  622
    PAGE 453                                           652
    PAGE 680  ''PROGRAM  OLDER.VERSION                 605
BTL.BL.HC.TEAM
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES        1164
    PAGE 311  ROUTINE INTER.HELO                      4532
    PAGE 375  EVENT HC.DEPART.BATTLE                  7007
    PAGE 402  EVENT START.BATTLE                      8152
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                 158
    PAGE 452  PROCESS HC.RETURN.FARRP                  621
    PAGE 488  PROCESS ASSESSMENT                      2487
    PAGE 680  ''PROGRAM  OLDER.VERSION                 603
BTL.BL.UNITS
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES        1159
```

```
    PAGE 145  ROUTINE FIN.BATTLE                    6699
    PAGE 146  ROUTINE INTER.BATTLE                  6753
    PAGE 396  EVENT START.BATTLE                    7801
    PAGE 449  PROCESS HC.ARRIVE.BATTLE               447
    PAGE 680  ''PROGRAM  OLDER.VERSION               598
BTL.BU
    PAGE  39  ''SECTION FOR EVENTS                  2235
    PAGE 698  ''PROGRAM  OLDER.VERSION              1672
BTL.CHECK
    PAGE 128  ROUTINE BTL.CHECK                     5984 6002
    PAGE 129                                        6061
    PAGE 452  PROCESS HC.RETURN.FARRP                631
BTL.ENDED
    PAGE   2  PROGRAM REVISIONS                       87
    PAGE  39  ''SECTION FOR EVENTS                  2233
    PAGE 130  ROUTINE CHECK.DEAD                    6074
    PAGE 145  ROUTINE FIN.BATTLE                    6736
    PAGE 356  EVENT BTL.ENDED                       6253 6280 6292
    PAGE 601  ROUTINE BETWEEN.ROUTINE               7560 7561
    PAGE 616  ROUTINE SNAP2                         8170
    PAGE 698  ''PROGRAM  OLDER.VERSION              1670
BTL.FIELD
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1157 1157
    PAGE  72  ROUTINE ORIENTATION                   3695 3698 3699 3700 3708 3709 3710
    PAGE  73                                        3753 3754 3755 3756 3757 3758 3770 3771 3772
    PAGE 397  EVENT START.BATTLE                    7902
    PAGE 680  ''PROGRAM  OLDER.VERSION               596  596
BTL.FORCE.SET
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1171
    PAGE  25                                        1414
    PAGE  58  ROUTINE CREATE.FORCE                  3110
    PAGE  71  ROUTINE ORIENTATION                   3665
    PAGE  93  ROUTINE INITIAL.DETECT                4597
    PAGE  94  ROUTINE INITIAL.MOVE                  4622
    PAGE 101  ROUTINE LOS.CHECK                     4877
    PAGE 133  ROUTINE CHECK.FORCE                   6159 6200
    PAGE 145  ROUTINE FIN.BATTLE                    6702
    PAGE 146  ROUTINE INTER.BATTLE                  6796
    PAGE 173  ROUTINE DUST.EFFECTS                  7902 7923 7929
    PAGE 200  ROUTINE ILLUM.EFFECTS                 9223 9230
    PAGE 216  ROUTINE REQUEST.FASCAM                9932
    PAGE 218  ROUTINE REQUEST.ILLUM                 9999    6
    PAGE 219                                          62
    PAGE 222  ROUTINE REQUEST.SMOKE                  188  194
    PAGE 223                                         235
    PAGE 226  ROUTINE REQUEST.WD.FASCAM              406  412
    PAGE 234  ROUTINE SMOKE.EFFECTS                  789
    PAGE 235                                         813  820
    PAGE 236                                         889
    PAGE 238  ROUTINE SWITCH.FO                      925
    PAGE 283  ROUTINE CAS.EVAL                      3182 3220
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS         3409
    PAGE 298  ROUTINE HC.COMPUTE.TIMES              3933
    PAGE 305  ROUTINE UNIT.PRIORITY                 4248 4275
    PAGE 329  ROUTINE EMPTY                         5233
    PAGE 330                                        5282
```

```
    PAGE 345  ROUTINE TERM.CHECK                    5772
    PAGE 398  EVENT START.BATTLE                    7944
    PAGE 400                                        8051
    PAGE 439  PROCESS FORWARD.OBSERVER              9932
    PAGE 444  PROCESS HC.ARRIVE.BATTLE               166  200
    PAGE 448                                         405
    PAGE 478  PROCESS WITH.DRAW                     1953
    PAGE 489  PROCESS ASSESSMENT                    2534
    PAGE 508  PROCESS CAS.MISSION                   3607
    PAGE 619  ROUTINE SNAP.R                        8260
    PAGE 680  ''PROGRAM  OLDER.VERSION               610
    PAGE 684                                         853
BTL.NUMBER
    PAGE 115  ROUTINE PROX.CHECK                    5526 5527
BTL.RD.FARRP
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1167
    PAGE 129  ROUTINE BTL.CHECK                     6052
    PAGE 311  ROUTINE INTER.HELO                    4527 4537
    PAGE 356  EVENT BTL.ENDED                       6283 6284
    PAGE 401  EVENT START.BATTLE                    8131 8146
    PAGE 444  PROCESS HC.ARRIVE.BATTLE               162
    PAGE 452  PROCESS HC.RETURN.FARRP                615  617
    PAGE 453                                         654
    PAGE 680  ''PROGRAM  OLDER.VERSION               606
BTL.RD.HC.TEAM
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1165
    PAGE 311  ROUTINE INTER.HELO                    4531
    PAGE 375  EVENT HC.DEPART.BATTLE                7009
    PAGE 401  EVENT START.BATTLE                    8147
    PAGE 444  PROCESS HC.ARRIVE.BATTLE               161
    PAGE 452  PROCESS HC.RETURN.FARRP                616
    PAGE 488  PROCESS ASSESSMENT                    2485
    PAGE 680  ''PROGRAM  OLDER.VERSION               604
BTL.RD.UNITS
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1160
    PAGE 145  ROUTINE FIN.BATTLE                    6700
    PAGE 146  ROUTINE INTER.BATTLE                  6752
    PAGE 398  EVENT START.BATTLE                    7802
    PAGE 449  PROCESS HC.ARRIVE.BATTLE               445
    PAGE 680  ''PROGRAM  OLDER.VERSION               599
BTL.RU
    PAGE  39  ''SECTION FOR EVENTS                  2236
    PAGE 698  ''PROGRAM  OLDER.VERSION              1673
BTL.SEQ.NO
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1155
    PAGE 115  ROUTINE PROX.CHECK                    5544
    PAGE 133  ROUTINE CHECK.FORCE                   6181
    PAGE 146  ROUTINE INTER.BATTLE                  6793
    PAGE 311  ROUTINE INTER.HELO                    4525
    PAGE 356  EVENT BTL.ENDED                       6265
    PAGE 395  EVENT START.BATTLE                    7794 7798
    PAGE 398                                        7941
    PAGE 401                                        8102 8126
    PAGE 478  PROCESS WITH.DRAW                     1936
    PAGE 515  PROCESS HELICOPTER.FIRE               3970 3995
    PAGE 517                                        4074
```

```
    PAGE 518                                        4133
    PAGE 680  ''PROGRAM  OLDER.VERSION               594
BTL.TERRAIN.TYPE
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1156
    PAGE  80  ROUTINE BLOCK.LOS                     4023
    PAGE  83  ROUTINE CHANGE.LOC                    4149
    PAGE 101  ROUTINE LOS.CHECK                     4894 4896 4901 4903
    PAGE 106  ROUTINE MIN.MOVE                      5083
    PAGE 107  ROUTINE NEW.SEGMENT                   5123 5126 5131 5134
    PAGE 109                                        5241 5242 5245 5246
    PAGE 115  ROUTINE PROX.CHECK                    5527
    PAGE 146  ROUTINE INTER.BATTLE                  6756
    PAGE 149  ROUTINE PK.COMPUTE                    6917 6923 6932
    PAGE 395  EVENT START.BATTLE                    7800
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION         713
    PAGE 478  PROCESS WITH.DRAW                     1961
    PAGE 479                                        1992
    PAGE 680  ''PROGRAM  OLDER.VERSION               595
BTL.TIME.OF.DAY
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1154
    PAGE 311  ROUTINE INTER.HELO                    4526
    PAGE 356  EVENT BTL.ENDED                       6266
    PAGE 395  EVENT START.BATTLE                    7793 7796
    PAGE 680  ''PROGRAM  OLDER.VERSION               593
BTL.UNITS.
    PAGE 125  ROUTINE WHAT.NEXT                     5876 5885 5890 5892 5893 5928 5930
    PAGE 126                                        5931 5933 5950 5951 5954 5956 5963 5969 5969
BTL.WIDTH
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1161
    PAGE 397  EVENT START.BATTLE                    7903
    PAGE 680  ''PROGRAM  OLDER.VERSION               600
BTL.WINNER
    PAGE  39  ''SECTION FOR EVENTS                  2234
    PAGE 698  ''PROGRAM  OLDER.VERSION              1671
BTRY
    PAGE   1  ROUTINE FOR CROSS_REFERENCING          44
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       320
    PAGE  17                                         928
    PAGE  18                                         985
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES      1358 1359
    PAGE  26                                        1487
    PAGE  89  ROUTINE FA.BN.MOVEMENT                4408
    PAGE  90                                        4471 4490
    PAGE  91                                        4549 4563
    PAGE  92                                        4580
    PAGE 163  ROUTINE BTRY.FM.DEQ                   7521 7523 7525 7526 7528 7528 7529 7533 7534 7540 7541
    PAGE 164  ROUTINE BTRY.FM.ENQ                   7544 7551 7552 7554 7558 7559 7563
    PAGE 166  ROUTINE CFR.DETECTION                 7620 7624 7631 7635 7636 7637 7673
    PAGE 167                                        7689 7692
    PAGE 178  ROUTINE EST.COVERAGE                  8174
    PAGE 181  ROUTINE FASCAM.COMPUTATION            8281
    PAGE 182  ROUTINE FA.BN.ASGN                    8303 8325 8327 8329 8330 8334 8334 8336 8336 8337 8341 8342 8343
    PAGE 183                                        8345 8346 8346 8348 8354 8355 8358 8365 8366 8370 8375 8379 8394 8395
    PAGE 184                                        8403 8408 8410 8416 8417 8419 8421 8424 8425 8429 8435 8440 8441
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION         9004 9007 9019 9027 9040 9045
    PAGE 197                                        9063 9083
```

```
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9175
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9310 9320 9323 9323 9325 9354
    PAGE 206  ROUTINE PDB.DETECTION                  9431 9438 9455 9458
    PAGE 208  ROUTINE PGM.MSN.ASGN                   9502 9514 9515 9516 9517 9517 9518 9519 9520 9525 9528 9529 9530 9533 9534
                                                     9538 9543
    PAGE 209                                         9547 9558 9561 9563 9566 9567 9568 9576 9581 9582 9583 9584 9584 9588 9588
                                                     9588 9589 9590 9601 9604
    PAGE 211                                         9678
    PAGE 233  ROUTINE SMOKE.COMPUTATION               732
    PAGE 246  ROUTINE VOLLEY                         1309 1315 1319 1322 1342 1347 1352 1357 1361
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS               1368 1394 1396 1415 1415 1423
    PAGE 260  ROUTINE BTRY.EFFECTS                   1985 2000 2001 2008 2008 2010 2010
    PAGE 262                                         2099 2110 2110 2112
    PAGE 263                                         2142 2143
    PAGE 270                                         2562
    PAGE 271                                         2646
    PAGE 272                                         2671 2674 2674 2710
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS         2736 2739 2741 2746
    PAGE 355  EVENT ARTY.OCCUPATION                  6240
    PAGE 358  EVENT CFR.ACTIVATION                   6342 6348 6350 6356 6368 6373
    PAGE 362  EVENT CFR.OPERATOR                     6484 6504 6505 6518 6520 6525
    PAGE 363                                         6555
    PAGE 386  EVENT PDB.ACTIVATION                   7468 7474 7477 7483 7494 7498
    PAGE 387  EVENT PDB.OPERATOR                     7510 7523 7524 7536 7538 7542 7543
    PAGE 393  EVENT START.ARTY.MOVEMENT              7657 7668 7675 7678 7680 7685 7686 7688 7691 7704 7705 7705
    PAGE 394                                         7724 7724 7725 7728 7729 7730 7735 7739
    PAGE 407  EVENT STOP.ARTY.MOVEMENT               8372 8380 8389 8389 8390 8393 8394 8396 8399 8400
    PAGE 463  PROCESS HOW.REPAIR                     1177 1179 1180 1181 1182 1183 1196 1212
    PAGE 483  PROCESS FIRE.MISSION                   2225
    PAGE 550  ROUTINE BTRY.INPUT                     5404 5409 5415 5417 5440 5441 5443
    PAGE 551                                         5450 5455 5456 5456 5457 5458 5458 5462 5468 5474 5475 5480 5480 5488 5491
                                                     5492
    PAGE 592  ROUTINE AMMO.RPT                       7125 7131 7132
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                7335 7337
    PAGE 597                                         7401 7403 7404 7406 7407
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7948 7948 7950 7951
    PAGE 618  ROUTINE SNAP.R                         8214
    PAGE 619                                         8305 8307 8308
    PAGE 629  FUNCTION BTRY.AVAILABLE                8645 8657 8658 8674 8675
    PAGE 634  FUNCTION FEBA.BAND                     8790 8809
    PAGE 636  FUNCTION HE.WLA                        8859 8871 8872 8874
    PAGE 638  FUNCTION ICM.WLA                       8957 8968 8970 8971
    PAGE 665  ''PROGRAM  OLDER.VERSION               9762
    PAGE 676                                          370
    PAGE 677                                          427
    PAGE 683                                          797  798
    PAGE 685                                          926
    PAGE 711                                         2428
BTRY.AVAILABLE
    PAGE  50  ''SECTION FOR DEFINITIONS              2850
    PAGE 183  ROUTINE FA.BN.ASGN                     8402
    PAGE 184                                         8403
    PAGE 629  FUNCTION BTRY.AVAILABLE                8645 8675
    PAGE 709  ''PROGRAM  OLDER.VERSION               2269
BTRY.FM.DEQ
    PAGE 163  ROUTINE BTRY.FM.DEQ                    7513 7526
```

```
    PAGE 486  PROCESS FIRE.MISSION                    2388
BTRY.FM.ENQ
    PAGE 164  ROUTINE BTRY.FM.ENQ                     7543 7552 7570
    PAGE 481  PROCESS FIRE.MISSION                    2118
BTRY.INPUT
    PAGE 520  ROUTINE MAIN2                           4230 4232
    PAGE 550  ROUTINE BTRY.INPUT                      5387
BTRY.INPUT.
    PAGE 551  ROUTINE BTRY.INPUT                      5451
BY.BN
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         321
    PAGE 252  ROUTINE MINE.EFFECTS                    1631
    PAGE 271  ROUTINE BTRY.EFFECTS                    2602
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                 2969
    PAGE 282  ROUTINE AC.DF.EFFECTS                   3149
    PAGE 355  EVENT ARTY.OCCUPATION                   6250
    PAGE 384  EVENT OFF.LINE.ATTRITION                7453
    PAGE 550  ROUTINE BTRY.INPUT                      5440
    PAGE 551                                          5458
    PAGE 665  ''PROGRAM  OLDER.VERSION                9763
BY.BN.RANK
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         328
    PAGE  44  ''SECTION FOR DEFINITIONS               2516
    PAGE  49                                          2797
    PAGE 665  ''PROGRAM  OLDER.VERSION                9770
    PAGE 703                                          1937
    PAGE 708                                          2218
BY.CUR.FM
    PAGE   3  PROGRAM REVISIONS                        131
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         325
    PAGE  89  ROUTINE FA.BN.MOVEMENT                  4429
    PAGE  90                                          4478 4479 4479 4496 4497 4497
    PAGE  91                                          4538 4539 4539
    PAGE 163  ROUTINE BTRY.FM.DEQ                     7528 7540 7541
    PAGE 164  ROUTINE BTRY.FM.ENQ                     7554 7563 7569
    PAGE 184  ROUTINE FA.BN.ASGN                      8410
    PAGE 209  ROUTINE PGM.MSN.ASGN                    9561 9566 9567 9582 9584 9584
    PAGE 270  ROUTINE BTRY.EFFECTS                    2589 2590
    PAGE 665  ''PROGRAM  OLDER.VERSION                9767
BY.FIRE.RATE
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         329
    PAGE  44  ''SECTION FOR DEFINITIONS               2517
    PAGE 666  ''PROGRAM  OLDER.VERSION                9771
    PAGE 703                                          1938
BY.FM.QUEUE
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES         335
    PAGE  35  ''SECTION FOR PROCESSES                 2014
    PAGE  49  ''SECTION FOR DEFINITIONS               2800
    PAGE  90  ROUTINE FA.BN.MOVEMENT                  4476 4494
    PAGE  91                                          4536
    PAGE 163  ROUTINE BTRY.FM.DEQ                     7529
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS          2740 2741 2742
    PAGE 463  PROCESS HOW.REPAIR                      1183
    PAGE 481  PROCESS FIRE.MISSION                    2117
    PAGE 486                                          2386
    PAGE 619  ROUTINE SNAP.R                          8308
```

```
    PAGE 666  ''PROGRAM  OLDER.VERSION                9777
    PAGE 694                                          1452
    PAGE 708                                          2221
BY.HOW.SET
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         333
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES        1491
    PAGE  49  ''SECTION FOR DEFINITIONS               2801
    PAGE 252  ROUTINE MINE.EFFECTS                    1613 1615 1620
    PAGE 271  ROUTINE BTRY.EFFECTS                    2607 2614
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                 2961
    PAGE 282  ROUTINE AC.DF.EFFECTS                   3140 3141
    PAGE 384  EVENT OFF.LINE.ATTRITION                7440 7445
    PAGE 463  PROCESS HOW.REPAIR                      1181 1212
    PAGE 485  PROCESS FIRE.MISSION                    2330
    PAGE 551  ROUTINE BTRY.INPUT                      5491
    PAGE 666  ''PROGRAM  OLDER.VERSION                9775
    PAGE 685                                           930
    PAGE 708                                          2222
BY.N.ROUNDS
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         326
    PAGE 665  ''PROGRAM  OLDER.VERSION                9768
BY.PGM.CAP
    PAGE   3  PROGRAM REVISIONS                        166
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         330
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9515
    PAGE 209                                          9588
    PAGE 258  ROUTINE FO.DETECTION                    1910 1911
    PAGE 363  EVENT CFR.OPERATOR                      6535
    PAGE 550  ROUTINE BTRY.INPUT                      5443
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                 7403 7406
    PAGE 666  ''PROGRAM  OLDER.VERSION                9772
BY.PGM.FM
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         324
    PAGE 665  ''PROGRAM  OLDER.VERSION                9766
BY.PGM.STATUS
    PAGE   3  PROGRAM REVISIONS                        165
BY.SCHD.LIST
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES         334
    PAGE  35  ''SECTION FOR PROCESSES                 2013
    PAGE  49  ''SECTION FOR DEFINITIONS               2802
    PAGE 185  ROUTINE FA.BN.ASGN                      8496
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS          2745 2746 2747
    PAGE 481  PROCESS FIRE.MISSION                    2110
    PAGE 629  FUNCTION BTRY.AVAILABLE                 8658
    PAGE 666  ''PROGRAM  OLDER.VERSION                9776
    PAGE 694                                          1451
    PAGE 708                                          2223
BY.STATUS
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES         322
    PAGE  89  ROUTINE FA.BN.MOVEMENT                  4433
    PAGE  90                                          4470 4475 4489 4493 4502
    PAGE  91                                          4548 4552 4562 4566
    PAGE  92                                          4579 4583
    PAGE 182  ROUTINE FA.BN.ASGN                      8327 8329
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9516
    PAGE 209                                          9588
```

```
    PAGE 355  EVENT ARTY.OCCUPATION                  6239 6243
    PAGE 394  EVENT START.ARTY.MOVEMENT              7724 7728
    PAGE 407  EVENT STOP.ARTY.MOVEMENT               8389 8393
    PAGE 665  ''PROGRAM  OLDER.VERSION               9764
BY.STOP.FASCAM.SUPP
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES        331
    PAGE  44  ''SECTION FOR DEFINITIONS              2489
    PAGE 182  ROUTINE FA.BN.ASGN                     8343
    PAGE 183                                         8346
    PAGE 208  ROUTINE PGM.MSN.ASGN                   9520
    PAGE 273  ROUTINE BTRY.EFFECTS                   2722 2724
    PAGE 666  ''PROGRAM  OLDER.VERSION               9773
    PAGE 702                                         1910
BY.TYPE
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES        323
    PAGE  89  ROUTINE FA.BN.MOVEMENT                 4429
    PAGE  90                                         4462 4463 4464 4464 4466 4482 4483 4484 4484 4486
    PAGE  91                                         4541 4542 4543 4543 4545 4555 4556 4557 4557 4559
    PAGE  92                                         4569 4570 4576
    PAGE 176  ROUTINE EST.COVERAGE                   8027
    PAGE 177                                         8083
    PAGE 178                                         8153 8154 8157 8158 8173 8177
    PAGE 181  ROUTINE FASCAM.COMPUTATION             8253
    PAGE 182  ROUTINE FA.BN.ASGN                     8334 8336 8342
    PAGE 183                                         8346 8395
    PAGE 184                                         8425 8429
    PAGE 188  ROUTINE FINAL.COVERAGE                 8570 8574 8576
    PAGE 189                                         8635
    PAGE 190                                         8703 8704
    PAGE 192                                         8809 8819
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9019 9027
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9144
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9323
    PAGE 208  ROUTINE PGM.MSN.ASGN                   9517 9519 9528 9530
    PAGE 233  ROUTINE SMOKE.COMPUTATION               698
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS               1415
    PAGE 252  ROUTINE MINE.EFFECTS                   1610 1626
    PAGE 260  ROUTINE BTRY.EFFECTS                   2001 2008 2010
    PAGE 262                                         2110 2112
    PAGE 263                                         2143
    PAGE 270                                         2562 2588
    PAGE 271                                         2604 2619
    PAGE 272                                         2710 2717
    PAGE 273                                         2718
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                2956
    PAGE 279                                         2964
    PAGE 282  ROUTINE AC.DF.EFFECTS                  3138 3143
    PAGE 355  EVENT ARTY.OCCUPATION                  6249
    PAGE 358  EVENT CFR.ACTIVATION                   6350
    PAGE 362  EVENT CFR.OPERATOR                     6525
    PAGE 384  EVENT OFF.LINE.ATTRITION               7438
    PAGE 386  EVENT PDB.ACTIVATION                   7477
    PAGE 387  EVENT PDB.OPERATOR                     7542
    PAGE 407  EVENT STOP.ARTY.MOVEMENT               8400
    PAGE 435  PROCESS ARTY.ASSESS                    9697
    PAGE 463  PROCESS HOW.REPAIR                     1180
```

```
    PAGE 476  PROCESS TARGET.REPORT                    1881
    PAGE 480  PROCESS FIRE.MISSION                     2080 2082
    PAGE 481                                           2086 2088 2095 2096 2098 2129 2135
    PAGE 482                                           2149 2150
    PAGE 483                                           2217 2229 2244 2249
    PAGE 484                                           2290
    PAGE 485                                           2332 2335
    PAGE 486                                           2398
    PAGE 550  ROUTINE BTRY.INPUT                       5441
    PAGE 551                                           5457
    PAGE 592  ROUTINE AMMO.RPT                         7132
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                  7337
    PAGE 597                                           7404 7407
    PAGE 611  ROUTINE OUTPUT.ATTRITION                 7950 7951 7969 7983
    PAGE 629  FUNCTION BTRY.AVAILABLE                  8657
    PAGE 636  FUNCTION HE.WLA                          8874
    PAGE 638  FUNCTION ICM.WLA                         8968
    PAGE 665  ''PROGRAM  OLDER.VERSION                 9765
BY.UNIT
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES          327
    PAGE  89  ROUTINE FA.BN.MOVEMENT                   4410 4424 4436
    PAGE  90                                           4455 4505
    PAGE  91                                           4516 4535
    PAGE 104  ROUTINE MINE.DELAY                       5031
    PAGE 165  ROUTINE CFR.DEGRADE                      7596
    PAGE 166  ROUTINE CFR.DETECTION                    7636 7637
    PAGE 167                                           7692
    PAGE 182  ROUTINE FA.BN.ASGN                       8341
    PAGE 183                                           8345 8365 8366 8370 8375 8379
    PAGE 184                                           8424
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ             9325
    PAGE 205  ROUTINE NOISE.DEGRADE                    9411
    PAGE 206  ROUTINE PDB.DETECTION                    9458
    PAGE 208  ROUTINE PGM.MSN.ASGN                     9518 9529 9533 9534 9538 9543
    PAGE 209                                           9547 9558 9604
    PAGE 246  ROUTINE VOLLEY                           1319 1322 1342
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS                 1423
    PAGE 260  ROUTINE BTRY.EFFECTS                     2010
    PAGE 262                                           2099
    PAGE 271                                           2646
    PAGE 272                                           2671 2674
    PAGE 355  EVENT ARTY.OCCUPATION                    6244
    PAGE 358  EVENT CFR.ACTIVATION                     6356 6368
    PAGE 362  EVENT CFR.OPERATOR                       6505 6518 6520
    PAGE 386  EVENT PDB.ACTIVATION                     7483 7494
    PAGE 387  EVENT PDB.OPERATOR                       7524 7536 7538
    PAGE 393  EVENT START.ARTY.MOVEMENT                7675 7678 7680 7685 7686 7688 7691 7704 7705 7705
    PAGE 394                                           7729 7730 7739
    PAGE 407  EVENT STOP.ARTY.MOVEMENT                 8394 8396
    PAGE 435  PROCESS ARTY.ASSESS                      9699
    PAGE 463  PROCESS HOW.REPAIR                       1196
    PAGE 480  PROCESS FIRE.MISSION                     2057 2058 2062 2068 2072
    PAGE 481                                           2127 2135
    PAGE 482                                           2144
    PAGE 551  ROUTINE BTRY.INPUT                       5451 5455 5456 5462 5468 5475 5475 5479
    PAGE 611  ROUTINE OUTPUT.ATTRITION                 7948
```

```
    PAGE 665  ''PROGRAM  OLDER.VERSION                9769
B.HQ.INC
    PAGE  69  ROUTINE GENERAL.BATTLE                  3588 3605
B.HQ.W
    PAGE  69  ROUTINE GENERAL.BATTLE                  3582 3586
B.MAN.INC
    PAGE  69  ROUTINE GENERAL.BATTLE                  3590 3612
B.MAN.W
    PAGE  69  ROUTINE GENERAL.BATTLE                  3583 3586
B.TF.LDR
    PAGE 112  ROUTINE PREPARE.LIST                    5375 5376
B.WIDTH.UNIT
    PAGE  42  ''SECTION FOR DEFINITIONS               2372
    PAGE  69  ROUTINE GENERAL.BATTLE                  3582 3583
    PAGE 569  ROUTINE TBF.INPUT                       6212 6217 6217
    PAGE 701  ''PROGRAM  OLDER.VERSION                1802
CALCULATIONS...
    PAGE 271  ROUTINE BTRY.EFFECTS                    2635
CALP.ON
    PAGE  43  ''SECTION FOR DEFINITIONS               2433
    PAGE 523  ROUTINE SYS.INPUT                       4326
    PAGE 530  ROUTINE EQ.TE.INPUT                     4606
CANDIDATE.REAR
    PAGE 119  ROUTINE RESET.FEBA.SECTOR               5663 5666 5670 5674
    PAGE 331  ROUTINE ENQ.FEBA.SET                    5328 5329 5330
CAS.MISSION
    PAGE  35  ''SECTION FOR PROCESSES                 1962
    PAGE 284  ROUTINE CAS.EVAL                        3259
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS           3398
    PAGE 288                                          3460 3467
    PAGE 309  ROUTINE AD.SHOOT                        4463 4464
    PAGE 349  EVENT AD.ENGAGEMENT                     5915
    PAGE 380  EVENT INIT.PREPLAN.CAS                  7238
    PAGE 421  PROCESS AC.ATK.TGT                      8975
    PAGE 426                                          9286
    PAGE 427                                          9323 9324
    PAGE 504  PROCESS CAS.MISSION                     3350 3368 3369
    PAGE 505                                          3434
    PAGE 506                                          3476 3477
    PAGE 508                                          3602 3603
    PAGE 509                                          3645 3646
    PAGE 599  ROUTINE BETWEEN.ROUTINE                 7468 7469
    PAGE 616  ROUTINE SNAP2                           8146
    PAGE 618  ROUTINE SNAP.R                          8251
    PAGE 694  ''PROGRAM  OLDER.VERSION                1400
CAS.MISSION.
    PAGE  58  ROUTINE CREATE.FORCE                    3116
CAS.MSN.RPT.FLAG
    PAGE  43  ''SECTION FOR DEFINITIONS               2430
    PAGE 294  ROUTINE END.CAS.MISSION                 3759
    PAGE 582  ROUTINE TACAIR.INPUT                    6750
    PAGE 702  ''PROGRAM  OLDER.VERSION                1860
CATEGORY
    PAGE   1  ROUTINE FOR CROSS_REFERENCING            44
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES         352  368  371  374
    PAGE   8                                           398
```

```
    PAGE  17                                          929
    PAGE 240  ROUTINE TARGET.ANALYSIS                1034
    PAGE 473  PROCESS TARGET.REPORT                  1673 1678
    PAGE 527  ROUTINE CAT.TU.INPUT                   4460 4461 4463 4464 4465 4466 4466 4467 4467
    PAGE 547  ROUTINE P.E.M.INPUT                    5282 5285 5287 5291 5293 5299 5300
    PAGE 560  ROUTINE  RUL.EN.INPUT                  5833 5840
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                7263
    PAGE 618  ROUTINE SNAP.R                         8215
    PAGE 666  ''PROGRAM  OLDER.VERSION               9794 9810 9813 9816
    PAGE 667                                         9840
    PAGE 676                                          371
    PAGE 711                                         2429
CAT.TU.INPUT
    PAGE 520  ROUTINE MAIN2                          4199 4201
    PAGE 527  ROUTINE CAT.TU.INPUT                   4450 4456
CDI.USAGE.INDICATOR
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        369
    PAGE  42  ''SECTION FOR DEFINITIONS              2380
    PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION          9058
    PAGE 560  ROUTINE  RUL.EN.INPUT                  5837
    PAGE 666  ''PROGRAM  OLDER.VERSION               9811
    PAGE 701                                         1810
CDT.MAX.VOLS
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        372
    PAGE  42  ''SECTION FOR DEFINITIONS              2379
    PAGE 181  ROUTINE FASCAM.COMPUTATION             8268
    PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION          9103
    PAGE 198                                         9111 9118
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9165
    PAGE 233  ROUTINE SMOKE.COMPUTATION               722
    PAGE 473  PROCESS TARGET.REPORT                  1678
    PAGE 560  ROUTINE  RUL.EN.INPUT                  5844
    PAGE 666  ''PROGRAM  OLDER.VERSION               9814
    PAGE 701                                         1809
CEM.WPN.NO
    PAGE 530  ROUTINE EQ.TE.INPUT                    4617 4618 4622 4628 4629
CENTERS.
    PAGE 448  PROCESS HC.ARRIVE.BATTLE                387
CENTER.
    PAGE 445  PROCESS HC.ARRIVE.BATTLE                216
CFPS.LIST
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES       1202
    PAGE  35  ''SECTION FOR PROCESSES                1983
    PAGE 293  ROUTINE END.CAS.MISSION                3689 3696
    PAGE 314  ROUTINE FLIGHT.PATH                    4678
    PAGE 315                                         4719 4745
    PAGE 316                                         4749 4759 4781
    PAGE 505  PROCESS CAS.MISSION                    3421 3423
    PAGE 508                                         3627
    PAGE 681  ''PROGRAM  OLDER.VERSION                641
    PAGE 694                                         1421
CFPS.TIME.LENGTH
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES       1198
    PAGE  47  ''SECTION FOR DEFINITIONS              2714
    PAGE 316  ROUTINE FLIGHT.PATH                    4768
    PAGE 505  PROCESS CAS.MISSION                    3425 3446
```

```
    PAGE 508                                    3638
    PAGE 680  ''PROGRAM  OLDER.VERSION           637
    PAGE 706                                    2136
CFPS.XEND
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES  1196
    PAGE  43  ''SECTION FOR DEFINITIONS         2471
    PAGE 316  ROUTINE FLIGHT.PATH               4762 4765 4769 4783 4784 4786
    PAGE 318                                    4867 4911 4915 4919
    PAGE 680  ''PROGRAM  OLDER.VERSION           635
    PAGE 702                                    1891
CFPS.XSTART
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES  1194
    PAGE  43  ''SECTION FOR DEFINITIONS         2470
    PAGE 314  ROUTINE FLIGHT.PATH               4679
    PAGE 315                                    4720
    PAGE 316                                    4746 4750 4765 4769 4783 4784 4786 4787
    PAGE 317                                    4857
    PAGE 318                                    4870 4874 4877 4895
    PAGE 319                                    4923
    PAGE 680  ''PROGRAM  OLDER.VERSION           633
    PAGE 702                                    1890
CFPS.YEND
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES  1197
    PAGE  43  ''SECTION FOR DEFINITIONS         2473
    PAGE 316  ROUTINE FLIGHT.PATH               4763 4766 4770 4785
    PAGE 318                                    4868 4916
    PAGE 319                                    4920
    PAGE 680  ''PROGRAM  OLDER.VERSION           636
    PAGE 702                                    1893
CFPS.YSTART
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES  1195
    PAGE  43  ''SECTION FOR DEFINITIONS         2472
    PAGE 314  ROUTINE FLIGHT.PATH               4680
    PAGE 315                                    4721 4730
    PAGE 316                                    4747 4751 4766 4770 4785 4787
    PAGE 317                                    4858
    PAGE 318                                    4871 4875 4896
    PAGE 319                                    4924
    PAGE 680  ''PROGRAM  OLDER.VERSION           634
    PAGE 702                                    1892
CFP.OBS.LINK
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES  1180
    PAGE 315  ROUTINE FLIGHT.PATH               4692 4713
    PAGE 316                                    4754
    PAGE 680  ''PROGRAM  OLDER.VERSION           619
CFP.SEGMENT
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES  1193
    PAGE 293  ROUTINE END.CAS.MISSION           3697
    PAGE 314  ROUTINE FLIGHT.PATH               4677
    PAGE 315                                    4718 4744
    PAGE 316                                    4748 4773
    PAGE 505  PROCESS CAS.MISSION               3421
    PAGE 680  ''PROGRAM  OLDER.VERSION           632
CFR.ACTIVATION
    PAGE  39  ''SECTION FOR EVENTS              2238
    PAGE 246  ROUTINE VOLLEY                    1359
```

```
    PAGE 358  EVENT CFR.ACTIVATION                    6341
    PAGE 601  ROUTINE BETWEEN.ROUTINE                 7564 7565
    PAGE 616  ROUTINE SNAP2                           8171
    PAGE 698  ''PROGRAM  OLDER.VERSION                1675
CFR.ACT.BTRY
    PAGE  39  ''SECTION FOR EVENTS                    2239
    PAGE 698  ''PROGRAM  OLDER.VERSION                1676
CFR.CIR.ERROR
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES         381
    PAGE  44  ''SECTION FOR DEFINITIONS               2518
    PAGE 167  ROUTINE CFR.DETECTION                   7679
    PAGE 562  ROUTINE MCFR.INPUT                      5907
    PAGE 666  ''PROGRAM  OLDER.VERSION                9823
    PAGE 703                                          1939
CFR.DET.PROB
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES         380
    PAGE  44  ''SECTION FOR DEFINITIONS               2519
    PAGE 167  ROUTINE CFR.DETECTION                   7678
    PAGE 562  ROUTINE MCFR.INPUT                      5906
    PAGE 666  ''PROGRAM  OLDER.VERSION                9822
    PAGE 703                                          1940
CFR.DET.UN
    PAGE  40  ''SECTION FOR EVENTS                    2251
    PAGE 699  ''PROGRAM  OLDER.VERSION                1688
CFR.LAST.ON.OR.OFF
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1224
    PAGE  48  ''SECTION FOR DEFINITIONS               2725
    PAGE 206  ROUTINE PDB.DETECTION                   9472
    PAGE 359  EVENT CFR.OFF                           6401
    PAGE 361  EVENT CFR.ON                            6473
    PAGE 681  ''PROGRAM  OLDER.VERSION                 663
    PAGE 706                                          2147
CFR.OFF
    PAGE  39  ''SECTION FOR EVENTS                    2241
    PAGE 161  ROUTINE ATTRIT.SENSOR                   7462 7463
    PAGE 359  EVENT CFR.OFF                           6380
    PAGE 360  EVENT CFR.ON                            6422
    PAGE 601  ROUTINE BETWEEN.ROUTINE                 7568 7569
    PAGE 616  ROUTINE SNAP2                           8172
    PAGE 698  ''PROGRAM  OLDER.VERSION                1678
CFR.OFF.ID
    PAGE  39  ''SECTION FOR EVENTS                    2242
    PAGE 161  ROUTINE ATTRIT.SENSOR                   7459
    PAGE 698  ''PROGRAM  OLDER.VERSION                1679
CFR.OFF.LINK
    PAGE  39  ''SECTION FOR EVENTS                    2243
    PAGE 698  ''PROGRAM  OLDER.VERSION                1680
CFR.ON
    PAGE   4  PROGRAM REVISIONS                        197
    PAGE  39  ''SECTION FOR EVENTS                    2245
    PAGE 161  ROUTINE ATTRIT.SENSOR                   7454 7455
    PAGE 206  ROUTINE PDB.DETECTION                   9478 9479 9482
    PAGE 359  EVENT CFR.OFF                           6395
    PAGE 360  EVENT CFR.ON                            6405
    PAGE 601  ROUTINE BETWEEN.ROUTINE                 7572 7573
    PAGE 616  ROUTINE SNAP2                           8173
```

```
    PAGE 698  ''PROGRAM  OLDER.VERSION                1682
CFR.ON.ID
    PAGE  39  ''SECTION FOR EVENTS                    2246
    PAGE 161  ROUTINE ATTRIT.SENSOR                   7453
    PAGE 206  ROUTINE PDB.DETECTION                   9477
    PAGE 698  ''PROGRAM  OLDER.VERSION                1683
CFR.ON.LINK
    PAGE  39  ''SECTION FOR EVENTS                    2247
    PAGE 698  ''PROGRAM  OLDER.VERSION                1684
CFR.OPERATOR
    PAGE  39  ''SECTION FOR EVENTS                    2249
    PAGE 161  ROUTINE ATTRIT.SENSOR                   7469 7470
    PAGE 167  ROUTINE CFR.DETECTION                   7695
    PAGE 362  EVENT CFR.OPERATOR                      6477 6494 6514 6529
    PAGE 363                                          6552 6586
    PAGE 601  ROUTINE BETWEEN.ROUTINE                 7576 7577
    PAGE 616  ROUTINE SNAP2                           8174
    PAGE 699  ''PROGRAM  OLDER.VERSION                1686
CFR.ORIENTATION
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1225
    PAGE  44  ''SECTION FOR DEFINITIONS               2520
    PAGE 166  ROUTINE CFR.DETECTION                   7660 7661
    PAGE 361  EVENT CFR.ON                            6468 6469 6470
    PAGE 681  ''PROGRAM  OLDER.VERSION                 664
    PAGE 703                                          1941
CFR.RH.RANGE
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES         379
    PAGE 167  ROUTINE CFR.DETECTION                   7675
    PAGE 358  EVENT CFR.ACTIVATION                    6372
    PAGE 562  ROUTINE MCFR.INPUT                      5908
    PAGE 666  ''PROGRAM  OLDER.VERSION                9821
CFR.RNG.HACK
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES         378
    PAGE 562  ROUTINE MCFR.INPUT                      5884
    PAGE 618  ROUTINE SNAP.R                          8216
    PAGE 666  ''PROGRAM  OLDER.VERSION                9820
    PAGE 711                                          2430
CFR.SENS.ID
    PAGE  39  ''SECTION FOR EVENTS                    2250
    PAGE 161  ROUTINE ATTRIT.SENSOR                   7466
    PAGE 699  ''PROGRAM  OLDER.VERSION                1687
CFR.US.LINK
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1226
    PAGE 166  ROUTINE CFR.DETECTION                   7627
    PAGE 206  ROUTINE PDB.DETECTION                   9468
    PAGE 358  EVENT CFR.ACTIVATION                    6364
    PAGE 362  EVENT CFR.OPERATOR                      6488
    PAGE 566  ROUTINE SENSOR.INPUT                    6121
    PAGE 681  ''PROGRAM  OLDER.VERSION                 665
CF.DET.UNIT
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1211
    PAGE 161  ROUTINE ATTRIT.SENSOR                   7475
    PAGE 167  ROUTINE CFR.DETECTION                   7688
    PAGE 363  EVENT CFR.OPERATOR                      6560
    PAGE 681  ''PROGRAM  OLDER.VERSION                 650
```

```
CF.D.BTRY
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1212
    PAGE  44  ''SECTION FOR DEFINITIONS               2521
    PAGE 166  ROUTINE CFR.DETECTION                   7631
    PAGE 167                                          7689
    PAGE 362  EVENT CFR.OPERATOR                      6504
    PAGE 681  ''PROGRAM  OLDER.VERSION                 651
    PAGE 703                                          1942
CF.D.CPE
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1213
    PAGE  44  ''SECTION FOR DEFINITIONS               2522
    PAGE 167  ROUTINE CFR.DETECTION                   7690
    PAGE 362  EVENT CFR.OPERATOR                      6510
    PAGE 681  ''PROGRAM  OLDER.VERSION                 652
    PAGE 703                                          1943
CF.D.PD
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1214
    PAGE  44  ''SECTION FOR DEFINITIONS               2523
    PAGE 167  ROUTINE CFR.DETECTION                   7691
    PAGE 363  EVENT CFR.OPERATOR                      6556
    PAGE 681  ''PROGRAM  OLDER.VERSION                 653
    PAGE 703                                          1944
CF.D.PRIORITY
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1215
    PAGE  44  ''SECTION FOR DEFINITIONS               2524
    PAGE  49                                          2803
    PAGE 167  ROUTINE CFR.DETECTION                   7692
    PAGE 681  ''PROGRAM  OLDER.VERSION                 654
    PAGE 703                                          1945
    PAGE 708                                          2224
CF.OPERATOR
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1227
    PAGE 167  ROUTINE CFR.DETECTION                   7694
    PAGE 362  EVENT CFR.OPERATOR                      6490 6491 6500
    PAGE 681  ''PROGRAM  OLDER.VERSION                 666
CF.OP.Q
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1217 1231
    PAGE 161  ROUTINE ATTRIT.SENSOR                   7472 7474
    PAGE 166  ROUTINE CFR.DETECTION                   7630
    PAGE 167                                          7693
    PAGE 362  EVENT CFR.OPERATOR                      6501
    PAGE 363                                          6584
    PAGE 681  ''PROGRAM  OLDER.VERSION                 656  670
CF.OP.Q.
    PAGE  49  ''SECTION FOR DEFINITIONS               2803
    PAGE 708  ''PROGRAM  OLDER.VERSION                2224
CF.RADAR
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1223
    PAGE 161  ROUTINE ATTRIT.SENSOR                   7484
    PAGE 566  ROUTINE SENSOR.INPUT                    6120
    PAGE 619  ROUTINE SNAP.R                          8293
    PAGE 681  ''PROGRAM  OLDER.VERSION                 662
CHANGE.
    PAGE 408  EVENT UPDATE.LOC                        8445
    PAGE 411                                          8579
```

```
CHANGE.LITE
    PAGE  39  ''SECTION FOR EVENTS                  2202
    PAGE  57  ROUTINE MAIN3                         3079
    PAGE 365  EVENT CHANGE.LITE                     6595 6603 6609
    PAGE 600  ROUTINE BETWEEN.ROUTINE               7524 7525
    PAGE 616  ROUTINE SNAP2                         8161
    PAGE 698  ''PROGRAM  OLDER.VERSION              1639
CHANGE.LOC
    PAGE  83  ROUTINE CHANGE.LOC                    4124
    PAGE 106  ROUTINE MIN.MOVE                      5065
    PAGE 173  ROUTINE DUST.EFFECTS                  7917
    PAGE 235  ROUTINE SMOKE.EFFECTS                  805
    PAGE 261  ROUTINE BTRY.EFFECTS                  2075
    PAGE 381  EVENT MOVE                            7274
CHANGE.WEATHE
    PAGE 366  EVENT CHANGE.WEATHER                  6629
CHANGE.WEATHER
    PAGE  40  ''SECTION FOR EVENTS                  2253
    PAGE 366  EVENT CHANGE.WEATHER                  6616
    PAGE 581  ROUTINE VIS.INPUT                     6726
    PAGE 601  ROUTINE BETWEEN.ROUTINE               7580 7581
    PAGE 699  ''PROGRAM  OLDER.VERSION              1690
CHECKED.
    PAGE 408  EVENT UPDATE.LOC                      8419
CHECK.CAS.CONSTRAINTS
    PAGE 284  ROUTINE CAS.EVAL                      3272
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS         3280 3290
    PAGE 327  ROUTINE DQ.CMSN.QUEUE                 5159
    PAGE 380  EVENT INIT.PREPLAN.CAS                7261
CHECK.DEAD
    PAGE 130  ROUTINE CHECK.DEAD                    6067 6107
    PAGE 147  ROUTINE INTER.BATTLE                  6841 6846
    PAGE 148                                        6878 6883
    PAGE 356  EVENT BTL.ENDED                       6257
    PAGE 357                                        6326 6328
CHECK.LIST
    PAGE 112  ROUTINE PREPARE.LIST                  5378
    PAGE 137  ROUTINE CHECK.LIST                    6332 6361
CHECK.UNIT
    PAGE 115  ROUTINE PROX.CHECK                    5498 5517 5518 5519 5523 5526 5529 5534 5541 5541 5544 5545 5552
    PAGE 116                                        5555 5558
    PAGE 133  ROUTINE CHECK.FORCE                   6155 6159 6160 6181 6200 6201
    PAGE 345  ROUTINE TERM.CHECK                    5767 5772 5773 5792
    PAGE 478  PROCESS WITH.DRAW                     1916 1930 1931 1934 1936 1943 1946 1947 1948 1951 1953 1954 1965 1968
    PAGE 479                                        1986 1988 1989 1996 2011 2020 2023
CHK.COMP.TR
    PAGE 168  ROUTINE CHK.COMP.TR                   7699 7706
    PAGE 363  EVENT CFR.OPERATOR                    6561
    PAGE 387  EVENT PDB.OPERATOR                    7549
    PAGE 471  PROCESS TARGET.REPORT                 1587
CHK.RANGE
    PAGE 138  ROUTINE CHECK.PROX                    6381
CHK.RANGE.
    PAGE 138  ROUTINE CHECK.PROX                    6373 6427
    PAGE 139                                        6444
```

```
CLOSEST.FARRP
    PAGE 297  ROUTINE FARRP.CHECK                    3861 3900 3904 3906
CMSN.AC.TYPE
    PAGE  35  ''SECTION FOR PROCESSES                1967
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2811 2837 2840
    PAGE 280  ROUTINE AC.DF.EFFECTS                  3047
    PAGE 281                                         3105
    PAGE 284  ROUTINE CAS.EVAL                       3264
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS          3298 3310 3315 3320
    PAGE 286                                         3377 3385
    PAGE 287                                         3404 3421
    PAGE 295  ROUTINE END.CAS.MISSION                3797 3828
    PAGE 307  ROUTINE AD.SHOOT                       4344 4347
    PAGE 314  ROUTINE FLIGHT.PATH                    4637 4662 4664
    PAGE 316                                         4771
    PAGE 351  EVENT AD.ENGAGEMENT                    6014
    PAGE 380  EVENT INIT.PREPLAN.CAS                 7248
    PAGE 419  PROCESS AC.ATK.TGT                     8857
    PAGE 504  PROCESS CAS.MISSION                    3355
    PAGE 694  ''PROGRAM  OLDER.VERSION               1405
CMSN.ANGLE
    PAGE  35  ''SECTION FOR PROCESSES                1981
    PAGE  47  ''SECTION FOR DEFINITIONS              2715
    PAGE 422  PROCESS AC.ATK.TGT                     9041
    PAGE 507  PROCESS CAS.MISSION                    3548
    PAGE 694  ''PROGRAM  OLDER.VERSION               1419
    PAGE 706                                         2137
CMSN.ASP.STATUS
    PAGE  35  ''SECTION FOR PROCESSES                1978
    PAGE  58  ROUTINE CREATE.FORCE                   3127
    PAGE 284  ROUTINE CAS.EVAL                       3270
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS          3439
    PAGE 293  ROUTINE END.CAS.MISSION                3707
    PAGE 380  EVENT INIT.PREPLAN.CAS                 7252
    PAGE 504  PROCESS CAS.MISSION                    3359
    PAGE 694  ''PROGRAM  OLDER.VERSION               1416
CMSN.FIRST.PASS.TIME
    PAGE  35  ''SECTION FOR PROCESSES                1977
    PAGE  48  ''SECTION FOR DEFINITIONS              2716
    PAGE 293  ROUTINE END.CAS.MISSION                3708
    PAGE 295                                         3799
    PAGE 508  PROCESS CAS.MISSION                    3595
    PAGE 694  ''PROGRAM  OLDER.VERSION               1415
    PAGE 706                                         2138
CMSN.FLIGHT.TIME
    PAGE  35  ''SECTION FOR PROCESSES                1976
    PAGE  48  ''SECTION FOR DEFINITIONS              2717
    PAGE 295  ROUTINE END.CAS.MISSION                3801
    PAGE 421  PROCESS AC.ATK.TGT                     8977
    PAGE 504  PROCESS CAS.MISSION                    3361
    PAGE 694  ''PROGRAM  OLDER.VERSION               1414
    PAGE 706                                         2139
CMSN.NR.ABORTED
    PAGE  35  ''SECTION FOR PROCESSES                1969
    PAGE 295  ROUTINE END.CAS.MISSION                3800 3815
    PAGE 504  PROCESS CAS.MISSION                    3358
```

```
    PAGE 694  ''PROGRAM  OLDER.VERSION                1407
CMSN.NR.AC
    PAGE  35  ''SECTION FOR PROCESSES                 1968
    PAGE 284  ROUTINE CAS.EVAL                        3266
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS           3305 3307 3318
    PAGE 286                                          3343 3374 3378 3378 3380 3391
    PAGE 287                                          3429 3450
    PAGE 295  ROUTINE END.CAS.MISSION                 3799 3815
    PAGE 380  EVENT INIT.PREPLAN.CAS                  7249
    PAGE 504  PROCESS CAS.MISSION                     3356
    PAGE 694  ''PROGRAM  OLDER.VERSION                1406
CMSN.NR.SURV.AC
    PAGE  35  ''SECTION FOR PROCESSES                 1970
    PAGE 295  ROUTINE END.CAS.MISSION                 3801
    PAGE 309  ROUTINE AD.SHOOT                        4461 4462
    PAGE 351  EVENT AD.ENGAGEMENT                     6013
    PAGE 426  PROCESS AC.ATK.TGT                      9307
    PAGE 504  PROCESS CAS.MISSION                     3357
    PAGE 694  ''PROGRAM  OLDER.VERSION                1408
CMSN.NR.SURV.AC.
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS           3447
CMSN.P1.ADU.RANGE
    PAGE  35  ''SECTION FOR PROCESSES                 1979
    PAGE  43  ''SECTION FOR DEFINITIONS               2480
    PAGE 422  PROCESS AC.ATK.TGT                      9038 9040 9042
    PAGE 506  PROCESS CAS.MISSION                     3514
    PAGE 694  ''PROGRAM  OLDER.VERSION                1417
    PAGE 702                                          1900
CMSN.P2.ADU.RANGE
    PAGE  43  ''SECTION FOR DEFINITIONS               2481
    PAGE 702  ''PROGRAM  OLDER.VERSION                1901
CMSN.P3.ADU.RANGE
    PAGE  35  ''SECTION FOR PROCESSES                 1980
    PAGE 424  PROCESS AC.ATK.TGT                      9161
    PAGE 506  PROCESS CAS.MISSION                     3520
    PAGE 694  ''PROGRAM  OLDER.VERSION                1418
CMSN.Q.FLAG
    PAGE  35  ''SECTION FOR PROCESSES                 1965
    PAGE  52  ''SECTION FOR SUBSTITUTIONS             2989
    PAGE 288  ROUTINE CHECK.CAS.CONSTRAINTS           3455
    PAGE 694  ''PROGRAM  OLDER.VERSION                1403
    PAGE 711                                          2396
CMSN.REQUEST.TIME
    PAGE  35  ''SECTION FOR PROCESSES                 1973
    PAGE  48  ''SECTION FOR DEFINITIONS               2718
    PAGE 284  ROUTINE CAS.EVAL                        3269
    PAGE 295  ROUTINE END.CAS.MISSION                 3795
    PAGE 380  EVENT INIT.PREPLAN.CAS                  7251
    PAGE 694  ''PROGRAM  OLDER.VERSION                1411
    PAGE 706                                          2140
CMSN.SCOREBOARD
    PAGE  35  ''SECTION FOR PROCESSES                 1972
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                 2813
    PAGE 282  ROUTINE AC.DF.EFFECTS                   3155
    PAGE 294  ROUTINE END.CAS.MISSION                 3762
    PAGE 504  PROCESS CAS.MISSION                     3384
```

```
    PAGE 694  ''PROGRAM  OLDER.VERSION                1410
CMSN.SEQ.NR
    PAGE  35  ''SECTION FOR PROCESSES                 1963
    PAGE  43  ''SECTION FOR DEFINITIONS               2478
    PAGE 284  ROUTINE CAS.EVAL                        3261
    PAGE 286  ROUTINE CHECK.CAS.CONSTRAINTS           3384 3392
    PAGE 287                                          3403
    PAGE 288                                          3465
    PAGE 294  ROUTINE END.CAS.MISSION                 3769
    PAGE 295                                          3794 3816
    PAGE 307  ROUTINE AD.SHOOT                        4322 4326
    PAGE 380  EVENT INIT.PREPLAN.CAS                  7245
    PAGE 419  PROCESS AC.ATK.TGT                      8885
    PAGE 423                                          9085 9117
    PAGE 424                                          9193
    PAGE 425                                          9224
    PAGE 426                                          9268
    PAGE 694  ''PROGRAM  OLDER.VERSION                1401
    PAGE 702                                          1898
CMSN.SIDE
    PAGE  35  ''SECTION FOR PROCESSES                 1966
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                 2812
    PAGE 280  ROUTINE AC.DF.EFFECTS                   3048
    PAGE 284  ROUTINE CAS.EVAL                        3263
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS           3287
    PAGE 287                                          3438
    PAGE 293  ROUTINE END.CAS.MISSION                 3683
    PAGE 294                                          3769
    PAGE 295                                          3794
    PAGE 307  ROUTINE AD.SHOOT                        4323 4327
    PAGE 314  ROUTINE FLIGHT.PATH                     4636
    PAGE 380  EVENT INIT.PREPLAN.CAS                  7247
    PAGE 419  PROCESS AC.ATK.TGT                      8885
    PAGE 420                                          8949
    PAGE 421                                          8983
    PAGE 423                                          9085 9117
    PAGE 424                                          9193
    PAGE 425                                          9224
    PAGE 426                                          9256 9268 9289
    PAGE 504  PROCESS CAS.MISSION                     3354
    PAGE 694  ''PROGRAM  OLDER.VERSION                1404
CMSN.START.TIME
    PAGE  35  ''SECTION FOR PROCESSES                 1974
    PAGE  48  ''SECTION FOR DEFINITIONS               2719
    PAGE 295  ROUTINE END.CAS.MISSION                 3796
    PAGE 504  PROCESS CAS.MISSION                     3372
    PAGE 694  ''PROGRAM  OLDER.VERSION                1412
    PAGE 706                                          2141
CMSN.TAKE.OFF.TIME
    PAGE  35  ''SECTION FOR PROCESSES                 1975
    PAGE  48  ''SECTION FOR DEFINITIONS               2720
    PAGE 295  ROUTINE END.CAS.MISSION                 3798
    PAGE 421  PROCESS AC.ATK.TGT                      8975
    PAGE 505  PROCESS CAS.MISSION                     3448
    PAGE 694  ''PROGRAM  OLDER.VERSION                1413
    PAGE 706                                          2142
```

```
CMSN.TGT.UNIT
    PAGE  35  ''SECTION FOR PROCESSES                1971
    PAGE  43  ''SECTION FOR DEFINITIONS              2479
    PAGE  58  ROUTINE CREATE.FORCE                   3118 3126
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2807
    PAGE 280  ROUTINE AC.DF.EFFECTS                  3044
    PAGE 284  ROUTINE CAS.EVAL                       3268
    PAGE 286  ROUTINE CHECK.CAS.CONSTRAINTS          3393
    PAGE 287                                         3408 3410 3411 3441 3442 3443 3444
    PAGE 288                                         3466
    PAGE 293  ROUTINE END.CAS.MISSION                3684 3706
    PAGE 314  ROUTINE FLIGHT.PATH                    4649 4661 4663 4666 4667
    PAGE 380  EVENT INIT.PREPLAN.CAS                 7250
    PAGE 419  PROCESS AC.ATK.TGT                     8858
    PAGE 504  PROCESS CAS.MISSION                    3360
    PAGE 694  ''PROGRAM  OLDER.VERSION               1409
    PAGE 702                                         1899
CMSN.TYPE
    PAGE  35  ''SECTION FOR PROCESSES                1964
    PAGE  52  ''SECTION FOR SUBSTITUTIONS            2995
    PAGE 284  ROUTINE CAS.EVAL                       3262
    PAGE 294  ROUTINE END.CAS.MISSION                3763
    PAGE 295                                         3816
    PAGE 380  EVENT INIT.PREPLAN.CAS                 7246
    PAGE 694  ''PROGRAM  OLDER.VERSION               1402
    PAGE 711                                         2402
COLOR
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           46
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES        584
    PAGE  58  ROUTINE CREATE.FORCE                   3092 3098 3106 3107
    PAGE  60  ROUTINE CREATE.TEAMS                   3159 3167
    PAGE 536  ROUTINE UNIT.INPUT                     4903
    PAGE 570  ROUTINE DECISION.INPUT                 6267 6270 6274 6275 6276
    PAGE 670  ''PROGRAM  OLDER.VERSION                25
COL.OBSTACLE.PTR
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES       1181
    PAGE 315  ROUTINE FLIGHT.PATH                    4693 4715
    PAGE 680  ''PROGRAM  OLDER.VERSION                620
COL.SET
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        827
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES       1187
    PAGE  49  ''SECTION FOR DEFINITIONS              2804
    PAGE 314  ROUTINE FLIGHT.PATH                    4643
    PAGE 315                                         4705 4710
    PAGE 316                                         4753
    PAGE 674  ''PROGRAM  OLDER.VERSION                269
    PAGE 680                                          626
    PAGE 708                                         2225
COL.XENTRY
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES       1182
    PAGE  43  ''SECTION FOR DEFINITIONS              2474
    PAGE  49                                         2804
    PAGE 315  ROUTINE FLIGHT.PATH                    4695 4700 4720
    PAGE 316                                         4746
    PAGE 680  ''PROGRAM  OLDER.VERSION                621
    PAGE 702                                         1894
```

```
    PAGE 708                                        2225
COL.XENTRY.
    PAGE 314  ROUTINE FLIGHT.PATH                   4644
COL.XEXIT
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1183
    PAGE  43  ''SECTION FOR DEFINITIONS             2475
    PAGE 315  ROUTINE FLIGHT.PATH                   4697 4702
    PAGE 316                                        4750
    PAGE 680  ''PROGRAM  OLDER.VERSION               622
    PAGE 702                                        1895
COL.YENTRY
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1184
    PAGE  43  ''SECTION FOR DEFINITIONS             2476
    PAGE 315  ROUTINE FLIGHT.PATH                   4696 4701 4727
    PAGE 680  ''PROGRAM  OLDER.VERSION               623
    PAGE 702                                        1896
COL.YEXIT
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1185
    PAGE  43  ''SECTION FOR DEFINITIONS             2477
    PAGE 315  ROUTINE FLIGHT.PATH                   4698 4703
    PAGE 680  ''PROGRAM  OLDER.VERSION               624
    PAGE 702                                        1897
COMBAT.UNITS
    PAGE 595  ROUTINE ANALYSIS.OUTPUT               7243 7246 7259 7261 7261
COMB.RESULT
    PAGE 231  ROUTINE SIZE.ESTIMATE                  632  666  672
COMMAND.
    PAGE 606  ROUTINE KV.PRINT                      7722
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES           8503
COMPLETED.
    PAGE 334  ROUTINE FINISH.COMPUTATION            5388
    PAGE 545  ROUTINE ORD.MOVDIS                    5191
COMPUTED.
    PAGE 182  ROUTINE FA.BN.ASGN                    8296
    PAGE 242  ROUTINE UNIT.ENVIR                    1126
CONFLICT.
    PAGE 539  ROUTINE READ.ORDERS                   4993
CONTINUE.
    PAGE 445  PROCESS HC.ARRIVE.BATTLE               217
CONTROLLER.
    PAGE 505  PROCESS CAS.MISSION                   3454
COORDINATES.
    PAGE 415  EVENT ACT.MOVCOR                      8743
    PAGE 546  ROUTINE ORD.MOVCOR                    5210
COST.CRITERIA
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES       728
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS              1426 1433
    PAGE 523  ROUTINE SYS.INPUT                     4343
    PAGE 672  ''PROGRAM  OLDER.VERSION               171
COS.F
    PAGE  73  ROUTINE ORIENTATION                   3794
    PAGE  74                                        3809 3819
    PAGE  75  ROUTINE UNIT.ASSIGNMENT               3865
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA            4798 4808 4811
    PAGE 111  ROUTINE PRED.POS                      5341
    PAGE 422  PROCESS AC.ATK.TGT                    9041
```

```
    PAGE 447  PROCESS HC.ARRIVE.BATTLE                354  372
    PAGE 448                                          392
COVERAGE.
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2829
CPM.UNWARNED.FRACT
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        376
    PAGE 266  ROUTINE BTRY.EFFECTS                   2315 2324 2337 2338 2339
    PAGE 547  ROUTINE P.E.M.INPUT                    5300
    PAGE 637  FUNCTION HE.WLA                        8944
    PAGE 666  ''PROGRAM  OLDER.VERSION               9818
CPM.WARNED.FRACT
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        375
    PAGE 266  ROUTINE BTRY.EFFECTS                   2316 2325 2333 2334 2335
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                2904
    PAGE 547  ROUTINE P.E.M.INPUT                    5299
    PAGE 637  FUNCTION HE.WLA                        8947
    PAGE 639  FUNCTION ICM.WLA                       9019
    PAGE 666  ''PROGRAM  OLDER.VERSION               9817
CPU.TIME
    PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT          8819
    PAGE 650  ''PROGRAM LIB$STAT_TIMER               9221
CRIT.NO
    PAGE  58  ROUTINE CREATE.FORCE                   3111 3134
    PAGE  59                                         3154
    PAGE 133  ROUTINE CHECK.FORCE                    6174 6177
CROSSED.
    PAGE  87  ROUTINE END.MOVE                       4356
CT.GROUP
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        354
    PAGE  51  ''SECTION FOR SUBSTITUTIONS            2904
    PAGE  66  ROUTINE FORM.TF.LIST                   3425
    PAGE  89  ROUTINE FA.BN.MOVEMENT                 4425
    PAGE 104  ROUTINE MINE.DELAY                     5029
    PAGE 243  ROUTINE UNIT.ENVIR                     1206
    PAGE 252  ROUTINE MINE.EFFECTS                   1604
    PAGE 270  ROUTINE BTRY.EFFECTS                   2583
    PAGE 272                                         2709
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                2953
    PAGE 282  ROUTINE AC.DF.EFFECTS                  3135
    PAGE 384  EVENT OFF.LINE.ATTRITION               7435
    PAGE 483  PROCESS FIRE.MISSION                   2227
    PAGE 484                                         2281
    PAGE 527  ROUTINE CAT.TU.INPUT                   4465 4467
    PAGE 533  ROUTINE UNIT.INPUT                     4735
    PAGE 539  ROUTINE READ.ORDERS                    5013
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                7257
    PAGE 666  ''PROGRAM  OLDER.VERSION               9796
    PAGE 709                                         2312
CT.MIN.FEBA
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        355
    PAGE 240  ROUTINE TARGET.ANALYSIS                1036 1037
    PAGE 527  ROUTINE CAT.TU.INPUT                   4464 4466 4466
    PAGE 666  ''PROGRAM  OLDER.VERSION               9797
CT.NAME
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        353
    PAGE  47  ''SECTION FOR DEFINITIONS              2661
```

```
    PAGE 476  PROCESS TARGET.REPORT                    1876
    PAGE 494  PROCESS SHOOT.OUT                        2814
    PAGE 506  PROCESS CAS.MISSION                      3495
    PAGE 527  ROUTINE CAT.TU.INPUT                     4463
    PAGE 533  ROUTINE UNIT.INPUT                       4736
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                  7268
    PAGE 666  ''PROGRAM  OLDER.VERSION                 9795
    PAGE 705                                           2085
CT.TU.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES          357
    PAGE  17                                            944
    PAGE 240  ROUTINE TARGET.ANALYSIS                  1041
    PAGE 527  ROUTINE CAT.TU.INPUT                     4487
    PAGE 666  ''PROGRAM  OLDER.VERSION                 9799
    PAGE 676                                            386
CUM.NO.BATTLES
    PAGE  41  ''SECTION FOR DEFINITIONS                2359
    PAGE 395  EVENT START.BATTLE                       7797 7798
    PAGE 700  ''PROGRAM  OLDER.VERSION                 1789
CUR.MISSION
    PAGE 164  ROUTINE BTRY.FM.ENQ                      7570
CUR.STRENGTH
    PAGE 130  ROUTINE CHECK.DEAD                       6085 6086
CW.VISIBILITY
    PAGE  40  ''SECTION FOR EVENTS                     2254
    PAGE 699  ''PROGRAM  OLDER.VERSION                 1691
DATA.
    PAGE 622  ROUTINE TACAIR.DATA.REPORT               8394
    PAGE 623                                           8457
DATA.ERROR
    PAGE 533  ROUTINE UNIT.INPUT                       4709 4715 4724
    PAGE 534                                           4745 4752 4794
    PAGE 535                                           4831
    PAGE 536                                           4866 4874
DB.ENEMY.UNITS
    PAGE  40  ''SECTION FOR EVENTS                     2277
    PAGE 339  ROUTINE HC.EMPTY                         5575 5582
    PAGE 699  ''PROGRAM  OLDER.VERSION                 1714
DB.FARRP
    PAGE  40  ''SECTION FOR EVENTS                     2275
    PAGE 338  ROUTINE HC.EMPTY                         5569
    PAGE 699  ''PROGRAM  OLDER.VERSION                 1712
DB.TEAM
    PAGE  40  ''SECTION FOR EVENTS                     2276
    PAGE 338  ROUTINE HC.EMPTY                         5571 5572
    PAGE 339                                           5574 5576 5581 5584 5588 5590 5591
    PAGE 699  ''PROGRAM  OLDER.VERSION                 1713
DEAD.CHOPPER
    PAGE 488  PROCESS ASSESSMENT                       2498 2503
    PAGE 489                                           2527 2530 2543 2551 2556 2557 2559
    PAGE 490                                           2591
DEAD.EQ
    PAGE 383  EVENT OFF.LINE.ATTRITION                 7373 7376 7387 7394 7395
DEAD.HOW
    PAGE 384  EVENT OFF.LINE.ATTRITION                 7439 7442 7447
```

```
DEAD.SENS
    PAGE 384  EVENT OFF.LINE.ATTRITION                7423 7427 7430
DEAD.SO
    PAGE 383  EVENT OFF.LINE.ATTRITION                7385 7389 7390 7392
DEAD.UNIT
    PAGE 130  ROUTINE CHECK.DEAD                      6073 6091 6104 6107
    PAGE 142  ROUTINE DEAD.UNIT                       6521
DEBUG
    PAGE   1  ROUTINE FOR CROSS_REFERENCING             54
    PAGE  41  ''SECTION FOR DEFINITIONS               2356
    PAGE  63  ROUTINE FILE.FD.SCHD                    3264
    PAGE 100  ROUTINE LOCATE.SECTOR                   4865
    PAGE 107  ROUTINE NEW.SEGMENT                     5143
    PAGE 108                                          5178
    PAGE 119  ROUTINE RESET.FEBA.SECTOR               5656
    PAGE 123  ROUTINE TIME.TO.DETECT                  5846
    PAGE 137  ROUTINE CHECK.LIST                      6359 6367
    PAGE 154  ROUTINE AO.DETECTION                    7098 7111
    PAGE 155                                          7142
    PAGE 156                                          7242
    PAGE 157                                          7252 7258 7291
    PAGE 159  ROUTINE ATTRIT.SENSOR                   7326 7333 7341 7360
    PAGE 160                                          7370
    PAGE 161                                          7479
    PAGE 162                                          7500
    PAGE 163  ROUTINE BTRY.FM.DEQ                     7524
    PAGE 164  ROUTINE BTRY.FM.ENQ                     7550 7568
    PAGE 168  ROUTINE CHK.COMP.TR                     7704
    PAGE 176  ROUTINE EST.COVERAGE                    8022 8032
    PAGE 177                                          8072 8088
    PAGE 178                                          8128
    PAGE 179                                          8213
    PAGE 180  ROUTINE EST.MIL.WORTH                   8237
    PAGE 182  ROUTINE FA.BN.ASGN                      8328 8335
    PAGE 183                                          8344 8353 8386
    PAGE 184                                          8415 8439
    PAGE 189  ROUTINE FINAL.COVERAGE                  8621
    PAGE 190                                          8678
    PAGE 192                                          8822
    PAGE 194  ROUTINE FIND.START.TIME                 8901
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION           9021
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ            9314
    PAGE 209  ROUTINE PGM.MSN.ASGN                    9574 9587 9595
    PAGE 210                                          9651
    PAGE 211                                          9685 9694
    PAGE 212  ROUTINE PIR.DETECTION                   9713
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION         9804 9820
    PAGE 229  ROUTINE RPV.DETECTION                    525
    PAGE 239  ROUTINE TARGET.ANALYSIS                  995
    PAGE 241                                          1089
    PAGE 242  ROUTINE UNIT.ENVIR                      1144
    PAGE 245                                          1295
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS                1393 1408
    PAGE 249                                          1448
    PAGE 254  ROUTINE FO.DETECTION                    1690 1711 1721
    PAGE 255                                          1733 1745 1757
```

```
    PAGE 256                                        1804 1828
    PAGE 257                                        1860
    PAGE 262  ROUTINE BTRY.EFFECTS                  2089
    PAGE 264                                        2200 2215
    PAGE 268                                        2436 2466
    PAGE 269                                        2526
    PAGE 270                                        2574 2591
    PAGE 325  ROUTINE DEQ.FEBA.SET                  5085
    PAGE 331  ROUTINE ENQ.FEBA.SET                  5307
    PAGE 334  ROUTINE FINISH.COMPUTATION            5394
    PAGE 355  EVENT ARTY.OCCUPATION                 6232
    PAGE 369  EVENT ENGAGEMENT                      6686
    PAGE 370                                        6741
    PAGE 376  EVENT HELO.ENGAGEMENT                 7050
    PAGE 392  EVENT SET.DEBUG                       7628 7633
    PAGE 398  EVENT START.BATTLE                    7963
    PAGE 408  EVENT UPDATE.LOC                      8451
    PAGE 432  PROCESS AIR.OBSERVER                  9589
    PAGE 438  PROCESS FORWARD.OBSERVER              9901
    PAGE 439                                        9949 9968
    PAGE 440                                        9986   10
    PAGE 441                                          29   50   75
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1371
    PAGE 470  PROCESS TARGET.REPORT                 1502 1524
    PAGE 471                                        1569 1595
    PAGE 472                                        1619 1645
    PAGE 473                                        1703
    PAGE 474                                        1716 1734
    PAGE 475                                        1772 1780 1827
    PAGE 480  PROCESS FIRE.MISSION                  2081
    PAGE 481                                        2087 2097 2103 2134
    PAGE 494  PROCESS SHOOT.OUT                     2812 2820
    PAGE 520  ROUTINE MAIN2                         4189
    PAGE 523  ROUTINE SYS.INPUT                     4323
    PAGE 527  ROUTINE CAT.TU.INPUT                  4455
    PAGE 599  ROUTINE BETWEEN.ROUTINE               7433
    PAGE 629  FUNCTION BTRY.AVAILABLE               8673
    PAGE 632  FUNCTION EST.RANGE                    8755
    PAGE 633  FUNCTION EST.TR.RANGE                 8768
    PAGE 635  FUNCTION FEBA.BAND                    8852
    PAGE 636  FUNCTION HE.WLA                       8870
    PAGE 638  FUNCTION ICM.WLA                      8969
    PAGE 656  ROUTINE AR.DETECTION                  9391
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9486
    PAGE 700  ''PROGRAM  OLDER.VERSION              1786
DECIDE.UNIT
    PAGE 324  ROUTINE DECIDE                        5036 5046 5047 5048 5049 5050 5054 5055 5056 5060 5063 5064 5065 5074
DECIMATED.
    PAGE 130  ROUTINE CHECK.DEAD                    6072
    PAGE 356  EVENT BTL.ENDED                       6259
DECISION
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           47
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES        585
    PAGE  58  ROUTINE CREATE.FORCE                   3107
    PAGE 145  ROUTINE FIN.BATTLE                     6723
    PAGE 570  ROUTINE DECISION.INPUT                 6274
```

```
    PAGE 670  ''PROGRAM  OLDER.VERSION                  26
DECISION INPUT
    PAGE 521  ROUTINE MAIN2                          4269 4271
    PAGE 570  ROUTINE DECISION.INPUT                 6260
DECISION.POINT
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES       1406
    PAGE  46  ''SECTION FOR DEFINITIONS              2619
    PAGE  58  ROUTINE CREATE.FORCE                   3107
    PAGE 133  ROUTINE CHECK.FORCE                    6165 6178 6206 6207
    PAGE 145  ROUTINE FIN.BATTLE                     6724 6726 6733
    PAGE 345  ROUTINE TERM.CHECK                     5778
    PAGE 684  ''PROGRAM  OLDER.VERSION                845
    PAGE 705                                         2042
DEFENDER.
    PAGE 115  ROUTINE PROX.CHECK                     5508
DEFILADE.DIST
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        845
    PAGE  44  ''SECTION FOR DEFINITIONS              2490
    PAGE 149  ROUTINE PK.COMPUTE                     6932
    PAGE 267  ROUTINE BTRY.EFFECTS                   2416
    PAGE 571  ROUTINE TT.FACTORS.INPUT               6321
    PAGE 674  ''PROGRAM  OLDER.VERSION                287
    PAGE 702                                         1911
DEFORDER
    PAGE   1  ROUTINE FOR CROSS_REFERENCING            51
    PAGE  39  ''SECTION FOR EVENTS                   2211
    PAGE 698  ''PROGRAM  OLDER.VERSION               1648
DEFUNIT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING            51
    PAGE  39  ''SECTION FOR EVENTS                   2210
    PAGE 698  ''PROGRAM  OLDER.VERSION               1647
DEF.ORDER
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES       1241
    PAGE 144  ROUTINE DEAD.UNIT                      6664
    PAGE 326  ROUTINE DESTROY.ORD                    5126
    PAGE 542  ROUTINE ORD.DEF                        5131
    PAGE 681  ''PROGRAM  OLDER.VERSION                680
    PAGE 712                                         2484
DEF.UNITS
    PAGE 112  ROUTINE PREPARE.LIST                   5350
DEF.UNITS.
    PAGE 112  ROUTINE PREPARE.LIST                   5362
    PAGE 113                                         5451 5455
DELAY.
    PAGE  95  ROUTINE INIT.REINF                     4644
DELAY.TIME
    PAGE 111  ROUTINE PRED.POS                       5302 5309 5332
DELTA.INTERCEPT
    PAGE 343  ROUTINE SEARCH.COVERAGE                5686 5688 5690 5691 5692
DELTA.TIME
    PAGE  83  ROUTINE CHANGE.LOC                     4133 4138 4140 4144 4156
    PAGE 106  ROUTINE MIN.MOVE                       5068 5093 5094 5098
    PAGE 194  ROUTINE FIND.START.TIME                8944 8946
    PAGE 195                                         8950 8968 8973 8977
    PAGE 395  EVENT START.BATTLE                     7754
    PAGE 402                                         8157 8164
```

```
DELTA.TIME.
    PAGE  83  ROUTINE CHANGE.LOC                    4135
DELTA.X
    PAGE  78  ROUTINE ADJUST                        3951 3952 3960
    PAGE 110  ROUTINE POSITION                      5276 5279
    PAGE 111  ROUTINE PRED.POS                      5333 5337
    PAGE 166  ROUTINE CFR.DETECTION                 7625 7636 7638 7646 7652
    PAGE 257  ROUTINE FO.DETECTION                  1855 1877
    PAGE 335  ROUTINE FRAC.COMPUTE                  5420 5423 5427 5429
    PAGE 429  PROCESS AIR.OBSERVER                  9436 9437 9439
    PAGE 430                                        9464 9465 9467 9469
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1403 1404 1406
    PAGE 628  FUNCTION ACT.RANGE                    8636 8639 8641 8641
    PAGE 632  FUNCTION EST.RANGE                    8752 8754
    PAGE 633  FUNCTION EST.TR.RANGE                 8765 8767
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9510 9511 9513
DELTA.Y
    PAGE  78  ROUTINE ADJUST                        3950 3960
    PAGE 110  ROUTINE POSITION                      5277 5280
    PAGE 111  ROUTINE PRED.POS                      5334 5338
    PAGE 166  ROUTINE CFR.DETECTION                 7625 7637 7639 7645 7652
    PAGE 257  ROUTINE FO.DETECTION                  1856 1879
    PAGE 429  PROCESS AIR.OBSERVER                  9434 9439
    PAGE 430                                        9462 9467 9469
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1402 1404 1406
    PAGE 628  FUNCTION ACT.RANGE                    8636 8640 8641 8641
    PAGE 632  FUNCTION EST.RANGE                    8753 8754
    PAGE 633  FUNCTION EST.TR.RANGE                 8766 8767
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9509 9511 9513
DEP.PREC
    PAGE 178  ROUTINE EST.COVERAGE                  8145 8159 8166
    PAGE 190  ROUTINE FINAL.COVERAGE                8695
    PAGE 191                                        8735
DEP.TM
    PAGE 178  ROUTINE EST.COVERAGE                  8148
    PAGE 179                                        8198 8201 8204
    PAGE 190  ROUTINE FINAL.COVERAGE                8697
    PAGE 193                                        8842 8843 8848
DEQ.FEBA.SET
    PAGE  84  ROUTINE CHANGE.LOC                    4187
    PAGE  85                                        4251
    PAGE  87  ROUTINE END.MOVE                      4323 4342
    PAGE 110  ROUTINE POSITION                      5285
    PAGE 325  ROUTINE DEQ.FEBA.SET                  5078 5087
    PAGE 400  EVENT START.BATTLE                    8055
    PAGE 403  EVENT START.MOVE                      8235
    PAGE 404                                        8248
    PAGE 409  EVENT UPDATE.LOC                      8463 8476
DESTIN.X
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES      1593
    PAGE  44  ''SECTION FOR DEFINITIONS             2500
    PAGE 415  EVENT ACT.MOVCOR                      8759
    PAGE 546  ROUTINE ORD.MOVCOR                    5234 5242 5242
    PAGE 687  ''PROGRAM  OLDER.VERSION              1032
    PAGE 703                                        1921
```

```
DESTIN.Y
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1594
    PAGE  44  ''SECTION FOR DEFINITIONS                 2501
    PAGE 415  EVENT ACT.MOVCOR                          8760
    PAGE 546  ROUTINE ORD.MOVCOR                        5235 5243 5243
    PAGE 687  ''PROGRAM  OLDER.VERSION                  1033
    PAGE 703                                            1922
DESTROYED.
    PAGE 350  EVENT AD.ENGAGEMENT                       5970
DESTRUCT.INDIC
    PAGE 435  PROCESS ARTY.ASSESS                       9694 9715 9730
    PAGE 436                                            9778
    PAGE 437                                            9825
DETECTED.
    PAGE 350  EVENT AD.ENGAGEMENT                       5988
    PAGE 351                                            6023 6030 6034
DETECTED.UNIT
    PAGE  40  ''SECTION FOR EVENTS                      2261
    PAGE  81  ROUTINE BLOCK.LOS                         4071 4073
    PAGE 107  ROUTINE NEW.SEGMENT                       5156
    PAGE 108                                            5164
    PAGE 124  ROUTINE TIME.TO.DETECT                    5854
    PAGE 699  ''PROGRAM  OLDER.VERSION                  1698
DETECTING.UNIT
    PAGE  40  ''SECTION FOR EVENTS                      2260
    PAGE  81  ROUTINE BLOCK.LOS                         4072 4074
    PAGE 107  ROUTINE NEW.SEGMENT                       5157
    PAGE 108                                            5165
    PAGE 124  ROUTINE TIME.TO.DETECT                    5853
    PAGE 328  ROUTINE EMPTY                             5191
    PAGE 369  EVENT ENGAGEMENT                          6700
    PAGE 699  ''PROGRAM  OLDER.VERSION                  1697
DETECT.QUANT
    PAGE 212  ROUTINE PIR.DETECTION                     9756
    PAGE 213                                            9760 9760 9763
    PAGE 229  ROUTINE RPV.DETECTION                      569
    PAGE 230                                             573  573  576
    PAGE 255  ROUTINE FO.DETECTION                      1774 1779 1779 1782
    PAGE 256                                            1824
    PAGE 656  ROUTINE AR.DETECTION                      9419 9420 9421
DET.CAND
    PAGE 238  ROUTINE SWITCH.FO                          923  939  941  942  955  957  958
DET.LINK
    PAGE 213  ROUTINE PIR.DETECTION                     9776 9777 9780 9781 9782 9783 9784 9786
    PAGE 230  ROUTINE RPV.DETECTION                      607  608  611  612  613  614  615  617
    PAGE 362  EVENT CFR.OPERATOR                        6524 6525 6526
    PAGE 363                                            6555 6556 6557 6578 6579
    PAGE 387  EVENT PDB.OPERATOR                        7541 7542 7543 7544 7545 7556 7557
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                   9562 9563
DET.PROB
    PAGE 165  ROUTINE CFR.DEGRADE                       7578 7582 7608 7611 7613
    PAGE 166  ROUTINE CFR.DETECTION                     7625
    PAGE 167                                            7684 7685 7691
    PAGE 205  ROUTINE NOISE.DEGRADE                     9385 9389 9418 9425
    PAGE 206  ROUTINE PDB.DETECTION                     9436 9449 9451 9453 9457
```

```
DET.QUANT
     PAGE 213  ROUTINE PIR.DETECTION                    9767
     PAGE 230  ROUTINE RPV.DETECTION                     580
     PAGE 268  ROUTINE BTRY.EFFECTS                     2454 2461 2462 2464 2467 2469
DET.UNIT
     PAGE 161  ROUTINE ATTRIT.SENSOR                    7430 7432 7433 7472 7474 7475
DFFB.MAX.RANGE
     PAGE   7  ''SECTION FOR PERMANENT_ENTITIES          390
     PAGE  42  ''SECTION FOR DEFINITIONS                2377
     PAGE 560  ROUTINE  RUL.EN.INPUT                    5830
     PAGE 635  FUNCTION FEBA.BAND                       8846
     PAGE 667  ''PROGRAM  OLDER.VERSION                 9832
     PAGE 701                                           1807
DF.NOISE
     PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES         1247
     PAGE 205  ROUTINE NOISE.DEGRADE                    9396
     PAGE 310  ROUTINE AD.SHOOT                         4484 4485 4486 4487
     PAGE 499  PROCESS SHOOT.OUT                        3106 3107 3108 3109
     PAGE 681  ''PROGRAM  OLDER.VERSION                  686
DF.RATE.LIST
     PAGE  15  ''SECTION FOR PERMANENT_ENTITIES          833
     PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES         1251
     PAGE  49  ''SECTION FOR DEFINITIONS                2805
     PAGE 205  ROUTINE NOISE.DEGRADE                    9392 9395
     PAGE 310  ROUTINE AD.SHOOT                         4487
     PAGE 499  PROCESS SHOOT.OUT                        3109
     PAGE 674  ''PROGRAM  OLDER.VERSION                  275
     PAGE 681                                            690
     PAGE 708                                           2226
DF.TIME
     PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES         1248
     PAGE  48  ''SECTION FOR DEFINITIONS                2726
     PAGE 205  ROUTINE NOISE.DEGRADE                    9394
     PAGE 310  ROUTINE AD.SHOOT                         4485
     PAGE 499  PROCESS SHOOT.OUT                        3107
     PAGE 681  ''PROGRAM  OLDER.VERSION                  687
     PAGE 706                                           2148
DF.UNIT
     PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES         1249
     PAGE 205  ROUTINE NOISE.DEGRADE                    9399
     PAGE 310  ROUTINE AD.SHOOT                         4486
     PAGE 499  PROCESS SHOOT.OUT                        3108
     PAGE 681  ''PROGRAM  OLDER.VERSION                  688
DIM.F
     PAGE  58  ROUTINE CREATE.FORCE                     3104
     PAGE  67  ROUTINE GENERAL.BATTLE                   3458 3459
     PAGE  75  ROUTINE UNIT.ASSIGNMENT                  3837
     PAGE  77  ROUTINE ADJUST                           3897
     PAGE  96  ROUTINE LINE.OF.SIGHT                    4679 4680
     PAGE 112  ROUTINE PREPARE.LIST                     5368
     PAGE 125  ROUTINE WHAT.NEXT                        5890 5928
     PAGE 130  ROUTINE CHECK.DEAD                       6078 6079 6097 6122
     PAGE 147  ROUTINE INTER.BATTLE                     6818
     PAGE 148                                           6855
     PAGE 268  ROUTINE BTRY.EFFECTS                     2443
     PAGE 300  ROUTINE HC.DISENGAGE                     4017
```

```
    PAGE 311  ROUTINE INTER.HELO                    4556
    PAGE 395  EVENT START.BATTLE                    7762 7763
    PAGE 413  EVENT ACT.ATK                         8685 8691
    PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT         8825
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION         712
    PAGE 529  ROUTINE KV.INPUT                      4564
    PAGE 530  ROUTINE EQ.TE.INPUT                   4621
    PAGE 609  ROUTINE KV.SCOREBOARD                 7847 7848 7858 7858 7862
    PAGE 620  ROUTINE SNAP.R                        8321
    PAGE 714  ROUTINE PLAT.COUNT                    2518
DIRECTION.
    PAGE 545  ROUTINE ORD.MOVDIS                    5185
DIR.
    PAGE 385  EVENT OFF.LINE.ATTRITION              7461
    PAGE 416  EVENT ACT.MOVDIS                      8780 8782 8786 8788 8792
    PAGE 499  PROCESS SHOOT.OUT                     3129
    PAGE 514  PROCESS HELICOPTER.FIRE               3942
DIR.OF.MOVE
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES      1603
    PAGE  47  ''SECTION FOR DEFINITIONS             2662
    PAGE 144  ROUTINE DEAD.UNIT                     6678
    PAGE 416  EVENT ACT.MOVDIS                      8779 8785
    PAGE 545  ROUTINE ORD.MOVDIS                    5196
    PAGE 687  ''PROGRAM  OLDER.VERSION              1042
    PAGE 705                                        2086
DISTANCE.INCREMENT
    PAGE  41  ''SECTION FOR DEFINITIONS             2361
    PAGE  56  ROUTINE MAIN1                         3041
    PAGE 499  PROCESS SHOOT.OUT                     3137
    PAGE 700  ''PROGRAM  OLDER.VERSION              1791
DIST.FACTOR
    PAGE 180  ROUTINE EST.MIL.WORTH                 8226 8232 8234 8236
DIST.FEBA
    PAGE 240  ROUTINE TARGET.ANALYSIS               1032 1036
    PAGE 241                                        1103
DIST.FROM.FEBA.BAND
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES       368  371  389
    PAGE   8                                         392
    PAGE 560  ROUTINE  RUL.EN.INPUT                 5826 5828 5830 5834 5841 5847
    PAGE 635  FUNCTION FEBA.BAND                    8845
    PAGE 666  ''PROGRAM  OLDER.VERSION              9810 9813
    PAGE 667                                        9831 9834
    PAGE 711                                        2431
DIST.MOVED
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES      1604
    PAGE 144  ROUTINE DEAD.UNIT                     6679
    PAGE 416  EVENT ACT.MOVDIS                      8792
    PAGE 545  ROUTINE ORD.MOVDIS                    5197
    PAGE 687  ''PROGRAM  OLDER.VERSION              1043
DIS.ATTACK
    PAGE  42  ''SECTION FOR DEFINITIONS             2369
    PAGE  67  ROUTINE GENERAL.BATTLE                3503
    PAGE  68                                        3507 3512 3514
    PAGE 569  ROUTINE TBF.INPUT                     6210 6215 6215
    PAGE 700  ''PROGRAM  OLDER.VERSION              1799
```

```
DIS.FARRP
    PAGE 300  ROUTINE HC.DISENGAGE                    4005 4020 4052
    PAGE 301                                          4092
DIS.TEAM
    PAGE 300  ROUTINE HC.DISENGAGE                    4006 4020 4026 4046 4048
    PAGE 301                                          4067 4075 4088 4112
DIS.WITH.DRAW
    PAGE  42  ''SECTION FOR DEFINITIONS               2368
    PAGE  67  ROUTINE GENERAL.BATTLE                  3477 3481 3504
    PAGE  68                                          3511 3515
    PAGE 569  ROUTINE TBF.INPUT                       6209 6214 6214
    PAGE 700  ''PROGRAM  OLDER.VERSION                1798
DOSQ.SIDE
    PAGE  40  ''SECTION FOR EVENTS                    2257
    PAGE 699  ''PROGRAM  OLDER.VERSION                1694
DQ.CMSN.QUEUE
    PAGE 327  ROUTINE DQ.CMSN.QUEUE                   5143 5155
    PAGE 365  EVENT CHANGE.LITE                       6612
    PAGE 366  EVENT CHANGE.WEATHER                    6625
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE               6644 6657
    PAGE 508  PROCESS CAS.MISSION                     3612
DQ.OLD.SORTIE.QUEUE
    PAGE  40  ''SECTION FOR EVENTS                    2256
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE               6634 6652
    PAGE 504  PROCESS CAS.MISSION                     3390
    PAGE 601  ROUTINE BETWEEN.ROUTINE                 7584 7585
    PAGE 616  ROUTINE SNAP2                           8175
    PAGE 699  ''PROGRAM  OLDER.VERSION                1693
DROP.DEAD.INDICATOR
    PAGE  38  ''SECTION FOR PROCESSES                 2134
    PAGE  46  ''SECTION FOR DEFINITIONS               2643
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                 2936 2938
    PAGE 282  ROUTINE AC.DF.EFFECTS                   3121 3122
    PAGE 383  EVENT OFF.LINE.ATTRITION                7379 7380
    PAGE 435  PROCESS ARTY.ASSESS                     9738 9740
    PAGE 436                                          9786 9799
    PAGE 464  PROCESS MINE.ASSESS                     1266
    PAGE 465                                          1283 1300
    PAGE 491  PROCESS ASSESSMENT                      2695 2696
    PAGE 493  PROCESS SHOOT.OUT                       2775 2795
    PAGE 496                                          2962
    PAGE 500                                          3167
    PAGE 502                                          3322
    PAGE 516  PROCESS HELICOPTER.FIRE                 4049 4050
    PAGE 697  ''PROGRAM  OLDER.VERSION                1572
    PAGE 705                                          2067
DT.MAX.BATS
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         393
    PAGE  42  ''SECTION FOR DEFINITIONS               2378
    PAGE 184  ROUTINE FA.BN.ASGN                      8451
    PAGE 560  ROUTINE  RUL.EN.INPUT                   5850
    PAGE 667  ''PROGRAM  OLDER.VERSION                9835
    PAGE 701                                          1808
DUM.UNITS.
    PAGE 112  ROUTINE PREPARE.LIST                    5363 5365 5382
    PAGE 113                                          5455 5458
```

```
    PAGE 137  ROUTINE CHECK.LIST                    6332 6341 6345 6357 6370
    PAGE 138  ROUTINE CHECK.PROX                    6388 6390 6415
    PAGE 139                                        6432 6449 6479 6480
    PAGE 140                                        6492 6494 6497 6501
    PAGE 412  EVENT ACT.ATK                         8618 8620 8634 8639 8650
    PAGE 413                                        8659 8664 8675 8698
DUPLICATIONS.
    PAGE 471  PROCESS TARGET.REPORT                 1552
DUP.
    PAGE 225  ROUTINE REQUEST.SMOKE                  385
DYNANL.LIS
    PAGE 647  ROUTINE OPEN.INPUT.OUTPUT.FILES       9185
D.B.H
    PAGE  69  ROUTINE GENERAL.BATTLE                3587 3588 3603 3605
D.B.M
    PAGE  69  ROUTINE GENERAL.BATTLE                3589 3590 3610 3612
D.R.H
    PAGE  69  ROUTINE GENERAL.BATTLE                3591 3592 3619 3621
D.R.M
    PAGE  69  ROUTINE GENERAL.BATTLE                3593 3594
    PAGE  70                                        3626 3628
D.X
    PAGE  67  ROUTINE GENERAL.BATTLE                3498 3500
    PAGE 111  ROUTINE PRED.POS                      5304 5313 5341
    PAGE 321  ROUTINE COMPUTE.D                     4958
    PAGE 322  ROUTINE COMPUTE.WD                    4992
D.X.
    PAGE 321  ROUTINE COMPUTE.D                     4960
    PAGE 322  ROUTINE COMPUTE.WD                    4994
    PAGE 342  ROUTINE RANGE.COMPUTE                 5667 5670 5672 5672
D.Y
    PAGE  67  ROUTINE GENERAL.BATTLE                3499 3500
    PAGE 111  ROUTINE PRED.POS                      5305 5314 5342
    PAGE 321  ROUTINE COMPUTE.D                     4959
    PAGE 322  ROUTINE COMPUTE.WD                    4993
D.Y.
    PAGE 321  ROUTINE COMPUTE.D                     4960
    PAGE 322  ROUTINE COMPUTE.WD                    4994
    PAGE 342  ROUTINE RANGE.COMPUTE                 5667 5671 5672 5672
EAAT.AC.DET.TIME
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES      1258
    PAGE 419  PROCESS AC.ATK.TGT                    8860
    PAGE 681  ''PROGRAM  OLDER.VERSION               697
EAAT.BLIND
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES      1259
    PAGE 419  PROCESS AC.ATK.TGT                    8861
    PAGE 681  ''PROGRAM  OLDER.VERSION               698
EAAT.DETECT.TIME
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES      1260
    PAGE  49  ''SECTION FOR DEFINITIONS             2784
    PAGE 419  PROCESS AC.ATK.TGT                    8862
    PAGE 682  ''PROGRAM  OLDER.VERSION               699
    PAGE 707                                        2205
EAAT.DIST.TO.FP
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES      1261
    PAGE 419  PROCESS AC.ATK.TGT                    8863
```

```
    PAGE 682  ''PROGRAM  OLDER.VERSION                700
EAAT.DIST.TO.P3
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES       1262
    PAGE 419  PROCESS AC.ATK.TGT                     8864
    PAGE 682  ''PROGRAM  OLDER.VERSION                701
EAAT.FIRING.RANGE
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1263
    PAGE 419  PROCESS AC.ATK.TGT                     8865
    PAGE 682  ''PROGRAM  OLDER.VERSION                702
EAAT.FIRING.TIME
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1264
    PAGE  49  ''SECTION FOR DEFINITIONS              2786
    PAGE 419  PROCESS AC.ATK.TGT                     8866
    PAGE 682  ''PROGRAM  OLDER.VERSION                703
    PAGE 707                                         2206
EAAT.MODEL.ADS
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1265
    PAGE 419  PROCESS AC.ATK.TGT                     8867
    PAGE 682  ''PROGRAM  OLDER.VERSION                704
EAAT.NUM.FIRES.P
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1266
    PAGE 419  PROCESS AC.ATK.TGT                     8868
    PAGE 682  ''PROGRAM  OLDER.VERSION                705
EAAT.NUM.PASSES
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1267
    PAGE 419  PROCESS AC.ATK.TGT                     8869
    PAGE 682  ''PROGRAM  OLDER.VERSION                706
EAAT.P1.TO.P2
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1268
    PAGE 419  PROCESS AC.ATK.TGT                     8870
    PAGE 682  ''PROGRAM  OLDER.VERSION                707
EAAT.P2.TO.P3
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1269
    PAGE 419  PROCESS AC.ATK.TGT                     8871
    PAGE 682  ''PROGRAM  OLDER.VERSION                708
EAAT.P3.TO.P1
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1270
    PAGE 419  PROCESS AC.ATK.TGT                     8872
    PAGE 682  ''PROGRAM  OLDER.VERSION                709
EAAT.RANGE
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1271
    PAGE 419  PROCESS AC.ATK.TGT                     8873
    PAGE 682  ''PROGRAM  OLDER.VERSION                710
EAAT.RANGE.AT.FP
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1272
    PAGE 419  PROCESS AC.ATK.TGT                     8874
    PAGE 682  ''PROGRAM  OLDER.VERSION                711
EAAT.RESULT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1273
    PAGE 419  PROCESS AC.ATK.TGT                     8875
    PAGE 682  ''PROGRAM  OLDER.VERSION                712
EAAT.ROW
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1274
    PAGE 419  PROCESS AC.ATK.TGT                     8876
    PAGE 682  ''PROGRAM  OLDER.VERSION                713
```

```
EAAT.SUM.CRIT.EQ
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES          1275
    PAGE 419  PROCESS AC.ATK.TGT                        8877
    PAGE 682  ''PROGRAM  OLDER.VERSION                   714
EAAT.TIME.TO.LEA
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES          1276
    PAGE  49  ''SECTION FOR DEFINITIONS                 2787
    PAGE 419  PROCESS AC.ATK.TGT                        8878
    PAGE 682  ''PROGRAM  OLDER.VERSION                   715
    PAGE 708                                            2207
EAAT.WAIT.TIME
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES          1277
    PAGE  49  ''SECTION FOR DEFINITIONS                 2785
    PAGE 419  PROCESS AC.ATK.TGT                        8879
    PAGE 682  ''PROGRAM  OLDER.VERSION                   716
    PAGE 708                                            2208
EC.FRACT
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES           399
    PAGE 242  ROUTINE UNIT.ENVIR                        1166 1167 1168
    PAGE 262  ROUTINE BTRY.EFFECTS                      2125
    PAGE 263                                            2172
    PAGE 547  ROUTINE P.E.M.INPUT                       5287
    PAGE 636  FUNCTION HE.WLA                           8905
    PAGE 637                                            8918 8924
    PAGE 638  FUNCTION ICM.WLA                          9010
    PAGE 639                                            9018
    PAGE 667  ''PROGRAM  OLDER.VERSION                  9841
EDGE1.DIST
    PAGE 343  ROUTINE SEARCH.COVERAGE                   5698 5701
EDGE2.DIST
    PAGE 343  ROUTINE SEARCH.COVERAGE                   5699 5702
EENT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING               54
    PAGE  48  ''SECTION FOR DEFINITIONS                 2734
    PAGE  80  ROUTINE BLOCK.LOS                         4024
    PAGE  83  ROUTINE CHANGE.LOC                        4150
    PAGE 106  ROUTINE MIN.MOVE                          5085
    PAGE 150  ROUTINE PK.COMPUTE                        7001
    PAGE 365  EVENT CHANGE.LITE                         6600 6601 6603
    PAGE 523  ROUTINE SYS.INPUT                         4353
    PAGE 707  ''PROGRAM  OLDER.VERSION                  2156
EE.ABS
    PAGE 149  ROUTINE PK.COMPUTE                        6907
    PAGE 150                                            6982 6983
EE.DIF
    PAGE 149  ROUTINE PK.COMPUTE                        6907
    PAGE 150                                            6969 6970 6978 6982
EE.FAC
    PAGE 149  ROUTINE PK.COMPUTE                        6907
    PAGE 150                                            6978 6983 6999
EE.INT
    PAGE 150  ROUTINE PK.COMPUTE                        6968 6969 6972 6973 6975 6977 6980 6981
EE.REAL
    PAGE 149  ROUTINE PK.COMPUTE                        6907
    PAGE 150                                            6967 6968 6969 6971
```

```
EFF.EXPOSED.SIZE
    PAGE 154  ROUTINE AO.DETECTION                    7092
    PAGE 156                                          7240 7245
    PAGE 157                                          7248 7249 7251
EFF.EXP.SIZE
    PAGE 156  ROUTINE AO.DETECTION                    7246
EFF.TIME.SCORE
    PAGE 154  ROUTINE AO.DETECTION                    7091
    PAGE 156                                          7228 7241
EFM.FIRE.RATE
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES        1281
    PAGE  48  ''SECTION FOR DEFINITIONS               2731
    PAGE 480  PROCESS FIRE.MISSION                    2049
    PAGE 682  ''PROGRAM  OLDER.VERSION                 720
    PAGE 707                                          2153
EFM.LAST.ARTY.ENGAGE
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES        1283
    PAGE 480  PROCESS FIRE.MISSION                    2053
    PAGE 682  ''PROGRAM  OLDER.VERSION                 722
EFM.OLD.CEP
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES        1284
    PAGE 480  PROCESS FIRE.MISSION                    2051
    PAGE 682  ''PROGRAM  OLDER.VERSION                 723
EFM.PREP.TIME
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES        1280
    PAGE  48  ''SECTION FOR DEFINITIONS               2730
    PAGE 480  PROCESS FIRE.MISSION                    2048
    PAGE 682  ''PROGRAM  OLDER.VERSION                 719
    PAGE 707                                          2152
EFM.SUPPRESS
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES        1282
    PAGE  48  ''SECTION FOR DEFINITIONS               2732
    PAGE 480  PROCESS FIRE.MISSION                    2050
    PAGE 682  ''PROGRAM  OLDER.VERSION                 721
    PAGE 707                                          2154
EFM.SUPP.TIME
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES        1285
    PAGE 480  PROCESS FIRE.MISSION                    2052
    PAGE 682  ''PROGRAM  OLDER.VERSION                 724
EFO.CANDIDATE
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES        1289
    PAGE 438  PROCESS FORWARD.OBSERVER                9859
    PAGE 682  ''PROGRAM  OLDER.VERSION                 728
EFO.ENEMY
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES        1290
    PAGE 438  PROCESS FORWARD.OBSERVER                9860
    PAGE 682  ''PROGRAM  OLDER.VERSION                 729
EFO.FO.UNIT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES        1292
    PAGE  44  ''SECTION FOR DEFINITIONS               2527
    PAGE 438  PROCESS FORWARD.OBSERVER                9862
    PAGE 682  ''PROGRAM  OLDER.VERSION                 731
    PAGE 703                                          1948
EFO.LINK
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES        1295
    PAGE 438  PROCESS FORWARD.OBSERVER                9865
```

```
    PAGE 682  ''PROGRAM  OLDER.VERSION                734
EFO.MAX.XMIT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1297
    PAGE 438  PROCESS FORWARD.OBSERVER               9867
    PAGE 682  ''PROGRAM  OLDER.VERSION                736
EFO.MIN.XMIT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1296
    PAGE 438  PROCESS FORWARD.OBSERVER               9866
    PAGE 682  ''PROGRAM  OLDER.VERSION                735
EFO.PERIOD.OF.SEARCH
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1298
    PAGE  48  ''SECTION FOR DEFINITIONS              2728
    PAGE 438  PROCESS FORWARD.OBSERVER               9868
    PAGE 682  ''PROGRAM  OLDER.VERSION                737
    PAGE 707                                         2150
EFO.SEARCH.TIME
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1291
    PAGE  48  ''SECTION FOR DEFINITIONS              2727
    PAGE 438  PROCESS FORWARD.OBSERVER               9861
    PAGE 682  ''PROGRAM  OLDER.VERSION                730
    PAGE 707                                         2149
EFO.START.TIME
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1301
    PAGE  48  ''SECTION FOR DEFINITIONS              2729
    PAGE 438  PROCESS FORWARD.OBSERVER               9871
    PAGE 682  ''PROGRAM  OLDER.VERSION                740
    PAGE 707                                         2151
EFO.TARGET
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1302
    PAGE 438  PROCESS FORWARD.OBSERVER               9872
    PAGE 682  ''PROGRAM  OLDER.VERSION                741
EFO.X.CORRECT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1293
    PAGE  44  ''SECTION FOR DEFINITIONS              2528
    PAGE 438  PROCESS FORWARD.OBSERVER               9863
    PAGE 682  ''PROGRAM  OLDER.VERSION                732
    PAGE 703                                         1949
EFO.X.SEARCH.GRID
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1299
    PAGE  44  ''SECTION FOR DEFINITIONS              2530
    PAGE 438  PROCESS FORWARD.OBSERVER               9869
    PAGE 682  ''PROGRAM  OLDER.VERSION                738
    PAGE 703                                         1951
EFO.Y.CORRECT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1294
    PAGE  44  ''SECTION FOR DEFINITIONS              2529
    PAGE 438  PROCESS FORWARD.OBSERVER               9864
    PAGE 682  ''PROGRAM  OLDER.VERSION                733
    PAGE 703                                         1950
EFO.Y.SEARCH.GRID
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1300
    PAGE  44  ''SECTION FOR DEFINITIONS              2531
    PAGE 438  PROCESS FORWARD.OBSERVER               9870
    PAGE 682  ''PROGRAM  OLDER.VERSION                739
    PAGE 703                                         1952
```

```
ELEM.ACQUIRE.PROB
    PAGE 212  ROUTINE PIR.DETECTION                9707 9751 9753
    PAGE 213                                       9769 9783
    PAGE 229  ROUTINE RPV.DETECTION                 519  564  566
    PAGE 230                                        582  614
ELEM.DETECT.PROB
    PAGE 212  ROUTINE PIR.DETECTION                9707 9749 9753
    PAGE 213                                       9759 9783
    PAGE 229  ROUTINE RPV.DETECTION                 519  562  566
    PAGE 230                                        572  614
ELEM.PROB.DETECT
    PAGE 154  ROUTINE AO.DETECTION                 7093
    PAGE 157                                       7251 7253 7254 7256 7257 7286
    PAGE 254  ROUTINE FO.DETECTION                 1681
    PAGE 255                                       1768 1770 1778
    PAGE 256                                       1793
    PAGE 656  ROUTINE AR.DETECTION                 9388 9406 9407 9420 9441
EMPLOY.HELIC
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS           3491
    PAGE 290                                       3569
    PAGE 291                                       3613
    PAGE 292                                       3667
EMPLOY.HELICOPTERS
    PAGE 129  ROUTINE BTL.CHECK                    6054
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS           3473 3519
    PAGE 401  EVENT START.BATTLE                   8110 8134
ENCLOSED.
    PAGE 315  ROUTINE FLIGHT.PATH                  4717
ENDED.
    PAGE 293  ROUTINE END.CAS.MISSION              3712
ENDS.
    PAGE 630  FUNCTION COLLISION                   8705
END.
    PAGE 315  ROUTINE FLIGHT.PATH                  4729
END.BATTLE
    PAGE 328  ROUTINE EMPTY                        5169
    PAGE 330                                       5287
END.MINUTE
    PAGE 428  PROCESS AIR.OBSERVER                 9357
    PAGE 429                                       9409 9410
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE         1397 1398
    PAGE 653  PROCESS AIRBORNE.RADAR               9299 9300
END.SIMULATION
    PAGE  39  ''SECTION FOR EVENTS                 2202
    PAGE  57  ROUTINE MAIN3                        3078
    PAGE 368  EVENT END.SIMULATION                 6661
    PAGE 592  ROUTINE AMMO.RPT                     7122
    PAGE 600  ROUTINE BETWEEN.ROUTINE              7528 7529
    PAGE 616  ROUTINE SNAP2                        8162
    PAGE 698  ''PROGRAM  OLDER.VERSION             1639
END.SUPP
    PAGE 273  ROUTINE BTRY.EFFECTS                 2720 2723
ENEMY.
    PAGE 138  ROUTINE CHECK.PROX                   6391 6393 6398 6403 6408 6422 6430
    PAGE 139                                       6447
```

```
ENEMY.DO
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1145
    PAGE 126  ROUTINE WHAT.NEXT                         5945
    PAGE 543  ROUTINE ORD.ATK                           5153
    PAGE 680  ''PROGRAM  OLDER.VERSION                   584
ENEMY.UNIT
    PAGE 379  EVENT HELO.ENGAGEMENT                     7221
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION             821
ENEMY.UNITS
    PAGE  37  ''SECTION FOR PROCESSES                   2080
    PAGE 300  ROUTINE HC.DISENGAGE                      4021
    PAGE 311  ROUTINE INTER.HELO                        4555 4560
    PAGE 338  ROUTINE HC.EMPTY                          5531 5532 5539
    PAGE 339                                            5576
    PAGE 375  EVENT HC.DEPART.BATTLE                    7002
    PAGE 696  ''PROGRAM  OLDER.VERSION                  1518
ENGAGEMENT.
    PAGE 634  FUNCTION FEBA.BAND                        8780
ENGAGE.MENT
    PAGE 107  ROUTINE NEW.SEGMENT                       5155 5156 5157
    PAGE 108                                            5160 5161 5163 5164 5165 5168 5169
ENQ.FEBA.SET
    PAGE  84  ROUTINE CHANGE.LOC                        4192
    PAGE  85                                            4256
    PAGE  87  ROUTINE END.MOVE                          4333 4348
    PAGE 110  ROUTINE POSITION                          5290
    PAGE 331  ROUTINE ENQ.FEBA.SET                      5301 5309
    PAGE 400  EVENT START.BATTLE                        8060
    PAGE 404  EVENT START.MOVE                          8240 8253
    PAGE 409  EVENT UPDATE.LOC                          8468 8481
ENVIRONMENT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING               44
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES            261  264
    PAGE   8                                              395  398  401  404
    PAGE  10                                              519  522
    PAGE  12                                              679
    PAGE  13                                              693
    PAGE  17                                              922
    PAGE 212  ROUTINE PIR.DETECTION                     9724 9729 9746
    PAGE 229  ROUTINE RPV.DETECTION                      537  539  542  559
    PAGE 243  ROUTINE UNIT.ENVIR                        1211
    PAGE 261  ROUTINE BTRY.EFFECTS                      2047
    PAGE 264                                            2245
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                   2895
    PAGE 547  ROUTINE P.E.M.INPUT                       5258 5270 5272 5273 5278 5282 5283 5287
    PAGE 557  ROUTINE SUBM.INPUT                        5727 5729 5733 5736 5740 5745
    PAGE 558  ROUTINE HE.LA.INPUT                       5762 5780 5781 5783 5790
    PAGE 559                                            5811 5812 5814
    PAGE 588  ROUTINE AC.MUNS.INPUT                     7031 7033 7037 7043
    PAGE 618  ROUTINE SNAP.R                            8217
    PAGE 636  FUNCTION HE.WLA                           8911 8911 8912
    PAGE 637                                            8918 8922
    PAGE 638  FUNCTION ICM.WLA                          9002
    PAGE 664  ''PROGRAM  OLDER.VERSION                  9703 9706
    PAGE 667                                            9837 9840 9843 9846
    PAGE 669                                            9960 9963
```

```
    PAGE 672                                          122  136
    PAGE 676                                          364
    PAGE 711                                         2432
ENV.FRACT.TOWN
    PAGE 637  FUNCTION HE.WLA                        8920 8925
ENV.PROP
    PAGE 637  FUNCTION HE.WLA                        8924 8927 8929 8936 8950
ENV.TEST
    PAGE 636  FUNCTION HE.WLA                        8916
    PAGE 637                                         8918
EN.DIS.OP
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES       1243
    PAGE 126  ROUTINE WHAT.NEXT                      5938
    PAGE 144  ROUTINE DEAD.UNIT                      6667
    PAGE 542  ROUTINE ORD.DEF                        5134
    PAGE 681  ''PROGRAM  OLDER.VERSION                682
EN.NAME
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES        396
    PAGE  47  ''SECTION FOR DEFINITIONS              2663
    PAGE 212  ROUTINE PIR.DETECTION                  9731 9733 9735
    PAGE 229  ROUTINE RPV.DETECTION                   544  546  548
    PAGE 261  ROUTINE BTRY.EFFECTS                   2049 2052 2055
    PAGE 265                                         2258 2264 2308
    PAGE 547  ROUTINE P.E.M.INPUT                    5273
    PAGE 636  FUNCTION HE.WLA                        8911
    PAGE 637                                         8926
    PAGE 667  ''PROGRAM  OLDER.VERSION               9838
    PAGE 705                                         2087
EN.UNITS
    PAGE 112  ROUTINE PREPARE.LIST                   5348
    PAGE 405  EVENT START.MOVE                       8334
    PAGE 409  EVENT UPDATE.LOC                       8517
EN.UNITS.
    PAGE  95  ROUTINE INIT.REINF                     4638 4648 4653
    PAGE 112  ROUTINE PREPARE.LIST                   5362 5368 5370 5371 5374
    PAGE 117  ROUTINE PROX.POS                       5567 5575 5592
    PAGE 118  ROUTINE REIN.ARRIVE                    5612 5615 5617 5635
    PAGE 138  ROUTINE CHECK.PROX                     6374 6387 6414
    PAGE 139                                         6488
    PAGE 140                                         6494 6500
    PAGE 347  EVENT ACT.REINF                        5807 5825
    PAGE 348                                         5875 5892
    PAGE 373  EVENT GET.NX.ORD                       6879 6887 6917 6930
    PAGE 374                                         6936 6945 6961 6969
    PAGE 405  EVENT START.MOVE                       8343 8349
    PAGE 406                                         8353
    PAGE 408  EVENT UPDATE.LOC                       8425
    PAGE 409                                         8496 8511
    PAGE 410                                         8521
    PAGE 412  EVENT ACT.ATK                          8603 8615 8655
    PAGE 413                                         8680 8698
EN.UN.PTR
    PAGE  39  ''SECTION FOR EVENTS                   2224
    PAGE 698  ''PROGRAM  OLDER.VERSION               1661
EPS.LA.PERS
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES        405
```

```
    PAGE 190  ROUTINE FINAL_COVERAGE                  8707
    PAGE 266  ROUTINE BTRY_EFFECTS                    2326
    PAGE 557  ROUTINE SUBM_INPUT                      5736
    PAGE 638  FUNCTION ICM_WLA                        9014
    PAGE 667  ''PROGRAM  OLDER_VERSION                9847
EQUIPMENT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING             45
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          407
    PAGE  16                                            881
    PAGE  17                                            923
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES         1687 1689
    PAGE  31                                           1761 1763 1781
    PAGE  32                                           1809 1811 1815
    PAGE 130  ROUTINE CHECK_DEAD                       6084
    PAGE 141  ROUTINE CHECK_STREN                      6506
    PAGE 239  ROUTINE TARGET_ANALYSIS                   978
    PAGE 260  ROUTINE BTRY_EFFECTS                     1970
    PAGE 278  ROUTINE AC_BOMB_EFFECTS                  2909
    PAGE 280  ROUTINE AC_DF_EFFECTS                    3028
    PAGE 283  ROUTINE CAS_EVAL                         3181
    PAGE 293  ROUTINE END_CAS_MISSION                  3733
    PAGE 352  EVENT AD_ENGAGEMENT                      6077
    PAGE 369  EVENT ENGAGEMENT                         6726
    PAGE 370                                           6748
    PAGE 382  EVENT OFF_LINE_ATTRITION                 7301 7307 7310 7316 7325
    PAGE 383                                           7368
    PAGE 421  PROCESS AC_ATK_TGT                       8970
    PAGE 464  PROCESS MINE_ASSESS                      1258
    PAGE 465                                           1307 1329
    PAGE 530  ROUTINE EQ_TE_INPUT                      4597 4599 4602 4603 4604 4605 4607 4609 4610 4611 4612 4612 4625 4626 4628
                                                       4629
    PAGE 534  ROUTINE UNIT_INPUT                       4768 4770 4771 4793
    PAGE 535                                           4851
    PAGE 536                                           4885 4887 4892 4895 4896 4898 4899
    PAGE 542  ROUTINE ORD_DEF                          5121
    PAGE 546  ROUTINE ORD_MOVCOR                       5219 5223 5225
    PAGE 548  ROUTINE TB_INPUT                         5328 5329 5333 5336
    PAGE 575  ROUTINE PGM_INPUT                        6473 6474 6480
    PAGE 583  ROUTINE TACAIR_INPUT                     6798 6799 6806 6811 6812 6818
    PAGE 585                                           6933 6934 6942
    PAGE 596  ROUTINE ANALYSIS_OUTPUT                  7301
    PAGE 609  ROUTINE KV_SCOREBOARD                    7882
    PAGE 610                                           7917 7919 7922
    PAGE 618  ROUTINE SNAP_R                           8218
    PAGE 636  FUNCTION HE_WLA                          8862 8863
    PAGE 638  FUNCTION ICM_WLA                         8962 8963
    PAGE 667  ''PROGRAM  OLDER_VERSION                 9849
    PAGE 675                                            323
    PAGE 676                                            365
    PAGE 689                                           1126 1128
    PAGE 690                                           1200 1202 1211 1220
    PAGE 691                                           1248 1250 1254
    PAGE 711                                           2433
EQUIPMENT.
    PAGE 584  ROUTINE TACAIR_INPUT                     6897
```

```
EQUIP.PK.PTR
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          414
    PAGE  44  ''SECTION FOR DEFINITIONS                2494
    PAGE 149  ROUTINE PK.COMPUTE                       6940
    PAGE 281  ROUTINE AC.DF.EFFECTS                    3052
    PAGE 307  ROUTINE AD.SHOOT                         4352
    PAGE 530  ROUTINE EQ.TE.INPUT                      4609
    PAGE 667  ''PROGRAM  OLDER.VERSION                 9856
    PAGE 702                                           1915
EQUIP.TYPE
    PAGE 154  ROUTINE AO.DETECTION                     7126 7128 7131
    PAGE 155                                           7132 7144
    PAGE 156                                           7229 7233 7236
    PAGE 157                                           7280 7285
    PAGE 212  ROUTINE PIR.DETECTION                    9745 9748 9750
    PAGE 213                                           9777 9782
    PAGE 229  ROUTINE RPV.DETECTION                     558  561  563
    PAGE 230                                            605  608  613
    PAGE 255  ROUTINE FO.DETECTION                     1767 1769
    PAGE 256                                           1787 1792 1796 1815
    PAGE 370  EVENT ENGAGEMENT                         6746
    PAGE 429  PROCESS AIR.OBSERVER                     9411 9413 9414 9416 9416 9418 9419
    PAGE 656  ROUTINE AR.DETECTION                     9404 9410
EQ.
    PAGE 280  ROUTINE AC.DF.EFFECTS                    3042
EQ.AD.INDICATOR
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          408
    PAGE 307  ROUTINE AD.SHOOT                         4331
    PAGE 352  EVENT AD.ENGAGEMENT                      6073
    PAGE 508  PROCESS CAS.MISSION                      3591
    PAGE 530  ROUTINE EQ.TE.INPUT                      4610
    PAGE 667  ''PROGRAM  OLDER.VERSION                 9850
EQ.KILLED
    PAGE 435  PROCESS ARTY.ASSESS                      9707
    PAGE 436                                           9747 9791
    PAGE 437                                           9829 9838 9844
EQ.KV.ID
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          410
    PAGE 250  ROUTINE MINE.EFFECTS                     1512
    PAGE 251                                           1560
    PAGE 260  ROUTINE BTRY.EFFECTS                     2001
    PAGE 264                                           2227 2232
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                  2818
    PAGE 279                                           2979 2979
    PAGE 280  ROUTINE AC.DF.EFFECTS                    3052
    PAGE 282                                           3154 3154
    PAGE 309  ROUTINE AD.SHOOT                         4453 4453
    PAGE 310                                           4472
    PAGE 384  EVENT OFF.LINE.ATTRITION                 7406 7406 7407 7407 7418 7420
    PAGE 435  PROCESS ARTY.ASSESS                      9698 9712
    PAGE 464  PROCESS MINE.ASSESS                      1251
    PAGE 465                                           1330
    PAGE 487  PROCESS ASSESSMENT                       2461 2462
    PAGE 489                                           2550 2551
    PAGE 490                                           2613 2614
    PAGE 492                                           2705 2706
```

```
    PAGE 499  PROCESS SHOOT.OUT                       3111 3114
    PAGE 514  PROCESS HELICOPTER.FIRE                 3920
    PAGE 516                                          4061 4062
    PAGE 530  ROUTINE EQ.TE.INPUT                     4625
    PAGE 535  ROUTINE UNIT.INPUT                      4801
    PAGE 536                                          4892 4896 4899
    PAGE 593  ROUTINE AMMO.RPT                        7181 7183 7197 7199 7214 7216
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                 7310
    PAGE 667  ''PROGRAM  OLDER.VERSION                9852
EQ.MAX.SPEED
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         412
    PAGE  44  ''SECTION FOR DEFINITIONS               2525
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                3962
    PAGE 530  ROUTINE EQ.TE.INPUT                     4604
    PAGE 583  ROUTINE TACAIR.INPUT                    6851
    PAGE 667  ''PROGRAM  OLDER.VERSION                9854
    PAGE 703                                          1946
EQ.NAME
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         409
    PAGE  47  ''SECTION FOR DEFINITIONS               2664
    PAGE 155  ROUTINE AO.DETECTION                    7143
    PAGE 157                                          7259
    PAGE 183  ROUTINE FA.BN.ASGN                      8387
    PAGE 227  ROUTINE REQUEST.WD.FASCAM                457
    PAGE 250  ROUTINE MINE.EFFECTS                    1509
    PAGE 252                                          1584 1594
    PAGE 263  ROUTINE BTRY.EFFECTS                    2140
    PAGE 268                                          2447
    PAGE 269                                          2518
    PAGE 270                                          2552
    PAGE 271                                          2649 2650
    PAGE 272                                          2660 2661
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS           3310 3320 3335
    PAGE 286                                          3357 3368 3384
    PAGE 287                                          3404
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS              3499 3508
    PAGE 295  ROUTINE END.CAS.MISSION                 3797 3823
    PAGE 297  ROUTINE FARRP.CHECK                     3877 3884
    PAGE 309  ROUTINE AD.SHOOT                        4423 4426 4439 4442
    PAGE 377  EVENT HELO.ENGAGEMENT                   7098
    PAGE 378                                          7203
    PAGE 382  EVENT OFF.LINE.ATTRITION                7302 7311
    PAGE 392  EVENT SET.DEBUG                         7644
    PAGE 421  PROCESS AC.ATK.TGT                      9003 9012
    PAGE 452  PROCESS HC.RETURN.FARRP                  596  604
    PAGE 487  PROCESS ASSESSMENT                      2465 2468
    PAGE 489                                          2579 2582
    PAGE 490                                          2617 2620
    PAGE 491                                          2671 2674
    PAGE 492                                          2724 2727
    PAGE 499  PROCESS SHOOT.OUT                       3144 3147
    PAGE 500                                          3177 3180 3196 3199
    PAGE 501                                          3218 3221 3246 3249
    PAGE 502                                          3299 3302
    PAGE 511  PROCESS HELICOPTER.FIRE                 3760 3761
    PAGE 514                                          3910
```

```
    PAGE 515                                        3971 3974 3996 3999
    PAGE 516                                        4012 4015
    PAGE 517                                        4075 4078 4091 4094
    PAGE 518                                        4134 4137 4150 4153
    PAGE 530  ROUTINE EQ.TE.INPUT                   4602
    PAGE 534  ROUTINE UNIT.INPUT                    4770 4784
    PAGE 536                                        4895 4898
    PAGE 537                                        4918
    PAGE 548  ROUTINE TB.INPUT                      5329
    PAGE 575  ROUTINE PGM.INPUT                     6474
    PAGE 583  ROUTINE TACAIR.INPUT                  6799 6812
    PAGE 585                                        6934
    PAGE 592  ROUTINE AMMO.RPT                      7138 7140
    PAGE 593                                        7186 7189 7203 7206
    PAGE 596  ROUTINE ANALYSIS.OUTPUT               7309
    PAGE 607  ROUTINE KV.PRINT                      7750 7762
    PAGE 608                                        7804 7821
    PAGE 609  ROUTINE KV.SCOREBOARD                 7867 7873
    PAGE 610                                        7923
    PAGE 611  ROUTINE OUTPUT.ATTRITION              7983 7984
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8413 8416
    PAGE 623                                        8451
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES           8537
    PAGE 625                                        8567 8602
    PAGE 626                                        8617
    PAGE 667  ''PROGRAM  OLDER.VERSION              9851
    PAGE 705                                        2088
EQ.PAX.KILL.RATE
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES       415
    PAGE 530  ROUTINE EQ.TE.INPUT                   4607
EQ.PERSONNEL.LOAD
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES       413
    PAGE  44  ''SECTION FOR DEFINITIONS             2526
    PAGE 530  ROUTINE EQ.TE.INPUT                   4605
    PAGE 667  ''PROGRAM  OLDER.VERSION              9855
    PAGE 703                                        1947
EQ.QUANT
    PAGE 610  ROUTINE KV.SCOREBOARD                 7907 7908
EQ.SEQ.NO
    PAGE 530  ROUTINE EQ.TE.INPUT                   4601
EQ.TE.INPUT
    PAGE 520  ROUTINE MAIN2                         4205 4207
    PAGE 530  ROUTINE EQ.TE.INPUT                   4573
EQ.TE.PTR
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES       411
    PAGE 123  ROUTINE TIME.TO.DETECT                5836
    PAGE 149  ROUTINE PK.COMPUTE                    6944 6945
    PAGE 154  ROUTINE AO.DETECTION                  7126
    PAGE 183  ROUTINE FA.BN.ASGN                    8376
    PAGE 208  ROUTINE PGM.MSN.ASGN                  9544
    PAGE 212  ROUTINE PIR.DETECTION                 9745
    PAGE 229  ROUTINE RPV.DETECTION                  558
    PAGE 251  ROUTINE MINE.EFFECTS                  1559
    PAGE 252                                        1610
    PAGE 255  ROUTINE FO.DETECTION                  1767
    PAGE 264  ROUTINE BTRY.EFFECTS                  2209 2226
```

```
    PAGE 268                                        2457
    PAGE 271                                        2604 2649
    PAGE 272                                        2660
    PAGE 277  ROUTINE AC.BOMB.EFFECTS               2890
    PAGE 278                                        2956
    PAGE 280  ROUTINE AC.DF.EFFECTS                 3021
    PAGE 282                                        3138 3139
    PAGE 283  ROUTINE CAS.EVAL                      3194
    PAGE 309  ROUTINE AD.SHOOT                      4424 4427 4440 4443
    PAGE 362  EVENT CFR.OPERATOR                    6525
    PAGE 387  EVENT PDB.OPERATOR                    7542
    PAGE 458  PROCESS HEL.TARGET.ACQUISITION         928
    PAGE 459                                        1035
    PAGE 480  PROCESS FIRE.MISSION                  2069
    PAGE 487  PROCESS ASSESSMENT                    2466
    PAGE 488                                        2469
    PAGE 489                                        2580 2583
    PAGE 490                                        2618 2621
    PAGE 491                                        2672 2675
    PAGE 492                                        2725 2728
    PAGE 499  PROCESS SHOOT.OUT                     3145 3148
    PAGE 500                                        3178 3181 3197 3200
    PAGE 501                                        3219 3222 3247 3250
    PAGE 502                                        3300 3303
    PAGE 516  PROCESS HELICOPTER.FIRE               4013 4016
    PAGE 517                                        4092 4095
    PAGE 518                                        4151 4154
    PAGE 530  ROUTINE EQ.TE.INPUT                   4603 4611 4612
    PAGE 537  ROUTINE UNIT.INPUT                    4921
    PAGE 596  ROUTINE ANALYSIS.OUTPUT               7309
    PAGE 656  ROUTINE AR.DETECTION                  9404
    PAGE 667  ''PROGRAM  OLDER.VERSION              9853
ERROR.
    PAGE 539  ROUTINE READ.ORDERS                   5008 5016 5021 5030
    PAGE 540                                        5046 5089
    PAGE 541                                        5106
ERROR.MSG
    PAGE  61  ROUTINE FEBA.INITIAL                  3189 3215 3221
    PAGE  62                                        3252
    PAGE 565  ROUTINE SENSOR.INPUT                  6036
    PAGE 567                                        6148 6170
    PAGE 568                                        6194
ERROR.STOP
    PAGE  75  ROUTINE UNIT.ASSIGNMENT               3849
    PAGE 101  ROUTINE LOS.CHECK                     4881
    PAGE 104  ROUTINE MINE.DELAY                    5034
    PAGE 113  ROUTINE PREPARE.LIST                  5434 5438
    PAGE 114  ROUTINE PREP.WITHDRAW                 5472 5485
    PAGE 118  ROUTINE REIN.ARRIVE                   5638 5641
    PAGE 125  ROUTINE WHAT.NEXT                     5901 5910
    PAGE 126                                        5958 5971
    PAGE 133  ROUTINE CHECK.FORCE                   6163 6204
    PAGE 142  ROUTINE DEAD.UNIT                     6539 6550
    PAGE 143                                        6583
    PAGE 145  ROUTINE FIN.BATTLE                    6730
    PAGE 146  ROUTINE INTER.BATTLE                  6779
```

```
    PAGE 147                                        6801 6844 6849
    PAGE 148                                        6881 6886
    PAGE 150  ROUTINE PK.COMPUTE                    6964 6989 6996
    PAGE 173  ROUTINE DUST.EFFECTS                  7913 7927 7933
    PAGE 174                                        7977 7992
    PAGE 200  ROUTINE ILLUM.EFFECTS                 9227 9234
    PAGE 216  ROUTINE REQUEST.FASCAM                9936
    PAGE 217                                        9951
    PAGE 218  ROUTINE REQUEST.ILLUM                    3   10
    PAGE 219                                           66   81
    PAGE 222  ROUTINE REQUEST.SMOKE                   192  198
    PAGE 223                                          239  254
    PAGE 226  ROUTINE REQUEST.WD.FASCAM               410  416
    PAGE 231  ROUTINE SIZE.ESTIMATE                   663  669
    PAGE 235  ROUTINE SMOKE.EFFECTS                   801  817  824
    PAGE 236                                          901
    PAGE 261  ROUTINE BTRY.EFFECTS                   2071
    PAGE 263                                         2170 2187
    PAGE 268                                         2449
    PAGE 326  ROUTINE DESTROY.ORD                    5138
    PAGE 328  ROUTINE EMPTY                          5180
    PAGE 345  ROUTINE TERM.CHECK                     5776
    PAGE 369  EVENT ENGAGEMENT                       6710
    PAGE 373  EVENT GET.NX.ORD                       6899
    PAGE 374                                         6972 6975
    PAGE 378  EVENT HELO.ENGAGEMENT                  7159
    PAGE 395  EVENT START.BATTLE                     7775
    PAGE 399                                         7976 7993
    PAGE 430  PROCESS AIR.OBSERVER                   9455
    PAGE 463  PROCESS HOW.REPAIR                     1200
    PAGE 478  PROCESS WITH.DRAW                      1957
    PAGE 479                                         2014
    PAGE 487  PROCESS ASSESSMENT                     2437
    PAGE 488                                         2504
    PAGE 489                                         2538
    PAGE 494  PROCESS SHOOT.OUT                      2856
    PAGE 496                                         2929
    PAGE 497                                         2989 3005 3031
    PAGE 499                                         3134
    PAGE 502                                         3290
    PAGE 513  PROCESS HELICOPTER.FIRE                3852 3863
    PAGE 514                                         3947
    PAGE 530  ROUTINE EQ.TE.INPUT                    4614
    PAGE 563  ROUTINE MPDB.INPUT                     5936
    PAGE 566  ROUTINE SENSOR.INPUT                   6108
    PAGE 568                                         6195
    PAGE 604  ROUTINE ERROR.STOP                     7665
    PAGE 630  FUNCTION COLLISION                     8699 8702
    PAGE 631  FUNCTION COMBINATIONS                  8721
    PAGE 636  FUNCTION HE.WLA                        8914
ERR.F
    PAGE 640  ROUTINE EXPONENTIAL.F                  9035
    PAGE 641  ROUTINE NORMAL.F                       9045
    PAGE 652  ROUTINE GAMMA.F                        9242 9245
ESO.ACQ.TIME
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1316
```

```
    PAGE  48  ''SECTION FOR DEFINITIONS              2733
    PAGE 493  PROCESS SHOOT.OUT                      2765
    PAGE 682  ''PROGRAM  OLDER.VERSION                755
    PAGE 707                                         2155
ESO.CUM1
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1305
    PAGE 493  PROCESS SHOOT.OUT                      2754
    PAGE 682  ''PROGRAM  OLDER.VERSION                744
ESO.CUM2
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1306
    PAGE 493  PROCESS SHOOT.OUT                      2755
    PAGE 682  ''PROGRAM  OLDER.VERSION                745
ESO.DUMMY
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1307
    PAGE 493  PROCESS SHOOT.OUT                      2756
    PAGE 682  ''PROGRAM  OLDER.VERSION                746
ESO.QUANT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1317
    PAGE 493  PROCESS SHOOT.OUT                      2766
    PAGE 682  ''PROGRAM  OLDER.VERSION                756
ESO.RANGE
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1308
    PAGE 487  PROCESS ASSESSMENT                     2425
    PAGE 493  PROCESS SHOOT.OUT                      2757
    PAGE 682  ''PROGRAM  OLDER.VERSION                747
ESO.SCR1
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1309
    PAGE 493  PROCESS SHOOT.OUT                      2758
    PAGE 682  ''PROGRAM  OLDER.VERSION                748
ESO.SCR2
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1310
    PAGE 493  PROCESS SHOOT.OUT                      2759
    PAGE 682  ''PROGRAM  OLDER.VERSION                749
ESO.TGT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1311
    PAGE 493  PROCESS SHOOT.OUT                      2760
    PAGE 682  ''PROGRAM  OLDER.VERSION                750
ESO.TGT.EQUIP
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1312
    PAGE 493  PROCESS SHOOT.OUT                      2761
    PAGE 682  ''PROGRAM  OLDER.VERSION                751
ESO.TGT.UNIT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1313
    PAGE 493  PROCESS SHOOT.OUT                      2762
    PAGE 682  ''PROGRAM  OLDER.VERSION                752
ESO.WEAPON
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1314
    PAGE 493  PROCESS SHOOT.OUT                      2763
    PAGE 682  ''PROGRAM  OLDER.VERSION                753
ESO.WPN
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1315
    PAGE 487  PROCESS ASSESSMENT                     2424
    PAGE 493  PROCESS SHOOT.OUT                      2764
    PAGE 682  ''PROGRAM  OLDER.VERSION                754
ESO..TGT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES       1318
```

```
    PAGE 493  PROCESS SHOOT OUT                        2767
    PAGE 683  ''PROGRAM  OLDER.VERSION                  757
EST.COV
    PAGE 176  ROUTINE EST.COVERAGE                     8034
    PAGE 179                                           8215
EST.COVERAGE
    PAGE   2  PROGRAM REVISIONS                         102
    PAGE   3                                            172
    PAGE 176  ROUTINE EST.COVERAGE                     8004 8024
    PAGE 179                                           8186
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS                 1399
EST.COV.
    PAGE 177  ROUTINE EST.COVERAGE                     8090
EST.EFFECTS
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION          9806
EST.FRACT.COV
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION            9002
    PAGE 197                                           9101
    PAGE 198                                           9108 9125
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ             9301 9311 9315 9316 9336
    PAGE 204                                           9358 9368
EST.MIL.WORTH
    PAGE 180  ROUTINE EST.MIL.WORTH                    8219 8239
    PAGE 241  ROUTINE TARGET.ANALYSIS                  1100
EST.MSN.EFFECTS
    PAGE 182  ROUTINE FA.BN.ASGN                       8304
    PAGE 185                                           8483 8512
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION            8994 9002 9015
    PAGE 197                                           9051
    PAGE 198                                           9129
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ             9304 9311 9324
    PAGE 204                                           9383
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION          9793 9801 9805 9811 9821
EST.RANGE
    PAGE  50  ''SECTION FOR DEFINITIONS                2853
    PAGE 184  ROUTINE FA.BN.ASGN                       8423 8424
    PAGE 208  ROUTINE PGM.MSN.ASGN                     9526 9529
    PAGE 209                                           9556 9558 9603 9604
    PAGE 240  ROUTINE TARGET.ANALYSIS                  1030 1031
    PAGE 632  FUNCTION EST.RANGE                       8744 8757
    PAGE 709  ''PROGRAM  OLDER.VERSION                 2272
EST.TR.RANGE
    PAGE  50  ''SECTION FOR DEFINITIONS                2854
    PAGE 171  ROUTINE COMPARE.TRS                      7834 7835
    PAGE 633  FUNCTION EST.TR.RANGE                    8761 8770
    PAGE 709  ''PROGRAM  OLDER.VERSION                 2273
ES.RELY
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          402
    PAGE 190  ROUTINE FINAL.COVERAGE                   8712
    PAGE 265  ROUTINE BTRY.EFFECTS                     2265
    PAGE 557  ROUTINE SUBM.INPUT                       5729
    PAGE 638  FUNCTION ICM.WLA                         9004
    PAGE 667  ''PROGRAM  OLDER.VERSION                 9844
ETC.
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES         1475
    PAGE 685  ''PROGRAM  OLDER.VERSION                  914
```

```
ETR.DUPLICATE
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1321
    PAGE 470  PROCESS TARGET.REPORT                     1491
    PAGE 683  ''PROGRAM  OLDER.VERSION                   760
ETR.DUR
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1322
    PAGE 470  PROCESS TARGET.REPORT                     1492
    PAGE 683  ''PROGRAM  OLDER.VERSION                   761
ETR.MAX.PREP
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1323
    PAGE 470  PROCESS TARGET.REPORT                     1493
    PAGE 683  ''PROGRAM  OLDER.VERSION                   762
ETR.NEW.STOP
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1324
    PAGE 470  PROCESS TARGET.REPORT                     1494
    PAGE 683  ''PROGRAM  OLDER.VERSION                   763
ETR.RFAF
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1325
    PAGE 470  PROCESS TARGET.REPORT                     1495
    PAGE 683  ''PROGRAM  OLDER.VERSION                   764
ETR.START.TIME
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1326
    PAGE 470  PROCESS TARGET.REPORT                     1496
    PAGE 683  ''PROGRAM  OLDER.VERSION                   765
ETR.TOT.FOLLOW
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1327
    PAGE 470  PROCESS TARGET.REPORT                     1497
    PAGE 683  ''PROGRAM  OLDER.VERSION                   766
EVENT.V
    PAGE 392  EVENT SET.DEBUG                           7631
    PAGE 599  ROUTINE BETWEEN.ROUTINE                   7437 7441 7445
    PAGE 611  ROUTINE OUTPUT.ATTRITION                  7971
    PAGE 614  ROUTINE SNAP2                             8029
    PAGE 615                                            8086
EV.S
    PAGE  58  ROUTINE CREATE.FORCE                      3125 3141
    PAGE  59                                            3147
    PAGE  81  ROUTINE BLOCK.LOS                         4070 4080
    PAGE 103  ROUTINE MINE.DELAY                        4971
    PAGE 104                                            5030 5036 5038 5046
    PAGE 105                                            5050 5052
    PAGE 107  ROUTINE NEW.SEGMENT                       5155
    PAGE 108                                            5163 5172
    PAGE 112  ROUTINE PREPARE.LIST                      5385 5392
    PAGE 114  ROUTINE PREP.WITHDRAW                     5481
    PAGE 124  ROUTINE TIME.TO.DETECT                    5852
    PAGE 159  ROUTINE ATTRIT.SENSOR                     7353
    PAGE 160                                            7404 7423
    PAGE 161                                            7451 7458 7465
    PAGE 173  ROUTINE DUST.EFFECTS                      7909
    PAGE 174                                            7973 7988
    PAGE 206  ROUTINE PDB.DETECTION                     9475
    PAGE 216  ROUTINE REQUEST.FASCAM                    9907 9915
    PAGE 219  ROUTINE REQUEST.ILLUM                       92
    PAGE 220                                              113
    PAGE 224  ROUTINE REQUEST.SMOKE                       290  321
```

```
    PAGE 235  ROUTINE SMOKE.EFFECTS                  797
    PAGE 236                                         897
    PAGE 238  ROUTINE SWITCH.FO                      935  950
    PAGE 261  ROUTINE BTRY.EFFECTS                  2067
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS         3437
    PAGE 293  ROUTINE END.CAS.MISSION               3700 3705
    PAGE 294                                        3748
    PAGE 301  ROUTINE HC.DISENGAGE                  4066 4074 4098 4110
    PAGE 311  ROUTINE INTER.HELO                    4546
    PAGE 312                                        4609 4617
    PAGE 328  ROUTINE EMPTY                         5183 5190 5197 5208 5217
    PAGE 329                                        5225
    PAGE 337  ROUTINE HC.EMPTY                      5474 5485 5500
    PAGE 338                                        5520 5554 5561 5568
    PAGE 350  EVENT AD.ENGAGEMENT                   6008
    PAGE 369  EVENT ENGAGEMENT                      6699
    PAGE 376  EVENT HELO.ENGAGEMENT                 7058
    PAGE 377                                        7105
    PAGE 393  EVENT START.ARTY.MOVEMENT             7667
    PAGE 402  EVENT START.BATTLE                    8159
    PAGE 407  EVENT STOP.ARTY.MOVEMENT              8379
    PAGE 425  PROCESS AC.ATK.TGT                    9247
    PAGE 426                                        9293 9298
    PAGE 427                                        9317
    PAGE 443  PROCESS HC.ARRIVE.BATTLE               130
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION        1081 1087 1092
    PAGE 479  PROCESS WITH.DRAW                     2010
    PAGE 482  PROCESS FIRE.MISSION                  2182
    PAGE 485                                        2365 2371
    PAGE 486                                        2375 2393
    PAGE 488  PROCESS ASSESSMENT                    2506 2513 2518
    PAGE 490                                        2607 2632 2637
    PAGE 505  PROCESS CAS.MISSION                   3430 3433
    PAGE 616  ROUTINE SNAP2                         8141 8142 8143 8144 8145 8146 8147 8148 8149 8150 8151 8153 8154 8155 8157
                                                    8158 8159 8160 8161 8162 8163 8164 8165 8166 8167 8168 8169 8170 8171 8172
                                                    8173 8174 8176 8177 8178 8179 8180 8181 8182 8183 8184 8185 8187 8188 8190
                                                    8192 8193
    PAGE 617                                        8194 8195 8196 8197
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8445
    PAGE 630  FUNCTION COLLISION                    8684
EXCEEDED.
    PAGE 356  EVENT BTL.ENDED                       6257
EXISTS.
    PAGE 138  ROUTINE CHECK.PROX                    6384
EXIST.
    PAGE 553  ROUTINE FBN.FD.INPUT                  5570 5589
EXIT.
    PAGE 353  EVENT AD.ENGAGEMENT                   6130
EXPECTED.
    PAGE 550  ROUTINE BTRY.INPUT                    5430
EXPENDITURES.
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2814
EXPONENTIAL.F
    PAGE 422  PROCESS AC.ATK.TGT                    9074 9075
    PAGE 463  PROCESS HOW.REPAIR                    1192 1193 1207 1208
    PAGE 551  ROUTINE BTRY.INPUT                    5494 5496 5500 5502
```

```
    PAGE 640  ROUTINE EXPONENTIAL.F                    9026
EXPOSURE.DIST
    PAGE 154  ROUTINE AO.DETECTION                     7090
    PAGE 155                                           7132 7133 7137 7144
    PAGE 156                                           7195 7201 7211 7231
EXP.F
    PAGE 157  ROUTINE AO.DETECTION                     7251
    PAGE 190  ROUTINE FINAL.COVERAGE                   8711
    PAGE 192                                           8824
    PAGE 193                                           8855
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ             9345
    PAGE 204                                           9359 9369
    PAGE 231  ROUTINE SIZE.ESTIMATE                     677
    PAGE 254  ROUTINE FO.DETECTION                     1719
    PAGE 265  ROUTINE BTRY.EFFECTS                     2281
    PAGE 266                                           2350 2351 2358 2359 2366 2367
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                  2903 2913
    PAGE 323  ROUTINE CONTRAST.TO.FREQ                 5013
    PAGE 335  ROUTINE FRAC.COMPUTE                     5421 5426 5430
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION          5746 5759
    PAGE 638  FUNCTION ICM.WLA                         9008
    PAGE 639                                           9016
    PAGE 657  FUNCTION AR.PROB.DETECT                  9464
EXT.RND
    PAGE 260  ROUTINE BTRY.EFFECTS                     1988
    PAGE 261                                           2033
    PAGE 268                                           2480 2481
EX.AC.ATK.TGT
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES         1257
    PAGE 419  PROCESS AC.ATK.TGT                       8897
    PAGE 427                                           9327
    PAGE 681  ''PROGRAM  OLDER.VERSION                  696
    PAGE 712                                           2485
EX.FIRE.MISSION
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES         1279
    PAGE 480  PROCESS FIRE.MISSION                     2056 2065 2075
    PAGE 481                                           2111
    PAGE 484                                           2313
    PAGE 485                                           2362
    PAGE 486                                           2387 2406
    PAGE 682  ''PROGRAM  OLDER.VERSION                  718
    PAGE 712                                           2487
EX.FWD.OBSERVER
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES         1288
    PAGE 438  PROCESS FORWARD.OBSERVER                 9877
    PAGE 440                                           9996
    PAGE 441                                             42   83
    PAGE 442                                             89
    PAGE 682  ''PROGRAM  OLDER.VERSION                  727
    PAGE 712                                           2486
EX.LETH.AREA
    PAGE 190  ROUTINE FINAL.COVERAGE                   8714
    PAGE 191                                           8724 8729
    PAGE 192                                           8817 8823
    PAGE 193                                           8854
```

```
EX.SHOOT.OUT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES          1304
    PAGE 493  PROCESS SHOOT.OUT                         2773 2791 2803
    PAGE 495                                            2879
    PAGE 496                                            2958 2970
    PAGE 500                                            3156 3189
    PAGE 502                                            3311
    PAGE 503                                            3330 3345
    PAGE 682  ''PROGRAM  OLDER.VERSION                   743
    PAGE 712                                            2488
EX.TGT.REPORT
    PAGE  23  ''SECTION FOR TEMPORARY_ENTITIES          1320
    PAGE 470  PROCESS TARGET.REPORT                     1499 1518 1535
    PAGE 471                                            1582
    PAGE 472                                            1605 1628
    PAGE 473                                            1659
    PAGE 474                                            1729
    PAGE 475                                            1795
    PAGE 477                                            1909
    PAGE 683  ''PROGRAM  OLDER.VERSION                   759
    PAGE 712                                            2489
EYE.
    PAGE 120  ROUTINE SEARCH                            5727
E.
    PAGE 664  ''PROGRAM  OLDER.VERSION                  9673
FARRP
    PAGE   1  ROUTINE FOR CROSS_REFERENCING               48
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES          1166 1167
    PAGE  24                                            1329
    PAGE 128  ROUTINE BTL.CHECK                         5991 6002
    PAGE 129                                            6061
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS                3475 3479 3483 3490 3491 3495 3529
    PAGE 290                                            3540 3549 3551 3564 3568 3569 3573 3583 3588
    PAGE 291                                            3607 3611 3613 3618 3628 3629
    PAGE 292                                            3662 3666 3667
    PAGE 297  ROUTINE FARRP.CHECK                       3865 3905
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                  3915 3925 3929 3932 3950 3951 3952 3954 3959
    PAGE 299                                            3973
    PAGE 300  ROUTINE HC.DISENGAGE                      4021
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE                 4143
    PAGE 304  ROUTINE REPLACE.HC                        4194 4209 4229 4229
    PAGE 311  ROUTINE INTER.HELO                        4560
    PAGE 337  ROUTINE HC.EMPTY                          5471
    PAGE 338                                            5541
    PAGE 339                                            5583
    PAGE 356  EVENT BTL.ENDED                           6280 6292
    PAGE 375  EVENT HC.DEPART.BATTLE                    6996 7002 7017
    PAGE 391  EVENT SEND.TEAM                           7612
    PAGE 401  EVENT START.BATTLE                        8099 8102 8103 8106 8107 8108 8112 8123 8126 8127 8130 8131 8132 8136
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                    95  108  112  113  115  118  121  131  150
    PAGE 444                                             151  159  162
    PAGE 445                                             226  231  253  258  260  265
    PAGE 448                                             425  431  435  437
    PAGE 450  PROCESS HC.RETURN.FARRP                    506
    PAGE 451                                             575
    PAGE 453                                             648  650
```

```
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          719
    PAGE 455                                          791
    PAGE 457                                          878  896
    PAGE 458                                          963  980
    PAGE 459                                         1013
    PAGE 494  PROCESS SHOOT.OUT                      2838 2839 2842
    PAGE 497                                         3014 3015 3018
    PAGE 535  ROUTINE UNIT.INPUT                     4837
    PAGE 573  ROUTINE FARRP.INPUT                    6413 6413 6419 6427 6428 6428 6429 6429 6430 6430 6433 6433 6434 6436 6436 6436
    PAGE 680  ''PROGRAM  OLDER.VERSION                605  606
    PAGE 683                                          768
FARRP.
    PAGE 535  ROUTINE UNIT.INPUT                     4839
FARRP.CHECK
    PAGE 297  ROUTINE FARRP.CHECK                    3856 3905
    PAGE 401  EVENT START.BATTLE                     8094 8118
FARRP.INPUT
    PAGE 521  ROUTINE MAIN2                          4275 4277
    PAGE 572  ROUTINE FARRP.INPUT                    6327 6374
FASCAM.COMPU
    PAGE 181  ROUTINE FASCAM.COMPUTATION             8280
FASCAM.MUNITION
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES        423
    PAGE 577  ROUTINE MINE.INPUT                     6563 6565
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                7373
    PAGE 618  ROUTINE SNAP.R                         8219
    PAGE 667  ''PROGRAM  OLDER.VERSION               9864
    PAGE 711                                         2434
FASCAM.SUPP.TIME
    PAGE 183  ROUTINE FA.BN.ASGN                     8349
FASCAM.VOLLEYS
    PAGE  42  ''SECTION FOR DEFINITIONS              2417
    PAGE 181  ROUTINE FASCAM.COMPUTATION             8266 8279
    PAGE 577  ROUTINE MINE.INPUT                     6557
    PAGE 701  ''PROGRAM  OLDER.VERSION               1847
FA.BN
    PAGE   2  PROGRAM REVISIONS                        77
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES        428
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES       1349
    PAGE 271  ROUTINE BTRY.EFFECTS                   2602 2603
    PAGE 390  EVENT SCHEDULE.ARTY.MOVEMENT           7587
    PAGE 550  ROUTINE BTRY.INPUT                     5398 5400 5401
    PAGE 618  ROUTINE SNAP.R                         8220
    PAGE 667  ''PROGRAM  OLDER.VERSION               9869
    PAGE 683                                          788
    PAGE 711                                         2435
FA.BN.ASGN
    PAGE   2  PROGRAM REVISIONS                        90
    PAGE   3                                          142
    PAGE 181  ROUTINE FASCAM.COMPUTATION             8246
    PAGE 182  ROUTINE FA.BN.ASGN                     8286 8330 8337
    PAGE 183                                         8348 8355
    PAGE 184                                         8417 8441
    PAGE 475  PROCESS TARGET.REPORT                  1814
```

```
FA.BN.UNIT
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          430
    PAGE 550  ROUTINE BTRY.INPUT                       5407 5435
    PAGE 553  ROUTINE FBN.FD.INPUT                     5565
    PAGE 667  ''PROGRAM  OLDER.VERSION                 9871
FBN.FD.INPUT
    PAGE 520  ROUTINE MAIN2                            4233 4235
    PAGE 553  ROUTINE FBN.FD.INPUT                     5539
FB.BN
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES         1349
    PAGE 208  ROUTINE PGM.MSN.ASGN                     9513
    PAGE 258  ROUTINE FO.DETECTION                     1909
    PAGE 362  EVENT CFR.OPERATOR                       6534
    PAGE 475  PROCESS TARGET.REPORT                    1817
    PAGE 553  ROUTINE FBN.FD.INPUT                     5573
    PAGE 683  ''PROGRAM  OLDER.VERSION                  788
FB.MISSION
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          429
    PAGE  47  ''SECTION FOR DEFINITIONS                2665
    PAGE 550  ROUTINE BTRY.INPUT                       5401
    PAGE 667  ''PROGRAM  OLDER.VERSION                 9870
    PAGE 705                                           2089
FC.NO.UNITS
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                 3946 3947 3948
    PAGE 299                                           3994 3998
FC.X.COORD
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                 3943 3947 3947 3955
    PAGE 299                                           3987 3996
FC.Y.COORD
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                 3944 3948 3948 3956
    PAGE 299                                           3988 3996
FDC
    PAGE   1  ROUTINE FOR CROSS_REFERENCING              45
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          438
    PAGE  38  ''SECTION FOR PROCESSES                  2175 2176
    PAGE  63  ROUTINE FILE.FD.SCHD                     3260 3263 3265 3266 3268 3271 3286 3297 3305 3315
    PAGE  64                                           3323 3339 3340 3349 3353
    PAGE 168  ROUTINE CHK.COMP.TR                      7700 7705 7706 7708
    PAGE 169  ROUTINE CHK.FD.TR                        7725 7735 7736
    PAGE 172  ROUTINE COPY                             7852
    PAGE 182  ROUTINE FA.BN.ASGN                       8293
    PAGE 194  ROUTINE FIND.START.TIME                  8896 8902 8903 8905 8909 8912 8932
    PAGE 208  ROUTINE PGM.MSN.ASGN                     9492 9510
    PAGE 216  ROUTINE REQUEST.FASCAM                   9924
    PAGE 219  ROUTINE REQUEST.ILLUM                      54
    PAGE 222  ROUTINE REQUEST.SMOKE                     227
    PAGE 258  ROUTINE FO.DETECTION                     1904 1928
    PAGE 332  ROUTINE FDC.TR.DEQ                       5342 5345 5346 5347 5349 5349 5350 5353 5354 5355
    PAGE 333  ROUTINE FDC.TR.ENQ                       5360 5365 5368 5370 5370 5371 5373
    PAGE 334  ROUTINE FINISH.COMPUTATION               5381 5395 5396 5403 5404
    PAGE 362  EVENT CFR.OPERATOR                       6529
    PAGE 363                                           6552
    PAGE 470  PROCESS TARGET.REPORT                    1486 1490 1531
    PAGE 471                                           1550 1577 1590
    PAGE 472                                           1614 1634 1638 1654
    PAGE 473                                           1688 1700
```

```
    PAGE 474                                          1724
    PAGE 475                                          1791 1800 1808 1809 1823 1825 1829
    PAGE 476                                          1834 1839 1847 1851 1887
    PAGE 477                                          1891 1895 1902
    PAGE 480  PROCESS FIRE.MISSION                    2033
    PAGE 553  ROUTINE FBN.FD.INPUT                    5542 5543 5549 5553 5555 5556 5557 5558 5559 5570 5574 5579 5582 5585 5587
                                                      5587 5588 5588 5594
    PAGE 565  ROUTINE SENSOR.INPUT                    6054 6064
    PAGE 618  ROUTINE SNAP.R                          8221
    PAGE 619                                          8279 8280
    PAGE 667  ''PROGRAM  OLDER.VERSION                9879
    PAGE 697                                          1613 1614
    PAGE 711                                          2436
FDC.THRESHOLD
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         729
    PAGE 672  ''PROGRAM  OLDER.VERSION                 172
FDC.TR.DEQ
    PAGE 332  ROUTINE FDC.TR.DEQ                      5341
    PAGE 333  ROUTINE FDC.TR.ENQ                      5364
    PAGE 334  ROUTINE FINISH.COMPUTATION              5404
FD.BN.LINK
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES        1348
    PAGE 553  ROUTINE FBN.FD.INPUT                    5563 5573 5574
    PAGE 683  ''PROGRAM  OLDER.VERSION                 787
FD.BN.LIST
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES         450
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES        1351
    PAGE  49  ''SECTION FOR DEFINITIONS               2824
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9510
    PAGE 258  ROUTINE FO.DETECTION                    1907
    PAGE 362  EVENT CFR.OPERATOR                      6532
    PAGE 475  PROCESS TARGET.REPORT                   1800
    PAGE 553  ROUTINE FBN.FD.INPUT                    5574
    PAGE 669  ''PROGRAM  OLDER.VERSION                9891
    PAGE 683                                           790
    PAGE 708                                          2245
FD.COMPLETE.LIST
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         448
    PAGE  38  ''SECTION FOR PROCESSES                 2176
    PAGE  49  ''SECTION FOR DEFINITIONS               2806
    PAGE 168  ROUTINE CHK.COMP.TR                     7708
    PAGE 477  PROCESS TARGET.REPORT                   1895 1902
    PAGE 668  ''PROGRAM  OLDER.VERSION                9889
    PAGE 697                                          1614
    PAGE 708                                          2227
FD.COMPLETE.LIST
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         442
    PAGE 168  ROUTINE CHK.FD.TR                       7736
    PAGE 553  ROUTINE FDC.TR.DEQ                      5349 5353 5354 5355
    PAGE 553  ROUTINE FDC.TR.ENQ                      5370 5371
    PAGE 669  ''PROGRAM  OLDER.VERSION                9883

              ROUTINE FBN.FD.INPUT                    1775

              ''SECTION FOR PERMANENT_ENTITIES         439
              ''SECTION FOR DEFINITIONS               2532
```

```
    PAGE 157  ROUTINE AO.DETECTION                    7302 7303 7305
    PAGE 158                                          7306
    PAGE 256  ROUTINE FO.DETECTION                    1836 1837
    PAGE 258                                          1921 1947 1948 1950 1951 1953 1954
    PAGE 363  EVENT CFR.OPERATOR                      6545
    PAGE 475  PROCESS TARGET.REPORT                   1823
    PAGE 476                                          1839
    PAGE 553  ROUTINE FBN.FD.INPUT                    5594
    PAGE 566  ROUTINE SENSOR.INPUT                    6107
    PAGE 667  ''PROGRAM  OLDER.VERSION                9880
    PAGE 703                                          1953
FD.MAX.TIME
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         441
    PAGE  44  ''SECTION FOR DEFINITIONS               2534
    PAGE 185  ROUTINE FA.BN.ASGN                      8503
    PAGE 211  ROUTINE PGM.MSN.ASGN                    9691
    PAGE 475  PROCESS TARGET.REPORT                   1809
    PAGE 553  ROUTINE FBN.FD.INPUT                    5558
    PAGE 667  ''PROGRAM  OLDER.VERSION                9882
    PAGE 703                                          1955
FD.MIN.TIME
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         440
    PAGE  44  ''SECTION FOR DEFINITIONS               2533
    PAGE 475  PROCESS TARGET.REPORT                   1808
    PAGE 553  ROUTINE FBN.FD.INPUT                    5557
    PAGE 667  ''PROGRAM  OLDER.VERSION                9881
    PAGE 703                                          1954
FD.N.LOST
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         444
    PAGE 667  ''PROGRAM  OLDER.VERSION                9885
FD.N.PROCESSED
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         443
    PAGE 476  PROCESS TARGET.REPORT                   1887
    PAGE 667  ''PROGRAM  OLDER.VERSION                9884
FD.PARENT
    PAGE 553  ROUTINE FBN.FD.INPUT                    5546 5551 5556 5582 5585 5587
    PAGE 554                                          5598
FD.SCHD.LIST
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         449
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES        1362
    PAGE  49  ''SECTION FOR DEFINITIONS               2807
    PAGE  63  ROUTINE FILE.FD.SCHD                    3268 3271 3286 3297 3305 3315
    PAGE  64                                          3339 3340 3349
    PAGE 194  ROUTINE FIND.START.TIME                 8909 8912 8932
    PAGE 668  ''PROGRAM  OLDER.VERSION                9890
    PAGE 683                                           801
    PAGE 708                                          2228
FD.SCHD.MSN
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES        1357
    PAGE  63  ROUTINE FILE.FD.SCHD                    3272 3282 3293 3301 3311
    PAGE  64                                          3331 3332 3345
    PAGE 194  ROUTINE FIND.START.TIME                 8913
    PAGE 683  ''PROGRAM  OLDER.VERSION                 796
FD.TOT.THRESHOLD
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         445
    PAGE 553  ROUTINE FBN.FD.INPUT                    5559
```

```
    PAGE 367  ''PROGRAM  OLDER.VERSION                9886
FD.TR.QUEUE
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES         447
    PAGE  38  ''SECTION FOR PROCESSES                 2175
    PAGE  49  ''SECTION FOR DEFINITIONS               2808
    PAGE 169  ROUTINE CHK.FD.TR                       7735
    PAGE 332  ROUTINE FDC.TR.DEQ                      5345 5350
    PAGE 333  ROUTINE FDC.TR.ENQ                      5364
    PAGE 334  ROUTINE FINISH.COMPUTATION              5399 5403
    PAGE 472  PROCESS TARGET.REPORT                   1634
    PAGE 477                                          1906 1907
    PAGE 619  ROUTINE SNAP.R                          8280
    PAGE 668  ''PROGRAM  OLDER.VERSION                9888
    PAGE 697                                          1613
    PAGE 708                                          2229
FD.UNIT
    PAGE  50  ''SECTION FOR DEFINITIONS               2843
    PAGE 553  ROUTINE FBN.FD.INPUT                    5551 5555 5582 5587
    PAGE 565  ROUTINE SENSOR.INPUT                    6059
    PAGE 568                                          6198
    PAGE 708  ''PROGRAM  OLDER.VERSION                2264
FEBA.ADV
    PAGE 239  ROUTINE TARGET.ANALYSIS                 1021 1025 1027
FEBA.BAND
    PAGE   4  PROGRAM REVISIONS                        185
    PAGE  50  ''SECTION FOR DEFINITIONS               2855
    PAGE 181  ROUTINE FASCAM.COMPUTATION              8254 8255
    PAGE 182  ROUTINE FA.BN.ASGN                      8310 8311
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION           9012 9013
    PAGE 199  ROUTINE ILLUM.COMPUTATION               9145 9146
    PAGE 233  ROUTINE SMOKE.COMPUTATION                699  700
    PAGE 473  PROCESS TARGET.REPORT                   1676 1679
    PAGE 634  FUNCTION FEBA.BAND                      8774
    PAGE 635                                          8854
    PAGE 709  ''PROGRAM  OLDER.VERSION                2274
FEBA.DIST
    PAGE 660  FUNCTION STAY.TIME                      9597 9606 9613
FEBA.DISTANCE
    PAGE 180  ROUTINE EST.MIL.WORTH                   8221 8228 8234 8238
FEBA.FLIGHT
    PAGE  51  ''SECTION FOR SUBSTITUTIONS             2909
    PAGE 428  PROCESS AIR.OBSERVER                    9355
    PAGE 430                                          9470
    PAGE 433                                          9636
    PAGE 709  ''PROGRAM  OLDER.VERSION                2317
FEBA.INITIAL
    PAGE  61  ROUTINE FEBA.INITIAL                    3184
    PAGE 521  ROUTINE MAIN2                           4257 4259
FEBA.SORTIE
    PAGE  40  ''SECTION FOR EVENTS                    2263
    PAGE 371  EVENT FEBA.SORTIE                       6775 6826
    PAGE 372                                          6867 6872
    PAGE 567  ROUTINE SENSOR.INPUT                    6185
    PAGE 601  ROUTINE BETWEEN.ROUTINE                 7592 7593
    PAGE 616  ROUTINE SNAP2                           8178
    PAGE 699  ''PROGRAM  OLDER.VERSION                1700
```

```
FEBA.WIDTH
    PAGE  41  ''SECTION FOR DEFINITIONS              2364
    PAGE  61  ROUTINE FEBA.INITIAL                   3193
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA             4824 4825
    PAGE 315  ROUTINE FLIGHT.PATH                    4725 4739
    PAGE 371  EVENT FEBA.SORTIE                      6819
    PAGE 372                                         6860
    PAGE 523  ROUTINE SYS.INPUT                      4355 4365 4365
    PAGE 700  ''PROGRAM  OLDER.VERSION               1794
FE.SECTOR
    PAGE  40  ''SECTION FOR EVENTS                   2264
    PAGE 699  ''PROGRAM  OLDER.VERSION               1701
FE.SIDE
    PAGE  40  ''SECTION FOR EVENTS                   2265
    PAGE 699  ''PROGRAM  OLDER.VERSION               1702
FE.TIME.INTERVAL
    PAGE  40  ''SECTION FOR EVENTS                   2266
    PAGE  48  ''SECTION FOR DEFINITIONS              2735
    PAGE 699  ''PROGRAM  OLDER.VERSION               1703
    PAGE 707                                         2157
FILENAME1.
    PAGE 606  ROUTINE KV.PRINT                       7715 7717
FILENAME2.
    PAGE 606  ROUTINE KV.PRINT                       7716 7718
FILENAME.
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES            8498 8499
FILE.
    PAGE 552  ROUTINE BTRY.INPUT                     5514
FILE.FD.SCHD
    PAGE  50  ''SECTION FOR DEFINITIONS              2856
    PAGE  63  ROUTINE FILE.FD.SCHD                   3260 3266
    PAGE  64                                         3323 3353
    PAGE 476  PROCESS TARGET.REPORT                  1849
    PAGE 709  ''PROGRAM  OLDER.VERSION               2275
FIND.START.TIME
    PAGE 194  ROUTINE FIND.START.TIME                8891 8903
    PAGE 473  PROCESS TARGET.REPORT                  1683
FINISH.COMPUTATION
    PAGE 334  ROUTINE FINISH.COMPUTATION             5379 5396
    PAGE 470  PROCESS TARGET.REPORT                  1528
    PAGE 471                                         1574
    PAGE 472                                         1651
    PAGE 474                                         1721
    PAGE 475                                         1788
    PAGE 476                                         1888
FIRED.
    PAGE 310  ROUTINE AD.SHOOT                       4490
FIRED.RNDS
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7968 7982 7983
FIRER.UNIT
    PAGE  37  ''SECTION FOR PROCESSES                2133
    PAGE 149  ROUTINE PK.COMPUTE                     6898 6913 6913 6915 6916
    PAGE 369  EVENT ENGAGEMENT                       6735
    PAGE 379  EVENT HELO.ENGAGEMENT                  7215
    PAGE 435  PROCESS ARTY.ASSESS                    9721
    PAGE 464  PROCESS MINE.ASSESS                    1265
```

```
    PAGE 493  PROCESS SHOOT.OUT                    2769 2796
    PAGE 496                                       2963
    PAGE 499                                       3113 3125
    PAGE 500                                       3168
    PAGE 502                                       3323
    PAGE 508  PROCESS CAS.MISSION                  3584
    PAGE 697  ''PROGRAM  OLDER.VERSION             1571
FIRE.
    PAGE 308  ROUTINE AD.SHOOT                     4398
    PAGE 422  PROCESS AC.ATK.TGT                   9071
FIRE.EQUIP
    PAGE 493  PROCESS SHOOT.OUT                    2768 2778 2800 2802
    PAGE 494                                       2807 2813 2821
    PAGE 495                                       2870 2875 2877 2878 2913
    PAGE 496                                       2936 2967 2969 2977
    PAGE 497                                       2984 2985 2986 2994 3001 3002 3003
    PAGE 498                                       3052 3062 3070
    PAGE 499                                       3144 3145
    PAGE 500                                       3172 3174 3177 3178 3196 3197
    PAGE 501                                       3214 3218 3219 3234 3240 3246 3247 3262
    PAGE 502                                       3299 3300 3327
    PAGE 503                                       3329 3335 3344
FIRE.MISSION
    PAGE   2  PROGRAM REVISIONS                      94  113
    PAGE  35  ''SECTION FOR PROCESSES              1994
    PAGE 163  ROUTINE BTRY.FM.DEQ                  7541
    PAGE 184  ROUTINE FA.BN.ASGN                   8414 8438 8459
    PAGE 185                                       8491 8498 8504
    PAGE 209  ROUTINE PGM.MSN.ASGN                 9593
    PAGE 210                                       9660
    PAGE 211                                       9692
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS       2754 2768
    PAGE 435  PROCESS ARTY.ASSESS                  9691 9696 9697 9699 9701
    PAGE 437                                       9846
    PAGE 480  PROCESS FIRE.MISSION                 2027
    PAGE 482                                       2156
    PAGE 486                                       2376
    PAGE 592  ROUTINE AMMO.RPT                     7122
    PAGE 599  ROUTINE BETWEEN.ROUTINE              7472 7473
    PAGE 616  ROUTINE SNAP2                        8147
    PAGE 619  ROUTINE SNAP.R                       8308
    PAGE 694  ''PROGRAM  OLDER.VERSION             1432
FIRE.MISSIONS
    PAGE   3  PROGRAM REVISIONS                      143
FIRE.ON.MOVE
    PAGE 149  ROUTINE PK.COMPUTE                   6918
    PAGE 151                                       7030
FIRE.OTM
    PAGE  50  ''SECTION FOR DEFINITIONS            2838
    PAGE 151  ROUTINE PK.COMPUTE                   7031
    PAGE 526  ROUTINE PK.INPUT                     4444 4446
    PAGE 708  ''PROGRAM  OLDER.VERSION             2259
FIRE.PLAN
    PAGE 260  ROUTINE BTRY.EFFECTS                 2017 2019
    PAGE 272                                       2673 2703
```

```
FIRE.RATE
    PAGE 480  PROCESS FIRE.MISSION                    2049 2079 2082
    PAGE 481                                          2086 2088 2092 2098
    PAGE 484                                          2259
    PAGE 485                                          2321
FIRE.UNIT
    PAGE 493  PROCESS SHOOT.OUT                       2769 2799
    PAGE 495                                          2895 2902 2908 2916
    PAGE 496                                          2939 2966
    PAGE 498                                          3042 3057 3086 3093
    PAGE 499                                          3099 3104 3108 3116 3150
    PAGE 500                                          3171 3183 3202
    PAGE 501                                          3224 3252 3265
    PAGE 502                                          3278 3305 3326
FIRING.
    PAGE 352  EVENT AD.ENGAGEMENT                     6071
FIRING.EQUIP
    PAGE  37  ''SECTION FOR PROCESSES                 2132
    PAGE 369  EVENT ENGAGEMENT                        6734
    PAGE 379  EVENT HELO.ENGAGEMENT                   7214
    PAGE 435  PROCESS ARTY.ASSESS                     9720
    PAGE 464  PROCESS MINE.ASSESS                     1264
    PAGE 487  PROCESS ASSESSMENT                      2461 2465 2466
    PAGE 489                                          2550 2579 2580
    PAGE 490                                          2596 2613 2617 2618
    PAGE 491                                          2658 2671 2672
    PAGE 492                                          2705 2715 2724 2725
    PAGE 493  PROCESS SHOOT.OUT                       2768
    PAGE 499                                          3111 3114
    PAGE 508  PROCESS CAS.MISSION                     3583
    PAGE 697  ''PROGRAM  OLDER.VERSION                1570
FIRING.TABLE
    PAGE   2  PROGRAM REVISIONS                        84
    PAGE   3                                          128
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES        1368
    PAGE  81  ROUTINE BLOCK.LOS                       4103 4117
    PAGE 108  ROUTINE NEW.SEGMENT                     5211 5212 5216
    PAGE 109                                          5220 5221 5228 5229 5233 5237 5238
    PAGE 293  ROUTINE END.CAS.MISSION                 3723
    PAGE 300  ROUTINE HC.DISENGAGE                    4060
    PAGE 301                                          4090 4091 4094 4096
    PAGE 312  ROUTINE INTER.HELO                      4603
    PAGE 329  ROUTINE EMPTY                           5249 5250 5252 5254
    PAGE 369  EVENT ENGAGEMENT                        6721 6722 6723 6724
    PAGE 377  EVENT HELO.ENGAGEMENT                   7092 7093 7094 7095
    PAGE 378                                          7188 7189 7198 7199
    PAGE 419  PROCESS AC.ATK.TGT                      8893
    PAGE 423                                          9092 9124
    PAGE 425                                          9200 9231
    PAGE 426                                          9275
    PAGE 461  PROCESS HEL.TARGET.ACQUISITION          1109 1119
    PAGE 489  PROCESS ASSESSMENT                      2572
    PAGE 491                                          2651
    PAGE 493  PROCESS SHOOT.OUT                       2770
    PAGE 494                                          2813 2814 2816 2817 2821 2822 2823 2826 2827 2828 2829 2830 2831 2833 2834
                                                      2859 2863
```

```
    PAGE 495                                    2871 2875 2876 2886 2887 2891 2901 2914 2917 2919 2921
    PAGE 496                                    2938 2941 2943 2944 2945 2948 2949 2977 2978
    PAGE 497                                    2981 2994 2995 2998 3010 3012 3013 3035
    PAGE 498                                    3051 3053 3061 3063 3072 3074 3075 3092
    PAGE 499                                    3102 3128 3138
    PAGE 500                                    3160 3161 3164 3165 3193 3211
    PAGE 501                                    3263 3266 3268
    PAGE 502                                    3275 3282 3285 3288 3292 3315 3316 3319 3320
    PAGE 507  PROCESS CAS.MISSION               3572
    PAGE 511  PROCESS HELICOPTER.FIRE           3736 3749
    PAGE 619  ROUTINE SNAP.R                    8271
    PAGE 683  ''PROGRAM  OLDER.VERSION           807
    PAGE 712                                    2490
FIRING.TABLES
    PAGE 300  ROUTINE HC.DISENGAGE              4011 4051
    PAGE 301                                    4083
    PAGE 312  ROUTINE INTER.HELO                4591
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION    1078
FIRING.TABLES.
    PAGE 420  PROCESS AC.ATK.TGT                8924
FIRING.UNIT
    PAGE 487  PROCESS ASSESSMENT                2416 2434 2443
    PAGE 488                                    2471
    PAGE 489                                    2534 2535 2553
    PAGE 490                                    2585 2612 2623
    PAGE 491                                    2645 2677
    PAGE 492                                    2708 2730
FIRST.EFFECTS
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ      9311 9345 9346
    PAGE 204                                    9360 9361 9370 9371
FLIGHT.LEG
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES  1385
    PAGE 371  EVENT FEBA.SORTIE                 6820
    PAGE 372                                    6861
    PAGE 429  PROCESS AIR.OBSERVER              9392 9402 9445
    PAGE 431                                    9523 9541
    PAGE 432                                    9572 9606
    PAGE 433                                    9624 9642 9661 9676
    PAGE 469  PROCESS REMOTE.PILOT.VEHICLE      1473
    PAGE 659  PROCESS PHOTO.IR.FLIGHT           9551
    PAGE 684  ''PROGRAM  OLDER.VERSION           824
FLIGHT.RISK
    PAGE 428  PROCESS AIR.OBSERVER              9353 9369 9371
    PAGE 430                                    9470
    PAGE 433                                    9636
FLIGHT.TIME
    PAGE 128  ROUTINE BTL.CHECK                 6016 6035
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS        3534 3548 3586
    PAGE 291                                    3632
    PAGE 297  ROUTINE FARRP.CHECK               3897 3899 3901 3903 3906
    PAGE 298  ROUTINE HC.COMPUTE.TIMES          3920
    PAGE 299                                    3972 3978 3992 3997
    PAGE 338  ROUTINE HC.EMPTY                  5542 5547
    PAGE 339                                    5584 5589
    PAGE 375  EVENT HC.DEPART.BATTLE            7018 7023
    PAGE 428  PROCESS AIR.OBSERVER              9356 9381 9386
```

```
    PAGE 429                                        9409
    PAGE 450  PROCESS HC.RETURN.FARRP                498
    PAGE 453                                         669  672  675
FLT.TIME
    PAGE 428  PROCESS AIR.OBSERVER                  9356 9379 9380 9381
FLT.TIME2
    PAGE 428  PROCESS AIR.OBSERVER                  9356 9380 9381
FLY.
    PAGE 327  ROUTINE DQ.CMSN.QUEUE                 5151
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE             6645
FL.X.END
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES      1388
    PAGE 371  EVENT FEBA.SORTIE                     6823
    PAGE 372                                        6864
    PAGE 429  PROCESS AIR.OBSERVER                  9435 9437
    PAGE 430                                        9448 9450 9463 9465 9491
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1403
    PAGE 468                                        1421 1467
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9510 9531
    PAGE 659                                        9550
    PAGE 684  ''PROGRAM  OLDER.VERSION               827
FL.X.START
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES      1386
    PAGE 371  EVENT FEBA.SORTIE                     6821
    PAGE 372                                        6862
    PAGE 429  PROCESS AIR.OBSERVER                  9435 9437
    PAGE 430                                        9448 9450 9463 9465 9488 9489
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1392 1393 1403 1405
    PAGE 468                                        1419 1467
    PAGE 653  PROCESS AIRBORNE.RADAR                9295
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9502 9510 9512 9529
    PAGE 659                                        9550
    PAGE 684  ''PROGRAM  OLDER.VERSION               825
FL.Y.END
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES      1389
    PAGE 371  EVENT FEBA.SORTIE                     6824
    PAGE 372                                        6865
    PAGE 429  PROCESS AIR.OBSERVER                  9434
    PAGE 430                                        9447 9449 9462 9492
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1402
    PAGE 468                                        1422 1466
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9509
    PAGE 659                                        9532 9549
    PAGE 684  ''PROGRAM  OLDER.VERSION               828
FL.Y.START
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES      1387
    PAGE 371  EVENT FEBA.SORTIE                     6822
    PAGE 372                                        6863
    PAGE 429  PROCESS AIR.OBSERVER                  9434
    PAGE 430                                        9447 9449 9462 9488 9490
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1391 1394 1402 1405
    PAGE 468                                        1420 1466
    PAGE 653  PROCESS AIRBORNE.RADAR                9296
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9501 9509 9512 9530
    PAGE 659                                        9549
    PAGE 684  ''PROGRAM  OLDER.VERSION               826
```

```
FMM.ID
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          424
    PAGE  47  ''SECTION FOR DEFINITIONS                2666
    PAGE 577  ROUTINE MINE.INPUT                       6567
    PAGE 592  ROUTINE AMMO.RPT                         7160
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                  7378
    PAGE 667  ''PROGRAM  OLDER.VERSION                 9865
    PAGE 705                                           2090
FMM.MAX.RANGE
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          425
    PAGE 181  ROUTINE FASCAM.COMPUTATION               8260
    PAGE 272  ROUTINE BTRY.EFFECTS                     2712
    PAGE 483  PROCESS FIRE.MISSION                     2231
    PAGE 577  ROUTINE MINE.INPUT                       6570
    PAGE 667  ''PROGRAM  OLDER.VERSION                 9866
FMM.RND.WT
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          426
    PAGE 577  ROUTINE MINE.INPUT                       6569
    PAGE 592  ROUTINE AMMO.RPT                         7161
    PAGE 667  ''PROGRAM  OLDER.VERSION                 9867
FM.BTRY
    PAGE  35  ''SECTION FOR PROCESSES                  1996
    PAGE  46  ''SECTION FOR DEFINITIONS                2642
    PAGE 163  ROUTINE BTRY.FM.DEQ                      7523
    PAGE 176  ROUTINE EST.COVERAGE                     8027
    PAGE 177                                           8083
    PAGE 178                                           8153 8154 8157 8158 8162 8172 8172 8173 8176 8177
    PAGE 181  ROUTINE FASCAM.COMPUTATION               8253 8278
    PAGE 184  ROUTINE FA.BN.ASGN                       8421
    PAGE 185                                           8496
    PAGE 188  ROUTINE FINAL.COVERAGE                   8570 8574 8576
    PAGE 189                                           8635
    PAGE 190                                           8703 8704
    PAGE 192                                           8808 8809 8818 8819 8823
    PAGE 193                                           8854
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION            9007
    PAGE 199  ROUTINE ILLUM.COMPUTATION                9144 9172
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ             9320
    PAGE 209  ROUTINE PGM.MSN.ASGN                     9601
    PAGE 233  ROUTINE SMOKE.COMPUTATION                 698  729
    PAGE 260  ROUTINE BTRY.EFFECTS                     2000
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS           2739
    PAGE 435  PROCESS ARTY.ASSESS                      9697 9699
    PAGE 476  PROCESS TARGET.REPORT                    1881 1881
    PAGE 480  PROCESS FIRE.MISSION                     2043
    PAGE 611  ROUTINE OUTPUT.ATTRITION                 7968 7969 7970
    PAGE 694  ''PROGRAM  OLDER.VERSION                 1434
    PAGE 705                                           2066
FM.EX.FIRE.MISSION
    PAGE   2  PROGRAM REVISIONS                         114
    PAGE  35  ''SECTION FOR PROCESSES                  2009
    PAGE 480  PROCESS FIRE.MISSION                     2048 2049 2050 2051 2052 2053 2056 2065 2075
    PAGE 481                                           2111 2112
    PAGE 484                                           2313
    PAGE 485                                           2362 2363
    PAGE 486                                           2387 2405 2406
```

```
    PAGE 694  ''PROGRAM  OLDER.VERSION                1447
FM.FIRED.VOLS
    PAGE  35  ''SECTION FOR PROCESSES                 2008
    PAGE  90  ROUTINE FA.BN.MOVEMENT                  4479 4497
    PAGE  91                                          4539
    PAGE 160  ROUTINE ATTRIT.SENSOR                   7402
    PAGE 209  ROUTINE PGM.MSN.ASGN                    9584
    PAGE 260  ROUTINE BTRY.EFFECTS                    2006
    PAGE 263                                          2142 2179 2190
    PAGE 264                                          2198
    PAGE 265                                          2280
    PAGE 266                                          2351 2359 2367
    PAGE 267                                          2372
    PAGE 270                                          2595
    PAGE 271                                          2648 2659
    PAGE 272                                          2673 2694
    PAGE 463  PROCESS HOW.REPAIR                      1185
    PAGE 483  PROCESS FIRE.MISSION                    2239 2240 2255
    PAGE 484                                          2263
    PAGE 611  ROUTINE OUTPUT.ATTRITION                7968
    PAGE 694  ''PROGRAM  OLDER.VERSION                1446
FM.N.VOLS
    PAGE  35  ''SECTION FOR PROCESSES                 2000
    PAGE  90  ROUTINE FA.BN.MOVEMENT                  4477 4479 4495 4497
    PAGE  91                                          4537 4539
    PAGE 160  ROUTINE ATTRIT.SENSOR                   7402
    PAGE 181  ROUTINE FASCAM.COMPUTATION              8267 8270
    PAGE 185  ROUTINE FA.BN.ASGN                      8488
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION           9016
    PAGE 197                                          9103
    PAGE 198                                          9111 9113 9118 9119 9119
    PAGE 199  ROUTINE ILLUM.COMPUTATION               9162 9164 9168 9173
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ            9319 9349
    PAGE 204                                          9363 9381
    PAGE 209  ROUTINE PGM.MSN.ASGN                    9584
    PAGE 211                                          9672 9677 9679 9680
    PAGE 233  ROUTINE SMOKE.COMPUTATION                719  721  725  730
    PAGE 264  ROUTINE BTRY.EFFECTS                    2213 2214 2216
    PAGE 269                                          2498
    PAGE 270                                          2595
    PAGE 463  PROCESS HOW.REPAIR                      1185
    PAGE 476  PROCESS TARGET.REPORT                   1881
    PAGE 481  PROCESS FIRE.MISSION                    2122 2130 2140
    PAGE 482                                          2147
    PAGE 483                                          2237 2240 2255
    PAGE 484                                          2265 2287
    PAGE 485                                          2320 2333 2336
    PAGE 486                                          2398 2402
    PAGE 629  FUNCTION BTRY.AVAILABLE                 8662 8666
    PAGE 694  ''PROGRAM  OLDER.VERSION                1438
FM.PREP.TIME
    PAGE  35  ''SECTION FOR PROCESSES                 2006
    PAGE 210  ROUTINE PGM.MSN.ASGN                    9648
    PAGE 482  PROCESS FIRE.MISSION                    2163
    PAGE 483                                          2212
    PAGE 694  ''PROGRAM  OLDER.VERSION                1444
```

```
FM.PRIORITY
    PAGE  35  ''SECTION FOR PROCESSES                2004
    PAGE  49  ''SECTION FOR DEFINITIONS              2800
    PAGE 184  ROUTINE FA.BN.ASGN                     8420
    PAGE 209  ROUTINE PGM.MSN.ASGN                   9600
    PAGE 694  ''PROGRAM  OLDER.VERSION               1442
    PAGE 708                                         2221
FM.Q.SIZE
    PAGE  35  ''SECTION FOR PROCESSES                2001
    PAGE  49  ''SECTION FOR DEFINITIONS              2798
    PAGE 184  ROUTINE FA.BN.ASGN                     8419
    PAGE 694  ''PROGRAM  OLDER.VERSION               1439
    PAGE 708                                         2219
FM.RANGE
    PAGE  35  ''SECTION FOR PROCESSES                1999
    PAGE  49  ''SECTION FOR DEFINITIONS              2799
    PAGE 176  ROUTINE EST.COVERAGE                   8021
    PAGE 181  ROUTINE FASCAM.COMPUTATION             8260
    PAGE 184  ROUTINE FA.BN.ASGN                     8424 8425 8429
    PAGE 188  ROUTINE FINAL.COVERAGE                 8568
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9009
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9151
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9322
    PAGE 209  ROUTINE PGM.MSN.ASGN                   9604
    PAGE 210                                         9637 9640
    PAGE 233  ROUTINE SMOKE.COMPUTATION               709
    PAGE 262  ROUTINE BTRY.EFFECTS                   2111
    PAGE 272                                         2712
    PAGE 483  PROCESS FIRE.MISSION                   2231
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7985
    PAGE 694  ''PROGRAM  OLDER.VERSION               1437
    PAGE 708                                         2220
FM.RAP.FLAG
    PAGE  35  ''SECTION FOR PROCESSES                1998
    PAGE 184  ROUTINE FA.BN.ASGN                     8426 8433
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9029
    PAGE 476  PROCESS TARGET.REPORT                  1882
    PAGE 694  ''PROGRAM  OLDER.VERSION               1436
FM.START.TIME
    PAGE  35  ''SECTION FOR PROCESSES                2005
    PAGE  48  ''SECTION FOR DEFINITIONS              2736
    PAGE  49                                         2802
    PAGE 185  ROUTINE FA.BN.ASGN                     8497 8499
    PAGE 481  PROCESS FIRE.MISSION                   2109
    PAGE 629  FUNCTION BTRY.AVAILABLE                8659 8661 8662 8665 8666
    PAGE 694  ''PROGRAM  OLDER.VERSION               1443
    PAGE 707                                         2158
    PAGE 708                                         2223
FM.STATUS
    PAGE  35  ''SECTION FOR PROCESSES                1997
    PAGE 163  ROUTINE BTRY.FM.DEQ                    7540
    PAGE 164  ROUTINE BTRY.FM.ENQ                    7564 7566
    PAGE 481  PROCESS FIRE.MISSION                   2123
    PAGE 482                                         2187
    PAGE 694  ''PROGRAM  OLDER.VERSION               1435
```

```
FM.TGT
    PAGE  35  ''SECTION FOR PROCESSES                1995
    PAGE 176  ROUTINE EST.COVERAGE                   8020
    PAGE 178                                         8147
    PAGE 181  ROUTINE FASCAM.COMPUTATION             8252
    PAGE 184  ROUTINE FA.BN.ASGN                     8410 8422
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9008
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9143
    PAGE 200  ROUTINE ILLUM.EFFECTS                  9191
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9318
    PAGE 209  ROUTINE PGM.MSN.ASGN                   9566 9567 9602
    PAGE 216  ROUTINE REQUEST.FASCAM                 9916 9917
    PAGE 220  ROUTINE REQUEST.ILLUM                   114  115  117  121
    PAGE 224  ROUTINE REQUEST.SMOKE                   322  323  325  329
    PAGE 233  ROUTINE SMOKE.COMPUTATION               697
    PAGE 234  ROUTINE SMOKE.EFFECTS                   748
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS               1391
    PAGE 260  ROUTINE BTRY.EFFECTS                   2002
    PAGE 262                                         2122 2125
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS         2751
    PAGE 435  PROCESS ARTY.ASSESS                    9701
    PAGE 480  PROCESS FIRE.MISSION                   2044
    PAGE 481                                         2104 2105
    PAGE 482                                         2148 2152 2155 2160 2161 2162 2175 2185 2194 2194
    PAGE 483                                         2205 2205 2211 2226 2227 2256
    PAGE 484                                         2266 2266 2271 2272 2273 2281 2289
    PAGE 485                                         2326 2329 2345 2346 2349 2352 2365 2370 2371 2374
    PAGE 486                                         2391 2392 2393 2394 2402
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7956 7963 7964 7976
    PAGE 694  ''PROGRAM  OLDER.VERSION               1433
FM.TM
    PAGE  35  ''SECTION FOR PROCESSES                2002
    PAGE 181  ROUTINE FASCAM.COMPUTATION             8264
    PAGE 188  ROUTINE FINAL.COVERAGE                 8563
    PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION          9099
    PAGE 198                                         9107
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9155
    PAGE 200  ROUTINE ILLUM.EFFECTS                  9211 9220
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9321
    PAGE 210  ROUTINE PGM.MSN.ASGN                   9611
    PAGE 233  ROUTINE SMOKE.COMPUTATION               713
    PAGE 234  ROUTINE SMOKE.EFFECTS                   775  784
    PAGE 235                                          811
    PAGE 260  ROUTINE BTRY.EFFECTS                   2011
    PAGE 262                                         2113 2124 2130 2131 2133 2134 2.
    PAGE 263                                         2154 2155 2155
    PAGE 483  PROCESS FIRE.MISSION                   2219
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7965 7966
    PAGE 694  ''PROGRAM  OLDER.VERSION               1440
FM.TM.CLASS
    PAGE  35  ''SECTION FOR PROCESSES                2003
    PAGE  47  ''SECTION FOR DEFINITIONS              2667
    PAGE 181  ROUTINE FASCAM.COMPUTATION             8265
    PAGE 188  ROUTINE FINAL.COVERAGE                 8564
    PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION          9097 9098
    PAGE 198                                         9105
```

```
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9156
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9326 9331 9333 9352
    PAGE 210  ROUTINE PGM.MSN.ASGN                   9606 9608 9612
    PAGE 211                                         9665
    PAGE 233  ROUTINE SMOKE.COMPUTATION               714
    PAGE 260  ROUTINE BTRY.EFFECTS                   2007 2011
    PAGE 262                                         2123
    PAGE 263                                         2153 2164 2181
    PAGE 265                                         2256
    PAGE 266                                         2312
    PAGE 271                                         2650
    PAGE 272                                         2661 2675
    PAGE 476  PROCESS TARGET.REPORT                  1882
    PAGE 483  PROCESS FIRE.MISSION                   2218
    PAGE 484                                         2264 2294 2299 2304 2305 2306 2307
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7965 7966
    PAGE 637  FUNCTION HE.WLA                        8919
    PAGE 694  ''PROGRAM  OLDER.VERSION               1441
    PAGE 706                                         2091
FM.TOF.TIME
    PAGE  35  ''SECTION FOR PROCESSES                2007
    PAGE 210  ROUTINE PGM.MSN.ASGN                   9649
    PAGE 485  PROCESS FIRE.MISSION                   2349 2366
    PAGE 694  ''PROGRAM  OLDER.VERSION               1445
FOLLOW.EFFECTS
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9312
    PAGE 204                                         9359 9360
FOLLOW.MINUS
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9312
    PAGE 204                                         9369 9370
FORCE
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           48
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES        931
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES       1405
    PAGE  58  ROUTINE CREATE.FORCE                   3105 3106 3107 3108 3109 3110 3115 3116 3121 3130 3136 3137
    PAGE  59                                         3154
    PAGE  71  ROUTINE ORIENTATION                    3665
    PAGE  77  ROUTINE ADJUST                         3889
    PAGE  93  ROUTINE INITIAL.DETECT                 4597 4600
    PAGE  94  ROUTINE INITIAL.MOVE                   4622 4625
    PAGE 101  ROUTINE LOS.CHECK                      4877 4878 4884
    PAGE 104  ROUTINE MINE.DELAY                     5028 5040
    PAGE 112  ROUTINE PREPARE.LIST                   5373
    PAGE 118  ROUTINE REIN.ARRIVE                    5605
    PAGE 133  ROUTINE CHECK.FORCE                    6159 6160 6165 6169 6173 6177 6178 6182 6184 6187 6206
    PAGE 134  ROUTINE CHECK.FOR.MINES                6223
    PAGE 137  ROUTINE CHECK.LIST                     6361
    PAGE 142  ROUTINE DEAD.UNIT                      6527 6528 6556 6558
    PAGE 143                                         6587
    PAGE 146  ROUTINE INTER.BATTLE                   6796
    PAGE 147                                         6797 6804
    PAGE 219  ROUTINE REQUEST.ILLUM                    40
    PAGE 223  ROUTINE REQUEST.SMOKE                   270  272
    PAGE 224                                          300  332
    PAGE 235  ROUTINE SMOKE.EFFECTS                   832
    PAGE 238  ROUTINE SWITCH.FO                       923  925  927
```

```
    PAGE 283  ROUTINE CAS.EVAL                          3219
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS             3409 3411 3412
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                  3931 3933 3934 3941 3946 3954
    PAGE 305  ROUTINE UNIT.PRIORITY                     4242 4243 4244 4248 4249 4257 4258 4258 4258 4263 4263 4265 4267 4270 4275
                                                        4276 4283
    PAGE 329  ROUTINE EMPTY                             5233 5235 5280
    PAGE 330                                            5282 5283
    PAGE 345  ROUTINE TERM.CHECK                        5772 5773 5778 5782 5793
    PAGE 347  EVENT ACT.REINF                           5812
    PAGE 396  EVENT START.BATTLE                        7819 7830
    PAGE 398                                            7944 7946
    PAGE 404  EVENT START.MOVE                          8285
    PAGE 410  EVENT UPDATE.LOC                          8540
    PAGE 412  EVENT ACT.ATK                             8626
    PAGE 439  PROCESS FORWARD.OBSERVER                  9932 9933 9935
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                   101
    PAGE 444                                             165
    PAGE 445                                             216  217
    PAGE 446                                             298  298
    PAGE 447                                             326
    PAGE 448                                             387  405  406  412
    PAGE 508  PROCESS CAS.MISSION                       3607 3608 3609
    PAGE 539  ROUTINE READ.ORDERS                       4987
    PAGE 619  ROUTINE SNAP.R                            8260 8262 8263
    PAGE 634  FUNCTION FEBA.BAND                        8791
    PAGE 676  ''PROGRAM  OLDER.VERSION                   373
    PAGE 684                                             844
FORCES.
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                   101
FORCE.
    PAGE  71  ROUTINE ORIENTATION                       3668
    PAGE 112  ROUTINE PREPARE.LIST                      5356
    PAGE 137  ROUTINE CHECK.LIST                        6338
    PAGE 141  ROUTINE CHECK.STREN                       6507 6508
FORWARD.DIST
    PAGE 428  PROCESS AIR.OBSERVER                      9348
    PAGE 430                                            9491 9493
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE              1370
    PAGE 468                                            1421 1423
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                   9485 9531
    PAGE 659                                            9533
FORWARD.OBSE
    PAGE 438  PROCESS FORWARD.OBSERVER                  9909
    PAGE 439                                            9954
    PAGE 440                                            9976 9990   18
    PAGE 441                                              36   79
FORWARD.OBSERVER
    PAGE  36  ''SECTION FOR PROCESSES                   2033
    PAGE 160  ROUTINE ATTRIT.SENSOR                     7405 7411
    PAGE 238  ROUTINE SWITCH.FO                          937  947  951  953
    PAGE 438  PROCESS FORWARD.OBSERVER                  9854
    PAGE 482  PROCESS FIRE.MISSION                      2177
    PAGE 485                                            2367 2372
    PAGE 566  ROUTINE SENSOR.INPUT                      6104
    PAGE 599  ROUTINE BETWEEN.ROUTINE                   7476 7477
    PAGE 616  ROUTINE SNAP2                             8148
```

```
    PAGE 619  ROUTINE SNAP.R                          8287
    PAGE 695  ''PROGRAM  OLDER.VERSION                1471
FOUND.
    PAGE 420  PROCESS AC.ATK.TGT                      8960
    PAGE 550  ROUTINE BTRY.INPUT                      5419
    PAGE 552                                          5525
FO.CAND.DET.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES        1427
    PAGE  36  ''SECTION FOR PROCESSES                 2040
    PAGE  49  ''SECTION FOR DEFINITIONS               2810
    PAGE 160  ROUTINE ATTRIT.SENSOR                   7394 7396
    PAGE 238  ROUTINE SWITCH.FO                        939  941  955  957
    PAGE 439  PROCESS FORWARD.OBSERVER                9924 9942 9957 9959
    PAGE 619  ROUTINE SNAP.R                          8289
    PAGE 684  ''PROGRAM  OLDER.VERSION                 866
    PAGE 695                                          1478
    PAGE 708                                          2231
FO.CIR.ERROR
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES         468
    PAGE 257  ROUTINE FO.DETECTION                    1867
    PAGE 538  ROUTINE MFO.INPUT                       4959
    PAGE 668  ''PROGRAM  OLDER.VERSION                9909
FO.CURRENT.TR
    PAGE   2  PROGRAM REVISIONS                         81
    PAGE   3                                           176
    PAGE  36  ''SECTION FOR PROCESSES                 2036
    PAGE 160  ROUTINE ATTRIT.SENSOR                   7379 7380 7385 7386 7409 7410
    PAGE 238  ROUTINE SWITCH.FO                        934  936  945
    PAGE 440  PROCESS FORWARD.OBSERVER                   2
    PAGE 441                                            65
    PAGE 695  ''PROGRAM  OLDER.VERSION                1474
FO.CUR.FM.LIST
    PAGE  35  ''SECTION FOR PROCESSES                 2015
    PAGE  36                                          2042
    PAGE  49  ''SECTION FOR DEFINITIONS               2809
    PAGE 160  ROUTINE ATTRIT.SENSOR                   7399 7401
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS          2774 2775 2776
    PAGE 482  PROCESS FIRE.MISSION                    2179 2186 2200
    PAGE 485                                          2355 2361 2364
    PAGE 695  ''PROGRAM  OLDER.VERSION                1453 1480
    PAGE 708                                          2230
FO.DC
    PAGE 160  ROUTINE ATTRIT.SENSOR                   7394 7396 7397
FO.DC.RANGE
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES        1425
    PAGE  49  ''SECTION FOR DEFINITIONS               2810
    PAGE 439  PROCESS FORWARD.OBSERVER                9922 9940
    PAGE 684  ''PROGRAM  OLDER.VERSION                 864
    PAGE 708                                          2231
FO.DC.UNIT
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES        1424
    PAGE 254  ROUTINE FO.DETECTION                    1688
    PAGE 439  PROCESS FORWARD.OBSERVER                9920 9938
    PAGE 440                                          9972 9974 9980 9982
    PAGE 684  ''PROGRAM  OLDER.VERSION                 863
```

```
FO.DET
    PAGE 254  ROUTINE FO.DETECTION                1697
    PAGE 440  PROCESS FORWARD.OBSERVER              18
FO.DETECT
    PAGE 254  ROUTINE FO.DETECTION                1713
    PAGE 256                                      1830
    PAGE 257                                      1871
FO.DETECTION
    PAGE   2  PROGRAM REVISIONS                     98
    PAGE   3                                       165
    PAGE 254  ROUTINE FO.DETECTION                1671 1693
    PAGE 255                                      1747
    PAGE 258                                      1904 1928
    PAGE 440  PROCESS FORWARD.OBSERVER               3
    PAGE 441                                        51
FO.DET.CANDIDATE
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES    1423
    PAGE 160  ROUTINE ATTRIT.SENSOR               7397
    PAGE 238  ROUTINE SWITCH.FO                    942  958
    PAGE 439  PROCESS FORWARD.OBSERVER            9919 9937
    PAGE 440                                      9995
    PAGE 441                                        41   67
    PAGE 619  ROUTINE SNAP.R                      8288
    PAGE 684  ''PROGRAM  OLDER.VERSION             862
FO.DISPLACEMENT.DIR
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA          4779 4804 4805 4806 4808 4809
FO.EX.FWD.OBSERVER
    PAGE  36  ''SECTION FOR PROCESSES             2038
    PAGE 438  PROCESS FORWARD.OBSERVER            9859 9860 9861 9862 9863 9864 9865 9866 9867 9868 9869 9870 9871 9872 9877
    PAGE 440                                      9997
    PAGE 441                                        43   84
    PAGE 442                                        89
    PAGE 695  ''PROGRAM  OLDER.VERSION            1476
FO.FST.INDIC
    PAGE  36  ''SECTION FOR PROCESSES             2037
    PAGE 257  ROUTINE FO.DETECTION                1861
    PAGE 566  ROUTINE SENSOR.INPUT                6106 6107
    PAGE 695  ''PROGRAM  OLDER.VERSION            1475
FO.RANGE
    PAGE 260  ROUTINE BTRY.EFFECTS                1979
    PAGE 267                                      2408 2416 2423
    PAGE 268                                      2437 2439
FO.RANGE.BAND
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES     465
    PAGE  12                                       657
    PAGE 538  ROUTINE MFO.INPUT                   4942
    PAGE 618  ROUTINE SNAP.R                      8222
    PAGE 668  ''PROGRAM  OLDER.VERSION            9906
    PAGE 671                                        99
    PAGE 712                                      2439
FO.RB.RANGE
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES     466
    PAGE  49  ''SECTION FOR DEFINITIONS           2811
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA          4797
    PAGE 254  ROUTINE FO.DETECTION                1710
    PAGE 255                                      1742
```

```
    PAGE 538  ROUTINE MFO.INPUT                    4956 4957 4957
    PAGE 668  ''PROGRAM  OLDER.VERSION             9907
    PAGE 708                                       2232
FO.REL.DIRECTION
    PAGE  36  ''SECTION FOR PROCESSES              2035
    PAGE  48  ''SECTION FOR DEFINITIONS            2737
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA           4804
    PAGE 566  ROUTINE SENSOR.INPUT                 6110
    PAGE 695  ''PROGRAM  OLDER.VERSION             1473
    PAGE 707                                       2159
FO.SEARCH.TIME
    PAGE 438  PROCESS FORWARD.OBSERVER             9861
    PAGE 439                                       9946
    PAGE 440                                       9984
FO.TGT.RPT.LIST
    PAGE  36  ''SECTION FOR PROCESSES              2041
    PAGE  38                                       2177
    PAGE  49  ''SECTION FOR DEFINITIONS            2812
    PAGE 160  ROUTINE ATTRIT.SENSOR                7388 7390
    PAGE 441  PROCESS FORWARD.OBSERVER               49
    PAGE 470  PROCESS TARGET.REPORT                1515 1516 1532 1533
    PAGE 471                                       1543 1544 1578 1580
    PAGE 472                                       1600 1602 1624 1626 1655
    PAGE 473                                       1657
    PAGE 474                                       1725 1727
    PAGE 475                                       1792 1793
    PAGE 476                                       1840 1841
    PAGE 477                                       1903 1904
    PAGE 695  ''PROGRAM  OLDER.VERSION             1479
    PAGE 697                                       1615
    PAGE 708                                       2233
FO.UNIT
    PAGE 438  PROCESS FORWARD.OBSERVER             9862 9880 9881 9889
    PAGE 439                                       9923 9941 9962
FO.US.LINK
    PAGE  36  ''SECTION FOR PROCESSES              2034
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA           4782
    PAGE 103  ROUTINE MINE.DELAY                   4972 4974
    PAGE 254  ROUTINE FO.DETECTION                 1686 1718
    PAGE 438  PROCESS FORWARD.OBSERVER             9879
    PAGE 441                                         53
    PAGE 482  PROCESS FIRE.MISSION                 2165
    PAGE 566  ROUTINE SENSOR.INPUT                 6113
    PAGE 695  ''PROGRAM  OLDER.VERSION             1472
FO.VISIBILITY
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES      467
    PAGE 255  ROUTINE FO.DETECTION                 1752
    PAGE 538  ROUTINE MFO.INPUT                    4958
    PAGE 668  ''PROGRAM  OLDER.VERSION             9908
FO.X
    PAGE 438  PROCESS FORWARD.OBSERVER             9910
    PAGE 440                                         19
FO.X.CORRECT
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA           4775 4808 4810
    PAGE 254  ROUTINE FO.DETECTION                 1676 1701
    PAGE 438  PROCESS FORWARD.OBSERVER             9863 9899 9907
```

```
    PAGE 440                                          8   16
FO.Y
    PAGE 438  PROCESS FORWARD.OBSERVER               9910
    PAGE 440                                           19
FO.Y.CORRECT
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA             4775 4809 4819
    PAGE 254  ROUTINE FO.DETECTION                   1677 1702
    PAGE 438  PROCESS FORWARD.OBSERVER               9864 9900 9908
    PAGE 440                                           9   17
FPO.XMAX
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES       1435
    PAGE  65  ROUTINE FILE.KAD.SENSOR                3384 3392 3393 3393 3394 3395 3395 3406 3406 3406
    PAGE 315  ROUTINE FLIGHT.PATH                    4689 4697 4700 4716
    PAGE 685  ''PROGRAM  OLDER.VERSION                874
FPO.XMIN
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES       1434
    PAGE  65  ROUTINE FILE.KAD.SENSOR                3383 3392 3392 3393 3394 3394 3395 3405 3405 3405
    PAGE 315  ROUTINE FLIGHT.PATH                    4688 4695 4702 4716
    PAGE 685  ''PROGRAM  OLDER.VERSION                873
FPO.YMAX
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES       1437
    PAGE  65  ROUTINE FILE.KAD.SENSOR                3386 3396 3397 3397 3398 3399 3399 3408 3408 3408
    PAGE 315  ROUTINE FLIGHT.PATH                    4690 4691 4725 4732 4735 4738
    PAGE 685  ''PROGRAM  OLDER.VERSION                876
FPO.YMIN
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES       1436
    PAGE  65  ROUTINE FILE.KAD.SENSOR                3385 3396 3396 3397 3398 3398 3399 3407 3407 3407
    PAGE 315  ROUTINE FLIGHT.PATH                    4690 4691 4724 4731 4733 4740
    PAGE 685  ''PROGRAM  OLDER.VERSION                875
FP.BATTLE
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES       1331
    PAGE 129  ROUTINE BTL.CHECK                      6046
    PAGE 297  ROUTINE FARRP.CHECK                    3871
    PAGE 337  ROUTINE HC.EMPTY                       5469
    PAGE 375  EVENT HC.DEPART.BATTLE                 7007 7009
    PAGE 401  EVENT START.BATTLE                     8108 8132
    PAGE 402                                         8155
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                112
    PAGE 450  PROCESS HC.RETURN.FARRP                 474
    PAGE 452                                          611  616  617  621  622  626  628
    PAGE 453                                          649
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          713  714  717
    PAGE 455                                          788
    PAGE 457                                          891
    PAGE 459                                         1008
    PAGE 683  ''PROGRAM  OLDER.VERSION                770
FP.NO.TEAMS
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES       1332
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS             3540 3573
    PAGE 291                                         3618
    PAGE 304  ROUTINE REPLACE.HC                     4223
    PAGE 683  ''PROGRAM  OLDER.VERSION                771
FP.OBSTACLE
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES       1433
    PAGE  65  ROUTINE FILE.KAD.SENSOR                3382 3410
    PAGE 314  ROUTINE FLIGHT.PATH                    4686
```

```
    PAGE 684  ''PROGRAM  OLDER.VERSION                872
FP.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        753
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES       1344
    PAGE  49  ''SECTION FOR DEFINITIONS              2814
    PAGE 297  ROUTINE FARRP.CHECK                    3871
    PAGE 494  PROCESS SHOOT.OUT                      2838
    PAGE 497                                         3014
    PAGE 573  ROUTINE FARRP.INPUT                    6430
    PAGE 673  ''PROGRAM  OLDER.VERSION                195
    PAGE 683                                          783
    PAGE 708                                         2235
FP.UNIT
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES       1330
    PAGE 128  ROUTINE BTL.CHECK                      5996
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS             3495
    PAGE 297  ROUTINE FARRP.CHECK                    3873
    PAGE 298  ROUTINE HC.COMPUTE.TIMES               3932 3951 3952 3959
    PAGE 299                                         3973
    PAGE 300  ROUTINE HC.DISENGAGE                   4052
    PAGE 301                                         4092
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4160 4171
    PAGE 304  ROUTINE REPLACE.HC                     4200
    PAGE 312  ROUTINE INTER.HELO                     4592
    PAGE 338  ROUTINE HC.EMPTY                       5567
    PAGE 375  EVENT HC.DEPART.BATTLE                 7006
    PAGE 376  EVENT HELO.ENGAGEMENT                  7048
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                113  115
    PAGE 445                                          265
    PAGE 450  PROCESS HC.RETURN.FARRP                 475
    PAGE 452                                          592
    PAGE 453                                          656
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          727
    PAGE 457                                          899
    PAGE 461                                         1114 1135
    PAGE 494  PROCESS SHOOT.OUT                      2839
    PAGE 496                                         2924
    PAGE 497                                         3015
    PAGE 510  PROCESS HELICOPTER.FIRE                3678
    PAGE 515                                         3978 4003
    PAGE 517                                         4082
    PAGE 518                                         4141
    PAGE 573  ROUTINE FARRP.INPUT                    6427 6430 6433 6434 6435
    PAGE 683  ''PROGRAM  OLDER.VERSION                769
FP.X.COORD
    PAGE 298  ROUTINE HC.COMPUTE.TIMES               3951 3955
    PAGE 299                                         3989 3996
FP.Y.COORD
    PAGE 298  ROUTINE HC.COMPUTE.TIMES               3952 3956
    PAGE 299                                         3990 3997
FRACTION.
    PAGE 281  ROUTINE AC.DF.EFFECTS                  3099
    PAGE 308  ROUTINE AD.SHOOT                       4393
FRACTION.COVERED
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS               1376 1388 1405
    PAGE 249                                         1439 1447
```

```
FRACT.COVERAGE
    PAGE   3  PROGRAM REVISIONS                    173
    PAGE 176  ROUTINE EST.COVERAGE                8011
    PAGE 179                                      8184 8197 8204 8204 8210 8211 8214
FRACT.COVERED
    PAGE 212  ROUTINE PIR.DETECTION               9703 9707 9744
    PAGE 229  ROUTINE RPV.DETECTION                514  519  557
    PAGE 343  ROUTINE SEARCH.COVERAGE             5683 5730
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE        1427 1428 1448
    PAGE 658  PROCESS PHOTO.IR.FLIGHT             9485
    PAGE 659                                      9537 9538 9541
FRACT.MAX.RANGE
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA          4780 4799 4801
FRACT.POST
    PAGE 636  FUNCTION HE.WLA                     8869
    PAGE 637                                      8944 8947 8950
FRAC.CAS
    PAGE 159  ROUTINE ATTRIT.SENSOR               7314 7319 7327 7329 7332
    PAGE 260  ROUTINE BTRY.EFFECTS                1983
    PAGE 264                                      2228 2233 2250 2253 2253
    PAGE 265                                      2282 2282 2293 2293 2303 2306 2306
    PAGE 267                                      2382 2386 2392 2392
    PAGE 269                                      2486 2535 2540 2541
    PAGE 271                                      2609 2632 2641 2644 2649
    PAGE 272                                      2660
FRAC.COV
    PAGE 179  ROUTINE EST.COVERAGE                8215
FRAC.COVERAGE
    PAGE   2  PROGRAM REVISIONS                    105
    PAGE 188  ROUTINE FINAL.COVERAGE              8550
    PAGE 191                                      8731 8763
    PAGE 192                                      8801
    PAGE 193                                      8852 8852 8853 8865 8866 8869
FRAC.F
    PAGE 268  ROUTINE BTRY.EFFECTS                2433
FRAC.WPN.RNG
    PAGE 447  PROCESS HC.ARRIVE.BATTLE             365  367  371  379  382
FRONT.DEPTH
    PAGE  41  ''SECTION FOR DEFINITIONS           2363
    PAGE  61  ROUTINE FEBA.INITIAL                3203 3204
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA          4815 4816
    PAGE 119  ROUTINE RESET.FEBA.SECTOR           5664
    PAGE 331  ROUTINE ENQ.FEBA.SET                5328
    PAGE 523  ROUTINE SYS.INPUT                   4356 4366 4366
    PAGE 700  ''PROGRAM  OLDER.VERSION            1793
FR.CAS.INDIC
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES    1410
    PAGE  58  ROUTINE CREATE.FORCE                3109 3115 3121 3130
    PAGE 283  ROUTINE CAS.EVAL                    3219 3221
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS       3412
    PAGE 508  PROCESS CAS.MISSION                 3609
    PAGE 684  ''PROGRAM  OLDER.VERSION             849
FR.COLOR
    PAGE 305  ROUTINE UNIT.PRIORITY               4236 4249 4264 4276 4290
    PAGE 306                                      4292
```

```
FR.CRIT.NO
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1407
    PAGE  46  ''SECTION FOR DEFINITIONS                 2620
    PAGE  59  ROUTINE CREATE.FORCE                      3154
    PAGE 133  ROUTINE CHECK.FORCE                       6177
    PAGE 145  ROUTINE FIN.BATTLE                        6722
    PAGE 684  ''PROGRAM  OLDER.VERSION                   846
    PAGE 705                                            2043
FR.MISSION
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1408
    PAGE  58  ROUTINE CREATE.FORCE                      3108
    PAGE 145  ROUTINE FIN.BATTLE                        6723
    PAGE 227  ROUTINE REQUEST.WD.FASCAM                  494  496
    PAGE 283  ROUTINE CAS.EVAL                          3184
    PAGE 284                                            3226 3229
    PAGE 305  ROUTINE UNIT.PRIORITY                     4257 4258 4260 4263
    PAGE 684  ''PROGRAM  OLDER.VERSION                   847
FR.SIDE
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1409
    PAGE  58  ROUTINE CREATE.FORCE                      3106
    PAGE 101  ROUTINE LOS.CHECK                         4878
    PAGE 133  ROUTINE CHECK.FORCE                       6160 6173 6182 6201
    PAGE 145  ROUTINE FIN.BATTLE                        6723 6732
    PAGE 147  ROUTINE INTER.BATTLE                      6797
    PAGE 173  ROUTINE DUST.EFFECTS                      7924
    PAGE 200  ROUTINE ILLUM.EFFECTS                     9224
    PAGE 216  ROUTINE REQUEST.FASCAM                    9933
    PAGE 218  ROUTINE REQUEST.ILLUM                        0
    PAGE 219                                              63
    PAGE 222  ROUTINE REQUEST.SMOKE                      189
    PAGE 223                                             238
    PAGE 226  ROUTINE REQUEST.WD.FASCAM                  407
    PAGE 227                                             494
    PAGE 235  ROUTINE SMOKE.EFFECTS                      814
    PAGE 283  ROUTINE CAS.EVAL                          3184 3207 3223
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS             3411
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                  3934
    PAGE 305  ROUTINE UNIT.PRIORITY                     4249 4258 4258 4260 4263 4276
    PAGE 345  ROUTINE TERM.CHECK                        5773
    PAGE 439  PROCESS FORWARD.OBSERVER                  9933
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                   170  201
    PAGE 448                                             406
    PAGE 478  PROCESS WITH.DRAW                         1954
    PAGE 489  PROCESS ASSESSMENT                        2535
    PAGE 508  PROCESS CAS.MISSION                       3608
    PAGE 684  ''PROGRAM  OLDER.VERSION                   848
FR.UNIT.SET
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          1009
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1412
    PAGE  58  ROUTINE CREATE.FORCE                      3136
    PAGE  71  ROUTINE ORIENTATION                       3668
    PAGE  93  ROUTINE INITIAL.DETECT                    4600
    PAGE  94  ROUTINE INITIAL.MOVE                      4625
    PAGE 101  ROUTINE LOS.CHECK                         4884
    PAGE 133  ROUTINE CHECK.FORCE                       6169 6187
    PAGE 145  ROUTINE FIN.BATTLE                        6709
```

```
    PAGE 147  ROUTINE INTER.BATTLE                    6804
    PAGE 157  ROUTINE AO.DETECTION                    7299
    PAGE 173  ROUTINE DUST.EFFECTS                    7897 7904 7938
    PAGE 174                                          7968
    PAGE 193  ROUTINE FINAL.COVERAGE                  8874
    PAGE 200  ROUTINE ILLUM.EFFECTS                   9195
    PAGE 201                                          9271 9274
    PAGE 216  ROUTINE REQUEST.FASCAM                  9938
    PAGE 218  ROUTINE REQUEST.ILLUM                     22
    PAGE 219                                            42   68
    PAGE 222  ROUTINE REQUEST.SMOKE                   210
    PAGE 223                                          241  279
    PAGE 226  ROUTINE REQUEST.WD.FASCAM               426
    PAGE 227                                          481
    PAGE 234  ROUTINE SMOKE.EFFECTS                   750  792
    PAGE 235                                          837  840
    PAGE 236                                          869  892
    PAGE 238  ROUTINE SWITCH.FO                       927
    PAGE 251  ROUTINE MINE.EFFECTS                    1529
    PAGE 257  ROUTINE FO.DETECTION                    1849 1895
    PAGE 261  ROUTINE BTRY.EFFECTS                    2063
    PAGE 262                                          2086
    PAGE 283  ROUTINE CAS.EVAL                        3189
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                3941
    PAGE 305  ROUTINE UNIT.PRIORITY                   4270 4283
    PAGE 329  ROUTINE EMPTY                           5235 5280
    PAGE 345  ROUTINE TERM.CHECK                      5762
    PAGE 383  EVENT OFF.LINE.ATTRITION                7372
    PAGE 384                                          7402
    PAGE 398  EVENT START.BATTLE                      7946
    PAGE 400                                          8053
    PAGE 438  PROCESS FORWARD.OBSERVER                8889
    PAGE 439                                          9935 9962
    PAGE 440                                          9982
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                  168  203
    PAGE 448                                            412
    PAGE 478  PROCESS WITH.DRAW                       1959
    PAGE 489  PROCESS ASSESSMENT                      2540 2563
    PAGE 619  ROUTINE SNAP.R                          8263
    PAGE 677  ''PROGRAM  OLDER.VERSION                  451
    PAGE 684                                            851
FS.BATS
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES        1360
    PAGE  63  ROUTINE FILE.FD.SCHD                    3280 3285 3291 3296 3304 3308 3314
    PAGE  64                                          3330 3337 3338 3344 3348
    PAGE 194  ROUTINE FIND.START.TIME                 8947
    PAGE 195                                          8950 8975 8978
    PAGE 683  ''PROGRAM  OLDER.VERSION                  799
FS.START
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES        1358
    PAGE  49  ''SECTION FOR DEFINITIONS               2807
    PAGE  63  ROUTINE FILE.FD.SCHD                    3278 3283 3294 3302 3312
    PAGE  64                                          3328 3333 3334 3343 3346
    PAGE 194  ROUTINE FIND.START.TIME                 8905 8916 8920 8922 8924 8925 8933 8935 8940 8944
    PAGE 195                                          8951 8964 8969
    PAGE 683  ''PROGRAM  OLDER.VERSION                  797
```

```
    PAGE 708                                        2228
FS.STOP
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES      1359
    PAGE  63  ROUTINE FILE.FD.SCHD                  3270 3275 3279 3284 3292 3295 3303 3313
    PAGE  64                                        3329 3335 3336 3347
    PAGE 194  ROUTINE FIND.START.TIME               8911 8919 8922 8928 8943 8944
    PAGE 195                                        8960 8968 8970
    PAGE 683  ''PROGRAM  OLDER.VERSION               798
FT.AC.ATK.TGT
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES      1369
    PAGE 329  ROUTINE EMPTY                         5250
    PAGE 425  PROCESS AC.ATK.TGT                    9244
    PAGE 426                                        9295
    PAGE 490  PROCESS ASSESSMENT                    2606 2634
    PAGE 491                                        2648
    PAGE 494  PROCESS SHOOT.OUT                     2822 2823 2833 2834 2859
    PAGE 495                                        2886 2887
    PAGE 497                                        3035
    PAGE 498                                        3074 3075
    PAGE 500                                        3211
    PAGE 507  PROCESS CAS.MISSION                   3576
    PAGE 683  ''PROGRAM  OLDER.VERSION               808
FT.CAS.MISSION
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES      1370
    PAGE 293  ROUTINE END.CAS.MISSION               3720 3730
    PAGE 507  PROCESS CAS.MISSION                   3577
    PAGE 683  ''PROGRAM  OLDER.VERSION               809
FT.ENTRY
    PAGE 493  PROCESS SHOOT.OUT                     2771
    PAGE 501                                        3240 3241
    PAGE 503                                        3335 3336
FT.FIRING.WPN
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES      1373
    PAGE 496  PROCESS SHOOT.OUT                     2943
    PAGE 498                                        3072
    PAGE 499                                        3102 3138
    PAGE 501                                        3215 3235
    PAGE 502                                        3282 3285 3288 3292
    PAGE 512  PROCESS HELICOPTER.FIRE               3791
    PAGE 513                                        3870
    PAGE 683  ''PROGRAM  OLDER.VERSION               812
FT.PK
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES      1374
    PAGE  48  ''SECTION FOR DEFINITIONS             2738
    PAGE 493  PROCESS SHOOT.OUT                     2770
    PAGE 494                                        2828
    PAGE 495                                        2919
    PAGE 496                                        2944
    PAGE 501                                        3268
    PAGE 502                                        3275
    PAGE 512  PROCESS HELICOPTER.FIRE               3792
    PAGE 515                                        3975 3989 4000
    PAGE 516                                        4018
    PAGE 517                                        4079 4097
    PAGE 518                                        4138 4156
    PAGE 683  ''PROGRAM  OLDER.VERSION               813
```

```
    PAGE 707                                        2160
FT.PK.BAR
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES      1375
    PAGE  48  ''SECTION FOR DEFINITIONS             2739
    PAGE 494  PROCESS SHOOT.OUT                     2829
    PAGE 496                                        2941 2945
    PAGE 512  PROCESS HELICOPTER.FIRE               3793
    PAGE 683  ''PROGRAM  OLDER.VERSION               814
    PAGE 707                                        2161
FT.SCORE1
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES      1376
    PAGE  46  ''SECTION FOR DEFINITIONS             2617
    PAGE 494  PROCESS SHOOT.OUT                     2830
    PAGE 496                                        2948 2978
    PAGE 512  PROCESS HELICOPTER.FIRE               3803
    PAGE 513                                        3848
    PAGE 684  ''PROGRAM  OLDER.VERSION               815
    PAGE 705                                        2040
FT.SCORE2
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES      1377
    PAGE  46  ''SECTION FOR DEFINITIONS             2618
    PAGE 494  PROCESS SHOOT.OUT                     2831
    PAGE 496                                        2949
    PAGE 497                                        2995
    PAGE 512  PROCESS HELICOPTER.FIRE               3804
    PAGE 513                                        3859
    PAGE 684  ''PROGRAM  OLDER.VERSION               816
    PAGE 705                                        2041
FT.TARGET.EQUIP
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES      1372
    PAGE  46  ''SECTION FOR DEFINITIONS             2616
    PAGE 369  EVENT ENGAGEMENT                      6723
    PAGE 377  EVENT HELO.ENGAGEMENT                 7094
    PAGE 378                                        7184 7198
    PAGE 461  PROCESS HEL.TARGET.ACQUISITION        1106
    PAGE 489  PROCESS ASSESSMENT                    2569
    PAGE 494  PROCESS SHOOT.OUT                     2826 2863
    PAGE 495                                        2891 2914 2921
    PAGE 497                                        3012
    PAGE 499                                        3128
    PAGE 500                                        3164
    PAGE 501                                        3263
    PAGE 502                                        3319
    PAGE 507  PROCESS CAS.MISSION                   3575
    PAGE 511  PROCESS HELICOPTER.FIRE               3729 3761
    PAGE 512                                        3775 3783 3796
    PAGE 513                                        3869
    PAGE 683  ''PROGRAM  OLDER.VERSION               811
    PAGE 705                                        2039
FT.TGT.UNIT
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES      1371
    PAGE  46  ''SECTION FOR DEFINITIONS             2615
    PAGE 108  ROUTINE NEW.SEGMENT                   5212
    PAGE 109                                        5229
    PAGE 300  ROUTINE HC.DISENGAGE                  4057
    PAGE 301                                        4091
```

```
    PAGE 312  ROUTINE INTER.HELO                    4599
    PAGE 369  EVENT ENGAGEMENT                      6722
    PAGE 377  EVENT HELO.ENGAGEMENT                 7093
    PAGE 378                                        7189
    PAGE 489  PROCESS ASSESSMENT                    2568
    PAGE 494  PROCESS SHOOT.OUT                     2814 2817 2827
    PAGE 495                                        2901 2917
    PAGE 496                                        2938
    PAGE 497                                        3013
    PAGE 498                                        3092
    PAGE 500                                        3165
    PAGE 501                                        3241 3266
    PAGE 502                                        3320
    PAGE 503                                        3336
    PAGE 507  PROCESS CAS.MISSION                   3574
    PAGE 511  PROCESS HELICOPTER.FIRE               3755 3764 3770
    PAGE 512                                        3778 3786
    PAGE 513                                        3868
    PAGE 683  ''PROGRAM  OLDER.VERSION               810
    PAGE 705                                        2038
FT.TIME
    PAGE 299  ROUTINE HC.COMPUTE.TIMES              3974 3976 3978
FULL.STRENGTH
    PAGE 130  ROUTINE CHECK.DEAD                    6082 6083 6086
FUZE
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           46
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES        476  479
    PAGE  10                                          519  522
    PAGE  51  ''SECTION FOR SUBSTITUTIONS           2926
    PAGE 188  ROUTINE FINAL.COVERAGE                8547 8553
    PAGE 190                                        8716 8718 8720
    PAGE 260  ROUTINE BTRY.EFFECTS                  1985
    PAGE 262                                        2126 2128 2130
    PAGE 263                                        2172 2173 2175 2178
    PAGE 265                                        2259 2275
    PAGE 266                                        2317 2345 2353 2361
    PAGE 272                                        2675 2694
    PAGE 555  ROUTINE MUNS.INPUT                    5614 5635 5636 5649 5651
    PAGE 558  ROUTINE HE.LA.INPUT                   5760 5780 5781 5783 5789
    PAGE 559                                        5811 5812 5814
    PAGE 618  ROUTINE SNAP.R                        8223
    PAGE 636  FUNCTION HE.WLA                       8868 8906 8908 8910
    PAGE 637                                        8934 8941
     AGE 668  ''PROGRAM  OLDER.VERSION              9917 9920
    PAGE 669                                        9960 9963
    PAGE 710                                        2333
    PAGE 712                                        2440
FZ.HE.RELY
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES       480
    PAGE 190  ROUTINE FINAL.COVERAGE                8716
    PAGE 262  ROUTINE BTRY.EFFECTS                  2130
    PAGE 263                                        2178
    PAGE 555  ROUTINE MUNS.INPUT                    5651
    PAGE 636  FUNCTION HE.WLA                       8910
    PAGE 668  ''PROGRAM  OLDER.VERSION              9921
```

```
FZ.NAME
     PAGE   9  ''SECTION FOR PERMANENT_ENTITIES          477
     PAGE  47  ''SECTION FOR DEFINITIONS                2668
     PAGE 555  ROUTINE MUNS.INPUT                       5636
     PAGE 668  ''PROGRAM  OLDER.VERSION                 9918
     PAGE 706                                           2092
F.AATT.LIST
     PAGE  34  ''SECTION FOR PROCESSES                  1902
     PAGE 693  ''PROGRAM  OLDER.VERSION                 1340
F.AA.SET
     PAGE  34  ''SECTION FOR PROCESSES                  1950
     PAGE 693  ''PROGRAM  OLDER.VERSION                 1388
F.AO.CAND.DET.LIST
     PAGE  34  ''SECTION FOR PROCESSES                  1935
     PAGE 693  ''PROGRAM  OLDER.VERSION                 1373
F.AO.DET.TGT.LIST
     PAGE  34  ''SECTION FOR PROCESSES                  1937
     PAGE 693  ''PROGRAM  OLDER.VERSION                 1375
F.AO.FLIGHT.LEG.LIST
     PAGE  34  ''SECTION FOR PROCESSES                  1933
     PAGE 693  ''PROGRAM  OLDER.VERSION                 1371
F.AR.CAND.DET.LIST
     PAGE  34  ''SECTION FOR PROCESSES                  1917
     PAGE 693  ''PROGRAM  OLDER.VERSION                 1355
F.AR.DET.TGT.LIST
     PAGE  34  ''SECTION FOR PROCESSES                  1915
     PAGE 693  ''PROGRAM  OLDER.VERSION                 1353
F.AU.LIST
     PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          577
     PAGE 670  ''PROGRAM  OLDER.VERSION                   18
F.AVAIL.AO.LIST
     PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          765
     PAGE 673  ''PROGRAM  OLDER.VERSION                  207
F.BN.BTRY.SET
     PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          434
     PAGE  89  ROUTINE FA.BN.MOVEMENT                   4410
     PAGE 667  ''PROGRAM  OLDER.VERSION                 9875
F.BTL.FORCE.SET
     PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES         1176
     PAGE 680  ''PROGRAM  OLDER.VERSION                  615
F.BY.FM.QUEUE
     PAGE   7  ''SECTION FOR PERMANENT_ENTITIES          343
     PAGE 163  ROUTINE BTRY.FM.DEQ                      7528
     PAGE 666  ''PROGRAM  OLDER.VERSION                 9785
F.BY.HOW.SET
     PAGE   7  ''SECTION FOR PERMANENT_ENTITIES          339
     PAGE 666  ''PROGRAM  OLDER.VERSION                 9781
F.BY.SCHD.LIST
     PAGE   7  ''SECTION FOR PERMANENT_ENTITIES          341
     PAGE 666  ''PROGRAM  OLDER.VERSION                 9783
F.CFPS.LIST
     PAGE  35  ''SECTION FOR PROCESSES                  1987
     PAGE 694  ''PROGRAM  OLDER.VERSION                 1425
F.CF.OP.Q
     PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES         1236
     PAGE 363  EVENT CFR.OPERATOR                       6585
```

```
    PAGE 681   ''PROGRAM  OLDER.VERSION                675
F.CT.TU.SET
    PAGE   7   ''SECTION FOR PERMANENT_ENTITIES         361
    PAGE 666   ''PROGRAM  OLDER.VERSION                9803
F.FD.BN.LIST
    PAGE   9   ''SECTION FOR PERMANENT_ENTITIES         458
    PAGE 668   ''PROGRAM  OLDER.VERSION                9899
F.FD.COMPLETE.LIST
    PAGE   9   ''SECTION FOR PERMANENT_ENTITIES         454
    PAGE 668   ''PROGRAM  OLDER.VERSION                9895
F.FD.SCHD.LIST
    PAGE   9   ''SECTION FOR PERMANENT_ENTITIES         456
    PAGE 194   ROUTINE FIND.START.TIME                 8905
    PAGE 668   ''PROGRAM  OLDER.VERSION                9897
F.FD.TR.QUEUE
    PAGE   9   ''SECTION FOR PERMANENT_ENTITIES         452
    PAGE 332   ROUTINE FDC.TR.DEQ                      5349
    PAGE 668   ''PROGRAM  OLDER.VERSION                9893
F.FO.CAND.DET.LIST
    PAGE  36   ''SECTION FOR PROCESSES                 2044
    PAGE 695   ''PROGRAM  OLDER.VERSION                1482
F.FO.CUR.FM.LIST
    PAGE  36   ''SECTION FOR PROCESSES                 2048
    PAGE 485   PROCESS FIRE.MISSION                    2370
    PAGE 486                                           2375 2376
    PAGE 695   ''PROGRAM  OLDER.VERSION                1486
F.FO.TGT.RPT.LIST
    PAGE  36   ''SECTION FOR PROCESSES                 2046
    PAGE 695   ''PROGRAM  OLDER.VERSION                1484
F.FP.SET
    PAGE  14   ''SECTION FOR PERMANENT_ENTITIES         758
    PAGE 673   ''PROGRAM  OLDER.VERSION                 200
F.FR.UNIT.SET
    PAGE  25   ''SECTION FOR TEMPORARY_ENTITIES        1416
    PAGE 684   ''PROGRAM  OLDER.VERSION                 855
F.GP.CAT.SET
    PAGE   9   ''SECTION FOR PERMANENT_ENTITIES         487
    PAGE 668   ''PROGRAM  OLDER.VERSION                9928
F.HE.TB.RH.LIST
    PAGE   9   ''SECTION FOR PERMANENT_ENTITIES         504
    PAGE 669   ''PROGRAM  OLDER.VERSION                9945
F.HF.SO.LIST
    PAGE  32   ''SECTION FOR TEMPORARY_ENTITIES        1833
    PAGE 691   ''PROGRAM  OLDER.VERSION                1272
F.HT.LIST
    PAGE  24   ''SECTION FOR TEMPORARY_ENTITIES        1340
    PAGE 683   ''PROGRAM  OLDER.VERSION                 779
F.HT.MEMBER.LIST
    PAGE  26   ''SECTION FOR TEMPORARY_ENTITIES        1452
    PAGE 685   ''PROGRAM  OLDER.VERSION                 891
F.HT.TARGET.LIST
    PAGE  26   ''SECTION FOR TEMPORARY_ENTITIES        1455
    PAGE 685   ''PROGRAM  OLDER.VERSION                 894
F.IC.TB.RH.LIST
    PAGE  10   ''SECTION FOR PERMANENT_ENTITIES         541
    PAGE 669   ''PROGRAM  OLDER.VERSION                9982
```

```
F.MADS.RH.SET
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          603
    PAGE 670  ''PROGRAM  OLDER.VERSION                   44
F.MA.SET
    PAGE  37  ''SECTION FOR PROCESSES                  2095
    PAGE 696  ''PROGRAM  OLDER.VERSION                 1533
F.MCFR.RH.LIST
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES          641
    PAGE 671  ''PROGRAM  OLDER.VERSION                   82
F.MFO.RB.SET
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES          653
    PAGE 671  ''PROGRAM  OLDER.VERSION                   94
F.MFP.LIST
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          574
    PAGE 670  ''PROGRAM  OLDER.VERSION                   15
F.MO.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES         1063
    PAGE 678  ''PROGRAM  OLDER.VERSION                  505
F.MPDB.RH.LIST
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES          667
    PAGE 671  ''PROGRAM  OLDER.VERSION                  110
F.MU.ORDER.SET
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES         1553
    PAGE 687  ''PROGRAM  OLDER.VERSION                  992
F.MU.TF.LIST
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES         1551
    PAGE 142  ROUTINE DEAD.UNIT                        6576
    PAGE 687  ''PROGRAM  OLDER.VERSION                  990
F.PATH.SET
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES         1801
    PAGE  72  ROUTINE ORIENTATION                      3701 3702 3711 3712
    PAGE 691  ''PROGRAM  OLDER.VERSION                 1240
F.PDB.KEYED.LIST
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES         1640
    PAGE 688  ''PROGRAM  OLDER.VERSION                 1079
F.PDB.OP.Q
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES         1643
    PAGE 388  EVENT PDB.OPERATOR                       7563
    PAGE 688  ''PROGRAM  OLDER.VERSION                 1082
F.PIR.FLIGHT.LEG.LIST
    PAGE  37  ''SECTION FOR PROCESSES                  2107
    PAGE 696  ''PROGRAM  OLDER.VERSION                 1545
F.PIR.RECORD.LIST
    PAGE  37  ''SECTION FOR PROCESSES                  2109
    PAGE 696  ''PROGRAM  OLDER.VERSION                 1547
F.PIR.RTD.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES         1682
    PAGE 689  ''PROGRAM  OLDER.VERSION                 1121
F.RPV.CAND.DET.LIST
    PAGE  37  ''SECTION FOR PROCESSES                  2123
    PAGE 696  ''PROGRAM  OLDER.VERSION                 1561
F.RPV.FLIGHT.LEG.LIST
    PAGE  37  ''SECTION FOR PROCESSES                  2126
    PAGE 696  ''PROGRAM  OLDER.VERSION                 1564
F.SIDE.CFR.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          763
```

```
    PAGE 673  ''PROGRAM  OLDER.VERSION                205
F.SIDE.PDB.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        761
    PAGE 673  ''PROGRAM  OLDER.VERSION                203
F.SI.LIST
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES       1204
    PAGE 681  ''PROGRAM  OLDER.VERSION                643
F.SO.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1827
    PAGE 691  ''PROGRAM  OLDER.VERSION               1266
F.SS.SET
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        809
    PAGE  89  ROUTINE FA.BN.MOVEMENT                 4451
    PAGE  91                                         4531
    PAGE 239  ROUTINE TARGET.ANALYSIS                1019
    PAGE 246  ROUTINE VOLLEY                         1339
    PAGE 331  ROUTINE ENQ.FEBA.SET                   5325
    PAGE 360  EVENT CFR.ON                           6447
    PAGE 371  EVENT FEBA.SORTIE                      6813
    PAGE 372                                         6854
    PAGE 634  FUNCTION FEBA.BAND                     8802 8820
    PAGE 660  FUNCTION STAY.TIME                     9606
    PAGE 674  ''PROGRAM  OLDER.VERSION                251
F.STRENGTH
    PAGE 130  ROUTINE CHECK.DEAD                     6100 6101
F.T
    PAGE 378  EVENT HELO.ENGAGEMENT                  7183 7184
F.TB.SORT.LIST
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        87?
    PAGE 675  ''PROGRAM  OLDER.VERSION                314
F.TB.TM.LIST
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        904
    PAGE 675  ''PROGRAM  OLDER.VERSION                346
F.TEAM.TYPES
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        875
    PAGE 675  ''PROGRAM  OLDER.VERSION                317
F.TR.DET.LIST
    PAGE  38  ''SECTION FOR PROCESSES                2181
    PAGE 697  ''PROGRAM  OLDER.VERSION               1619
F.TR.FM.LIST
    PAGE  38  ''SECTION FOR PROCESSES                2179
    PAGE 697  ''PROGRAM  OLDER.VERSION               1617
F.TU.NTE.SET
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES        948
    PAGE 676  ''PROGRAM  OLDER.VERSION                390
F.TU.TE.LIST
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES        946
    PAGE 676  ''PROGRAM  OLDER.VERSION                388
F.UE.TARGET.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1824
    PAGE 691  ''PROGRAM  OLDER.VERSION               1263
F.UE.WEAPON.SET
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1830
    PAGE 307  ROUTINE AD.SHOOT                       4338
    PAGE 352  EVENT AD.ENGAGEMENT                    6081
    PAGE 493  PROCESS SHOOT.OUT                      2778
```

```
    PAGE 691  ''PROGRAM  OLDER.VERSION                1269
F.UNIT.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES         774
    PAGE 673  ''PROGRAM  OLDER.VERSION                 216
F.UN.EQUIP.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES        1020
    PAGE 677  ''PROGRAM  OLDER.VERSION                 462
F.UN.HC.LOS.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES        1059
    PAGE 678  ''PROGRAM  OLDER.VERSION                 501
F.UN.LOS.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES        1028
    PAGE 678  ''PROGRAM  OLDER.VERSION                 470
F.UN.PATH
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES        1024
    PAGE 322  ROUTINE COMPUTE.WD                      4990
    PAGE 677  ''PROGRAM  OLDER.VERSION                 466
F.UN.SEGMENT.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES        1026
    PAGE 678  ''PROGRAM  OLDER.VERSION                 468
F.UN.SENSOR.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES        1030
    PAGE 678  ''PROGRAM  OLDER.VERSION                 472
F.UN.SUB.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES        1022
    PAGE 142  ROUTINE DEAD.UNIT                       6574
    PAGE 143                                          6625
    PAGE 677  ''PROGRAM  OLDER.VERSION                 464
GET.FLAG
    PAGE  40  ''SECTION FOR EVENTS                    2272
    PAGE 699  ''PROGRAM  OLDER.VERSION                1709
GET.NX.ORD
    PAGE  40  ''SECTION FOR EVENTS                    2268
    PAGE  88  ROUTINE END.MOVE                        4381
    PAGE 117  ROUTINE PROX.POS                        5588
    PAGE 118  ROUTINE REIN.ARRIVE                     5631
    PAGE 125  ROUTINE WHAT.NEXT                       5881
    PAGE 126                                          5961
    PAGE 146  ROUTINE INTER.BATTLE                    6769 6782
    PAGE 348  EVENT ACT.REINF                         5888
    PAGE 373  EVENT GET.NX.ORD                        6875
    PAGE 601  ROUTINE BETWEEN.ROUTINE                 7596 7597
    PAGE 616  ROUTINE SNAP2                           8179
    PAGE 699  ''PROGRAM  OLDER.VERSION                1705
GET.ORDER
    PAGE  40  ''SECTION FOR EVENTS                    2270
    PAGE 699  ''PROGRAM  OLDER.VERSION                1707
GET.OTHER
    PAGE  40  ''SECTION FOR EVENTS                    2271
    PAGE 699  ''PROGRAM  OLDER.VERSION                1708
GET.UNIT
    PAGE  40  ''SECTION FOR EVENTS                    2269
    PAGE 699  ''PROGRAM  OLDER.VERSION                1706
GOOD.WEATHER.PROB
    PAGE  42  ''SECTION FOR DEFINITIONS               2416
    PAGE 256  ROUTINE FO.DETECTION                    1798
```

```
    PAGE 575  ROUTINE PGM.INPUT                    6487
    PAGE 701  ''PROGRAM  OLDER.VERSION             1846
GP.CAT.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES      359
    PAGE   9                                        485
    PAGE 527  ROUTINE CAT.TU.INPUT                 4467
    PAGE 666  ''PROGRAM  OLDER.VERSION             9801
    PAGE 668                                       9926
GP.NAME
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES      483
    PAGE  47  ''SECTION FOR DEFINITIONS            2669
    PAGE  56  ROUTINE MAIN1                        3058 3059 3060 3061
    PAGE 668  ''PROGRAM  OLDER.VERSION             9924
    PAGE 706                                       2093
GRAND.PARENT
    PAGE 125  ROUTINE WHAT.NEXT                    5908 5909 5912 5914 5918 5919
GROUPING
    PAGE   1  ROUTINE FOR CROSS_REFERENCING          46
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES      482
    PAGE  14                                        771
    PAGE  56  ROUTINE MAIN1                        3057
    PAGE 243  ROUTINE UNIT.ENVIR                   1206 1212
    PAGE 244                                       1228 1261
    PAGE 270  ROUTINE BTRY.EFFECTS                 2583 2584
    PAGE 428  PROCESS AIR.OBSERVER                 9342
    PAGE 430                                       9477 9479
    PAGE 439  PROCESS FORWARD.OBSERVER             9913 9915
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE         1365 1407 1409
    PAGE 618  ROUTINE SNAP.R                       8224
    PAGE 653  PROCESS AIRBORNE.RADAR               9272 9313 9316
    PAGE 658  PROCESS PHOTO.IR.FLIGHT              9481 9515 9517
    PAGE 668  ''PROGRAM  OLDER.VERSION             9923
    PAGE 673                                        213
    PAGE 712                                       2441
G.
    PAGE 664  ''PROGRAM  OLDER.VERSION             9676 9683
HALF.WIDTH
    PAGE 343  ROUTINE SEARCH.COVERAGE              5679 5686 5686 5708 5710 5713 5719 5721 5725
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE         1379
    PAGE 468                                       1426
    PAGE 658  PROCESS PHOTO.IR.FLIGHT              9484 9503
    PAGE 659                                       9536
HCEN.ENEMY
    PAGE  40  ''SECTION FOR EVENTS                 2283
    PAGE 376  EVENT HELO.ENGAGEMENT                7060
    PAGE 699  ''PROGRAM  OLDER.VERSION             1720
HCEN.FARRP
    PAGE  40  ''SECTION FOR EVENTS                 2281
    PAGE 699  ''PROGRAM  OLDER.VERSION             1718
HCEN.HELICOPTER
    PAGE  40  ''SECTION FOR EVENTS                 2282
    PAGE 699  ''PROGRAM  OLDER.VERSION             1719
HCEN.TEAM
    PAGE  40  ''SECTION FOR EVENTS                 2280
    PAGE 301  ROUTINE HC.DISENGAGE                 4067
    PAGE 312  ROUTINE INTER.HELO                   4610
```

```
    PAGE 338  ROUTINE HC.EMPTY                    5555
    PAGE 376  EVENT HELO.ENGAGEMENT               7059
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION      1082
    PAGE 699  ''PROGRAM  OLDER.VERSION            1717
HC.ALT
    PAGE 510  PROCESS HELICOPTER.FIRE             3706
HC.ALTITUDE
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES    1474
    PAGE  52  ''SECTION FOR SUBSTITUTIONS         2975
    PAGE 377  EVENT HELO.ENGAGEMENT               7112 7121
    PAGE 378                                      7168 7181
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION       749
    PAGE 455                                       776  781  801
    PAGE 456                                       862
    PAGE 458                                       945
    PAGE 510  PROCESS HELICOPTER.FIRE             3703
    PAGE 511                                      3720
    PAGE 685  ''PROGRAM  OLDER.VERSION             913
    PAGE 711                                      2382
HC.ARRIVE
    PAGE 448  PROCESS HC.ARRIVE.BATTLE             437
    PAGE 449                                       455
HC.ARRIVE.BA
    PAGE 443  PROCESS HC.ARRIVE.BATTLE             120
    PAGE 445                                       231  260
    PAGE 446                                       294
HC.ARRIVE.BATTLE
    PAGE  36  ''SECTION FOR PROCESSES             2054
    PAGE 337  ROUTINE HC.EMPTY                    5500 5501 5503 5504 5505 5508 5509 5511 5512 5513
    PAGE 338                                      5515
    PAGE 391  EVENT SEND.TEAM                     7615
    PAGE 402  EVENT START.BATTLE                  8165
    PAGE 443  PROCESS HC.ARRIVE.BATTLE              93
    PAGE 447                                       380
    PAGE 448                                       398
    PAGE 599  ROUTINE BETWEEN.ROUTINE             7480 7481
    PAGE 616  ROUTINE SNAP2                       8149
    PAGE 695  ''PROGRAM  OLDER.VERSION            1492
HC.BTL.TEAM
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES    1470
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS          3545 3558 3582
    PAGE 291                                      3595 3638
    PAGE 292                                      3647
    PAGE 301  ROUTINE HC.DISENGAGE                4112
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE           4146 4148 4152
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION      1094
    PAGE 489  PROCESS ASSESSMENT                  2556
    PAGE 495  PROCESS SHOOT.OUT                   2866
    PAGE 496                                      2925
    PAGE 498                                      3040
    PAGE 685  ''PROGRAM  OLDER.VERSION             909
HC.COMPUTE.TIMES
    PAGE 128  ROUTINE BTL.CHECK                   6009 6028
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS          3527
    PAGE 297  ROUTINE FARRP.CHECK                 3890
    PAGE 298  ROUTINE HC.COMPUTE.TIMES            3913
```

```
    PAGE 299                                        3995
    PAGE 450  PROCESS HC.RETURN.FARRP                491
HC.DEBUG
    PAGE  47  ''SECTION FOR DEFINITIONS             2705
    PAGE 128  ROUTINE BTL.CHECK                     6000
    PAGE 129                                        6059
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS            3489 3517
    PAGE 290                                        3567
    PAGE 291                                        3610
    PAGE 292                                        3665
    PAGE 297  ROUTINE FARRP.CHECK                   3902
    PAGE 299  ROUTINE HC.COMPUTE.TIMES              3985
    PAGE 300  ROUTINE HC.DISENGAGE                  4019
    PAGE 303  ROUTINE HEL.RANGE.COMPUTE             4179
    PAGE 304  ROUTINE REPLACE.HC                    4205 4227
    PAGE 305  ROUTINE UNIT.PRIORITY                 4256 4289
    PAGE 311  ROUTINE INTER.HELO                    4558
    PAGE 337  ROUTINE HC.EMPTY                      5468 5479 5493 5514
    PAGE 338                                        5529
    PAGE 339                                        5573
    PAGE 356  EVENT BTL.ENDED                       6278 6290
    PAGE 375  EVENT HC.DEPART.BATTLE                7000
    PAGE 376  EVENT HELO.ENGAGEMENT                 7050
    PAGE 377                                        7096
    PAGE 378                                        7148 7201
    PAGE 379                                        7217
    PAGE 391  EVENT SEND.TEAM                       7609
    PAGE 401  EVENT START.BATTLE                    8101 8125
    PAGE 443  PROCESS HC.ARRIVE.BATTLE               117
    PAGE 445                                         225  257
    PAGE 446                                         289
    PAGE 447                                         376
    PAGE 448                                         395  434
    PAGE 449                                         453
    PAGE 450  PROCESS HC.RETURN.FARRP                503
    PAGE 451                                         519  554  571
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION         716
    PAGE 455                                         787
    PAGE 456                                         818
    PAGE 457                                         874
    PAGE 458                                         958  975
    PAGE 460                                        1050
    PAGE 462                                        1160
    PAGE 510  PROCESS HELICOPTER.FIRE               3701
    PAGE 511                                        3740 3758
    PAGE 512                                        3805 3827
    PAGE 513                                        3878
    PAGE 514                                        3907
    PAGE 515                                        3954
    PAGE 516                                        4063
    PAGE 518                                        4170
    PAGE 572  ROUTINE FARRP.INPUT                   6354
    PAGE 706  ''PROGRAM  OLDER.VERSION              2129
HC.DEPART
    PAGE 375  EVENT HC.DEPART.BATTLE                7002
```

```
HC.DEPART.BATTLE
    PAGE  40  ''SECTION FOR EVENTS                  2274
    PAGE 338  ROUTINE HC.EMPTY                      5568 5569 5571 5572
    PAGE 339                                        5574 5575 5581 5582 5584 5588 5590 5591 5594 5595
    PAGE 375  EVENT HC.DEPART.BATTLE                6988
    PAGE 461  PROCESS HEL.TARGET.ACQUISITION        1154
    PAGE 601  ROUTINE BETWEEN.ROUTINE               7600 7601
    PAGE 616  ROUTINE SNAP2                         8180
    PAGE 699  ''PROGRAM  OLDER.VERSION              1711
HC.DIS
    PAGE 443  PROCESS HC.ARRIVE.BATTLE               110
    PAGE 448                                         389  392  393  394  394  396  399
HC.DISENGAGE
    PAGE 300  ROUTINE HC.DISENGAGE                  4003 4021
    PAGE 338  ROUTINE HC.EMPTY                      5535
    PAGE 339                                        5578
    PAGE 375  EVENT HC.DEPART.BATTLE                7011
HC.EMPTY
    PAGE 337  ROUTINE HC.EMPTY                      5457 5471 5482 5495
    PAGE 338                                        5517 5532
    PAGE 339                                        5576
    PAGE 356  EVENT BTL.ENDED                       6274 6286
    PAGE 448  PROCESS HC.ARRIVE.BATTLE               423
HC.FIRE.MASK
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES      1477
    PAGE 685  ''PROGRAM  OLDER.VERSION               916
HC.OUTPUT.SWITCH
    PAGE  42  ''SECTION FOR DEFINITIONS             2391
    PAGE 701  ''PROGRAM  OLDER.VERSION              1821
HC.PAIRED
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES      1471
    PAGE 455  PROCESS HEL.TARGET.ACQUISITION         750  760  763  764  780
    PAGE 685  ''PROGRAM  OLDER.VERSION               910
HC.RETURN
    PAGE 450  PROCESS HC.RETURN.FARRP                506
    PAGE 451                                         522
HC.RETURN.FARRP
    PAGE  36  ''SECTION FOR PROCESSES               2058
    PAGE 337  ROUTINE HC.EMPTY                      5474 5475 5477 5478 5480 5488 5506
    PAGE 338                                        5543
    PAGE 339                                        5585
    PAGE 375  EVENT HC.DEPART.BATTLE                7019
    PAGE 445  PROCESS HC.ARRIVE.BATTLE               251
    PAGE 448                                         429
    PAGE 450  PROCESS HC.RETURN.FARRP                460
    PAGE 451                                         556  574
    PAGE 599  ROUTINE BETWEEN.ROUTINE               7484 7485
    PAGE 616  ROUTINE SNAP2                         8150
    PAGE 695  ''PROGRAM  OLDER.VERSION              1496
HC.SPACING
    PAGE 447  PROCESS HC.ARRIVE.BATTLE               359  363
    PAGE 448                                         388  389  394
HC.STATUS
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES      1475
    PAGE 377  EVENT HELO.ENGAGEMENT                 7122
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION         748
```

```
    PAGE 455                                        777  782
    PAGE 460                                       1066
    PAGE 461                                       1124 1152
    PAGE 489  PROCESS ASSESSMENT                   2560 2560
    PAGE 494  PROCESS SHOOT.OUT                    2847 2848
    PAGE 497                                       3022 3023
    PAGE 510  PROCESS HELICOPTER.FIRE              3704 3706
    PAGE 511                                       3721
    PAGE 685  ''PROGRAM  OLDER.VERSION              914
HC.SWITCH
    PAGE  42  ''SECTION FOR DEFINITIONS            2390
    PAGE 356  EVENT BTL.ENDED                      6269
    PAGE 400  EVENT START.BATTLE                   8086
    PAGE 572  ROUTINE FARRP.INPUT                  6333 6334
    PAGE 701  ''PROGRAM  OLDER.VERSION             1820
HC.TEAM
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES     1445
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS           3544 3545 3546 3547 3548 3549 3550 3551 3558 3559 3565 3568 3581 3582 3583
                                                   3584 3585 3586 3587 3588
    PAGE 291                                       3595 3597 3608 3611 3627 3628 3629 3630 3631 3632 3638 3639
    PAGE 292                                       3647 3648 3663 3666
    PAGE 304  ROUTINE REPLACE.HC                   4225
    PAGE 494  PROCESS SHOOT.OUT                    2842 2844
    PAGE 497                                       3018 3020
    PAGE 685  ''PROGRAM  OLDER.VERSION              884
HC.TIME.ALOFT
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES     1476
    PAGE 685  ''PROGRAM  OLDER.VERSION              915
HC.TYPE
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES     1468
    PAGE  52  ''SECTION FOR SUBSTITUTIONS          2971
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS           3542 3556 3579
    PAGE 291                                       3593 3636 3645
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE            4147 4149
    PAGE 376  EVENT HELO.ENGAGEMENT                7085
    PAGE 377                                       7110
    PAGE 445  PROCESS HC.ARRIVE.BATTLE              239  246
    PAGE 450  PROCESS HC.RETURN.FARRP               479
    PAGE 451                                        530  544
    PAGE 455  PROCESS HEL.TARGET.ACQUISITION        756  759  774
    PAGE 457                                        910
    PAGE 458                                        942  947
    PAGE 488  PROCESS ASSESSMENT                   2490 2490 2494 2494
    PAGE 496  PROCESS SHOOT.OUT                    2923
    PAGE 685  ''PROGRAM  OLDER.VERSION              907
    PAGE 710                                       2378
HC.UE.ID
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES     1469
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS           3543 3557 3580
    PAGE 291                                       3594 3637 3646
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE            4161
    PAGE 304  ROUTINE REPLACE.HC                   4202
    PAGE 376  EVENT HELO.ENGAGEMENT                7079
    PAGE 377                                       7115
    PAGE 378                                       7155 7185 7192
    PAGE 445  PROCESS HC.ARRIVE.BATTLE              266
```

```
    PAGE 450  PROCESS HC.RETURN.FARRP                481  484
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION         901
    PAGE 459                                        1035
    PAGE 461                                        1105
    PAGE 488  PROCESS ASSESSMENT                    2483
    PAGE 489                                        2551 2559 2559
    PAGE 494  PROCESS SHOOT.OUT                     2845
    PAGE 495                                        2890
    PAGE 497                                        3021 3037
    PAGE 498                                        3082
    PAGE 501                                        3213
    PAGE 510  PROCESS HELICOPTER.FIRE               3680
    PAGE 515                                        3971 3996
    PAGE 517                                        4075 4091 4092 4109
    PAGE 518                                        4134 4150 4151
    PAGE 685  ''PROGRAM  OLDER.VERSION               908
HC.UNIT
    PAGE 510  PROCESS HELICOPTER.FIRE               3678 3679 3695
    PAGE 511                                        3763 3769
    PAGE 512                                        3777 3787
    PAGE 514                                        3917 3919 3928 3939
    PAGE 515                                        3968 3969 3970 3993 3994 3995
    PAGE 516                                        4018
    PAGE 517                                        4072 4073 4074 4097
    PAGE 518                                        4131 4132 4133 4156
HC.UN.LOS.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES      1484
    PAGE  33                                        1863
    PAGE 300  ROUTINE HC.DISENGAGE                  4028 4030
    PAGE 311  ROUTINE INTER.HELO                    4567 4569
    PAGE 376  EVENT HELO.ENGAGEMENT                 7068
    PAGE 377                                        7133
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION         828  839  864
    PAGE 460                                        1077
    PAGE 461                                        1125
    PAGE 489  PROCESS ASSESSMENT                    2527 2530
    PAGE 685  ''PROGRAM  OLDER.VERSION               923
    PAGE 692                                        1302
HC.VISIBLE
    PAGE 497  PROCESS SHOOT.OUT                     3038
    PAGE 498                                        3046 3050
HC.WPN.TYPE
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES      1878
    PAGE 377  EVENT HELO.ENGAGEMENT                 7118
    PAGE 513  PROCESS HELICOPTER.FIRE               3880 3883 3886 3889
    PAGE 516                                        4028
    PAGE 535  ROUTINE UNIT.INPUT                    4838 4846
    PAGE 692  ''PROGRAM  OLDER.VERSION              1317
HC.X
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES      1472
    PAGE  44  ''SECTION FOR DEFINITIONS             2535
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE             4133
    PAGE 448  PROCESS HC.ARRIVE.BATTLE               392  396  399
    PAGE 685  ''PROGRAM  OLDER.VERSION               911
    PAGE 703                                        1956
```

```
HC.Y
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES          1473
    PAGE  44  ''SECTION FOR DEFINITIONS                 2536
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE                 4134
    PAGE 448  PROCESS HC.ARRIVE.BATTLE                   393  396  399
    PAGE 685  ''PROGRAM  OLDER.VERSION                   912
    PAGE 703                                            1957
HDB.FARRP
    PAGE 375  EVENT HC.DEPART.BATTLE                    6990 7001 7006 7007 7009 7013 7021
HDB.TEAM
    PAGE 375  EVENT HC.DEPART.BATTLE                    6991 7001 7014 7018 7022 7024 7025
HDM.
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE                 4129
HEADING.V
    PAGE   2  PROGRAM REVISIONS                          110
    PAGE  56  ROUTINE MAIN1                             3044
HELICOPTER
    PAGE   1  ROUTINE FOR CROSS_REFERENCING               48
    PAGE   5  ''PROGRAM'' PREAMBLE                       221
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES          1467
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS                3541 3542 3543 3545 3550 3555 3556 3557 3558 3559 3578 3579 3580 3582 3587
    PAGE 291                                            3592 3593 3594 3595 3596 3635 3636 3637 3638 3639 3644 3645 3646
    PAGE 292                                            3647 3648
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                  3924
    PAGE 299                                            3982
    PAGE 300  ROUTINE HC.DISENGAGE                      4012
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE                 4128
    PAGE 304  ROUTINE REPLACE.HC                        4218 4219
    PAGE 375  EVENT HC.DEPART.BATTLE                    6995
    PAGE 376  EVENT HELO.ENGAGEMENT                     7046 7054 7055
    PAGE 378                                            7165
    PAGE 379                                            7221
    PAGE 396  EVENT START.BATTLE                        7842
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                    99  102
    PAGE 449                                             443
    PAGE 450  PROCESS HC.RETURN.FARRP                    466
    PAGE 451                                             534  548
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION             698  702
    PAGE 456                                             821
    PAGE 457                                             895
    PAGE 458                                             967
    PAGE 459                                            1012
    PAGE 490  PROCESS ASSESSMENT                        2591
    PAGE 494  PROCESS SHOOT.OUT                         2844 2845 2847 2848 2851
    PAGE 497                                            3020 3021 3022 3023 3026
    PAGE 510  PROCESS HELICOPTER.FIRE                   3663
    PAGE 535  ROUTINE UNIT.INPUT                        4839
    PAGE 572  ROUTINE FARRP.INPUT                       6332
    PAGE 664  ''PROGRAM  OLDER.VERSION                  9659
    PAGE 685                                             906
HELICOPTERS.
    PAGE 445  PROCESS HC.ARRIVE.BATTLE                   237
HELICOPTER.
    PAGE 458  PROCESS HEL.TARGET.ACQUISITION             938
HELICOPTER.FIRE
    PAGE  36  ''SECTION FOR PROCESSES                   2062
```

```
    PAGE 301  ROUTINE HC.DISENGAGE                  4074 4075 4077 4078 4079 4080
    PAGE 312  ROUTINE INTER.HELO                    4617 4618 4620 4621 4622 4623
    PAGE 338  ROUTINE HC.EMPTY                      5561 5562 5564
    PAGE 376  EVENT HELO.ENGAGEMENT                 7042
    PAGE 377                                        7105 7106 7137
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION        1087 1088 1090
    PAGE 488  PROCESS ASSESSMENT                    2506 2507 2510 2511
    PAGE 510  PROCESS HELICOPTER.FIRE               3658 3670 3677 3678 3687 3689 3690 3691 3711 3713 3714
    PAGE 511                                        3716 3724 3738
    PAGE 512                                        3826
    PAGE 513                                        3834 3836 3837 3838 3854 3865
    PAGE 514                                        3904
    PAGE 515                                        3953 3961 3963 3964 3965 3977 3978 3979 3990 4002 4003 4004
    PAGE 517                                        4081 4082 4083 4118 4119 4120
    PAGE 518                                        4122 4129 4140 4141 4142 4168
    PAGE 600  ROUTINE BETWEEN.ROUTINE               7488
    PAGE 616  ROUTINE SNAP2                         8151
    PAGE 695  ''PROGRAM  OLDER.VERSION              1500
HELICOPTER.FIRES
    PAGE 300  ROUTINE HC.DISENGAGE                  4010
    PAGE 301                                        4073
    PAGE 312  ROUTINE INTER.HELO                    4616
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION        1074
HELI.COPTER
    PAGE   5  ''PROGRAM'' PREAMBLE                   221
    PAGE 664  ''PROGRAM  OLDER.VERSION              9659
HELO.EN
    PAGE 378  EVENT HELO.ENGAGEMENT                 7151
    PAGE 379                                        7204 7221
HELO.ENGAGEMENT
    PAGE  40  ''SECTION FOR EVENTS                  2279
    PAGE 301  ROUTINE HC.DISENGAGE                  4066 4067 4069 4070
    PAGE 312  ROUTINE INTER.HELO                    4609 4610 4612 4613
    PAGE 338  ROUTINE HC.EMPTY                      5554 5555 5557 5558
    PAGE 376  EVENT HELO.ENGAGEMENT                 7030 7058 7059 7060 7062 7063
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION         868
    PAGE 458                                         950  968
    PAGE 460                                        1043 1081 1082 1084 1085
    PAGE 602  ROUTINE BETWEEN.ROUTINE               7604 7605
    PAGE 616  ROUTINE SNAP2                         8181
    PAGE 699  ''PROGRAM  OLDER.VERSION              1716
HELO.ENGAGEMENTS
    PAGE 300  ROUTINE HC.DISENGAGE                  4010
    PAGE 301                                        4065
    PAGE 312  ROUTINE INTER.HELO                    4608
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION        1076
HELO.FIRE
    PAGE 510  PROCESS HELICOPTER.FIRE               3705
    PAGE 511                                        3744
    PAGE 512                                        3831
    PAGE 513                                        3883
    PAGE 515                                        3958
    PAGE 517                                        4067
    PAGE 600  ROUTINE BETWEEN.ROUTINE               7489
HEL.ENGAGEMENT
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION         703
```

```
HEL.IS.KILLED.INDIC
     PAGE  36  ''SECTION FOR PROCESSES                2065
     PAGE 488  PROCESS ASSESSMENT                     2510
     PAGE 510  PROCESS HELICOPTER.FIRE                3687 3711
     PAGE 512                                         3826
     PAGE 513                                         3834
     PAGE 515                                         3953 3961
     PAGE 517                                         4118
     PAGE 695  ''PROGRAM  OLDER.VERSION               1503
HEL.RNG.COM
     PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4153
HEL.TARGET.A
     PAGE 454  PROCESS HEL.TARGET.ACQUISITION          719
     PAGE 455                                          791
     PAGE 456                                          821
HEL.TARGET.ACQUISITION
     PAGE  37  ''SECTION FOR PROCESSES                2078
     PAGE 311  ROUTINE INTER.HELO                     4545 4547 4549 4550 4551 4552 4553 4555
     PAGE 312                                         4625
     PAGE 338  ROUTINE HC.EMPTY                       5520 5521 5523 5524 5525 5526 5527 5530 5531 5538 5539 5542 5546 5548 5549
     PAGE 376  EVENT HELO.ENGAGEMENT                  7039
     PAGE 443  PROCESS HC.ARRIVE.BATTLE                130  131  134  135  136  137
     PAGE 449                                          449
     PAGE 454  PROCESS HEL.TARGET.ACQUISITION          692  714  723
     PAGE 460                                         1063
     PAGE 461                                         1143
     PAGE 600  ROUTINE BETWEEN.ROUTINE                7492 7493
     PAGE 616  ROUTINE SNAP2                          8152
     PAGE 696  ''PROGRAM  OLDER.VERSION               1516
HE.COST
     PAGE   9  ''SECTION FOR PERMANENT_ENTITIES        494
     PAGE  44  ''SECTION FOR DEFINITIONS              2537
     PAGE 249  ROUTINE WEIGHTED.VOLLEYS               1434
     PAGE 555  ROUTINE MUNS.INPUT                     5643
     PAGE 668  ''PROGRAM  OLDER.VERSION               9935
     PAGE 703                                         1958
HE.DUST.DURATION
     PAGE   9  ''SECTION FOR PERMANENT_ENTITIES        498
     PAGE  44  ''SECTION FOR DEFINITIONS              2538
     PAGE 193  ROUTINE FINAL.COVERAGE                 8887
     PAGE 555  ROUTINE MUNS.INPUT                     5647
     PAGE 668  ''PROGRAM  OLDER.VERSION               9939
     PAGE 703                                         1959
HE.FARRP
     PAGE 337  ROUTINE HC.EMPTY                       5459 5469 5469 5475 5486 5501
     PAGE 338                                         5521 5537 5545 5555 5562 5567 5569
     PAGE 339                                         5580 5587
HE.FRACT.COV
     PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9001
     PAGE 197                                         9091
     PAGE 198                                         9108
HE.ICM
     PAGE 471  PROCESS TARGET.REPORT                  1556
HE.ID
     PAGE   9  ''SECTION FOR PERMANENT_ENTITIES        492
     PAGE  47  ''SECTION FOR DEFINITIONS              2670
```

```
    PAGE 262  ROUTINE BTRY.EFFECTS                    2124
    PAGE 263                                          2165
    PAGE 555  ROUTINE MUNS INPUT                      5640
    PAGE 592  ROUTINE AMMO.RPT                        7152
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                 7362
    PAGE 611  ROUTINE OUTPUT.ATTRITION                7965
    PAGE 668  ''PROGRAM  OLDER.VERSION                9933
    PAGE 706                                          2094
HE.LA.INPUT
    PAGE 521  ROUTINE MAIN2                           4242 4244
    PAGE 558  ROUTINE HE.LA.INPUT                     5750
HE.MIN.MARG.EFF
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES         499
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ            9332
    PAGE 555  ROUTINE MUNS.INPUT                      5641
    PAGE 668  ''PROGRAM  OLDER.VERSION                9940
HE.MUNITION
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES         479  491  501
    PAGE 555  ROUTINE MUNS.INPUT                      5614 5638 5640 5641 5642 5643 5644 5645 5647 5648 5651 5659
    PAGE 556                                          5666
    PAGE 558  ROUTINE HE.LA.INPUT                     5758 5763 5770 5787 5794 5801
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                 7357
    PAGE 618  ROUTINE SNAP.R                          8225
    PAGE 668  ''PROGRAM  OLDER.VERSION                9920 9932 9942
    PAGE 712                                          2442
HE.N.TUBES
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                 7243 7248
    PAGE 596                                          7341
    PAGE 597                                          7362 7397
HE.N.VOLS
    PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION           9090
    PAGE 198                                          9111
HE.OR.ICM.
    PAGE 185  ROUTINE FA.BN.ASGN                      8462 8484 8508
HE.OR.ICM.COMPUTATION
    PAGE   2  PROGRAM REVISIONS                        117
    PAGE 185  ROUTINE FA.BN.ASGN                      8478
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION           8989 9023
HE.OR.ICM.COMPUTE
    PAGE 634  FUNCTION FEBA.BAND                      8778
HE.RANGE.HACK
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         508  519  522
    PAGE 188  ROUTINE FINAL.COVERAGE                  8576 8577
    PAGE 555  ROUTINE MUNS.INPUT                      5632
    PAGE 618  ROUTINE SNAP.R                          8226
    PAGE 669  ''PROGRAM  OLDER.VERSION                9949 9960 9963
    PAGE 712                                          2443
HE.RH.RANGE
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         509
    PAGE 176  ROUTINE EST.COVERAGE                    8028 8052 8053 8054
    PAGE 177                                          8063 8064 8064 8068 8074 8075
    PAGE 188  ROUTINE FINAL.COVERAGE                  8571 8595 8596 8597
    PAGE 189                                          8609 8610 8610 8617 8623 8624
    PAGE 263  ROUTINE BTRY.EFFECTS                    2167
    PAGE 556  ROUTINE MUNS.INPUT                      5670
    PAGE 636  FUNCTION HE.WLA                         8879 8883
```

```
    PAGE 669  ''PROGRAM  OLDER.VERSION                9950
HE.RH.RANGE.
    PAGE 177  ROUTINE EST.COVERAGE                    8062
    PAGE 189  ROUTINE FINAL.COVERAGE                  8608
HE.RH.ROUND.CPE
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         511
    PAGE  45  ''SECTION FOR DEFINITIONS               2544
    PAGE 176  ROUTINE EST.COVERAGE                    8059 8060
    PAGE 177                                          8069 8069 8070 8073
    PAGE 189  ROUTINE FINAL.COVERAGE                  8605 8606 8618 8618 8619 8622
    PAGE 556  ROUTINE MUNS.INPUT                      5669
    PAGE 669  ''PROGRAM  OLDER.VERSION                9952
    PAGE 703                                          1965
HE.RH.TOTAL.CPE
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         510
    PAGE  45  ''SECTION FOR DEFINITIONS               2543
    PAGE 176  ROUTINE EST.COVERAGE                    8055 8057
    PAGE 177                                          8065 8065 8067 8073
    PAGE 188  ROUTINE FINAL.COVERAGE                  8598 8600
    PAGE 189                                          8611 8611 8613 8622
    PAGE 556  ROUTINE MUNS.INPUT                      5668
    PAGE 669  ''PROGRAM  OLDER.VERSION                9951
    PAGE 703                                          1964
HE.ROUND.RAD
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES         496
    PAGE  44  ''SECTION FOR DEFINITIONS               2539
    PAGE 555  ROUTINE MUNS.INPUT                      5645
    PAGE 636  FUNCTION HE.WLA                         8899
    PAGE 668  ''PROGRAM  OLDER.VERSION                9937
    PAGE 703                                          1960
HE.SIDE
    PAGE 337  ROUTINE HC.EMPTY                        5460 5469
HE.TB.RH.LIST
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES         502
    PAGE  10                                           513
    PAGE 176  ROUTINE EST.COVERAGE                    8027
    PAGE 188  ROUTINE FINAL.COVERAGE                  8570 8576
    PAGE 263  ROUTINE BTRY.EFFECTS                    2166
    PAGE 556  ROUTINE MUNS.INPUT                      5666
    PAGE 558  ROUTINE HE.LA.INPUT                     5763 5770 5794 5801
    PAGE 636  FUNCTION HE.WLA                         8882
    PAGE 668  ''PROGRAM  OLDER.VERSION                9943
    PAGE 669                                          9954
HE.VOLLEY.RAD
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES         495
    PAGE  44  ''SECTION FOR DEFINITIONS               2541
    PAGE 177  ROUTINE EST.COVERAGE                    8081
    PAGE 189  ROUTINE FINAL.COVERAGE                  8630
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ            9329
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS                1432
    PAGE 262  ROUTINE BTRY.EFFECTS                    2131
    PAGE 263                                          2177
    PAGE 555  ROUTINE MUNS.INPUT                      5644
    PAGE 668  ''PROGRAM  OLDER.VERSION                9936
    PAGE 703                                          1962
```

```
HE.VOL.DUST.RAD
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES          497
    PAGE  44  ''SECTION FOR DEFINITIONS                2540
    PAGE 193  ROUTINE FINAL.COVERAGE                   8886
    PAGE 555  ROUTINE MUNS.INPUT                       5648
    PAGE 668  ''PROGRAM  OLDER.VERSION                 9938
    PAGE 703                                           1961
HE.WEIGHT
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES          493
    PAGE  45  ''SECTION FOR DEFINITIONS                2542
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS                 1436
    PAGE 555  ROUTINE MUNS.INPUT                       5642
    PAGE 592  ROUTINE AMMO.RPT                         7153
    PAGE 668  ''PROGRAM  OLDER.VERSION                 9934
    PAGE 703                                           1963
HE.WLA
    PAGE  50  ''SECTION FOR DEFINITIONS                2857
    PAGE 178  ROUTINE EST.COVERAGE                     8161 8162
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION            9042 9045
    PAGE 197                                           9098
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ             9353 9354
    PAGE 636  FUNCTION HE.WLA                          8859 8872 8892
    PAGE 709  ''PROGRAM  OLDER.VERSION                 2276
HE.W.VOLS
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION            9001
    PAGE 197                                           9089 9093 9096
HF.DESTRUCT.INDIC
    PAGE  36  ''SECTION FOR PROCESSES                  2067
    PAGE 338  ROUTINE HC.EMPTY                         5564
    PAGE 488  PROCESS ASSESSMENT                       2511
    PAGE 510  PROCESS HELICOPTER.FIRE                  3689 3713
    PAGE 513                                           3836
    PAGE 515                                           3963
    PAGE 517                                           4119
    PAGE 695  ''PROGRAM  OLDER.VERSION                 1505
HF.ENEMY.UNIT
    PAGE  36  ''SECTION FOR PROCESSES                  2064
    PAGE 695  ''PROGRAM  OLDER.VERSION                 1502
HF.FIRING.TABLE
    PAGE  36  ''SECTION FOR PROCESSES                  2068
    PAGE 511  PROCESS HELICOPTER.FIRE                  3731 3733
    PAGE 513                                           3854 3865
    PAGE 695  ''PROGRAM  OLDER.VERSION                 1506
HF.RANGE
    PAGE  36  ''SECTION FOR PROCESSES                  2069
    PAGE 378  EVENT HELO.ENGAGEMENT                    7146
    PAGE 695  ''PROGRAM  OLDER.VERSION                 1507
HF.REINFORCE.IND
    PAGE  36  ''SECTION FOR PROCESSES                  2071
    PAGE 312  ROUTINE INTER.HELO                       4621
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION           1090
    PAGE 510  PROCESS HELICOPTER.FIRE                  3690 3714
    PAGE 513                                           3837
    PAGE 515                                           3964
    PAGE 517                                           4120
    PAGE 695  ''PROGRAM  OLDER.VERSION                 1509
```

```
HF.RESULTS
     PAGE  36  ''SECTION FOR PROCESSES                2066
     PAGE 511  PROCESS HELICOPTER.FIRE                3724
     PAGE 514                                         3904
     PAGE 515                                         3990
     PAGE 518                                         4129
     PAGE 695  ''PROGRAM  OLDER.VERSION               1504
HF.SO.LIST
     PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1819
     PAGE  37  ''SECTION FOR PROCESSES                2076
     PAGE 378  EVENT HELO.ENGAGEMENT                  7161
     PAGE 510  PROCESS HELICOPTER.FIRE                3691
     PAGE 511                                         3716 3730 3739
     PAGE 513                                         3839
     PAGE 515                                         3966
     PAGE 518                                         4123 4168
     PAGE 691  ''PROGRAM  OLDER.VERSION               1258
     PAGE 696                                         1514
HF.TEAM
     PAGE  36  ''SECTION FOR PROCESSES                2070
     PAGE 301  ROUTINE HC.DISENGAGE                   4075 4078
     PAGE 312  ROUTINE INTER.HELO                     4618
     PAGE 338  ROUTINE HC.EMPTY                       5562
     PAGE 377  EVENT HELO.ENGAGEMENT                  7106
     PAGE 378                                         7147
     PAGE 460  PROCESS HEL.TARGET.ACQUISITION         1088
     PAGE 510  PROCESS HELICOPTER.FIRE                3677 3678
     PAGE 515                                         3977 3978 3979 4002 4003 4004
     PAGE 517                                         4081 4082 4083
     PAGE 518                                         4140 4141 4142
     PAGE 695  ''PROGRAM  OLDER.VERSION               1508
HIGH.FRAC.RANGE
     PAGE 443  PROCESS HC.ARRIVE.BATTLE                109
     PAGE 447                                          358  362  366
HIT.
     PAGE 282  ROUTINE AC.DF.EFFECTS                  3112
HOME.
     PAGE 508  PROCESS CAS.MISSION                    3625
HOW
     PAGE   1  ROUTINE FOR CROSS_REFERENCING            49
     PAGE  16  ''SECTION FOR PERMANENT_ENTITIES         883
     PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES        1486
     PAGE  77  ROUTINE ADJUST                         3916
     PAGE 252  ROUTINE MINE.EFFECTS                   1608 1621
     PAGE 271  ROUTINE BTRY.EFFECTS                   2607 2614 2615
     PAGE 279  ROUTINE AC.BOMB.EFFECTS                2961 2962
     PAGE 282  ROUTINE AC.DF.EFFECTS                  3141 3142
     PAGE 384  EVENT OFF.LINE.ATTRITION               7446
     PAGE 463  PROCESS HOW.REPAIR                     1170 1179 1181 1188 1193 1208 1212
     PAGE 485  PROCESS FIRE.MISSION                   2330 2333 2336 2338 2339 2340
     PAGE 551  ROUTINE BTRY.INPUT                     5490 5491 5492 5495 5501
     PAGE 609  ROUTINE KV.SCOREBOARD                  7882
     PAGE 656  ROUTINE AR.DETECTION                   9414
     PAGE 675  ''PROGRAM  OLDER.VERSION                325
     PAGE 685                                          925
```

```
HOW.NAME
    PAGE 548  ROUTINE TB.INPUT                    5314 5327 5329 5332
HOW.REPAIR
    PAGE  37  ''SECTION FOR PROCESSES             2084
    PAGE 463  PROCESS HOW.REPAIR                  1168
    PAGE 485  PROCESS FIRE.MISSION                2340
    PAGE 600  ROUTINE BETWEEN.ROUTINE             7496 7497
    PAGE 616  ROUTINE SNAP2                       8154
    PAGE 696  ''PROGRAM  OLDER.VERSION            1522
HQ.SET.BACK
    PAGE  41  ''SECTION FOR DEFINITIONS           2367
    PAGE  68  ROUTINE GENERAL.BATTLE              3537 3545
    PAGE  69                                      3567 3569 3574 3577
    PAGE 569  ROUTINE TBF.INPUT                   6208 6213 6213
    PAGE 700  ''PROGRAM  OLDER.VERSION            1797
HRF.FARRP
    PAGE 450  PROCESS HC.RETURN.FARRP              462  474  475  493  504
    PAGE 451                                       563  564  564  565  572
    PAGE 452                                       584  587  592  611  615  616  617  620  621  622  626  627  628  633
    PAGE 453                                       645  647  649  656  663  666  671  680  686
HRF.TEAM
    PAGE 450  PROCESS HC.RETURN.FARRP              463  477  488  490  500  504
    PAGE 451                                       529  533  543  547  561  562  572  573
    PAGE 452                                       579  580  581  585
    PAGE 453                                       642  646  667  669  670  681  687
HR.HOW
    PAGE  37  ''SECTION FOR PROCESSES             2085
    PAGE 696  ''PROGRAM  OLDER.VERSION            1523
HTA.BATTLE
    PAGE  37  ''SECTION FOR PROCESSES             2081
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION       714
    PAGE 696  ''PROGRAM  OLDER.VERSION            1519
HTA.REINFORCE.IND
    PAGE  37  ''SECTION FOR PROCESSES             2082
    PAGE 311  ROUTINE INTER.HELO                  4550
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION       723
    PAGE 460                                      1063
    PAGE 461                                      1143
    PAGE 696  ''PROGRAM  OLDER.VERSION            1520
HTA.TEAM
    PAGE  37  ''SECTION FOR PROCESSES             2079
    PAGE 311  ROUTINE INTER.HELO                  4547 4552
    PAGE 338  ROUTINE HC.EMPTY                    5521 5523 5525 5530 5532 5538 5542 5546 5548 5549
    PAGE 443  PROCESS HC.ARRIVE.BATTLE             131  137
    PAGE 696  ''PROGRAM  OLDER.VERSION            1517
HT.ARR.BTL.TIME
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES    1450
    PAGE  49  ''SECTION FOR DEFINITIONS           2782
    PAGE 402  EVENT START.BATTLE                  8158
    PAGE 443  PROCESS HC.ARRIVE.BATTLE             114
    PAGE 685  ''PROGRAM  OLDER.VERSION             889
    PAGE 707                                      2204
HT.FARRP
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES    1446
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS          3549 3583
    PAGE 291                                      3629
```

```
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4146
    PAGE 311  ROUTINE INTER.HELO                     4547
    PAGE 338  ROUTINE HC.EMPTY                       5521 5555 5562
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                131
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          713  714  717  718  727
    PAGE 455                                          788  789
    PAGE 457                                          871  876  891  892  899
    PAGE 458                                          953  961  971  978
    PAGE 459                                         1008 1009
    PAGE 460                                         1046
    PAGE 461                                         1114 1135
    PAGE 462                                         1156 1161 1163
    PAGE 496  PROCESS SHOOT.OUT                      2924
    PAGE 510  PROCESS HELICOPTER.FIRE                3678
    PAGE 515                                         3977 3978 4002 4003
    PAGE 517                                         4081 4082
    PAGE 518                                         4140 4141
    PAGE 685  ''PROGRAM  OLDER.VERSION                885
HT.LIST
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES       1346
    PAGE  26                                         1465
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS             3551 3588
    PAGE 291                                         3628
    PAGE 304  ROUTINE REPLACE.HC                     4224
    PAGE 494  PROCESS SHOOT.OUT                      2842
    PAGE 497                                         3018
    PAGE 683  ''PROGRAM  OLDER.VERSION                785
    PAGE 685                                          904
HT.LOITER.TIME
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES       1449
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS             3547 3585
    PAGE 291                                         3631
    PAGE 402  EVENT START.BATTLE                     8164
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                118
    PAGE 450  PROCESS HC.RETURN.FARRP                 500
    PAGE 453                                          670
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          741
    PAGE 685  ''PROGRAM  OLDER.VERSION                888
HT.MEMBER.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES       1462 1482
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS             3550 3559 3587
    PAGE 291                                         3597 3639
    PAGE 292                                         3648
    PAGE 300  ROUTINE HC.DISENGAGE                   4026
    PAGE 301                                         4088
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4148
    PAGE 304  ROUTINE REPLACE.HC                     4198 4216 4218
    PAGE 311  ROUTINE INTER.HELO                     4565
    PAGE 376  EVENT HELO.ENGAGEMENT                  7066 7078
    PAGE 377                                         7109
    PAGE 378                                         7167 7180
    PAGE 445  PROCESS HC.ARRIVE.BATTLE                238  245
    PAGE 448                                          390
    PAGE 450  PROCESS HC.RETURN.FARRP                 477
    PAGE 451                                          529  533  543  547
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          746
```

```
    PAGE 455                                        755  758  772  800
    PAGE 456                                        861
    PAGE 458                                        944
    PAGE 459                                        993
    PAGE 460                                       1064
    PAGE 461                                       1104 1122 1151
    PAGE 488  PROCESS ASSESSMENT                   2489 2493
    PAGE 489                                       2556 2558
    PAGE 494  PROCESS SHOOT.OUT                    2844
    PAGE 497                                       3020
    PAGE 498                                       3040
    PAGE 685  ''PROGRAM  OLDER.VERSION              901  921
HT.MOVE.TIME
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES     1448
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS           3548 3586
    PAGE 291                                       3632
    PAGE 337  ROUTINE HC.EMPTY                     5503
    PAGE 338                                       5542
    PAGE 339                                       5584
    PAGE 375  EVENT HC.DEPART.BATTLE               7018
    PAGE 391  EVENT SEND.TEAM                      7619
    PAGE 445  PROCESS HC.ARRIVE.BATTLE              255
    PAGE 448                                        433
    PAGE 453  PROCESS HC.RETURN.FARRP               669
    PAGE 685  ''PROGRAM  OLDER.VERSION              887
HT.SIDE
    PAGE 298  ROUTINE HC.COMPUTE.TIMES             3932 3934
    PAGE 443  PROCESS HC.ARRIVE.BATTLE              115  119  121
    PAGE 444                                        157  170  188  201
    PAGE 445                                        218
    PAGE 447                                        356
    PAGE 448                                        406  418  426
    PAGE 449                                        444
HT.STATUS
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES     1447
    PAGE  52  ''SECTION FOR SUBSTITUTIONS          2980
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS           3546 3584
    PAGE 291                                       3630
    PAGE 337  ROUTINE HC.EMPTY                     5477 5488 5504
    PAGE 338                                       5523 5549 5571
    PAGE 339                                       5591
    PAGE 375  EVENT HC.DEPART.BATTLE               7025
    PAGE 391  EVENT SEND.TEAM                      7607 7610 7612
    PAGE 443  PROCESS HC.ARRIVE.BATTLE              137
    PAGE 445                                        250
    PAGE 448                                        428
    PAGE 451  PROCESS HC.RETURN.FARRP               561  562  573  575
    PAGE 452                                        579  580
    PAGE 453                                        642
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION       1070
    PAGE 461                                       1146
    PAGE 685  ''PROGRAM  OLDER.VERSION              886
    PAGE 711                                       2387
HT.TARGET.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES     1012
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES     1463
```

```
    PAGE 300  ROUTINE HC.DISENGAGE                  4045 4046 4048
    PAGE 312  ROUTINE INTER.HELO                    4585 4586 4588
    PAGE 458  PROCESS HEL.TARGET.ACQUISITION         939  940
    PAGE 677  ''PROGRAM  OLDER.VERSION               454
    PAGE 685                                         902
HT.TERMINATOR
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES      1451
    PAGE 301  ROUTINE HC.DISENGAGE                  4078
    PAGE 337  ROUTINE HC.EMPTY                      5478 5492 5505
    PAGE 338                                        5525 5548 5572
    PAGE 339                                        5590
    PAGE 375  EVENT HC.DEPART.BATTLE                7024
    PAGE 443  PROCESS HC.ARRIVE.BATTLE               125  145
    PAGE 445                                         249
    PAGE 448                                         427
    PAGE 450  PROCESS HC.RETURN.FARRP                488  490
    PAGE 452                                         581
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION         722
    PAGE 460                                        1062
    PAGE 461                                        1142
    PAGE 510  PROCESS HELICOPTER.FIRE               3688 3712
    PAGE 513                                        3835
    PAGE 515                                        3962
    PAGE 517                                        4121
    PAGE 685  ''PROGRAM  OLDER.VERSION               890
HW.BTRY
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES      1487
    PAGE  46  ''SECTION FOR DEFINITIONS             2621
    PAGE 463  PROCESS HOW.REPAIR                    1179
    PAGE 551  ROUTINE BTRY.INPUT                    5492
    PAGE 685  ''PROGRAM  OLDER.VERSION               926
    PAGE 705                                        2044
HW.LFAIL.RNDS
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES      1489
    PAGE  46  ''SECTION FOR DEFINITIONS             2623
    PAGE 463  PROCESS HOW.REPAIR                    1208
    PAGE 485  PROCESS FIRE.MISSION                  2336 2339
    PAGE 551  ROUTINE BTRY.INPUT                    5501
    PAGE 685  ''PROGRAM  OLDER.VERSION               928
    PAGE 705                                        2046
HW.SFAIL.RNDS
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES      1488
    PAGE  46  ''SECTION FOR DEFINITIONS             2622
    PAGE 463  PROCESS HOW.REPAIR                    1188 1193
    PAGE 485  PROCESS FIRE.MISSION                  2333 2338
    PAGE 551  ROUTINE BTRY.INPUT                    5495
    PAGE 685  ''PROGRAM  OLDER.VERSION               927
    PAGE 705                                        2045
HYP.
    PAGE 408  EVENT UPDATE.LOC                      8426 8437 8438 8439
ICM.FRACT.COV
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION         9002
    PAGE 197                                        9071 9101
ICM.N.VOLS
    PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION         9070 9103
```

```
ICM.WLA
    PAGE  50  ''SECTION FOR DEFINITIONS              2858
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9037 9040
    PAGE 638  FUNCTION ICM.WLA                       8957 8971
    PAGE 709  ''PROGRAM  OLDER.VERSION               2277
ICM.W.VOLS
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9001
    PAGE 197                                         9069 9073 9077 9096
IC.COST
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES        528
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS               1427
    PAGE 556  ROUTINE MUNS.INPUT                     5682
    PAGE 669  ''PROGRAM  OLDER.VERSION               9969
IC.ID
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES        526
    PAGE  47  ''SECTION FOR DEFINITIONS              2671
    PAGE 262  ROUTINE BTRY.EFFECTS                   2133
    PAGE 263                                         2182
    PAGE 556  ROUTINE MUNS.INPUT                     5679
    PAGE 592  ROUTINE AMMO.RPT                       7156
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                7370
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7966
    PAGE 669  ''PROGRAM  OLDER.VERSION               9967
    PAGE 706                                         2095
IC.MIN.MARG.EFF
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES        533
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9334
    PAGE 556  ROUTINE MUNS.INPUT                     5680
    PAGE 669  ''PROGRAM  OLDER.VERSION               9974
IC.MUNITION
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        368
    PAGE  10                                          525  535
    PAGE 555  ROUTINE MUNS.INPUT                     5614
    PAGE 556                                         5677 5679 5680 5681 5682 5683 5684 5685 5686 5689 5692 5693 5701
    PAGE 560  ROUTINE  RUL.EN.INPUT                  5835
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                7365
    PAGE 618  ROUTINE SNAP.R                         8227
    PAGE 666  ''PROGRAM  OLDER.VERSION               9810
    PAGE 669                                         9966 9976
    PAGE 712                                         2444
IC.N.SUBM
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES        532
    PAGE  45  ''SECTION FOR DEFINITIONS              2547
    PAGE 190  ROUTINE FINAL.COVERAGE                 8711
    PAGE 263  ROUTINE BTRY.EFFECTS                   2155 2191
    PAGE 556  ROUTINE MUNS.INPUT                     5686
    PAGE 638  FUNCTION ICM.WLA                       8988
    PAGE 669  ''PROGRAM  OLDER.VERSION               9973
    PAGE 703                                         1968
IC.N.TUBES
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                7243 7249
    PAGE 596                                         7344
    PAGE 597                                         7370 7397
IC.RANGE.HACK
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES        545
    PAGE 555  ROUTINE MUNS.INPUT                     5632
```

```
    PAGE 618  ROUTINE SNAP.R                          8228
    PAGE 669  ''PROGRAM  OLDER.VERSION                9986
    PAGE 712                                          2445
IC.RELIABILITY
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         529
    PAGE  45  ''SECTION FOR DEFINITIONS               2545
    PAGE 190  ROUTINE FINAL.COVERAGE                  8701
    PAGE 262  ROUTINE BTRY.EFFECTS                    2134
    PAGE 263                                          2192
    PAGE 556  ROUTINE MUNS.INPUT                      5683
    PAGE 638  FUNCTION ICM.WLA                        9004
    PAGE 669  ''PROGRAM  OLDER.VERSION                9970
    PAGE 703                                          1966
IC.RH.RANGE
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         546
    PAGE 177  ROUTINE EST.COVERAGE                    8084 8108 8109 8110 8119
    PAGE 178                                          8120 8120 8124 8130 8131
    PAGE 189  ROUTINE FINAL.COVERAGE                  8636 8652 8653 8654
    PAGE 190                                          8666 8667 8667 8674 8680 8681
    PAGE 263  ROUTINE BTRY.EFFECTS                    2184
    PAGE 556  ROUTINE MUNS.INPUT                      5705
    PAGE 638  FUNCTION ICM.WLA                        8973 8979
    PAGE 669  ''PROGRAM  OLDER.VERSION                9987
IC.RH.RANGE.
    PAGE 177  ROUTINE EST.COVERAGE                    8118
    PAGE 190  ROUTINE FINAL.COVERAGE                  8665
IC.RH.ROUND.CPE
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         548
    PAGE  45  ''SECTION FOR DEFINITIONS               2549
    PAGE 177  ROUTINE EST.COVERAGE                    8115 8116
    PAGE 178                                          8125 8125 8126 8129
    PAGE 189  ROUTINE FINAL.COVERAGE                  8662
    PAGE 190                                          8663 8675 8675 8676 8679
    PAGE 556  ROUTINE MUNS.INPUT                      5704
    PAGE 669  ''PROGRAM  OLDER.VERSION                9989
    PAGE 703                                          1970
IC.RH.TOTAL.CPE
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         547
    PAGE  45  ''SECTION FOR DEFINITIONS               2548
    PAGE 177  ROUTINE EST.COVERAGE                    8111 8113
    PAGE 178                                          8121 8121 8123 8129
    PAGE 189  ROUTINE FINAL.COVERAGE                  8655 8657
    PAGE 190                                          8668 8668 8670 8679
    PAGE 556  ROUTINE MUNS.INPUT                      5703
    PAGE 669  ''PROGRAM  OLDER.VERSION                9988
    PAGE 703                                          1969
IC.SUBM.INDEX
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         530
    PAGE 190  ROUTINE FINAL.COVERAGE                  8707 8709 8712
    PAGE 263  ROUTINE BTRY.EFFECTS                    2193
    PAGE 556  ROUTINE MUNS.INPUT                      5684
    PAGE 638  FUNCTION ICM.WLA                        8989
    PAGE 669  ''PROGRAM  OLDER.VERSION                9971
IC.TB.INTERCEPT
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         537
    PAGE  48  ''SECTION FOR DEFINITIONS               2740
```

```
    PAGE 178  ROUTINE EST.COVERAGE                    8154 8158
    PAGE 190  ROUTINE FINAL.COVERAGE                  8704
    PAGE 263  ROUTINE BTRY.EFFECTS                    2155
    PAGE 556  ROUTINE MUNS.INPUT                      5693
    PAGE 638  FUNCTION ICM.WLA                        8986
    PAGE 669  ''PROGRAM  OLDER.VERSION                9978
    PAGE 707                                          2162
IC.TB.RH.LIST
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         539  550
    PAGE 177  ROUTINE EST.COVERAGE                    8083
    PAGE 189  ROUTINE FINAL.COVERAGE                  8635
    PAGE 263  ROUTINE BTRY.EFFECTS                    2183
    PAGE 556  ROUTINE MUNS.INPUT                      5701
    PAGE 638  FUNCTION ICM.WLA                        8978
    PAGE 669  ''PROGRAM  OLDER.VERSION                9980 9991
IC.TB.SLOPE
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         536
    PAGE  48  ''SECTION FOR DEFINITIONS               2741
    PAGE 178  ROUTINE EST.COVERAGE                    8153 8157
    PAGE 190  ROUTINE FINAL.COVERAGE                  8703
    PAGE 263  ROUTINE BTRY.EFFECTS                    2154
    PAGE 556  ROUTINE MUNS.INPUT                      5692
    PAGE 638  FUNCTION ICM.WLA                        8987
    PAGE 669  ''PROGRAM  OLDER.VERSION                9977
    PAGE 707                                          2163
IC.VOLLEY.RAD
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         531
    PAGE  45  ''SECTION FOR DEFINITIONS               2546
    PAGE 178  ROUTINE EST.COVERAGE                    8137
    PAGE 190  ROUTINE FINAL.COVERAGE                  8687
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ            9327
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS                1425
    PAGE 262  ROUTINE BTRY.EFFECTS                    2135
    PAGE 263                                          2189
    PAGE 556  ROUTINE MUNS.INPUT                      5685
    PAGE 669  ''PROGRAM  OLDER.VERSION                9972
    PAGE 703                                          1967
IC.WEIGHT
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         527
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS                1429
    PAGE 556  ROUTINE MUNS.INPUT                      5681
    PAGE 592  ROUTINE AMMO.RPT                        7157
    PAGE 669  ''PROGRAM  OLDER.VERSION                9968
IF.RATE.LIST
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES         832
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES        1501
    PAGE  49  ''SECTION FOR DEFINITIONS               2813
    PAGE 165  ROUTINE CFR.DEGRADE                     7594 7598
    PAGE 205  ROUTINE NOISE.DEGRADE                   9404 9407
    PAGE 246  ROUTINE VOLLEY                          1358
    PAGE 674  ''PROGRAM  OLDER.VERSION                 274
    PAGE 686                                           940
    PAGE 708                                          2234
IF.V
    PAGE 246  ROUTINE VOLLEY                          1355 1356 1357 1358
```

```
IF.VOLLEY
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1497
    PAGE 165  ROUTINE CFR.DEGRADE                       7599
    PAGE 205  ROUTINE NOISE.DEGRADE                     9408
    PAGE 246  ROUTINE VOLLEY                            1355
    PAGE 686  ''PROGRAM  OLDER.VERSION                   936
IF.V.BTRY
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1499
    PAGE 165  ROUTINE CFR.DEGRADE                       7596
    PAGE 205  ROUTINE NOISE.DEGRADE                     9411
    PAGE 246  ROUTINE VOLLEY                            1357
    PAGE 686  ''PROGRAM  OLDER.VERSION                   938
IF.V.TIME
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1498
    PAGE  48  ''SECTION FOR DEFINITIONS                 2742
    PAGE 165  ROUTINE CFR.DEGRADE                       7597
    PAGE 205  ROUTINE NOISE.DEGRADE                     9406
    PAGE 246  ROUTINE VOLLEY                            1356
    PAGE 686  ''PROGRAM  OLDER.VERSION                   937
    PAGE 707                                            2164
ILLUMINATION.COMP
    PAGE 634  FUNCTION FEBA.BAND                        8778
ILLUM.COMPUT
    PAGE 199  ROUTINE ILLUM.COMPUTATION                 9175
ILLUM.DEBUG
    PAGE  42  ''SECTION FOR DEFINITIONS                 2418
    PAGE 199  ROUTINE ILLUM.COMPUTATION                 9171
    PAGE 200  ROUTINE ILLUM.EFFECTS                     9204
    PAGE 201                                            9265 9289
    PAGE 220  ROUTINE REQUEST.ILLUM                      153
    PAGE 576  ROUTINE ILLUM.INPUT                       6504
    PAGE 701  ''PROGRAM  OLDER.VERSION                  1848
ILLUM.DURATION
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES           560
    PAGE  45  ''SECTION FOR DEFINITIONS                 2552
    PAGE 200  ROUTINE ILLUM.EFFECTS                     9211 9220
    PAGE 576  ROUTINE ILLUM.INPUT                       6514
    PAGE 669  ''PROGRAM  OLDER.VERSION                     1
    PAGE 703                                            1973
ILLUM.EFFECT
    PAGE 200  ROUTINE ILLUM.EFFECTS                     9212
    PAGE 201                                            9268 9293
ILLUM.ID
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES           557
    PAGE  47  ''SECTION FOR DEFINITIONS                 2672
    PAGE 576  ROUTINE ILLUM.INPUT                       6511
    PAGE 592  ROUTINE AMMO.RPT                          7164
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                   7386
    PAGE 669  ''PROGRAM  OLDER.VERSION                  9998
    PAGE 706                                            2096
ILLUM.INPUT
    PAGE 521  ROUTINE MAIN2                             4281 4283
    PAGE 576  ROUTINE ILLUM.INPUT                       6494
ILLUM.MAX.RANGE
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES           559
    PAGE  45  ''SECTION FOR DEFINITIONS                 2551
```

```
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9151
    PAGE 576  ROUTINE ILLUM.INPUT                    6516
    PAGE 669  ''PROGRAM  OLDER.VERSION                  0
    PAGE 703                                         1972
ILLUM.MUNITION
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES        556
    PAGE 576  ROUTINE ILLUM.INPUT                    6507 6509
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                7381
    PAGE 618  ROUTINE SNAP.R                         8229
    PAGE 669  ''PROGRAM  OLDER.VERSION               9997
    PAGE 712                                         2446
ILLUM.RADIUS
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES        558
    PAGE  45  ''SECTION FOR DEFINITIONS              2550
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9158
    PAGE 576  ROUTINE ILLUM.INPUT                    6512
    PAGE 669  ''PROGRAM  OLDER.VERSION               9999
    PAGE 703                                         1971
ILLUM.RND.WT
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES        561
    PAGE 576  ROUTINE ILLUM.INPUT                    6515
    PAGE 592  ROUTINE AMMO.RPT                       7165
    PAGE 669  ''PROGRAM  OLDER.VERSION                  2
ILLUM.RULE
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        796
    PAGE 200  ROUTINE ILLUM.EFFECTS                  9209 9218
    PAGE 370  EVENT ENGAGEMENT                       6763
    PAGE 576  ROUTINE ILLUM.INPUT                    6527
    PAGE 674  ''PROGRAM  OLDER.VERSION                238
ILLUM.SWITCH
    PAGE  42  ''SECTION FOR DEFINITIONS              2419
    PAGE 218  ROUTINE REQUEST.ILLUM                  9995
    PAGE 576  ROUTINE ILLUM.INPUT                    6499 6500
    PAGE 701  ''PROGRAM  OLDER.VERSION               1849
IMMEDIATELY.
    PAGE 412  EVENT ACT.ATK                          8611
    PAGE 416  EVENT ACT.MOVDIS                       8774
INC.
    PAGE 652  ROUTINE GAMMA.F                        9237
IND.
    PAGE 260  ROUTINE BTRY.EFFECTS                   2012
INFORMATION.
    PAGE 239  ROUTINE TARGET.ANALYSIS                 981
INF.C
    PAGE 103  ROUTINE MINE.DELAY                     4969
    PAGE 106  ROUTINE MIN.MOVE                       5071
    PAGE 138  ROUTINE CHECK.PROX                     6419
    PAGE 218  ROUTINE REQUEST.ILLUM                    20
    PAGE 222  ROUTINE REQUEST.SMOKE                   208
    PAGE 226  ROUTINE REQUEST.WD.FASCAM               423
    PAGE 227                                          479
    PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT          8812
    PAGE 420  PROCESS AC.ATK.TGT                     8936 8941
    PAGE 506  PROCESS CAS.MISSION                    3493
    PAGE 578  ROUTINE MINE.INPUT                     6614 6616
    PAGE 631  FUNCTION COMBINATIONS                  8737 8740
```

```
INITIATED.
    PAGE 347  EVENT ACT.REINF                          5817
INIT.PREPLAN.CAS
    PAGE  40  ''SECTION FOR EVENTS                     2285
    PAGE 380  EVENT INIT.PREPLAN.CAS                   7229 7256
    PAGE 585  ROUTINE TACAIR.INPUT                     6968
    PAGE 602  ROUTINE BETWEEN.ROUTINE                  7608 7609
    PAGE 616  ROUTINE SNAP2                            8182
    PAGE 699  ''PROGRAM  OLDER.VERSION                 1722
INIT.X.FEBA
    PAGE  41  ''SECTION FOR DEFINITIONS                2365
    PAGE  61  ROUTINE FEBA.INITIAL                     3203 3204
    PAGE 119  ROUTINE RESET.FEBA.SECTOR                5666
    PAGE 523  ROUTINE SYS.INPUT                        4357 4367 4367
    PAGE 700  ''PROGRAM  OLDER.VERSION                 1795
INIT.Y.FEBA
    PAGE  41  ''SECTION FOR DEFINITIONS                2366
    PAGE  61  ROUTINE FEBA.INITIAL                     3194 3214 3217
    PAGE 100  ROUTINE LOCATE.SECTOR                    4850
    PAGE 523  ROUTINE SYS.INPUT                        4358 4368 4368
    PAGE 700  ''PROGRAM  OLDER.VERSION                 1796
INT.F
    PAGE  63  ROUTINE FILE.FD.SCHD                     3270
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA               4808 4809 4811 4820
    PAGE 157  ROUTINE AO.DETECTION                     7286
    PAGE 165  ROUTINE CFR.DEGRADE                      7584
    PAGE 167  ROUTINE CFR.DETECTION                    7691
    PAGE 180  ROUTINE EST.MIL.WORTH                    8236
    PAGE 184  ROUTINE FA.BN.ASGN                       8424
    PAGE 194  ROUTINE FIND.START.TIME                  8911
    PAGE 206  ROUTINE PDB.DETECTION                    9457
    PAGE 213  ROUTINE PIR.DETECTION                    9783
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION          9815 9818
    PAGE 230  ROUTINE RPV.DETECTION                     614
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS                 1454
    PAGE 256  ROUTINE FO.DETECTION                     1793
    PAGE 238  ROUTINE BTRY.EFFECTS                     2464
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS               3574
    PAGE 291                                           3619 3622
    PAGE 358  EVENT CFR.ACTIVATION                     6368
    PAGE 362  EVENT CFR.OPERATOR                       6519 6521
    PAGE 383  EVENT OFF.LINE.ATTRITION                 7369
    PAGE 386  EVENT PDB.ACTIVATION                     7494
    PAGE 428  PROCESS AIR.OBSERVER                     9365 9373
    PAGE 430                                           9489 9491
    PAGE 473  PROCESS TARGET.REPORT                    1685 1686 1686 1687
    PAGE 475                                           1818
    PAGE 476                                           1848
    PAGE 491  PROCESS ASSESSMENT                       2688
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT                4671
    PAGE 607  ROUTINE KV.PRINT                         7785
    PAGE 653  PROCESS AIRBORNE.RADAR                   9280 9281 9304
    PAGE 656  ROUTINE AR.DETECTION                     9441
IPC.AC.TYPE
    PAGE  40  ''SECTION FOR EVENTS                     2287
    PAGE 623  ROUTINE TACAIR.DATA.REPORT               8451
```

```
    PAGE 699  ''PROGRAM  OLDER.VERSION                1724
IPC.NR.AC
    PAGE  40  ''SECTION FOR EVENTS                    2288
    PAGE 623  ROUTINE TACAIR.DATA.REPORT              8452
    PAGE 699  ''PROGRAM  OLDER.VERSION                1725
IPC.TARGET.UNIT
    PAGE  40  ''SECTION FOR EVENTS                    2286
    PAGE 623  ROUTINE TACAIR.DATA.REPORT              8450
    PAGE 699  ''PROGRAM  OLDER.VERSION                1723
IT.
    PAGE 118  ROUTINE REIN.ARRIVE                     5609
I.ACT.ATK
    PAGE  59  ROUTINE CREATE.FORCE                    3147
    PAGE 614  ROUTINE SNAP2                           8051
    PAGE 615                                          8108
    PAGE 616                                          8163
I.ACT.DEF
    PAGE 614  ROUTINE SNAP2                           8052
    PAGE 615                                          8109
    PAGE 616                                          8164
I.ACT.MOVCOR
    PAGE  58  ROUTINE CREATE.FORCE                    3141
    PAGE 614  ROUTINE SNAP2                           8053
    PAGE 615                                          8110
    PAGE 616                                          8165
I.ACT.MOVDIS
    PAGE 614  ROUTINE SNAP2                           8054
    PAGE 615                                          8111
    PAGE 616                                          8166
I.ACT.REINF
    PAGE 614  ROUTINE SNAP2                           8055
    PAGE 615                                          8112
    PAGE 616                                          8167
I.AC.ATK.TGT
    PAGE 427  PROCESS AC.ATK.TGT                      9317
    PAGE 490  PROCESS ASSESSMENT                      2607
    PAGE 614  ROUTINE SNAP2                           8030
    PAGE 615                                          8087
    PAGE 616                                          8141
I.AD.ENGAGEMENT
    PAGE 293  ROUTINE END.CAS.MISSION                 3700
    PAGE 614  ROUTINE SNAP2                           8056
    PAGE 615                                          8113
    PAGE 616                                          8168
I.AIRBORNE.RADAR
    PAGE 614  ROUTINE SNAP2                           8031
    PAGE 615                                          8088
    PAGE 616                                          8142
I.AIR.OBSERVER
    PAGE 614  ROUTINE SNAP2                           8032
    PAGE 615                                          8089
    PAGE 616                                          8143
I.ARTY.ASSESS
    PAGE 328  ROUTINE EMPTY                           5217
    PAGE 611  ROUTINE OUTPUT.ATTRITION                7971
    PAGE 614  ROUTINE SNAP2                           8033
```

```
    PAGE 615                                        8090
    PAGE 616                                        8144
I.ARTY.OCCUPATION
    PAGE 614  ROUTINE SNAP2                         8057
    PAGE 615                                        8114
    PAGE 616                                        8169
I.ASSESSMENT
    PAGE  81  ROUTINE BLOCK.LOS                     4080
    PAGE 108  ROUTINE NEW.SEGMENT                   5172
    PAGE 294  ROUTINE END.CAS.MISSION               3748
    PAGE 301  ROUTINE HC.DISENGAGE                  4098
    PAGE 328  ROUTINE EMPTY                         5197
    PAGE 425  PROCESS AC.ATK.TGT                    9247
    PAGE 426                                        9298
    PAGE 488  PROCESS ASSESSMENT                    2518
    PAGE 490                                        2637
    PAGE 614  ROUTINE SNAP2                         8034
    PAGE 615                                        8091
    PAGE 616                                        8145
I.BTL.ENDED
    PAGE 614  ROUTINE SNAP2                         8058
    PAGE 615                                        8115
    PAGE 616                                        8170
I.CAS.MISSION
    PAGE  58  ROUTINE CREATE.FORCE                  3125
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS         3437
    PAGE 293  ROUTINE END.CAS.MISSION               3705
    PAGE 350  EVENT AD.ENGAGEMENT                   6008
    PAGE 505  PROCESS CAS.MISSION                   3430 3433
    PAGE 614  ROUTINE SNAP2                         8035
    PAGE 615                                        8092
    PAGE 616                                        8146
I.CEP1.SQ
    PAGE 170  ROUTINE COMBINE.TRS                   7773 7775 7775 7776
I.CEP2.SQ
    PAGE 170  ROUTINE COMBINE.TRS                   7774 7775 7776 7776
I.CFR.ACTIVATION
    PAGE 614  ROUTINE SNAP2                         8059
    PAGE 615                                        8116
    PAGE 616                                        8171
I.CFR.OFF
    PAGE 161  ROUTINE ATTRIT.SENSOR                 7458
    PAGE 614  ROUTINE SNAP2                         8060
    PAGE 615                                        8117
    PAGE 616                                        8172
I.CFR.ON
    PAGE 161  ROUTINE ATTRIT.SENSOR                 7451
    PAGE 206  ROUTINE PDB.DETECTION                 9475
    PAGE 614  ROUTINE SNAP2                         8061
    PAGE 615                                        8118
    PAGE 616                                        8173
I.CFR.OPERATOR
    PAGE 161  ROUTINE ATTRIT.SENSOR                 7465
    PAGE 614  ROUTINE SNAP2                         8062
    PAGE 615                                        8119
    PAGE 616                                        8174
```

```
I.CHANGE.LITE
    PAGE 614  ROUTINE SNAP2                    8049
    PAGE 615                                   8106
    PAGE 616                                   8161
I.CHANGE.WEATHER
    PAGE 614  ROUTINE SNAP2                    8063
    PAGE 615                                   8120
I.DQ.OLD.SORTIE.QUEUE
    PAGE 614  ROUTINE SNAP2                    8064
    PAGE 615                                   8121
    PAGE 616                                   8176
I.END.SIMULATION
    PAGE 614  ROUTINE SNAP2                    8050
    PAGE 615                                   8107
    PAGE 616                                   8162
I.ENGAGEMENT
    PAGE  81  ROUTINE BLOCK.LOS                4070
    PAGE 107  ROUTINE NEW.SEGMENT              5155
    PAGE 108                                   5163
    PAGE 124  ROUTINE TIME.TO.DETECT           5852
    PAGE 328  ROUTINE EMPTY                    5190
    PAGE 369  EVENT ENGAGEMENT                 6699
    PAGE 614  ROUTINE SNAP2                    8065
    PAGE 615                                   8122
    PAGE 616                                   8177
I.EQ
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES      8491 8512 8514 8514 8515 8518 8519 8521 8523 8524 8524 8527 8528 8532 8534
                                               8537 8543 8545
    PAGE 625                                   8548 8554 8556 8559 8562 8564 8567 8573 8575 8578 8584 8586 8589 8599 8601
                                               8602 8603
    PAGE 626                                   8613 8615 8617 8617
I.FEBA.SORTIE
    PAGE 614  ROUTINE SNAP2                    8066
    PAGE 615                                   8123
    PAGE 616                                   8178
I.FIRE.MISSION
    PAGE 216  ROUTINE REQUEST.FASCAM           9915
    PAGE 220  ROUTINE REQUEST.ILLUM             113
    PAGE 224  ROUTINE REQUEST.SMOKE             321
    PAGE 614  ROUTINE SNAP2                    8036
    PAGE 615                                   8093
    PAGE 616                                   8147
I.FORWARD.OBSERVER
    PAGE 103  ROUTINE MINE.DELAY               4971
    PAGE 614  ROUTINE SNAP2                    8037
    PAGE 615                                   8094
    PAGE 616                                   8148
I.GET.NX.ORD
    PAGE 614  ROUTINE SNAP2                    8067
    PAGE 615                                   8124
    PAGE 616                                   8179
I.HC.ARRIVE.BATTLE
    PAGE 337  ROUTINE HC.EMPTY                 5500
    PAGE 402  EVENT START.BATTLE               8159
    PAGE 614  ROUTINE SNAP2                    8038
    PAGE 615                                   8095
```

```
    PAGE 616                                          8149
I.HC.DEPART.BATTLE
    PAGE 338  ROUTINE HC.EMPTY                        5568
    PAGE 614  ROUTINE SNAP2                           8068
    PAGE 615                                          8125
    PAGE 616                                          8180
I.HC.RETURN.FARRP
    PAGE 337  ROUTINE HC.EMPTY                        5474
    PAGE 614  ROUTINE SNAP2                           8039
    PAGE 615                                          8096
    PAGE 616                                          8150
I.HELICOPTER.FIRE
    PAGE 301  ROUTINE HC.DISENGAGE                    4074
    PAGE 312  ROUTINE INTER.HELO                      4617
    PAGE 338  ROUTINE HC.EMPTY                        5561
    PAGE 377  EVENT HELO.ENGAGEMENT                   7105
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION          1087
    PAGE 488  PROCESS ASSESSMENT                      2506
    PAGE 614  ROUTINE SNAP2                           8040
    PAGE 615                                          8097
    PAGE 616                                          8151
I.HELO.ENGAGEMENT
    PAGE 301  ROUTINE HC.DISENGAGE                    4066
    PAGE 312  ROUTINE INTER.HELO                      4609
    PAGE 338  ROUTINE HC.EMPTY                        5554
    PAGE 376  EVENT HELO.ENGAGEMENT                   7058
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION          1081
    PAGE 614  ROUTINE SNAP2                           8069
    PAGE 615                                          8126
    PAGE 616                                          8181
I.HEL.TARGET.ACQUISITION
    PAGE 311  ROUTINE INTER.HELO                      4546
    PAGE 338  ROUTINE HC.EMPTY                        5520
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                 130
    PAGE 614  ROUTINE SNAP2                           8041
    PAGE 615                                          8098
    PAGE 616                                          8153
I.HOW.REPAIR
    PAGE 614  ROUTINE SNAP2                           8042
    PAGE 615                                          8099
    PAGE 616                                          8154
I.INIT.PREPLAN.CAS
    PAGE 614  ROUTINE SNAP2                           8070
    PAGE 615                                          8127
    PAGE 616                                          8182
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8445
I.MINE.ASSESS
    PAGE 329  ROUTINE EMPTY                           5225
    PAGE 614  ROUTINE SNAP2                           8043
    PAGE 615                                          8100
I.MOVE
    PAGE 114  ROUTINE PREP.WITHDRAW                   5481
    PAGE 173  ROUTINE DUST.EFFECTS                    7909
    PAGE 174                                          7973 7988
    PAGE 235  ROUTINE SMOKE.EFFECTS                    797
    PAGE 236                                           897
```

```
    PAGE 261  ROUTINE BTRY.EFFECTS                2067
    PAGE 328  ROUTINE EMPTY                       5183
    PAGE 479  PROCESS WITH.DRAW                   2010
    PAGE 614  ROUTINE SNAP2                       8071
    PAGE 615                                      8128
    PAGE 616                                      8183
I.OFF.LINE.ATTRITION
    PAGE 615  ROUTINE SNAP2                       8083
    PAGE 616                                      8140
I.PDB.ACTIVATION
    PAGE 614  ROUTINE SNAP2                       8072
    PAGE 615                                      8129
    PAGE 616                                      8184
I.PDB.OPERATOR
    PAGE 160  ROUTINE ATTRIT.SENSOR               7423
    PAGE 614  ROUTINE SNAP2                       8073
    PAGE 615                                      8130
    PAGE 616                                      8185
I.PHOTO.IR.FLIGHT
    PAGE 614  ROUTINE SNAP2                       8044
    PAGE 615                                      8101
    PAGE 616                                      8155
I.POSITION.REPORT
    PAGE 615  ROUTINE SNAP2                       8081
    PAGE 616                                      8138
    PAGE 617                                      8196
I.REMOTE.PILOT.VEHICLE
    PAGE 614  ROUTINE SNAP2                       8045
    PAGE 615                                      8102
    PAGE 616                                      8157
I.SCHEDULE.ARTY.MOVEMENT
    PAGE 614  ROUTINE SNAP2                       8074
    PAGE 615                                      8131
    PAGE 616                                      8187
I.SEND.TEAM
    PAGE 337  ROUTINE HC.EMPTY                    5485
    PAGE 614  ROUTINE SNAP2                       8075
    PAGE 615                                      8132
    PAGE 616                                      8188
I.SET.DEBUG
    PAGE 615  ROUTINE SNAP2                       8082
    PAGE 616                                      8139
    PAGE 617                                      8197
I.SHOOT.OUT
    PAGE 301  ROUTINE HC.DISENGAGE                4110
    PAGE 426  PROCESS AC.ATK.TGT                  9293
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION      1092
    PAGE 488  PROCESS ASSESSMENT                  2513
    PAGE 490                                      2632
    PAGE 614  ROUTINE SNAP2                       8046
    PAGE 615                                      8103
    PAGE 616                                      8158
I.START.ARTY.MOVEMENT
    PAGE 393  EVENT START.ARTY.MOVEMENT           7667
    PAGE 614  ROUTINE SNAP2                       8076
    PAGE 615                                      8133
```

```
    PAGE 616                                          8190
I.START.BATTLE
    PAGE 614  ROUTINE SNAP2                           8077
    PAGE 615                                          8134
    PAGE 616                                          8193
I.START.MOVE
    PAGE 112  ROUTINE PREPARE.LIST                    5392
    PAGE 615  ROUTINE SNAP2                           8078 8135
    PAGE 617                                          8194
I.STOP.ARTY.MOVEMENT
    PAGE 104  ROUTINE MINE.DELAY                      5030 5036 5038
    PAGE 407  EVENT STOP.ARTY.MOVEMENT                8379
    PAGE 615  ROUTINE SNAP2                           8079
    PAGE 616                                          8136 8192
I.TARGET.REPORT
    PAGE 216  ROUTINE REQUEST.FASCAM                  9907
    PAGE 219  ROUTINE REQUEST.ILLUM                     92
    PAGE 224  ROUTINE REQUEST.SMOKE                    290
    PAGE 614  ROUTINE SNAP2                           8047
    PAGE 615                                          8104
    PAGE 616                                          8159
I.UPDATE.LOC
    PAGE 104  ROUTINE MINE.DELAY                      5046
    PAGE 105                                          5050 5052
    PAGE 112  ROUTINE PREPARE.LIST                    5385
    PAGE 615  ROUTINE SNAP2                           8080
    PAGE 616                                          8137
    PAGE 617                                          8195
    PAGE 630  FUNCTION COLLISION                      8684
I.WITH.DRAW
    PAGE 328  ROUTINE EMPTY                           5208
    PAGE 614  ROUTINE SNAP2                           8048
    PAGE 615                                          8105
    PAGE 616                                          8160
I.WPN
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES             8491 8511 8514 8515 8515 8520 8523 8523 8524 8540 8542
    PAGE 625                                          8548 8550 8570 8572 8578 8580 8597 8601 8602 8603
    PAGE 626                                          8611 8615 8616 8617
JR.
    PAGE 664  ''PROGRAM  OLDER.VERSION                9684
J.
    PAGE 664  ''PROGRAM  OLDER.VERSION                9669 9677 9678 9682
KAS.AD.UNIT
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES        1518
    PAGE  65  ROUTINE FILE.KAD.SENSOR                 3368 3379
    PAGE 317  ROUTINE FLIGHT.PATH                     4823
    PAGE 686  ''PROGRAM  OLDER.VERSION                 957
KEYED.SENSOR
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES        1507
    PAGE 160  ROUTINE ATTRIT.SENSOR                   7421
    PAGE 161                                          7447
    PAGE 567  ROUTINE SENSOR.INPUT                    6137
    PAGE 686  ''PROGRAM  OLDER.VERSION                 946
KEY.TIME
    PAGE 206  ROUTINE PDB.DETECTION                   9436 9474 9482
```

```
KILLED.
    PAGE 329  ROUTINE EMPTY                          5257
KILLER.SIDE
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        784
    PAGE 673  ''PROGRAM  OLDER.VERSION                226
KILLER.TB
    PAGE 435  PROCESS ARTY.ASSESS                    9697 9698
    PAGE 437                                         9842
KILLER.VICTIM
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        778  784  784  784
    PAGE 609  ROUTINE KV.SCOREBOARD                  7845 7863
    PAGE 673  ''PROGRAM  OLDER.VERSION                220  226  226  226
    PAGE 711                                         2437
KILL.TOTAL
    PAGE 606  ROUTINE KV.PRINT                       7699 7700 7705 7705 7731
    PAGE 607                                         7777
    PAGE 608                                         7832 7840
KNOWN.AD.SENSOR
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES       1517
    PAGE  65  ROUTINE FILE.KAD.SENSOR                3377
    PAGE 686  ''PROGRAM  OLDER.VERSION                956
KS.SENSOR.ID
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES       1509
    PAGE 161  ROUTINE ATTRIT.SENSOR                  7445
    PAGE 206  ROUTINE PDB.DETECTION                  9466
    PAGE 567  ROUTINE SENSOR.INPUT                   6153 6165 6172
    PAGE 686  ''PROGRAM  OLDER.VERSION                948
KS.TYPE.SENSOR
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES       1508
    PAGE 206  ROUTINE PDB.DETECTION                  9467
    PAGE 567  ROUTINE SENSOR.INPUT                   6139 6166 6169
    PAGE 686  ''PROGRAM  OLDER.VERSION                947
KV.AMMO.CONSUMED
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        780
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2818
    PAGE 280  ROUTINE AC.DF.EFFECTS                  3052
    PAGE 310  ROUTINE AD.SHOOT                       4472
    PAGE 384  EVENT OFF.LINE.ATTRITION               7419
    PAGE 499  PROCESS SHOOT.OUT                      3113
    PAGE 514  PROCESS HELICOPTER.FIRE                3925
    PAGE 529  ROUTINE KV.INPUT                       4530 4539 4548
    PAGE 593  ROUTINE AMMO.RPT                       7182 7199 7216
    PAGE 609  ROUTINE KV.SCOREBOARD                  7869 7875
    PAGE 673  ''PROGRAM  OLDER.VERSION                222
KV.CEM.WPN.NO
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        779
    PAGE 529  ROUTINE KV.INPUT                       4529 4538 4547 4564 4567
    PAGE 530  ROUTINE EQ.TE.INPUT                    4621 4622
    PAGE 607  ROUTINE KV.PRINT                       7758 7762
    PAGE 608                                         7809 7826
    PAGE 609  ROUTINE KV.SCOREBOARD                  7847 7848 7858 7858 7862 7866 7872
    PAGE 673  ''PROGRAM  OLDER.VERSION                221
KV.COUNTER
    PAGE 609  ROUTINE KV.SCOREBOARD                  7887 7888
    PAGE 610                                         7908 7909 7923 7924 7927
```

```
KV.EQ.ID
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          782
    PAGE 529  ROUTINE KV.INPUT                         4532 4541 4550
    PAGE 530  ROUTINE EQ.TE.INPUT                      4626
    PAGE 607  ROUTINE KV.PRINT                         7750 7762
    PAGE 608                                           7804 7821
    PAGE 609  ROUTINE KV.SCOREBOARD                    7867 7873
    PAGE 673  ''PROGRAM  OLDER.VERSION                  224
KV.INITIAL.DENSITY
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          781
    PAGE 529  ROUTINE KV.INPUT                         4531 4540 4549
    PAGE 535  ROUTINE UNIT.INPUT                       4803
    PAGE 536                                           4896 4899
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                  7310
    PAGE 606  ROUTINE KV.PRINT                         7730 7732 7742
    PAGE 607                                           7763 7799
    PAGE 608                                           7822
    PAGE 609  ROUTINE KV.SCOREBOARD                    7868 7874
    PAGE 673  ''PROGRAM  OLDER.VERSION                  223
KV.INPUT
    PAGE 520  ROUTINE MAIN2                            4202 4204
    PAGE 529  ROUTINE KV.INPUT                         4523
    PAGE 530  ROUTINE EQ.TE.INPUT                      4629
KV.SCORE
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          785
    PAGE 252  ROUTINE MINE.EFFECTS                     1580
    PAGE 270  ROUTINE BTRY.EFFECTS                     2572 2576
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                  2978
    PAGE 282  ROUTINE AC.DF.EFFECTS                    3153
    PAGE 309  ROUTINE AD.SHOOT                         4452
    PAGE 384  EVENT OFF.LINE.ATTRITION                 7407
    PAGE 435  PROCESS ARTY.ASSESS                      9742
    PAGE 436                                           9801
    PAGE 465  PROCESS MINE.ASSESS                      1296 1316
    PAGE 489  PROCESS ASSESSMENT                       2553
    PAGE 490                                           2612
    PAGE 492                                           2708
    PAGE 517  PROCESS HELICOPTER.FIRE                  4069
    PAGE 529  ROUTINE KV.INPUT                         4555 4557 4559
    PAGE 606  ROUTINE KV.PRINT                         7706
    PAGE 607                                           7769
    PAGE 608                                           7823
    PAGE 673  ''PROGRAM  OLDER.VERSION                  227
KV.SIDE
    PAGE 435  PROCESS ARTY.ASSESS                      9691 9699 9742
    PAGE 436                                           9801
KV.WPN.NAME
    PAGE  47  ''SECTION FOR DEFINITIONS                2673
    PAGE 706  ''PROGRAM  OLDER.VERSION                 2097
K.SIDE
    PAGE 382  EVENT OFF.LINE.ATTRITION                 7297 7299
    PAGE 383                                           7338 7345
    PAGE 384                                           7407 7410 7411 7420
K.WPN
    PAGE 382  EVENT OFF.LINE.ATTRITION                 7283 7284 7286 7328 7330 7333
    PAGE 383                                           7340 7350
```

```
    PAGE 384                                        7409
LAST.ARTY.ENGAGE
    PAGE 480  PROCESS FIRE.MISSION                  2053
    PAGE 481                                        2127 2131 2133
    PAGE 482                                        2143
LAST.ENG.TIME
    PAGE 183  ROUTINE FA.BN.ASGN                    8348
LAST.ORD.ATK
    PAGE 373  EVENT GET.NX.ORD                      6896 6925
    PAGE 374                                        6947
LAST.SENSOR
    PAGE 565  ROUTINE SENSOR.INPUT                  6042 6043 6048
LAST.UNIT
    PAGE 565  ROUTINE SENSOR.INPUT                  6020 6021 6025
LAUNCH.SITE
    PAGE 229  ROUTINE RPV.DETECTION                  531
    PAGE 230                                         594
LEAVE.TIME
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS            3485
    PAGE 291                                        3601 3603 3609 3611 3613
    PAGE 292                                        3652 3655 3657 3664 3666 3667
LEFT.BNDRY.INT
    PAGE  41  ''SECTION FOR DEFINITIONS             2362
    PAGE  61  ROUTINE FEBA.INITIAL                  3194 3198
    PAGE 100  ROUTINE LOCATE.SECTOR                 4838
    PAGE 700  ''PROGRAM  OLDER.VERSION              1792
LEG.LENGTH
    PAGE 428  PROCESS AIR.OBSERVER                  9343
    PAGE 429                                        9439 9441
    PAGE 430                                        9469 9493
    PAGE 432                                        9595 9596
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1366 1406
    PAGE 468                                        1423
    PAGE 653  PROCESS AIRBORNE.RADAR                9273 9305
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9482 9513 9514
    PAGE 659                                        9533
LEG.NUMBER
    PAGE 429  PROCESS AIR.OBSERVER                  9429
    PAGE 433                                        9654 9654
LEG.REMAINING
    PAGE 432  PROCESS AIR.OBSERVER                  9596 9597
LEG.SLOPE
    PAGE 343  ROUTINE SEARCH.COVERAGE               5681 5686 5687 5688 5688 5689 5690 5690 5691 5692
    PAGE 428  PROCESS AIR.OBSERVER                  9343
    PAGE 430                                        9467 9468 9481 9482 9482 9483
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1366 1404 1405 1411 1412 1412 1413
    PAGE 468                                        1426
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9482 9511 9512 9524 9525 9525 9526
    PAGE 659                                        9536
LEG.TIME.REMAINING
    PAGE 428  PROCESS AIR.OBSERVER                  9347
    PAGE 432                                        9597 9598 9619
LEG.Y.INTERCEPT
    PAGE 343  ROUTINE SEARCH.COVERAGE               5682 5688 5690 5691 5692
    PAGE 428  PROCESS AIR.OBSERVER                  9343
    PAGE 430                                        9468 9482 9483
```

```
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1366 1405 1412 1413
    PAGE 468                                        1427
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9482 9512 9525 9526
    PAGE 659                                        9537
LENGTH.ADJ.PAT
    PAGE 191  ROUTINE FINAL.COVERAGE                8734 8736 8739
LENGTH.DAM
    PAGE 191  ROUTINE FINAL.COVERAGE                8739 8741 8745 8747 8759
    PAGE 192                                        8784 8785 8832 8833
    PAGE 193                                        8857 8864
LENGTH.TGT
    PAGE 178  ROUTINE EST.COVERAGE                  8149
    PAGE 179                                        8191 8194 8197 8210
    PAGE 190  ROUTINE FINAL.COVERAGE                8698
    PAGE 191                                        8759
    PAGE 192                                        8786 8787 8801 8825 8832 8833
    PAGE 193                                        8838 8856
LENGTH.VOL
    PAGE 191  ROUTINE FINAL.COVERAGE                8737 8739
LENGTH.1RND.PAT
    PAGE 178  ROUTINE EST.COVERAGE                  8152 8163 8168 8169
    PAGE 190  ROUTINE FINAL.COVERAGE                8702 8705 8713 8714
    PAGE 191                                        8722 8723 8734
    PAGE 192                                        8824
    PAGE 193                                        8855
LETHAL.AREA
    PAGE 178  ROUTINE EST.COVERAGE                  8162 8163 8165
    PAGE 190  ROUTINE FINAL.COVERAGE                8707 8709 8713 8718 8720
    PAGE 191                                        8722 8724
LINES.V
    PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT         8812
LOCATE.SECTOR
    PAGE   4  PROGRAM REVISIONS                      187  199
    PAGE  61  ROUTINE FEBA.INITIAL                  3225
    PAGE  89  ROUTINE FA.BN.MOVEMENT                4436
    PAGE  91                                        4516
    PAGE 100  ROUTINE LOCATE.SECTOR                 4831 4867
    PAGE 119  ROUTINE RESET.FEBA.SECTOR             5681
    PAGE 239  ROUTINE TARGET.ANALYSIS               1000
    PAGE 246  ROUTINE VOLLEY                        1317
    PAGE 260  ROUTINE BTRY.EFFECTS                  2020
    PAGE 325  ROUTINE DEQ.FEBA.SET                  5089 5095
    PAGE 331  ROUTINE ENQ.FEBA.SET                  5312
    PAGE 347  EVENT ACT.REINF                       5840
    PAGE 360  EVENT CFR.ON                          6429
    PAGE 371  EVENT FEBA.SORTIE                     6797
    PAGE 372                                        6840
    PAGE 634  FUNCTION FEBA.BAND                    8792 8810
LOCATION.
    PAGE 182  ROUTINE FA.BN.ASGN                    8324
LOC.UPDATE.FREQ
    PAGE  42  ''SECTION FOR DEFINITIONS             2398
    PAGE 406  EVENT START.MOVE                      8365
    PAGE 411  EVENT UPDATE.LOC                      8593
    PAGE 523  ROUTINE SYS.INPUT                     4363
    PAGE 701  ''PROGRAM  OLDER.VERSION              1828
```

```
LOG.E.F
    PAGE 192  ROUTINE FINAL.COVERAGE                8817
    PAGE 231  ROUTINE SIZE.ESTIMATE                  671  673  674  675  676
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS              1446
    PAGE 341  ROUTINE PROB.TIME                     5656
    PAGE 631  FUNCTION COMBINATIONS                 8742 8742
    PAGE 640  ROUTINE EXPONENTIAL.F                 9037
    PAGE 641  ROUTINE NORMAL.F                      9059
    PAGE 642  ROUTINE WEIBULL.F                     9070
    PAGE 652  ROUTINE GAMMA.F                       9259
LOG.10.F
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION       5755 5756
LOITER.TIME
    PAGE 128  ROUTINE BTL.CHECK                     6017 6019 6036 6038
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS            3535 3547 3585
    PAGE 291                                        3603 3631
    PAGE 292                                        3655 3657
    PAGE 297  ROUTINE FARRP.CHECK                   3898 3899 3904 3906
    PAGE 298  ROUTINE HC.COMPUTE.TIMES              3921
    PAGE 299                                        3978 3979 3980 3982 3993 3997
    PAGE 443  PROCESS HC.ARRIVE.BATTLE               121
    PAGE 450  PROCESS HC.RETURN.FARRP                499  500
    PAGE 453                                         670  672  675
LOOK.
    PAGE 310  ROUTINE AD.SHOOT                      4500
LOOK.SHOOT.LOOK
    PAGE  52  ''SECTION FOR SUBSTITUTIONS           3000
    PAGE 711  ''PROGRAM  OLDER.VERSION              2407
LOSSES.
    PAGE 508  PROCESS CAS.MISSION                   3625
LOS.BAND
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES       848  851
    PAGE  96  ROUTINE LINE.OF.SIGHT                 4704 4705 4711
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION         814  815  825
    PAGE 571  ROUTINE TT.FACTORS.INPUT              6290 6292 6295 6299 6303
    PAGE 674  ''PROGRAM  OLDER.VERSION               290
    PAGE 675                                         293
    PAGE 712                                        2471
LOS.PROB
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES       852
    PAGE  96  ROUTINE LINE.OF.SIGHT                 4710 4711
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION         825
    PAGE 571  ROUTINE TT.FACTORS.INPUT              6303
    PAGE 675  ''PROGRAM  OLDER.VERSION               294
LOW.FRAC.RANGE
    PAGE 443  PROCESS HC.ARRIVE.BATTLE               109
    PAGE 447                                         357  361  366
LOW.MIL.WORTH
    PAGE 208  ROUTINE PGM.MSN.ASGN                  9509 9511
    PAGE 209                                        9562 9566 9567 9582
L.
    PAGE 664  ''PROGRAM  OLDER.VERSION              9672 9675
L.AATT.LIST
    PAGE  34  ''SECTION FOR PROCESSES               1903
    PAGE 693  ''PROGRAM  OLDER.VERSION              1341
```

```
L.AA.SET
    PAGE  34  ''SECTION FOR PROCESSES                1951
    PAGE 693  ''PROGRAM  OLDER.VERSION               1389
L.AO.CAND.DET.LIST
    PAGE  34  ''SECTION FOR PROCESSES                1936
    PAGE 693  ''PROGRAM  OLDER.VERSION               1374
L.AO.DET.TGT.LIST
    PAGE  34  ''SECTION FOR PROCESSES                1938
    PAGE 693  ''PROGRAM  OLDER.VERSION               1376
L.AO.FLIGHT.LEG.LIST
    PAGE  34  ''SECTION FOR PROCESSES                1934
    PAGE 693  ''PROGRAM  OLDER.VERSION               1372
L.AR.CAND.DET.LIST
    PAGE  34  ''SECTION FOR PROCESSES                1918
    PAGE 693  ''PROGRAM  OLDER.VERSION               1356
L.AR.DET.TGT.LIST
    PAGE  34  ''SECTION FOR PROCESSES                1916
    PAGE 693  ''PROGRAM  OLDER.VERSION               1354
L.AU.LIST
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES        578
    PAGE 670  ''PROGRAM  OLDER.VERSION                 19
L.AVAIL.AO.LIST
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        766
    PAGE 673  ''PROGRAM  OLDER.VERSION                208
L.BN.BTRY.SET
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES        435
    PAGE 667  ''PROGRAM  OLDER.VERSION               9876
L.BTL.FORCE.SET
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES       1177
    PAGE 680  ''PROGRAM  OLDER.VERSION                616
L.BY.FM.QUEUE
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        344
    PAGE 666  ''PROGRAM  OLDER.VERSION               9786
L.BY.HOW.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        340
    PAGE 666  ''PROGRAM  OLDER.VERSION               9782
L.BY.SCHD.LIST
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        342
    PAGE 666  ''PROGRAM  OLDER.VERSION               9784
L.CFPS.LIST
    PAGE  35  ''SECTION FOR PROCESSES                1988
    PAGE 694  ''PROGRAM  OLDER.VERSION               1426
L.CF.OP.Q
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES       1237
    PAGE 681  ''PROGRAM  OLDER.VERSION                676
L.CT.TU.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        362
    PAGE 666  ''PROGRAM  OLDER.VERSION               9804
L.FD.BN.LIST
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES        459
    PAGE 668  ''PROGRAM  OLDER.VERSION               9900
L.FD.COMPLETE.LIST
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES        455
    PAGE 668  ''PROGRAM  OLDER.VERSION               9896
L.FD.SCHD.LIST
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES        457
```

```
    PAGE 668  ''PROGRAM  OLDER.VERSION                9898
L.FD.TR.QUEUE
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES         453
    PAGE 668  ''PROGRAM  OLDER.VERSION                9894
L.FO.CAND.DET.LIST
    PAGE  36  ''SECTION FOR PROCESSES                 2045
    PAGE 695  ''PROGRAM  OLDER.VERSION                1483
L.FO.CUR.FM.LIST
    PAGE  36  ''SECTION FOR PROCESSES                 2049
    PAGE 695  ''PROGRAM  OLDER.VERSION                1487
L.FO.TGT.RPT.LIST
    PAGE  36  ''SECTION FOR PROCESSES                 2047
    PAGE 695  ''PROGRAM  OLDER.VERSION                1485
L.FP.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES         759
    PAGE 673  ''PROGRAM  OLDER.VERSION                 201
L.FR.UNIT.SET
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES        1417
    PAGE 684  ''PROGRAM  OLDER.VERSION                 856
L.GP.CAT.SET
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES         488
    PAGE 668  ''PROGRAM  OLDER.VERSION                9929
L.HE.TB.RH.LIST
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES         505
    PAGE 636  FUNCTION HE.WLA                         8879
    PAGE 669  ''PROGRAM  OLDER.VERSION                9946
L.HF.SO.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES        1834
    PAGE 691  ''PROGRAM  OLDER.VERSION                1273
L.HT.LIST
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES        1341
    PAGE 683  ''PROGRAM  OLDER.VERSION                 780
L.HT.MEMBER.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES        1453
    PAGE 685  ''PROGRAM  OLDER.VERSION                 892
L.HT.TARGET.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES        1456
    PAGE 685  ''PROGRAM  OLDER.VERSION                 895
L.IC.TB.RH.LIST
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         542
    PAGE 638  FUNCTION ICM.WLA                        8973
    PAGE 669  ''PROGRAM  OLDER.VERSION                9983
L.MADS.RH.SET
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         604
    PAGE  65  ROUTINE FILE.KAD.SENSOR                 3381
    PAGE 316  ROUTINE FLIGHT.PATH                     4794
    PAGE 506  PROCESS CAS.MISSION                     3506
    PAGE 670  ''PROGRAM  OLDER.VERSION                  45
L.MA.SET
    PAGE  37  ''SECTION FOR PROCESSES                 2096
    PAGE 696  ''PROGRAM  OLDER.VERSION                1534
L.MCFR.RH.LIST
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES         642
    PAGE 165  ROUTINE CFR.DEGRADE                     7587 7590
    PAGE 358  EVENT CFR.ACTIVATION                    6372
    PAGE 671  ''PROGRAM  OLDER.VERSION                  83
```

```
L.MFO.RB.SET
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES          654
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA               4797
    PAGE 254  ROUTINE FO.DETECTION                     1710
    PAGE 671  ''PROGRAM  OLDER.VERSION                   95
L.MFP.LIST
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          575
    PAGE 670  ''PROGRAM  OLDER.VERSION                   16
L.MO.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES         1064
    PAGE 678  ''PROGRAM  OLDER.VERSION                  506
L.MPDB.RH.LIST
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES          668
    PAGE 386  EVENT PDB.ACTIVATION                     7497
    PAGE 671  ''PROGRAM  OLDER.VERSION                  111
L.MU.ORDER.SET
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES         1554
    PAGE  95  ROUTINE INIT.REINF                       4654
    PAGE 144  ROUTINE DEAD.UNIT                        6689
    PAGE 687  ''PROGRAM  OLDER.VERSION                  993
L.MU.TF.LIST
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES         1552
    PAGE 687  ''PROGRAM  OLDER.VERSION                  991
L.PATH.SET
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES         1802
    PAGE  72  ROUTINE ORIENTATION                      3704 3705 3714 3715
    PAGE 691  ''PROGRAM  OLDER.VERSION                 1241
L.PDB.KEYED.LIST
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES         1641
    PAGE 688  ''PROGRAM  OLDER.VERSION                 1080
L.PDB.OP.Q
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES         1644
    PAGE 688  ''PROGRAM  OLDER.VERSION                 1083
L.PIR.FLIGHT.LEG.LIST
    PAGE  37  ''SECTION FOR PROCESSES                  2108
    PAGE 696  ''PROGRAM  OLDER.VERSION                 1546
L.PIR.RECORD.LIST
    PAGE  37  ''SECTION FOR PROCESSES                  2110
    PAGE 696  ''PROGRAM  OLDER.VERSION                 1548
L.PIR.RTD.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES         1683
    PAGE 689  ''PROGRAM  OLDER.VERSION                 1122
L.RPV.CAND.DET.LIST
    PAGE  37  ''SECTION FOR PROCESSES                  2124
    PAGE 696  ''PROGRAM  OLDER.VERSION                 1562
L.RPV.FLIGHT.LEG.LIST
    PAGE  37  ''SECTION FOR PROCESSES                  2127
    PAGE 696  ''PROGRAM  OLDER.VERSION                 1565
L.SIDE.CFR.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          764
    PAGE 673  ''PROGRAM  OLDER.VERSION                  206
L.SIDE.PDB.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          762
    PAGE 673  ''PROGRAM  OLDER.VERSION                  204
L.SI.LIST
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES         1205
```

```
    PAGE 681  ''PROGRAM  OLDER.VERSION                644
L.SO.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1828
    PAGE 691  ''PROGRAM  OLDER.VERSION               1267
L.SS.SET
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        810
    PAGE  90  ROUTINE FA.BN.MOVEMENT                 4453
    PAGE  91                                         4533
    PAGE 239  ROUTINE TARGET.ANALYSIS                1023
    PAGE 246  ROUTINE VOLLEY                         1337
    PAGE 331  ROUTINE ENQ.FEBA.SET                   5327
    PAGE 360  EVENT CFR.ON                           6450
    PAGE 371  EVENT FEBA.SORTIE                      6815
    PAGE 372                                         6856
    PAGE 634  FUNCTION FEBA.BAND                     8805 8823
    PAGE 674  ''PROGRAM  OLDER.VERSION                252
L.TB.SORT.LIST
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        873
    PAGE 675  ''PROGRAM  OLDER.VERSION                315
L.TB.TM.LIST
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        905
    PAGE 675  ''PROGRAM  OLDER.VERSION                347
L.TEAM.TYPES
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        876
    PAGE 675  ''PROGRAM  OLDER.VERSION                318
L.TGT
    PAGE 192  ROUTINE FINAL.COVERAGE                 8829
    PAGE 193                                         8861
L.TR.DET.LIST
    PAGE  38  ''SECTION FOR PROCESSES                2182
    PAGE 697  ''PROGRAM  OLDER.VERSION               1620
L.TR.FM.LIST
    PAGE  38  ''SECTION FOR PROCESSES                2180
    PAGE 697  ''PROGRAM  OLDER.VERSION               1618
L.TU.NTE.SET
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES        949
    PAGE 676  ''PROGRAM  OLDER.VERSION                391
L.TU.TE.LIST
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES        947
    PAGE 676  ''PROGRAM  OLDER.VERSION                389
L.UE.TARGET.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1825
    PAGE 497  PROCESS SHOOT.OUT                      2984 3001
    PAGE 691  ''PROGRAM  OLDER.VERSION               1264
L.UE.WEAPON.SET
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1831
    PAGE 691  ''PROGRAM  OLDER.VERSION               1270
L.UNIT.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        775
    PAGE 673  ''PROGRAM  OLDER.VERSION                217
L.UN.EQUIP.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES       1021
    PAGE 677  ''PROGRAM  OLDER.VERSION                463
L.UN.HC.LOS.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES       1060
    PAGE 678  ''PROGRAM  OLDER.VERSION                502
```

```
L.UN.LOS.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          1029
    PAGE 678  ''PROGRAM  OLDER.VERSION                   471
L.UN.PATH
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          1025
    PAGE 399  EVENT START.BATTLE                        8000 8001
    PAGE 678  ''PROGRAM  OLDER.VERSION                   467
L.UN.SEGMENT.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          1027
    PAGE 678  ''PROGRAM  OLDER.VERSION                   469
L.UN.SENSOR.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1031
    PAGE 678  ''PROGRAM  OLDER.VERSION                   473
L.UN.SUB.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          1023
    PAGE 677  ''PROGRAM  OLDER.VERSION                   465
L.1
    PAGE   2  PROGRAM REVISIONS                           74
L.1RND
    PAGE 192  ROUTINE FINAL.COVERAGE                    8828
    PAGE 193                                            8860
MADS.DELAY.TIME
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES           591
    PAGE 351  EVENT AD.ENGAGEMENT                       6039 6046
    PAGE 422  PROCESS AC.ATK.TGT                        9044
    PAGE 587  ROUTINE MADS.INPUT                        6989
    PAGE 623  ROUTINE TACAIR.DATA.REPORT                8472
    PAGE 670  ''PROGRAM  OLDER.VERSION                    32
MADS.DETECT
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES           597
    PAGE 317  ROUTINE FLIGHT.PATH                       4841
    PAGE 318                                            4876 4904
    PAGE 350  EVENT AD.ENGAGEMENT                       5992
    PAGE 670  ''PROGRAM  OLDER.VERSION                    38
MADS.FCM
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES           593
    PAGE  52  ''SECTION FOR SUBSTITUTIONS               2999
    PAGE 308  ROUTINE AD.SHOOT                          4408
    PAGE 310                                            4497 4497
    PAGE 587  ROUTINE MADS.INPUT                        6991
    PAGE 623  ROUTINE TACAIR.DATA.REPORT                8474
    PAGE 670  ''PROGRAM  OLDER.VERSION                    34
    PAGE 711                                            2406
MADS.INPUT
    PAGE 522  ROUTINE MAIN2                             4300 4302
    PAGE 587  ROUTINE MADS.INPUT                        6978
MADS.NAME
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES           590
    PAGE  47  ''SECTION FOR DEFINITIONS                 2660
    PAGE 316  ROUTINE FLIGHT.PATH                       4792
    PAGE 350  EVENT AD.ENGAGEMENT                       5974 5997
    PAGE 587  ROUTINE MADS.INPUT                        6988
    PAGE 623  ROUTINE TACAIR.DATA.REPORT                8471
    PAGE 670  ''PROGRAM  OLDER.VERSION                    31
    PAGE 705                                            2084
```

```
MADS.PW.DEGRADE
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          592
    PAGE 350  EVENT AD.ENGAGEMENT                      6003
    PAGE 587  ROUTINE MADS.INPUT                       6990
    PAGE 623  ROUTINE TACAIR.DATA.REPORT               8473
    PAGE 670  ''PROGRAM  OLDER.VERSION                   33
MADS.RDY.RDS
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          598
    PAGE 670  ''PROGRAM  OLDER.VERSION                   39
MADS.RESUP.TIME
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          599
    PAGE 670  ''PROGRAM  OLDER.VERSION                   40
MADS.RH
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES         1526
    PAGE 587  ROUTINE MADS.INPUT                       6998
    PAGE 686  ''PROGRAM  OLDER.VERSION                  965
MADS.RH.SET
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          601
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES         1531
    PAGE  49  ''SECTION FOR DEFINITIONS                2817
    PAGE 350  EVENT AD.ENGAGEMENT                      5975
    PAGE 587  ROUTINE MADS.INPUT                       7004
    PAGE 623  ROUTINE TACAIR.DATA.REPORT               8480
    PAGE 670  ''PROGRAM  OLDER.VERSION                   42
    PAGE 686                                            970
    PAGE 708                                           2238
MADS.RIPL
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          594
    PAGE 310  ROUTINE AD.SHOOT                         4498
    PAGE 587  ROUTINE MADS.INPUT                       6992
    PAGE 623  ROUTINE TACAIR.DATA.REPORT               8475
    PAGE 670  ''PROGRAM  OLDER.VERSION                   35
MADS.WPN.RELOAD.TIME
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          595
    PAGE 310  ROUTINE AD.SHOOT                         4493
    PAGE 587  ROUTINE MADS.INPUT                       6993
    PAGE 623  ROUTINE TACAIR.DATA.REPORT               8476
    PAGE 670   'PROGRAM  OLDER.VERSION                   36
MADS.XMIT.PCT
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          596
    PAGE 350  EVENT AD.ENGAGEMENT                      6001
    PAGE 587  ROUTINE MADS.INPUT                       6994
    PAGE 623  ROUTINE TACAIR.DATA.REPORT               8477
    PAGE 670  ''PROGRAM  OLDER.VERSION                   37
MANUEVER.SUPPORT
    PAGE 239  ROUTINE TARGET.ANALYSIS                   987
    PAGE 241                                           1099
    PAGE 254  ROUTINE FO.DETECTION                     1728
    PAGE 257                                           1882
MAN.UNIT
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES         1537
    PAGE  87  ROUTINE END.MOVE                         4319 4321 4325 4328 4329 4330 4331 4332 4335
    PAGE 137  ROUTINE CHECK.LIST                       6342 6363 6370
    PAGE 141  ROUTINE CHECK.STREN                      6511 6514 6516 6517
    PAGE 143  ROUTINE DEAD.UNIT                        6592 6593 6594
    PAGE 403  EVENT START.MOVE                         8195 8233
```

```
    PAGE 404                                          8237 8242
    PAGE 405                                          8322 8324
    PAGE 408  EVENT UPDATE.LOC                        8459 8461 8461
    PAGE 409                                          8462 8462 8465 8470
    PAGE 412  EVENT ACT.ATK                           8614 8636 8639
    PAGE 413                                          8661 8664
    PAGE 414  EVENT ACT.DEF                           8724 8728 8730 8731
    PAGE 415  EVENT ACT.MOVCOR                        8748 8751 8753
    PAGE 534  ROUTINE UNIT.INPUT                      4738 4739
    PAGE 686  ''PROGRAM  OLDER.VERSION                 976
    PAGE 712                                          2491
MAN.UNT
    PAGE 142  ROUTINE DEAD.UNIT                       6537 6538 6542 6547 6548 6554 6560 6562 6563 6565 6576
    PAGE 143                                          6579 6590 6592 6597 6598 6600
    PAGE 144                                          6645 6647 6650 6654 6669 6683 6685 6689
    PAGE 373  EVENT GET.NX.ORD                        6890 6891 6892 6906 6910
MAOR.CIR.ERROR
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         619
    PAGE 157  ROUTINE AO.DETECTION                    7272
    PAGE 564  ROUTINE MAO.INPUT                       6006
    PAGE 670  ''PROGRAM  OLDER.VERSION                  60
MAO.ALTITUDE
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         610
    PAGE 428  PROCESS AIR.OBSERVER                    9373
    PAGE 564  ROUTINE MAO.INPUT                       5970
    PAGE 670  ''PROGRAM  OLDER.VERSION                  51
MAO.EQ.ID
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         616
    PAGE 564  ROUTINE MAO.INPUT                       5968
    PAGE 566  ROUTINE SENSOR.INPUT                    6084
    PAGE 670  ''PROGRAM  OLDER.VERSION                  57
MAO.INPUT
    PAGE 521  ROUTINE MAIN2                           4260 4262
    PAGE 564  ROUTINE MAO.INPUT                       5953
MAO.MAGNIFICATION
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         611
    PAGE 156  ROUTINE AO.DETECTION                    7230 7234 7237
    PAGE 429  PROCESS AIR.OBSERVER                    9413
    PAGE 564  ROUTINE MAO.INPUT                       5971
    PAGE 670  ''PROGRAM  OLDER.VERSION                  52
MAO.MAX.ALOFT.TIME
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         614
    PAGE 428  PROCESS AIR.OBSERVER                    9379
    PAGE 564  ROUTINE MAO.INPUT                       5972
    PAGE 670  ''PROGRAM  OLDER.VERSION                  55
MAO.MAX.PREP
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         613
    PAGE  48  ''SECTION FOR DEFINITIONS               2743
    PAGE 429  PROCESS AIR.OBSERVER                    9397
    PAGE 564  ROUTINE MAO.INPUT                       5974
    PAGE 670  ''PROGRAM  OLDER.VERSION                  54
    PAGE 707                                          2165
MAO.MIN.PREP
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         612
    PAGE  48  ''SECTION FOR DEFINITIONS               2745
    PAGE 429  PROCESS AIR.OBSERVER                    9397
```

```
    PAGE 564  ROUTINE MAO.INPUT                    5973
    PAGE 670  ''PROGRAM  OLDER.VERSION               53
    PAGE 707                                       2167
MAO.NAME
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES       608
    PAGE  47  ''SECTION FOR DEFINITIONS             2674
    PAGE 564  ROUTINE MAO.INPUT                     5967
    PAGE 670  ''PROGRAM  OLDER.VERSION                49
    PAGE 706                                        2098
MAO.PGM.CAP
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES       615
    PAGE  45  ''SECTION FOR DEFINITIONS             2553
    PAGE 157  ROUTINE AO.DETECTION                  7299
    PAGE 564  ROUTINE MAO.INPUT                     5975
    PAGE 670  ''PROGRAM  OLDER.VERSION                56
    PAGE 703                                        1974
MAO.VELOCITY
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES       609
    PAGE 428  PROCESS AIR.OBSERVER                  9364 9365
    PAGE 564  ROUTINE MAO.INPUT                     5969
    PAGE 670  ''PROGRAM  OLDER.VERSION                50
MARCH.ORDER
    PAGE  89  ROUTINE FA.BN.MOVEMENT                4402
    PAGE  90                                        4475 4493
    PAGE  91                                        4552 4566
    PAGE  92                                        4583
MARGINAL.EFFECTS
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ          9313
    PAGE 204                                        9372 9373 9378
MARGINAL.EFFECTS.ADJ
    PAGE 198  ROUTINE HE.OR.ICM.COMPUTATION         9122
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ          9300 9316
MAR.CEP
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES       629
    PAGE 671  ''PROGRAM  OLDER.VERSION                70
MAR.MAX.ALOFT.TIME
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES       626
    PAGE  45  ''SECTION FOR DEFINITIONS             2564
    PAGE 653  PROCESS AIRBORNE.RADAR                9299
    PAGE 671  ''PROGRAM  OLDER.VERSION                67
    PAGE 704                                        1985
MAR.MAX.PREP
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES       625
    PAGE  45  ''SECTION FOR DEFINITIONS             2555
    PAGE 653  PROCESS AIRBORNE.RADAR                9293
    PAGE 671  ''PROGRAM  OLDER.VERSION                66
    PAGE 704                                        1976
MAR.MAX.SEARCH.RNG
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES       627
    PAGE 653  PROCESS AIRBORNE.RADAR                9281
    PAGE 671  ''PROGRAM  OLDER.VERSION                68
MAR.MIN.PREP
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES       624
    PAGE  45  ''SECTION FOR DEFINITIONS             2554
    PAGE 653  PROCESS AIRBORNE.RADAR                9293
    PAGE 671  ''PROGRAM  OLDER.VERSION                65
```

```
    PAGE 704                                          1975
MAR.MIN.SEARCH.RNG
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES         628
    PAGE 653  PROCESS AIRBORNE.RADAR                  9280
    PAGE 671  ''PROGRAM  OLDER.VERSION                  69
MAR.NAME
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         622
    PAGE  47  ''SECTION FOR DEFINITIONS               2675
    PAGE 671  ''PROGRAM  OLDER.VERSION                  63
    PAGE 706                                          2099
MAR.VELOCITY
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         623
    PAGE 653  PROCESS AIRBORNE.RADAR                  9279
    PAGE 657  FUNCTION AR.PROB.DETECT                 9450 9453 9456
    PAGE 671  ''PROGRAM  OLDER.VERSION                  64
MAX.APPAR.SIZE
    PAGE 154  ROUTINE AO.DETECTION                    7092
    PAGE 156                                          7233 7236 7239 7243
MAX.APP.SIZE
    PAGE 156  ROUTINE AO.DETECTION                    7244
MAX.ATT.FASCAM
    PAGE  42  ''SECTION FOR DEFINITIONS               2422
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM              9861
    PAGE 226  ROUTINE REQUEST.WD.FASCAM                430
    PAGE 577  ROUTINE MINE.INPUT                      6554
    PAGE 701  ''PROGRAM  OLDER.VERSION                1852
MAX.BAT
    PAGE  90  ROUTINE FA.BN.MOVEMENT                  4456 4459 4482 4482 4483 4484 4484 4485 4486 4489 4489 4493 4494 4496 4497
                                                      4497
MAX.CPE.INT
    PAGE 176  ROUTINE EST.COVERAGE                    8017 8046
    PAGE 177                                          8065 8067 8069 8070 8102
    PAGE 178                                          8121 8123 8125 8126
    PAGE 188  ROUTINE FINAL.COVERAGE                  8558 8589
    PAGE 189                                          8611 8613 8618 8619 8646
    PAGE 190                                          8668 8670 8675 8676
MAX.DAT
    PAGE   2  PROGRAM REVISIONS                         77
MAX.DBANK.V
    PAGE  56  ROUTINE MAIN1                           3042
    PAGE 618  ROUTINE SNAP.R                          8207
MAX.DIST
    PAGE  90  ROUTINE FA.BN.MOVEMENT                  4456 4482 4483
    PAGE 154  ROUTINE AO.DETECTION                    7082 7094
    PAGE 155                                          7132
    PAGE 428  PROCESS AIR.OBSERVER                    9350 9352
    PAGE 429                                          9394 9405 9416 9419 9426
    PAGE 431                                          9531 9550 9556
    PAGE 432                                          9582 9614
    PAGE 433                                          9632 9650 9669 9672
MAX.EFFECTS
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ            9313
    PAGE 204                                          9360 9372 9376 9383
MAX.F
    PAGE  65  ROUTINE FILE.KAD.SENSOR                 3406 3408
    PAGE  69  ROUTINE GENERAL.BATTLE                  3586
```

```
    PAGE  78  ROUTINE ADJUST                          3979 3981
    PAGE  80  ROUTINE BLOCK.LOS                       4032 4034 4047 4049
    PAGE  97  ROUTINE LINE.OF.SIGHT                   4740 4759
    PAGE 104  ROUTINE MINE.DELAY                      5023
    PAGE 106  ROUTINE MIN.MOVE                        5092
    PAGE 109  ROUTINE NEW.SEGMENT                     5258
    PAGE 134  ROUTINE CHECK.FOR.MINES                 6230 6232
    PAGE 135                                          6310 6314
    PAGE 168  ROUTINE CHK.COMP.TR                     7715
    PAGE 170  ROUTINE COMBINE.TRS                     7787
    PAGE 273  ROUTINE BTRY.EFFECTS                    2723
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                 2832
    PAGE 305  ROUTINE UNIT.PRIORITY                   4265
    PAGE 316  ROUTINE FLIGHT.PATH                     4784
    PAGE 341  ROUTINE PROB.TIME                       5657
    PAGE 349  EVENT AD.ENGAGEMENT                     5923 5925
    PAGE 351                                          6043 6045
    PAGE 422  PROCESS AC.ATK.TGT                      9028
    PAGE 439  PROCESS FORWARD.OBSERVER                9947
    PAGE 451  PROCESS HC.RETURN.FARRP                  563
    PAGE 479  PROCESS WITH.DRAW                       1985 2008
    PAGE 515  PROCESS HELICOPTER.FIRE                 3949
    PAGE 578  ROUTINE MINE.INPUT                      6630 6632
    PAGE 585  ROUTINE TACAIR.INPUT                    6966
MAX.FASCAM.RANGE
    PAGE  42  ''SECTION FOR DEFINITIONS               2420
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM              9872
    PAGE 226  ROUTINE REQUEST.WD.FASCAM                445
    PAGE 227                                           499
    PAGE 577  ROUTINE MINE.INPUT                      6556 6559 6559
    PAGE 701  ''PROGRAM  OLDER.VERSION                1850
MAX.FOV.RANGE
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4780 4798 4799
MAX.GRID.SQUARE
    PAGE 653  PROCESS AIRBORNE.RADAR                  9281 9284 9289
MAX.MINUS
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ            9312
    PAGE 204                                          9370 9372 9376
MAX.PREP
    PAGE 470  PROCESS TARGET.REPORT                   1493
    PAGE 473                                          1670 1685
MAX.PROB.LOS
    PAGE 154  ROUTINE AO.DETECTION                    7089
    PAGE 155                                          7188
    PAGE 156                                          7190 7227
MAX.R
    PAGE 115  ROUTINE PROX.CHECK                      5515 5530 5532
MAX.RANGE
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4780 4797 4798 4799 4803
    PAGE 154  ROUTINE AO.DETECTION                    7081 7094 7128
    PAGE 428  PROCESS AIR.OBSERVER                    9350 9352
    PAGE 429                                          9394 9405 9413 9416 9426
    PAGE 431                                          9531 9550 9556
    PAGE 432                                          9582 9614
    PAGE 433                                          9632 9650 9669 9672
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4642 4658 4672
```

```
MAX.RNG.INT
     PAGE 176  ROUTINE EST.COVERAGE                    8017 8048
     PAGE 177                                          8063 8064 8075 8104 8119
     PAGE 178                                          8120 8131
     PAGE 188  ROUTINE FINAL.COVERAGE                  8558 8591
     PAGE 189                                          8609 8610 8624 8648
     PAGE 190                                          8666 8667 8681
MAX.SUPP
     PAGE 272  ROUTINE BTRY.EFFECTS                    2717
     PAGE 273                                          2719 2721
MAX.TU
     PAGE 240  ROUTINE TARGET.ANALYSIS                 1069 1081 1085
     PAGE 241                                          1090 1094 1095
MAX.WD.FASCAM
     PAGE  42  ''SECTION FOR DEFINITIONS               2423
     PAGE 227  ROUTINE REQUEST.WD.FASCAM                497
     PAGE 577  ROUTINE MINE.INPUT                      6553
     PAGE 701  ''PROGRAM  OLDER.VERSION                1853
MAX.WPN.RNG
     PAGE 302  ROUTINE HEL.RANGE.COMPUTE               4167 4169 4170 4172 4175
     PAGE 446  PROCESS HC.ARRIVE.BATTLE                 273  275  277  279
     PAGE 447                                           365
     PAGE 448                                           404
MA.BATTLE.ENDED
     PAGE  37  ''SECTION FOR PROCESSES                 2090
     PAGE 329  ROUTINE EMPTY                           5229
     PAGE 464  PROCESS MINE.ASSESS                     1257 1272
     PAGE 696  ''PROGRAM  OLDER.VERSION                1528
MA.CASUALTIES
     PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES        1563
     PAGE  45  ''SECTION FOR DEFINITIONS               2556
     PAGE 251  ROUTINE MINE.EFFECTS                    1569
     PAGE 464  PROCESS MINE.ASSESS                     1249
     PAGE 687  ''PROGRAM  OLDER.VERSION                1002
     PAGE 704                                          1977
MA.LINK
     PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES        1561
     PAGE 251  ROUTINE MINE.EFFECTS                    1566
     PAGE 464  PROCESS MINE.ASSESS                     1250
     PAGE 687  ''PROGRAM  OLDER.VERSION                1000
     PAGE 712                                          2492
MA.MINEFIELD
     PAGE  37  ''SECTION FOR PROCESSES                 2089
     PAGE 696  ''PROGRAM  OLDER.VERSION                1527
MA.MSN
     PAGE  37  ''SECTION FOR PROCESSES                 2091
     PAGE 251  ROUTINE MINE.EFFECTS                    1537
     PAGE 466  PROCESS MINE.ASSESS                     1342
     PAGE 696  ''PROGRAM  OLDER.VERSION                1529
MA.SET
     PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES        1565
     PAGE  37  ''SECTION FOR PROCESSES                 2093
     PAGE 251  ROUTINE MINE.EFFECTS                    1567
     PAGE 464  PROCESS MINE.ASSESS                     1245 1247
     PAGE 687  ''PROGRAM  OLDER.VERSION                1004
     PAGE 696                                          1531
```

```
MA.UE.LINK
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1562
    PAGE 251  ROUTINE MINE.EFFECTS                      1568
    PAGE 464  PROCESS MINE.ASSESS                       1248
    PAGE 687  ''PROGRAM  OLDER.VERSION                  1001
MA.UNIT
    PAGE  37  ''SECTION FOR PROCESSES                   2088
    PAGE 329  ROUTINE EMPTY                             5226
    PAGE 696  ''PROGRAM  OLDER.VERSION                  1526
MCFR.EQ.ID
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           637
    PAGE 562  ROUTINE MCFR.INPUT                        5894
    PAGE 566  ROUTINE SENSOR.INPUT                      6119
    PAGE 671  ''PROGRAM  OLDER.VERSION                    78
MCFR.INPUT
    PAGE 521  ROUTINE MAIN2                             4251 4253
    PAGE 562  ROUTINE MCFR.INPUT                        5878
MCFR.MAX.ON
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           633
    PAGE 360  EVENT CFR.ON                              6426
    PAGE 562  ROUTINE MCFR.INPUT                        5896
    PAGE 671  ''PROGRAM  OLDER.VERSION                    74
MCFR.MIN.OFF
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           632
    PAGE 206  ROUTINE PDB.DETECTION                     9472
    PAGE 359  EVENT CFR.OFF                             6394
    PAGE 562  ROUTINE MCFR.INPUT                        5895
    PAGE 671  ''PROGRAM  OLDER.VERSION                    73
MCFR.NAME
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           636
    PAGE  47  ''SECTION FOR DEFINITIONS                 2676
    PAGE 562  ROUTINE MCFR.INPUT                        5893
    PAGE 671  ''PROGRAM  OLDER.VERSION                    77
    PAGE 706                                            2100
MCFR.RH.LIST
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES           383
    PAGE  12                                             639
    PAGE 166  ROUTINE CFR.DETECTION                     7674
    PAGE 562  ROUTINE MCFR.INPUT                        5904
    PAGE 666  ''PROGRAM  OLDER.VERSION                  9825
    PAGE 671                                              80
MCFR.SEARCH.WIDTH
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           635
    PAGE 165  ROUTINE CFR.DEGRADE                       7584
    PAGE 360  EVENT CFR.ON                              6458
    PAGE 562  ROUTINE MCFR.INPUT                        5899
    PAGE 671  ''PROGRAM  OLDER.VERSION                    76
MCFR.SWEEP.ANGLE
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           634
    PAGE 166  ROUTINE CFR.DETECTION                     7659
    PAGE 360  EVENT CFR.ON                              6462
    PAGE 562  ROUTINE MCFR.INPUT                        5897
    PAGE 671  ''PROGRAM  OLDER.VERSION                    75
MCORDER
    PAGE   1  ROUTINE FOR CROSS_REFERENCING               52
    PAGE  39  ''SECTION FOR EVENTS                      2215
```

```
    PAGE 698  ''PROGRAM  OLDER.VERSION                1652
MCUNIT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING            51
    PAGE  39  ''SECTION FOR EVENTS                   2214
    PAGE  58  ROUTINE CREATE.FORCE                   3142
    PAGE 698  ''PROGRAM  OLDER.VERSION               1651
MDORDER
    PAGE   1  ROUTINE FOR CROSS_REFERENCING            52
    PAGE  39  ''SECTION FOR EVENTS                   2219
    PAGE 698  ''PROGRAM  OLDER.VERSION               1656
MDUNIT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING            52
    PAGE  39  ''SECTION FOR EVENTS                   2218
    PAGE 698  ''PROGRAM  OLDER.VERSION               1655
MEMBERS.
    PAGE 142  ROUTINE DEAD.UNIT                      6529
METERS.
    PAGE 403  EVENT START.MOVE                       8190
MFB.DELAY
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        789
    PAGE 104  ROUTINE MINE.DELAY                     5002
    PAGE 577  ROUTINE MINE.INPUT                     6579 6581 6583
    PAGE 673  ''PROGRAM  OLDER.VERSION                231
MFB.UPPER.LIMIT
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        788
    PAGE 104  ROUTINE MINE.DELAY                     4997
    PAGE 577  ROUTINE MINE.INPUT                     6584 6585
    PAGE 673  ''PROGRAM  OLDER.VERSION                230
MFO.EQ.ID
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES        649
    PAGE 538  ROUTINE MFO.INPUT                      4949
    PAGE 566  ROUTINE SENSOR.INPUT                   6097
    PAGE 671  ''PROGRAM  OLDER.VERSION                 90
MFO.INPUT
    PAGE 520  ROUTINE MAIN2                          4218 4220
    PAGE 538  ROUTINE MFO.INPUT                      4930
MFO.NAME
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES        646
    PAGE  47  ''SECTION FOR DEFINITIONS              2677
    PAGE 254  ROUTINE FO.DETECTION                   1690 1695 1711 1721
    PAGE 255                                         1733 1745 1757
    PAGE 256                                         1799 1804 1814 1828
    PAGE 257                                         1860
    PAGE 258                                         1937
    PAGE 441  PROCESS FORWARD.OBSERVER                 53
    PAGE 538  ROUTINE MFO.INPUT                      4948
    PAGE 671  ''PROGRAM  OLDER.VERSION                 87
    PAGE 706                                         2101
MFO.PGM.CAP
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES        648
    PAGE 256  ROUTINE FO.DETECTION                   1795 1813
    PAGE 257                                         1894
    PAGE 258                                         1936
    PAGE 538  ROUTINE MFO.INPUT                      4951
    PAGE 671  ''PROGRAM  OLDER.VERSION                 89
```

```
MFO.RB.SET
     PAGE   9  ''SECTION FOR PERMANENT_ENTITIES          470
     PAGE  12                                            651
     PAGE  49  ''SECTION FOR DEFINITIONS                2811
     PAGE 255  ROUTINE FO.DETECTION                     1741
     PAGE 538  ROUTINE MFO.INPUT                        4960
     PAGE 668  ''PROGRAM  OLDER.VERSION                 9911
     PAGE 671                                             92
     PAGE 708                                           2232
MFO.SEARCH.RATE
     PAGE  12  ''SECTION FOR PERMANENT_ENTITIES          647
     PAGE 439  PROCESS FORWARD.OBSERVER                 9928 9930
     PAGE 538  ROUTINE MFO.INPUT                        4950
     PAGE 671  ''PROGRAM  OLDER.VERSION                   88
MFP.LIST
     PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          571
     PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES         1586
     PAGE 135  ROUTINE CHECK.FOR.MINES                  6276
     PAGE 578  ROUTINE MINE.INPUT                       6622
     PAGE 579                                           6653
     PAGE 670  ''PROGRAM  OLDER.VERSION                   12
     PAGE 687                                           1025
MFP.X.COORD
     PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES         1583
     PAGE  44  ''SECTION FOR DEFINITIONS                2491
     PAGE 135  ROUTINE CHECK.FOR.MINES                  6280 6280 6283 6284 6296 6298 6304 6308 6309 6310 6311
     PAGE 578  ROUTINE MINE.INPUT                       6625 6629 6630
     PAGE 579                                           6656
     PAGE 687  ''PROGRAM  OLDER.VERSION                 1022
     PAGE 702                                           1912
MFP.Y.COORD
     PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES         1584
     PAGE  44  ''SECTION FOR DEFINITIONS                2492
     PAGE 135  ROUTINE CHECK.FOR.MINES                  6281 6282 6296 6305 6312 6313 6314 6315
     PAGE 578  ROUTINE MINE.INPUT                       6627 6631 6632
     PAGE 579                                           6657
     PAGE 687  ''PROGRAM  OLDER.VERSION                 1023
     PAGE 702                                           1913
MF.BAND
     PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          787
     PAGE 104  ROUTINE MINE.DELAY                       4996
     PAGE 577  ROUTINE MINE.INPUT                       6574
     PAGE 673  ''PROGRAM  OLDER.VERSION                   229
MF.COLOR
     PAGE  10  ''SECTION FOR PERMANENT_ENTITIES          565
     PAGE  45  ''SECTION FOR DEFINITIONS                2557
     PAGE 134  ROUTINE CHECK.FOR.MINES                  6245
     PAGE 410  EVENT UPDATE.LOC                         8528
     PAGE 578  ROUTINE MINE.INPUT                       6609 6611 6634
     PAGE 670  ''PROGRAM  OLDER.VERSION                    6
     PAGE 704                                           1978
MF.DEBUG
     PAGE  42  ''SECTION FOR DEFINITIONS                2424
     PAGE 103  ROUTINE MINE.DELAY                       4958
     PAGE 104                                           5008
     PAGE 181  ROUTINE FASCAM.COMPUTATION               8273
```

```
    PAGE 217  ROUTINE REQUEST.FASCAM                  9973
    PAGE 221  ROUTINE REQUEST.ILLUM                    160
    PAGE 251  ROUTINE MINE.EFFECTS                    1539 1547
    PAGE 253                                          1642 1664
    PAGE 393  EVENT START.ARTY.MOVEMENT               7703
    PAGE 405  EVENT START.MOVE                        8317
    PAGE 410  EVENT UPDATE.LOC                        8572
    PAGE 464  PROCESS MINE.ASSESS                     1236
    PAGE 465                                          1319
    PAGE 466                                          1346
    PAGE 577  ROUTINE MINE.INPUT                      6551
    PAGE 701  ''PROGRAM  OLDER.VERSION                1854
MF.ID
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         564
    PAGE  47  ''SECTION FOR DEFINITIONS               2678
    PAGE 578  ROUTINE MINE.INPUT                      6605 6645
    PAGE 670  ''PROGRAM  OLDER.VERSION                   5
    PAGE 706                                          2102
MF.POINT
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES        1582
    PAGE 578  ROUTINE MINE.INPUT                      6621 6622 6625 6627 6629 6630 6631 6632
    PAGE 687  ''PROGRAM  OLDER.VERSION                1021
    PAGE 712                                          2493
MF.PRINT
    PAGE 577  ROUTINE MINE.INPUT                      6552
    PAGE 578                                          6643
MF.SWITCH
    PAGE  42  ''SECTION FOR DEFINITIONS               2425
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM              9839
    PAGE 226  ROUTINE REQUEST.WD.FASCAM                402
    PAGE 577  ROUTINE MINE.INPUT                      6546 6547
    PAGE 701  ''PROGRAM  OLDER.VERSION                1855
MF.TYPE
    PAGE 577  ROUTINE MINE.INPUT                      6540
    PAGE 578                                          6607 6634
MF.X.HIGH
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         566
    PAGE  45  ''SECTION FOR DEFINITIONS               2558
    PAGE 134  ROUTINE CHECK.FOR.MINES                 6260 6268
    PAGE 578  ROUTINE MINE.INPUT                      6639
    PAGE 670  ''PROGRAM  OLDER.VERSION                   7
    PAGE 704                                          1979
MF.X.LOW
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         567
    PAGE  45  ''SECTION FOR DEFINITIONS               2559
    PAGE 134  ROUTINE CHECK.FOR.MINES                 6259 6267
    PAGE 578  ROUTINE MINE.INPUT                      6638
    PAGE 670  ''PROGRAM  OLDER.VERSION                   8
    PAGE 704                                          1980
MF.Y.HIGH
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         568
    PAGE  45  ''SECTION FOR DEFINITIONS               2560
    PAGE 134  ROUTINE CHECK.FOR.MINES                 6262 6267
    PAGE 578  ROUTINE MINE.INPUT                      6641
    PAGE 670  ''PROGRAM  OLDER.VERSION                   9
    PAGE 704                                          1981
```

```
MF.Y.LOW
     PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          569
     PAGE  45  ''SECTION FOR DEFINITIONS                2561
     PAGE 134  ROUTINE CHECK.FOR.MINES                  6261 6268
     PAGE 578  ROUTINE MINE.INPUT                       6640
     PAGE 670  ''PROGRAM  OLDER.VERSION                   10
     PAGE 704                                           1982
MIL.FLAG
     PAGE 239  ROUTINE TARGET.ANALYSIS                   974  989
     PAGE 241                                           1096
MIL.WORTH
     PAGE 180  ROUTINE EST.MIL.WORTH                    8222 8236 8238
     PAGE 241  ROUTINE TARGET.ANALYSIS                  1105 1106
     PAGE 474  PROCESS TARGET.REPORT                    1736
MINEFIELD
     PAGE   1  ROUTINE FOR CROSS_REFERENCING              46
     PAGE  10  ''SECTION FOR PERMANENT_ENTITIES           563
     PAGE  83  ROUTINE CHANGE.LOC                       4137 4143
     PAGE 103  ROUTINE MINE.DELAY                       4944
     PAGE 134  ROUTINE CHECK.FOR.MINES                  6244
     PAGE 135                                           6273 6289
     PAGE 143  ROUTINE DEAD.UNIT                        6606
     PAGE 216  ROUTINE REQUEST.FASCAM                   9895
     PAGE 410  EVENT UPDATE.LOC                         8524 8527
     PAGE 484  PROCESS FIRE.MISSION                     2268
     PAGE 578  ROUTINE MINE.INPUT                       6600 6603 6605 6609 6611 6622 6634 6638 6639 6640 6641 6645
     PAGE 579                                           6653
     PAGE 670  ''PROGRAM  OLDER.VERSION                    4
     PAGE 712                                           2447
MINEFIELD.
     PAGE 410  EVENT UPDATE.LOC                         8540
MINEFIELD.PRINT
     PAGE  43  ''SECTION FOR DEFINITIONS                2441
MINES.
     PAGE 630  FUNCTION COLLISION                       8705
MINE.ASSESS
     PAGE  37  ''SECTION FOR PROCESSES                  2087
     PAGE 251  ROUTINE MINE.EFFECTS                     1531
     PAGE 329  ROUTINE EMPTY                            5225 5226 5228 5229 5230
     PAGE 464  PROCESS MINE.ASSESS                      1216 1229
     PAGE 600  ROUTINE BETWEEN.ROUTINE                  7500 7501
     PAGE 696  ''PROGRAM  OLDER.VERSION                 1525
MINE.DELAY
     PAGE 103  ROUTINE MINE.DELAY                       4932 4962
     PAGE 104                                           5015
     PAGE 251  ROUTINE MINE.EFFECTS                     1519
     PAGE 279  ROUTINE AC.BOMB.EFFECTS                  2984
MINE.EFFECTS
     PAGE 227  ROUTINE REQUEST.WD.FASCAM                 462
     PAGE 250  ROUTINE MINE.EFFECTS                     1459 1515
     PAGE 251                                           1541 1551
     PAGE 253                                           1648 1666
     PAGE 393  EVENT START.ARTY.MOVEMENT                7689
     PAGE 405  EVENT START.MOVE                         8298
     PAGE 410  EVENT UPDATE.LOC                         8553
     PAGE 484  PROCESS FIRE.MISSION                     2269
```

```
MINE.INPUT
    PAGE 521  ROUTINE MAIN2                          4284 4286
    PAGE 577  ROUTINE MINE.INPUT                     6533
MINE.OBSTACLE
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES       1571
    PAGE  87  ROUTINE END.MOVE                       4360
    PAGE 135  ROUTINE CHECK.FOR.MINES                6320
    PAGE 393  EVENT START.ARTY.MOVEMENT              7700
    PAGE 405  EVENT START.MOVE                       8313
    PAGE 409  EVENT UPDATE.LOC                       8503
    PAGE 410                                         8568
    PAGE 630  FUNCTION COLLISION                     8709
    PAGE 687  ''PROGRAM  OLDER.VERSION               1010
    PAGE 712                                         2494
MINE.USE.RULE
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        797
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM             9859
    PAGE 227  ROUTINE REQUEST.WD.FASCAM               494
    PAGE 578  ROUTINE MINE.INPUT                     6591
    PAGE 674  ''PROGRAM  OLDER.VERSION                239
MINE.WD.RULE
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        798
    PAGE 226  ROUTINE REQUEST.WD.FASCAM               419
    PAGE 578  ROUTINE MINE.INPUT                     6593
    PAGE 674  ''PROGRAM  OLDER.VERSION                240
MINUTES.V
    PAGE  80  ROUTINE BLOCK.LOS                      4031 4046
    PAGE  83  ROUTINE CHANGE.LOC                     4138 4140 4144
    PAGE 104  ROUTINE MINE.DELAY                     5037
    PAGE 105                                         5051
    PAGE 106  ROUTINE MIN.MOVE                       5094
    PAGE 165  ROUTINE CFR.DEGRADE                    7597
    PAGE 182  ROUTINE FA.BN.ASGN                     8341 8343
    PAGE 183                                         8345
    PAGE 205  ROUTINE NOISE.DEGRADE                  9394 9406
    PAGE 206  ROUTINE PDB.DETECTION                  9473
    PAGE 208  ROUTINE PGM.MSN.ASGN                   9518 9520
    PAGE 266  ROUTINE BTRY.EFFECTS                   2336
    PAGE 270                                         2587 2588
    PAGE 272                                         2706
    PAGE 273                                         2720
    PAGE 356  EVENT BTL.ENDED                        6266
    PAGE 394  EVENT START.ARTY.MOVEMENT              7732
    PAGE 402  EVENT START.BATTLE                     8158
    PAGE 429  PROCESS AIR.OBSERVER                   9409 9440
    PAGE 430                                         9459
    PAGE 431                                         9507 9515
    PAGE 432                                         9594 9598
    PAGE 438  PROCESS FORWARD.OBSERVER               9892
    PAGE 441                                           70   72
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          741  743
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE           1397
    PAGE 468                                         1441 1461
    PAGE 481  PROCESS FIRE.MISSION                   2127
    PAGE 482                                         2143
    PAGE 653  PROCESS AIRBORNE.RADAR                 9299 9304
```

```
    PAGE 654                                          9344 9369
MIN.APPAR.SIZE
    PAGE 154  ROUTINE AO.DETECTION                    7091
    PAGE 156                                          7229 7239 7243
MIN.APP.SIZE
    PAGE 156  ROUTINE AO.DETECTION                    7244
MIN.BAT
    PAGE   2  PROGRAM REVISIONS                         77
    PAGE  90  ROUTINE FA.BN.MOVEMENT                  4457 4459 4462 4462 4463 4464 4464 4465 4466 4470 4470 4475 4476 4478 4479
                                                      4479
MIN.CPE.INT
    PAGE 176  ROUTINE EST.COVERAGE                    8017 8043 8047 8057 8059 8060
    PAGE 177                                          8065 8069 8099 8103 8113 8115 8116
    PAGE 178                                          8121 8125
    PAGE 188  ROUTINE FINAL.COVERAGE                  8558 8586 8590 8600
    PAGE 189                                          8605 8606 8611 8618 8643 8647 8657 8662
    PAGE 190                                          8663 8668 8675
MIN.DIST
    PAGE  90  ROUTINE FA.BN.MOVEMENT                  4457 4462 4463
MIN.DISTANCE
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                 196  207  208
    PAGE 445                                           228  234
    PAGE 447                                           354
    PAGE 448                                           404
MIN.DIST.2
    PAGE 261  ROUTINE BTRY.EFFECTS                    2026
MIN.F
    PAGE  65  ROUTINE FILE.KAD.SENSOR                 3405 3407
    PAGE  78  ROUTINE ADJUST                          3978 3980
    PAGE 106  ROUTINE MIN.MOVE                        5074
    PAGE 123  ROUTINE TIME.TO.DETECT                  5840
    PAGE 134  ROUTINE CHECK.FOR.MINES                 6229 6231
    PAGE 135                                          6308 6312
    PAGE 181  ROUTINE FASCAM.COMPUTATION              8267
    PAGE 183  ROUTINE FA.BN.ASGN                      8394
    PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION           9103
    PAGE 198                                          9111
    PAGE 199  ROUTINE ILLUM.COMPUTATION               9164
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9525
    PAGE 233  ROUTINE SMOKE.COMPUTATION                721
    PAGE 251  ROUTINE MINE.EFFECTS                    1563
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                 2831
    PAGE 278                                          2920
    PAGE 286  ROUTINE CHECK.CAS.CONSTRAINTS           3378
    PAGE 316  ROUTINE FLIGHT.PATH                     4783
    PAGE 349  EVENT AD.ENGAGEMENT                     5924 5926
    PAGE 351                                          6044
    PAGE 422  PROCESS AC.ATK.TGT                      9029
    PAGE 458  PROCESS HEL.TARGET.ACQUISITION           932
    PAGE 459                                          1039
    PAGE 464  PROCESS MINE.ASSESS                     1252
    PAGE 465                                          1308
    PAGE 514  PROCESS HELICOPTER.FIRE                 3891
    PAGE 516                                          4029
    PAGE 578  ROUTINE MINE.INPUT                      6629 6631
```

```
MIN.FASCAM.RANGE
    PAGE  42  ''SECTION FOR DEFINITIONS              2421
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM             9871
    PAGE 226  ROUTINE REQUEST.WD.FASCAM               444
    PAGE 227                                          454  498
    PAGE 577  ROUTINE MINE.INPUT                     6555 6558 6558
    PAGE 701  ''PROGRAM  OLDER.VERSION               1851
MIN.FLIGHT.TIME
    PAGE 297  ROUTINE FARRP.CHECK                    3870 3899 3901
MIN.GRID.SQUARE
    PAGE 653  PROCESS AIRBORNE.RADAR                 9280 9285 9288
MIN.MARG
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9312 9332 9334 9346
    PAGE 204                                         9373 9378
MIN.NO.SUP.UNITS
    PAGE  42  ''SECTION FOR DEFINITIONS              2396
    PAGE 305  ROUTINE UNIT.PRIORITY                  4265
    PAGE 572  ROUTINE FARRP.INPUT                    6357 6382
    PAGE 701  ''PROGRAM  OLDER.VERSION               1826
MIN.PROB.LOS
    PAGE 154  ROUTINE AO.DETECTION                   7089
    PAGE 156                                         7221 7223 7227
MIN.R
    PAGE 115  ROUTINE PROX.CHECK                     5515 5531 5532
MIN.RANGE
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT              4642 4659 4673
MIN.RNG.INT
    PAGE 176  ROUTINE EST.COVERAGE                   8017 8042 8049 8052 8053 8054 8055
    PAGE 177                                         8064 8074 8098 8105 8108 8109 8110 8111
    PAGE 178                                         8120 8130
    PAGE 188  ROUTINE FINAL.COVERAGE                 8558 8585 8592 8595 8596 8597 8598
    PAGE 189                                         8610 8623 8642 8649 8652 8653 8654 8655
    PAGE 190                                         8667 8680
MIN.SEGMENT
    PAGE 106  ROUTINE MIN.MOVE                       5071 5074 5074 5075 5076 5077 5078 5092 5092 5093
MIN.SPEED
    PAGE 298  ROUTINE HC.COMPUTE.TIMES               3962 3964 3969 3969
    PAGE 299                                         3972 3991 3997
MIN.SUPP
    PAGE 273  ROUTINE BTRY.EFFECTS                   2718 2721
MIN.X.COORD
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                198
    PAGE 445                                          209  215  229  234
    PAGE 447                                          327
MIN.Y.COORD
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                199
    PAGE 445                                          210  215  229  234
    PAGE 447                                          328
MISSILES.
    PAGE 280  ROUTINE AC.DF.EFFECTS                  3017
MISSION
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           45
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        374
    PAGE  11                                          581  584
    PAGE  14                                          791
    PAGE  15                                          800
```

PAGE 16		883
PAGE 35	''SECTION FOR PROCESSES	1996 2012
PAGE 48	''SECTION FOR DEFINITIONS	2761 2762 2763 2764
PAGE 58	ROUTINE CREATE.FORCE	3094 3107 3108 3124
PAGE 103	ROUTINE MINE.DELAY	4945
PAGE 115	ROUTINE PROX.CHECK	5505 5508 5512 5533 5536
PAGE 163	ROUTINE BTRY.FM.DEQ	7513 7516 7523 7525 7526
PAGE 164	ROUTINE BTRY.FM.ENQ	7547 7549 7565
PAGE 176	ROUTINE EST.COVERAGE	8008 8014 8020 8021 8023 8024 8027
PAGE 177		8083
PAGE 178		8147 8153 8154 8157 8158 8162 8162 8172 8172 8173 8176 8177
PAGE 182	ROUTINE FA.BN.ASGN	8298
PAGE 183		8361 8401
PAGE 185		8461 8501
PAGE 188	ROUTINE FINAL.COVERAGE	8547 8555 8563 8564 8568 8570 8574 8576
PAGE 189		8635
PAGE 190		8703 8704
PAGE 191		8743
PAGE 192		8808 8809 8818 8819 8823
PAGE 193		8854
PAGE 200	ROUTINE ILLUM.EFFECTS	9185
PAGE 203	ROUTINE MARGINAL.EFFECTS.ADJ	9300 9307 9318 9319 9320 9321 9322 9326 9331 9333 9343 9349 9352 9355
PAGE 204		9363 9381
PAGE 208	ROUTINE PGM.MSN.ASGN	9502 9522
PAGE 210		9653
PAGE 211		9690
PAGE 214	ROUTINE REM.EFFECTS.COMPUTATION	9797
PAGE 216	ROUTINE REQUEST.FASCAM	9893 9914
PAGE 219	ROUTINE REQUEST.ILLUM	87
PAGE 222	ROUTINE REQUEST.SMOKE	179
PAGE 223		285
PAGE 227	ROUTINE REQUEST.WD.FASCAM	468
PAGE 234	ROUTINE SMOKE.EFFECTS	743
PAGE 248	ROUTINE WEIGHTED.VOLLEYS	1372 1387 1391 1394 1401
PAGE 256	ROUTINE FO.DETECTION	1806 1819
PAGE 258		1945
PAGE 260	ROUTINE BTRY.EFFECTS	1969 1985
PAGE 262		2094 2095 2096 2097
PAGE 263		2145
PAGE 264		2202
PAGE 266		2315 2316 2324 2325 2333 2334 2335 2337 2338 2339
PAGE 267		2395
PAGE 270		2593
PAGE 272		2682
PAGE 285	ROUTINE CHECK.CAS.CONSTRAINTS	3294 3322
PAGE 286		3388 3395
PAGE 287		3406
PAGE 293	ROUTINE END.CAS.MISSION	3688 3711
PAGE 294		3771
PAGE 295		3803 3805 3808
PAGE 305	ROUTINE UNIT.PRIORITY	4243
PAGE 314	ROUTINE FLIGHT.PATH	4641 4647
PAGE 327	ROUTINE DQ.CMSN.QUEUE	5147
PAGE 334	ROUTINE FINISH.COMPUTATION	5402
PAGE 335	ROUTINE FRAC.COMPUTE	5417
PAGE 380	EVENT INIT.PREPLAN.CAS	7257

```
    PAGE 398  EVENT START.BATTLE                    7961
    PAGE 405  EVENT START.MOVE                      8304
    PAGE 414  EVENT ACT.DEF                         8720
    PAGE 415  EVENT ACT.MOVCOR                      8744
    PAGE 426  PROCESS AC.ATK.TGT                    9291
    PAGE 427                                        9322
    PAGE 473  PROCESS TARGET.REPORT                 1702
    PAGE 474                                        1739 1760
    PAGE 477                                        1900
    PAGE 480  PROCESS FIRE.MISSION                  2030 2032
    PAGE 481                                        2106 2109
    PAGE 482                                        2151 2170 2184 2190 2199
    PAGE 484                                        2267
    PAGE 485                                        2356 2358
    PAGE 486                                        2382 2384
    PAGE 542  ROUTINE ORD.DEF                       5123
    PAGE 546  ROUTINE ORD.MOVCOR                    5217
    PAGE 547  ROUTINE P.E.M.INPUT                   5258 5276 5279 5280 5291 5295 5299 5300
    PAGE 549  ROUTINE TB.INPUT                      5376
    PAGE 570  ROUTINE DECISION.INPUT                6268 6272 6274 6275 6276
    PAGE 576  ROUTINE ILLUM.INPUT                   6524 6527
    PAGE 577  ROUTINE MINE.INPUT                    6588
    PAGE 578                                        6591 6592 6593
    PAGE 580  ROUTINE SMOKE.INPUT                   6697 6700 6701 6702 6703 6704
    PAGE 582  ROUTINE TACAIR.INPUT                  6775
    PAGE 618  ROUTINE SNAP.R                        8230
    PAGE 621  ROUTINE TACAIR.DATA.REPORT            8373 8382
    PAGE 622                                        8394
    PAGE 629  FUNCTION BTRY.AVAILABLE               8649
    PAGE 666  ''PROGRAM  OLDER.VERSION              9816
    PAGE 670                                          22   25
    PAGE 673                                         233
    PAGE 674                                         242
    PAGE 675                                         325
    PAGE 694                                        1434 1450
    PAGE 707                                        2183 2184 2185 2186
    PAGE 712                                        2448
MISSIONS.
    PAGE 549  ROUTINE TB.INPUT                      5378
    PAGE 585  ROUTINE TACAIR.INPUT                  6913
MISSION.
    PAGE  58  ROUTINE CREATE.FORCE                  3123
    PAGE 350  EVENT AD.ENGAGEMENT                   6007 6012
    PAGE 351                                        6017
    PAGE 420  PROCESS AC.ATK.TGT                    8958
MISSION.NR
    PAGE 295  ROUTINE END.CAS.MISSION               3817
MISSION.TYPE
    PAGE 295  ROUTINE END.CAS.MISSION               3817
MIX.TERRAIN
    PAGE  52  ''SECTION FOR SUBSTITUTIONS           2969
    PAGE 336  ROUTINE GET.TERRAIN                   5448
    PAGE 710  ''PROGRAM  OLDER.VERSION              2376
MNMTR.PROB.DETECT
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES       658
    PAGE 255  ROUTINE FO.DETECTION                  1768
```

```
    PAGE 538  ROUTINE MFO.INPUT                    4976
    PAGE 671  ''PROGRAM  OLDER.VERSION              100
MN.NAME
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES      582
    PAGE  47  ''SECTION FOR DEFINITIONS            2679
    PAGE  52  ''SECTION FOR SUBSTITUTIONS          2952
    PAGE 398  EVENT START.BATTLE                   7959
    PAGE 547  ROUTINE P.E.M.INPUT                  5280
    PAGE 621  ROUTINE TACAIR.DATA.REPORT           8384
    PAGE 670  ''PROGRAM  OLDER.VERSION               23
    PAGE 706                                       2103
    PAGE 710                                       2359
MODEL.AD.SENSOR
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES      589
    PAGE 587  ROUTINE MADS.INPUT                   6984 6985
    PAGE 623  ROUTINE TACAIR.DATA.REPORT           8457 8468
    PAGE 670  ''PROGRAM  OLDER.VERSION               30
    PAGE 712                                       2449
MODEL.AO
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES      607  618
    PAGE 564  ROUTINE MAO.INPUT                    5963 5965 6003
    PAGE 618  ROUTINE SNAP.R                       8231
    PAGE 670  ''PROGRAM  OLDER.VERSION               48   59
    PAGE 712                                       2450
MODEL.AR
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES      621
    PAGE 618  ROUTINE SNAP.R                       8232
    PAGE 671  ''PROGRAM  OLDER.VERSION               62
    PAGE 712                                       2451
MODEL.CFR
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES      631
    PAGE 562  ROUTINE MCFR.INPUT                   5887 5891
    PAGE 618  ROUTINE SNAP.R                       8233
    PAGE 671  ''PROGRAM  OLDER.VERSION               72
    PAGE 712                                       2452
MODEL.FO
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES      645  657
    PAGE 538  ROUTINE MFO.INPUT                    4939 4946 4967
    PAGE 618  ROUTINE SNAP.R                       8234
    PAGE 671  ''PROGRAM  OLDER.VERSION               86   99
    PAGE 712                                       2453
MODEL.PDB
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES      660
    PAGE 563  ROUTINE MPDB.INPUT                   5924 5928
    PAGE 618  ROUTINE SNAP.R                       8235
    PAGE 671  ''PROGRAM  OLDER.VERSION              103
    PAGE 712                                       2454
MODEL.PIR
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES      671  679
    PAGE 618  ROUTINE SNAP.R                       8236
    PAGE 671  ''PROGRAM  OLDER.VERSION              114
    PAGE 672                                        122
    PAGE 712                                       2455
MODEL.RPV
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES      683  693
    PAGE 618  ROUTINE SNAP.R                       8237
```

```
    PAGE 672  ''PROGRAM  OLDER.VERSION                126  136
    PAGE 712                                         2456
MODINPUT.DAT
    PAGE 647  ROUTINE OPEN.INPUT.OUTPUT.FILES        9182
MOD.F
    PAGE 176  ROUTINE ES1.COVERAGE                   8040
    PAGE 177                                         8096
    PAGE 188  ROUTINE FINAL.COVERAGE                 8583
    PAGE 189                                         8640
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                7265 7274 7283 7293
    PAGE 596                                         7306 7322
    PAGE 597                                         7359 7367 7375 7383 7391
    PAGE 606  ROUTINE KV.PRINT                       7739
    PAGE 607                                         7747 7755 7766 7774 7782
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES            8534 8545
    PAGE 625                                         8556 8564 8575 8586
MOTION.
    PAGE 415  EVENT ACT.MOVCOR                       8745
MOVCOR.ORDER
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES       1592
    PAGE 326  ROUTINE DESTROY.ORD                    5134
    PAGE 546  ROUTINE ORD.MOVCOR                     5232
    PAGE 687  ''PROGRAM  OLDER.VERSION               1031
    PAGE 712                                         2495
MOVDIS.ORDER
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES       1602
    PAGE 144  ROUTINE DEAD.UNIT                      6677
    PAGE 326  ROUTINE DESTROY.ORD                    5132
    PAGE 545  ROUTINE ORD.MOVDIS                     5195
    PAGE 687  ''PROGRAM  OLDER.VERSION               1041
    PAGE 712                                         2496
MOVED.
    PAGE  77  ROUTINE ADJUST                         3888
MOVED.UNIT
    PAGE 415  EVENT ACT.MOVCOR                       8751 8753
MOVED.UNIT.
    PAGE 415  EVENT ACT.MOVCOR                       8738 8750 8755 8758
MOVEMENT.STATUS
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES        254
    PAGE  12                                          657  679
    PAGE  13                                          693
    PAGE 538  ROUTINE MFO.INPUT                      4936 4973
    PAGE 664  ''PROGRAM  OLDER.VERSION               9696
    PAGE 671                                           99
    PAGE 672                                          122  136
MOVE.
    PAGE  87  ROUTINE END.MOVE                       4313
    PAGE 138  ROUTINE CHECK.PROX                     6378
    PAGE 403  EVENT START.MOVE                       8188
    PAGE 545  ROUTINE ORD.MOVDIS                     5187
MOVE.DIST
    PAGE 393  EVENT START.ARTY.MOVEMENT              7658 7675
    PAGE 394                                         7732 7736 7739
    PAGE 407  EVENT STOP.ARTY.MOVEMENT               8373 8396
MOVE.FACTOR
    PAGE 106  ROUTINE MIN.MOVE                       5069 5083 5088 5088 5090 5093
```

```
MOVE.FACTOR.NR
    PAGE  42  ''SECTION FOR DEFINITIONS              2376
    PAGE 701  ''PROGRAM  OLDER.VERSION               1806
MOVE.FIRE.DIST
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        844
    PAGE  43  ''SECTION FOR DEFINITIONS              2446
    PAGE 149  ROUTINE PK.COMPUTE                     6917 6923
    PAGE 571  ROUTINE TT.FACTORS.INPUT               6319
    PAGE 674  ''PROGRAM  OLDER.VERSION                286
    PAGE 702                                         1866
MOVE.RATE
    PAGE  83  ROUTINE CHANGE.LOC                     4133 4148 4151 4151 4154 4156
MOVE.STATUS
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7932 7978 7980 7986
MOVE.TIME
    PAGE 337  ROUTINE HC.EMPTY                       5503 5510
    PAGE 393  EVENT START.ARTY.MOVEMENT              7663
    PAGE 394                                         7732 7737
    PAGE 546  ROUTINE ORD.MOVCOR                     5250
MOVE.TIME.
    PAGE 546  ROUTINE ORD.MOVCOR                     5230 5233 5245
MOVE.TYPE
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES       1606
    PAGE  47  ''SECTION FOR DEFINITIONS              2680
    PAGE 144  ROUTINE DEAD.UNIT                      6681
    PAGE 416  EVENT ACT.MOVDIS                       8798
    PAGE 545  ROUTINE ORD.MOVDIS                     5198
    PAGE 687  ''PROGRAM  OLDER.VERSION               1045
    PAGE 706                                         2104
MOVING.TO.BATTLE
    PAGE  52  ''SECTION FOR SUBSTITUTIONS            2982
    PAGE 391  EVENT SEND.TEAM                        7607
    PAGE 711  ''PROGRAM  OLDER.VERSION               2389
MOVREINF.ORD
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES       1608
    PAGE  95  ROUTINE INIT.REINF                     4651
    PAGE 326  ROUTINE DESTROY.ORD                    5136
    PAGE 688  ''PROGRAM  OLDER.VERSION               1047
    PAGE 713                                         2497
MOV.FAC
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        846
    PAGE  48  ''SECTION FOR DEFINITIONS              2746
    PAGE  80  ROUTINE BLOCK.LOS                      4023
    PAGE  83  ROUTINE CHANGE.LOC                     4149
    PAGE 106  ROUTINE MIN.MOVE                       5083
    PAGE 394  EVENT START.ARTY.MOVEMENT              7731
    PAGE 403  EVENT START.MOVE                       8217
    PAGE 571  ROUTINE TT.FACTORS.INPUT               6322
    PAGE 674  ''PROGRAM  OLDER.VERSION                288
    PAGE 707                                         2168
MOV.MISSION
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES       1595
    PAGE 415  EVENT ACT.MOVCOR                       8750
    PAGE 546  ROUTINE ORD.MOVCOR                     5236
    PAGE 687  ''PROGRAM  OLDER.VERSION               1034
```

```
MO.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          1005
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1576
    PAGE  87  ROUTINE END.MOVE                          4357 4359
    PAGE 135  ROUTINE CHECK.FOR.MINES                   6324
    PAGE 393  EVENT START.ARTY.MOVEMENT                 7686 7688
    PAGE 404  EVENT START.MOVE                          8280
    PAGE 405                                            8312
    PAGE 409  EVENT UPDATE.LOC                          8500 8502
    PAGE 410                                            8525 8567
    PAGE 630  FUNCTION COLLISION                        8706 8708
    PAGE 677  ''PROGRAM  OLDER.VERSION                   447
    PAGE 687                                            1015
MO.MINEFIELD
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1572
    PAGE 135  ROUTINE CHECK.FOR.MINES                   6325
    PAGE 405  EVENT START.MOVE                          8311
    PAGE 410  EVENT UPDATE.LOC                          8530 8566
    PAGE 687  ''PROGRAM  OLDER.VERSION                  1011
MO.X.INTER
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1573
    PAGE  45  ''SECTION FOR DEFINITIONS                 2562
    PAGE 135  ROUTINE CHECK.FOR.MINES                   6322
    PAGE 393  EVENT START.ARTY.MOVEMENT                 7692
    PAGE 404  EVENT START.MOVE                          8282
    PAGE 405                                            8301
    PAGE 410  EVENT UPDATE.LOC                          8536 8556
    PAGE 687  ''PROGRAM  OLDER.VERSION                  1012
    PAGE 704                                            1983
MO.Y.INTER
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1574
    PAGE  45  ''SECTION FOR DEFINITIONS                 2563
    PAGE 135  ROUTINE CHECK.FOR.MINES                   6323
    PAGE 393  EVENT START.ARTY.MOVEMENT                 7693
    PAGE 404  EVENT START.MOVE                          8283
    PAGE 405                                            8302
    PAGE 410  EVENT UPDATE.LOC                          8537 8557
    PAGE 687  ''PROGRAM  OLDER.VERSION                  1013
    PAGE 704                                            1984
MPDB.CIR.ERROR
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES           699
    PAGE 206  ROUTINE PDB.DETECTION                     9456
    PAGE 563  ROUTINE MPDB.INPUT                        5945
    PAGE 672  ''PROGRAM  OLDER.VERSION                   142
MPDB.DET.PROB
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES           698
    PAGE 206  ROUTINE PDB.DETECTION                     9449
    PAGE 563  ROUTINE MPDB.INPUT                        5944
    PAGE 672  ''PROGRAM  OLDER.VERSION                   141
MPDB.EQ.ID
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           663
    PAGE 563  ROUTINE MPDB.INPUT                        5931
    PAGE 567  ROUTINE SENSOR.INPUT                      6129
    PAGE 671  ''PROGRAM  OLDER.VERSION                   106
MPDB.INPUT
    PAGE 521  ROUTINE MAIN2                             4254 4256
```

```
    PAGE 563  ROUTINE MPDB.INPUT                        5915
MPDB.KEY.TIME
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           661
    PAGE 206  ROUTINE PDB.DETECTION                     9474
    PAGE 563  ROUTINE MPDB.INPUT                        5932
    PAGE 671  ''PROGRAM  OLDER.VERSION                   104
MPDB.NAME
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           662
    PAGE  47  ''SECTION FOR DEFINITIONS                 2681
    PAGE 563  ROUTINE MPDB.INPUT                        5930
    PAGE 671  ''PROGRAM  OLDER.VERSION                   105
    PAGE 706                                            2105
MPDB.RH.LIST
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           665
    PAGE  13                                             704
    PAGE 206  ROUTINE PDB.DETECTION                     9442
    PAGE 563  ROUTINE MPDB.INPUT                        5940
    PAGE 671  ''PROGRAM  OLDER.VERSION                   108
    PAGE 672                                             147
MPIR.CIR.ERROR
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           676
    PAGE  45  ''SECTION FOR DEFINITIONS                 2567
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                   9576
    PAGE 672  ''PROGRAM  OLDER.VERSION                   119
    PAGE 704                                            1988
MPIR.HALF.COV.WIDTH
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           677
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                   9503
    PAGE 672  ''PROGRAM  OLDER.VERSION                   120
MPIR.MAX.PROCESS
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           674
    PAGE  45  ''SECTION FOR DEFINITIONS                 2565
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                   9519
    PAGE 659                                            9554
    PAGE 671  ''PROGRAM  OLDER.VERSION                   117
    PAGE 704                                            1986
MPIR.MIN.PROCESS
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           675
    PAGE  45  ''SECTION FOR DEFINITIONS                 2566
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                   9554
    PAGE 671  ''PROGRAM  OLDER.VERSION                   118
    PAGE 704                                            1987
MPIR.NAME
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           672
    PAGE  47  ''SECTION FOR DEFINITIONS                 2682
    PAGE 671  ''PROGRAM  OLDER.VERSION                   115
    PAGE 706                                            2106
MPIR.VELOCITY
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           673
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                   9495
    PAGE 671  ''PROGRAM  OLDER.VERSION                   116
MPMET.ACQUIRE.PROB
    PAGE 212  ROUTINE PIR.DETECTION                     9750
    PAGE 229  ROUTINE RPV.DETECTION                      563
MPMET.DETECT.PROB
    PAGE 212  ROUTINE PIR.DETECTION                     9748
```

```
    PAGE 229  ROUTINE RPV.DETECTION                    561
MPMET.PROB.ACQUIRE
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES         681
    PAGE 672  ''PROGRAM  OLDER.VERSION                 124
MPMET.PROB.DETECT
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES         680
    PAGE 672  ''PROGRAM  OLDER.VERSION                 123
MRH.MIN.ALT
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES        1528
    PAGE 350  EVENT AD.ENGAGEMENT                     5987
    PAGE 587  ROUTINE MADS.INPUT                      7003
    PAGE 623  ROUTINE TACAIR.DATA.REPORT              8482
    PAGE 686  ''PROGRAM  OLDER.VERSION                 967
MRH.PD
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES        1529
    PAGE  43  ''SECTION FOR DEFINITIONS               2482
    PAGE 350  EVENT AD.ENGAGEMENT                     6001
    PAGE 587  ROUTINE MADS.INPUT                      7001
    PAGE 623  ROUTINE TACAIR.DATA.REPORT              8483
    PAGE 686  ''PROGRAM  OLDER.VERSION                 968
    PAGE 702                                          1902
MRH.RANGE
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES        1527
    PAGE  49  ''SECTION FOR DEFINITIONS               2817
    PAGE  65  ROUTINE FILE.KAD.SENSOR                 3381
    PAGE 316  ROUTINE FLIGHT.PATH                     4794
    PAGE 350  EVENT AD.ENGAGEMENT                     5976 5984
    PAGE 506  PROCESS CAS.MISSION                     3506
    PAGE 587  ROUTINE MADS.INPUT                      7002
    PAGE 623  ROUTINE TACAIR.DATA.REPORT              8481
    PAGE 686  ''PROGRAM  OLDER.VERSION                 966
    PAGE 708                                          2238
MRMET.PROB.ACQUIRE
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         695
    PAGE 672  ''PROGRAM  OLDER.VERSION                 138
MRMET.PROB.DETECT
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         694
    PAGE 672  ''PROGRAM  OLDER.VERSION                 137
MRPV.CIR.ERROR
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         690
    PAGE  45  ''SECTION FOR DEFINITIONS               2570
    PAGE 230  ROUTINE RPV.DETECTION                    595
    PAGE 672  ''PROGRAM  OLDER.VERSION                 133
    PAGE 704                                          1991
MRPV.HALF.COV.WIDTH
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         688
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE            1379
    PAGE 672  ''PROGRAM  OLDER.VERSION                 131
MRPV.MAX.ALOFT.TIME
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         689
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE            1397
    PAGE 672  ''PROGRAM  OLDER.VERSION                 132
MRPV.MAX.PREP
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         686
    PAGE  45  ''SECTION FOR DEFINITIONS               2568
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE            1390
```

```
    PAGE 672  ''PROGRAM  OLDER.VERSION                 129
    PAGE 704                                          1989
MRPV.MIN.PREP
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         687
    PAGE  45  ''SECTION FOR DEFINITIONS               2569
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE            1390
    PAGE 672  ''PROGRAM  OLDER.VERSION                 130
    PAGE 704                                          1990
MRPV.NAME
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         684
    PAGE  47  ''SECTION FOR DEFINITIONS               2683
    PAGE 672  ''PROGRAM  OLDER.VERSION                 127
    PAGE 706                                          2107
MRPV.PGM.CAP
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         691
    PAGE 230  ROUTINE RPV.DETECTION                    605
    PAGE 672  ''PROGRAM  OLDER.VERSION                 134
MRPV.VELOCITY
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         685
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE            1381
    PAGE 672  ''PROGRAM  OLDER.VERSION                 128
MUNS.INPUT
    PAGE 520  ROUTINE MAIN2                           4236 4238
    PAGE 555  ROUTINE MUNS.INPUT                      5602
MU.CRIT.NO
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES        1538
    PAGE 130  ROUTINE CHECK.DEAD                      6085
    PAGE 133  ROUTINE CHECK.FORCE                     6171 6174
    PAGE 138  ROUTINE CHECK.PROX                      6428
    PAGE 139                                          6445
    PAGE 141  ROUTINE CHECK.STREN                     6512 6516
    PAGE 142  ROUTINE DEAD.UNIT                       6542
    PAGE 145  ROUTINE FIN.BATTLE                      6711
    PAGE 252  ROUTINE MINE.EFFECTS                    1576
    PAGE 270  ROUTINE BTRY.EFFECTS                    2569
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                 2950
    PAGE 282  ROUTINE AC.DF.EFFECTS                   3133
    PAGE 324  ROUTINE DECIDE                          5054
    PAGE 384  EVENT OFF.LINE.ATTRITION                7401
    PAGE 398  EVENT START.BATTLE                      7954
    PAGE 436  PROCESS ARTY.ASSESS                     9753 9797
    PAGE 437                                          9834
    PAGE 465  PROCESS MINE.ASSESS                     1292 1312
    PAGE 478  PROCESS WITH.DRAW                       1930
    PAGE 492  PROCESS ASSESSMENT                      2738
    PAGE 516  PROCESS HELICOPTER.FIRE                 4059
    PAGE 535  ROUTINE UNIT.INPUT                      4807
    PAGE 686  ''PROGRAM  OLDER.VERSION                 977
MU.CUR.ORDER
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES        1540
    PAGE  61  ROUTINE FEBA.INITIAL                    3211
    PAGE  95  ROUTINE INIT.REINF                      4658
    PAGE 113  ROUTINE PREPARE.LIST                    5409 5423
    PAGE 118  ROUTINE REIN.ARRIVE                     5625
    PAGE 119  ROUTINE RESET.FEBA.SECTOR               5677
    PAGE 126  ROUTINE WHAT.NEXT                       5933 5956
```

```
    PAGE 143  ROUTINE DEAD.UNIT                    6597 6597
    PAGE 144                                       6650 6650 6685
    PAGE 146  ROUTINE INTER.BATTLE                 6776
    PAGE 373  EVENT GET.NX.ORD                     6892 6910
    PAGE 415  EVENT ACT.MOVCOR                     8755
    PAGE 541  ROUTINE READ.ORDERS                  5102
    PAGE 630  FUNCTION COLLISION                   8694 8696
    PAGE 686  ''PROGRAM  OLDER.VERSION              979
MU.OFFSET.X
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES     1542
    PAGE  44  ''SECTION FOR DEFINITIONS            2498
    PAGE  66  ROUTINE FORM.TF.LIST                 3440
    PAGE  87  ROUTINE END.MOVE                     4329
    PAGE 408  EVENT UPDATE.LOC                     8461
    PAGE 686  ''PROGRAM  OLDER.VERSION              981
    PAGE 703                                       1919
MU.OFFSET.Y
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES     1543
    PAGE  44  ''SECTION FOR DEFINITIONS            2499
    PAGE  66  ROUTINE FORM.TF.LIST                 3441
    PAGE  87  ROUTINE END.MOVE                     4331
    PAGE 147  ROUTINE INTER.BATTLE                 6826
    PAGE 148                                       6863
    PAGE 409  EVENT UPDATE.LOC                     8462
    PAGE 686  ''PROGRAM  OLDER.VERSION              982
    PAGE 703                                       1920
MU.ORDER.SET
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES     1547
    PAGE  29                                       1625
    PAGE  49  ''SECTION FOR DEFINITIONS            2815
    PAGE  66  ROUTINE FORM.TF.LIST                 3431
    PAGE  95  ROUTINE INIT.REINF                   4656
    PAGE 113  ROUTINE PREPARE.LIST                 5408 5419
    PAGE 118  ROUTINE REIN.ARRIVE                  5624
    PAGE 125  ROUTINE WHAT.NEXT                    5896 5903 5912
    PAGE 126                                       5931 5954
    PAGE 143  ROUTINE DEAD.UNIT                    6596 6598 6600 6601
    PAGE 144                                       6644 6645 6647 6648 6669 6683
    PAGE 146  ROUTINE INTER.BATTLE                 6764
    PAGE 326  ROUTINE DESTROY.ORD                  5122 5124
    PAGE 373  EVENT GET.NX.ORD                     6891 6906
    PAGE 539  ROUTINE READ.ORDERS                  5018
    PAGE 540                                       5098
    PAGE 630  FUNCTION COLLISION                   8695
    PAGE 686  ''PROGRAM  OLDER.VERSION              986
    PAGE 688                                       1064
    PAGE 708                                       2236
MU.REINF.IND
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES     1539
    PAGE 347  EVENT ACT.REINF                      5836
    PAGE 534  ROUTINE UNIT.INPUT                   4741 4742 4742
    PAGE 686  ''PROGRAM  OLDER.VERSION              978
MU.TF.LDR
    PAGE 142  ROUTINE DEAD.UNIT                    6552 6555 6560
    PAGE 147  ROUTINE INTER.BATTLE                 6815 6831 6837
    PAGE 148                                       6868 6874
```

```
MU.TF.LIST
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1546 1549
    PAGE  49  ''SECTION FOR DEFINITIONS                 2816
    PAGE  66  ROUTINE FORM.TF.LIST                      3430 3438
    PAGE  87  ROUTINE END.MOVE                          4321
    PAGE 113  ROUTINE PREPARE.LIST                      5428
    PAGE 125  ROUTINE WHAT.NEXT                         5897 5904 5905 5916 5918
    PAGE 130  ROUTINE CHECK.DEAD                        6102
    PAGE 137  ROUTINE CHECK.LIST                        6363
    PAGE 138  ROUTINE CHECK.PROX                        6424
    PAGE 139                                            6440
    PAGE 141  ROUTINE CHECK.STREN                       6514
    PAGE 142  ROUTINE DEAD.UNIT                         6560 6563 6565
    PAGE 143                                            6579 6590 6592 6594
    PAGE 147  ROUTINE INTER.BATTLE                      6831 6833 6835 6837
    PAGE 148                                            6868 6870 6872 6874
    PAGE 403  EVENT START.MOVE                          8233
    PAGE 405                                            8293 8322
    PAGE 408  EVENT UPDATE.LOC                          8459
    PAGE 410                                            8548
    PAGE 412  EVENT ACT.ATK                             8636
    PAGE 413                                            8661
    PAGE 414  EVENT ACT.DEF                             8728
    PAGE 415  EVENT ACT.MOVCOR                          8751
    PAGE 686  ''PROGRAM  OLDER.VERSION                   985  988
    PAGE 708                                            2237
MU.TF.MEM
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1541
    PAGE  61  ROUTINE FEBA.INITIAL                      3212
    PAGE  66  ROUTINE FORM.TF.LIST                      3432 3439
    PAGE 104  ROUTINE MINE.DELAY                        5041 5042
    PAGE 112  ROUTINE PREPARE.LIST                      5370 5376
    PAGE 119  ROUTINE RESET.FEBA.SECTOR                 5678
    PAGE 125  ROUTINE WHAT.NEXT                         5894 5898 5906 5919 5930
    PAGE 142  ROUTINE DEAD.UNIT                         6547 6548 6562
    PAGE 143                                            6589 6593
    PAGE 144                                            6654
    PAGE 147  ROUTINE INTER.BATTLE                      6806 6832 6836
    PAGE 148                                            6869 6873
    PAGE 347  EVENT ACT.REINF                           5837
    PAGE 412  EVENT ACT.ATK                             8629 8630
    PAGE 539  ROUTINE READ.ORDERS                       5026
    PAGE 686  ''PROGRAM  OLDER.VERSION                   980
MU.UNIT.ID
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1544
    PAGE  87  ROUTINE END.MOVE                          4325 4328 4330 4332 4335
    PAGE 113  ROUTINE PREPARE.LIST                      5404 5430
    PAGE 137  ROUTINE CHECK.LIST                        6354 6370
    PAGE 139  ROUTINE CHECK.PROX                        6443
    PAGE 141  ROUTINE CHECK.STREN                       6517
    PAGE 142  ROUTINE DEAD.UNIT                         6567
    PAGE 143                                            6580
    PAGE 147  ROUTINE INTER.BATTLE                      6839
    PAGE 148                                            6876
    PAGE 404  EVENT START.MOVE                          8237 8242
    PAGE 405                                            8296 8324
```

```
    PAGE 408  EVENT UPDATE.LOC                          8461
    PAGE 409                                            8462 8465 8470
    PAGE 410                                            8551
    PAGE 412  EVENT ACT.ATK                             8639
    PAGE 413                                            8664
    PAGE 414  EVENT ACT.DEF                             8730 8731
    PAGE 415  EVENT ACT.MOVCOR                          8753
    PAGE 534  ROUTINE UNIT.INPUT                        4740
    PAGE 686  ''PROGRAM  OLDER.VERSION                   983
MV.UNIT
    PAGE  40  ''SECTION FOR EVENTS                      2291
    PAGE 114  ROUTINE PREP.WITHDRAW                     5482
    PAGE 173  ROUTINE DUST.EFFECTS                      7910
    PAGE 174                                            7974 7989
    PAGE 235  ROUTINE SMOKE.EFFECTS                      798
    PAGE 236                                             898
    PAGE 261  ROUTINE BTRY.EFFECTS                      2068
    PAGE 328  ROUTINE EMPTY                             5184
    PAGE 479  PROCESS WITH.DRAW                         2011
    PAGE 699  ''PROGRAM  OLDER.VERSION                  1728
M.AATT.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1097
    PAGE 679  ''PROGRAM  OLDER.VERSION                   538
M.AA.SET
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1121
    PAGE 679  ''PROGRAM  OLDER.VERSION                   560
M.AO.CAND.DET.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1132
    PAGE 679  ''PROGRAM  OLDER.VERSION                   571
M.AO.DET.TGT.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1053
    PAGE 678  ''PROGRAM  OLDER.VERSION                   495
M.AO.EB.SET
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES           305
    PAGE 665  ''PROGRAM  OLDER.VERSION                  9747
M.AO.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1402
    PAGE 684  ''PROGRAM  OLDER.VERSION                   841
M.AO.RB.SET
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES           318
    PAGE 665  ''PROGRAM  OLDER.VERSION                  9760
M.AR.CAND.DET.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1142
    PAGE 679  ''PROGRAM  OLDER.VERSION                   581
M.AR.DET.TGT.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1054
    PAGE 678  ''PROGRAM  OLDER.VERSION                   496
M.AVAIL.AO.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1851
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1290
M.BATTLE.SET
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES          1175
    PAGE 680  ''PROGRAM  OLDER.VERSION                   614
M.BN.BTRY.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES           347
    PAGE 666  ''PROGRAM  OLDER.VERSION                  9789
```

```
M.BN.CAN FM.SET
     PAGE  36   ''SECTION FOR PROCESSES                2028
     PAGE 695   ''PROGRAM  OLDER.VERSION               1466
M.BTL.FORCE.SET
     PAGE  25   ''SECTION FOR TEMPORARY_ENTITIES       1421
     PAGE 684   ''PROGRAM  OLDER.VERSION                860
M.BY.FM.QUEUE
     PAGE  36   ''SECTION FOR PROCESSES                2030
     PAGE 695   ''PROGRAM  OLDER.VERSION               1468
M.BY.HOW.SET
     PAGE  27   ''SECTION FOR TEMPORARY_ENTITIES       1495
     PAGE 686   ''PROGRAM  OLDER.VERSION                934
M.BY.SCHD.LIST
     PAGE  36   ''SECTION FOR PROCESSES                2029
     PAGE 695   ''PROGRAM  OLDER.VERSION               1467
M.CFPS.LIST
     PAGE  22   ''SECTION FOR TEMPORARY_ENTITIES       1209
     PAGE 681   ''PROGRAM  OLDER.VERSION                648
M.CF.OP.Q
     PAGE  22   ''SECTION FOR TEMPORARY_ENTITIES       1221
     PAGE 681   ''PROGRAM  OLDER.VERSION                660
M.COL.SET
     PAGE  21   ''SECTION FOR TEMPORARY_ENTITIES       1191
     PAGE 680   ''PROGRAM  OLDER.VERSION                630
M.CT.TU.SET
     PAGE  17   ''SECTION FOR PERMANENT_ENTITIES        954
     PAGE 676   ''PROGRAM  OLDER.VERSION                396
M.DF.RATE.LIST
     PAGE  22   ''SECTION FOR TEMPORARY_ENTITIES       1255
     PAGE 681   ''PROGRAM  OLDER.VERSION                694
M.FD.BN.LIST
     PAGE  24   ''SECTION FOR TEMPORARY_ENTITIES       1355
     PAGE 683   ''PROGRAM  OLDER.VERSION                794
M.FD.COMPLETE.LIST
     PAGE  39   ''SECTION FOR PROCESSES                2192
     PAGE 698   ''PROGRAM  OLDER.VERSION               1630
M.FD.SCHD.LIST
     PAGE  24   ''SECTION FOR TEMPORARY_ENTITIES       1366
     PAGE 683   ''PROGRAM  OLDER.VERSION                805
M.FD.TR.QUEUE
     PAGE  38   ''SECTION FOR PROCESSES                2191
     PAGE 698   ''PROGRAM  OLDER.VERSION               1629
M.FO.CAND.DET.LIST
     PAGE  25   ''SECTION FOR TEMPORARY_ENTITIES       1431
     PAGE 684   ''PROGRAM  OLDER.VERSION                870
M.FO.CUR.FM.LIST
     PAGE  36   ''SECTION FOR PROCESSES                2031
     PAGE 695   ''PROGRAM  OLDER.VERSION               1469
M.FO.TGT.RPT.LIST
     PAGE  39   ''SECTION FOR PROCESSES                2193
     PAGE 698   ''PROGRAM  OLDER.VERSION               1631
M.FP.SET
     PAGE  24   ''SECTION FOR TEMPORARY_ENTITIES       1339
     PAGE 683   ''PROGRAM  OLDER.VERSION                778
M.FR.UNIT.SET
     PAGE  19   ''SECTION FOR PERMANENT_ENTITIES       1052
```

Missing 946

```
M.MU.TF.LIST
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES        1559
    PAGE 687  ''PROGRAM  OLDER.VERSION                 998
M.PATH.SET
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1670
    PAGE 689  ''PROGRAM  OLDER.VERSION                1109
M.PDB.KEYED.LIST
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES        1515
    PAGE 686  ''PROGRAM  OLDER.VERSION                 954
M.PDB.OP.Q
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES        1660
    PAGE 688  ''PROGRAM  OLDER.VERSION                1099
M.PIR.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES        1403
    PAGE 684  ''PROGRAM  OLDER.VERSION                 842
M.PIR.RECORD.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1681
    PAGE 689  ''PROGRAM  OLDER.VERSION                1120
M.PIR.RTD.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1695
    PAGE 689  ''PROGRAM  OLDER.VERSION                1134
M.RPV.CAND.DET.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1720
    PAGE 689  ''PROGRAM  OLDER.VERSION                1159
M.RPV.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES        1401
    PAGE 684  ''PROGRAM  OLDER.VERSION                 840
M.SD.ADS.SET
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES        1111
    PAGE 679  ''PROGRAM  OLDER.VERSION                 550
M.SD.CMSN.QUEUE
    PAGE  35  ''SECTION FOR PROCESSES                 1992
    PAGE 694  ''PROGRAM  OLDER.VERSION                1430
M.SD.FPO.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES        1443
    PAGE 685  ''PROGRAM  OLDER.VERSION                 882
M.SD.KAS.SET
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES        1524
    PAGE 686  ''PROGRAM  OLDER.VERSION                 963
M.SD.OLD.SORTIE.QUEUE
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES        1619
    PAGE 688  ''PROGRAM  OLDER.VERSION                1058
M.SIDE.CFR.SET
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1235
    PAGE 681  ''PROGRAM  OLDER.VERSION                 674
M.SIDE.PDB.SET
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES        1648
    PAGE 688  ''PROGRAM  OLDER.VERSION                1087
M.SI.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES        1735
    PAGE 690  ''PROGRAM  OLDER.VERSION                1174
M.SO.LIST
    PAGE  38  ''SECTION FOR PROCESSES                 2145
    PAGE 697  ''PROGRAM  OLDER.VERSION                1583
M.SS.SET
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES        1050
```

```
    PAGE 119  ROUTINE RESET.FEBA.SECTOR                5679
    PAGE 331  ROUTINE ENQ.FEBA.SET                     5322
    PAGE 678  ''PROGRAM  OLDER.VERSION                  492
M.TB.SORT.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         1019
    PAGE 677  ''PROGRAM  OLDER.VERSION                  461
M.TB.TM.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES         1758
    PAGE 690  ''PROGRAM  OLDER.VERSION                 1197
M.TEAM.TYPES
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES         1806
    PAGE 691  ''PROGRAM  OLDER.VERSION                 1245
M.TE.SET
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES          421
    PAGE 667  ''PROGRAM  OLDER.VERSION                 9862
M.TR.DET.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES         1769
    PAGE 690  ''PROGRAM  OLDER.VERSION                 1208
M.TR.FM.LIST
    PAGE  36  ''SECTION FOR PROCESSES                  2027
    PAGE 695  ''PROGRAM  OLDER.VERSION                 1465
M.TU.NTE.SET
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES         1778
    PAGE 690  ''PROGRAM  OLDER.VERSION                 1217
M.TU.TE.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES         1789
    PAGE 691  ''PROGRAM  OLDER.VERSION                 1228
M.U
    PAGE 139  ROUTINE CHECK.PROX                       6440 6443
M.UE.TARGET.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES         1383
    PAGE 684  ''PROGRAM  OLDER.VERSION                  822
M.UE.WEAPON.SET
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES         1885
    PAGE 692  ''PROGRAM  OLDER.VERSION                 1324
M.UNIT.SET
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES         1055
    PAGE 534  ROUTINE UNIT.INPUT                       4749
    PAGE 678  ''PROGRAM  OLDER.VERSION                  497
M.UN.EQUIP.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES         1823
    PAGE 691  ''PROGRAM  OLDER.VERSION                 1262
M.UN.HC.LOS.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES         1873
    PAGE 692  ''PROGRAM  OLDER.VERSION                 1312
M.UN.LOS.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES         1867
    PAGE 692  ''PROGRAM  OLDER.VERSION                 1306
M.UN.PATH
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES         1705
    PAGE 689  ''PROGRAM  OLDER.VERSION                 1144
M.UN.SEGMENT.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES         1746
    PAGE 690  ''PROGRAM  OLDER.VERSION                 1185
M.UN.SENSOR.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES         1850
```

```
    PAGE 692  ''PROGRAM  OLDER.VERSION                1289
M.UN.SUB.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES        1051
    PAGE 678  ''PROGRAM  OLDER.VERSION                 493
NECESSARY.
    PAGE 450  PROCESS HC.RETURN.FARRP                  467
NEWX.
    PAGE 408  EVENT UPDATE.LOC                        8440 8443 8448 8453 8461
    PAGE 410                                          8536 8574
NEWY.
    PAGE 408  EVENT UPDATE.LOC                        8441 8444 8449 8453
    PAGE 409                                          8462
    PAGE 410                                          8537 8574
NEW.BATS
    PAGE  63  ROUTINE FILE.FD.SCHD                    3280 3285 3285 3291 3304 3308 3314
    PAGE  64                                          3324 3330 3344 3354
NEW.BLUE
    PAGE 146  ROUTINE INTER.BATTLE                    6748
    PAGE 147                                          6846 6851 6852
    PAGE 148                                          6883 6888 6889
    PAGE 356  EVENT BTL.ENDED                         6262
    PAGE 357                                          6327 6333 6338
NEW.BTL
    PAGE 128  ROUTINE BTL.CHECK                       5998 6021 6040
    PAGE 129                                          6045 6046 6048 6049 6051 6052 6057 6060 6061
NEW.COUNT
    PAGE 130  ROUTINE CHECK.DEAD                      6078 6093 6109 6114 6120
NEW.ITEM
    PAGE  63  ROUTINE FILE.FD.SCHD                    3282 3283 3284 3285 3286
    PAGE  64                                          3345 3346 3347 3348 3349
NEW.ITEM1
    PAGE  63  ROUTINE FILE.FD.SCHD                    3293 3294 3295 3296 3297 3311 3312 3313 3314 3315
    PAGE  64                                          3331 3333 3335 3337 3339
NEW.ITEM2
    PAGE  63  ROUTINE FILE.FD.SCHD                    3301 3302 3303 3304 3305
    PAGE  64                                          3332 3334 3336 3338 3340
NEW.LEG
    PAGE 429  PROCESS AIR.OBSERVER                    9445
    PAGE 430                                          9446 9447 9448 9449 9450 9451
NEW.MISSION
    PAGE 164  ROUTINE BTRY.FM.ENQ                     7544 7551 7552 7563 7564 7566
NEW.MU
    PAGE 143  ROUTINE DEAD.UNIT                       6581 6582 6586 6589 6594 6596 6597 6601
NEW.PARENT
    PAGE 143  ROUTINE DEAD.UNIT                       6623 6625 6628 6632 6633 6634 6635 6635
    PAGE 144                                          6640 6641 6643
NEW.PRIORITY
    PAGE 128  ROUTINE BTL.CHECK                       5997 6018 6020 6037 6039
NEW.RED
    PAGE 146  ROUTINE INTER.BATTLE                    6748
    PAGE 147                                          6841 6852 6852
    PAGE 148                                          6878 6889 6889
    PAGE 356  EVENT BTL.ENDED                         6262
    PAGE 357                                          6329 6336 6339
NEW.SECTOR
    PAGE  84  ROUTINE CHANGE.LOC                      4197
```

```
    PAGE  85                                          4261
    PAGE 110  ROUTINE POSITION                        5274 5295
NEW.START
    PAGE  63  ROUTINE FILE.FD.SCHD                    3260 3265 3275 3283 3289 3298 3302 3309 3312
    PAGE  64                                          3326 3328 3335 3343 3347
NEW.STOP
    PAGE  63  ROUTINE FILE.FD.SCHD                    3260 3265 3281 3284 3290 3292 3294
    PAGE  64                                          3318 3324 3327 3329 3334 3350 3354
    PAGE 470  PROCESS TARGET.REPORT                   1494
    PAGE 476                                          1848 1853
NEW.TARGET
    PAGE 333  ROUTINE FDC.TR.ENQ                      5359 5368 5371 5372 5374
NEW.TF
    PAGE 142  ROUTINE DEAD.UNIT                       6564 6568 6572 6574 6576
    PAGE 143                                          6579 6580 6582
NEW.UNITS
    PAGE 130  ROUTINE CHECK.DEAD                      6068 6077 6116 6120
    PAGE 131                                          6125
NEW.UNIT.LIST
    PAGE 146  ROUTINE INTER.BATTLE                    6748
    PAGE 147                                          6819 6822 6830 6839 6846
    PAGE 148                                          6856 6859 6867 6876 6878
NEW.UN.FWD.ADV
    PAGE 331  ROUTINE ENQ.FEBA.SET                    5317 5319 5321 5328
NEXT.
    PAGE  87  ROUTINE END.MOVE                        4364
    PAGE  88                                          4367 4377 4384
    PAGE 117  ROUTINE PROX.POS                        5583 5585 5591
    PAGE 118  ROUTINE REIN.ARRIVE                     5629 5634
    PAGE 126  ROUTINE WHAT.NEXT                       5938 5940 5945 5947 5964
    PAGE 348  EVENT ACT.REINF                         5881 5891
    PAGE 373  EVENT GET.NX.ORD                        6878 6907 6910
    PAGE 374                                          6979 6982
    PAGE 408  EVENT UPDATE.LOC                        8421
NEXT.UNIT
    PAGE 362  EVENT CFR.OPERATOR                      6484
    PAGE 363                                          6585 6589
    PAGE 388  EVENT PDB.OPERATOR                      7563 7564
NITE.DAY.FACTOR
    PAGE 106  ROUTINE MIN.MOVE                        5069 5084 5086 5088
NITE.MOV.FAC
    PAGE  48  ''SECTION FOR DEFINITIONS               2747
    PAGE  80  ROUTINE BLOCK.LOS                       4025
    PAGE  83  ROUTINE CHANGE.LOC                      4151
    PAGE 106  ROUTINE MIN.MOVE                        5086
    PAGE 210  ROUTINE PGM.MSN.ASGN                    9624
    PAGE 394  EVENT START.ARTY.MOVEMENT               7713
    PAGE 403  EVENT START.MOVE                        8206
    PAGE 523  ROUTINE SYS.INPUT                       4370
    PAGE 707  ''PROGRAM  OLDER.VERSION                2169
NITE.OR.DAY
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES         253
    PAGE  12                                           657
    PAGE  13                                           697
    PAGE  15                                           800
    PAGE  56  ROUTINE MAIN1                           3048
```

```
    PAGE  57  ROUTINE MAIN3                          3074
    PAGE 200  ROUTINE ILLUM.EFFECTS                  9195
    PAGE 201                                         9264 9296
    PAGE 206  ROUTINE PDB.DETECTION                  9449 9456
    PAGE 210  ROUTINE PGM.MSN.ASGN                   9623
    PAGE 218  ROUTINE REQUEST.ILLUM                  9995
    PAGE 234  ROUTINE SMOKE.EFFECTS                   774  782
    PAGE 255  ROUTINE FO.DETECTION                   1753 1768
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS          3420
    PAGE 323  ROUTINE CONTRAST.TO.FREQ               5014
    PAGE 365  EVENT CHANGE.LITE                      6598 6599 6605
    PAGE 366  EVENT CHANGE.WEATHER                   6628 6630
    PAGE 370  EVENT ENGAGEMENT                       6769
    PAGE 393  EVENT START.ARTY.MOVEMENT              7712
    PAGE 403  EVENT START.MOVE                       8205
    PAGE 478  PROCESS WITH.DRAW                      1948
    PAGE 538  ROUTINE MFO.INPUT                      4972
    PAGE 563  ROUTINE MPDB.INPUT                     5942
    PAGE 664  ''PROGRAM  OLDER.VERSION               9695
    PAGE 671                                           99
    PAGE 672                                          140
    PAGE 674                                          242
NITE.VIS.PCT
    PAGE  43  ''SECTION FOR DEFINITIONS              2432
    PAGE 255  ROUTINE FO.DETECTION                   1754
    PAGE 366  EVENT CHANGE.WEATHER                   6628 6630
    PAGE 581  ROUTINE VIS.INPUT                      6717
    PAGE 702  ''PROGRAM  OLDER.VERSION               1862
NON.RAP
    PAGE 182  ROUTINE FA.BN.ASGN                     8306
    PAGE 184                                         8426
NORMAL.F
    PAGE 157  ROUTINE AO.DETECTION                   7273 7275 7277
    PAGE 193  ROUTINE FINAL.COVERAGE                 8878 8879 8880
    PAGE 230  ROUTINE RPV.DETECTION                   596  598  600
    PAGE 242  ROUTINE UNIT.ENVIR                     1154 1155
    PAGE 243                                         1213 1214
    PAGE 244                                         1229 1230 1240 1241 1263 1264 1272 1273
    PAGE 257  ROUTINE FO.DETECTION                   1874 1875 1876
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2842 2843 2844
    PAGE 351  EVENT AD.ENGAGEMENT                    6041 6042
    PAGE 362  EVENT CFR.OPERATOR                     6517 6519 6521
    PAGE 387  EVENT PDB.OPERATOR                     7535 7536 7538
    PAGE 641  ROUTINE NORMAL.F                       9039
    PAGE 656  ROUTINE AR.DETECTION                   9429 9431 9433
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                9577 9578 9580
NORTH
    PAGE   1  ROUTINE FOR CROSS_REFERENCING            56
    PAGE  51  ''SECTION FOR SUBSTITUTIONS            2899
    PAGE 709  ''PROGRAM  OLDER.VERSION               2307
NOT.READY
    PAGE  53  ''SECTION FOR SUBSTITUTIONS            3010
NOT.THIS.TU
    PAGE 240  ROUTINE TARGET.ANALYSIS                1045 1051 1055 1059
NO.
    PAGE 573  ROUTINE FARRP.INPUT                    6411
```

```
NO.AH.TEAM
    PAGE 714  ROUTINE PLAT.COUNT                    2512 2536
NO.ARMOR.PLAT
    PAGE 714  ROUTINE PLAT.COUNT                    2511 2532
NO.ATTACK
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS            3509 3518 3519 3522
    PAGE 290                                        3538 3539
    PAGE 291                                        3622 3624
    PAGE 450  PROCESS HC.RETURN.FARRP                480  487  505  507
    PAGE 451                                         525  526  537  537  563  564  572  574
    PAGE 452                                         627
    PAGE 453                                         641
NO.ATTACK.CONFIGURATIONS
    PAGE  42  ''SECTION FOR DEFINITIONS             2386
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS            3506
    PAGE 297  ROUTINE FARRP.CHECK                   3882
    PAGE 452  PROCESS HC.RETURN.FARRP                594
    PAGE 572  ROUTINE FARRP.INPUT                   6347 6348 6349
    PAGE 701  ''PROGRAM  OLDER.VERSION              1816
NO.ATT.HC
    PAGE 446  PROCESS HC.ARRIVE.BATTLE               294
NO.AUTH
    PAGE 141  ROUTINE CHECK.STREN                   6513 6517 6519
NO.BLUE.AH.TEAM
    PAGE 396  EVENT START.BATTLE                    7827
NO.BLUE.ARMOR.PLAT
    PAGE 396  EVENT START.BATTLE                    7826
    PAGE 397                                        7869 7888
NO.BLUE.HQ
    PAGE  68  ROUTINE GENERAL.BATTLE                3543
    PAGE  69                                        3582 3587
    PAGE 396  EVENT START.BATTLE                    7828
    PAGE 397                                        7875 7894
NO.BLUE.INF.PLAT
    PAGE 396  EVENT START.BATTLE                    7824
    PAGE 397                                        7873 7892
NO.BLUE.MAN
    PAGE  68  ROUTINE GENERAL.BATTLE                3546
    PAGE  69                                        3583 3589
NO.BLUE.MECH.PLAT
    PAGE 396  EVENT START.BATTLE                    7825
    PAGE 397                                        7871 7890
NO.BLUE.UNITS
    PAGE  67  ROUTINE GENERAL.BATTLE                3458 3483 3488 3489
    PAGE  68                                        3518 3523
    PAGE  71  ROUTINE ORIENTATION                   3673 3681 3682
    PAGE  72                                        3719 3720
    PAGE  96  ROUTINE LINE.OF.SIGHT                 4679 4681
    PAGE 395  EVENT START.BATTLE                    7762 7787
    PAGE 397                                        7864 7883
    PAGE 399                                        8013
NO.ENEMY.UNITS
    PAGE 300  ROUTINE HC.DISENGAGE                  4017 4033 4054
    PAGE 301                                        4084
NO.FLY.VIS
    PAGE  52  ''SECTION FOR SUBSTITUTIONS           2990
```

```
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS          3418
    PAGE 711  ''PROGRAM  OLDER.VERSION               2397
NO.GER.PLAIN
    PAGE  52  ''SECTION FOR SUBSTITUTIONS            2968
    PAGE 336  ROUTINE GET.TERRAIN                    5445
    PAGE 710  ''PROGRAM  OLDER.VERSION               2375
NO.HC
    PAGE 447  PROCESS HC.ARRIVE.BATTLE                382
NO.HQ
    PAGE 714  ROUTINE PLAT.COUNT                     2513 2540
NO.INF.PLAT
    PAGE 714  ROUTINE PLAT.COUNT                     2509 2524
NO.MECH.PLAT
    PAGE 714  ROUTINE PLAT.COUNT                     2510 2528
NO.OF.FARRPS
    PAGE 572  ROUTINE FARRP.INPUT                    6369
    PAGE 573                                         6394 6417
NO.ON.HAND
    PAGE 141  ROUTINE CHECK.STREN                    6512 6516 6519
NO.PATH.POINTS
    PAGE  60  ROUTINE CREATE.TEAMS                   3171 3172
NO.PK.PTR
    PAGE  52  ''SECTION FOR SUBSTITUTIONS            3005
    PAGE 307  ROUTINE AD.SHOOT                       4354
    PAGE 354  EVENT AD.ENGAGEMENT                    6199
    PAGE 422  PROCESS AC.ATK.TGT                     9046
    PAGE 424                                         9164
    PAGE 711  ''PROGRAM  OLDER.VERSION               2412
NO.RED.AH.TEAM
    PAGE 396  EVENT START.BATTLE                     7838
NO.RED.ARMOR.PLAT
    PAGE 396  EVENT START.BATTLE                     7837
    PAGE 397                                         7868 7887
NO.RED.HQ
    PAGE  69  ROUTINE GENERAL.BATTLE                 3575 3584 3591
    PAGE 396  EVENT START.BATTLE                     7839
    PAGE 397                                         7874 7893
NO.RED.INF.PLAT
    PAGE 396  EVENT START.BATTLE                     7835
    PAGE 397                                         7872 7891
NO.RED.MAN
    PAGE  69  ROUTINE GENERAL.BATTLE                 3578 3585 3593
NO.RED.MECH.PLAT
    PAGE 396  EVENT START.BATTLE                     7836
    PAGE 397                                         7870 7889
NO.RED.UNITS
    PAGE  67  ROUTINE GENERAL.BATTLE                 3459 3491 3496 3497
    PAGE  68                                         3550 3555
    PAGE  71  ROUTINE ORIENTATION                    3677 3683 3684
    PAGE  72                                         3721 3722
    PAGE  96  ROUTINE LINE.OF.SIGHT                  4680 4683
    PAGE 395  EVENT START.BATTLE                     7763 7787
    PAGE 397                                         7865 7884
    PAGE 399                                         8013
NO.SCOUT
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS             3500 3518 3519
```

```
    PAGE 290                                        3552 3553 3574 3575
    PAGE 291                                        3619 3621
    PAGE 450  PROCESS HC.RETURN.FARRP                483  487  505  507
    PAGE 451                                         539  540  551  551  563  572  574
NO.SCOUT.CONFIGURATIONS
    PAGE  42  ''SECTION FOR DEFINITIONS             2387
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS            3497
    PAGE 297  ROUTINE FARRP.CHECK                   3875
    PAGE 452  PROCESS HC.RETURN.FARRP                602
    PAGE 572  ROUTINE FARRP.INPUT                   6339 6340 6341
    PAGE 701  ''PROGRAM  OLDER.VERSION              1817
NO.SHOT.MADE
    PAGE  52  ''SECTION FOR SUBSTITUTIONS           3004
    PAGE 308  ROUTINE AD.SHOOT                      4360
    PAGE 422  PROCESS AC.ATK.TGT                    9047
    PAGE 424                                        9164
    PAGE 711  ''PROGRAM  OLDER.VERSION              2411
NO.TF
    PAGE 112  ROUTINE PREPARE.LIST                  5367 5381 5400
    PAGE 113                                        5442 5443 5444 5446 5451 5453
    PAGE 137  ROUTINE CHECK.LIST                    6332 6343 6351
NO.TGTS
    PAGE 488  PROCESS ASSESSMENT                    2491 2492
NO.UNITS
    PAGE  58  ROUTINE CREATE.FORCE                  3104 3113
    PAGE  75  ROUTINE UNIT.ASSIGNMENT               3837 3838
    PAGE  77  ROUTINE ADJUST                        3894 3897 3907
    PAGE 714  ROUTINE PLAT.COUNT                    2518 2520
NO.VICTIM
    PAGE 488  PROCESS ASSESSMENT                    2496 2497
NO.VICTIMS
    PAGE 517  PROCESS HELICOPTER.FIRE               4067
NR.
    PAGE 380  EVENT INIT.PREPLAN.CAS                7257
NUMBER.DETECTED
    PAGE 157  ROUTINE AO.DETECTION                  7257 7259 7263 7265 7288 7290
NUMBER.IN.LOS
    PAGE 255  ROUTINE FO.DETECTION                  1775 1776
NUMBER.OF.LEGS
    PAGE 429  PROCESS AIR.OBSERVER                  9440 9443
    PAGE 653  PROCESS AIRBORNE.RADAR                9304 9309
NUMBER.VOLLEYS
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS              1375
    PAGE 249                                        1440 1454
NUM.
    PAGE 294  ROUTINE END.CAS.MISSION               3771
NUM.ABORTED
    PAGE 295  ROUTINE END.CAS.MISSION               3817
NUM.AC
    PAGE 295  ROUTINE END.CAS.MISSION               3817
NUM.ANG.INCREMENTS
    PAGE  41  ''SECTION FOR DEFINITIONS             2358
    PAGE 335  ROUTINE FRAC.COMPUTE                  5423 5424
    PAGE 523  ROUTINE SYS.INPUT                     4360
    PAGE 700  ''PROGRAM  OLDER.VERSION              1788
```

```
NUM.AO.SORTIE.PER.SECTOR.PER.DAY
    PAGE 567  ROUTINE SENSOR.INPUT                  6178 6179 6182
NUM.DET.PGM
    PAGE 157  ROUTINE AO.DETECTION                  7290 7292 7298
NUM.DET.SDM
    PAGE 257  ROUTINE FO.DETECTION                  1899 1900
    PAGE 258                                        1941 1942
NUM.KEYED.SENSORS
    PAGE 567  ROUTINE SENSOR.INPUT                  6134 6135
NUM.KILL.PGM
    PAGE  41  ''SECTION FOR DEFINITIONS             2355
    PAGE 270  ROUTINE BTRY.EFFECTS                  2555
    PAGE 437  PROCESS ARTY.ASSESS                   9822
    PAGE 593  ROUTINE AMMO.RPT                      7221
    PAGE 700  ''PROGRAM  OLDER.VERSION              1785
NUM.PGM.FIRED
    PAGE  41  ''SECTION FOR DEFINITIONS             2354
    PAGE 264  ROUTINE BTRY.EFFECTS                  2205
    PAGE 593  ROUTINE AMMO.RPT                      7193 7194 7198
    PAGE 700  ''PROGRAM  OLDER.VERSION              1784
NUM.POSITION.REPORT
    PAGE 523  ROUTINE SYS.INPUT                     4325 4340
NUM.RAD.INCREMENTS
    PAGE  41  ''SECTION FOR DEFINITIONS             2357
    PAGE 523  ROUTINE SYS.INPUT                     4359
    PAGE 700  ''PROGRAM  OLDER.VERSION              1787
NUM.RH
    PAGE 538  ROUTINE MFO.INPUT                     4952 4953 4962
    PAGE 562  ROUTINE MCFR.INPUT                    5900 5901 5910
    PAGE 563  ROUTINE MPDB.INPUT                    5933 5934 5948
NUM.ROUNDS
    PAGE 260  ROUTINE BTRY.EFFECTS                  1985
    PAGE 263                                        2180 2190 2191
    PAGE 264                                        2198 2201 2205 2214
    PAGE 268                                        2454
    PAGE 272                                        2681 2695
NUM.SECT
    PAGE 138  ROUTINE CHECK.PROX                    6397 6399 6400 6404 6405 6409 6410 6412 6420
NUM.SENSOR
    PAGE 565  ROUTINE SENSOR.INPUT                  6052
    PAGE 566                                        6072 6093 6100 6100 6101
NX.ORDER
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES      1598
    PAGE  87  ROUTINE END.MOVE                      4364
    PAGE 546  ROUTINE ORD.MOVCOR                    5239
    PAGE 687  ''PROGRAM  OLDER.VERSION              1037
NX.ORD.ABOVE
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES      1599
    PAGE 117  ROUTINE PROX.POS                      5583
    PAGE 146  ROUTINE INTER.BATTLE                  6760
    PAGE 546  ROUTINE ORD.MOVCOR                    5240
    PAGE 630  FUNCTION COLLISION                    8694
    PAGE 687  ''PROGRAM  OLDER.VERSION              1038
NX.ORD.BELOW
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES      1600
    PAGE 117  ROUTINE PROX.POS                      5585
```

```
    PAGE 146  ROUTINE INTER.BATTLE                    6762
    PAGE 546  ROUTINE ORD.MOVCOR                      5241
    PAGE 687  ''PROGRAM  OLDER.VERSION                1039
N.AATT.LIST
    PAGE  34  ''SECTION FOR PROCESSES                 1904
    PAGE 693  ''PROGRAM  OLDER.VERSION                1342
N.AA.SET
    PAGE  34  ''SECTION FOR PROCESSES                 1952
    PAGE 693  ''PROGRAM  OLDER.VERSION                1390
N.AC.MUNS
    PAGE 588  ROUTINE AC.MUNS.INPUT                   7016 7048 7050 7064
N.AC.TYPE
    PAGE 283  ROUTINE CAS.EVAL                        3175
    PAGE 582  ROUTINE TACAIR.INPUT                    6792
N.AO.CAND.DET.LIST
    PAGE  34  ''SECTION FOR PROCESSES                 1940
    PAGE 693  ''PROGRAM  OLDER.VERSION                1378
N.AO.DET.TGT.LIST
    PAGE  34  ''SECTION FOR PROCESSES                 1941
    PAGE 693  ''PROGRAM  OLDER.VERSION                1379
N.AO.ELEVATION.BAND
    PAGE 564  ROUTINE MAO.INPUT                       5978
N.AO.FLIGHT.LEG.LIST
    PAGE  34  ''SECTION FOR PROCESSES                 1939
    PAGE 693  ''PROGRAM  OLDER.VERSION                1377
N.AO.RANGE.BAND
    PAGE 564  ROUTINE MAO.INPUT                       5988
N.AR.CAND.DET.LIST
    PAGE  34  ''SECTION FOR PROCESSES                 1920
    PAGE 693  ''PROGRAM  OLDER.VERSION                1358
N.AR.DET.TGT.LIST
    PAGE  34  ''SECTION FOR PROCESSES                 1919
    PAGE 693  ''PROGRAM  OLDER.VERSION                1357
N.AU.LIST
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         579
    PAGE 670  ''PROGRAM  OLDER.VERSION                  20
N.AVAIL.AO.LIST
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES         769
    PAGE 673  ''PROGRAM  OLDER.VERSION                 211
N.BLUE.TYPE.EQP
    PAGE  42  ''SECTION FOR DEFINITIONS               2401
    PAGE 250  ROUTINE MINE.EFFECTS                    1503 1505
    PAGE 281  ROUTINE AC.DF.EFFECTS                   3056
    PAGE 310  ROUTINE AD.SHOOT                        4476
    PAGE 384  EVENT OFF.LINE.ATTRITION                7412
    PAGE 420  PROCESS AC.ATK.TGT                      8927
    PAGE 499  PROCESS SHOOT.OUT                       3118
    PAGE 514  PROCESS HELICOPTER.FIRE                 3931
    PAGE 520  ROUTINE MAIN2                           4212
    PAGE 530  ROUTINE EQ.TE.INPUT                     4594 4596
    PAGE 534  ROUTINE UNIT.INPUT                      4778 4780
    PAGE 536                                          4887
    PAGE 592  ROUTINE AMMO.RPT                        7145
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                 7299 7303
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES             8510 8521 8526 8527 8530 8537
    PAGE 625                                          8562 8573 8583 8602
```

```
    PAGE 626                                        8613
    PAGE 701  ''PROGRAM  OLDER.VERSION              1831
N.BN.BTRY.SET
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES       436
    PAGE 667  ''PROGRAM  OLDER.VERSION              9877
N.BTL.FORCE.SET
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES      1178
    PAGE 680  ''PROGRAM  OLDER.VERSION               617
N.BTRY
    PAGE 550  ROUTINE BTRY.INPUT                    5403 5415 5417
N.BTRY.HOW.SET
    PAGE 182  ROUTINE FA.BN.ASGN                    8337
N.BY.FM.QUEUE
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES       350
    PAGE 164  ROUTINE BTRY.FM.ENQ                   7569
    PAGE 183  ROUTINE FA.BN.ASGN                    8394
    PAGE 184                                        8419
    PAGE 208  ROUTINE PGM.MSN.ASGN                  9525
    PAGE 666  ''PROGRAM  OLDER.VERSION              9792
N.BY.HOW.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES       348
    PAGE 178  ROUTINE EST.COVERAGE                  8172 8176
    PAGE 182  ROUTINE FA.BN.ASGN                    8334 8336
    PAGE 192  ROUTINE FINAL.COVERAGE                8808 8818 8823
    PAGE 193                                        8854
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ          9323
    PAGE 208  ROUTINE PGM.MSN.ASGN                  9517
    PAGE 209                                        9589
    PAGE 211                                        9678
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS              1415
    PAGE 252  ROUTINE MINE.EFFECTS                  1625
    PAGE 260  ROUTINE BTRY.EFFECTS                  2008
    PAGE 262                                        2110
    PAGE 263                                        2142
    PAGE 271                                        2618
    PAGE 278  ROUTINE AC.BOMB.EFFECTS               2957
    PAGE 279                                        2964
    PAGE 282  ROUTINE AC.DF.EFFECTS                 3143
    PAGE 363  EVENT CFR.OPERATOR                    6555
    PAGE 387  EVENT PDB.OPERATOR                    7543
    PAGE 463  PROCESS HOW.REPAIR                    1182
    PAGE 483  PROCESS FIRE.MISSION                  2243 2248
    PAGE 484                                        2284 2287
    PAGE 551  ROUTINE BTRY.INPUT                    5488
    PAGE 611  ROUTINE OUTPUT.ATTRITION              7968
    PAGE 666  ''PROGRAM  OLDER.VERSION              9790
N.BY.SCHD.LIST
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES       349
    PAGE 666  ''PROGRAM  OLDER.VERSION              9791
N.B.WPN.TYPE
    PAGE  42  ''SECTION FOR DEFINITIONS             2403
    PAGE 281  ROUTINE AC.DF.EFFECTS                 3059
    PAGE 310  ROUTINE AD.SHOOT                      4479
    PAGE 383  EVENT OFF.LINE.ATTRITION              7339 7349
    PAGE 384                                        7414
    PAGE 499  PROCESS SHOOT.OUT                     3120
```

```
    PAGE 514  PROCESS HELICOPTER.FIRE                3934
    PAGE 520  ROUTINE MAIN2                          4213
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT              4644 4646
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                7319
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES            8509 8511 8530 8540
    PAGE 625                                         8572 8597
    PAGE 626                                         8616
    PAGE 701  ''PROGRAM  OLDER.VERSION               1833
N.CATEGORY
    PAGE 527  ROUTINE CAT.TU.INPUT                   4459
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                7261
N.CFPS.LIST
    PAGE  35  ''SECTION FOR PROCESSES                1989
    PAGE 316  ROUTINE FLIGHT.PATH                    4774
    PAGE 694  ''PROGRAM  OLDER.VERSION               1427
N.CFR.RNG.HACK
    PAGE 562  ROUTINE MCFR.INPUT                     5883
N.CF.OP.Q
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES       1238
    PAGE 681  ''PROGRAM  OLDER.VERSION                677
N.COLOR
    PAGE 570  ROUTINE DECISION.INPUT                 6265
N.CT.TU.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        363
    PAGE 666  ''PROGRAM  OLDER.VERSION               9805
N.DIST.FROM.FEBA.BAND
    PAGE 560  ROUTINE  RUL.EN.INPUT                  5825
    PAGE 635  FUNCTION FEBA.BAND                     8849
N.ENVIRONMENT
    PAGE 212  ROUTINE PIR.DETECTION                  9711
    PAGE 229  ROUTINE RPV.DETECTION                   523
    PAGE 260  ROUTINE BTRY.EFFECTS                   1998
    PAGE 547  ROUTINE P.E.M.INPUT                    5269
N.EQUIPMENT
    PAGE 250  ROUTINE MINE.EFFECTS                   1506
    PAGE 530  ROUTINE EQ.TE.INPUT                    4596
    PAGE 534  ROUTINE UNIT.INPUT                     4781
    PAGE 609  ROUTINE KV.SCOREBOARD                  7888
N.FAC
    PAGE 149  ROUTINE PK.COMPUTE                     6907
    PAGE 150                                         7002 7004 7006
N.FASCAM.MUNITION
    PAGE 577  ROUTINE MINE.INPUT                     6561 6562
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                7250
    PAGE 596                                         7330
N.FA.BN
    PAGE 550  ROUTINE BTRY.INPUT                     5397 5426 5428
    PAGE 552                                         5521 5523
    PAGE 553  ROUTINE FBN.FD.INPUT                   5564
N.FDC
    PAGE 553  ROUTINE FBN.FD.INPUT                   5548 5551 5581
    PAGE 565  ROUTINE SENSOR.INPUT                   6058
N.FD.BN.LIST
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES        463
    PAGE 668  ''PROGRAM  OLDER.VERSION               9904
```

```
N.FD.COMPLETE.LIST
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES          461
    PAGE 668  ''PROGRAM  OLDER.VERSION                 9902
N.FD.SCHD.LIST
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES          462
    PAGE 668  ''PROGRAM  OLDER.VERSION                 9903
N.FD.TR.QUEUE
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES          460
    PAGE 668  ''PROGRAM  OLDER.VERSION                 9901
N.FO.CAND.DET.LIST
    PAGE  36  ''SECTION FOR PROCESSES                  2050
    PAGE 439  PROCESS FORWARD.OBSERVER                 9947 9953 9954
    PAGE 440                                           9973 9977
    PAGE 695  ''PROGRAM  OLDER.VERSION                 1488
N.FO.CUR.FM.LIST
    PAGE  36  ''SECTION FOR PROCESSES                  2052
    PAGE 482  PROCESS FIRE.MISSION                     2173 2181
    PAGE 483                                           2203
    PAGE 695  ''PROGRAM  OLDER.VERSION                 1490
N.FO.RANGE.BAND
    PAGE 538  ROUTINE MFO.INPUT                        4941
N.FO.TGT.RPT.LIST
    PAGE  36  ''SECTION FOR PROCESSES                  2051
    PAGE 695  ''PROGRAM  OLDER.VERSION                 1489
N.FP.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          760
    PAGE 673  ''PROGRAM  OLDER.VERSION                  202
N.FR.UNIT.SET
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES         1418
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                 3946
    PAGE 305  ROUTINE UNIT.PRIORITY                    4265 4267
    PAGE 684  ''PROGRAM  OLDER.VERSION                  857
N.FUZE
    PAGE 555  ROUTINE MUNS.INPUT                       5612
N.GP.CAT.SET
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES          489
    PAGE 668  ''PROGRAM  OLDER.VERSION                 9930
N.GROUPING
    PAGE  56  ROUTINE MAIN1                            3056
N.HE.MUNITION
    PAGE 555  ROUTINE MUNS.INPUT                       5610
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                  7248
    PAGE 596                                           7330
N.HE.RANGE.HACK
    PAGE 555  ROUTINE MUNS.INPUT                       5621
N.HE.TB.RH.LIST
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES          506
    PAGE 669  ''PROGRAM  OLDER.VERSION                 9947
N.HF.SO.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES         1835
    PAGE 691  ''PROGRAM  OLDER.VERSION                 1274
N.HOW
    PAGE 192  ROUTINE FINAL.COVERAGE                   8826
    PAGE 193                                           8858
    PAGE 209  ROUTINE PGM.MSN.ASGN                     9590
```

```
N.HT.LIST
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1342
    PAGE 304  ROUTINE REPLACE.HC                       4228 4229
    PAGE 452  PROCESS HC.RETURN.FARRP                   587  627
    PAGE 453                                            647  663  671
    PAGE 683  ''PROGRAM  OLDER.VERSION                  781
N.HT.MEMBER.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES          1454
    PAGE 447  PROCESS HC.ARRIVE.BATTLE                  379
    PAGE 448                                            389
    PAGE 495  PROCESS SHOOT.OUT                        2865
    PAGE 515  PROCESS HELICOPTER.FIRE                  3979 4004
    PAGE 517                                           4083
    PAGE 518                                           4142
    PAGE 685  ''PROGRAM  OLDER.VERSION                  893
N.HT.TARGET.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES          1457
    PAGE 685  ''PROGRAM  OLDER.VERSION                  896
N.IC.MUNITION
    PAGE 555  ROUTINE MUNS.INPUT                       5611
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                  7249
    PAGE 596                                           7330
N.IC.RANGE.HACK
    PAGE 555  ROUTINE MUNS.INPUT                       5629
N.IC.TB.RH.LIST
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES           543
    PAGE 669  ''PROGRAM  OLDER.VERSION                 9984
N.ILLUM.MUNITION
    PAGE 576  ROUTINE ILLUM.INPUT                      6505
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                  7251
    PAGE 596                                           7331
N.KILLED
    PAGE 260  ROUTINE BTRY.EFFECTS                     1987
    PAGE 268                                           2473 2477 2480 2483
    PAGE 269                                           2486 2528 2529
    PAGE 270                                           2544
    PAGE 435  PROCESS ARTY.ASSESS                      9692
    PAGE 436                                           9775 9782
    PAGE 437                                           9813
    PAGE 611  ROUTINE OUTPUT.ATTRITION                 7930 7937 7939 7945 7951 7984 7984
N.KILLER.VICTIM
    PAGE 609  ROUTINE KV.SCOREBOARD                    7862
N.LOS.BAND
    PAGE 571  ROUTINE TT.FACTORS.INPUT                 6289
N.MADS.RH.SET
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES           605
    PAGE 670  ''PROGRAM  OLDER.VERSION                    46
N.MA.SET
    PAGE  37  ''SECTION FOR PROCESSES                  2097
    PAGE 696  ''PROGRAM  OLDER.VERSION                 1535
N.MCFR.RH.LIST
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           643
    PAGE 671  ''PROGRAM  OLDER.VERSION                    84
N.MFO.RB.SET
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES           655
    PAGE 538  ROUTINE MFO.INPUT                        4969 4978
```

```
    PAGE 671  ''PROGRAM  OLDER.VERSION                  96
N.MFP.LIST
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         576
    PAGE 670  ''PROGRAM  OLDER.VERSION                  17
N.MINEFIELD
    PAGE 578  ROUTINE MINE.INPUT                       6598 6599 6601
N.MISSION
    PAGE 262  ROUTINE BTRY.EFFECTS                     2096
    PAGE 547  ROUTINE P.E.M.INPUT                      5275
    PAGE 570  ROUTINE DECISION.INPUT                   6266
N.MODEL.AD.SENSOR
    PAGE 587  ROUTINE MADS.INPUT                       6983
N.MODEL.AO
    PAGE 564  ROUTINE MAO.INPUT                        5962
N.MODEL.CFR
    PAGE 562  ROUTINE MCFR.INPUT                       5886
N.MODEL.FO
    PAGE 538  ROUTINE MFO.INPUT                        4938
N.MODEL.PDB
    PAGE 563  ROUTINE MPDB.INPUT                       5923
N.MOVEMENT.STATUS
    PAGE 538  ROUTINE MFO.INPUT                        4935
N.MO.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES         1062
    PAGE 393  EVENT START.ARTY.MOVEMENT                7685
    PAGE 404  EVENT START.MOVE                         8272
    PAGE 678  ''PROGRAM  OLDER.VERSION                  504
N.MPDB.RH.LIST
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES          669
    PAGE 671  ''PROGRAM  OLDER.VERSION                  112
N.MU.ORDER.SET
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES         1556
    PAGE 687  ''PROGRAM  OLDER.VERSION                  995
N.MU.TF.LIST
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES         1555
    PAGE 147  ROUTINE INTER.BATTLE                     6819
    PAGE 148                                           6856
    PAGE 404  EVENT START.MOVE                         8287 8290
    PAGE 410  EVENT UPDATE.LOC                         8542 8545
    PAGE 687  ''PROGRAM  OLDER.VERSION                  994
N.NITE.OR.DAY
    PAGE  56  ROUTINE MAIN1                            3047
N.PATH.SET
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES         1803
    PAGE 691  ''PROGRAM  OLDER.VERSION                 1242
N.PDB.KEYED.LIST
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES         1642
    PAGE 688  ''PROGRAM  OLDER.VERSION                 1081
N.PDB.OP.Q
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES         1645
    PAGE 688  ''PROGRAM  OLDER.VERSION                 1084
N.PDB.RNG.HACK
    PAGE 563  ROUTINE MPDB.INPUT                       5920 5935
N.PGM.TGTS
    PAGE 575  ROUTINE PGM.INPUT                        6465 6467 6468 6470
```

```
N.PGM.TUBES
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                    7399 7404 7411
N.PIR.FLIGHT.LEG.LIST
    PAGE  37  ''SECTION FOR PROCESSES                    2111
    PAGE 696  ''PROGRAM  OLDER.VERSION                   1549
N.PIR.RECORD.LIST
    PAGE  37  ''SECTION FOR PROCESSES                    2112
    PAGE 696  ''PROGRAM  OLDER.VERSION                   1550
N.PIR.RTD.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES           1684
    PAGE 689  ''PROGRAM  OLDER.VERSION                   1123
N.PK.BAND
    PAGE 525  ROUTINE PK.INPUT                           4388
N.PK.F.MOVE.FACTOR
    PAGE 525  ROUTINE PK.INPUT                           4428 4432
N.PK.MOVE.BAND
    PAGE 525  ROUTINE PK.INPUT                           4413 4432
N.PK.MOVE.FACTOR
    PAGE 525  ROUTINE PK.INPUT                           4412
N.PK.VECTOR
    PAGE 525  ROUTINE PK.INPUT                           4387
N.POSTURE
    PAGE 547  ROUTINE P.E.M.INPUT                        5263
N.QUEUE
    PAGE 164  ROUTINE BTRY.FM.ENQ                        7570
N.RED.TYPE.EQP
    PAGE  42  ''SECTION FOR DEFINITIONS                  2402
    PAGE 520  ROUTINE MAIN2                              4213
    PAGE 530  ROUTINE EQ.TE.INPUT                        4595 4596
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                    7299
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES                8508 8512 8517 8518 8530 8532 8543
    PAGE 625                                             8553 8599
    PAGE 701  ''PROGRAM  OLDER.VERSION                   1832
N.RPV.CAND.DET.LIST
    PAGE  37  ''SECTION FOR PROCESSES                    2125
    PAGE 696  ''PROGRAM  OLDER.VERSION                   1563
N.RPV.FLIGHT.LEG.LIST
    PAGE  37  ''SECTION FOR PROCESSES                    2128
    PAGE 696  ''PROGRAM  OLDER.VERSION                   1566
N.R.WPN.TYPE
    PAGE  42  ''SECTION FOR DEFINITIONS                  2404
    PAGE 520  ROUTINE MAIN2                              4212
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT                  4645 4646
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                    7319
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES                8509 8520 8530
    PAGE 625                                             8570
    PAGE 626                                             8611
    PAGE 701  ''PROGRAM  OLDER.VERSION                   1834
N.SDM.TUBES
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                    7400 7407 7411
N.SD.OLD.SORTIE.QUEUE
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE                  6653
N.SEARCH.POINT
    PAGE  56  ROUTINE MAIN1                              3050
N.SECTOR
    PAGE  61  ROUTINE FEBA.INITIAL                       3193 3200 3202
```

```
    PAGE  89  ROUTINE FA.BN.MOVEMENT                    4438
    PAGE  91                                            4518
    PAGE 100  ROUTINE LOCATE.SECTOR                     4857 4859
    PAGE 138  ROUTINE CHECK.PROX                        6402
    PAGE 246  ROUTINE VOLLEY                            1329
    PAGE 371  EVENT FEBA.SORTIE                         6819
    PAGE 372                                            6860
    PAGE 523  ROUTINE SYS.INPUT                         4354
N.SENSOR.TYPE
    PAGE 561  ROUTINE ST.INPUT                          5862
N.SIDE
    PAGE  56  ROUTINE MAIN1                             3053
    PAGE 529  ROUTINE KV.INPUT                          4533 4556
N.SIDE.CFR.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES           768
    PAGE 673  ''PROGRAM  OLDER.VERSION                   210
N.SIDE.PDB.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES           767
    PAGE 673  ''PROGRAM  OLDER.VERSION                   209
N.SI.LIST
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES          1206
    PAGE 681  ''PROGRAM  OLDER.VERSION                   645
N.SMOKE.MUNITION
    PAGE 580  ROUTINE SMOKE.INPUT                       6679
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                   7252
    PAGE 596                                            7331
N.SO.LIST
    PAGE   3  PROGRAM REVISIONS                          168
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES          1829
    PAGE 282  ROUTINE AC.DF.EFFECTS                     3119
    PAGE 369  EVENT ENGAGEMENT                          6730
    PAGE 379  EVENT HELO.ENGAGEMENT                     7210
    PAGE 383  EVENT OFF.LINE.ATTRITION                  7372
    PAGE 435  PROCESS ARTY.ASSESS                       9716
    PAGE 464  PROCESS MINE.ASSESS                       1257
    PAGE 491  PROCESS ASSESSMENT                        2687
    PAGE 493  PROCESS SHOOT.OUT                         2788
    PAGE 507  PROCESS CAS.MISSION                       3570 3578 3579
    PAGE 691  ''PROGRAM  OLDER.VERSION                  1268
N.SS.SET
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES           811
    PAGE 674  ''PROGRAM  OLDER.VERSION                   253
N.SUBMUNITION
    PAGE 557  ROUTINE SUBM.INPUT                        5721
N.TB.N.FM
    PAGE 548  ROUTINE TB.INPUT                          5316
N.TB.SORT.LIST
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           874
    PAGE 675  ''PROGRAM  OLDER.VERSION                   316
N.TB.TM.LIST
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           906
    PAGE 598  ROUTINE ANALYSIS.OUTPUT                   7415
    PAGE 675  ''PROGRAM  OLDER.VERSION                   348
N.TEAM.TYPES
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           877
    PAGE 675  ''PROGRAM  OLDER.VERSION                   319
```

```
N.TERRAIN.TYPE
    PAGE 571  ROUTINE TT.FACTORS.INPUT                6288
N.TGTS
    PAGE 260  ROUTINE BTRY.EFFECTS                    1987
    PAGE 262                                          2131 2135
    PAGE 263                                          2143
    PAGE 268                                          2464 2474 2480
    PAGE 269                                          2528 2529
N.TR.DET.LIST
    PAGE  38  ''SECTION FOR PROCESSES                 2190
    PAGE 698  ''PROGRAM  OLDER.VERSION                1628
N.TR.FM.LIST
    PAGE  38  ''SECTION FOR PROCESSES                 2189
    PAGE 184  ROUTINE FA.BN.ASGN                      8451
    PAGE 476  PROCESS TARGET.REPORT                   1854
    PAGE 698  ''PROGRAM  OLDER.VERSION                1627
N.TU.NTE.SET
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         953
    PAGE 676  ''PROGRAM  OLDER.VERSION                 395
N.TU.TE.LIST
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         952
    PAGE 676  ''PROGRAM  OLDER.VERSION                 394
N.TYPE.BATTLE.FIELD
    PAGE 569  ROUTINE TBF.INPUT                       6219
N.TYPE.BTRY
    PAGE 548  ROUTINE TB.INPUT                        5319
    PAGE 592  ROUTINE AMMO.RPT                        7129
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                 7411
N.TYPE.EQUIPMENT
    PAGE 428  PROCESS AIR.OBSERVER                    9352
    PAGE 530  ROUTINE EQ.TE.INPUT                     4579
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                 7290
N.TYPE.UNIT
    PAGE 527  ROUTINE CAT.TU.INPUT                    4470
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                 7271
N.TYPE.WEAPON
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4646
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                 7247
N.UE.TARGET.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES        1826
    PAGE 370  EVENT ENGAGEMENT                        6744
    PAGE 497  PROCESS SHOOT.OUT                       2986 3003
    PAGE 691  ''PROGRAM  OLDER.VERSION                1265
N.UE.WEAPON.SET
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES        1832
    PAGE 369  EVENT ENGAGEMENT                        6716
    PAGE 378  EVENT HELO.ENGAGEMENT                   7178
    PAGE 496  PROCESS SHOOT.OUT                       2926
    PAGE 504  PROCESS CAS.MISSION                     3382
    PAGE 507                                          3567
    PAGE 691  ''PROGRAM  OLDER.VERSION                1271
N.UNIT
    PAGE 533  ROUTINE UNIT.INPUT                      4686 4689
    PAGE 536                                          4855 4859 4910
    PAGE 537                                          4924
    PAGE 539  ROUTINE READ.ORDERS                     5001
```

```
    PAGE 551  ROUTINE BTRY.INPUT                    5445
    PAGE 565  ROUTINE SENSOR.INPUT                  6029
    PAGE 567                                        6141
    PAGE 595  ROUTINE ANALYSIS.OUTPUT               7280
    PAGE 613  ROUTINE FOR POSITION.OUT              8001
N.UNIT.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES       776
    PAGE 673  ''PROGRAM  OLDER.VERSION               218
N.UN.EQUIP.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1044
    PAGE 295  ROUTINE END.CAS.MISSION               3825 3845
    PAGE 504  PROCESS CAS.MISSION                   3383
    PAGE 678  ''PROGRAM  OLDER.VERSION               486
N.UN.HC.LOS.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1061
    PAGE 678  ''PROGRAM  OLDER.VERSION               503
N.UN.LOS.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1048
    PAGE 678  ''PROGRAM  OLDER.VERSION               490
N.UN.PATH
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1046
    PAGE  78  ROUTINE ADJUST                        3939 3992
    PAGE  83  ROUTINE CHANGE.LOC                    4167 4177
    PAGE  84                                        4198 4199
    PAGE 399  EVENT START.BATTLE                    8002
    PAGE 678  ''PROGRAM  OLDER.VERSION               488
N.UN.SEGMENT.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1047
    PAGE 678  ''PROGRAM  OLDER.VERSION               489
N.UN.SENSOR.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1049
    PAGE 678  ''PROGRAM  OLDER.VERSION               491
N.UN.SUB.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1045
    PAGE 678  ''PROGRAM  OLDER.VERSION               487
N.VOLS
    PAGE 163  ROUTINE BTRY.FM.DEQ                   7520
OBJECTIVE.
    PAGE 314  ROUTINE FLIGHT.PATH                   4672
OBSTACLES.
    PAGE 314  ROUTINE FLIGHT.PATH                   4643
    PAGE 315                                        4709
OBSTACLE.
    PAGE  65  ROUTINE FILE.KAD.SENSOR               3376
OBS.TIME.REMAINING
    PAGE 428  PROCESS AIR.OBSERVER                  9347
    PAGE 431                                        9516 9517
    PAGE 432                                        9600 9601
OCCUR.
    PAGE 539  ROUTINE READ.ORDERS                   4991
OFF.LINE.ATTRITION
    PAGE  41  ''SECTION FOR EVENTS                  2342 2344
    PAGE 382  EVENT OFF.LINE.ATTRITION              7280 7287 7293 7306 7315 7324 7334
    PAGE 603  ROUTINE BETWEEN.ROUTINE               7661 7662
    PAGE 700  ''PROGRAM  OLDER.VERSION              1776 1778
```

```
OF.
     PAGE 412  EVENT ACT.ATK                          8626
OLD.BATS
     PAGE  63  ROUTINE FILE.FD.SCHD                   3280 3291 3296 3308
     PAGE  64                                         3330 3337 3338 3344 3348
OLD.CEP
     PAGE 480  PROCESS FIRE.MISSION                   2051
     PAGE 482                                         2160
     PAGE 485                                         2326
OLD.DIST
     PAGE 428  PROCESS AIR.OBSERVER                   9345
     PAGE 430                                         9503
     PAGE 431                                         9511 9513
     PAGE 432                                         9595 9596
     PAGE 467  PROCESS REMOTE.PILOT.VEHICLE           1368
     PAGE 468                                         1438 1445 1446
OLD.LEG
     PAGE 429  PROCESS AIR.OBSERVER                   9442
     PAGE 430                                         9447 9448 9449 9450 9451
OLD.SECTOR
     PAGE  84  ROUTINE CHANGE.LOC                     4191 4195
     PAGE  85                                         4255 4259
     PAGE  87  ROUTINE END.MOVE                       4327 4336 4346 4351
     PAGE 110  ROUTINE POSITION                       5289 5293
     PAGE 119  ROUTINE RESET.FEBA.SECTOR              5650 5654 5657 5658 5661 5663 5674 5691 5692 5693
     PAGE 325  ROUTINE DEQ.FEBA.SET                   5107 5111 5114
     PAGE 331  ROUTINE ENQ.FEBA.SET                   5302 5306 5334 5337 5338
     PAGE 400  EVENT START.BATTLE                     8059 8063
     PAGE 404  EVENT START.MOVE                       8239 8243 8252 8256
     PAGE 409  EVENT UPDATE.LOC                       8467 8471 8480 8484
OLD.SORTIE
     PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES       1612
     PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE              6642 6646 6648 6649
     PAGE 504  PROCESS CAS.MISSION                    3386 3387 3388 3389
     PAGE 618  ROUTINE SNAP.R                         8253
     PAGE 688  ''PROGRAM  OLDER.VERSION               1051
OLD.START
     PAGE  63  ROUTINE FILE.FD.SCHD                   3278 3281 3289 3298 3303 3309 3313
     PAGE  64                                         3326 3333 3346
OLD.STOP
     PAGE  63  ROUTINE FILE.FD.SCHD                   3279 3290 3295
     PAGE  64                                         3318 3323 3327 3336 3350 3353
OLD.TOT.RH
     PAGE 538  ROUTINE MFO.INPUT                      4945 4954 4962 4966 4974 4978
     PAGE 562  ROUTINE MCFR.INPUT                     5890 5902 5910
     PAGE 563  ROUTINE MPDB.INPUT                     5927 5938 5948
OLD.Y
     PAGE 653  PROCESS AIRBORNE.RADAR                 9273
     PAGE 654                                         9340 9349 9350
ONLY.
     PAGE 142  ROUTINE DEAD.UNIT                      6532
     PAGE 500  PROCESS CAS.MISSION                    3588
ON.
     PAGE 115  ROUTINE PROX.CHECK                     5504
     PAGE 242  ROUTINE UNIT.ENVIR                     1132
     PAGE 422  PROCESS AC.ATK.TGT                     9035
```

```
ON.CALL
    PAGE  35  ''SECTION FOR PROCESSES                1964
    PAGE  52  ''SECTION FOR SUBSTITUTIONS            2997
    PAGE 284  ROUTINE CAS.EVAL                       3262
    PAGE 694  ''PROGRAM  OLDER.VERSION               1402
    PAGE 711                                         2404
ON.CLL
    PAGE 294  ROUTINE END.CAS.MISSION                3766
ON.GROUND
    PAGE  52  ''SECTION FOR SUBSTITUTIONS            2978
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          749
    PAGE 711  ''PROGRAM  OLDER.VERSION               2385
ON.TIME
    PAGE 359  EVENT CFR.OFF                          6387 6394 6399
OPEN.INPUT.OUTPUT.FILES
    PAGE  55  ''PROGRAM'' MAIN                       3027
    PAGE 647  ROUTINE OPEN.INPUT.OUTPUT.FILES        9180 9188
    PAGE 648  ROUTINE PERFORM.INSTRUMENTATION        9209
OPEN.IO.FILES
    PAGE   2  PROGRAM REVISIONS                       110
OPEN.PCT
    PAGE 242  ROUTINE UNIT.ENVIR                     1137 1166
    PAGE 243                                         1177 1184 1188 1193 1196 1207 1214
    PAGE 244                                         1241
OPP.NO.UNITS
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                177  183  184
    PAGE 445                                          228  233
OPP.UNITS
    PAGE 449  PROCESS HC.ARRIVE.BATTLE                445  447  452  454  455
OPP.X.COORD
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                175  183  183  186  196  198
    PAGE 445                                          228
    PAGE 446                                          299
OPP.Y.COORD
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                176  184  184  186  197  199
    PAGE 445                                          228  233  233
    PAGE 446                                          300
ORDER
    PAGE   1  ROUTINE FOR CROSS_REFERENCING            49
    PAGE   2  PROGRAM REVISIONS                       117
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES       1621
    PAGE  88  ROUTINE END.MOVE                       4376 4390
    PAGE  95  ROUTINE INIT.REINF                     4642 4642 4649 4650 4651 4652 4653 4654 4656 4658 4661
    PAGE 112  ROUTINE PREPARE.LIST                   5357 5357 5358
    PAGE 118  ROUTINE REIN.ARRIVE                    5606 5624 5625 5628 5629
    PAGE 125  ROUTINE WHAT.NEXT                      5879 5880 5886
    PAGE 126                                         5931 5932 5936 5938 5940 5943 5945 5947 5952 5953 5957
    PAGE 142  ROUTINE DEAD.UNIT                      6531 6535
    PAGE 143                                         6600 6601
    PAGE 144                                         6647 6648 6660 6661 6662 6664 6665 6666 6667 6668 6669 6670 6673 6674 6675
                                                     6677 6678 6679 6681 6682 6683
    PAGE 146  ROUTINE INTER.BATTLE                   6751 6764 6765 6768 6773 6774 6775 6781
    PAGE 169  ROUTINE CHK.FD.TR                      7735
    PAGE 172  ROUTINE COPY                           7852
    PAGE 182  ROUTINE FA.BN.ASGN                     8299
    PAGE 184                                         8447
```

```
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS                1382
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE               4140
    PAGE 314  ROUTINE FLIGHT.PATH                     4646
    PAGE 326  ROUTINE DESTROY.ORD                     5121 5124 5125 5126 5127 5128 5129 5130 5131 5132 5133 5134 5135 5136 5140
    PAGE 347  EVENT ACT.REINF                         5817 5818 5829
    PAGE 373  EVENT GET.NX.ORD                        6883 6891 6892 6895 6902 6903 6904 6906 6907 6911 6915 6919 6929
    PAGE 374                                          6935 6940 6944 6946 6951 6956 6963 6967 6979 6980 6982 6983
    PAGE 415  EVENT ACT.MOVCOR                        8744
    PAGE 508  PROCESS CAS.MISSION                     3627 3629
    PAGE 539  ROUTINE READ.ORDERS                     4988 4989 4990
    PAGE 540                                          5045 5051 5052 5053 5054 5057 5062 5065 5068 5070 5073 5075 5078 5080 5083
                                                      5086 5088 5098
    PAGE 541                                          5108
    PAGE 542  ROUTINE ORD.DEF                         5124 5126
    PAGE 543  ROUTINE ORD.ATK                         5145 5147
    PAGE 544  ROUTINE ORD.REINF                       5163 5164 5164 5166
    PAGE 545  ROUTINE ORD.MOVDIS                      5184 5190
    PAGE 546  ROUTINE ORD.MOVCOR                      5209 5214 5215 5221 5222 5224
    PAGE 688  ''PROGRAM  OLDER.VERSION                1060
    PAGE 713                                          2498
ORDERS.
    PAGE 539  ROUTINE READ.ORDERS                     4988
ORDER.
    PAGE 112  ROUTINE PREPARE.LIST                    5359
    PAGE 117  ROUTINE PROX.POS                        5572
    PAGE 125  ROUTINE WHAT.NEXT                       5882
    PAGE 373  EVENT GET.NX.ORD                        6884
    PAGE 403  EVENT START.MOVE                        8187
    PAGE 414  EVENT ACT.DEF                           8721
    PAGE 416  EVENT ACT.MOVDIS                        8772
    PAGE 539  ROUTINE READ.ORDERS                     4990
    PAGE 542  ROUTINE ORD.DEF                         5119
    PAGE 543  ROUTINE ORD.ATK                         5144
ORDR.
    PAGE  87  ROUTINE END.MOVE                        4310 4363 4364
    PAGE  88                                          4366 4367 4369 4373 4388 4389
    PAGE  95  ROUTINE INIT.REINF                      4637
    PAGE 117  ROUTINE PROX.POS                        5566 5582 5583 5585
    PAGE 118  ROUTINE REIN.ARRIVE                     5600 5614 5615 5622
    PAGE 146  ROUTINE INTER.BATTLE                    6739 6759 6760 6762 6772 6772
    PAGE 347  EVENT ACT.REINF                         5806 5828
    PAGE 348                                          5874 5881
    PAGE 403  EVENT START.MOVE                        8183
    PAGE 405                                          8326 8342 8348
    PAGE 406                                          8363
    PAGE 408  EVENT UPDATE.LOC                        8411 8431 8432
    PAGE 409                                          8488 8510 8516
    PAGE 411                                          8583 8592
    PAGE 412  EVENT ACT.ATK                           8602
    PAGE 414  EVENT ACT.DEF                           8717 8726
    PAGE 415  EVENT ACT.MOVCOR                        8739 8750 8755 8759 8760 8761 8762
    PAGE 416  EVENT ACT.MOVDIS                        8769 8779 8785 8792 8798 8799
    PAGE 542  ROUTINE ORD.DEF                         5116 5131 5132 5133 5134 5135
    PAGE 543  ROUTINE ORD.ATK                         5141 5152 5153 5154
    PAGE 544  ROUTINE ORD.REINF                       5160 5173 5174 5175
    PAGE 545  ROUTINE ORD.MOVDIS                      5181 5195 5196 5197 5198 5199
```

```
    PAGE 546  ROUTINE ORD.MOVCOR                    5205 5232 5234 5235 5236 5237 5238 5239 5240 5241 5242 5242 5243 5243 5249
ORD.ID
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES      1623
    PAGE  87  ROUTINE END.MOVE                      4364
    PAGE  88                                        4367
    PAGE  95  ROUTINE INIT.REINF                    4651 4652 4653
    PAGE 113  ROUTINE PREPARE.LIST                  5426 5431
    PAGE 117  ROUTINE PROX.POS                      5582 5583 5585
    PAGE 118  ROUTINE REIN.ARRIVE                   5614 5615 5629
    PAGE 126  ROUTINE WHAT.NEXT                     5945 5947
    PAGE 144  ROUTINE DEAD.UNIT                     6664 6665 6666 6667 6668 6670 6677 6678 6679 6681 6682
    PAGE 146  ROUTINE INTER.BATTLE                  6759 6760 6762
    PAGE 326  ROUTINE DESTROY.ORD                   5126 5128 5130 5132 5134 5136
    PAGE 348  EVENT ACT.REINF                       5881
    PAGE 408  EVENT UPDATE.LOC                      8431 8432
    PAGE 414  EVENT ACT.DEF                         8726
    PAGE 415  EVENT ACT.MOVCOR                      8750 8759 8760 8761
    PAGE 416  EVENT ACT.MOVDIS                      8779 8785 8792 8798
    PAGE 542  ROUTINE ORD.DEF                       5131 5132 5133 5134 5135
    PAGE 543  ROUTINE ORD.ATK                       5152 5153 5154
    PAGE 544  ROUTINE ORD.REINF                     5173 5174 5175
    PAGE 545  ROUTINE ORD.MOVDIS                    5195 5196 5197 5198 5199
    PAGE 546  ROUTINE ORD.MOVCOR                    5232 5234 5235 5236 5237 5238 5239 5240 5241 5242 5242 5243 5243
    PAGE 630  FUNCTION COLLISION                    8694
    PAGE 688  ''PROGRAM  OLDER.VERSION              1062
ORD.ID.
    PAGE 126  ROUTINE WHAT.NEXT                     5938 5940
ORD.MISSION
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES      1245
    PAGE 113  ROUTINE PREPARE.LIST                  5426 5431
    PAGE 144  ROUTINE DEAD.UNIT                     6665 6670
    PAGE 414  EVENT ACT.DEF                         8726
    PAGE 542  ROUTINE ORD.DEF                       5133
    PAGE 681  ''PROGRAM  OLDER.VERSION               684
ORD.NEXT
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES      1605
    PAGE  88  ROUTINE END.MOVE                      4367
    PAGE 144  ROUTINE DEAD.UNIT                     6682
    PAGE 545  ROUTINE ORD.MOVDIS                    5199
    PAGE 687  ''PROGRAM  OLDER.VERSION              1044
ORD.SEQ.NO
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES      1624
    PAGE  88  ROUTINE END.MOVE                      4388
    PAGE  95  ROUTINE INIT.REINF                    4654 4654 4658
    PAGE 113  ROUTINE PREPARE.LIST                  5409 5424
    PAGE 118  ROUTINE REIN.ARRIVE                   5625
    PAGE 126  ROUTINE WHAT.NEXT                     5932
    PAGE 144  ROUTINE DEAD.UNIT                     6662 6675
    PAGE 146  ROUTINE INTER.BATTLE                  6765 6772
    PAGE 347  EVENT ACT.REINF                       5828
    PAGE 373  EVENT GET.NX.ORD                      6892 6902 6907
    PAGE 415  EVENT ACT.MOVCOR                      8755
    PAGE 540  ROUTINE READ.ORDERS                   5053
    PAGE 630  FUNCTION COLLISION                    8696
    PAGE 688  ''PROGRAM  OLDER.VERSION              1063
```

```
ORD.TYPE
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES          1622
    PAGE  47  ''SECTION FOR DEFINITIONS                 2684
    PAGE  87  ROUTINE END.MOVE                          4363
    PAGE  88                                            4366 4369 4389
    PAGE  95  ROUTINE INIT.REINF                        4650
    PAGE 113  ROUTINE PREPARE.LIST                      5413 5418 5420
    PAGE 118  ROUTINE REIN.ARRIVE                       5628
    PAGE 126  ROUTINE WHAT.NEXT                         5936 5943
    PAGE 144  ROUTINE DEAD.UNIT                         6661 6674
    PAGE 146  ROUTINE INTER.BATTLE                      6768 6772 6773 6781
    PAGE 326  ROUTINE DESTROY.ORD                       5125 5127 5129 5131 5133 5135
    PAGE 373  EVENT GET.NX.ORD                          6895 6903 6911 6919
    PAGE 374                                            6940 6946 6963 6979 6982
    PAGE 405  EVENT START.MOVE                          8326
    PAGE 409  EVENT UPDATE.LOC                          8488
    PAGE 540  ROUTINE READ.ORDERS                       5052 5054 5065 5070 5075 5080 5086
    PAGE 630  FUNCTION COLLISION                        8701
    PAGE 688  ''PROGRAM  OLDER.VERSION                  1061
    PAGE 706                                            2108
OS.QTY
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES          1613
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE                 6648
    PAGE 504  PROCESS CAS.MISSION                       3389
    PAGE 688  ''PROGRAM  OLDER.VERSION                  1052
OTHER.FORCE
    PAGE 133  ROUTINE CHECK.FORCE                       6200 6201 6207
    PAGE 145  ROUTINE FIN.BATTLE                        6705 6729 6732 6733
    PAGE 478  PROCESS WITH.DRAW                         1952 1954 1959
OTHER.UNIT
    PAGE 101  ROUTINE LOS.CHECK                         4884 4886 4892 4893 4899 4900 4909 4916 4918 4925
OVERLAP.COORD
    PAGE 188  ROUTINE FINAL.COVERAGE                    8560
    PAGE 191                                            8766 8767 8768 8769 8770 8774 8776 8778
    PAGE 192                                            8779 8781 8784 8785 8786 8787 8791 8793 8795 8796 8798 8803
OVERLAP.LENGTH
    PAGE 192  ROUTINE FINAL.COVERAGE                    8795 8798 8801
OVERLAP.WIDTH
    PAGE 191  ROUTINE FINAL.COVERAGE                    8778
    PAGE 192                                            8781 8801
OVER.
    PAGE  58  ROUTINE CREATE.FORCE                      3124
OWN.DIS.OP
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES          1244
    PAGE 126  ROUTINE WHAT.NEXT                         5940
    PAGE 144  ROUTINE DEAD.UNIT                         6668
    PAGE 542  ROUTINE ORD.DEF                           5135
    PAGE 681  ''PROGRAM  OLDER.VERSION                   683
OWN.DO
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1146
    PAGE 126  ROUTINE WHAT.NEXT                         5947
    PAGE 543  ROUTINE ORD.ATK                           5154
    PAGE 680  ''PROGRAM  OLDER.VERSION                   585
O.ST.DEV
    PAGE 242  ROUTINE UNIT.ENVIR                        1139
    PAGE 243                                            1207 1214
```

```
    PAGE 244                                          1241
PASSES.
    PAGE 420  PROCESS AC.ATK.TGT                      8947
    PAGE 427                                          9321
PASSIVE.DETECTION.BASE
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES        1631
    PAGE 161  ROUTINE ATTRIT.SENSOR                   7436
    PAGE 567  ROUTINE SENSOR.INPUT                    6130
    PAGE 619  ROUTINE SNAP.R                          8297
    PAGE 688  ''PROGRAM  OLDER.VERSION                1070
PASS.
    PAGE 424  PROCESS AC.ATK.TGT                      9186
    PAGE 505  PROCESS CAS.MISSION                     3450
PATH.
    PAGE  77  ROUTINE ADJUST                          3931
    PAGE 314  ROUTINE FLIGHT.PATH                     4676 4682
    PAGE 315                                          4709
PATH.POINT
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES        1662
    PAGE  60  ROUTINE CREATE.TEAMS                    3174 3175 3176 3179
    PAGE  67  ROUTINE GENERAL.BATTLE                  3465 3468 3469
    PAGE  68                                          3533 3535 3537 3539 3540 3542 3545 3548
    PAGE  69                                          3565 3567 3569 3571 3572 3574 3577 3580 3601 3603 3608 3610 3617 3619
    PAGE  70                                          3624 3626
    PAGE 688  ''PROGRAM  OLDER.VERSION                1101
PATH.SET
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES        1666
    PAGE  32                                          1797
    PAGE  60  ROUTINE CREATE.TEAMS                    3179
    PAGE  67  ROUTINE GENERAL.BATTLE                  3465 3466 3468
    PAGE  68                                          3539 3548
    PAGE  69                                          3571 3580 3601 3608 3617
    PAGE  70                                          3624
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                 3853 3854
    PAGE 689  ''PROGRAM  OLDER.VERSION                1105
    PAGE 691                                          1236
PATT.RAD.2
    PAGE 638  FUNCTION ICM.WLA                        8966 8986 9007 9009
    PAGE 639                                          9015 9017
PCT.ON.HAND
    PAGE 117  ROUTINE PROX.POS                        5581
    PAGE 146  ROUTINE INTER.BATTLE                    6758 6759
    PAGE 348  EVENT ACT.REINF                         5863 5864
PCT.ON.HAND.
    PAGE 117  ROUTINE PROX.POS                        5582
PCT.STREN
    PAGE 141  ROUTINE CHECK.STREN                     6503 6519
PDB.ACTIVATION
    PAGE  40  ''SECTION FOR EVENTS                    2293
    PAGE 246  ROUTINE VOLLEY                          1345 1350
    PAGE 386  EVENT PDB.ACTIVATION                    7467
    PAGE 602  ROUTINE BETWEEN.ROUTINE                 7616 7617
    PAGE 616  ROUTINE SNAP2                           8184
    PAGE 699  ''PROGRAM  OLDER.VERSION                1730
PDB.ACT.ACTIVITY.TYPE
    PAGE  40  ''SECTION FOR EVENTS                    2295
```

```
    PAGE  47  ''SECTION FOR DEFINITIONS              2685
    PAGE 699  ''PROGRAM  OLDER.VERSION               1732
    PAGE 706                                         2109
PDB.ACT.BTRY
    PAGE  40  ''SECTION FOR EVENTS                   2294
    PAGE 699  ''PROGRAM  OLDER.VERSION               1731
PDB.DET.UNIT
    PAGE  40  ''SECTION FOR EVENTS                   2299
    PAGE 699  ''PROGRAM  OLDER.VERSION               1736
PDB.KEYED.LIST
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES       1511
    PAGE  29                                         1635
    PAGE 160  ROUTINE ATTRIT.SENSOR                  7418 7420
    PAGE 161                                         7443 7446
    PAGE 206  ROUTINE PDB.DETECTION                  9464
    PAGE 567  ROUTINE SENSOR.INPUT                   6138 6163
    PAGE 686  ''PROGRAM  OLDER.VERSION                950
    PAGE 688                                         1074
PDB.OPERATOR
    PAGE  40  ''SECTION FOR EVENTS                   2297
    PAGE 160  ROUTINE ATTRIT.SENSOR                  7427
    PAGE 161                                         7428
    PAGE 206  ROUTINE PDB.DETECTION                  9461
    PAGE 387  EVENT PDB.OPERATOR                     7504 7518 7532
    PAGE 388                                         7564
    PAGE 602  ROUTINE BETWEEN.ROUTINE                7620 7621
    PAGE 616  ROUTINE SNAP2                          8185
    PAGE 699  ''PROGRAM  OLDER.VERSION               1734
PDB.OP.Q
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES       1636 1656
    PAGE  49  ''SECTION FOR DEFINITIONS              2818
    PAGE 161  ROUTINE ATTRIT.SENSOR                  7430 7432
    PAGE 206  ROUTINE PDB.DETECTION                  9459
    PAGE 387  EVENT PDB.OPERATOR                     7521
    PAGE 388                                         7562
    PAGE 688  ''PROGRAM  OLDER.VERSION               1075 1095
    PAGE 708                                         2239
PDB.RH.RANGE
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES        702
    PAGE 206  ROUTINE PDB.DETECTION                  9443
    PAGE 386  EVENT PDB.ACTIVATION                   7497
    PAGE 563  ROUTINE MPDB.INPUT                     5941
    PAGE 672  ''PROGRAM  OLDER.VERSION                145
PDB.RNG.HACK
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES        697  701
    PAGE 563  ROUTINE MPDB.INPUT                     5921
    PAGE 618  ROUTINE SNAP.R                         8238
    PAGE 672  ''PROGRAM  OLDER.VERSION                140  144
    PAGE 712                                         2457
PDB.SENS.ID
    PAGE  40  ''SECTION FOR EVENTS                   2298
    PAGE 160  ROUTINE ATTRIT.SENSOR                  7424
    PAGE 699  ''PROGRAM  OLDER.VERSION               1735
PDB.US.LINK
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES       1632
    PAGE 205  ROUTINE NOISE.DEGRADE                  9390
```

```
    PAGE 206  ROUTINE PDB.DETECTION                  9439
    PAGE 386  EVENT PDB.ACTIVATION                   7491
    PAGE 387  EVENT PDB.OPERATOR                     7512
    PAGE 567  ROUTINE SENSOR.INPUT                   6131
    PAGE 688  ''PROGRAM  OLDER.VERSION               1071
PD.DET.UNIT
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES       1650
    PAGE 161  ROUTINE ATTRIT.SENSOR                  7433
    PAGE 206  ROUTINE PDB.DETECTION                  9454
    PAGE 387  EVENT PDB.OPERATOR                     7548
    PAGE 688  ''PROGRAM  OLDER.VERSION               1089
PD.D.BTRY
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES       1651
    PAGE 206  ROUTINE PDB.DETECTION                  9455
    PAGE 387  EVENT PDB.OPERATOR                     7523
    PAGE 688  ''PROGRAM  OLDER.VERSION               1090
PD.D.CEP
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES       1652
    PAGE 206  ROUTINE PDB.DETECTION                  9456
    PAGE 387  EVENT PDB.OPERATOR                     7529
    PAGE 688  ''PROGRAM  OLDER.VERSION               1091
PD.D.PD
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES       1653
    PAGE 206  ROUTINE PDB.DETECTION                  9457
    PAGE 387  EVENT PDB.OPERATOR                     7544
    PAGE 688  ''PROGRAM  OLDER.VERSION               1092
PD.D.PRIORITY
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES       1654
    PAGE  49  ''SECTION FOR DEFINITIONS              2818
    PAGE 206  ROUTINE PDB.DETECTION                  9458
    PAGE 688  ''PROGRAM  OLDER.VERSION               1093
    PAGE 708                                         2239
PD.OPERATOR
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES       1633
    PAGE  51  ''SECTION FOR SUBSTITUTIONS            2914
    PAGE 206  ROUTINE PDB.DETECTION                  9460
    PAGE 387  EVENT PDB.OPERATOR                     7514 7515 7520
    PAGE 688  ''PROGRAM  OLDER.VERSION               1072
    PAGE 709                                         2322
PERCENTS.
    PAGE 212  ROUTINE PIR.DETECTION                  9724
    PAGE 229  ROUTINE RPV.DETECTION                   538  539
PERCENT.
    PAGE 421  PROCESS AC.ATK.TGT                     8970
PERCENT.LOSS
    PAGE 606  ROUTINE KV.PRINT                       7726 7727 7731
    PAGE 607                                         7785
    PAGE 608                                         7836 7840
PERIOD.
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE              6644
PERPENDICULAR.DIST
    PAGE 154  ROUTINE AO.DETECTION                   7093 7106 7112 7127 7131
    PAGE 155                                         7134 7137 7138 7139 7162 7168 7178
    PAGE 156                                         7232 7237
    PAGE 428  PROCESS AIR.OBSERVER                   9345
    PAGE 430                                         9484 9486 9499
```

```
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1368
    PAGE 468                                        1414 1416 1426 1434
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9484 9527
    PAGE 659                                        9536
PERP.DIST
    PAGE 154  ROUTINE AO.DETECTION                  7113
PGM.HIT
    PAGE  50  ''SECTION FOR DEFINITIONS             2842
    PAGE 268  ROUTINE BTRY.EFFECTS                  2433 2433 2434
    PAGE 575  ROUTINE PGM.INPUT                     6455 6461
    PAGE 708  ''PROGRAM  OLDER.VERSION              2263
PGM.INPUT
    PAGE 521  ROUTINE MAIN2                         4278 4280
    PAGE 575  ROUTINE PGM.INPUT                     6449
PGM.KILL
    PAGE  50  ''SECTION FOR DEFINITIONS             2842
    PAGE 268  ROUTINE BTRY.EFFECTS                  2452
    PAGE 575  ROUTINE PGM.INPUT                     6468 6483
    PAGE 708  ''PROGRAM  OLDER.VERSION              2263
PGM.LINK
    PAGE  50  ''SECTION FOR DEFINITIONS             2843
    PAGE 268  ROUTINE BTRY.EFFECTS                  2443 2444
    PAGE 575  ROUTINE PGM.INPUT                     6467 6480
    PAGE 708  ''PROGRAM  OLDER.VERSION              2264
PGM.LOS.TIME
    PAGE 210  ROUTINE PGM.MSN.ASGN                  9626 9630 9633 9650 9652
PGM.MARK
    PAGE  47  ''SECTION FOR DEFINITIONS             2686
    PAGE 575  ROUTINE PGM.INPUT                     6490
    PAGE 593  ROUTINE AMMO.RPT                      7185
    PAGE 706  ''PROGRAM  OLDER.VERSION              2110
PGM.MISSION.TIME
    PAGE 210  ROUTINE PGM.MSN.ASGN                  9647 9650 9652
PGM.PREP.TIME
    PAGE 210  ROUTINE PGM.MSN.ASGN                  9636 9647 9648
PGM.RELY
    PAGE  48  ''SECTION FOR DEFINITIONS             2744
    PAGE 575  ROUTINE PGM.INPUT                     6487
    PAGE 707  ''PROGRAM  OLDER.VERSION              2166
PGM.TOF.TIME
    PAGE 210  ROUTINE PGM.MSN.ASGN                  9638 9641 9643 9647 9649
PHOTO.IR.FLIGHT
    PAGE  37  ''SECTION FOR PROCESSES               2099
    PAGE 600  ROUTINE BETWEEN.ROUTINE               7504 7505
    PAGE 616  ROUTINE SNAP2                         8155
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9474 9488
    PAGE 696  ''PROGRAM  OLDER.VERSION              1537
PIR.DETECTION
    PAGE 212  ROUTINE PIR.DETECTION                 9701 9715
    PAGE 242  ROUTINE UNIT.ENVIR                    1127
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9541
PIR.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES      1393
    PAGE  37  ''SECTION FOR PROCESSES               2104
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9504 9506 9508
    PAGE 684  ''PROGRAM  OLDER.VERSION               832
```

```
    PAGE 696                                        1542
PIR.RECORD.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1675
    PAGE  37  ''SECTION FOR PROCESSES               2105
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9547 9557 9559
    PAGE 689  ''PROGRAM  OLDER.VERSION              1114
    PAGE 696                                        1543
PIR.REC.TARGET
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1672
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9523
    PAGE 659                                        9543 9565
    PAGE 689  ''PROGRAM  OLDER.VERSION              1111
PIR.RTD.ELEM.PROB
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1688
    PAGE  46  ''SECTION FOR DEFINITIONS             2626
    PAGE 213  ROUTINE PIR.DETECTION                 9783
    PAGE 689  ''PROGRAM  OLDER.VERSION              1127
    PAGE 705                                        2049
PIR.RTD.LINK
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1686
    PAGE 213  ROUTINE PIR.DETECTION                 9780
    PAGE 689  ''PROGRAM  OLDER.VERSION              1125
PIR.RTD.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1677 1691
    PAGE 213  ROUTINE PIR.DETECTION                 9776 9781
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9542 9560 9562
    PAGE 689  ''PROGRAM  OLDER.VERSION              1116 1130
PIR.RTD.QUANT
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1689
    PAGE  46  ''SECTION FOR DEFINITIONS             2627
    PAGE 213  ROUTINE PIR.DETECTION                 9784 9786
    PAGE 689  ''PROGRAM  OLDER.VERSION              1128
    PAGE 705                                        2050
PIR.RTD.TE
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1687
    PAGE 213  ROUTINE PIR.DETECTION                 9777 9782
    PAGE 689  ''PROGRAM  OLDER.VERSION              1126
PIR.RT.UNIT
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1673
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9546 9570
    PAGE 689  ''PROGRAM  OLDER.VERSION              1112
PIR.US.LINK
    PAGE  37  ''SECTION FOR PROCESSES               2100
    PAGE 212  ROUTINE PIR.DETECTION                 9717
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9490
    PAGE 696  ''PROGRAM  OLDER.VERSION              1538
PIR.X.START
    PAGE  37  ''SECTION FOR PROCESSES               2101
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9502
    PAGE 696  ''PROGRAM  OLDER.VERSION              1539
PIR.Y.START
    PAGE  37  ''SECTION FOR PROCESSES               2102
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9501
    PAGE 696  ''PROGRAM  OLDER.VERSION              1540
PI.C
    PAGE  51  ''SECTION FOR SUBSTITUTIONS           2899 2900 2902
```

```
    PAGE  73  ROUTINE ORIENTATION                    3791
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA             4789 4792 4792 4795 4806
    PAGE 166  ROUTINE CFR.DETECTION                  7640 7642 7649 7654 7659 7662 7663 7665 7666 7667 7668 7669
    PAGE 178  ROUTINE EST.COVERAGE                   8149 8150 8152 8156 8163 8165 8169 8170
    PAGE 190  ROUTINE FINAL.COVERAGE                 8698 8699 8702
    PAGE 191                                         8722 8737 8738 8745 8746
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9337
    PAGE 204                                         9358 9368
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS               1445 1449 1451
    PAGE 263  ROUTINE BTRY.EFFECTS                   2177 2189
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                2861 2862 2864 2865 2870 2873 2877
    PAGE 335  ROUTINE FRAC.COMPUTE                   5431
    PAGE 343  ROUTINE SEARCH.COVERAGE                5707
    PAGE 360  EVENT CFR.ON                           6449 6460 6462
    PAGE 361                                         6470
    PAGE 446  PROCESS HC.ARRIVE.BATTLE                307  309  314  321
    PAGE 447                                          335  337  342  349
    PAGE 448                                          392  393
    PAGE 507  PROCESS CAS.MISSION                    3532 3534 3539
    PAGE 584  ROUTINE TACAIR.INPUT                   6867 6869 6874
    PAGE 636  FUNCTION HE.WLA                        8899
    PAGE 638  FUNCTION ICM.WLA                       9007 9009
    PAGE 639                                         9015 9017
    PAGE 644  ROUTINE ANGLE.COMPUTE                  9098 9105 9107 9107
    PAGE 709  ''PROGRAM  OLDER.VERSION               2307 2308 2310
PI.RAD.2
    PAGE 260  ROUTINE BTRY.EFFECTS                   1983
    PAGE 263                                         2177 2189
    PAGE 272                                         2682
    PAGE 636  FUNCTION HE.WLA                        8869 8899
    PAGE 637                                         8935 8936 8942 8950
PK.ADJ.MOV.FAC
    PAGE 149  ROUTINE PK.COMPUTE                     6907
    PAGE 150                                         7008
    PAGE 151                                         7015 7021 7023 7026 7036 7042 7044 7048
PK.BAND
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES       1069 1072
    PAGE 150  ROUTINE PK.COMPUTE                     6985 6986 6991 6992 6993 6998
    PAGE 281  ROUTINE AC.DF.EFFECTS                  3080 3081 3087 3089 3090 3096
    PAGE 308  ROUTINE AD.SHOOT                       4374 4375 4381 4383 4384 4390
    PAGE 310                                         4507
    PAGE 525  ROUTINE PK.INPUT                       4389 4391 4393 4397 4401
    PAGE 678  ''PROGRAM  OLDER.VERSION                511  514
    PAGE 712                                         2458
PK.BAND.RNG
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES       1070
    PAGE  42  ''SECTION FOR DEFINITIONS              2373
    PAGE 150  ROUTINE PK.COMPUTE                     6986 6993
    PAGE 281  ROUTINE AC.DF.EFFECTS                  3081 3090
    PAGE 308  ROUTINE AD.SHOOT                       4375 4384
    PAGE 525  ROUTINE PK.INPUT                       4393
    PAGE 678  ''PROGRAM  OLDER.VERSION                512
    PAGE 701                                         1803
PK.DEFILADE
    PAGE  43  ''SECTION FOR DEFINITIONS              2442
    PAGE 151  ROUTINE PK.COMPUTE                     7064
```

```
    PAGE 488  PROCESS ASSESSMENT                    2470
    PAGE 489                                        2584
    PAGE 490                                        2622
    PAGE 491                                        2676
    PAGE 492                                        2729
    PAGE 499  PROCESS SHOOT.OUT                     3149
    PAGE 500                                        3182 3201
    PAGE 501                                        3223 3251
    PAGE 502                                        3304
    PAGE 516  PROCESS HELICOPTER.FIRE               4017
    PAGE 517                                        4096
    PAGE 518                                        4155
PK.DEF.POINTER
    PAGE  50  ''SECTION FOR DEFINITIONS             2836
    PAGE 150  ROUTINE PK.COMPUTE                    6955
    PAGE 525  ROUTINE PK.INPUT                      4409 4411
    PAGE 708  ''PROGRAM  OLDER.VERSION              2257
PK.F.MOVE.FACTOR
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1081
    PAGE 151  ROUTINE PK.COMPUTE                    7032 7033 7045
    PAGE 525  ROUTINE PK.INPUT                      4429 4434 4437
    PAGE 678  ''PROGRAM  OLDER.VERSION               523
    PAGE 712                                        2459
PK.F.MOV.FAC
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1082
    PAGE  48  ''SECTION FOR DEFINITIONS             2765
    PAGE 151  ROUTINE PK.COMPUTE                    7045
    PAGE 525  ROUTINE PK.INPUT                      4430 4432 4437
    PAGE 678  ''PROGRAM  OLDER.VERSION               524
    PAGE 707                                        2187
PK.HE
    PAGE 636  FUNCTION HE.WLA                       8869
    PAGE 637                                        8935 8936 8942 8950
PK.ICM
    PAGE 638  FUNCTION ICM.WLA                      8966 9008 9009
    PAGE 639                                        9016 9017
PK.INPUT
    PAGE 281  ROUTINE AC.DF.EFFECTS                 3101
    PAGE 308  ROUTINE AD.SHOOT                      4395
    PAGE 520  ROUTINE MAIN2                         4196 4198
    PAGE 525  ROUTINE PK.INPUT                      4381
PK.MOVE.BAND
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1075 1078 1081
    PAGE 151  ROUTINE PK.COMPUTE                    7017 7018 7024 7038 7039 7046
    PAGE 525  ROUTINE PK.INPUT                      4414 4416 4419 4423 4425 4435 4437
    PAGE 678  ''PROGRAM  OLDER.VERSION               517  520  523
    PAGE 712                                        2460
PK.MOVE.FACTOR
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1078
    PAGE 151  ROUTINE PK.COMPUTE                    7011 7012 7024
    PAGE 525  ROUTINE PK.INPUT                      4414 4422 4425
    PAGE 678  ''PROGRAM  OLDER.VERSION               520
    PAGE 712                                        2461
PK.MOVE.FACTOR.NO
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1069
    PAGE 678  ''PROGRAM  OLDER.VERSION               511
```

```
PK.MOV.FAC
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1079
    PAGE  48  ''SECTION FOR DEFINITIONS                 2760
    PAGE 151  ROUTINE PK.COMPUTE                        7024
    PAGE 525  ROUTINE PK.INPUT                          4425
    PAGE 678  ''PROGRAM  OLDER.VERSION                   521
    PAGE 707                                            2182
PK.MOV.RNG
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1076
    PAGE  44  ''SECTION FOR DEFINITIONS                 2495
    PAGE 151  ROUTINE PK.COMPUTE                        7018 7039
    PAGE 525  ROUTINE PK.INPUT                          4419
    PAGE 678  ''PROGRAM  OLDER.VERSION                   518
    PAGE 702                                            1916
PK.POINTER
    PAGE  50  ''SECTION FOR DEFINITIONS                 2835
    PAGE 150  ROUTINE PK.COMPUTE                        6957
    PAGE 281  ROUTINE AC.DF.EFFECTS                     3062
    PAGE 307  ROUTINE AD.SHOOT                          4351
    PAGE 525  ROUTINE PK.INPUT                          4406 4408
    PAGE 708  ''PROGRAM  OLDER.VERSION                  2256
PK.PROB
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1073
    PAGE  42  ''SECTION FOR DEFINITIONS                 2374
    PAGE 150  ROUTINE PK.COMPUTE                        6991 6998
    PAGE 281  ROUTINE AC.DF.EFFECTS                     3077 3087 3096
    PAGE 308  ROUTINE AD.SHOOT                          4372 4381 4390
    PAGE 525  ROUTINE PK.INPUT                          4401
    PAGE 678  ''PROGRAM  OLDER.VERSION                   515
    PAGE 701                                            1804
PK.VECTOR
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1072
    PAGE 150  ROUTINE PK.COMPUTE                        6960 6961 6991 6998
    PAGE 525  ROUTINE PK.INPUT                          4389 4396 4401
    PAGE 678  ''PROGRAM  OLDER.VERSION                   514
    PAGE 712                                            2462
PK.VECTOR.NO
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1069
    PAGE 678  ''PROGRAM  OLDER.VERSION                   511
PK.VECTOR.NR
    PAGE  42  ''SECTION FOR DEFINITIONS                 2375
    PAGE 701  ''PROGRAM  OLDER.VERSION                  1805
POINT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING               49
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES          1697
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                   3856 3865 3866 3867
    PAGE  77  ROUTINE ADJUST                            3889 3894 3914 3916 3917 3919 3920 3922 3927 3931
    PAGE  78                                            3937 3938 3946 3947 3948 3949 3993 3994
    PAGE  83  ROUTINE CHANGE.LOC                        4132 4165 4170 4171
    PAGE  84                                            4185 4221 4229 4233 4234
    PAGE  85                                            4249 4292
    PAGE 110  ROUTINE POSITION                          5269 5276 5277
    PAGE 111  ROUTINE PRED.POS                          5324 5333 5334
    PAGE 135  ROUTINE CHECK.FOR.MINES                   6272
    PAGE 314  ROUTINE FLIGHT.PATH                       4660 4676
    PAGE 321  ROUTINE COMPUTE.D                         4947 4953 4958 4959
```

```
    PAGE 322  ROUTINE COMPUTE.WD                    4974 4975 4976 4983 4986
    PAGE 329  ROUTINE EMPTY                         5271 5272
    PAGE 349  EVENT AD.ENGAGEMENT                   5921 5941
    PAGE 352                                        6085
    PAGE 353                                        6129
    PAGE 399  EVENT START.BATTLE                    7989 7995 7996
    PAGE 422  PROCESS AC.ATK.TGT                    9036 9073
    PAGE 423                                        9116
    PAGE 425                                        9238 9239
    PAGE 584  ROUTINE TACAIR.INPUT                  6898
    PAGE 619  ROUTINE SNAP.R                        8266
    PAGE 689  ''PROGRAM  OLDER.VERSION              1136
POINTS.
    PAGE  65  ROUTINE FILE.KAD.SENSOR               3404
POINT.
    PAGE  77  ROUTINE ADJUST                        3924 3934
    PAGE 422  PROCESS AC.ATK.TGT                    9034
POINT.ER
    PAGE 112  ROUTINE PREPARE.LIST                  5403
    PAGE 113                                        5404 5405
    PAGE 137  ROUTINE CHECK.LIST                    6353 6354
POINT.RANGE.RATIO
    PAGE  77  ROUTINE ADJUST                        3890
POSITION.REPORT
    PAGE  41  ''SECTION FOR EVENTS                  2338
    PAGE  57  ROUTINE MAIN3                         3080
    PAGE 389  EVENT POSITION.REPORT                 7568 7575
    PAGE 602  ROUTINE BETWEEN.ROUTINE               7652 7653
    PAGE 617  ROUTINE SNAP2                         8196
    PAGE 700  ''PROGRAM  OLDER.VERSION              1775
POSTURE
    PAGE   1  ROUTINE FOR CROSS_REFERENCING          44
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES      261
    PAGE   7                                        374
    PAGE   8                                        404
    PAGE  10                                        519
    PAGE  13                                        710
    PAGE 266  ROUTINE BTRY.EFFECTS                  2313 2322
    PAGE 277  ROUTINE AC.BOMB.EFFECTS               2898
    PAGE 547  ROUTINE P.E.M.INPUT                   5258 5284 5266 5267 5291 5297 5299 5300
    PAGE 557  ROUTINE SUBM.INPUT                    5734 5736
    PAGE 558  ROUTINE HE.LA.INPUT                   5761 5780 5781 5783
    PAGE 588  ROUTINE AC.MUNS.INPUT                 7032 7033
    PAGE 618  ROUTINE SNAP.R                        8239
    PAGE 637  FUNCTION HE.WLA                       8938
    PAGE 638  FUNCTION ICM.WLA                      9012
    PAGE 664  ''PROGRAM  OLDER.VERSION              9703
    PAGE 666                                        9816
    PAGE 667                                        9846
    PAGE 669                                        9960
    PAGE 672                                         153
    PAGE 712                                        2463
POS.REP.INT
    PAGE  48  ''SECTION FOR DEFINITIONS             2748
    PAGE 389  EVENT POSITION.REPORT                 7575
    PAGE 523  ROUTINE SYS.INPUT                     4340
```

```
    PAGE 707  ''PROGRAM  OLDER.VERSION                2170
PP.X.POINT
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES        1663
    PAGE  46  ''SECTION FOR DEFINITIONS               2624
    PAGE  60  ROUTINE CREATE.TEAMS                    3175 3177 3177
    PAGE  68  ROUTINE GENERAL.BATTLE                  3535 3537 3542 3545
    PAGE  69                                          3567 3569 3574 3577
    PAGE  72  ROUTINE ORIENTATION                     3701 3704 3711 3714
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                 3859 3864
    PAGE 688  ''PROGRAM  OLDER.VERSION                1102
    PAGE 705                                          2047
PP.Y.POINT
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES        1664
    PAGE  46  ''SECTION FOR DEFINITIONS               2625
    PAGE  60  ROUTINE CREATE.TEAMS                    3176 3178 3178
    PAGE  69  ROUTINE GENERAL.BATTLE                  3603 3610 3619
    PAGE  70                                          3626
    PAGE  72  ROUTINE ORIENTATION                     3702 3705 3712 3715
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                 3860 3864
    PAGE 688  ''PROGRAM  OLDER.VERSION                1103
    PAGE 705                                          2048
PREPARE.LIST
    PAGE 112  ROUTINE PREPARE.LIST                    5346
    PAGE 113                                          5406
    PAGE 412  EVENT ACT.ATK                           8653
    PAGE 413                                          8678
PREP.TIME
    PAGE 480  PROCESS FIRE.MISSION                    2048
    PAGE 482                                          2149 2153 2167 2167 2170
    PAGE 483                                          2214
PRESENT.SECTOR
    PAGE 325  ROUTINE DEQ.FEBA.SET                    5089 5112
PRINT.SWITCH
    PAGE 572  ROUTINE FARRP.INPUT                     6354 6371
    PAGE 573                                          6432
PRIN.EQ
    PAGE 636  FUNCTION HE.WLA                         8897
    PAGE 637                                          8932 8933
    PAGE 638  FUNCTION ICM.WLA                        8995 9005 9006
PROB.LINE.OF.SITE
    PAGE 254  ROUTINE FO.DETECTION                    1681 1719 1720 1722
PROB.LOS
    PAGE 654  PROCESS AIRBORNE.RADAR                  9328 9329
PROCESS.
    PAGE 240  ROUTINE TARGET.ANALYSIS                 1076
PROCESS.TIME
    PAGE 185  ROUTINE FA.BN.ASGN                      8502 8505
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9499
    PAGE 211                                          9691 9692 9695
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                 9554 9555
PROCESS.V
    PAGE   2  PROGRAM REVISIONS                         75
    PAGE 419  PROCESS AC.ATK.TGT                      8851
    PAGE 428  PROCESS AIR.OBSERVER                    9335
    PAGE 435  PROCESS ARTY.ASSESS                     9690
    PAGE 438  PROCESS FORWARD.OBSERVER                9858
```

```
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1359
    PAGE 470  PROCESS TARGET.REPORT                 1489
    PAGE 478  PROCESS WITH.DRAW                     1920
    PAGE 480  PROCESS FIRE.MISSION                  2042
    PAGE 487  PROCESS ASSESSMENT                    2423
    PAGE 493  PROCESS SHOOT.OUT                     2753
    PAGE 504  PROCESS CAS.MISSION                   3353
    PAGE 510  PROCESS HELICOPTER.FIRE               3670
    PAGE 653  PROCESS AIRBORNE.RADAR                9266
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9477
PROX.TEST
    PAGE 138  ROUTINE CHECK.PROX                    6382
    PAGE 139                                        6467
PROX.TEST.
    PAGE 138  ROUTINE CHECK.PROX                    6374 6413
    PAGE 139                                        6462
    PAGE 347  EVENT ACT.REINF                       5851 5857
    PAGE 405  EVENT START.MOVE                      8333 8336
    PAGE 409  EVENT UPDATE.LOC                      8495 8498
PROX.UNITS.
    PAGE 347  EVENT ACT.REINF                       5825 5852 5855
PR.1RND
    PAGE 192  ROUTINE FINAL.COVERAGE                8828
    PAGE 193                                        8860
PR.1RND.PAT.DAM
    PAGE 190  ROUTINE FINAL.COVERAGE                8711 8714
    PAGE 192                                        8825
    PAGE 193                                        8856
PT.NAME
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES       711
    PAGE  47  ''SECTION FOR DEFINITIONS             2687
    PAGE 266  ROUTINE BTRY.EFFECTS                  2315 2324
    PAGE 547  ROUTINE P.E.M.INPUT                   5267
    PAGE 672  ''PROGRAM  OLDER.VERSION               154
    PAGE 706                                        2111
P.AATT.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES      1095
    PAGE 679  ''PROGRAM  OLDER.VERSION               536
P.AA.SET
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES      1119
    PAGE 679  ''PROGRAM  OLDER.VERSION               558
P.AO.CAND.DET.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES      1130
    PAGE 679  ''PROGRAM  OLDER.VERSION               569
P.AO.DET.TGT.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES      1038
    PAGE 678  ''PROGRAM  OLDER.VERSION               480
P.AO.EB.SET
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       303
    PAGE 665  ''PROGRAM  OLDER.VERSION              9745
P.AO.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES      1397
    PAGE 684  ''PROGRAM  OLDER.VERSION               836
P.AO.RB.SET
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES       316
    PAGE 665  ''PROGRAM  OLDER.VERSION              9758
```

```
P.AR.CAND.DET.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1140
    PAGE 679  ''PROGRAM  OLDER.VERSION                   579
P.AR.DET.TGT.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1040
    PAGE 678  ''PROGRAM  OLDER.VERSION                   482
P.AVAIL.AO.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1852
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1291
P.BATTLE.SET
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES          1173
    PAGE 680  ''PROGRAM  OLDER.VERSION                   612
P.BN.BTRY.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES           345
    PAGE 666  ''PROGRAM  OLDER.VERSION                  9787
P.BN.CAN.FM.SET
    PAGE  36  ''SECTION FOR PROCESSES                   2019
    PAGE 695  ''PROGRAM  OLDER.VERSION                  1457
P.BTL.FORCE.SET
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1419
    PAGE 684  ''PROGRAM  OLDER.VERSION                   858
P.BY.FM.QUEUE
    PAGE  36  ''SECTION FOR PROCESSES                   2023
    PAGE 695  ''PROGRAM  OLDER.VERSION                  1461
P.BY.HOW.SET
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES          1493
    PAGE 686  ''PROGRAM  OLDER.VERSION                   932
P.BY.SCHD.LIST
    PAGE  36  ''SECTION FOR PROCESSES                   2021
    PAGE 695  ''PROGRAM  OLDER.VERSION                  1459
P.CFPS.LIST
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES          1207
    PAGE 681  ''PROGRAM  OLDER.VERSION                   646
P.CF.OP.Q
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES          1219
    PAGE 681  ''PROGRAM  OLDER.VERSION                   658
P.COL.SET
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES          1189
    PAGE 680  ''PROGRAM  OLDER.VERSION                   628
P.CT.TU.SET
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES           950
    PAGE 676  ''PROGRAM  OLDER.VERSION                   392
P.DF.RATE.LIST
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES          1253
    PAGE 681  ''PROGRAM  OLDER.VERSION                   692
P.E.M.INPUT
    PAGE 242  ROUTINE UNIT.ENVIR                        1121
    PAGE 520  ROUTINE MAIN2                             4224 4226
    PAGE 547  ROUTINE P.E.M.INPUT                       5255
P.FD.BN.LIST
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1353
    PAGE 683  ''PROGRAM  OLDER.VERSION                   792
P.FD.COMPLETE.LIST
    PAGE  38  ''SECTION FOR PROCESSES                   2185
    PAGE 697  ''PROGRAM  OLDER.VERSION                  1623
```

```
P.FD.SCHD.LIST
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1364
    PAGE 683  ''PROGRAM  OLDER.VERSION                   803
P.FD.TR.QUEUE
    PAGE  38  ''SECTION FOR PROCESSES                   2183
    PAGE 697  ''PROGRAM  OLDER.VERSION                  1621
P.FO.CAND.DET.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1429
    PAGE 684  ''PROGRAM  OLDER.VERSION                   868
P.FO.CUR.FM.LIST
    PAGE  36  ''SECTION FOR PROCESSES                   2025
    PAGE 695  ''PROGRAM  OLDER.VERSION                  1463
P.FO.TGT.RPT.LIST
    PAGE  38  ''SECTION FOR PROCESSES                   2187
    PAGE 697  ''PROGRAM  OLDER.VERSION                  1625
P.FP.SET
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1337
    PAGE 683  ''PROGRAM  OLDER.VERSION                   776
P.FR.UNIT.SET
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1036
    PAGE 678  ''PROGRAM  OLDER.VERSION                   478
P.GP.CAT.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES           364
    PAGE 666  ''PROGRAM  OLDER.VERSION                  9806
P.HC.UN.LOS.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1868
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1307
P.HE.TB.RH.LIST
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES           515
    PAGE 669  ''PROGRAM  OLDER.VERSION                  9956
P.HF.SO.LIST
    PAGE  36  ''SECTION FOR PROCESSES                   2072
    PAGE 695  ''PROGRAM  OLDER.VERSION                  1510
P.HIT
    PAGE 260  ROUTINE BTRY.EFFECTS                      1990
    PAGE 268                                            2433 2437 2439 2452
P.HT.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES          1458
    PAGE 685  ''PROGRAM  OLDER.VERSION                   897
P.HT.MEMBER.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES          1478
    PAGE 685  ''PROGRAM  OLDER.VERSION                   917
P.HT.TARGET.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1056
    PAGE 678  ''PROGRAM  OLDER.VERSION                   498
P.IC.TB.RH.LIST
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES           552
    PAGE 669  ''PROGRAM  OLDER.VERSION                  9993
P.IF.RATE.LIST
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1503
    PAGE 686  ''PROGRAM  OLDER.VERSION                   942
P.MADS.RH.SET
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1533
    PAGE 686  ''PROGRAM  OLDER.VERSION                   972
P.MA.SET
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1567
```

```
    PAGE 687  ''PROGRAM  OLDER.VERSION                1006
P.MCFR.RH.LIST
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES         385
    PAGE 686  ''PROGRAM  OLDER.VERSION                9827
P.MFO.RB.SET
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES         472
    PAGE 668  ''PROGRAM  OLDER.VERSION                9913
P.MFP.LIST
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES        1588
    PAGE 687  ''PROGRAM  OLDER.VERSION                1027
P.MO.LIST
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES        1578
    PAGE 687  ''PROGRAM  OLDER.VERSION                1017
P.MPDB.RH.LIST
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         706
    PAGE 672  ''PROGRAM  OLDER.VERSION                 149
P.MU.ORDER.SET
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES        1627
    PAGE 688  ''PROGRAM  OLDER.VERSION                1066
P.MU.TF.LIST
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES        1557
    PAGE 687  ''PROGRAM  OLDER.VERSION                 996
P.PATH.SET
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES        1668
    PAGE 689  ''PROGRAM  OLDER.VERSION                1107
P.PDB.KEYED.LIST
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES        1513
    PAGE 686  ''PROGRAM  OLDER.VERSION                 952
P.PDB.OP.Q
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES        1658
    PAGE 688  ''PROGRAM  OLDER.VERSION                1097
P.PIR.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES        1395
    PAGE 684  ''PROGRAM  OLDER.VERSION                 834
P.PIR.RECORD.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1679
    PAGE 689  ''PROGRAM  OLDER.VERSION                1118
P.PIR.RTD.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1693
    PAGE 689  ''PROGRAM  OLDER.VERSION                1132
P.RPV.CAND.DET.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1718
    PAGE 689  ''PROGRAM  OLDER.VERSION                1157
P.RPV.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES        1399
    PAGE 684  ''PROGRAM  OLDER.VERSION                 838
P.SD.ADS.SET
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES        1109
    PAGE 679  ''PROGRAM  OLDER.VERSION                 548
P.SD.CMSN.QUEUE
    PAGE  35  ''SECTION FOR PROCESSES                 1990
    PAGE 694  ''PROGRAM  OLDER.VERSION                1428
P.SD.FPO.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES        1441
    PAGE 685  ''PROGRAM  OLDER.VERSION                 880
```

```
P.SD.KAS.SET
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1522
    PAGE 686  ''PROGRAM  OLDER.VERSION                   961
P.SD.OLD.SORTIE.QUEUE
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES          1617
    PAGE 688  ''PROGRAM  OLDER.VERSION                  1056
P.SIDE.CFR.SET
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES          1233
    PAGE 681  ''PROGRAM  OLDER.VERSION                   672
P.SIDE.PDB.SET
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES          1646
    PAGE 688  ''PROGRAM  OLDER.VERSION                  1085
P.SI.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1733
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1172
P.SO.LIST
    PAGE  38  ''SECTION FOR PROCESSES                   2143
    PAGE 697  ''PROGRAM  OLDER.VERSION                  1581
P.SS.SET
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1032
    PAGE 678  ''PROGRAM  OLDER.VERSION                   474
P.TB.SORT.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          1017
    PAGE 677  ''PROGRAM  OLDER.VERSION                   459
P.TB.TM.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1756
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1195
P.TEAM.TYPES
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES          1804
    PAGE 691  ''PROGRAM  OLDER.VERSION                  1243
P.TE.SET
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES           419
    PAGE 667  ''PROGRAM  OLDER.VERSION                  9860
P.TR.DET.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1767
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1206
P.TR.FM.LIST
    PAGE  35  ''SECTION FOR PROCESSES                   2017
    PAGE 695  ''PROGRAM  OLDER.VERSION                  1455
P.TU.NTE.SET
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1776
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1215
P.TU.TE.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES          1787
    PAGE 691  ''PROGRAM  OLDER.VERSION                  1226
P.UE.TARGET.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1381
    PAGE 684  ''PROGRAM  OLDER.VERSION                   820
P.UE.WEAPON.SET
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1883
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1322
P.UNIT.SET
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1042
    PAGE 678  ''PROGRAM  OLDER.VERSION                   484
P.UN.EQUIP.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES          1821
```

```
    PAGE 691  ''PROGRAM  OLDER.VERSION                1260
P.UN.HC.LOS.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES        1871
    PAGE 692  ''PROGRAM  OLDER.VERSION                1310
P.UN.LOS.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES        1865
    PAGE 692  ''PROGRAM  OLDER.VERSION                1304
P.UN.PATH
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1703
    PAGE  78  ROUTINE ADJUST                          3948 3949
    PAGE 689  ''PROGRAM  OLDER.VERSION                1142
P.UN.SEGMENT.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES        1744
    PAGE 690  ''PROGRAM  OLDER.VERSION                1183
P.UN.SENSOR.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES        1848
    PAGE 692  ''PROGRAM  OLDER.VERSION                1287
P.UN.SUB.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES        1034
    PAGE 678  ''PROGRAM  OLDER.VERSION                 476
P.X
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1698
    PAGE  46  ''SECTION FOR DEFINITIONS               2628
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                 3865
    PAGE  77  ROUTINE ADJUST                          3919
    PAGE  78                                          3937 3947 3949 3993
    PAGE  83  ROUTINE CHANGE.LOC                      4170
    PAGE  84                                          4233
    PAGE  85                                          4281
    PAGE 110  ROUTINE POSITION                        5276
    PAGE 111  ROUTINE PRED.POS                        5333
    PAGE 321  ROUTINE COMPUTE.D                       4958
    PAGE 322  ROUTINE COMPUTE.WD                      4975 4992
    PAGE 399  EVENT START.BATTLE                      7995 8000
    PAGE 689  ''PROGRAM  OLDER.VERSION                1137
    PAGE 705                                          2051
P.Y
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1699
    PAGE  46  ''SECTION FOR DEFINITIONS               2629
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                 3866
    PAGE  77  ROUTINE ADJUST                          3920
    PAGE  78                                          3938 3946 3948 3994
    PAGE  83  ROUTINE CHANGE.LOC                      4171
    PAGE  84                                          4234
    PAGE  85                                          4282
    PAGE 110  ROUTINE POSITION                        5277
    PAGE 111  ROUTINE PRED.POS                        5334
    PAGE 321  ROUTINE COMPUTE.D                       4959
    PAGE 322  ROUTINE COMPUTE.WD                      4976 4993
    PAGE 399  EVENT START.BATTLE                      7996 8001
    PAGE 689  ''PROGRAM  OLDER.VERSION                1138
    PAGE 705                                          2052
QTY.KILLED
    PAGE 435  PROCESS ARTY.ASSESS                     9714 9739
    PAGE 436                                          9767 9769 9782 9787
    PAGE 437                                          9813 9815 9819 9822 9826 9829 9834
```

```
RADAR.TYPE
    PAGE 206  ROUTINE PDB.DETECTION                9432 9437 9470
    PAGE 358  EVENT CFR.ACTIVATION                 6347 6352 6354 6370
    PAGE 386  EVENT PDB.ACTIVATION                 7475 7479 7481 7498
RAD.OF.EFFECTS
    PAGE 176  ROUTINE EST.COVERAGE                 8015
    PAGE 177                                       8081
    PAGE 178                                       8137 8169 8170
    PAGE 188  ROUTINE FINAL.COVERAGE               8556
    PAGE 189                                       8630 8632
    PAGE 190                                       8687
    PAGE 191                                       8737 8738 8745 8746
RAD.OF.TARGET
    PAGE 188  ROUTINE FINAL.COVERAGE               8555 8562
    PAGE 190                                       8698 8699
RANDI.F
    PAGE 210  ROUTINE PGM.MSN.ASGN                 9636 9638 9641 9643
    PAGE 255  ROUTINE FO.DETECTION                 1775
    PAGE 269  ROUTINE BTRY.EFFECTS                 2508
    PAGE 273                                       2721
    PAGE 280  ROUTINE AC.DF.EFFECTS                3033
    PAGE 282                                       3116
    PAGE 336  ROUTINE GET.TERRAIN                  5443 5446 5449
    PAGE 404  EVENT START.MOVE                     8290
    PAGE 410  EVENT UPDATE.LOC                     8545
    PAGE 428  PROCESS AIR.OBSERVER                 9366
    PAGE 488  PROCESS ASSESSMENT                   2492
    PAGE 491                                       2688
    PAGE 496  PROCESS SHOOT.OUT                    2976
    PAGE 497                                       2993
    PAGE 513  PROCESS HELICOPTER.FIRE              3846 3857
                                                        PAGE 514                3893
    PAGE 516                                       4041
RANDOM.F
    PAGE  96  ROUTINE LINE.OF.SIGHT                4712
    PAGE 123  ROUTINE TIME.TO.DETECT               5835
    PAGE 159  ROUTINE ATTRIT.SENSOR                7332
    PAGE 167  ROUTINE CFR.DETECTION                7685
    PAGE 206  ROUTINE PDB.DETECTION                9453
    PAGE 211  ROUTINE PGM.MSN.ASGN                 9691
    PAGE 213  ROUTINE PIR.DETECTION                9759 9769
    PAGE 230  ROUTINE RPV.DETECTION                 572  582
    PAGE 252  ROUTINE MINE.EFFECTS                 1618
    PAGE 254  ROUTINE FO.DETECTION                 1720
    PAGE 255                                       1756 1778
    PAGE 256                                       1798
    PAGE 268  ROUTINE BTRY.EFFECTS                 2476 2482
    PAGE 269                                       2507
    PAGE 271                                       2609
    PAGE 278  ROUTINE AC.BOMB.EFFECTS              2935
    PAGE 281  ROUTINE AC.DF.EFFECTS                3110
    PAGE 309  ROUTINE AD.SHOOT                     4416
    PAGE 341  ROUTINE PROB.TIME                    5650
    PAGE 350  EVENT AD.ENGAGEMENT                  5993
    PAGE 359  EVENT CFR.OFF                        6394
    PAGE 406  EVENT START.MOVE                     8365
```

```
    PAGE 411  EVENT UPDATE.LOC                    8593
    PAGE 428  PROCESS AIR.OBSERVER                9367
    PAGE 435  PROCESS ARTY.ASSESS                 9734
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION       826
    PAGE 458                                       927
    PAGE 459                                      1034
    PAGE 465  PROCESS MINE.ASSESS                 1287
    PAGE 487  PROCESS ASSESSMENT                  2458
    PAGE 515  PROCESS HELICOPTER.FIRE             3988
    PAGE 523  ROUTINE SYS.INPUT                   4335
    PAGE 637  FUNCTION HE.WLA                     8917
    PAGE 640  ROUTINE EXPONENTIAL.F               9032
    PAGE 641  ROUTINE NORMAL.F                    9047 9048 9057
    PAGE 642  ROUTINE WEIBULL.F                   9068
    PAGE 652  ROUTINE GAMMA.F                     9249 9254
    PAGE 654  PROCESS AIRBORNE.RADAR              9329
RANGES.
    PAGE  77  ROUTINE ADJUST                      3900
RANGE.
    PAGE  77  ROUTINE ADJUST                      3923
    PAGE 138  ROUTINE CHECK.PROX                  6426 6427
    PAGE 139                                      6433 6435 6443 6444 6450 6452
    PAGE 352  EVENT AD.ENGAGEMENT                 6085 6089 6113
    PAGE 353                                      6166
RANGE.RATIO
    PAGE  71  ROUTINE ORIENTATION                 3637 3648
    PAGE  73                                      3796
    PAGE  77  ROUTINE ADJUST                      3884 3902 3903 3913
    PAGE 395  EVENT START.BATTLE                  7754
    PAGE 397                                      7910
    PAGE 399                                      8019 8026
    PAGE 400                                      8034 8045
RANGE.1
    PAGE 115  ROUTINE PROX.CHECK                  5528 5530 5531 5535
RAP.MSN
    PAGE 182  ROUTINE FA.BN.ASGN                  8305
    PAGE 184                                      8433
RATE.
    PAGE 504  PROCESS CAS.MISSION                 3386
RATE.MOVE
    PAGE 403  EVENT START.MOVE                    8210 8216
    PAGE 404                                      8271
    PAGE 406                                      8361 8365
RATE.MOVE.
    PAGE 408  EVENT UPDATE.LOC                    8409
    PAGE 411                                      8590 8593
RATIO.
    PAGE 110  ROUTINE POSITION                    5273 5278 5279 5280
    PAGE 408  EVENT UPDATE.LOC                    8427 8439 8440 8441
RD.ATK.FAIL.PROB
    PAGE  48  ''SECTION FOR DEFINITIONS           2763
    PAGE 450  PROCESS HC.RETURN.FARRP              515
    PAGE 572  ROUTINE FARRP.INPUT                 6367
    PAGE 573                                      6391
    PAGE 707  ''PROGRAM  OLDER.VERSION            2185
```

```
RD.HC.SPACING
     PAGE  42  ''SECTION FOR DEFINITIONS          2395
     PAGE 447  PROCESS HC.ARRIVE.BATTLE            363
     PAGE 572  ROUTINE FARRP.INPUT                6356 6381
     PAGE 574                                     6445 6445
     PAGE 701  ''PROGRAM  OLDER.VERSION           1825
RD.HIGH.FRAC.RANGE
     PAGE  49  ''SECTION FOR DEFINITIONS          2781
     PAGE 302  ROUTINE HEL.RANGE.COMPUTE          4176
     PAGE 447  PROCESS HC.ARRIVE.BATTLE            362
     PAGE 572  ROUTINE FARRP.INPUT                6365
     PAGE 573                                     6390
     PAGE 707  ''PROGRAM  OLDER.VERSION           2203
RD.LOW.FRAC.RANGE
     PAGE  49  ''SECTION FOR DEFINITIONS          2779
     PAGE 302  ROUTINE HEL.RANGE.COMPUTE          4175
     PAGE 447  PROCESS HC.ARRIVE.BATTLE            361
     PAGE 572  ROUTINE FARRP.INPUT                6365
     PAGE 573                                     6389
     PAGE 707  ''PROGRAM  OLDER.VERSION           2201
RD.MAX.FL.TIME
     PAGE  42  ''SECTION FOR DEFINITIONS          2393
     PAGE 299  ROUTINE HC.COMPUTE.TIMES           3976
     PAGE 572  ROUTINE FARRP.INPUT                6355 6380
     PAGE 701  ''PROGRAM  OLDER.VERSION           1823
RD.MAX.HANDOFF.TIME
     PAGE  48  ''SECTION FOR DEFINITIONS          2769
     PAGE 454  PROCESS HEL.TARGET.ACQUISITION      735
     PAGE 572  ROUTINE FARRP.INPUT                6359 6384
     PAGE 707  ''PROGRAM  OLDER.VERSION           2191
RD.MAX.MASK.TIME
     PAGE  48  ''SECTION FOR DEFINITIONS          2773
     PAGE 461  PROCESS HEL.TARGET.ACQUISITION     1139
     PAGE 572  ROUTINE FARRP.INPUT                6361
     PAGE 573                                     6386
     PAGE 707  ''PROGRAM  OLDER.VERSION           2195
RD.MAX.UNMASK.TIME
     PAGE  49  ''SECTION FOR DEFINITIONS          2777
     PAGE 454  PROCESS HEL.TARGET.ACQUISITION      738
     PAGE 572  ROUTINE FARRP.INPUT                6363
     PAGE 573                                     6388
     PAGE 707  ''PROGRAM  OLDER.VERSION           2199
RD.MIN.HANDOFF.TIME
     PAGE  48  ''SECTION FOR DEFINITIONS          2767
     PAGE 454  PROCESS HEL.TARGET.ACQUISITION      735
     PAGE 572  ROUTINE FARRP.INPUT                6359 6383
     PAGE 707  ''PROGRAM  OLDER.VERSION           2189
RD.MIN.MASK.TIME
     PAGE  48  ''SECTION FOR DEFINITIONS          2771
     PAGE 461  PROCESS HEL.TARGET.ACQUISITION     1139
     PAGE 572  ROUTINE FARRP.INPUT                6361
     PAGE 573                                     6385
     PAGE 707  ''PROGRAM  OLDER.VERSION           2193
RD.MIN.UNMASK.TIME
     PAGE  49  ''SECTION FOR DEFINITIONS          2775
     PAGE 454  PROCESS HEL.TARGET.ACQUISITION      737
```

```
    PAGE 572  ROUTINE FARRP.INPUT                    6363
    PAGE 573                                         6387
    PAGE 707  ''PROGRAM  OLDER.VERSION               2197
RD.PRIORITY
    PAGE 396  EVENT START.BATTLE                     7855 7856
RD.ROUNDS.PER.POPUP
    PAGE  42  ''SECTION FOR DEFINITIONS              2388
    PAGE 510  PROCESS HELICOPTER.FIRE                3696
    PAGE 572  ROUTINE FARRP.INPUT                    6368
    PAGE 573                                         6393
    PAGE 701  ''PROGRAM  OLDER.VERSION               1818
RD.SCT.FAIL.PROB
    PAGE  48  ''SECTION FOR DEFINITIONS              2764
    PAGE 450  PROCESS HC.RETURN.FARRP                 516
    PAGE 572  ROUTINE FARRP.INPUT                    6367
    PAGE 573                                         6392
    PAGE 707  ''PROGRAM  OLDER.VERSION               2186
RD.STATUS
    PAGE 147  ROUTINE INTER.BATTLE                   6841 6842
    PAGE 148                                         6878 6879
    PAGE 357  EVENT BTL.ENDED                        6329 6335
RD.UNITS
    PAGE 146  ROUTINE INTER.BATTLE                   6748 6752
    PAGE 147                                         6841
    PAGE 148                                         6855 6859 6860 6862 6863 6890
    PAGE 357  EVENT BTL.ENDED                        6328
    PAGE 413  EVENT ACT.ATK                          8691 8694 8695
RD.UNITS.
    PAGE 356  EVENT BTL.ENDED                        6253 6262 6303
    PAGE 357                                         6311 6319
    PAGE 412  EVENT ACT.ATK                          8616 8646 8650
    PAGE 413                                         8682 8707 8709
RD.WIN
    PAGE 356  EVENT BTL.ENDED                        6300 6308
    PAGE 357                                         6316 6336
READ.
    PAGE 565  ROUTINE SENSOR.INPUT                   6054
READ.ORDERS
    PAGE 520  ROUTINE MAIN2                          4221 4223
    PAGE 539  ROUTINE READ.ORDERS                    4983
REAL.F
    PAGE  90  ROUTINE FA.BN.MOVEMENT                 4466 4486
    PAGE  91                                         4545 4559
    PAGE  92                                         4576
    PAGE 165  ROUTINE CFR.DEGRADE                    7611 7613
    PAGE 166  ROUTINE CFR.DETECTION                  7636 7637 7659 7660 7661
    PAGE 167                                         7678
    PAGE 178  ROUTINE EST.COVERAGE                   8176
    PAGE 180  ROUTINE EST.MIL.WORTH                  8234
    PAGE 183  ROUTINE FA.BN.ASGN                     8384 8385 8388
    PAGE 190  ROUTINE FINAL.COVERAGE                 8701 8707 8709 8716 8718 8720
    PAGE 191                                         8755 8756 8757 8758
    PAGE 194  ROUTINE FIND.START.TIME                8906 8928
    PAGE 195                                         8987
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9354
    PAGE 206  ROUTINE PDB.DETECTION                  9472 9474
```

PAGE 209	ROUTINE PGM.MSN.ASGN	9552 9553
PAGE 214	ROUTINE REM.EFFECTS.COMPUTATION	9809 9810
PAGE 231	ROUTINE SIZE.ESTIMATE	673 674
PAGE 248	ROUTINE WEIGHTED.VOLLEYS	1416
PAGE 251	ROUTINE MINE.EFFECTS	1561 1562
PAGE 262	ROUTINE BTRY.EFFECTS	2130
PAGE 263		2154 2178 2192
PAGE 265		2258 2265 2280
PAGE 266		2315 2316 2317 2324 2325 2326 2333 2334 2335 2337 2338 2339
PAGE 270		2587 2588
PAGE 278	ROUTINE AC.BOMB.EFFECTS	2918 2919
PAGE 287	ROUTINE CHECK.CAS.CONSTRAINTS	3445
PAGE 316	ROUTINE FLIGHT.PATH	4768
PAGE 317		4856
PAGE 318		4870
PAGE 319		4923
PAGE 350	EVENT AD.ENGAGEMENT	5956
PAGE 351		6035 6063
PAGE 352		6083 6098
PAGE 353		6159 6160
PAGE 354		6209
PAGE 360	EVENT CFR.ON	6458
PAGE 362	EVENT CFR.OPERATOR	6492 6493
PAGE 387	EVENT PDB.OPERATOR	7516 7517
PAGE 420	PROCESS AC.ATK.TGT	8914 8917
PAGE 422		9026 9027 9038 9076
PAGE 428	PROCESS AIR.OBSERVER	9379
PAGE 432		9562 9563
PAGE 440	PROCESS FORWARD.OBSERVER	26 27
PAGE 446	PROCESS HC.ARRIVE.BATTLE	319 319
PAGE 447		347 347
PAGE 463	PROCESS HOW.REPAIR	1190 1194 1204 1209
PAGE 468	PROCESS REMOTE.PILOT.VEHICLE	1453 1454
PAGE 473	PROCESS TARGET.REPORT	1678
PAGE 475		1808 1809
PAGE 480	PROCESS FIRE.MISSION	2079
PAGE 481		2086 2092 2093
PAGE 482		2149 2150
PAGE 485		2349 2366
PAGE 504	PROCESS CAS.MISSION	3397 3398
PAGE 506		3514 3520
PAGE 507		3544 3545
PAGE 551	ROUTINE BTRY.INPUT	5496 5502
PAGE 584	ROUTINE TACAIR.INPUT	6855 6857 6859 6879 6880 6885 6888 6891
PAGE 585		6951 6952 6961
PAGE 628	FUNCTION ACT.RANGE	8641
PAGE 629	FUNCTION BTRY.AVAILABLE	8656 8662 8666
PAGE 631	FUNCTION COMBINATIONS	8742 8742
PAGE 636	FUNCTION HE.WLA	8899 8910
PAGE 637		8933 8940 8944 8947
PAGE 638	FUNCTION ICM.WLA	9004 9006 9010 9014
PAGE 639		9018 9019
PAGE 644	ROUTINE ANGLE.COMPUTE	9112 9112
PAGE 654	PROCESS AIRBORNE.RADAR	9364 9365
PAGE 659	PROCESS PHOTO.IR.FLIGHT	9566 9567

```
REAL.TIME.CPU
    PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT          8807 8817 8820
REAL.UNIT
    PAGE 550  ROUTINE BTRY.INPUT                     5442
    PAGE 551                                         5446 5450
REAL.UN.NO.
    PAGE 539  ROUTINE READ.ORDERS                    5011 5014 5019 5028
    PAGE 540                                         5044
REAL.VOLLEYS
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS               1388
    PAGE 249                                         1441 1445 1450 1452 1454 1455
REARMING.REFUELING
    PAGE  52  ''SECTION FOR SUBSTITUTIONS            2986
    PAGE 451  PROCESS HC.RETURN.FARRP                 562
    PAGE 711  ''PROGRAM  OLDER.VERSION               2393
REARM.CAP
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES       1336
    PAGE 451  PROCESS HC.RETURN.FARRP                 564
    PAGE 573  ROUTINE FARRP.INPUT                    6429 6436
    PAGE 683  ''PROGRAM  OLDER.VERSION                775
REARM.TIME
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES       1334
    PAGE 451  PROCESS HC.RETURN.FARRP                 565
    PAGE 573  ROUTINE FARRP.INPUT                    6428 6436
    PAGE 683  ''PROGRAM  OLDER.VERSION                773
REAR.
    PAGE 142  ROUTINE DEAD.UNIT                      6531
RECORDS.....
    PAGE 176  ROUTINE EST.COVERAGE                   8039
    PAGE 177                                         8095
    PAGE 188  ROUTINE FINAL.COVERAGE                 8582
    PAGE 189                                         8639
REC.TARGET
    PAGE 212  ROUTINE PIR.DETECTION                  9703 9714 9715
    PAGE 213                                         9778 9781
    PAGE 656  ROUTINE AR.DETECTION                   9392
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                9523
    PAGE 659                                         9541 9543 9546 9547 9557 9559 9560 9562 9565 9570
REDUCED.
    PAGE  77  ROUTINE ADJUST                         3891
RED.
    PAGE 120  ROUTINE SEARCH                         5734
    PAGE 314  ROUTINE FLIGHT.PATH                    4645
RED.AH.TEAM
    PAGE  52  ''SECTION FOR SUBSTITUTIONS            2949
    PAGE 710  ''PROGRAM  OLDER.VERSION               2356
    PAGE 714  ROUTINE PLAT.COUNT                     2534
RED.CO.HQ
    PAGE  51  ''SECTION FOR SUBSTITUTIONS            2944
    PAGE 569  ROUTINE TBF.INPUT                      6255
    PAGE 710  ''PROGRAM  OLDER.VERSION               2351
    PAGE 714  ROUTINE PLAT.COUNT                     2538
RED.HB.PRIORITY
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES       1162
    PAGE 128  ROUTINE BTL.CHECK                      6026 6037 6039
    PAGE 129                                         6051
```

```
    PAGE 311  ROUTINE INTER.HELO                        4529
    PAGE 396  EVENT START.BATTLE                        7856
    PAGE 397                                            7859
    PAGE 401                                            8116 8133
    PAGE 448  PROCESS HC.ARRIVE.BATTLE                   421
    PAGE 680  ''PROG*AM  OLDER.VERSION                   601
RED.INF.PLT
    PAGE  51  ''SECTION FOR SUBSTITUTIONS               2947
    PAGE 569  ROUTINE TBF.INPUT                         6253
    PAGE 710  ''PROGRAM  OLDER.VERSION                  2354
    PAGE 714  ROUTINE PLAT.COUNT                        2522
RED.MECH.PLT
    PAGE  51  ''SECTION FOR SUBSTITUTIONS               2942
    PAGE 569  ROUTINE TBF.INPUT                         6251
    PAGE 710  ''PROGRAM  OLDER.VERSION                  2349
    PAGE 714  ROUTINE PLAT.COUNT                        2526
RED.MISSION
    PAGE  41  ''SECTION FOR EVENTS                      2314
    PAGE 395  EVENT START.BATTLE                        7747
    PAGE 396                                            7817
    PAGE 397                                            7867 7886
    PAGE 398                                            7952
    PAGE 399                                            8010 8011 8012
    PAGE 400                                            8037 8038 8039
    PAGE 700  ''PROGRAM  OLDER.VERSION                  1751
RED.N.KV
    PAGE 529  ROUTINE KV.INPUT                          4536 4542 4558 4560
    PAGE 609  ROUTINE KV.SCOREBOARD                     7845 7848 7850 7852
RED.TK.PLT
    PAGE  51  ''SECTION FOR SUBSTITUTIONS               2941
    PAGE 569  ROUTINE TBF.INPUT                         6249
    PAGE 710  ''PROGRAM  OLDER.VERSION                  2348
    PAGE 714  ROUTINE PLAT.COUNT                        2530
RED.UNITS
    PAGE  41  ''SECTION FOR EVENTS                      2316
    PAGE  67  ROUTINE GENERAL.BATTLE                    3451 3456 3459 3493 3494
    PAGE  68                                            3552 3558 3562 3563
    PAGE  69                                            3566 3573
    PAGE  96  ROUTINE LINE.OF.SIGHT                     4665 4677 4680 4688 4691 4716
    PAGE  97                                            4722 4730 4746 4763
    PAGE 395  EVENT START.BATTLE                        7749 7759 7763 7770 7772
    PAGE 396                                            7802 7816 7833
    PAGE 397                                            7899
    PAGE 398                                            7933
    PAGE 399                                            8016
    PAGE 400                                            8042 8072
    PAGE 700  ''PROGRAM  OLDER.VERSION                  1753
REFUEL.CAP
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1335
    PAGE 451  PROCESS HC.RETURN.FARRP                    563
    PAGE 573  ROUTINE FARRP.INPUT                       6429 6436
    PAGE 683  ''PROGRAM  OLDER.VERSION                   774
REFUEL.TIME
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1333
    PAGE 451  PROCESS HC.RETURN.FARRP                    564
    PAGE 573  ROUTINE FARRP.INPUT                       6428 6435
```

```
    PAGE 683  ''PROGRAM  OLDER.VERSION                772
REINFORCED.UN
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES       1609
    PAGE  95  ROUTINE INIT.REINF                     4652
    PAGE 118  ROUTINE REIN.ARRIVE                    5614
    PAGE 408  EVENT UPDATE.LOC                       8431 8432
    PAGE 688  ''PROGRAM  OLDER.VERSION               1048
REINFORCEMENTS.
    PAGE 347  EVENT ACT.REINF                        5811
    PAGE 544  ROUTINE ORD.REINF                      5164
REINFORCEMENT.
    PAGE 347  EVENT ACT.REINF                        5816
REINF.
    PAGE 534  ROUTINE UNIT.INPUT                     4744
REINF.ORDER
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES       1707
    PAGE 326  ROUTINE DESTROY.ORD                    5130
    PAGE 544  ROUTINE ORD.REINF                      5173
    PAGE 689  ''PROGRAM  OLDER.VERSION               1146
    PAGE 713                                         2499
REINF.THRESH
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES       1242
    PAGE 144  ROUTINE DEAD.UNIT                      6666
    PAGE 542  ROUTINE ORD.DEF                        5132
    PAGE 681  ''PROGRAM  OLDER.VERSION                681
REINORDER
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           53
    PAGE  39  ''SECTION FOR EVENTS                   2223
    PAGE 698  ''PROGRAM  OLDER.VERSION               1660
REINUNIT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           52
    PAGE  39  ''SECTION FOR EVENTS                   2222
    PAGE 698  ''PROGRAM  OLDER.VERSION               1659
REIN.DELAY
    PAGE  42  ''SECTION FOR DEFINITIONS              2383
    PAGE  95  ROUTINE INIT.REINF                     4661
    PAGE 523  ROUTINE SYS.INPUT                      4374
    PAGE 701  ''PROGRAM  OLDER.VERSION               1813
REIN.PROX
    PAGE  42  ''SECTION FOR DEFINITIONS              2384
    PAGE 347  EVENT ACT.REINF                        5849
    PAGE 523  ROUTINE SYS.INPUT                      4372
    PAGE 524                                         4377 4377
    PAGE 701  ''PROGRAM  OLDER.VERSION               1814
REIN.THRESH
    PAGE  42  ''SECTION FOR DEFINITIONS              2385
    PAGE 348  EVENT ACT.REINF                        5864
    PAGE 523  ROUTINE SYS.INPUT                      4373
    PAGE 701  ''PROGRAM  OLDER.VERSION               1815
REMOTE.PILOT.VEHICLE
    PAGE  37  ''SECTION FOR PROCESSES                2114
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE           1356 1373
    PAGE 474  PROCESS TARGET.REPORT                  1742
    PAGE 485  PROCESS FIRE.MISSION                   2352
    PAGE 600  ROUTINE BETWEEN.ROUTINE                7508 7509
    PAGE 616  ROUTINE SNAP2                          8156
```

```
    PAGE 696  ''PROGRAM  OLDER.VERSION                1552
REM.EFF
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION         9823
    PAGE 475  PROCESS TARGET.REPORT                   1775
REM.EFFECTS.COMPUTATION
    PAGE 185  ROUTINE FA.BN.ASGN                      8510
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION         9791 9806
REPF.LA.PERS
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES         520
    PAGE 190  ROUTINE FINAL.COVERAGE                  8718
    PAGE 266  ROUTINE BTRY.EFFECTS                    2317
    PAGE 558  ROUTINE HE.LA.INPUT                     5780 5781 5783
    PAGE 637  FUNCTION HE.WLA                         8940
    PAGE 669  ''PROGRAM  OLDER.VERSION                9961
REPLACE.HC
    PAGE 304  ROUTINE REPLACE.HC                      4187 4209 4229
    PAGE 452  PROCESS HC.RETURN.FARRP                  582
    PAGE 453                                           643
REPORTED.
    PAGE  65  ROUTINE FILE.KAD.SENSOR                 3371
REPORT.
    PAGE 182  ROUTINE FA.BN.ASGN                      8294
    PAGE 294  ROUTINE END.CAS.MISSION                 3760
REP.PREC
    PAGE 178  ROUTINE EST.COVERAGE                    8144 8155 8164
    PAGE 190  ROUTINE FINAL.COVERAGE                  8694
    PAGE 191                                          8734
REP.TM
    PAGE   3  PROGRAM REVISIONS                        172
    PAGE 178  ROUTINE EST.COVERAGE                    8146 8148
    PAGE 179                                          8182 8186 8191 8194 8197
    PAGE 190  ROUTINE FINAL.COVERAGE                  8696
    PAGE 192                                          8832 8833
    PAGE 193                                          8838 8856 8862
REQUESTED.
    PAGE 283  ROUTINE CAS.EVAL                        3176
    PAGE 542  ROUTINE ORD.DEF                         5121
REQUEST.FASC
    PAGE 217  ROUTINE REQUEST.FASCAM                  9975
REQUEST.ILLU
    PAGE 221  ROUTINE REQUEST.ILLUM                    156  166
REQUEST.SMOK
    PAGE 225  ROUTINE REQUEST.SMOKE                    375
REQUIRED.
    PAGE 474  PROCESS TARGET.REPORT                   1762
REQ.EFF
    PAGE 475  PROCESS TARGET.REPORT                   1775
REQ.EFFECTS
    PAGE 187  ROUTINE FD.EFFECTS.REQ                  8526 8534 8536 8542
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION         9801 9810 9812 9816
REQ.EFF.MOVING
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         732
    PAGE  42  ''SECTION FOR DEFINITIONS               2399
    PAGE 187  ROUTINE FD.EFFECTS.REQ                  8534
    PAGE 523  ROUTINE SYS.INPUT                       4347
    PAGE 672  ''PROGRAM  OLDER.VERSION                 175
```

```
    PAGE 701                                          1829
REQ.EFF.STA
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         733
    PAGE  42  ''SECTION FOR DEFINITIONS               2400
    PAGE 187  ROUTINE FD.EFFECTS.REQ                  8536
    PAGE 523  ROUTINE SYS.INPUT                       4348
    PAGE 672  ''PROGRAM  OLDER.VERSION                 176
    PAGE 701                                          1830
REQ.PROXIMITY
    PAGE 171  ROUTINE COMPARE.TRS                     7833 7835
    PAGE 662  ROUTINE PROXIMITY.REQ                   9631 9636
REQ.TIME
    PAGE 171  ROUTINE COMPARE.TRS                     7807 7821 7822
    PAGE 663  ROUTINE TIME.REQ                        9645 9649 9651
RESET.FEBA.SECTOR
    PAGE  62  ROUTINE FEBA.INITIAL                    3247
    PAGE 119  ROUTINE RESET.FEBA.SECTOR               5647 5658
    PAGE 331  ROUTINE ENQ.FEBA.SET                    5338
RETURNED.
    PAGE 240  ROUTINE TARGET.ANALYSIS                 1072
RETURNING.FROM.BATTLE
    PAGE  52  ''SECTION FOR SUBSTITUTIONS             2983
    PAGE 338  ROUTINE HC.EMPTY                        5550
    PAGE 339                                          5592
    PAGE 375  EVENT HC.DEPART.BATTLE                  7025
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                 138
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION          1070
    PAGE 461                                          1146
    PAGE 711  ''PROGRAM  OLDER.VERSION                2390
RETURN.
    PAGE 421  PROCESS AC.ATK.TGT                      8974
RETURN.MINUTE
    PAGE 428  PROCESS AIR.OBSERVER                    9358
    PAGE 429                                          9410 9440
    PAGE 430                                          9459
    PAGE 431                                          9507 9515 9516
    PAGE 432                                          9594 9599 9600
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE            1398
    PAGE 468                                          1441 1461
    PAGE 653  PROCESS AIRBORNE.RADAR                  9300 9304
    PAGE 654                                          9344 9369
RETURN.TIME
    PAGE 428  PROCESS AIR.OBSERVER                    9346 9385 9386
    PAGE 429                                          9410
    PAGE 433                                          9656
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE            1369 1396 1398
    PAGE 468                                          1469
    PAGE 653  PROCESS AIRBORNE.RADAR                  9274 9298 9300
    PAGE 655                                          9378
RE.COUNT
    PAGE 139  ROUTINE CHECK.PROX                      6476 6482 6486 6488
RF.FARRP
    PAGE  36  ''SECTION FOR PROCESSES                 2059
    PAGE 337  ROUTINE HC.EMPTY                        5475
    PAGE 695  ''PROGRAM  OLDER.VERSION                1497
```

```
RF.TEAM
    PAGE  36  ''SECTION FOR PROCESSES                2060
    PAGE 337  ROUTINE HC.EMPTY                       5477 5478 5480 5482
    PAGE 695  ''PROGRAM  OLDER.VERSION               1498
RH.FARRP
    PAGE 304  ROUTINE REPLACE.HC                     4189 4200 4207 4223 4224 4228 4228
RH.TEAM
    PAGE 304  ROUTINE REPLACE.HC                     4190 4198 4207 4216 4218 4224 4225
RINF.C
    PAGE 108  ROUTINE NEW.SEGMENT                    5188
    PAGE 120  ROUTINE SEARCH                         5721
    PAGE 123  ROUTINE TIME.TO.DETECT                 5814
    PAGE 134  ROUTINE CHECK.FOR.MINES                6240
    PAGE 135                                         6286 6292 6292 6293
    PAGE 159  ROUTINE ATTRIT.SENSOR                  7366
    PAGE 160                                         7391 7407
    PAGE 191  ROUTINE FINAL.COVERAGE                 8776
    PAGE 192                                         8779 8793 8796
    PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION          9073 9077 9093
    PAGE 238  ROUTINE SWITCH.FO                       946  952
    PAGE 284  ROUTINE CAS.EVAL                       3234 3246
    PAGE 301  ROUTINE HC.DISENGAGE                   4079 4104
    PAGE 311  ROUTINE INTER.HELO                     4551
    PAGE 312                                         4622
    PAGE 328  ROUTINE EMPTY                          5203 5212
    PAGE 329                                         5262
    PAGE 337  ROUTINE HC.EMPTY                       5512
    PAGE 338                                         5526
    PAGE 341  ROUTINE PROB.TIME                      5648
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                135
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION          881
    PAGE 459                                         1015
RND.INT.CPE
    PAGE 176  ROUTINE EST.COVERAGE                   8016 8060
    PAGE 177                                         8070 8074 8116
    PAGE 178                                         8126 8130 8144 8145
    PAGE 188  ROUTINE FINAL.COVERAGE                 8557
    PAGE 189                                         8606 8619 8623
    PAGE 190                                         8663 8676 8680 8694 8695
RNG.
    PAGE  96  ROUTINE LINE.OF.SIGHT                  4701 4713
    PAGE 101  ROUTINE LOS.CHECK                      4894 4896 4901 4903 4912
    PAGE 107  ROUTINE NEW.SEGMENT                    5123 5126 5131 5134 5138 5139
RNG.BND
    PAGE 255  ROUTINE FO.DETECTION                   1741 1742 1752 1769
    PAGE 257                                         1867
RNG.HACK
    PAGE 176  ROUTINE EST.COVERAGE                   8011 8015 8027 8028 8031 8040 8042 8043 8046 8047 8048 8049
    PAGE 177                                         8073 8073 8075 8079 8083 8084 8087 8096 8098 8099 8102 8103 8104 8105
    PAGE 178                                         8129 8129 8131 8135
    PAGE 188  ROUTINE FINAL.COVERAGE                 8555 8570 8571 8583 8585 8586 8589 8590 8591 8592
    PAGE 189                                         8622 8622 8624 8628 8635 8636 8640 8642 8643 8646 8647 8648 8649
    PAGE 190                                         8679 8679 8681 8685 8718 8720
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS               1406 1407 1409 1410
    PAGE 263  ROUTINE BTRY.EFFECTS                   2166 2167 2183 2184
    PAGE 265                                         2258
```

```
    PAGE 266                                                2317
    PAGE 272                                                2681
    PAGE 636  FUNCTION HE.WLA                               8882 8883
    PAGE 637                                                8933 8940
    PAGE 638  FUNCTION ICM.WLA                              8978 8979
RN.SEED
    PAGE  42  ''SECTION FOR DEFINITIONS                     2370
    PAGE  96  ROUTINE LINE.OF.SIGHT                         4712
    PAGE  97                                                4739 4758
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA                    4791 4802
    PAGE 104  ROUTINE MINE.DELAY                            5004
    PAGE 109  ROUTINE NEW.SEGMENT                           5257
    PAGE 115  ROUTINE PROX.CHECK                            5532
    PAGE 123  ROUTINE TIME.TO.DETECT                        5835
    PAGE 157  ROUTINE AO.DETECTION                          7257
    PAGE 193  ROUTINE FINAL.COVERAGE                        8879 8880
    PAGE 210  ROUTINE PGM.MSN.ASGN                          9619 9636 9638 9641 9643
    PAGE 211                                                9691
    PAGE 242  ROUTINE UNIT.ENVIR                            1155
    PAGE 243                                                1214 1220
    PAGE 244                                                1230 1241 1264 1273
    PAGE 251  ROUTINE MINE.EFFECTS                          1562
    PAGE 252                                                1618
    PAGE 254  ROUTINE FO.DETECTION                          1720
    PAGE 255                                                1756 1775 1778
    PAGE 256                                                1798
    PAGE 268  ROUTINE BTRY.EFFECTS                          2476 2482
    PAGE 269                                                2507 2508 2542
    PAGE 271                                                2609
    PAGE 273                                                2721
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                       2843 2844
    PAGE 278                                                2919 2930 2935 2958
    PAGE 280  ROUTINE AC.DF.EFFECTS                         3033
    PAGE 281                                                3110
    PAGE 282                                                3116
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE                     4154 4173 4176
    PAGE 309  ROUTINE AD.SHOOT                              4416
    PAGE 310                                                4502
    PAGE 336  ROUTINE GET.TERRAIN                           5443 5446 5449
    PAGE 341  ROUTINE PROB.TIME                             5650
    PAGE 350  EVENT AD.ENGAGEMENT                           5993
    PAGE 351                                                6024 6042
    PAGE 359  EVENT CFR.OFF                                 6394
    PAGE 362  EVENT CFR.OPERATOR                            6519 6521
    PAGE 371  EVENT FEBA.SORTIE                             6827
    PAGE 372                                                6869 6873
    PAGE 404  EVENT START.MOVE                              8291
    PAGE 406                                                8365
    PAGE 410  EVENT UPDATE.LOC                              8546
    PAGE 411                                                8593
    PAGE 422  PROCESS AC.ATK.TGT                            9027 9076
    PAGE 428  PROCESS AIR.OBSERVER                          9366 9367 9381
    PAGE 429                                                9397
    PAGE 432                                                9563
    PAGE 435  PROCESS ARTY.ASSESS                           9734
    PAGE 436                                                9767
```

```
    PAGE 438  PROCESS FORWARD.OBSERVER                9904
    PAGE 440                                          9989   13
    PAGE 441                                            32   36   78
    PAGE 447  PROCESS HC.ARRIVE.BATTLE                 366
    PAGE 451  PROCESS HC.RETURN.FARRP                  520  522  526  540
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION           730  732  736  738
    PAGE 456                                           826
    PAGE 458                                           927
    PAGE 459                                          1034
    PAGE 461                                          1137 1140
    PAGE 463  PROCESS HOW.REPAIR                      1194 1209
    PAGE 465  PROCESS MINE.ASSESS                     1287
    PAGE 479  PROCESS WITH.DRAW                       1984 2007
    PAGE 482  PROCESS FIRE.MISSION                    2150
    PAGE 487  PROCESS ASSESSMENT                      2458
    PAGE 488                                          2492
    PAGE 496  PROCESS SHOOT.OUT                       2976
    PAGE 497                                          2993
    PAGE 504  PROCESS CAS.MISSION                     3398 3402
    PAGE 513  PROCESS HELICOPTER.FIRE                 3846 3857
    PAGE 514                                          3893
    PAGE 515                                          3988
    PAGE 516                                          4042
    PAGE 523  ROUTINE SYS.INPUT                       4324 4331 4332 4333 4334 4336
    PAGE 551  ROUTINE BTRY.INPUT                      5497
    PAGE 552                                          5503
    PAGE 585  ROUTINE TACAIR.INPUT                    6953
    PAGE 599  ROUTINE BETWEEN.ROUTINE                 7437 7441
    PAGE 637  FUNCTION HE.WLA                         8917
    PAGE 700  ''PROGRAM  OLDER.VERSION                1800
ROUNAM.DAT
    PAGE 647  ROUTINE OPEN.INPUT.OUTPUT.FILES         9184
ROUNDS.
    PAGE 420  PROCESS AC.ATK.TGT                      8947
ROUND.NAME
    PAGE 260  ROUTINE BTRY.EFFECTS                    1984
    PAGE 262                                          2124 2133
    PAGE 263                                          2141 2165 2182
    PAGE 264                                          2197 2207
    PAGE 271                                          2650
    PAGE 272                                          2661 2675 2694
    PAGE 611  ROUTINE OUTPUT.ATTRITION                7932 7963 7964 7965 7966 7984
ROUND.RELY
    PAGE 190  ROUTINE FINAL.COVERAGE                  8701 8716
    PAGE 192                                          8820 8824
    PAGE 193                                          8855
    PAGE 260  ROUTINE BTRY.EFFECTS                    1983
    PAGE 262                                          2130 2134
    PAGE 263                                          2141 2178 2192
    PAGE 264                                          2199
    PAGE 268                                          2452
    PAGE 272                                          2681
ROUTINE.
    PAGE 115  ROUTINE PROX.CHECK                      5511
    PAGE 138  ROUTINE CHECK.PROX                      6381 6383
    PAGE 212  ROUTINE PIR.DETECTION                   9726
```

```
    PAGE 242  ROUTINE UNIT.ENVIR                    1121 1129
    PAGE 564  ROUTINE MAO.INPUT                     5958
RPV.CAND.DET.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1716
    PAGE  37  ''SECTION FOR PROCESSES               2121
    PAGE  49  ''SECTION FOR DEFINITIONS             2819
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1435 1439 1444
    PAGE 469                                        1475 1477
    PAGE 689  ''PROGRAM  OLDER.VERSION              1155
    PAGE 696                                        1559
    PAGE 708                                        2240
RPV.CURRENT.TARGET
    PAGE  37  ''SECTION FOR PROCESSES               2115
    PAGE 696  ''PROGRAM  OLDER.VERSION              1553
RPV.CURRENT.TR
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1389
    PAGE 468                                        1463
RPV.DC.DIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1714
    PAGE  46  ''SECTION FOR DEFINITIONS             2632
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1434
    PAGE 689  ''PROGRAM  OLDER.VERSION              1153
    PAGE 705                                        2055
RPV.DC.LEG.DIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1713
    PAGE  46  ''SECTION FOR DEFINITIONS             2631
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1433 1445 1446
    PAGE 689  ''PROGRAM  OLDER.VERSION              1152
    PAGE 705                                        2054
RPV.DC.UNIT
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1712
    PAGE  46  ''SECTION FOR DEFINITIONS             2630
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1432
    PAGE 689  ''PROGRAM  OLDER.VERSION              1151
    PAGE 705                                        2053
RPV.DETECTION
    PAGE 229  ROUTINE RPV.DETECTION                  512  527
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1448
RPV.DETECTION.
    PAGE 242  ROUTINE UNIT.ENVIR                    1128
RPV.DET.CANDIDATE
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES      1711
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1431 1452
    PAGE 469                                        1478
    PAGE 689  ''PROGRAM  OLDER.VERSION              1150
RPV.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES      1391
    PAGE  37  ''SECTION FOR PROCESSES               2120
    PAGE  49  ''SECTION FOR DEFINITIONS             2820
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1400
    PAGE 468                                        1470
    PAGE 469                                        1472
    PAGE 684  ''PROGRAM  OLDER.VERSION               830
    PAGE 696                                        1558
    PAGE 708                                        2241
```

```
RPV.US.LINK
    PAGE  37  ''SECTION FOR PROCESSES                2116
    PAGE 229  ROUTINE RPV.DETECTION                   529
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE           1375
    PAGE 696  ''PROGRAM  OLDER.VERSION               1554
RPV.X.START
    PAGE  37  ''SECTION FOR PROCESSES                2117
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE           1392
    PAGE 696  ''PROGRAM  OLDER.VERSION               1555
RPV.Y.START
    PAGE  37  ''SECTION FOR PROCESSES                2118
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE           1391
    PAGE 696  ''PROGRAM  OLDER.VERSION               1556
RRECORD.V
    PAGE 522  ROUTINE MAIN2                          4309
RR.TIME
    PAGE 451  PROCESS HC.RETURN.FARRP                 563  567  573
    PAGE 453                                          672  675
RSA.A
    PAGE 219  ROUTINE REQUEST.ILLUM                    94
    PAGE 224  ROUTINE REQUEST.SMOKE                   292
RTEF.LA.EQUIP
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES        523
    PAGE 190  ROUTINE FINAL.COVERAGE                 8720
    PAGE 265  ROUTINE BTRY.EFFECTS                   2258
    PAGE 559  ROUTINE HE.LA.INPUT                    5811 5812 5814
    PAGE 637  FUNCTION HE.WLA                        8933
    PAGE 669  ''PROGRAM  OLDER.VERSION               9964
RUL.EN.INPUT
    PAGE 521  ROUTINE MAIN2                          4245 4247
    PAGE 560  ROUTINE  RUL.EN.INPUT                  5820
RUNSTREAM.
    PAGE 606  ROUTINE KV.PRINT                       7720
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES            8501
RUN.
    PAGE 522  ROUTINE MAIN2                          4299
R.
    PAGE 664  ''PROGRAM  OLDER.VERSION               9668
R.H
    PAGE 558  ROUTINE HE.LA.INPUT                    5770 5771 5801 5802
R.HQ.INC
    PAGE  69  ROUTINE GENERAL.BATTLE                 3592 3621
R.HQ.W
    PAGE  69  ROUTINE GENERAL.BATTLE                 3584 3586
R.MAN.INC
    PAGE  69  ROUTINE GENERAL.BATTLE                 3594
    PAGE  70                                         3628
R.MAN.W
    PAGE  69  ROUTINE GENERAL.BATTLE                 3585 3586
R.RANGE
    PAGE 556  ROUTINE MUNS.INPUT                     5667 5670 5702 5705
R.T
    PAGE  78  ROUTINE ADJUST                         3975 3976 3982
R.UNIT
    PAGE 347  EVENT ACT.REINF                        5805 5820 5828 5831 5835
    PAGE 348                                         5866 5872 5883 5890
```

```
R.WIDTH.UNIT
    PAGE  42  ''SECTION FOR DEFINITIONS              2371
    PAGE  69  ROUTINE GENERAL.BATTLE                 3584 3585
    PAGE 569  ROUTINE TBF.INPUT                      6211 6216 6216
    PAGE 701  ''PROGRAM  OLDER.VERSION               1801
SADARM.THRESHOLD
    PAGE  42  ''SECTION FOR DEFINITIONS              2406
    PAGE 157  ROUTINE AO.DETECTION                   7298
    PAGE 257  ROUTINE FO.DETECTION                   1900
    PAGE 258                                         1942
    PAGE 575  ROUTINE PGM.INPUT                      6488
    PAGE 701  ''PROGRAM  OLDER.VERSION               1836
SAD.KILL
    PAGE 260  ROUTINE BTRY.EFFECTS                   1989
    PAGE 269                                         2503 2508 2514 2515 2521
SAVE.OUTPUT.UNIT
    PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT          8805 8809 8836
    PAGE 619  ROUTINE SNAP.R                         8311 8313
    PAGE 620                                         8329
SB.X.COORD
    PAGE 447  PROCESS HC.ARRIVE.BATTLE                368  372  377  381
    PAGE 448                                          392
SB.Y.COORD
    PAGE 447  PROCESS HC.ARRIVE.BATTLE                369  373  377  381
    PAGE 448                                          393
SCAN.RATE
    PAGE  42  ''SECTION FOR DEFINITIONS              2397
    PAGE 523  ROUTINE SYS.INPUT                      4364
    PAGE 701  ''PROGRAM  OLDER.VERSION               1827
SCHEDULED.
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          703
SCHEDULE.ARTY.MOVEMENT
    PAGE  40  ''SECTION FOR EVENTS                   2301
    PAGE  57  ROUTINE MAIN3                          3076
    PAGE 390  EVENT SCHEDULE.ARTY.MOVEMENT           7579 7584
    PAGE 602  ROUTINE BETWEEN.ROUTINE                7624 7625
    PAGE 616  ROUTINE SNAP2                          8186
    PAGE 699  ''PROGRAM  OLDER.VERSION               1738
SCHED.
    PAGE 225  ROUTINE REQUEST.SMOKE                   376
SCT.ATTRITION
    PAGE 451  PROCESS HC.RETURN.FARRP                 540  541  551  555  557
SCT.FAIL.PROB
    PAGE 450  PROCESS HC.RETURN.FARRP                 472  513  516
    PAGE 451                                          520  522  539  540
SCT.HC
    PAGE  50  ''SECTION FOR DEFINITIONS              2839
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS             3499
    PAGE 297  ROUTINE FARRP.CHECK                    3877
    PAGE 452  PROCESS HC.RETURN.FARRP                 604
    PAGE 572  ROUTINE FARRP.INPUT                    6340 6343 6376
    PAGE 708  ''PROGRAM  OLDER.VERSION               2260
SCT.UE.ID
    PAGE 128  ROUTINE BTL.CHECK                      5987 6013 6032
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS             3502
    PAGE 290                                         3531 3557
```

```
    PAGE 291                                        3594 3646
    PAGE 297  ROUTINE FARRP.CHECK                   3878 3894
    PAGE 298  ROUTINE HC.COMPUTE.TIMES              3917 3960
    PAGE 450  PROCESS HC.RETURN.FARRP                484  495
    PAGE 452                                         605
    PAGE 453                                         634
SDM.CB
    PAGE  42  ''SECTION FOR DEFINITIONS             2407
    PAGE 575  ROUTINE PGM.INPUT                     6488
    PAGE 701  ''PROGRAM  OLDER.VERSION              1837
SDM.MARK
    PAGE  47  ''SECTION FOR DEFINITIONS             2689
    PAGE 575  ROUTINE PGM.INPUT                     6490
    PAGE 593  ROUTINE AMMO.RPT                      7202
    PAGE 706  ''PROGRAM  OLDER.VERSION              2113
SDM.MAX.RANGE
    PAGE  42  ''SECTION FOR DEFINITIONS             2412
    PAGE 209  ROUTINE PGM.MSN.ASGN                  9558
    PAGE 575  ROUTINE PGM.INPUT                     6488
    PAGE 701  ''PROGRAM  OLDER.VERSION              1842
SDM.MNV
    PAGE  42  ''SECTION FOR DEFINITIONS             2410
    PAGE 211  ROUTINE PGM.MSN.ASGN                  9677
    PAGE 575  ROUTINE PGM.INPUT                     6488
    PAGE 701  ''PROGRAM  OLDER.VERSION              1840
SDM.PGM.RNG
    PAGE  42  ''SECTION FOR DEFINITIONS             2408
    PAGE 256  ROUTINE FO.DETECTION                  1797
    PAGE 257                                        1895
    PAGE 575  ROUTINE PGM.INPUT                     6489
    PAGE 701  ''PROGRAM  OLDER.VERSION              1838
SDM.SSPK
    PAGE  42  ''SECTION FOR DEFINITIONS             2411
    PAGE 211  ROUTINE PGM.MSN.ASGN                  9677
    PAGE 269  ROUTINE BTRY.EFFECTS                  2507
    PAGE 575  ROUTINE PGM.INPUT                     6488
    PAGE 701  ''PROGRAM  OLDER.VERSION              1841
SDM.TM
    PAGE  42  ''SECTION FOR DEFINITIONS             2409
    PAGE 188  ROUTINE FINAL.COVERAGE                8566
    PAGE 575  ROUTINE PGM.INPUT                     6489
    PAGE 701  ''PROGRAM  OLDER.VERSION              1839
SDM.VOLLEY.RADIUS
    PAGE  42  ''SECTION FOR DEFINITIONS             2413
    PAGE 189  ROUTINE FINAL.COVERAGE                8632
    PAGE 575  ROUTINE PGM.INPUT                     6489
    PAGE 701  ''PROGRAM  OLDER.VERSION              1843
SD.ADS.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES       748
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES      1107
    PAGE 162  ROUTINE ATTRIT.SENSOR                 7488
    PAGE 308  ROUTINE AD.SHOOT                      4399
    PAGE 316  ROUTINE FLIGHT.PATH                   4788
    PAGE 350  EVENT AD.ENGAGEMENT                   5961
    PAGE 506  PROCESS CAS.MISSION                   3494
    PAGE 566  ROUTINE SENSOR.INPUT                  6090
```

```
    PAGE 673  ''PROGRAM  OLDER.VERSION                190
    PAGE 679                                          546
SD.AIRFIELD
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES        736
    PAGE  43  ''SECTION FOR DEFINITIONS              2483
    PAGE 149  ROUTINE PK.COMPUTE                     6913 6914
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS          3297 3327
    PAGE 286                                         3348
    PAGE 293  ROUTINE END.CAS.MISSION                3683
    PAGE 314  ROUTINE FLIGHT.PATH                    4657 4658
    PAGE 328  ROUTINE EMPTY                          5200
    PAGE 420  PROCESS AC.ATK.TGT                     8949
    PAGE 487  PROCESS ASSESSMENT                     2432
    PAGE 490                                         2604
    PAGE 504  PROCESS CAS.MISSION                    3374
    PAGE 507                                         3574
    PAGE 582  ROUTINE TACAIR.INPUT                   6764
    PAGE 585                                         6957 6959
    PAGE 588  ROUTINE AC.MUNS.INPUT                  7048 7049 7052 7062
    PAGE 621  ROUTINE TACAIR.DATA.REPORT             8348 8348
    PAGE 673  ''PROGRAM  OLDER.VERSION                179
    PAGE 702                                         1903
SD.ASC.MAX.SORTIE
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        740
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS          3450
    PAGE 582  ROUTINE TACAIR.INPUT                   6767
    PAGE 621  ROUTINE TACAIR.DATA.REPORT             8351 8351
    PAGE 673  ''PROGRAM  OLDER.VERSION                183
SD.ASC.RADIUS
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        741
    PAGE  44  ''SECTION FOR DEFINITIONS              2485
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS          3446
    PAGE 582  ROUTINE TACAIR.INPUT                   6772
    PAGE 621  ROUTINE TACAIR.DATA.REPORT             8352 8352
    PAGE 673  ''PROGRAM  OLDER.VERSION                184
    PAGE 702                                         1906
SD.CAS.BRKPT
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        745
    PAGE 421  PROCESS AC.ATK.TGT                     8983
    PAGE 426                                         9256 9289
    PAGE 582  ROUTINE TACAIR.INPUT                   6771
    PAGE 673  ''PROGRAM  OLDER.VERSION                188
SD.CMSN.QUEUE
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        749
    PAGE  35  ''SECTION FOR PROCESSES                1985
    PAGE  58  ROUTINE CREATE.FORCE                   3117
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS          3296
    PAGE 288                                         3457 3458 3462 3463
    PAGE 327  ROUTINE DQ.CMSN.QUEUE                  5158
    PAGE 505  PROCESS CAS.MISSION                    3409 3410 3465
    PAGE 506                                         3466 3485 3486
    PAGE 508                                         3619 3620
    PAGE 509                                         3653 3654
    PAGE 618  ROUTINE SNAP.R                         8251
    PAGE 673  ''PROGRAM  OLDER.VERSION                191
    PAGE 694                                         1423
```

```
SD.FPO.LIST
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          750
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES         1439
    PAGE  65  ROUTINE FILE.KAD.SENSOR                  3389 3391 3402 3409
    PAGE 314  ROUTINE FLIGHT.PATH                      4683
    PAGE 673  ''PROGRAM  OLDER.VERSION                  192
    PAGE 685                                            878
SD.KAS.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          751
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES         1520
    PAGE  65  ROUTINE FILE.KAD.SENSOR                  3367 3378
    PAGE 317  ROUTINE FLIGHT.PATH                      4822
    PAGE 673  ''PROGRAM  OLDER.VERSION                  193
    PAGE 686                                            959
SD.MAX.SORTIE.TP
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES          738
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS            3430
    PAGE 582  ROUTINE TACAIR.INPUT                     6765
    PAGE 621  ROUTINE TACAIR.DATA.REPORT               8349 8349
    PAGE 673  ''PROGRAM  OLDER.VERSION                  181
SD.NO.FLY.VIS
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          743
    PAGE  44  ''SECTION FOR DEFINITIONS                2486
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS            3417
    PAGE 582  ROUTINE TACAIR.INPUT                     6773
    PAGE 621  ROUTINE TACAIR.DATA.REPORT               8353 8353
    PAGE 673  ''PROGRAM  OLDER.VERSION                  186
    PAGE 702                                           1907
SD.NR.CAS.MISSIONS
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES          737
    PAGE  44  ''SECTION FOR DEFINITIONS                2484
    PAGE 284  ROUTINE CAS.EVAL                         3260 3261
    PAGE 380  EVENT INIT.PREPLAN.CAS                   7244 7245 7254
    PAGE 673  ''PROGRAM  OLDER.VERSION                  180
    PAGE 702                                           1904
SD.OLD.SORTIE.QUEUE
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          752
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES         1615
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE                6647
    PAGE 504  PROCESS CAS.MISSION                      3388
    PAGE 618  ROUTINE SNAP.R                           8253
    PAGE 673  ''PROGRAM  OLDER.VERSION                  194
    PAGE 688                                           1054
SD.PAX.KV.ID
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          746
SD.POOR.FLY.VIS
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          744
    PAGE  44  ''SECTION FOR DEFINITIONS                2487
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                  2838
    PAGE 281  ROUTINE AC.DF.EFFECTS                    3104
    PAGE 350  EVENT AD.ENGAGEMENT                      6002
    PAGE 582  ROUTINE TACAIR.INPUT                     6774
    PAGE 621  ROUTINE TACAIR.DATA.REPORT               8354 8354
    PAGE 673  ''PROGRAM  OLDER.VERSION                  187
    PAGE 702                                           1908
```

```
SD.SORTIES.THIS.TP
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          742
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS            3429
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE                6648 6654
    PAGE 504  PROCESS CAS.MISSION                      3370 3394
    PAGE 673  ''PROGRAM  OLDER.VERSION                  185
SD.TP.SORTIE
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES          739
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE                6641
    PAGE 504  PROCESS CAS.MISSION                      3393
    PAGE 582  ROUTINE TACAIR.INPUT                     6766
    PAGE 621  ROUTINE TACAIR.DATA.REPORT               8350 8350
    PAGE 673  ''PROGRAM  OLDER.VERSION                  182
    PAGE 702                                           1905
SEARCH.DIRECTION
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA               4778 4793 4794 4795 4811 4820
SEARCH.DIST
    PAGE 428  PROCESS AIR.OBSERVER                     9345
    PAGE 431                                           9511 9514
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE             1368
    PAGE 468                                           1445 1447
    PAGE 653  PROCESS AIRBORNE.RADAR                   9273
    PAGE 654                                           9349 9352
SEARCH.PERIOD
    PAGE 438  PROCESS FORWARD.OBSERVER                 9868
    PAGE 439                                           9928 9930 9946
    PAGE 441                                             70   72
SEARCH.POINT
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES          713
    PAGE  56  ROUTINE MAIN1                            3051
    PAGE 618  ROUTINE SNAP.R                           8240
    PAGE 672  ''PROGRAM  OLDER.VERSION                  156
    PAGE 712                                           2464
SEARCH.RANGE
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA               4803 4811 4820
SEARCH.SECTOR
    PAGE 138  ROUTINE CHECK.PROX                       6389 6396 6400 6405 6410 6416 6422
    PAGE 139                                           6461
SEARCH.TIME
    PAGE 428  PROCESS AIR.OBSERVER                     9349
    PAGE 431                                           9514 9515 9536
SEARCH.X
    PAGE 438  PROCESS FORWARD.OBSERVER                 9910
SEARCH.X.GRID
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA               4774
    PAGE  99                                           4828
    PAGE 438  PROCESS FORWARD.OBSERVER                 9869 9897 9905
    PAGE 439                                           9917
SEARCH.Y
    PAGE 438  PROCESS FORWARD.OBSERVER                 9910
SEARCH.Y.GRID
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA               4775
    PAGE  99                                           4829
    PAGE 438  PROCESS FORWARD.OBSERVER                 9870 9898 9906
    PAGE 439                                           9918
```

```
SECTOR
    PAGE   1  ROUTINE FOR CROSS_REFERENCING          47
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES      717  718
    PAGE  15                                        804
    PAGE  61  ROUTINE FEBA.INITIAL                 3187 3191 3196 3198 3199 3200 3201 3203 3204 3225 3226 3228 3229 3232 3233
                                                   3240
    PAGE  62                                       3244 3246 3247
    PAGE 100  ROUTINE LOCATE.SECTOR                4833 4840 4842 4843 4845 4847 4852 4853 4857 4857 4866 4867
    PAGE 119  ROUTINE RESET.FEBA.SECTOR            5654 5685 5691 5696 5697
    PAGE 239  ROUTINE TARGET.ANALYSIS               985 1004 1009 1011 1012 1014 1019 1023
    PAGE 246  ROUTINE VOLLEY                       1315 1321 1327 1328 1329 1330 1332 1337 1339
    PAGE 261  ROUTINE BTRY.EFFECTS                 2024 2025
    PAGE 325  ROUTINE DEQ.FEBA.SET                 5080 5091 5094 5094 5095 5099 5100 5102 5106 5107 5112 5114
    PAGE 331  ROUTINE ENQ.FEBA.SET                 5302 5309 5312 5321 5323 5325 5327 5329 5330 5335
    PAGE 350  EVENT AD.ENGAGEMENT                  5993
    PAGE 360  EVENT CFR.ON                         6412 6433 6436 6438 6439 6441 6447 6450 6455
    PAGE 371  EVENT FEBA.SORTIE                    6776 6798 6806 6807 6813 6815 6818 6826
    PAGE 372                                       6841 6854 6856 6859 6867 6872
    PAGE 564  ROUTINE MAO.INPUT                    5957
    PAGE 567  ROUTINE SENSOR.INPUT                 6183
    PAGE 618  ROUTINE SNAP.R                       8241
    PAGE 634  FUNCTION FEBA.BAND                   8783 8785 8796 8797 8802 8805 8814 8815 8820 8823
    PAGE 660  FUNCTION STAY.TIME                   9596 9604 9606
    PAGE 672  ''PROGRAM  OLDER.VERSION              160  161
    PAGE 674                                       246
    PAGE 712                                      2465
SECTORS.
    PAGE 138  ROUTINE CHECK.PROX                   6379
SECTOR.
    PAGE  87  ROUTINE END.MOVE                     4315
SECTOR.LEADER
    PAGE  89  ROUTINE FA.BN.MOVEMENT               4451
    PAGE  90                                       4453 4455
    PAGE  91                                       4531 4533 4535
SECTOR.WIDTH
    PAGE  61  ROUTINE FEBA.INITIAL                 3193 3199
    PAGE 371  EVENT FEBA.SORTIE                    6819 6824
    PAGE 372                                       6860 6865
SECT.
    PAGE  87  ROUTINE END.MOVE                     4338 4353
    PAGE 404  EVENT START.MOVE                     8245 8258
    PAGE 405                                       8330
SEED.V
    PAGE 438  PROCESS FORWARD.OBSERVER             9904 9909
    PAGE 440                                       9989 9990   13   18
    PAGE 441                                         32   33   36   37   78   79
    PAGE 599  ROUTINE BETWEEN.ROUTINE              7437 7438 7441 7442
SEGMENT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING          49
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES     1737
    PAGE  84  ROUTINE CHANGE.LOC                   4210 4211
    PAGE  85                                       4274 4275
    PAGE  97  ROUTINE LINE.OF.SIGHT                4728 4729 4730 4740 4741 4742 4744 4747 4748 4749 4759 4760 4761 4763
    PAGE 106  ROUTINE MIN.MOVE                     5072 5074
    PAGE 107  ROUTINE NEW.SEGMENT                  5105 5116 5120 5122 5130 5139 5142 5148 5157
    PAGE 108                                       5164 5174 5175 5194 5200 5204 5208
```

```
    PAGE 109                                    5229 5244 5250 5252 5258 5259 5259 5260
    PAGE 122  ROUTINE SEGMENT.ADJUST            5781 5783 5784 5788
    PAGE 135  ROUTINE CHECK.FOR.MINES           6272
    PAGE 236  ROUTINE SMOKE.EFFECTS              887
    PAGE 317  ROUTINE FLIGHT.PATH               4832 4861
    PAGE 329  ROUTINE EMPTY                     5277 5278
    PAGE 349  EVENT AD.ENGAGEMENT               5942
    PAGE 479  PROCESS WITH.DRAW                 1980 1981 1982 1985 1986 1989 1991 1995 1999 2008
    PAGE 619  ROUTINE SNAP.R                    8268
    PAGE 690  ''PROGRAM  OLDER.VERSION          1176
    PAGE 713                                    2500
SEGMENT.
    PAGE 317  ROUTINE FLIGHT.PATH               4814
    PAGE 353  EVENT AD.ENGAGEMENT               6175
SEG.LENGTH
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES  1738
    PAGE  46  ''SECTION FOR DEFINITIONS         2633
    PAGE  80  ROUTINE BLOCK.LOS                 4034 4034 4037 4049 4049 4052
    PAGE  97  ROUTINE LINE.OF.SIGHT             4740 4741 4742 4759 4760 4761
    PAGE 106  ROUTINE MIN.MOVE                  5074
    PAGE 107  ROUTINE NEW.SEGMENT               5139
    PAGE 109                                    5258 5259 5260
    PAGE 122  ROUTINE SEGMENT.ADJUST            5783 5784
    PAGE 479  PROCESS WITH.DRAW                 1985 2008
    PAGE 690  ''PROGRAM  OLDER.VERSION          1177
    PAGE 705                                    2056
SEG.TYPE
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES  1740
    PAGE  46  ''SECTION FOR DEFINITIONS         2635
    PAGE  80  ROUTINE BLOCK.LOS                 4033 4036 4048 4051
    PAGE  97  ROUTINE LINE.OF.SIGHT             4729 4748
    PAGE 107  ROUTINE NEW.SEGMENT               5120 5142
    PAGE 109                                    5244 5259
    PAGE 479  PROCESS WITH.DRAW                 1981 1999
    PAGE 690  ''PROGRAM  OLDER.VERSION          1179
    PAGE 705                                    2058
SEG.UNIT
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES  1739
    PAGE  46  ''SECTION FOR DEFINITIONS         2634
    PAGE  80  ROUTINE BLOCK.LOS                 4018 4042
    PAGE  97  ROUTINE LINE.OF.SIGHT             4730 4749
    PAGE 107  ROUTINE NEW.SEGMENT               5116 5122 5130 5148 5157
    PAGE 108                                    5164 5174 5175 5194 5200 5204 5208
    PAGE 109                                    5229 5250 5252
    PAGE 479  PROCESS WITH.DRAW                 1982 1991 1995
    PAGE 690  ''PROGRAM  OLDER.VERSION          1178
    PAGE 705                                    2057
SELECTED.WPN
    PAGE 149  ROUTINE PK.COMPUTE                6902 6942
    PAGE 151                                    7056 7059
SEND.TEAM
    PAGE  40  ''SECTION FOR EVENTS              2303
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS        3562
    PAGE 291                                    3605
    PAGE 292                                    3660
    PAGE 337  ROUTINE HC.EMPTY                  5485 5486 5490 5491 5492 5494 5497 5498
```

```
    PAGE 391  EVENT SEND.TEAM                          7596 7612
    PAGE 453  PROCESS HC.RETURN.FARRP                   664  678  684
    PAGE 602  ROUTINE BETWEEN.ROUTINE                  7628 7629
    PAGE 616  ROUTINE SNAP2                            8188
    PAGE 699  ''PROGRAM  OLDER.VERSION                 1740
SENSOR.
    PAGE 315  ROUTINE FLIGHT.PATH                      4723
    PAGE 317                                           4832 4861
    PAGE 349  EVENT AD.ENGAGEMENT                      5942
    PAGE 350                                           5960
SENSOR.FDC
    PAGE 565  ROUTINE SENSOR.INPUT                     6055
    PAGE 566                                           6070 6079 6107
SENSOR.INPUT
    PAGE 521  ROUTINE MAIN2                            4263 4265
    PAGE 564  ROUTINE MAO.INPUT                        5958
    PAGE 565  ROUTINE SENSOR.INPUT                     6011
SENSOR.INTERSECT
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES         1722
    PAGE 293  ROUTINE END.CAS.MISSION                  3694
    PAGE 317  ROUTINE FLIGHT.PATH                      4818 4844
    PAGE 318                                           4879 4907
    PAGE 319                                           4929
    PAGE 349  EVENT AD.ENGAGEMENT                      5912
    PAGE 505  PROCESS CAS.MISSION                      3427
    PAGE 689  ''PROGRAM  OLDER.VERSION                 1161
SENSOR.MODEL
    PAGE 565  ROUTINE SENSOR.INPUT                     6046
    PAGE 566                                           6077 6084 6092 6097 6119
    PAGE 567                                           6129
SENSOR.TYPE
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES          720
    PAGE 154  ROUTINE AO.DETECTION                     7080 7100
    PAGE 157                                           7268
    PAGE 212  ROUTINE PIR.DETECTION                    9708 9720
    PAGE 229  ROUTINE RPV.DETECTION                     520  533
    PAGE 230                                            591  602
    PAGE 254  ROUTINE FO.DETECTION                     1683 1699
    PAGE 256                                           1839
    PAGE 428  PROCESS AIR.OBSERVER                     9342 9363
    PAGE 430                                           9486
    PAGE 431                                           9556
    PAGE 432                                           9562 9563
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE             1365 1380
    PAGE 468                                           1416 1453 1454
    PAGE 561  ROUTINE ST.INPUT                         5863 5865 5867 5868 5869 5870 5871 5873 5873
    PAGE 566  ROUTINE SENSOR.INPUT                     6080 6082 6087 6096 6117 6118
    PAGE 567                                           6127 6128
    PAGE 618  ROUTINE SNAP.R                           8242
    PAGE 653  PROCESS AIRBORNE.RADAR                   9272 9278
    PAGE 654                                           9355 9364 9365
    PAGE 656  ROUTINE AR.DETECTION                     9385 9389 9399 9425 9435
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                  9481 9494
    PAGE 659                                           9566 9567 9572 9583
    PAGE 672  ''PROGRAM  OLDER.VERSION                  163
    PAGE 712                                           2466
```

```
SENSOR.UNIT
    PAGE 565  ROUTINE SENSOR.INPUT                  6037 6039 6041
    PAGE 566                                        6075 6078 6090 6091 6099
SENS.ID
    PAGE 206  ROUTINE PDB.DETECTION                 9466 9468 9472 9477 9482
    PAGE 359  EVENT CFR.OFF                         6382 6397 6401
    PAGE 360  EVENT CFR.ON                          6407 6424
    PAGE 361                                        6468 6469 6470 6473
SENS.LINK
    PAGE 206  ROUTINE PDB.DETECTION                 9468 9469 9471 9482
    PAGE 359  EVENT CFR.OFF                         6383 6389 6393 6398 6400
    PAGE 360  EVENT CFR.ON                          6408 6417 6421 6425 6427
    PAGE 361                                        6472
SENS.TYPE
    PAGE 565  ROUTINE SENSOR.INPUT                  6045 6047 6053
    PAGE 566                                        6076 6080
SENS.UN
    PAGE 567  ROUTINE SENSOR.INPUT                  6149 6151 6153
SENS.UNIT.NO.
    PAGE 565  ROUTINE SENSOR.INPUT                  6023 6024 6030 6034 6037
SENS.UN.NO.
    PAGE 567  ROUTINE SENSOR.INPUT                  6140 6142 6146 6149
SEN.UNIT
    PAGE 441  PROCESS FORWARD.OBSERVER               59
SEQ.
    PAGE 539  ROUTINE READ.ORDERS                   5025
    PAGE 540                                        5042 5053 5058 5097
SETS.
    PAGE 547  ROUTINE P.E.M.INPUT                   5259
SET.BACK
    PAGE 360  EVENT CFR.ON                          6413 6453 6454 6456 6458
SET.DEBUG
    PAGE  41  ''SECTION FOR EVENTS                  2342 2343
    PAGE 392  EVENT SET.DEBUG                       7623 7630
    PAGE 602  ROUTINE BETWEEN.ROUTINE               7656 7657
    PAGE 617  ROUTINE SNAP2                         8197
    PAGE 700  ''PROGRAM  OLDER.VERSION              1776 1777
SE.BNDRY.INT
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES       718
    PAGE  61  ROUTINE FEBA.INITIAL                  3198 3201
    PAGE 100  ROUTINE LOCATE.SECTOR                 4843 4845
    PAGE 371  EVENT FEBA.SORTIE                     6818
    PAGE 372                                        6859
    PAGE 672  ''PROGRAM  OLDER.VERSION               161
SHADE
    PAGE   1  ROUTINE FOR CROSS_REFERENCING          55
    PAGE  50  ''SECTION FOR DEFINITIONS             2834
    PAGE  56  ROUTINE MAIN1                         3063 3064 3065
    PAGE 133  ROUTINE CHECK.FORCE                   6182
    PAGE 286  ROUTINE CHECK.CAS.CONSTRAINTS         3393
    PAGE 288                                        3465
    PAGE 294  ROUTINE END.CAS.MISSION               3769
    PAGE 398  EVENT START.BATTLE                    7959
    PAGE 607  ROUTINE KV.PRINT                      7795 7795
    PAGE 621  ROUTINE TACAIR.DATA.REPORT            8381
    PAGE 708  ''PROGRAM  OLDER.VERSION              2255
```

```
SHOOT.HC
    PAGE 379  EVENT HELO.ENGAGEMENT                    7213 7214 7215 7216
SHOOT.OUT
    PAGE   2  PROGRAM REVISIONS                         84
    PAGE   3                                           127  133
    PAGE  37  ''SECTION FOR PROCESSES                 2130
    PAGE 107  ROUTINE NEW.SEGMENT                     5111
    PAGE 108                                          5215 5216
    PAGE 109                                          5218 5232 5233 5235
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                 2939 2940
    PAGE 282  ROUTINE AC.DF.EFFECTS                   3123 3124
    PAGE 294  ROUTINE END.CAS.MISSION                 3741 3744
    PAGE 301  ROUTINE HC.DISENGAGE                    4110 4111 4112 4114
    PAGE 329  ROUTINE EMPTY                           5260 5261 5262 5263 5264
    PAGE 369  EVENT ENGAGEMENT                        6733
    PAGE 379  EVENT HELO.ENGAGEMENT                   7213
    PAGE 383  EVENT OFF.LINE.ATTRITION                7381 7383
    PAGE 398  EVENT START.BATTLE                      7970
    PAGE 419  PROCESS AC.ATK.TGT                      8891
    PAGE 423                                          9090 9107 9108 9122
    PAGE 425                                          9198 9214 9215 9229
    PAGE 426                                          9273
    PAGE 435  PROCESS ARTY.ASSESS                     9719 9720 9721 9722 9731 9738 9740
    PAGE 436                                          9756 9757 9758 9779 9786 9799
    PAGE 437                                          9804 9805 9806
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION          1092 1093 1094 1096
    PAGE 464  PROCESS MINE.ASSESS                     1262
    PAGE 465                                          1299 1302
    PAGE 487  PROCESS ASSESSMENT                      2447
    PAGE 488                                          2513 2514 2517 2519
    PAGE 491                                          2690 2695 2696 2697 2698 2699
    PAGE 493  PROCESS SHOOT.OUT                       2749 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766
                                                      2767 2768 2769 2773 2775 2776 2777 2790 2791 2795 2796 2802 2803
    PAGE 494                                          2809 2825 2832 2851 2855 2862
    PAGE 495                                          2866 2878 2880 2889 2890 2894 2922
    PAGE 496                                          2923 2925 2957 2959 2962 2963 2969 2971
    PAGE 497                                          2981 2983 2998 3000 3010 3011 3026 3030 3036 3037
    PAGE 498                                          3040 3069 3071 3078 3081 3082 3085
    PAGE 499                                          3111 3113 3114 3125 3141 3149
    PAGE 500                                          3157 3160 3167 3168 3174 3182 3190 3201 3212
    PAGE 501                                          3213 3215 3223 3235 3251
    PAGE 502                                          3279 3281 3289 3296 3304 3312 3315 3322 3323
    PAGE 503                                          3329 3331 3344 3346
    PAGE 507  PROCESS CAS.MISSION                     3581
    PAGE 508                                          3582 3583 3584
    PAGE 516  PROCESS HELICOPTER.FIRE                 4051 4052
    PAGE 600  ROUTINE BETWEEN.ROUTINE                 7512 7513
    PAGE 616  ROUTINE SNAP2                           8158
    PAGE 619  ROUTINE SNAP.R                          8273
    PAGE 696  ''PROGRAM  OLDER.VERSION                1568
SHOOT.OUTS
    PAGE 300  ROUTINE HC.DISENGAGE                    4011
    PAGE 301                                          4083
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION          1075 1078
SHOOT.OUTS.
    PAGE 507  PROCESS CAS.MISSION                     3553
```

SHOOT.OUT.
PAGE 376 EVENT HELO.ENGAGEMENT 7042
PAGE 510 PROCESS HELICOPTER.FIRE 3665
SHOOT.OUT.NO
PAGE 487 PROCESS ASSESSMENT 2417 2424 2425 2431 2460 2461 2465 2466
PAGE 488 2470 2481 2482 2483 2490 2494 2499 2508 2515
PAGE 489 2550 2555 2579 2580 2584
PAGE 490 2596 2606 2613 2617 2618 2622
PAGE 491 2658 2671 2672 2676
PAGE 492 2705 2715 2724 2725 2729
SHOT.
PAGE 354 EVENT AD.ENGAGEMENT 6188
SIDE
PAGE 1 ROUTINE FOR CROSS_REFERENCING 47
PAGE 13 ''SECTION FOR PERMANENT_ENTITIES 727
PAGE 14 771 778 784 787 791
PAGE 15 800 804 813
PAGE 56 ROUTINE MAIN1 3054
PAGE 61 ROUTINE FEBA.INITIAL 3187 3207 3209 3228 3229 3231 3232 3233 3238
PAGE 62 3242 3246 3247
PAGE 75 ROUTINE UNIT.ASSIGNMENT 3825 3843 3848
PAGE 98 ROUTINE LOCATE.SEARCH.AREA 4781 4785 4786
PAGE 119 ROUTINE RESET.FEBA.SECTOR 5649 5654 5657 5658 5661 5663 5665 5674 5675 5692 5693 5696 5697
PAGE 130 ROUTINE CHECK.DEAD 6081 6090
PAGE 133 ROUTINE CHECK.FORCE 6185
PAGE 159 ROUTINE ATTRIT.SENSOR 7318 7323 7358 7361 7363
PAGE 160 7369 7371 7373
PAGE 161 7435 7441 7477
PAGE 187 ROUTINE FD.EFFECTS.REQ 8524 8534 8536
PAGE 203 ROUTINE MARGINAL.EFFECTS.ADJ 9310 9325
PAGE 208 ROUTINE PGM.MSN.ASGN 9533
PAGE 209 9553
PAGE 212 ROUTINE PIR.DETECTION 9708 9721
PAGE 229 ROUTINE RPV.DETECTION 520 534
PAGE 239 ROUTINE TARGET.ANALYSIS 985 1005 1009 1018 1019 1023
PAGE 240 1042
PAGE 248 ROUTINE WEIGHTED.VOLLEYS 1387 1423
PAGE 249 1426 1433
PAGE 254 ROUTINE FO.DETECTION 1683 1700
PAGE 260 ROUTINE BTRY.EFFECTS 1985
PAGE 262 2100 2103
PAGE 270 2572 2575 2576 2578
PAGE 271 2647 2648 2658 2659
PAGE 272 2672 2673 2687 2688 2692
PAGE 286 ROUTINE CHECK.CAS.CONSTRAINTS 3395
PAGE 288 3467
PAGE 295 ROUTINE END.CAS.MISSION 3803
PAGE 297 ROUTINE FARRP.CHECK 3859 3865 3871
PAGE 311 ROUTINE INTER.HELO 4534 4536
PAGE 312 4626
PAGE 324 ROUTINE DECIDE 5056 5057
PAGE 325 ROUTINE DEQ.FEBA.SET 5084 5090 5091 5102 5106
PAGE 327 ROUTINE DQ.CMSN.QUEUE 5157 5158
PAGE 331 ROUTINE ENQ.FEBA.SET 5306 5311 5316 5321 5323 5325 5325 5327 5327 5329 5330 5337 5338
PAGE 337 ROUTINE HC.EMPTY 5464 5471
PAGE 356 EVENT BTL.ENDED 6273 6279 6280 6285 6289 6291 6292 6304

```
    PAGE 357                                    6312 6331
    PAGE 360  EVENT CFR.ON                      6412 6428 6436 6445 6447 6450
    PAGE 371  EVENT FEBA.SORTIE                 6777 6790 6812 6813 6814 6815 6826
    PAGE 372                                    6833 6853 6854 6855 6856 6868 6872
    PAGE 380  EVENT INIT.PREPLAN.CAS            7257
    PAGE 398  EVENT START.BATTLE                7961
    PAGE 401                                    8093 8097 8102 8103 8117 8121 8126 8127 8141 8143
    PAGE 428  PROCESS AIR.OBSERVER              9342 9361 9374
    PAGE 432                                    9590
    PAGE 448  PROCESS HC.ARRIVE.BATTLE           403
    PAGE 450  PROCESS HC.RETURN.FARRP            506  514
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE      1365 1377 1383
    PAGE 474  PROCESS TARGET.REPORT             1763 1767
    PAGE 476                                    1878
    PAGE 480  PROCESS FIRE.MISSION              2057 2078 2079 2082 2083
    PAGE 481                                    2085 2088 2089 2093 2093 2094 2095 2096 2098 2099
    PAGE 484                                    2291
    PAGE 510  PROCESS HELICOPTER.FIRE           3697
    PAGE 523  ROUTINE SYS.INPUT                 4341 4343 4344 4345 4346 4347 4348
    PAGE 529  ROUTINE KV.INPUT                  4527 4562 4564 4567
    PAGE 530  ROUTINE EQ.TE.INPUT               4620 4621 4622 4626
    PAGE 534  ROUTINE UNIT.INPUT                4751
    PAGE 536                                    4888 4890 4896 4899
    PAGE 567  ROUTINE SENSOR.INPUT              6161 6162 6176
    PAGE 568                                    6188
    PAGE 573  ROUTINE FARRP.INPUT               6414
    PAGE 576  ROUTINE ILLUM.INPUT               6520 6527
    PAGE 577  ROUTINE MINE.INPUT                6576 6579 6581 6583 6584 6585
    PAGE 578                                    6591 6593
    PAGE 579                                    6651
    PAGE 580  ROUTINE SMOKE.INPUT               6693 6700 6701 6703 6704
    PAGE 582  ROUTINE TACAIR.INPUT              6752
    PAGE 588  ROUTINE AC.MUNS.INPUT             7051 7052 7061 7062
    PAGE 592  ROUTINE AMMO.RPT                  7146 7148
    PAGE 593                                    7182 7199 7216
    PAGE 606  ROUTINE KV.PRINT                  7691 7692
    PAGE 609  ROUTINE KV.SCOREBOARD             7846 7860 7862 7865 7866 7867 7868 7869 7871 7872 7873 7874 7875
    PAGE 618  ROUTINE SNAP.R                    8243 8250 8251 8252 8253
    PAGE 621  ROUTINE TACAIR.DATA.REPORT        8345 8373
    PAGE 634  FUNCTION FEBA.BAND                8783 8785 8786 8797 8801 8802 8805 8815 8819 8820 8823
    PAGE 653  PROCESS AIRBORNE.RADAR            9272 9276 9283
    PAGE 654                                    9326
    PAGE 656  ROUTINE AR.DETECTION              9389 9400
    PAGE 658  PROCESS PHOTO.IR.FLIGHT           9481 9492 9496
    PAGE 660  FUNCTION STAY.TIME                9596 9599 9606
    PAGE 672  ''PROGRAM  OLDER.VERSION           170
    PAGE 673                                     213  220  226  229  233
    PAGE 674                                     242  246  255
    PAGE 711                                    2438
    PAGE 712                                    2467
SIDE.
    PAGE 283  ROUTINE CAS.EVAL                  3181
SIDE.ALF
    PAGE 611  ROUTINE OUTPUT.ATTRITION          7932 7934 7935 7985
SIDE.CFR.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES   755
```

```
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES          1229
    PAGE  49  ''SECTION FOR DEFINITIONS                 2822
    PAGE 161  ROUTINE ATTRIT.SENSOR                     7477
    PAGE 358  EVENT CFR.ACTIVATION                      6362
    PAGE 566  ROUTINE SENSOR.INPUT                      6124
    PAGE 673  ''PROGRAM  OLDER.VERSION                   197
    PAGE 681                                             668
    PAGE 708                                            2243
SIDE.PDB.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES           754
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES          1638
    PAGE  49  ''SECTION FOR DEFINITIONS                 2821
    PAGE 161  ROUTINE ATTRIT.SENSOR                     7435 7441
    PAGE 386  EVENT PDB.ACTIVATION                      7489
    PAGE 567  ROUTINE SENSOR.INPUT                      6133 6162
    PAGE 673  ''PROGRAM  OLDER.VERSION                   196
    PAGE 688                                            1077
    PAGE 708                                            2242
SIDE.TU.TOTAL
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES           731
    PAGE 231  ROUTINE SIZE.ESTIMATE                      674
    PAGE 533  ROUTINE UNIT.INPUT                        4730
    PAGE 672  ''PROGRAM  OLDER.VERSION                   174
SIGN.F
    PAGE 565  ROUTINE SENSOR.INPUT                      6016
    PAGE 566                                            6110
SIN.F
    PAGE  73  ROUTINE ORIENTATION                       3795
    PAGE  74                                            3810 3820
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                   3866
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA                4809 4820
    PAGE 111  ROUTINE PRED.POS                          5342
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                   2862 2865 2870 2873
    PAGE 447  PROCESS HC.ARRIVE.BATTLE                   373
    PAGE 448                                             393
SIZE.ESTIMATE
    PAGE 231  ROUTINE SIZE.ESTIMATE                      623
    PAGE 240  ROUTINE TARGET.ANALYSIS                   1063
    PAGE 631  FUNCTION COMBINATIONS                     8718
SIZE.MEASURE
    PAGE 231  ROUTINE SIZE.ESTIMATE                      626  631  634  649  671  677  678  679
    PAGE 232                                             684
    PAGE 239  ROUTINE TARGET.ANALYSIS                    986
    PAGE 240                                            1068 1069 1072
SI.AD.UNIT
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1728
    PAGE 317  ROUTINE FLIGHT.PATH                       4830
    PAGE 349  EVENT AD.ENGAGEMENT                       5917
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1167
SI.ALTITUDE
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1729
    PAGE 317  ROUTINE FLIGHT.PATH                       4826 4828
    PAGE 350  EVENT AD.ENGAGEMENT                       5987
    PAGE 351                                            6068
    PAGE 352                                            6084
    PAGE 354                                            6214
```

```
    PAGE 690  ''PROGRAM  OLDER.VERSION                1168
SI.LIST
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES        1200
    PAGE  31                                          1731
    PAGE 293  ROUTINE END.CAS.MISSION                 3691 3693
    PAGE 317  ROUTINE FLIGHT.PATH                     4819 4843
    PAGE 318                                          4878 4906
    PAGE 505  PROCESS CAS.MISSION                     3427 3437
    PAGE 508                                          3629
    PAGE 680  ''PROGRAM  OLDER.VERSION                 639
    PAGE 690                                          1170
SI.TIME.TIL.INTERSECT
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES        1727
    PAGE  48  ''SECTION FOR DEFINITIONS               2721
    PAGE 317  ROUTINE FLIGHT.PATH                     4856
    PAGE 318                                          4869 4890 4899
    PAGE 319                                          4922
    PAGE 505  PROCESS CAS.MISSION                     3444
    PAGE 508                                          3636
    PAGE 690  ''PROGRAM  OLDER.VERSION                1166
    PAGE 706                                          2143
SI.X.ENTRY
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1723
    PAGE  46  ''SECTION FOR DEFINITIONS               2650
    PAGE 317  ROUTINE FLIGHT.PATH                     4834 4839 4857
    PAGE 318                                          4865 4874 4895 4902
    PAGE 349  EVENT AD.ENGAGEMENT                     5923 5924 5927 5930 5935 5945 5949
    PAGE 351                                          6036 6064
    PAGE 352                                          6087
    PAGE 353                                          6148 6149 6160
    PAGE 354                                          6210
    PAGE 689  ''PROGRAM  OLDER.VERSION                1162
    PAGE 705                                          2074
SI.X.EXIT
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1725
    PAGE  46  ''SECTION FOR DEFINITIONS               2652
    PAGE 317  ROUTINE FLIGHT.PATH                     4836 4853
    PAGE 318                                          4867 4888 4897 4912 4915 4919
    PAGE 349  EVENT AD.ENGAGEMENT                     5923 5924 5927 5946 5952
    PAGE 351                                          6036
    PAGE 690  ''PROGRAM  OLDER.VERSION                1164
    PAGE 705                                          2076
SI.Y.ENTRY
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1724
    PAGE  46  ''SECTION FOR DEFINITIONS               2651
    PAGE 317  ROUTINE FLIGHT.PATH                     4835 4840 4858
    PAGE 318                                          4866 4875 4896 4903
    PAGE 349  EVENT AD.ENGAGEMENT                     5925 5926 5928 5934 5947 5950
    PAGE 351                                          6037 6066
    PAGE 353                                          6150 6151 6161
    PAGE 354                                          6212
    PAGE 690  ''PROGRAM  OLDER.VERSION                1163
    PAGE 705                                          2075
SI.Y.EXIT
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1726
    PAGE  46  ''SECTION FOR DEFINITIONS               2653
```

```
    PAGE 317  ROUTINE FLIGHT.PATH                    4837 4854
    PAGE 318                                         4868 4889 4898 4913 4916
    PAGE 319                                         4920
    PAGE 349  EVENT AD.ENGAGEMENT                    5925 5926 5928 5948 5953
    PAGE 351                                         6037
    PAGE 690  ''PROGRAM  OLDER.VERSION               1165
    PAGE 705                                         2077
SLANT.RANGE
    PAGE 154  ROUTINE AO.DETECTION                   7089 7127 7128
SMK.BURN.TIME
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        820
    PAGE  45  ''SECTION FOR DEFINITIONS              2580
    PAGE 234  ROUTINE SMOKE.EFFECTS                   775  784
    PAGE 235                                          811
    PAGE 580  ROUTINE SMOKE.INPUT                    6687
    PAGE 674  ''PROGRAM  OLDER.VERSION                262
    PAGE 704                                         2001
SMK.DEBUG
    PAGE  43  ''SECTION FOR DEFINITIONS              2426
    PAGE 225  ROUTINE REQUEST.SMOKE                   371  379
    PAGE 233  ROUTINE SMOKE.COMPUTATION               728
    PAGE 234  ROUTINE SMOKE.EFFECTS                   751  768
    PAGE 235                                          849
    PAGE 236                                          877
    PAGE 580  ROUTINE SMOKE.INPUT                    6678
    PAGE 701  ''PROGRAM  OLDER.VERSION               1856
SMK.ID
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        817
    PAGE  47  ''SECTION FOR DEFINITIONS              2688
    PAGE 580  ROUTINE SMOKE.INPUT                    6684
    PAGE 592  ROUTINE AMMO.RPT                       7168
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                7394
    PAGE 674  ''PROGRAM  OLDER.VERSION                259
    PAGE 706                                         2112
SMK.MAX.RANGE
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        819
    PAGE  45  ''SECTION FOR DEFINITIONS              2579
    PAGE 233  ROUTINE SMOKE.COMPUTATION               709
    PAGE 580  ROUTINE SMOKE.INPUT                    6689
    PAGE 674  ''PROGRAM  OLDER.VERSION                261
    PAGE 704                                         2000
SMK.RND.WT
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        821
    PAGE 580  ROUTINE SMOKE.INPUT                    6688
    PAGE 592  ROUTINE AMMO.RPT                       7169
    PAGE 674  ''PROGRAM  OLDER.VERSION                263
SMK.SWITCH
    PAGE  43  ''SECTION FOR DEFINITIONS              2427
    PAGE 222  ROUTINE REQUEST.SMOKE                   184
    PAGE 580  ROUTINE SMOKE.INPUT                    6673 6674
    PAGE 701  ''PROGRAM  OLDER.VERSION               1857
SMK.USE.RULE
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        801
    PAGE 234  ROUTINE SMOKE.EFFECTS                   773  782
    PAGE 370  EVENT ENGAGEMENT                       6769
    PAGE 580  ROUTINE SMOKE.INPUT                    6700 6701
```

```
    PAGE 674  ''PROGRAM  OLDER.VERSION                243
SMK.WD.RULE
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        802
    PAGE 478  PROCESS WITH.DRAW                      1947
    PAGE 580  ROUTINE SMOKE.INPUT                    6703 6704
    PAGE 674  ''PROGRAM  OLDER.VERSION                244
SMK.WIDTH
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        818
    PAGE  45  ''SECTION FOR DEFINITIONS              2578
    PAGE 233  ROUTINE SMOKE.COMPUTATION               715
    PAGE 580  ROUTINE SMOKE.INPUT                    6685
    PAGE 674  ''PROGRAM  OLDER.VERSION                260
    PAGE 704                                         1999
SMOKE.COMPUT
    PAGE 233  ROUTINE SMOKE.COMPUTATION               732
SMOKE.COMPUTE
    PAGE 634  FUNCTION FEBA.BAND                     8779
SMOKE.EFFECT
    PAGE 234  ROUTINE SMOKE.EFFECTS                   757  776
    PAGE 235                                          853
    PAGE 236                                          881
SMOKE.INPUT
    PAGE 521  ROUTINE MAIN2                          4287 4289
    PAGE 580  ROUTINE SMOKE.INPUT                    6668
SMOKE.MUNITION
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        816
    PAGE 580  ROUTINE SMOKE.INPUT                    6680 6682
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                7389
    PAGE 618  ROUTINE SNAP.R                         8244
    PAGE 674  ''PROGRAM  OLDER.VERSION                258
    PAGE 712                                         2468
SM.LOG.VALUE
    PAGE 231  ROUTINE SIZE.ESTIMATE                   632  671  677
SM.MAX.TANK.RATIO
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        794
    PAGE  48  ''SECTION FOR DEFINITIONS              2723
    PAGE 284  ROUTINE CAS.EVAL                       3257
    PAGE 582  ROUTINE TACAIR.INPUT                   6787
    PAGE 621  ROUTINE TACAIR.DATA.REPORT             8387
    PAGE 674  ''PROGRAM  OLDER.VERSION                236
    PAGE 706                                         2145
SM.MIN.CEQ
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        795
    PAGE 284  ROUTINE CAS.EVAL                       3256
    PAGE 582  ROUTINE TACAIR.INPUT                   6788
    PAGE 621  ROUTINE TACAIR.DATA.REPORT             8388
    PAGE 674  ''PROGRAM  OLDER.VERSION                237
SM.MIN.TANK.RATIO
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        793
    PAGE  48  ''SECTION FOR DEFINITIONS              2722
    PAGE 284  ROUTINE CAS.EVAL                       3258
    PAGE 582  ROUTINE TACAIR.INPUT                   6786
    PAGE 621  ROUTINE TACAIR.DATA.REPORT             8386
    PAGE 674  ''PROGRAM  OLDER.VERSION                235
    PAGE 706                                         2144
```

```
SM.NAME
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES          824
    PAGE  47  ''SECTION FOR DEFINITIONS                2690
    PAGE 557  ROUTINE SUBM.INPUT                       5724
    PAGE 674  ''PROGRAM  OLDER.VERSION                  266
    PAGE 706                                           2114
SM.ORDER
    PAGE  41  ''SECTION FOR EVENTS                     2324
    PAGE 700  ''PROGRAM  OLDER.VERSION                 1761
SM.TANK.TE
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES          792
    PAGE 283  ROUTINE CAS.EVAL                         3184
    PAGE 582  ROUTINE TACAIR.INPUT                     6785
    PAGE 621  ROUTINE TACAIR.DATA.REPORT               8385
    PAGE 673  ''PROGRAM  OLDER.VERSION                  234
SM.TYP.MOV
    PAGE  41  ''SECTION FOR EVENTS                     2323
    PAGE  47  ''SECTION FOR DEFINITIONS                2691
    PAGE 700  ''PROGRAM  OLDER.VERSION                 1760
    PAGE 706                                           2115
SM.UNIT
    PAGE  41  ''SECTION FOR EVENTS                     2320
    PAGE 112  ROUTINE PREPARE.LIST                     5393
    PAGE 700  ''PRC RAM  OLDER.VERSION                 1757
SM.XCOR
    PAGE  41  ''SECTION FOR EVENTS                     2321
    PAGE  44  ''SECTION FOR DEFINITIONS                2511
    PAGE 700  ''PROGRAM  OLDER.VERSION                 1758
    PAGE 703                                           1932
SM.YCOR
    PAGE  41  ''SECTION FOR EVENTS                     2322
    PAGE  44  ''SECTION FOR DEFINITIONS                2512
    PAGE 700  ''PROGRAM  OLDER.VERSION                 1759
    PAGE 703                                           1933
SORTIE.RATE
    PAGE  52  ''SECTION FOR SUBSTITUTIONS              2992
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS            3431
    PAGE 711  ''PROGRAM  OLDER.VERSION                 2399
SOUND.RANGE
    PAGE 246  ROUTINE VOLLEY                           1313 1342 1343
SOUND.TIME
    PAGE 246  ROUTINE VOLLEY                           1313 1343 1354
SOUTH
    PAGE   1  ROUTINE FOR CROSS_REFERENCING              56
    PAGE  51  ''SECTION FOR SUBSTITUTIONS              2900
    PAGE 709  ''PROGRAM  OLDER.VERSION                 2308
SO.AIR.ATK.INDIC
    PAGE  37  ''SECTION FOR PROCESSES                  2131
    PAGE 294  ROUTINE END.CAS.MISSION                  3735
    PAGE 423  PROCESS AC.ATK.TGT                       9105
    PAGE 425                                           9212
    PAGE 493  PROCESS SHOOT.OUT                        2777
    PAGE 494                                           2832
    PAGE 502                                           3281
    PAGE 508  PROCESS CAS.MISSION                      3590
    PAGE 697  ''PROGRAM  OLDER.VERSION                 1569
```

```
SO.DESTRUCT.INDIC
    PAGE  38  ''SECTION FOR PROCESSES                2136
    PAGE 294  ROUTINE END.CAS.MISSION                3742
    PAGE 329  ROUTINE EMPTY                          5263
    PAGE 493  PROCESS SHOOT.OUT                      2790
    PAGE 496                                         2957
    PAGE 499                                         3141
    PAGE 502                                         3296
    PAGE 697  ''PROGRAM  OLDER.VERSION               1574
SO.EX.SHOOT.OUT
    PAGE  38  ''SECTION FOR PROCESSES                2139
    PAGE 487  PROCESS ASSESSMENT                     2424 2425
    PAGE 493  PROCESS SHOOT.OUT                      2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2773
                                                     2791 2803
    PAGE 495                                         2880
    PAGE 496                                         2959 2971
    PAGE 500                                         3157 3190
    PAGE 502                                         3312
    PAGE 503                                         3331 3346
    PAGE 697  ''PROGRAM  OLDER.VERSION               1577
SO.FIRING.TABLE
    PAGE  38  ''SECTION FOR PROCESSES                2138
    PAGE  81  ROUTINE BLOCK.LOS                      4098 4100 4112 4114
    PAGE 108  ROUTINE NEW.SEGMENT                    5216
    PAGE 109                                         5218 5233 5235
    PAGE 293  ROUTINE END.CAS.MISSION                3729 3730 3731
    PAGE 301  ROUTINE HC.DISENGAGE                   4114
    PAGE 423  PROCESS AC.ATK.TGT                     9104
    PAGE 425                                         9211 9243 9244 9246
    PAGE 426                                         9294 9295 9297
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION         1096
    PAGE 488  PROCESS ASSESSMENT                     2517
    PAGE 489                                         2555
    PAGE 490                                         2606 2633 2634 2636
    PAGE 493  PROCESS SHOOT.OUT                      2776
    PAGE 495                                         2871 2873
    PAGE 497                                         2981 2983 2998 3000 3010
    PAGE 498                                         3071
    PAGE 500                                         3160
    PAGE 501                                         3215 3235
    PAGE 502                                         3315
    PAGE 697  ''PROGRAM  OLDER.VERSION               1576
SO.HELICOPTER
    PAGE  38  ''SECTION FOR PROCESSES                2137
    PAGE 301  ROUTINE HC.DISENGAGE                   4099 4111 4112
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION         1093 1094
    PAGE 487  PROCESS ASSESSMENT                     2431
    PAGE 488                                         2482 2483 2490 2494 2499 2508 2514 2515
    PAGE 494  PROCESS SHOOT.OUT                      2825 2851 2855 2862
    PAGE 495                                         2866 2889 2890 2894 2922
    PAGE 496                                         2923 2925
    PAGE 497                                         3011 3026 3030 3036 3037
    PAGE 498                                         3040 3078 3081 3082 3085
    PAGE 500                                         3212
    PAGE 501                                         3213
    PAGE 697  ''PROGRAM  OLDER.VERSION               1575
```

```
SO.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES          1816
    PAGE  38  ''SECTION FOR PROCESSES                   2141
    PAGE  81  ROUTINE BLOCK.LOS                         4097 4111
    PAGE 108  ROUTINE NEW.SEGMENT                       5215
    PAGE 109                                            5232
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                   2927 2934
    PAGE 282  ROUTINE AC.DF.EFFECTS                     3113 3118
    PAGE 293  ROUTINE END.CAS.MISSION                   3725
    PAGE 294                                            3743
    PAGE 329  ROUTINE EMPTY                             5258 5260
    PAGE 370  EVENT ENGAGEMENT                          6736
    PAGE 379  EVENT HELO.ENGAGEMENT                     7216
    PAGE 383  EVENT OFF.LINE.ATTRITION                  7374
    PAGE 398  EVENT START.BATTLE                        7969
    PAGE 419  PROCESS AC.ATK.TGT                        8890
    PAGE 423                                            9089 9103 9121
    PAGE 425                                            9197 9210 9228 9242
    PAGE 426                                            9272
    PAGE 435  PROCESS ARTY.ASSESS                       9722 9732
    PAGE 436                                            9780
    PAGE 464  PROCESS MINE.ASSESS                       1267
    PAGE 465                                            1282
    PAGE 491  PROCESS ASSESSMENT                        2686 2690
    PAGE 493  PROCESS SHOOT.OUT                         2802
    PAGE 495                                            2870 2878
    PAGE 496                                            2969
    PAGE 500                                            3174
    PAGE 503                                            3329 3344
    PAGE 508  PROCESS CAS.MISSION                       3582 3589
    PAGE 516  PROCESS HELICOPTER.FIRE                   4035 4044
    PAGE 619  ROUTINE SNAP.R                            8274
    PAGE 691  ''PROGRAM  OLDER.VERSION                  1255
    PAGE 697                                            1579
SO.RESULTS
    PAGE  38  ''SECTION FOR PROCESSES                   2135
    PAGE 487  PROCESS ASSESSMENT                        2460
    PAGE 488                                            2481
    PAGE 494  PROCESS SHOOT.OUT                         2809
    PAGE 498                                            3069
    PAGE 697  ''PROGRAM  OLDER.VERSION                  1573
SPAM.BTRY
    PAGE  41  ''SECTION FOR EVENTS                      2327
    PAGE 104  ROUTINE MINE.DELAY                        5031
    PAGE 407  EVENT STOP.ARTY.MOVEMENT                  8380
    PAGE 700  ''PROGRAM  OLDER.VERSION                  1764
SPAM.DIRECTION
    PAGE  41  ''SECTION FOR EVENTS                      2328
    PAGE  45  ''SECTION FOR DEFINITIONS                 2572
    PAGE 700  ''PROGRAM  OLDER.VERSION                  1765
    PAGE 704                                            1993
SPEED.
    PAGE 583  ROUTINE TACAIR.INPUT                      6850
SP.X.COORD
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES           714
    PAGE  45  ''SECTION FOR DEFINITIONS                 2573
```

```
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4810 4812 4813 4815 4816
    PAGE 672  ''PROGRAM  OLDER.VERSION                 157
    PAGE 704                                          1994
SP.X.GRID
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         714
    PAGE  45  ''SECTION FOR DEFINITIONS               2575
    PAGE  99  ROUTINE LOCATE.SEARCH.AREA              4828
    PAGE 672  ''PROGRAM  OLDER.VERSION                 157
    PAGE 704                                          1996
SP.Y.COORD
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         715
    PAGE  45  ''SECTION FOR DEFINITIONS               2574
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4819 4821 4822 4824 4825
    PAGE 672  ''PROGRAM  OLDER.VERSION                 158
    PAGE 704                                          1995
SP.Y.GRID
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         715
    PAGE  45  ''SECTION FOR DEFINITIONS               2576
    PAGE  99  ROUTINE LOCATE.SEARCH.AREA              4829
    PAGE 672  ''PROGRAM  OLDER.VERSION                 158
    PAGE 704                                          1997
SQRT.F
    PAGE  78  ROUTINE ADJUST                          3956 3970 3975
    PAGE 103  ROUTINE MINE.DELAY                      4977
    PAGE 154  ROUTINE AO.DETECTION                    7127
    PAGE 155                                          7133 7137 7151
    PAGE 156                                          7240
    PAGE 174  ROUTINE DUST.EFFECTS                    7950
    PAGE 178  ROUTINE EST.COVERAGE                    8146 8149 8150 8152 8156 8163 8165 8169 8170
    PAGE 180  ROUTINE EST.MIL.WORTH                   8234
    PAGE 190  ROUTINE FINAL.COVERAGE                  8696 8697 8698 8699 8702
    PAGE 191                                          8722 8737 8738 8745 8746
    PAGE 254  ROUTINE FO.DETECTION                    1701
    PAGE 267  ROUTINE BTRY.EFFECTS                    2408
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                 2846
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS           3445
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                3955
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE               4137
    PAGE 316  ROUTINE FLIGHT.PATH                     4768
    PAGE 317                                          4856
    PAGE 318                                          4869
    PAGE 319                                          4922
    PAGE 335  ROUTINE FRAC.COMPUTE                    5431
    PAGE 342  ROUTINE RANGE.COMPUTE                   5672
    PAGE 343  ROUTINE SEARCH.COVERAGE                 5686 5693 5695 5705 5706
    PAGE 350  EVENT AD.ENGAGEMENT                     5956
    PAGE 351                                          6035 6063
    PAGE 352                                          6083 6098 6123
    PAGE 353                                          6159 6160
    PAGE 354                                          6209
    PAGE 404  EVENT START.MOVE                        8284
    PAGE 408  EVENT UPDATE.LOC                        8437
    PAGE 410                                          8538
    PAGE 420  PROCESS AC.ATK.TGT                      8914 8917
    PAGE 422                                          9038
    PAGE 428  PROCESS AIR.OBSERVER                    9382
```

```
    PAGE 429                                          9414 9416 9418 9439
    PAGE 430                                          9469 9484 9489 9491
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                 196  205
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE            1393 1406
    PAGE 468                                          1414 1419 1421
    PAGE 506  PROCESS CAS.MISSION                     3514 3520
    PAGE 584  ROUTINE TACAIR.INPUT                    6855 6857 6859 6885 6888 6891
    PAGE 585                                          6961
    PAGE 628  FUNCTION ACT.RANGE                      8641
    PAGE 632  FUNCTION EST.RANGE                      8754
    PAGE 633  FUNCTION EST.TR.RANGE                   8767
    PAGE 641  ROUTINE NORMAL.F                        9059
    PAGE 645  ROUTINE LINE.CIRCLE                     9146
    PAGE 653  PROCESS AIRBORNE.RADAR                  9295
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                 9513 9527 9529 9531
SQ.
    PAGE  12  ''SECTION FOR PERMANENT_ENTITIES         647
    PAGE 671  ''PROGRAM  OLDER.VERSION                  88
SQ.METERS
    PAGE 557  ROUTINE SUBM.INPUT                      5736
SQ.M.
    PAGE 557  ROUTINE SUBM.INPUT                      5745
SS.REAR
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES         805
    PAGE  61  ROUTINE FEBA.INITIAL                    3203 3204 3228 3232
    PAGE 119  ROUTINE RESET.FEBA.SECTOR               5663 5674 5692 5696
    PAGE 331  ROUTINE ENQ.FEBA.SET                    5321 5329 5330
    PAGE 674  ''PROGRAM  OLDER.VERSION                 247
SS.SET
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES         807
    PAGE  18                                          1007
    PAGE  49  ''SECTION FOR DEFINITIONS               2823
    PAGE  61  ROUTINE FEBA.INITIAL                    3229 3233
    PAGE  62                                          3246
    PAGE  89  ROUTINE FA.BN.MOVEMENT                  4437 4439 4443
    PAGE  91                                          4517 4519 4523
    PAGE 119  ROUTINE RESET.FEBA.SECTOR               5661 5693 5697
*   PAGE 138  ROUTINE CHECK.PROX                      6398 6403 6408 6422
    PAGE 239  ROUTINE TARGET.ANALYSIS                 1009
    PAGE 246  ROUTINE VOLLEY                          1327 1332
    PAGE 261  ROUTINE BTRY.EFFECTS                    2025
    PAGE 325  ROUTINE DEQ.FEBA.SET                    5091 5102 5106 5111
    PAGE 331  ROUTINE ENQ.FEBA.SET                    5323 5337
    PAGE 360  EVENT CFR.ON                            6436
    PAGE 634  FUNCTION FEBA.BAND                      8797 8815 8827
    PAGE 674  ''PROGRAM  OLDER.VERSION                 249
    PAGE 677                                           449
    PAGE 708                                          2244
STAM.BTRY
    PAGE  40  ''SECTION FOR EVENTS                    2308
    PAGE 393  EVENT START.ARTY.MOVEMENT               7668
    PAGE 700  ''PROGRAM  OLDER.VERSION                1745
STAM.DIRECTION
    PAGE  41  ''SECTION FOR EVENTS                    2309
    PAGE  45  ''SECTION FOR DEFINITIONS               2577
    PAGE 700  ''PROGRAM  OLDER.VERSION                1746
```

```
    PAGE 704                                    1998
STARTED.
    PAGE 146  ROUTINE INTER.BATTLE              6745
    PAGE 327  ROUTINE DQ.CMSN.QUEUE             5150
START.ARTY.M
    PAGE 393  EVENT START.ARTY.MOVEMENT         7707
START.ARTY.MOVEMENT
    PAGE  40  ''SECTION FOR EVENTS              2307
    PAGE  90  ROUTINE FA.BN.MOVEMENT            4465 4485
    PAGE  91                                    4544 4558
    PAGE  92                                    4575
    PAGE 393  EVENT START.ARTY.MOVEMENT         7655
    PAGE 602  ROUTINE BETWEEN.ROUTINE           7632 7633
    PAGE 616  ROUTINE SNAP2                     8189
    PAGE 700  ''PROGRAM  OLDER.VERSION          1744
START.BATTLE
    PAGE  41  ''SECTION FOR EVENTS              2311
    PAGE 147  ROUTINE INTER.BATTLE              6851
    PAGE 148                                    6888
    PAGE 395  EVENT START.BATTLE                7743
    PAGE 401                                    8103 8127
    PAGE 412  EVENT ACT.ATK                     8610
    PAGE 413                                    8703
    PAGE 602  ROUTINE BETWEEN.ROUTINE           7636 7637
    PAGE 616  ROUTINE SNAP2                     8193
    PAGE 700  ''PROGRAM  OLDER.VERSION          1748
START.MOVE
    PAGE  41  ''SECTION FOR EVENTS              2319
    PAGE  95  ROUTINE INIT.REINF                4643 4660
    PAGE 112  ROUTINE PREPARE.LIST              5396 5397
    PAGE 403  EVENT START.MOVE                  8177
    PAGE 405                                    8319
    PAGE 415  EVENT ACT.MOVCOR                  8756
    PAGE 416  EVENT ACT.MOVDIS                  8774 8793
    PAGE 602  ROUTINE BETWEEN.ROUTINE           7640 7641
    PAGE 617  ROUTINE SNAP2                     8194
    PAGE 700  ''PROGRAM  OLDER.VERSION          1756
START.TIME
    PAGE 194  ROUTINE FIND.START.TIME           8897 8900 8906 8928
    PAGE 195                                    8987
    PAGE 438  PROCESS FORWARD.OBSERVER          9871 9892
    PAGE 441                                      70   72
    PAGE 470  PROCESS TARGET.REPORT             1496
    PAGE 473                                    1690 1691 1691
    PAGE 476                                    1848 1852
STATUS.
    PAGE 130  ROUTINE CHECK.DEAD                6068 6113 6115
STAY.TIME
    PAGE  50  ''SECTION FOR DEFINITIONS         2859
    PAGE 658  PROCESS PHOTO.IR.FLIGHT           9518 9519
    PAGE 660  FUNCTION STAY.TIME                9590
    PAGE 709  ''PROGRAM  OLDER.VERSION          2278
STA.TO.WITH
    PAGE  51  ''SECTION FOR SUBSTITUTIONS       2936
    PAGE  84  ROUTINE CHANGE.LOC                4203
    PAGE 101  ROUTINE LOS.CHECK                 4891 4893 4900
```

```
    PAGE 103  ROUTINE MINE.DELAY                    4957
    PAGE 111  ROUTINE PRED.POS                      5312
    PAGE 114  ROUTINE PREP.WITHDRAW                 5471 5477
    PAGE 115  ROUTINE PROX.CHECK                    5518
    PAGE 200  ROUTINE ILLUM.EFFECTS                 9203
    PAGE 234  ROUTINE SMOKE.EFFECTS                  766
    PAGE 252  ROUTINE MINE.EFFECTS                  1607
    PAGE 256  ROUTINE FO.DETECTION                  1843
    PAGE 267  ROUTINE BTRY.EFFECTS                  2411
    PAGE 324  ROUTINE DECIDE                        5048
    PAGE 407  EVENT STOP.ARTY.MOVEMENT              8394
    PAGE 448  PROCESS HC.ARRIVE.BATTLE               414
    PAGE 478  PROCESS WITH.DRAW                     1951
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9520
    PAGE 710  ''PROGRAM  OLDER.VERSION              2343
STOP.ARTY.MOVEMENT
    PAGE  41  ''SECTION FOR EVENTS                  2326
    PAGE 394  EVENT START.ARTY.MOVEMENT             7733
    PAGE 407  EVENT STOP.ARTY.MOVEMENT              8370
    PAGE 602  ROUTINE BETWEEN.ROUTINE               7644 7645
    PAGE 616  ROUTINE SNAP2                         8191
    PAGE 700  ''PROGRAM  OLDER.VERSION              1763
STOP.SIMULATION.TIME
    PAGE  48  ''SECTION FOR DEFINITIONS             2749
    PAGE  57  ROUTINE MAIN3                         3078
    PAGE 368  EVENT END.SIMULATION                  8666
    PAGE 438  PROCESS FORWARD.OBSERVER              9887
    PAGE 523  ROUTINE SYS.INPUT                     4351
    PAGE 606  ROUTINE KV.PRINT                      7725
    PAGE 707  ''PROGRAM  OLDER.VERSION              2171
STW.RND.FIRED
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES       814
    PAGE 227  ROUTINE REQUEST.WD.FASCAM              473
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2816
    PAGE 280  ROUTINE AC.DF.EFFECTS                 3050
    PAGE 310  ROUTINE AD.SHOOT                      4473
    PAGE 384  EVENT OFF.LINE.ATTRITION              7410
    PAGE 499  PROCESS SHOOT.OUT                     3104
    PAGE 514  PROCESS HELICOPTER.FIRE               3917
    PAGE 593  ROUTINE AMMO.RPT                      7233 7233
    PAGE 674  ''PROGRAM  OLDER.VERSION               256
STY.BLUE.EXP
    PAGE  50  ''SECTION FOR DEFINITIONS             2841
    PAGE 281  ROUTINE AC.DF.EFFECTS                 3056
    PAGE 310  ROUTINE AD.SHOOT                      4476
    PAGE 384  EVENT OFF.LINE.ATTRITION              7412
    PAGE 499  PROCESS SHOOT.OUT                     3117
    PAGE 514  PROCESS HELICOPTER.FIRE               3932
    PAGE 520  ROUTINE MAIN2                         4213
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES           8514 8515
    PAGE 625                                        8548 8601 8603
    PAGE 708  ''PROGRAM  OLDER.VERSION              2262
STY.RED.EXP
    PAGE  50  ''SECTION FOR DEFINITIONS             2841
    PAGE 281  ROUTINE AC.DF.EFFECTS                 3059
    PAGE 310  ROUTINE AD.SHOOT                      4479
```

```
    PAGE 384  EVENT OFF.LINE.ATTRITION                7414
    PAGE 499  PROCESS SHOOT.OUT                       3120
    PAGE 514  PROCESS HELICOPTER.FIRE                 3936
    PAGE 520  ROUTINE MAIN2                           4212
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES             8523 8524
    PAGE 625                                          8578
    PAGE 626                                          8615 8617
    PAGE 708  ''PROGRAM  OLDER.VERSION                2262
ST.FARRP
    PAGE  40  ''SECTION FOR EVENTS                    2304
    PAGE 337  ROUTINE HC.EMPTY                        5486 5490
    PAGE 699  ''PROGRAM  OLDER.VERSION                1741
ST.INPUT
    PAGE 521  ROUTINE MAIN2                           4248 4250
    PAGE 561  ROUTINE ST.INPUT                        5855
ST.MAX.RANGE
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         725
    PAGE 430  PROCESS AIR.OBSERVER                    9486
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE            1416
    PAGE 561  ROUTINE ST.INPUT                        5871 5873 5873
    PAGE 672  ''PROGRAM  OLDER.VERSION                 168
ST.MAX.XMIT
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         723
    PAGE 362  EVENT CFR.OPERATOR                      6493
    PAGE 387  EVENT PDB.OPERATOR                      7517
    PAGE 432  PROCESS AIR.OBSERVER                    9563
    PAGE 440  PROCESS FORWARD.OBSERVER                  24
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE            1454
    PAGE 561  ROUTINE ST.INPUT                        5869
    PAGE 654  PROCESS AIRBORNE.RADAR                  9365
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                 9567
    PAGE 672  ''PROGRAM  OLDER.VERSION                 166
ST.MIN.XMIT
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         722
    PAGE 362  EVENT CFR.OPERATOR                      6492
    PAGE 387  EVENT PDB.OPERATOR                      7516
    PAGE 432  PROCESS AIR.OBSERVER                    9562
    PAGE 440  PROCESS FORWARD.OBSERVER                  23
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE            1453
    PAGE 561  ROUTINE ST.INPUT                        5868
    PAGE 654  PROCESS AIRBORNE.RADAR                  9364
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                 9566
    PAGE 672  ''PROGRAM  OLDER.VERSION                 165
ST.NAME
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         721
    PAGE  47  ''SECTION FOR DEFINITIONS               2692
    PAGE 157  ROUTINE AO.DETECTION                    7268
    PAGE 159  ROUTINE ATTRIT.SENSOR                   7343 7347 7351
    PAGE 160                                          7378 7416 7417
    PAGE 161                                          7439 7440
    PAGE 162                                          7487 7506 7507
    PAGE 206  ROUTINE PDB.DETECTION                   9448 9450 9470
    PAGE 216  ROUTINE REQUEST.FASCAM                  9926 9940
    PAGE 217                                          9948
    PAGE 219  ROUTINE REQUEST.ILLUM                     56   70   78
    PAGE 222  ROUTINE REQUEST.SMOKE                    229
```

```
    PAGE 223                                    243  251
    PAGE 230  ROUTINE RPV.DETECTION             591
    PAGE 238  ROUTINE SWITCH.FO                 930
    PAGE 256  ROUTINE FO.DETECTION             1839
    PAGE 358  EVENT CFR.ACTIVATION             6370
    PAGE 362  EVENT CFR.OPERATOR               6506
    PAGE 386  EVENT PDB.ACTIVATION             7496
    PAGE 387  EVENT PDB.OPERATOR               7525
    PAGE 419  PROCESS AC.ATK.TGT               8904
    PAGE 561  ROUTINE ST.INPUT                 5867
    PAGE 565  ROUTINE SENSOR.INPUT             6053
    PAGE 566                                   6082 6087 6096 6117 6118
    PAGE 567                                   6127 6128
    PAGE 619  ROUTINE SNAP.R                   8286 8291 8292 8295 8296
    PAGE 656  ROUTINE AR.DETECTION             9425
    PAGE 659  PROCESS PHOTO.IR.FLIGHT          9572
    PAGE 672  ''PROGRAM  OLDER.VERSION          164
    PAGE 706                                   2116
ST.TEAM
    PAGE  40  ''SECTION FOR EVENTS             2305
    PAGE 337  ROUTINE HC.EMPTY                 5491 5492 5494 5495
    PAGE 699  ''PROGRAM  OLDER.VERSION         1742
ST.TE.PTR
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES  724
    PAGE 561  ROUTINE ST.INPUT                 5870
    PAGE 672  ''PROGRAM  OLDER.VERSION          167
SUBMUNITION
    PAGE   1  ROUTINE FOR CROSS_REFERENCING      45
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES   401  404
    PAGE  15                                     823  824
    PAGE  17                                     922
    PAGE 557  ROUTINE SUBM.INPUT               5716 5722 5723 5724 5726 5729 5732 5736 5739 5745
    PAGE 618  ROUTINE SNAP.R                   8245
    PAGE 667  ''PROGRAM  OLDER.VERSION         9843 9846
    PAGE 674                                    265  266
    PAGE 676                                    364
    PAGE 712                                   2469
SUBM.INPUT
    PAGE 520  ROUTINE MAIN2                    4239
    PAGE 521                                   4241
    PAGE 557  ROUTINE SUBM.INPUT               5713
SUBSTITUTED.
    PAGE 309  ROUTINE AD.SHOOT                 4420
SUBST.
    PAGE 622  ROUTINE TACAIR.DATA.REPORT       8398
SUC.REINF.OP
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES 1708
    PAGE 118  ROUTINE REIN.ARRIVE              5629
    PAGE 544  ROUTINE ORD.REINF                5174
    PAGE 689  ''PROGRAM  OLDER.VERSION         1147
SUPPRESSED.
    PAGE 183  ROUTINE FA.BN.ASGN               8359
SUPP.TIME
    PAGE 480  PROCESS FIRE.MISSION             2052
    PAGE 481                                   2129 2131 2133
```

```
SUP.MISSION.PRIORITY
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES          586
    PAGE 305  ROUTINE UNIT.PRIORITY                    4257 4263
    PAGE 570  ROUTINE DECISION.INPUT                   6275
    PAGE 670  ''PROGRAM  OLDER.VERSION                   27
SUP.MIS.PRIOR
    PAGE 305  ROUTINE UNIT.PRIORITY                    4260
SUP.NO.UNITS
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                  173  181  182
    PAGE 445                                            227  232
SUP.X.COORD
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                  171  181  181  186  189  191  196  205
    PAGE 445                                            215  219  221  227  232
    PAGE 446                                            299
    PAGE 447                                            327  368  372
SUP.Y.COORD
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                  172  182  182  186  197  206
    PAGE 445                                            215  227  232
    PAGE 446                                            300
    PAGE 447                                            328  369  373
SYS.INPUT
    PAGE 520  ROUTINE MAIN2                            4188 4195
    PAGE 523  ROUTINE SYS.INPUT                        4317 4332
S1.X.PERPEND
    PAGE 343  ROUTINE SEARCH.COVERAGE                  5687 5691 5693
S1.Y.PERPEND
    PAGE 343  ROUTINE SEARCH.COVERAGE                  5691 5694
S2.X.PERPEND
    PAGE 343  ROUTINE SEARCH.COVERAGE                  5689 5692 5695
S2.Y.PERPEND
    PAGE 343  ROUTINE SEARCH.COVERAGE                  5692 5696
S.
    PAGE 664  ''PROGRAM  OLDER.VERSION                 9681
S.AATT.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES         1096
    PAGE 679  ''PROGRAM  OLDER.VERSION                  537
S.AA.SET
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES         1120
    PAGE 679  ''PROGRAM  OLDER.VERSION                  559
S.AO.CAND.DET.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES         1131
    PAGE 679  ''PROGRAM  OLDER.VERSION                  570
S.AO.DET.TGT.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES         1039
    PAGE 678  ''PROGRAM  OLDER.VERSION                  481
S.AO.EB.SET
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES          304
    PAGE 665  ''PROGRAM  OLDER.VERSION                 9746
S.AO.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES         1398
    PAGE 684  ''PROGRAM  OLDER.VERSION                  837
S.AO.RB.SET
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES          317
    PAGE 665  ''PROGRAM  OLDER.VERSION                 9759
S.AR.CAND.DET.LIST
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES         1141
```

```
    PAGE 679  ''PROGRAM  OLDER.VERSION                580
S.AR.DET.TGT.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES        1041
    PAGE 678  ''PROGRAM  OLDER.VERSION                483
S.AVAIL.AO.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES        1853
    PAGE 692  ''PROGRAM  OLDER.VERSION                1292
S.BATTLE.SET
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES        1174
    PAGE 680  ''PROGRAM  OLDER.VERSION                613
S.BN.BTRY.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES        346
    PAGE 666  ''PROGRAM  OLDER.VERSION                9788
S.BN.CAN.FM.SET
    PAGE  36  ''SECTION FOR PROCESSES                 2020
    PAGE 695  ''PROGRAM  OLDER.VERSION                1458
S.BTL.FORCE.SET
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES        1420
    PAGE 684  ''PROGRAM  OLDER.VERSION                859
S.BY.FM.QUEUE
    PAGE  36  ''SECTION FOR PROCESSES                 2024
    PAGE 695  ''PROGRAM  OLDER.VERSION                1462
S.BY.HOW.SET
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES        1494
    PAGE 686  ''PROGRAM  OLDER.VERSION                933
S.BY.SCHD.LIST
    PAGE  36  ''SECTION FOR PROCESSES                 2022
    PAGE 695  ''PROGRAM  OLDER.VERSION                1460
S.CFPS.LIST
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1208
    PAGE 316  ROUTINE FLIGHT.PATH                     4761 4765 4766 4774
    PAGE 681  ''PROGRAM  OLDER.VERSION                647
S.CF.OP.Q
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1220
    PAGE 681  ''PROGRAM  OLDER.VERSION                659
S.COL.SET
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES        1190
    PAGE 680  ''PROGRAM  OLDER.VERSION                629
S.CT.TU.SET
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES        951
    PAGE 676  ''PROGRAM  OLDER.VERSION                393
S.DF.RATE.LIST
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES        1254
    PAGE 681  ''PROGRAM  OLDER.VERSION                693
S.FD.BN.LIST
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES        1354
    PAGE 683  ''PROGRAM  OLDER.VERSION                793
S.FD.COMPLETE.LIST
    PAGE  38  ''SECTION FOR PROCESSES                 2186
    PAGE 697  ''PROGRAM  OLDER.VERSION                1624
S.FD.SCHD.LIST
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES        1365
    PAGE 683  ''PROGRAM  OLDER.VERSION                804
S.FD.TR.QUEUE
    PAGE  38  ''SECTION FOR PROCESSES                 2184
    PAGE 697  ''PROGRAM  OLDER.VERSION                1622
```

```
S.FO.CAND.DET.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1430
    PAGE 684  ''PROGRAM  OLDER.VERSION                   869
S.FO.CUR.FM.LIST
    PAGE  36  ''SECTION FOR PROCESSES                   2026
    PAGE 695  ''PROGRAM  OLDER.VERSION                  1464
S.FO.TGT.RPT.LIST
    PAGE  38  ''SECTION FOR PROCESSES                   2188
    PAGE 697  ''PROGRAM  OLDER.VERSION                  1626
S.FP.SET
    PAGE  24  ''SECTION FOR TEMPORARY_ENTITIES          1338
    PAGE 683  ''PROGRAM  OLDER.VERSION                   777
S.FR.UNIT.SET
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1037
    PAGE 678  ''PROGRAM  OLDER.VERSION                   479
S.GP.CAT.SET
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES           365
    PAGE 666  ''PROGRAM  OLDER.VERSION                  9807
S.HC.UN.LOS.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1869
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1308
S.HE.TB.RH.LIST
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES           516
    PAGE 669  ''PROGRAM  OLDER.VERSION                  9957
S.HF.SO.LIST
    PAGE  36  ''SECTION FOR PROCESSES                   2073
    PAGE 696  ''PROGRAM  OLDER.VERSION                  1511
S.HT.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES          1459
    PAGE 685  ''PROGRAM  OLDER.VERSION                   898
S.HT.MEMBER.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES          1479
    PAGE 685  ''PROGRAM  OLDER.VERSION                   918
S.HT.TARGET.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1057
    PAGE 678  ''PROGRAM  OLDER.VERSION                   499
S.IC.TB.RH.LIST
    PAGE  10  ''SECTION FOR PERMANENT_ENTITIES           553
    PAGE 669  ''PROGRAM  OLDER.VERSION                  9994
S.IF.RATE.LIST
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1504
    PAGE 686  ''PROGRAM  OLDER.VERSION                   943
S.M
    PAGE 112  ROUTINE PREPARE.LIST                      5392 5393 5396 5397
S.MADS.RH.SET
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1534
    PAGE 686  ''PROGRAM  OLDER.VERSION                   973
S.MA.SET
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1568
    PAGE 687  ''PROGRAM  OLDER.VERSION                  1007
S.MCFR.RH.LIST
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES           386
    PAGE 666  ''PROGRAM  OLDER.VERSION                  9828
S.MFO.RB.SET
    PAGE   9  ''SECTION FOR PERMANENT_ENTITIES           473
    PAGE 668  ''PROGRAM  OLDER.VERSION                  9914
```

```
S.MFP.LIST
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1589
    PAGE 135  ROUTINE CHECK.FOR.MINES                   6277 6280 6281 6283 6309 6311 6313 6315
    PAGE 687  ''PROGRAM  OLDER.VERSION                  1028
S.MO.LIST
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1579
    PAGE 687  ''PROGRAM  OLDER.VERSION                  1018
S.MPDB.RH.LIST
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES           707
    PAGE 672  ''PROGRAM  OLDER.VERSION                   150
S.MU.ORDER.SET
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES          1628
    PAGE 688  ''PROGRAM  OLDER.VERSION                  1067
S.MU.TF.LIST
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1558
    PAGE 687  ''PROGRAM  OLDER.VERSION                   997
S.O
    PAGE 493  PROCESS SHOOT.OUT                         2771
S.O.
    PAGE 495  PROCESS SHOOT.OUT                         2870 2871 2873
S.PATH.SET
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES          1669
    PAGE 689  ''PROGRAM  OLDER.VERSION                  1108
S.PDB.KEYED.LIST
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1514
    PAGE 686  ''PROGRAM  OLDER.VERSION                   953
S.PDB.OP.Q
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES          1659
    PAGE 688  ''PROGRAM  OLDER.VERSION                  1098
S.PIR.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1396
    PAGE 684  ''PROGRAM  OLDER.VERSION                   835
S.PIR.RECORD.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES          1680
    PAGE 689  ''PROGRAM  OLDER.VERSION                  1119
S.PIR.RTD.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES          1694
    PAGE 689  ''PROGRAM  OLDER.VERSION                  1133
S.RPV.CAND.DET.LIST
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES          1719
    PAGE 689  ''PROGRAM  OLDER.VERSION                  1158
S.RPV.FLIGHT.LEG.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1400
    PAGE 684  ''PROGRAM  OLDER.VERSION                   839
S.SD.ADS.SET
    PAGE  20  ''SECTION FOR TEMPORARY_ENTITIES          1110
    PAGE 679  ''PROGRAM  OLDER.VERSION                   549
S.SD.CMSN.QUEUE
    PAGE  35  ''SECTION FOR PROCESSES                   1991
    PAGE 694  ''PROGRAM  OLDER.VERSION                  1429
S.SD.FPO.LIST
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES          1442
    PAGE 685  ''PROGRAM  OLDER.VERSION                   881
S.SD.KAS.SET
    PAGE  27  ''SECTION FOR TEMPORARY_ENTITIES          1523
    PAGE 686  ''PROGRAM  OLDER.VERSION                   962
```

```
S.SD.OLD.SORTIE.QUEUE
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES          1618
    PAGE 688  ''PROGRAM  OLDER.VERSION                  1057
S.SIDE.CFR.SET
    PAGE  22  ''SECTION FOR TEMPORARY_ENTITIES          1234
    PAGE 681  ''PROGRAM  OLDER.VERSION                   673
S.SIDE.PDB.SET
    PAGE  29  ''SECTION FOR TEMPORARY_ENTITIES          1647
    PAGE 688  ''PROGRAM  OLDER.VERSION                  1086
S.SI.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1734
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1173
S.SO.LIST
    PAGE  38  ''SECTION FOR PROCESSES                   2144
    PAGE 697  ''PROGRAM  OLDER.VERSION                  1582
S.SS.SET
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1033
    PAGE 678  ''PROGRAM  OLDER.VERSION                   475
S.TB.SORT.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          1018
    PAGE 677  ''PROGRAM  OLDER.VERSION                   460
S.TB.TM.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1757
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1196
S.TEAM.TYPES
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES          1805
    PAGE 691  ''PROGRAM  OLDER.VERSION                  1244
S.TE.SET
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES           420
    PAGE 667  ''PROGRAM  OLDER.VERSION                  9861
S.TR.DET.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1768
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1207
S.TR.FM.LIST
    PAGE  36  ''SECTION FOR PROCESSES                   2018
    PAGE 695  ''PROGRAM  OLDER.VERSION                  1456
S.TU.NTE.SET
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1777
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1216
S.TU.TE.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES          1788
    PAGE 691  ''PROGRAM  OLDER.VERSION                  1227
S.UE.TARGET.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES          1382
    PAGE 684  ''PROGRAM  OLDER.VERSION                   821
S.UE.WEAPON.SET
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1884
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1323
S.UNIT.SET
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES          1043
    PAGE 678  ''PROGRAM  OLDER.VERSION                   485
S.UN.EQUIP.LIST
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES          1822
    PAGE 691  ''PROGRAM  OLDER.VERSION                  1261
S.UN.HC.LOS.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1872
```

```
    PAGE 692  ''PROGRAM  OLDER.VERSION                1311
S.UN.LOS.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES        1866
    PAGE 692  ''PROGRAM  OLDER.VERSION                1305
S.UN.PATH
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES        1704
    PAGE 689  ''PROGRAM  OLDER.VERSION                1143
S.UN.SEGMENT.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES        1745
    PAGE 690  ''PROGRAM  OLDER.VERSION                1184
S.UN.SENSOR.LIST
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES        1849
    PAGE 692  ''PROGRAM  OLDER.VERSION                1288
S.UN.SUB.LIST
    PAGE  19  ''SECTION FOR PERMANENT_ENTITIES        1035
    PAGE 678  ''PROGRAM  OLDER.VERSION                 477
TACAIR.DEBUG
    PAGE  43  ''SECTION FOR DEFINITIONS               2429
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                 2794
    PAGE 277                                          2878
    PAGE 280  ROUTINE AC.DF.EFFECTS                   3005
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS           3288 3309 3319 3334
    PAGE 286                                          3356 3367 3383
    PAGE 287                                          3402
    PAGE 307  ROUTINE AD.SHOOT                        4315
    PAGE 310                                          4505
    PAGE 314  ROUTINE FLIGHT.PATH                     4665 4685
    PAGE 315                                          4712
    PAGE 316                                          4772
    PAGE 327  ROUTINE DQ.CMSN.QUEUE                   5153
    PAGE 349  EVENT AD.ENGAGEMENT                     5909
    PAGE 350                                          5983 5996
    PAGE 351                                          6025 6047 6052
    PAGE 352                                          6090 6114
    PAGE 353                                          6139 6176 6184
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE               6650
    PAGE 380  EVENT INIT.PREPLAN.CAS                  7253
    PAGE 419  PROCESS AC.ATK.TGT                      8881
    PAGE 421                                          9002 9011 9019
    PAGE 423                                          9082 9114
    PAGE 424                                          9190
    PAGE 425                                          9221
    PAGE 426                                          9265 9284
    PAGE 504  PROCESS CAS.MISSION                     3366
    PAGE 505                                          3420 3426 3429
    PAGE 506                                          3474
    PAGE 507                                          3569
    PAGE 508                                          3600
    PAGE 509                                          3643
    PAGE 582  ROUTINE TACAIR.INPUT                    6751
    PAGE 588  ROUTINE AC.MUNS.INPUT                   7047
    PAGE 621  ROUTINE TACAIR.DATA.REPORT              8339
    PAGE 702  ''PROGRAM  OLDER.VERSION                1859
TACAIR.INPUT
    PAGE   3  PROGRAM REVISIONS                        134
    PAGE 521  ROUTINE MAIN2                           4293 4297
```

```
    PAGE 582  ROUTINE TACAIR.INPUT                    6736
TAC.FAC
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9497
    PAGE 210                                          9621 9626
    PAGE 393  EVENT START.ARTY.MOVEMENT               7664 7711
    PAGE 394                                          7730
    PAGE 403  EVENT START.MOVE                        8193 8200 8202 8216
TAC.MOV.FAC
    PAGE  48  ''SECTION FOR DEFINITIONS               2750
    PAGE  80  ROUTINE BLOCK.LOS                       4022
    PAGE  83  ROUTINE CHANGE.LOC                      4148
    PAGE 106  ROUTINE MIN.MOVE                        5088
    PAGE 210  ROUTINE PGM.MSN.ASGN                    9621
    PAGE 523  ROUTINE SYS.INPUT                       4369
    PAGE 707  ''PROGRAM  OLDER.VERSION                2172
TAC.MOV.FAC.
    PAGE 403  EVENT START.MOVE                        8202
TAKEOFF.
    PAGE 504  PROCESS CAS.MISSION                     3396
TARGETED.
    PAGE 242  ROUTINE UNIT.ENVIR                      1125
    PAGE 283  ROUTINE CAS.EVAL                        3219
TARGETS.
    PAGE 420  PROCESS AC.ATK.TGT                      8922
TARGET.
    PAGE 293  ROUTINE END.CAS.MISSION                 3711
    PAGE 420  PROCESS AC.ATK.TGT                      8961
    PAGE 422                                          9073
    PAGE 505  PROCESS CAS.MISSION                     3455
    PAGE 506                                          3471
TARGET.ACQUISITION
    PAGE 239  ROUTINE TARGET.ANALYSIS                  988
    PAGE 480  PROCESS FIRE.MISSION                    2039
    PAGE 482                                          2165
TARGET.ANALYSIS
    PAGE   3  PROGRAM REVISIONS                        150  151
    PAGE   4                                           181
    PAGE 180  ROUTINE EST.MIL.WORTH                   8223
    PAGE 231  ROUTINE SIZE.ESTIMATE                    629
    PAGE 239  ROUTINE TARGET.ANALYSIS                  971  997
    PAGE 470  PROCESS TARGET.REPORT                   1540
    PAGE 471                                          1563
    PAGE 473                                          1709
    PAGE 632  FUNCTION EST.RANGE                      8750
TARGET.EQUIP
    PAGE 487  PROCESS ASSESSMENT                      2414 2462 2468
    PAGE 488                                          2469 2483
    PAGE 489                                          2569 2582 2583
    PAGE 490                                          2597 2614 2620 2621
    PAGE 491                                          2659 2668 2674 2675 2686 2687 2690
    PAGE 492                                          2706 2716 2721 2727 2728 2737
    PAGE 493  PROCESS SHOOT.OUT                       2761
    PAGE 494                                          2826 2846
    PAGE 497                                          3012 3021 3037
    PAGE 498                                          3077 3082
    PAGE 499                                          3118 3121 3147 3148
```

```
    PAGE 500                                    3164 3180 3181 3199 3200
    PAGE 501                                    3213 3221 3222 3233 3249 3250
    PAGE 502                                    3276 3302 3303 3319
TARGET.EQUIPMENT
    PAGE 377  EVENT HELO.ENGAGEMENT             7089 7090 7094 7098 7099 7100
TARGET.HELICOPTER
    PAGE 379  EVENT HELO.ENGAGEMENT             7204
TARGET.OF.OPPORT
    PAGE 238  ROUTINE SWITCH.FO                  922  963
TARGET.OF.OPPORTUNITY
    PAGE 254  ROUTINE FO.DETECTION              1684 1718
    PAGE 438  PROCESS FORWARD.OBSERVER          9873 9891
TARGET.REPORT
    PAGE   3  PROGRAM REVISIONS                  149
    PAGE   4                                     208
    PAGE  35  ''SECTION FOR PROCESSES           1995
    PAGE  38                                    2147
    PAGE 160  ROUTINE ATTRIT.SENSOR             7386
    PAGE 182  ROUTINE FA.BN.ASGN                8319
    PAGE 217  ROUTINE REQUEST.FASCAM            9957
    PAGE 220  ROUTINE REQUEST.ILLUM              136
    PAGE 225  ROUTINE REQUEST.SMOKE              354
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS    2769
    PAGE 332  ROUTINE FDC.TR.DEQ                5354 5355
    PAGE 334  ROUTINE FINISH.COMPUTATION        5384
    PAGE 362  EVENT CFR.OPERATOR                6503
    PAGE 363                                    6580 6582
    PAGE 387  EVENT PDB.OPERATOR                7522 7558 7560
    PAGE 431  PROCESS AIR.OBSERVER              9553 9558
    PAGE 432                                    9581 9585
    PAGE 440  PROCESS FORWARD.OBSERVER             1
    PAGE 441                                     48   63
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE      1388
    PAGE 468                                    1457 1462
    PAGE 470  PROCESS TARGET.REPORT             1482 1526
    PAGE 475                                    1782
    PAGE 476                                    1832 1858
    PAGE 486  PROCESS FIRE.MISSION              2394
    PAGE 600  ROUTINE BETWEEN.ROUTINE           7516 7517
    PAGE 616  ROUTINE SNAP2                     8159
    PAGE 619  ROUTINE SNAP.R                    8280
    PAGE 636  FUNCTION HE.WLA                   8894
    PAGE 653  PROCESS AIRBORNE.RADAR            9306
    PAGE 654                                    9370
    PAGE 659  PROCESS PHOTO.IR.FLIGHT           9556 9585 9586
    PAGE 694  ''PROGRAM  OLDER.VERSION          1433
    PAGE 697                                    1585
TARGET.UNIT
    PAGE 149  ROUTINE PK.COMPUTE                6899 6911 6914 6914 6922 6926 6930 6931
    PAGE 487  PROCESS ASSESSMENT                2415 2432 2432 2433 2442
    PAGE 488                                    2484 2485 2487
    PAGE 489                                    2568
    PAGE 490                                    2595 2604 2604
    PAGE 491                                    2657
    PAGE 492                                    2714 2736 2738 2741 2744
    PAGE 493  PROCESS SHOOT.OUT                 2762
```

```
    PAGE 494                                          2827 2838 2839
    PAGE 497                                          3013 3014 3015
    PAGE 498                                          3058
    PAGE 500                                          3165
    PAGE 501                                          3241
    PAGE 502                                          3277 3320
    PAGE 503                                          3336
TAR.REP
    PAGE 475  PROCESS TARGET.REPORT                   1775
TBF.BL.ARMOR.UNITS
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES         861
    PAGE 397  EVENT START.BATTLE                      7869 7888
    PAGE 569  ROUTINE TBF.INPUT                       6226 6240
    PAGE 675  ''PROGRAM  OLDER.VERSION                 303
TBF.BL.HQ.UNITS
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES         867
    PAGE 397  EVENT START.BATTLE                      7875 7894
    PAGE 569  ROUTINE TBF.INPUT                       6229 6246
    PAGE 675  ''PROGRAM  OLDER.VERSION                 309
TBF.BL.INF.UNITS
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES         865
    PAGE 397  EVENT START.BATTLE                      7873 7892
    PAGE 569  ROUTINE TBF.INPUT                       6228 6244
    PAGE 675  ''PROGRAM  OLDER.VERSION                 307
TBF.BL.MECH.UNITS
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES         863
    PAGE 397  EVENT START.BATTLE                      7871 7890
    PAGE 569  ROUTINE TBF.INPUT                       6227 6242
    PAGE 675  ''PROGRAM  OLDER.VERSION                 305
TBF.BL.MISSION
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES         857
    PAGE  67  ROUTINE GENERAL.BATTLE                  3474 3475 3476
    PAGE  72  ROUTINE ORIENTATION                     3698 3699 3700
    PAGE  73                                          3753 3754 3755
    PAGE 397  EVENT START.BATTLE                      7866 7885
    PAGE 569  ROUTINE TBF.INPUT                       6225
    PAGE 675  ''PROGRAM  OLDER.VERSION                 299
TBF.INPUT
    PAGE 521  ROUTINE MAIN2                           4266 4268
    PAGE 569  ROUTINE TBF.INPUT                       6202
TBF.NO.BL.UNITS
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES         855
    PAGE 397  EVENT START.BATTLE                      7864 7883
    PAGE 569  ROUTINE TBF.INPUT                       6224
    PAGE 675  ''PROGRAM  OLDER.VERSION                 297
TBF.NO.RD.UNITS
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES         856
    PAGE 397  EVENT START.BATTLE                      7865 7884
    PAGE 569  ROUTINE TBF.INPUT                       6230
    PAGE 675  ''PROGRAM  OLDER.VERSION                 298
TBF.RD.ARMOR.UNITS
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES         860
    PAGE 397  EVENT START.BATTLE                      7868 7887
    PAGE 569  ROUTINE TBF.INPUT                       6232 6248
    PAGE 675  ''PROGRAM  OLDER.VERSION                 302
```

```
TBF.RD.HQ.UNITS
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES          866
    PAGE 397  EVENT START.BATTLE                       7874 7893
    PAGE 569  ROUTINE TBF.INPUT                        6235 6254
    PAGE 675  ''PROGRAM  OLDER.VERSION                  308
TBF.RD.INF.UNITS
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES          864
    PAGE 397  EVENT START.BATTLE                       7872 7891
    PAGE 569  ROUTINE TBF.INPUT                        6234 6252
    PAGE 675  ''PROGRAM  OLDER.VERSION                  306
TBF.RD.MECH.UNITS
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES          862
    PAGE 397  EVENT START.BATTLE                       7870 7889
    PAGE 569  ROUTINE TBF.INPUT                        6233 6250
    PAGE 675  ''PROGRAM  OLDER.VERSION                  304
TBF.RD.MISSION
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES          858
    PAGE  67  ROUTINE GENERAL.BATTLE                   3478 3479 3480
    PAGE  68                                           3508 3509 3510
    PAGE  72  ROUTINE ORIENTATION                      3708 3709 3710
    PAGE  73                                           3756 3757 3758 3770 3771 3772
    PAGE 397  EVENT START.BATTLE                       7867 7886
    PAGE 569  ROUTINE TBF.INPUT                        6231
    PAGE 675  ''PROGRAM  OLDER.VERSION                  300
TBF.WIDTH
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES          859
    PAGE  69  ROUTINE GENERAL.BATTLE                   3586 3587 3589 3591 3593
    PAGE 397  EVENT START.BATTLE                       7903
    PAGE 569  ROUTINE TBF.INPUT                        6237 6238 6238
    PAGE 675  ''PROGRAM  OLDER.VERSION                  301
TB.HOW.EQ.ID
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES          881
    PAGE  45  ''SECTION FOR DEFINITIONS                2581
    PAGE 252  ROUTINE MINE.EFFECTS                     1610
    PAGE 260  ROUTINE BTRY.EFFECTS                     2001
    PAGE 263                                           2140
    PAGE 270                                           2562
    PAGE 271                                           2604 2650
    PAGE 272                                           2661
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                  2956
    PAGE 282  ROUTINE AC.DF.EFFECTS                    3138
    PAGE 362  EVENT CFR.OPERATOR                       6525
    PAGE 384  EVENT OFF.LINE.ATTRITION                 7438
    PAGE 387  EVENT PDB.OPERATOR                       7542
    PAGE 435  PROCESS ARTY.ASSESS                      9698
    PAGE 437                                           9842
    PAGE 463  PROCESS HOW.REPAIR                       1197
    PAGE 548  ROUTINE TB.INPUT                         5336
    PAGE 551  ROUTINE BTRY.INPUT                       5463
    PAGE 592  ROUTINE AMMO.RPT                         7138 7140 7144
    PAGE 593                                           7186 7189 7203 7206
    PAGE 598  ROUTINE ANALYSIS.OUTPUT                  7415
    PAGE 611  ROUTINE OUTPUT.ATTRITION                 7950
    PAGE 675  ''PROGRAM  OLDER.VERSION                  323
    PAGE 704                                           2002
```

```
TB.INPUT
    PAGE 520  ROUTINE MAIN2                          4227 4229
    PAGE 548  ROUTINE TB.INPUT                       5307
TB.LFAIL.MEAN.RNDS
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        888
    PAGE  45  ''SECTION FOR DEFINITIONS              2585
    PAGE 463  PROCESS HOW.REPAIR                     1209
    PAGE 485  PROCESS FIRE.MISSION                   2335
    PAGE 548  ROUTINE TB.INPUT                       5343
    PAGE 551  ROUTINE BTRY.INPUT                     5499 5502
    PAGE 675  ''PROGRAM  OLDER.VERSION                330
    PAGE 704                                         2006
TB.LFAIL.REPAIR
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        890
    PAGE  45  ''SECTION FOR DEFINITIONS              2587
    PAGE 463  PROCESS HOW.REPAIR                     1204
    PAGE 548  ROUTINE TB.INPUT                       5345
    PAGE 675  ''PROGRAM  OLDER.VERSION                332
    PAGE 704                                         2008
TB.MARCH.ORDER
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        896
    PAGE  90  ROUTINE FA.BN.MOVEMENT                 4466 4486
    PAGE  91                                         4545 4559
    PAGE  92                                         4576
    PAGE 548  ROUTINE TB.INPUT                       5351
    PAGE 675  ''PROGRAM  OLDER.VERSION                338
TB.MAX.FEBA
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        895
    PAGE  45  ''SECTION FOR DEFINITIONS              2592
    PAGE  90  ROUTINE FA.BN.MOVEMENT                 4464 4482 4483 4484
    PAGE  91                                         4543 4555 4556 4557
    PAGE  92                                         4569
    PAGE 548  ROUTINE TB.INPUT                       5350 5357 5357
    PAGE 675  ''PROGRAM  OLDER.VERSION                337
    PAGE 704                                         2013
TB.MAX.PREP
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        893
    PAGE  45  ''SECTION FOR DEFINITIONS              2590
    PAGE 473  PROCESS TARGET.REPORT                  1671
    PAGE 482  PROCESS FIRE.MISSION                   2150
    PAGE 548  ROUTINE TB.INPUT                       5348
    PAGE 675  ''PROGRAM  OLDER.VERSION                335
    PAGE 704                                         2011
TB.MAX.RANGE
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        885
    PAGE  45  ''SECTION FOR DEFINITIONS              2583
    PAGE 184  ROUTINE FA.BN.ASGN                     8425
    PAGE 208  ROUTINE PGM.MSN.ASGN                   9530
    PAGE 358  EVENT CFR.ACTIVATION                   6351
    PAGE 386  EVENT PDB.ACTIVATION                   7478
    PAGE 548  ROUTINE TB.INPUT                       5340 5355 5355
    PAGE 675  ''PROGRAM  OLDER.VERSION                327
    PAGE 704                                         2004
TB.MAX.RAP.RANGE
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        886
    PAGE 184  ROUTINE FA.BN.ASGN                     8429
```

```
    PAGE 548  ROUTINE TB.INPUT                          5341 5356 5356
    PAGE 675  ''PROGRAM  OLDER.VERSION                   328
TB.MIN.FEBA
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           894
    PAGE  45  ''SECTION FOR DEFINITIONS                 2591
    PAGE  90  ROUTINE FA.BN.MOVEMENT                    4462 4463 4464 4484
    PAGE  91                                            4541 4542 4543 4557
    PAGE  92                                            4570
    PAGE 548  ROUTINE TB.INPUT                          5349 5358 5358
    PAGE 675  ''PROGRAM  OLDER.VERSION                   336
    PAGE 704                                            2012
TB.MIN.HOW
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           883
    PAGE 182  ROUTINE FA.BN.ASGN                        8334 8336 8337
    PAGE 208  ROUTINE PGM.MSN.ASGN                      9517
    PAGE 252  ROUTINE MINE.EFFECTS                      1626
    PAGE 271  ROUTINE BTRY.EFFECTS                      2619
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                   2964
    PAGE 282  ROUTINE AC.DF.EFFECTS                     3143
    PAGE 463  PROCESS HOW.REPAIR                        1182
    PAGE 548  ROUTINE TB.INPUT                          5338
    PAGE 675  ''PROGRAM  OLDER.VERSION                   325
TB.MIN.PREP
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           892
    PAGE  45  ''SECTION FOR DEFINITIONS                 2589
    PAGE 482  PROCESS FIRE.MISSION                      2149
    PAGE 548  ROUTINE TB.INPUT                          5347
    PAGE 675  ''PROGRAM  OLDER.VERSION                   334
    PAGE 704                                            2010
TB.MN.FASCAM.SUPP
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           899
    PAGE 273  ROUTINE BTRY.EFFECTS                      2718
    PAGE 548  ROUTINE TB.INPUT                          5353
    PAGE 675  ''PROGRAM  OLDER.VERSION                   341
TB.MW.THRESHOLD
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           909
    PAGE 183  ROUTINE FA.BN.ASGN                        8395
    PAGE 208  ROUTINE PGM.MSN.ASGN                      9528
    PAGE 549  ROUTINE TB.INPUT                          5381
    PAGE 676  ''PROGRAM  OLDER.VERSION                   351
TB.MX.FASCAM.SUPP
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           900
    PAGE 272  ROUTINE BTRY.EFFECTS                      2717
    PAGE 548  ROUTINE TB.INPUT                          5354
    PAGE 675  ''PROGRAM  OLDER.VERSION                   342
TB.NAME
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           880
    PAGE  47  ''SECTION FOR DEFINITIONS                 2693
    PAGE 263  ROUTINE BTRY.EFFECTS                      2139
    PAGE 271                                            2651
    PAGE 272                                            2662 2692
    PAGE 481  PROCESS FIRE.MISSION                      2135
    PAGE 484                                            2290
    PAGE 548  ROUTINE TB.INPUT                          5325
    PAGE 592  ROUTINE AMMO.RPT                          7138 7140
    PAGE 593                                            7185 7186 7189 7202 7203 7206
```

```
    PAGE 598  ROUTINE ANALYSIS.OUTPUT                7415
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7983
    PAGE 675  ''PROGRAM  OLDER.VERSION                322
    PAGE 706                                         2117
TB.N.FM
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES        252
    PAGE  16                                          908
    PAGE 548  ROUTINE TB.INPUT                       5317
    PAGE 549                                         5379 5381
    PAGE 664  ''PROGRAM  OLDER.VERSION               9694
    PAGE 675                                          350
TB.N.HOW
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                7341 7344 7347 7350 7353
    PAGE 597                                         7404 7407
TB.OCCUPY
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        897
    PAGE 407  EVENT STOP.ARTY.MOVEMENT               8400
    PAGE 548  ROUTINE TB.INPUT                       5352
    PAGE 675  ''PROGRAM  OLDER.VERSION                339
TB.RND.PER.LAUNCH
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        882
    PAGE 178  ROUTINE EST.COVERAGE                   8173 8177
    PAGE 192  ROUTINE FINAL.COVERAGE                 8809 8819
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9323
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS               1415
    PAGE 260  ROUTINE BTRY.EFFECTS                   2008
    PAGE 262                                         2110
    PAGE 263                                         2143
    PAGE 483  PROCESS FIRE.MISSION                   2244 2249
    PAGE 548  ROUTINE TB.INPUT                       5337
    PAGE 551  ROUTINE BTRY.INPUT                     5466 5485
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7969
    PAGE 675  ''PROGRAM  OLDER.VERSION                324
TB.SEQ.NO
    PAGE 548  ROUTINE TB.INPUT                       5324
TB.SFAIL.MEAN.RNDS
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        887
    PAGE  45  ''SECTION FOR DEFINITIONS              2584
    PAGE 463  PROCESS HOW.REPAIR                     1194
    PAGE 485  PROCESS FIRE.MISSION                   2332
    PAGE 548  ROUTINE TB.INPUT                       5342
    PAGE 551  ROUTINE BTRY.INPUT                     5493 5496
    PAGE 675  ''PROGRAM  OLDER.VERSION                329
    PAGE 704                                         2005
TB.SFAIL.REPAIR
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        889
    PAGE  45  ''SECTION FOR DEFINITIONS              2586
    PAGE 463  PROCESS HOW.REPAIR                     1190
    PAGE 548  ROUTINE TB.INPUT                       5344
    PAGE 675  ''PROGRAM  OLDER.VERSION                331
    PAGE 704                                         2007
TB.SHOOT.SCOOT.IND
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES        898
    PAGE  89  ROUTINE FA.BN.MOVEMENT                 4429
    PAGE 355  EVENT ARTY.OCCUPATION                  6249
    PAGE 486  PROCESS FIRE.MISSION                   2398
```

```
    PAGE 548  ROUTINE TB.INPUT                          5326
    PAGE 675  ''PROGRAM  OLDER.VERSION                   340
TB.SORT.LIST
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           870
    PAGE  18                                            1013
    PAGE  49  ''SECTION FOR DEFINITIONS                 2825
    PAGE  68  ROUTINE GENERAL.BATTLE                    3520 3525 3552 3557
    PAGE 675  ''PROGRAM  OLDER.VERSION                   312
    PAGE 677                                             455
    PAGE 708                                            2246
TB.SUPPRESS.TIME
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           891
    PAGE  45  ''SECTION FOR DEFINITIONS                 2588
    PAGE 182  ROUTINE FA.BN.ASGN                        8342
    PAGE 183                                            8346 8349
    PAGE 208  ROUTINE PGM.MSN.ASGN                      9519
    PAGE 270  ROUTINE BTRY.EFFECTS                      2588
    PAGE 481  PROCESS FIRE.MISSION                      2129
    PAGE 548  ROUTINE TB.INPUT                          5346
    PAGE 675  ''PROGRAM  OLDER.VERSION                   333
    PAGE 704                                            2009
TB.SUST.FIRE.RATE
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES           884
    PAGE  45  ''SECTION FOR DEFINITIONS                 2582
    PAGE 473  PROCESS TARGET.REPORT                     1680
    PAGE 480  PROCESS FIRE.MISSION                      2080 2082
    PAGE 481                                            2086 2088 2095 2096 2098
    PAGE 548  ROUTINE TB.INPUT                          5339
    PAGE 629  FUNCTION BTRY.AVAILABLE                   8663 8667
    PAGE 675  ''PROGRAM  OLDER.VERSION                   326
    PAGE 704                                            2003
TB.TM
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1751
    PAGE 181  ROUTINE FASCAM.COMPUTATION                8260 8263
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION             9032
    PAGE 199  ROUTINE ILLUM.COMPUTATION                 9151 9154
    PAGE 233  ROUTINE SMOKE.COMPUTATION                  709  712
    PAGE 272  ROUTINE BTRY.EFFECTS                      2712
    PAGE 483  PROCESS FIRE.MISSION                      2219 2231
    PAGE 549  ROUTINE TB.INPUT                          5367
    PAGE 555  ROUTINE MUNS.INPUT                        5659
    PAGE 556                                            5689
    PAGE 592  ROUTINE AMMO.RPT                          7152 7153 7156 7157 7160 7161 7164 7165 7168 7169
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                   7341 7344 7347 7350 7353
    PAGE 598                                            7419
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1190
TB.TM.CLASS
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1750
    PAGE  47  ''SECTION FOR DEFINITIONS                 2694
    PAGE 181  ROUTINE FASCAM.COMPUTATION                8259
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION             9030 9031 9033
    PAGE 199  ROUTINE ILLUM.COMPUTATION                 9150
    PAGE 233  ROUTINE SMOKE.COMPUTATION                  708
    PAGE 272  ROUTINE BTRY.EFFECTS                      2711
    PAGE 483  PROCESS FIRE.MISSION                      2218 2230
    PAGE 549  ROUTINE TB.INPUT                          5366
```

```
    PAGE 555  ROUTINE MUNS.INPUT                    5619 5627
    PAGE 556                                        5660 5690
    PAGE 592  ROUTINE AMMO.RPT                      7142 7143 7151 7155 7159 7163 7167
    PAGE 593                                        7185 7202
    PAGE 596  ROUTINE ANALYSIS.OUTPUT               7340 7343 7346 7349 7352
    PAGE 598                                        7419
    PAGE 690  ''PROGRAM  OLDER.VERSION              1189
    PAGE 706                                        2118
TB.TM.FIRED
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES      1752
    PAGE 483  PROCESS FIRE.MISSION                  2245 2250
    PAGE 593  ROUTINE AMMO.RPT                      7177 7178 7182
    PAGE 690  ''PROGRAM  OLDER.VERSION              1191
TB.TM.LINK
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES      1748
    PAGE 548  ROUTINE TB.INPUT                      5364
    PAGE 549                                        5365 5366 5367 5368 5369 5372
    PAGE 555  ROUTINE MUNS.INPUT                    5618 5619 5626 5627 5658 5659
    PAGE 556                                        5660 5688 5689 5690
    PAGE 690  ''PROGRAM  OLDER.VERSION              1187
TB.TM.LIST
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES       902
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES      1754
    PAGE  49  ''SECTION FOR DEFINITIONS             2826
    PAGE 181  ROUTINE FASCAM.COMPUTATION            8258
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION         9027
    PAGE 199  ROUTINE ILLUM.COMPUTATION             9149
    PAGE 233  ROUTINE SMOKE.COMPUTATION              707
    PAGE 272  ROUTINE BTRY.EFFECTS                  2710
    PAGE 483  PROCESS FIRE.MISSION                  2217 2229
    PAGE 549  ROUTINE TB.INPUT                      5372
    PAGE 555  ROUTINE MUNS.INPUT                    5618 5626 5658
    PAGE 556                                        5688
    PAGE 592  ROUTINE AMMO.RPT                      7136
    PAGE 596  ROUTINE ANALYSIS.OUTPUT               7338
    PAGE 598                                        7417
    PAGE 675  ''PROGRAM  OLDER.VERSION               344
    PAGE 690                                        1193
    PAGE 708                                        2247
TB.TM.RAP
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES      1749
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION         9029
    PAGE 549  ROUTINE TB.INPUT                      5365 5368
    PAGE 690  ''PROGRAM  OLDER.VERSION              1188
TB.WPN.NAME
    PAGE  47  ''SECTION FOR DEFINITIONS             2695
    PAGE 706  ''PROGRAM  OLDER.VERSION              2119
TEAM.
    PAGE 300  ROUTINE HC.DISENGAGE                  4012
    PAGE 304  ROUTINE REPLACE.HC                    4194
    PAGE 446  PROCESS HC.ARRIVE.BATTLE               284
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION         702
TEAM.SIDE
    PAGE 450  PROCESS HC.RETURN.FARRP                475  504  511
    PAGE 452                                         614
    PAGE 453                                         651
```

```
TEAM.TYPES
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES          869
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES         1799
    PAGE  60  ROUTINE CREATE.TEAMS                     3169
    PAGE  67  ROUTINE GENERAL.BATTLE                   3461 3462 3471
    PAGE  68                                           3532
    PAGE  69                                           3564 3596
    PAGE  72  ROUTINE ORIENTATION                      3695
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                  3839 3841
    PAGE 398  EVENT START.BATTLE                       7918
    PAGE 675  ''PROGRAM  OLDER.VERSION                  311
    PAGE 691                                           1238
TEMP.POINT
    PAGE 322  ROUTINE COMPUTE.WD                       4969 4986 4989 4990 4992 4993
TERRAIN
    PAGE   1  ROUTINE FOR CROSS_REFERENCING              53
    PAGE  41  ''SECTION FOR EVENTS                     2312
    PAGE 115  ROUTINE PROX.CHECK                       5504 5506 5511
    PAGE 146  ROUTINE INTER.BATTLE                     6795
    PAGE 242  ROUTINE UNIT.ENVIR                       1132 1163
    PAGE 243                                           1180 1187
    PAGE 336  ROUTINE GET.TERRAIN                      5444
    PAGE 395  EVENT START.BATTLE                       7745 7800
    PAGE 400                                           8074
    PAGE 403  EVENT START.MOVE                         8188
    PAGE 700  ''PROGRAM  OLDER.VERSION                 1749
TERRAIN.PAR
    PAGE  42  ''SECTION FOR DEFINITIONS                2405
    PAGE  52  ''SECTION FOR SUBSTITUTIONS              2966
    PAGE 336  ROUTINE GET.TERRAIN                      5442 5445 5448 5451
    PAGE 523  ROUTINE SYS.INPUT                        4361
    PAGE 701  ''PROGRAM  OLDER.VERSION                 1835
    PAGE 710                                           2373
TERRAIN.TYPE
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES          836  851
    PAGE  21  ''SECTION FOR TEMPORARY_ENTITIES         1156
    PAGE 571  ROUTINE TT.FACTORS.INPUT                 6290 6298 6303 6306 6308 6309 6310 6311 6313 6315 6317 6319 6321 6322
    PAGE 618  ROUTINE SNAP.R                           8246
    PAGE 674  ''PROGRAM  OLDER.VERSION                  278
    PAGE 675                                            293
    PAGE 680                                            595
    PAGE 712                                           2470
    PAGE
    PAGE 242  ROUTINE UNIT.ENVIR                       1135
    PAGE 243                                           1172 1173 1174 1175 1176 1180 1181 1182 1183 1187 1191 1192 1195
    PAGE 336  ROUTINE GET.TERRAIN                      5443 5446 5449 5451
    PAGE 644  EVENT START ARTY MOVEMENT                7720 7731
    PAGE 647  EVENT START MOVE                         8215

              ROUTINE GET.TERRAIN                      5435
              EVENT START MOVE                         8217
              EVENT ACT ATK                            8702 8705

                  N FOR DEFINITIONS                    2431
                  UNIT ENVIR                           1165
                  FACTORS INPUT                        6287
```

```
    PAGE 702  ''PROGRAM  OLDER.VERSION                1861
TER.W.INC.
    PAGE  97  ROUTINE LINE.OF.SIGHT                   4740 4759
    PAGE 109  ROUTINE NEW.SEGMENT                     5258
    PAGE 210  ROUTINE PGM.MSN.ASGN                    9619
    PAGE 479  PROCESS WITH.DRAW                       1985 2008
TES.LA.EQUIP
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         923
    PAGE 190  ROUTINE FINAL.COVERAGE                  8709
    PAGE 265  ROUTINE BTRY.EFFECTS                    2264
    PAGE 557  ROUTINE SUBM.INPUT                      5745
    PAGE 638  FUNCTION ICM.WLA                        9006
    PAGE 676  ''PROGRAM  OLDER.VERSION                 365
TE.DELTA.T
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         917
    PAGE  45  ''SECTION FOR DEFINITIONS               2593
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION         5761
    PAGE 530  ROUTINE EQ.TE.INPUT                     4590
    PAGE 676  ''PROGRAM  OLDER.VERSION                 359
    PAGE 704                                          2014
TE.HEIGHT
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         918
    PAGE  45  ''SECTION FOR DEFINITIONS               2594
    PAGE 340  ROUTINE PROB.INF                        5622
    PAGE 530  ROUTINE EQ.TE.INPUT                     4591
    PAGE 676  ''PROGRAM  OLDER.VERSION                 360
    PAGE 704                                          2015
TE.LINK
    PAGE 183  ROUTINE FA.BN.ASGN                      8366 8367 8384 8387 8388 8388
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9534 9535
    PAGE 209                                          9552
    PAGE 480  PROCESS FIRE.MISSION                    2058 2059 2078
    PAGE 481                                          2085 2092
TE.MAX.MF.LOSS
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         916
    PAGE 251  ROUTINE MINE.EFFECTS                    1562
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                 2919
    PAGE 530  ROUTINE EQ.TE.INPUT                     4587
    PAGE 676  ''PROGRAM  OLDER.VERSION                 358
TE.MIN.MF.LOSS
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         915
    PAGE 251  ROUTINE MINE.EFFECTS                    1561
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                 2918
    PAGE 530  ROUTINE EQ.TE.INPUT                     4586
    PAGE 676  ''PROGRAM  OLDER.VERSION                 357
TE.NAME
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES         912
    PAGE  47  ''SECTION FOR DEFINITIONS               2696
    PAGE 149  ROUTINE PK.COMPUTE                      6944 6945
    PAGE 183  ROUTINE FA.BN.ASGN                      8388
    PAGE 250  ROUTINE MINE.EFFECTS                    1498
    PAGE 264  ROUTINE BTRY.EFFECTS                    2243
    PAGE 271                                          2649
    PAGE 272                                          2660 2692
    PAGE 530  ROUTINE EQ.TE.INPUT                     4583
    PAGE 538  ROUTINE MFO.INPUT                       4971
```

```
    PAGE 557  ROUTINE SUBM.INPUT                    5743
    PAGE 558  ROUTINE HE.LA.INPUT                   5793
    PAGE 582  ROUTINE TACAIR.INPUT                  6779
    PAGE 588  ROUTINE AC.MUNS.INPUT                 7042
    PAGE 595  ROUTINE ANALYSIS.OUTPUT               7296
    PAGE 621  ROUTINE TACAIR.DATA.REPORT            8385
    PAGE 656  ROUTINE AR.DETECTION                  9410
    PAGE 676  ''PROGRAM  OLDER.VERSION               354
    PAGE 706                                        2120
TE.PGM.INDIC
    PAGE  16  ''SECTION FOR PERMANENT_ENTITIES       913
    PAGE 157  ROUTINE AO.DETECTION                  7289
    PAGE 211  ROUTINE PGM.MSN.ASGN                  9668 9675
    PAGE 230  ROUTINE RPV.DETECTION                  605
    PAGE 256  ROUTINE FO.DETECTION                  1796 1815
    PAGE 257                                        1898
    PAGE 258                                        1940
    PAGE 264  ROUTINE BTRY.EFFECTS                  2209
    PAGE 267                                        2396 2399
    PAGE 268                                        2457
    PAGE 362  EVENT CFR.OPERATOR                    6526
    PAGE 530  ROUTINE EQ.TE.INPUT                   4584
    PAGE 676  ''PROGRAM  OLDER.VERSION               355
TE.PROJECTED.AREA
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES       914
    PAGE 156  ROUTINE AO.DETECTION                  7229 7233 7236
    PAGE 280  ROUTINE AC.DF.EFFECTS                 3021
    PAGE 429  PROCESS AIR.OBSERVER                  9414
    PAGE 530  ROUTINE EQ.TE.INPUT                   4585
    PAGE 537  ROUTINE UNIT.INPUT                    4921
    PAGE 676  ''PROGRAM  OLDER.VERSION               356
TE.SET
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES       417
    PAGE  17                                         920
    PAGE 530  ROUTINE EQ.TE.INPUT                   4612
    PAGE 667  ''PROGRAM  OLDER.VERSION              9858
    PAGE 676                                         362
TF.LDR
    PAGE 125  ROUTINE WHAT.NEXT                     5894 5895 5895 5897 5904 5914 5916
    PAGE 126                                        5970
    PAGE 142  ROUTINE DEAD.UNIT                     6548 6549 6552 6554 6557
    PAGE 143                                        6580 6581 6586 6593 6611 6622 6623
    PAGE 147  ROUTINE INTER.BATTLE                  6803 6807 6811 6815 6832 6836
    PAGE 148                                        6869 6873
TGT.BTRY
    PAGE 270  ROUTINE BTRY.EFFECTS                  2585 2586 2588 2589 2590
    PAGE 271                                        2602 2604 2607 2614 2618 2619
    PAGE 384  EVENT OFF.LINE.ATTRITION              7437 7438 7440 7445 7453
TGT.ON.MOVE
    PAGE 149  ROUTINE PK.COMPUTE                    6924 6927
    PAGE 151                                        7009
TGT.OTM
    PAGE  50  ''SECTION FOR DEFINITIONS             2837
    PAGE 151  ROUTINE PK.COMPUTE                    7010
    PAGE 526  ROUTINE PK.INPUT                      4440 4442
    PAGE 706  ''PROGRAM  OLDER.VERSION              2258
```

```
TGT.RADIUS
    PAGE 343  ROUTINE SEARCH.COVERAGE                5682 5704 5705 5706 5710 5713 5719 5721 5725
TGT.RAD.SQ
    PAGE 343  ROUTINE SEARCH.COVERAGE                5704 5705 5705 5706 5706 5707
TGT.UNIT
    PAGE 242  ROUTINE UNIT.ENVIR                     1113 1146 1151 1152 1153 1159 1164
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION          878
    PAGE 458                                          963  980
TGW.ATK.CRITERIA
    PAGE  43  ''SECTION FOR DEFINITIONS              2435
TGW.AVG.PK
    PAGE  49  ''SECTION FOR DEFINITIONS              2783
TGW.FIRED.RNDS
    PAGE  43  ''SECTION FOR DEFINITIONS              2439
TGW.MARK
    PAGE  47  ''SECTION FOR DEFINITIONS              2706
TGW.MIN.RANGE
    PAGE  43  ''SECTION FOR DEFINITIONS              2437
TGW.SDM.RANGE
    PAGE  43  ''SECTION FOR DEFINITIONS              2434
TGW.THRESHOLD
    PAGE  43  ''SECTION FOR DEFINITIONS              2436
TGW.TM
    PAGE  43  ''SECTION FOR DEFINITIONS              2438
TGW.TOT.KILLS
    PAGE  43  ''SECTION FOR DEFINITIONS              2440
THEM.
    PAGE  65  ROUTINE FILE.KAD.SENSOR                3388
    PAGE 314  ROUTINE FLIGHT.PATH                    4647
THERE.
    PAGE 349  EVENT AD.ENGAGEMENT                    5922
THETA.FOUR
    PAGE  71  ROUTINE ORIENTATION                    3647
    PAGE  73                                         3806 3807
    PAGE  74                                         3816 3817
THETA.ONE
    PAGE  71  ROUTINE ORIENTATION                    3647
    PAGE  72                                         3744
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                3834 3862 3863
THETA.ONE.
    PAGE  73  ROUTINE ORIENTATION                    3751
THETA.THREE
    PAGE  71  ROUTINE ORIENTATION                    3647
    PAGE  73                                         3789 3791
THETA.TWO
    PAGE  71  ROUTINE ORIENTATION                    3647
    PAGE  73                                         3750
THETA.TWO.
    PAGE  73  ROUTINE ORIENTATION                    3751
THIS.DET
    PAGE 167  ROUTINE CFR.DETECTION                  7688 7689 7690 7691 7692 7693 7695
THIS.UNIT
    PAGE 206  ROUTINE PDB.DETECTION                  9454 9455 9456 9457 9458 9459 9461
    PAGE 362  EVENT CFR.OPERATOR                     6480 6484 6497 6501 6504 6510
    PAGE 363                                         6558 6560
    PAGE 387  EVENT PDB.OPERATOR                     7507 7518 7521 7523 7529 7544 7548
```

```
THRESH.REIN
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES          1597
    PAGE  44  ''SECTION FOR DEFINITIONS                 2497
    PAGE 117  ROUTINE PROX.POS                          5582
    PAGE 146  ROUTINE INTER.BATTLE                      6759
    PAGE 546  ROUTINE ORD.MOVCOR                        5238
    PAGE 687  ''PROGRAM  OLDER.VERSION                  1036
    PAGE 703                                            1918
THRU.
    PAGE 315  ROUTINE FLIGHT.PATH                       4726
TIME.
    PAGE 310  ROUTINE AD.SHOOT                          4500
    PAGE 334  ROUTINE FINISH.COMPUTATION                5390
    PAGE 585  ROUTINE TACAIR.INPUT                      6949 6955
TIME.A
    PAGE 104  ROUTINE MINE.DELAY                        5037
    PAGE 105                                            5051
    PAGE 108  ROUTINE NEW.SEGMENT                       5188
    PAGE 159  ROUTINE ATTRIT.SENSOR                     7366
    PAGE 160                                            7407
    PAGE 238  ROUTINE SWITCH.FO                          946  952
    PAGE 301  ROUTINE HC.DISENGAGE                      4079 4104
    PAGE 311  ROUTINE INTER.HELO                        4551
    PAGE 312                                            4622
    PAGE 328  ROUTINE EMPTY                             5203 5212
    PAGE 329                                            5262
    PAGE 337  ROUTINE HC.EMPTY                          5512
    PAGE 338                                            5526
    PAGE 383  EVENT OFF.LINE.ATTRITION                  7382
    PAGE 436  PROCESS ARTY.ASSESS                       9757
    PAGE 437                                            9805
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                   135
    PAGE 465  PROCESS MINE.ASSESS                       1301
    PAGE 485  PROCESS FIRE.MISSION                      2349 2366
    PAGE 491  PROCESS ASSESSMENT                        2698
    PAGE 623  ROUTINE TACAIR.DATA.REPORT                8449
TIME.BETWEEN.ARTY.MOVE
    PAGE  48  ''SECTION FOR DEFINITIONS                 2751
    PAGE  57  ROUTINE MAIN3                             3077
    PAGE  89  ROUTINE FA.BN.MOVEMENT                    4432
    PAGE 252  ROUTINE MINE.EFFECTS                      1628
    PAGE 270  ROUTINE BTRY.EFFECTS                      2599
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                   2966
    PAGE 282  ROUTINE AC.DF.EFFECTS                     3146
    PAGE 355  EVENT ARTY.OCCUPATION                     6247
    PAGE 384  EVENT OFF.LINE.ATTRITION                  7452
    PAGE 390  EVENT SCHEDULE.ARTY.MOVEMENT              7585
    PAGE 523  ROUTINE SYS.INPUT                         4362
    PAGE 707  ''PROGRAM  OLDER.VERSION                  2173
TIME.CPU
    PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT             8806 8816 8817
TIME.EXP
    PAGE 155  ROUTINE AO.DETECTION                      7146
TIME.EXPOSED
    PAGE 154  ROUTINE AO.DETECTION                      7090
    PAGE 155                                            7133 7135 7135 7139 7140 7140 7144 7148 7151
```

```
TIME.INTERVAL
    PAGE 371  EVENT FEBA.SORTIE                    6778 6783 6784 6826 6827
    PAGE 372                                       6868 6869 6872 6873
TIME.SCORE
    PAGE 154  ROUTINE AO.DETECTION                 7090
    PAGE 155                                       7149 7151
    PAGE 156                                       7228
TIME.TO.DETE
    PAGE 123  ROUTINE TIME.TO.DETECT               5812 5849
TIME.V
    PAGE   2  PROGRAM REVISIONS                      72
    PAGE   3                                        122
    PAGE   4                                        179
    PAGE   6  ''SECTION FOR PERMANENT_ENTITIES      331
    PAGE  63  ROUTINE FILE.FD.SCHD                 3270
    PAGE  80  ROUTINE BLOCK.LOS                    4024 4024
    PAGE  83  ROUTINE CHANGE.LOC                   4135 4147 4150 4150
    PAGE  88  ROUTINE END.MOVE                     4389
    PAGE  89  ROUTINE FA.BN.MOVEMENT               4404 4405 4432
    PAGE  90                                       4470 4489
    PAGE  91                                       4548 4562
    PAGE  92                                       4579
    PAGE 106  ROUTINE MIN.MOVE                     5085 5085
    PAGE 107  ROUTINE NEW.SEGMENT                  5145 5151
    PAGE 108                                       5180 5184
    PAGE 113  ROUTINE PREPARE.LIST                 5443
    PAGE 114  ROUTINE PREP.WITHDRAW                5476
    PAGE 115  ROUTINE PROX.CHECK                   5543 5548
    PAGE 133  ROUTINE CHECK.FORCE                  6180 6185
    PAGE 142  ROUTINE DEAD.UNIT                    6544
    PAGE 146  ROUTINE INTER.BATTLE                 6773 6792 6795
    PAGE 150  ROUTINE PK.COMPUTE                   7000 7001
    PAGE 163  ROUTINE BTRY.FM.DEQ                  7533
    PAGE 164  ROUTINE BTRY.FM.ENQ                  7558
    PAGE 165  ROUTINE CFR.DEGRADE                  7597
    PAGE 178  ROUTINE EST.COVERAGE                 8171 8171
    PAGE 181  ROUTINE FASCAM.COMPUTATION           8275
    PAGE 182  ROUTINE FA.BN.ASGN                   8341 8343
    PAGE 183                                       8345 8354
    PAGE 184                                       8416 8440
    PAGE 185                                       8494
    PAGE 194  ROUTINE FIND.START.TIME              8911
    PAGE 199  ROUTINE ILLUM.COMPUTATION            9172
    PAGE 200  ROUTINE ILLUM.EFFECTS                9206
    PAGE 201                                       9266
    PAGE 205  ROUTINE NOISE.DEGRADE                9394 9406
    PAGE 206  ROUTINE PDB.DETECTION                9472
    PAGE 208  ROUTINE PGM.MSN.ASGN                 9518 9520
    PAGE 209                                       9596
    PAGE 217  ROUTINE REQUEST.FASCAM               9968 9969
    PAGE 220  ROUTINE REQUEST.ILLUM                 149  150  154
    PAGE 221                                        162
    PAGE 225  ROUTINE REQUEST.SMOKE                 367  368  372  381
    PAGE 230  ROUTINE RPV.DETECTION                 602
    PAGE 233  ROUTINE SMOKE.COMPUTATION             729
    PAGE 234  ROUTINE SMOKE.EFFECTS                 753  770
```

```
PAGE 235                                          850
PAGE 236                                          878
PAGE 246  ROUTINE VOLLEY                         1356
PAGE 252  ROUTINE MINE.EFFECTS                   1584 1594 1627
PAGE 254  ROUTINE FO.DETECTION                   1692 1696
PAGE 256                                         1818
PAGE 258                                         1944
PAGE 260  ROUTINE BTRY.EFFECTS                   2010
PAGE 266                                         2336
PAGE 270                                         2587 2599
PAGE 271                                         2648 2659
PAGE 272                                         2673 2695 2706
PAGE 273                                         2720
PAGE 276  ROUTINE AC.BOMB.EFFECTS                2795
PAGE 279                                         2965
PAGE 280  ROUTINE AC.DF.EFFECTS                  3006
PAGE 282                                         3145
PAGE 284  ROUTINE CAS.EVAL                       3269
PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS          3289
PAGE 286                                         3394
PAGE 287                                         3396
PAGE 288                                         3466 3467
PAGE 295  ROUTINE END.CAS.MISSION                3800
PAGE 307  ROUTINE AD.SHOOT                       4316
PAGE 309                                         4422 4438
PAGE 310                                         4485
PAGE 327  ROUTINE DQ.CMSN.QUEUE                  5154
PAGE 330  ROUTINE EMPTY                          5292
PAGE 347  EVENT ACT.REINF                        5828
PAGE 349  EVENT AD.ENGAGEMENT                    5910
PAGE 351                                         6045
PAGE 354                                         6218
PAGE 355  EVENT ARTY.OCCUPATION                  6239 6245
PAGE 356  EVENT BTL.ENDED                        6266 6302 6310
PAGE 357                                         6318
PAGE 359  EVENT CFR.OFF                          6401
PAGE 361  EVENT CFR.ON                           6473
PAGE 363  EVENT CFR.OPERATOR                     6558
PAGE 365  EVENT CHANGE.LITE                      6600 6606
PAGE 366  EVENT CHANGE.WEATHER                   6627
PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE              6651
PAGE 368  EVENT END.SIMULATION                   6669
PAGE 369  EVENT ENGAGEMENT                       6688 6694
PAGE 373  EVENT GET.NX.ORD                       6903
PAGE 376  EVENT HELO.ENGAGEMENT                  7051
PAGE 380  EVENT INIT.PREPLAN.CAS                 7251 7254
PAGE 382  EVENT OFF.LINE.ATTRITION               7285 7287 7292 7305 7314 7323 7333
PAGE 384                                         7452
PAGE 385                                         7460
PAGE 387  EVENT PDB.OPERATOR                     7546
PAGE 389  EVENT POSITION.REPORT                  7575
PAGE 392  EVENT SET.DEBUG                        7631 7637
PAGE 393  EVENT START.ARTY.MOVEMENT              7704
PAGE 394                                         7724
PAGE 395  EVENT START.BATTLE                     7786 7796
PAGE 398                                         7941 7942
```

```
PAGE 399                                    7984
PAGE 402                                    8158
PAGE 404  EVENT START.MOVE                  8270
PAGE 407  EVENT STOP.ARTY.MOVEMENT          8389
PAGE 408  EVENT UPDATE.LOC                  8453
PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT     8814 8814
PAGE 419  PROCESS AC.ATK.TGT                8882
PAGE 421                                    8981
PAGE 422                                    9021
PAGE 423                                    9083 9115
PAGE 424                                    9191
PAGE 425                                    9222
PAGE 426                                    9266 9287
PAGE 429  PROCESS AIR.OBSERVER              9409 9440
PAGE 430                                    9459
PAGE 431                                    9507 9515 9516
PAGE 432                                    9586 9587 9594 9598 9600
PAGE 438  PROCESS FORWARD.OBSERVER          9887 9892
PAGE 439                                    9952
PAGE 441                                      46   47   55   59   70   72
PAGE 443  PROCESS HC.ARRIVE.BATTLE          114  150
PAGE 454  PROCESS HEL.TARGET.ACQUISITION    741  743
PAGE 464  PROCESS MINE.ASSESS               1240
PAGE 466                                    1350
PAGE 467  PROCESS REMOTE.PILOT.VEHICLE      1397
PAGE 468                                    1441 1455 1456 1461
PAGE 470  PROCESS TARGET.REPORT             1504 1523 1526
PAGE 472                                    1643 1647
PAGE 473                                    1685
PAGE 475                                    1779 1782 1811
PAGE 476                                    1874
PAGE 478  PROCESS WITH.DRAW                 1935
PAGE 481  PROCESS FIRE.MISSION              2105 2127
PAGE 482                                    2143
PAGE 484                                    2289
PAGE 487  PROCESS ASSESSMENT                2464
PAGE 489                                    2578
PAGE 490                                    2616
PAGE 491                                    2670
PAGE 492                                    2723
PAGE 499  PROCESS SHOOT.OUT                 3107 3126 3143
PAGE 500                                    3176 3195
PAGE 501                                    3217 3245
PAGE 502                                    3298
PAGE 504  PROCESS CAS.MISSION               3367 3372
PAGE 505                                    3448
PAGE 506                                    3475
PAGE 508                                    3595 3601
PAGE 509                                    3644
PAGE 514  PROCESS HELICOPTER.FIRE           3940
PAGE 516                                    4011
PAGE 517                                    4090
PAGE 518                                    4149
PAGE 590  ROUTINE TR.INPUT                  7091 7092
PAGE 599  ROUTINE BETWEEN.ROUTINE           7437 7439 7441 7442
PAGE 604  ROUTINE ERROR.STOP                7670
```

```
    PAGE 606  ROUTINE KV.PRINT                    7725
    PAGE 611  ROUTINE OUTPUT.ATTRITION            7939
    PAGE 613  ROUTINE FOR POSITION.OUT            8001
    PAGE 614  ROUTINE SNAP2                       8028
    PAGE 615                                      8085
    PAGE 618  ROUTINE SNAP.R                      8207
    PAGE 620                                      8317 8317
    PAGE 653  PROCESS AIRBORNE.RADAR              9299 9304
    PAGE 654                                      9344 9367 9368 9369
    PAGE 656  ROUTINE AR.DETECTION                9435
    PAGE 659  PROCESS PHOTO.IR.FLIGHT             9568 9569 9583
    PAGE 666  ''PROGRAM  OLDER.VERSION            9773
TOTAL.CAS
    PAGE 269  ROUTINE BTRY.EFFECTS                2511 2515 2518 2541
    PAGE 270                                      2544 2547 2551 2555 2557 2564 2569 2572 2577
    PAGE 271                                      2605 2611 2644 2644 2648 2659
    PAGE 272                                      2682
TOTAL.EFF.NEW
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION     9801 9811 9812 9816 9816 9818 9821
TOTAL.EFF.OLD
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION     9801 9809 9811 9811 9821
TOTAL.TIME
    PAGE 194  ROUTINE FIND.START.TIME             8941 8943 8945 8946
    PAGE 195                                      8951 8959 8962 8964 8965 8970 8972 8974
TOT.FIRED.SDM
    PAGE  42  ''SECTION FOR DEFINITIONS           2415
    PAGE 264  ROUTINE BTRY.EFFECTS                2213
    PAGE 593  ROUTINE AMMO.RPT                    7210 7211 7215
    PAGE 701  ''PROGRAM  OLDER.VERSION            1845
TOT.FOLLOW
    PAGE 182  ROUTINE FA.BN.ASGN                  8307
    PAGE 184                                      8409
    PAGE 185                                      8495
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION       9005
    PAGE 198                                      9117
    PAGE 334  ROUTINE FINISH.COMPUTATION          5393 5398
    PAGE 470  PROCESS TARGET.REPORT               1497
    PAGE 476                                      1858 1863 1866
    PAGE 480  PROCESS FIRE.MISSION                2038
TOT.HOW.LOSS
    PAGE 271  ROUTINE BTRY.EFFECTS                2606 2610 2611
TOT.INT.CPE
    PAGE 176  ROUTINE EST.COVERAGE                8016 8057
    PAGE 177                                      8067 8074 8113
    PAGE 178                                      8123 8130 8146
    PAGE 188  ROUTINE FINAL.COVERAGE              8557 8600 8601 8602
    PAGE 189                                      8613 8614 8615 8623 8657 8658 8659
    PAGE 190                                      8670 8671 8672 8680 8696 8697
    PAGE 193                                      8877
TOT.KILL.SDM
    PAGE  42  ''SECTION FOR DEFINITIONS           2414
    PAGE 269  ROUTINE BTRY.EFFECTS                2514
    PAGE 593  ROUTINE AMMO.RPT                    7221
    PAGE 701  ''PROGRAM  OLDER.VERSION            1844
TOT.RH
    PAGE 538  ROUTINE MFO.INPUT                   4944 4953 4954 4965 4969 4974
```

```
    PAGE 562  ROUTINE MCFR.INPUT                    5889 5901 5902
    PAGE 563  ROUTINE MPDB.INPUT                    5926 5934 5935 5938
TOT.TIME
    PAGE 356  EVENT BTL.ENDED                       6266
    PAGE 357                                        6323
TOWN.PCT
    PAGE 242  ROUTINE UNIT.ENVIR                    1137 1168
    PAGE 243                                        1202 1209
    PAGE 244                                        1264
TRANSITION.DIST
    PAGE 154  ROUTINE AO.DETECTION                  7083 7094 7131
    PAGE 155                                        7144
    PAGE 428  PROCESS AIR.OBSERVER                  9350 9352
    PAGE 429                                        9394 9405 9418 9426
    PAGE 431                                        9531 9550 9556
    PAGE 432                                        9582 9614
    PAGE 433                                        9632 9650 9669 9672
TRANS.DIST
    PAGE 155  ROUTINE AO.DETECTION                  7146
TRIAL.PK.PTR
    PAGE 150  ROUTINE PK.COMPUTE                    6955 6957 6959 6961
TRIANGLE.
    PAGE 314  ROUTINE FLIGHT.PATH                   4660
TRUNC.F
    PAGE 267  ROUTINE BTRY.EFFECTS                  2424
    PAGE 281  ROUTINE AC.DF.EFFECTS                 3073
    PAGE 308  ROUTINE AD.SHOOT                      4367
    PAGE 421  PROCESS AC.ATK.TGT                    8964 8990 8998 9009
    PAGE 429  PROCESS AIR.OBSERVER                  9440
    PAGE 505  PROCESS CAS.MISSION                   3457
    PAGE 652  ROUTINE GAMMA.F                       9248
TR.
    PAGE 474  PROCESS TARGET.REPORT                 1762
TR.ABORT.TIME
    PAGE  38  ''SECTION FOR PROCESSES               2166
    PAGE  48  ''SECTION FOR DEFINITIONS             2752
    PAGE 160  ROUTINE ATTRIT.SENSOR                 7391
    PAGE 168  ROUTINE CHK.COMP.TR                   7715 7715 7716
    PAGE 172  ROUTINE COPY                          7875 7875
    PAGE 217  ROUTINE REQUEST.FASCAM                9969
    PAGE 220  ROUTINE REQUEST.ILLUM                  150
    PAGE 225  ROUTINE REQUEST.SMOKE                  368
    PAGE 363  EVENT CFR.OPERATOR                    6559
    PAGE 387  EVENT PDB.OPERATOR                    7547
    PAGE 432  PROCESS AIR.OBSERVER                  9587
    PAGE 441  PROCESS FORWARD.OBSERVER                47
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1456
    PAGE 470  PROCESS TARGET.REPORT                 1523
    PAGE 472                                        1643
    PAGE 473                                        1686
    PAGE 475                                        1779 1811
    PAGE 590  ROUTINE TR.INPUT                      7092
    PAGE 654  PROCESS AIRBORNE.RADAR                9368
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9569
    PAGE 697  ''PROGRAM  OLDER.VERSION              1604
    PAGE 707                                        2174
```

```
TR.ASGND.BATS
    PAGE  38  ''SECTION FOR PROCESSES                2169
    PAGE 172  ROUTINE COPY                           7878 7878
    PAGE 697  ''PROGRAM  OLDER.VERSION               1607
TR.CEP
    PAGE   4  PROGRAM REVISIONS                       207
    PAGE  38  ''SECTION FOR PROCESSES                2167
    PAGE 157  ROUTINE AO.DETECTION                   7272 7275 7277
    PAGE 170  ROUTINE COMBINE.TRS                    7773 7774 7777 7777 7777
    PAGE 172  ROUTINE COPY                           7876 7876
    PAGE 178  ROUTINE EST.COVERAGE                   8147
    PAGE 190  ROUTINE FINAL.COVERAGE                 8696 8697
    PAGE 230  ROUTINE RPV.DETECTION                   595  598  600
    PAGE 257  ROUTINE FO.DETECTION                   1867 1868 1872 1878 1880
    PAGE 362  EVENT CFR.OPERATOR                     6510 6511 6515 6519 6521
    PAGE 387  EVENT PDB.OPERATOR                     7529 7530 7533 7537 7539
    PAGE 470  PROCESS TARGET.REPORT                  1514
    PAGE 482  PROCESS FIRE.MISSION                   2160 2194 2194
    PAGE 483                                         2205 2205
    PAGE 485                                         2326
    PAGE 590  ROUTINE TR.INPUT                       7090
    PAGE 656  ROUTINE AR.DETECTION                   9431 9433
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                9576 9579 9581
    PAGE 697  ''PROGRAM  OLDER.VERSION               1605
TR.CUM.EFFECTS
    PAGE  38  ''SECTION FOR PROCESSES                2159
    PAGE 172  ROUTINE COPY                           7868 7868
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION        9809 9818
    PAGE 697  ''PROGRAM  OLDER.VERSION               1597
TR.DET.ELEM.PROB
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES       1762
    PAGE  46  ''SECTION FOR DEFINITIONS              2637
    PAGE 157  ROUTINE AO.DETECTION                   7286
    PAGE 230  ROUTINE RPV.DETECTION                   614
    PAGE 231  ROUTINE SIZE.ESTIMATE                   653
    PAGE 256  ROUTINE FO.DETECTION                   1793
    PAGE 363  EVENT CFR.OPERATOR                     6556
    PAGE 387  EVENT PDB.OPERATOR                     7544
    PAGE 590  ROUTINE TR.INPUT                       7101
    PAGE 656  ROUTINE AR.DETECTION                   9441
    PAGE 690  ''PROGRAM  OLDER.VERSION               1201
    PAGE 705                                         2060
TR.DET.LINK
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES       1760
    PAGE 157  ROUTINE AO.DETECTION                   7283
    PAGE 170  ROUTINE COMBINE.TRS                    7789
    PAGE 230  ROUTINE RPV.DETECTION                   611
    PAGE 231  ROUTINE SIZE.ESTIMATE                   636  639  646  653
    PAGE 256  ROUTINE FO.DETECTION                   1790
    PAGE 334  ROUTINE FINISH.COMPUTATION             5410
    PAGE 362  EVENT CFR.OPERATOR                     6524
    PAGE 363                                         6579
    PAGE 387  EVENT PDB.OPERATOR                     7541 7557
    PAGE 590  ROUTINE TR.INPUT                       7099 7104
    PAGE 656  ROUTINE AR.DETECTION                   9438
    PAGE 690  ''PROGRAM  OLDER.VERSION               1199
```

```
TR.DET.LIST
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1765
    PAGE  38  ''SECTION FOR PROCESSES                   2173
    PAGE  49  ''SECTION FOR DEFINITIONS                 2827
    PAGE 157  ROUTINE AO.DETECTION                      7279 7284
    PAGE 170  ROUTINE COMBINE.TRS                       7780 7782 7783 7792
    PAGE 211  ROUTINE PGM.MSN.ASGN                      9666 9674
    PAGE 230  ROUTINE RPV.DETECTION                      607  612
    PAGE 231  ROUTINE SIZE.ESTIMATE                      636
    PAGE 240  ROUTINE TARGET.ANALYSIS                   1046
    PAGE 255  ROUTINE FO.DETECTION                      1786
    PAGE 256                                            1791 1827 1830
    PAGE 257                                            1897
    PAGE 258                                            1939
    PAGE 334  ROUTINE FINISH.COMPUTATION                5407 5409
    PAGE 363  EVENT CFR.OPERATOR                        6557 6578
    PAGE 387  EVENT PDB.OPERATOR                        7545 7556
    PAGE 431  PROCESS AIR.OBSERVER                      9557
    PAGE 440  PROCESS FORWARD.OBSERVER                    22
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE              1449
    PAGE 590  ROUTINE TR.INPUT                          7107
    PAGE 654  PROCESS AIRBORNE.RADAR                    9357
    PAGE 656  ROUTINE AR.DETECTION                      9439
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                   9563
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1204
    PAGE 697                                            1611
    PAGE 708                                            2248
TR.DET.QUANT
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1763
    PAGE  46  ''SECTION FOR DEFINITIONS                 2638
    PAGE 157  ROUTINE AO.DETECTION                      7288
    PAGE 170  ROUTINE COMBINE.TRS                       7787 7787 7788
    PAGE 211  ROUTINE PGM.MSN.ASGN                      9669 9676
    PAGE 230  ROUTINE RPV.DETECTION                      615  617
    PAGE 231  ROUTINE SIZE.ESTIMATE                      646
    PAGE 256  ROUTINE FO.DETECTION                      1822 1824 1824
    PAGE 257                                            1899
    PAGE 258                                            1941
    PAGE 363  EVENT CFR.OPERATOR                        6555
    PAGE 387  EVENT PDB.OPERATOR                        7543
    PAGE 590  ROUTINE TR.INPUT                          7102
    PAGE 690  ''PROGRAM  OLDER.VERSION                  1202
    PAGE 705                                            2061
TR.DET.TE
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES          1761
    PAGE  46  ''SECTION FOR DEFINITIONS                 2636
    PAGE 157  ROUTINE AO.DETECTION                      7280 7285 7289
    PAGE 170  ROUTINE COMBINE.TRS                       7784 7784
    PAGE 211  ROUTINE PGM.MSN.ASGN                      9668 9675
    PAGE 230  ROUTINE RPV.DETECTION                      608  613
    PAGE 231  ROUTINE SIZE.ESTIMATE                      639
    PAGE 240  ROUTINE TARGET.ANALYSIS                   1050
    PAGE 256  ROUTINE FO.DETECTION                      1787 1792
    PAGE 257                                            1898
    PAGE 258                                            1940
    PAGE 362  EVENT CFR.OPERATOR                        6525 6526
```

```
    PAGE 387  EVENT PDB.OPERATOR                    7542
    PAGE 590  ROUTINE TR.INPUT                      7100 7103
    PAGE 656  ROUTINE AR.DETECTION                  9440
    PAGE 690  ''PROGRAM  OLDER.VERSION              1200
    PAGE 705                                        2059
TR.EST.RAD
    PAGE   2  PROGRAM REVISIONS                      102
TR.EST.RADIUS
    PAGE   2  PROGRAM REVISIONS                      103
    PAGE  38  ''SECTION FOR PROCESSES               2163
    PAGE  46  ''SECTION FOR DEFINITIONS             2654
    PAGE 172  ROUTINE COPY                          7872 7872
    PAGE 178  ROUTINE EST.COVERAGE                  8149 8150
    PAGE 199  ROUTINE ILLUM.COMPUTATION             9158 9174
    PAGE 220  ROUTINE REQUEST.ILLUM                  148
    PAGE 225  ROUTINE REQUEST.SMOKE                  366
    PAGE 233  ROUTINE SMOKE.COMPUTATION              715  731
    PAGE 241  ROUTINE TARGET.ANALYSIS               1095
    PAGE 697  ''PROGRAM  OLDER.VERSION              1601
    PAGE 705                                        2078
TR.EST.TU
    PAGE   3  PROGRAM REVISIONS                      149
    PAGE  38  ''SECTION FOR PROCESSES               2164
    PAGE 171  ROUTINE COMPARE.TRS                   7824 7824
    PAGE 172  ROUTINE COPY                          7873 7873
    PAGE 182  ROUTINE FA.BN.ASGN                    8317
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION         9010
    PAGE 240  ROUTINE TARGET.ANALYSIS               1085
    PAGE 241                                        1094
    PAGE 262  ROUTINE BTRY.EFFECTS                  2125
    PAGE 263                                        2172
    PAGE 471  PROCESS TARGET.REPORT                 1542 1567
    PAGE 473                                        1713
    PAGE 636  FUNCTION HE.WLA                       8889
    PAGE 638  FUNCTION ICM.WLA                      8990
    PAGE 697  ''PROGRAM  OLDER.VERSION              1602
TR.EST.X
    PAGE  38  ''SECTION FOR PROCESSES               2161
    PAGE  46  ''SECTION FOR DEFINITIONS             2648
    PAGE 157  ROUTINE AO.DETECTION                  7274
    PAGE 170  ROUTINE COMBINE.TRS                   7778 7778 7778
    PAGE 172  ROUTINE COPY                          7870 7870
    PAGE 191  ROUTINE FINAL.COVERAGE                8755
    PAGE 217  ROUTINE REQUEST.FASCAM                9966
    PAGE 220  ROUTINE REQUEST.ILLUM                  146
    PAGE 225  ROUTINE REQUEST.SMOKE                  364
    PAGE 230  ROUTINE RPV.DETECTION                  597
    PAGE 257  ROUTINE FO.DETECTION                  1877
    PAGE 261  ROUTINE BTRY.EFFECTS                  2027
    PAGE 362  EVENT CFR.OPERATOR                    6518
    PAGE 387  EVENT PDB.OPERATOR                    7536
    PAGE 590  ROUTINE TR.INPUT                      7088 7093 7093
    PAGE 632  FUNCTION EST.RANGE                    8752
    PAGE 633  FUNCTION EST.TR.RANGE                 8765 8765
    PAGE 634  FUNCTION FEBA.BAND                    8803 8806
    PAGE 656  ROUTINE AR.DETECTION                  9430
```

```
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                9578
    PAGE 697  ''PROGRAM  OLDER.VERSION               1599
    PAGE 705                                         2072
TR.EST.Y
    PAGE  38  ''SECTION FOR PROCESSES                2162
    PAGE  46  ''SECTION FOR DEFINITIONS              2649
    PAGE 157  ROUTINE AO.DETECTION                   7276
    PAGE 170  ROUTINE COMBINE.TRS                    7779 7779 7779
    PAGE 172  ROUTINE COPY                           7871 7871
    PAGE 191  ROUTINE FINAL.COVERAGE                 8756
    PAGE 217  ROUTINE REQUEST.FASCAM                 9967
    PAGE 220  ROUTINE REQUEST.ILLUM                   147
    PAGE 225  ROUTINE REQUEST.SMOKE                   365
    PAGE 230  ROUTINE RPV.DETECTION                   599
    PAGE 257  ROUTINE FO.DETECTION                   1879
    PAGE 261  ROUTINE BTRY.EFFECTS                   2022 2028
    PAGE 362  EVENT CFR.OPERATOR                     6520
    PAGE 387  EVENT PDB.OPERATOR                     7538
    PAGE 590  ROUTINE TR.INPUT                       7089 7094 7094
    PAGE 632  FUNCTION EST.RANGE                     8753
    PAGE 633  FUNCTION EST.TR.RANGE                  8766 8766
    PAGE 634  FUNCTION FEBA.BAND                     8794
    PAGE 656  ROUTINE AR.DETECTION                   9432
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                9580
    PAGE 697  ''PROGRAM  OLDER.VERSION               1600
    PAGE 705                                         2073
TR.ES.TU
    PAGE   2  PROGRAM REVISIONS                       90
TR.EX.TGT.REPORT
    PAGE  38  ''SECTION FOR PROCESSES                2170
    PAGE 470  PROCESS TARGET.REPORT                  1491 1492 1493 1494 1495 1496 1497 1499 1519 1536
    PAGE 471                                         1583
    PAGE 472                                         1606 1629
    PAGE 473                                         1660
    PAGE 474                                         1730
    PAGE 475                                         1796
    PAGE 477                                         1910
    PAGE 697  ''PROGRAM  OLDER.VERSION               1608
TR.FDC
    PAGE  38  ''SECTION FOR PROCESSES                2149
    PAGE 157  ROUTINE AO.DETECTION                   7270 7301 7302 7303 7303 7305
    PAGE 158                                         7306 7306
    PAGE 172  ROUTINE COPY                           7858 7858
    PAGE 185  ROUTINE FA.BN.ASGN                     8503
    PAGE 211  ROUTINE PGM.MSN.ASGN                   9691
    PAGE 217  ROUTINE REQUEST.FASCAM                 9961
    PAGE 220  ROUTINE REQUEST.ILLUM                   141
    PAGE 225  ROUTINE REQUEST.SMOKE                   359
    PAGE 230  ROUTINE RPV.DETECTION                   593  603
    PAGE 256  ROUTINE FO.DETECTION                   1835 1836 1837 1837
    PAGE 257                                         1901 1902
    PAGE 258                                         1915 1927 1946 1947 1948 1948 1950 1951 1951 1953 1954 1954
    PAGE 362  EVENT CFR.OPERATOR                     6508 6527
    PAGE 363                                         6538 6551
    PAGE 387  EVENT PDB.OPERATOR                     7527
    PAGE 470  PROCESS TARGET.REPORT                  1490
```

```
    PAGE 476                                        1839
    PAGE 590  ROUTINE TR.INPUT                      7082
    PAGE 656  ROUTINE AR.DETECTION                  9427 9436
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9574 9584
    PAGE 697  ''PROGRAM  OLDER.VERSION              1587
TR.FDC.STATUS
    PAGE  38  ''SECTION FOR PROCESSES               2150
    PAGE 172  ROUTINE COPY                          7859 7859
    PAGE 332  ROUTINE FDC.TR.DEQ                    5353
    PAGE 333  ROUTINE FDC.TR.ENQ                    5372 5374
    PAGE 472  PROCESS TARGET.REPORT                 1639
    PAGE 697  ''PROGRAM  OLDER.VERSION              1588
TR.FM.LIST
    PAGE  35  ''SECTION FOR PROCESSES               2011
    PAGE  38                                        2172
    PAGE  49  ''SECTION FOR DEFINITIONS             2828
    PAGE 185  ROUTINE FA.BN.ASGN                    8493
    PAGE 211  ROUTINE PGM.MSN.ASGN                  9683 9687
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS        2759 2760 2761
    PAGE 473  PROCESS TARGET.REPORT                 1701
    PAGE 476                                        1879
    PAGE 477                                        1896
    PAGE 486  PROCESS FIRE.MISSION                  2391 2392
    PAGE 694  ''PROGRAM  OLDER.VERSION              1449
    PAGE 697                                        1610
    PAGE 708                                        2249
TR.MIL.WORTH
    PAGE  38  ''SECTION FOR PROCESSES               2155
    PAGE  46  ''SECTION FOR DEFINITIONS             2647
    PAGE  49                                        2808
    PAGE 158  ROUTINE AO.DETECTION                  7308
    PAGE 172  ROUTINE COPY                          7864 7864
    PAGE 183  ROUTINE FA.BN.ASGN                    8395
    PAGE 184                                        8420
    PAGE 208  ROUTINE PGM.MSN.ASGN                  9527
    PAGE 209                                        9566 9567 9596 9600
    PAGE 217  ROUTINE REQUEST.FASCAM                9964
    PAGE 220  ROUTINE REQUEST.ILLUM                  144
    PAGE 225  ROUTINE REQUEST.SMOKE                  362
    PAGE 241  ROUTINE TARGET.ANALYSIS               1099 1106
    PAGE 257  ROUTINE FO.DETECTION                  1884 1886 1890
    PAGE 258                                        1925 1956
    PAGE 363  EVENT CFR.OPERATOR                    6549
    PAGE 471  PROCESS TARGET.REPORT                 1557 1558 1559 1560
    PAGE 473                                        1708
    PAGE 474                                        1735
    PAGE 697  ''PROGRAM  OLDER.VERSION              1593
    PAGE 705                                        2071
    PAGE 708                                        2229
TR.MISSION.TYPE
    PAGE  38  ''SECTION FOR PROCESSES               2148
    PAGE  47  ''SECTION FOR DEFINITIONS             2697
    PAGE 169  ROUTINE CHK.FD.TR                     7737 7738 7739
    PAGE 172  ROUTINE COPY                          7857 7857
    PAGE 184  ROUTINE FA.BN.ASGN                    8430 8431 8432
    PAGE 185                                        8463 8468 8473
```

```
    PAGE 187  ROUTINE FD.EFFECTS.REQ                8539 8540 8541
    PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION         9100
    PAGE 198                                        9106
    PAGE 216  ROUTINE REQUEST.FASCAM                9908 9916
    PAGE 217                                        9960
    PAGE 219  ROUTINE REQUEST.ILLUM                   93
    PAGE 220                                         114  140
    PAGE 224  ROUTINE REQUEST.SMOKE                  291  322
    PAGE 225                                         358
    PAGE 471  PROCESS TARGET.REPORT                 1553 1554 1555 1556
    PAGE 474                                        1747 1749
    PAGE 634  FUNCTION FEBA.BAND                    8787 8788 8789
    PAGE 635                                        8832 8833 8834
    PAGE 697  ''PROGRAM  OLDER.VERSION              1586
    PAGE 706                                        2121
TR.MOVE
    PAGE  38  ''SECTION FOR PROCESSES               2160
    PAGE 171  ROUTINE COMPARE.TRS                   7810 7810
    PAGE 172  ROUTINE COPY                          7869 7869
    PAGE 187  ROUTINE FD.EFFECTS.REQ                8533
    PAGE 230  ROUTINE RPV.DETECTION                  603
    PAGE 257  ROUTINE FO.DETECTION                  1846 1848
    PAGE 272  ROUTINE BTRY.EFFECTS                  2694
    PAGE 590  ROUTINE TR.INPUT                      7087
    PAGE 656  ROUTINE AR.DETECTION                  9437
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9584
    PAGE 697  ''PROGRAM  OLDER.VERSION              1598
TR.PGM.STATUS
    PAGE  38  ''SECTION FOR PROCESSES               2156
    PAGE 157  ROUTINE AO.DETECTION                  7300
    PAGE 160  ROUTINE ATTRIT.SENSOR                 7380 7410
    PAGE 172  ROUTINE COPY                          7865 7865
    PAGE 208  ROUTINE PGM.MSN.ASGN                  9503 9515
    PAGE 209                                        9557
    PAGE 210                                        9605
    PAGE 217  ROUTINE REQUEST.FASCAM                9965
    PAGE 220  ROUTINE REQUEST.ILLUM                  145
    PAGE 225  ROUTINE REQUEST.SMOKE                  363
    PAGE 230  ROUTINE RPV.DETECTION                  606
    PAGE 238  ROUTINE SWITCH.FO                      936  945
    PAGE 256  ROUTINE FO.DETECTION                  1794 1803 1813 1817 1836
    PAGE 257                                        1883 1889 1894
    PAGE 258                                        1914 1920 1924 1927 1936 1943
    PAGE 261  ROUTINE BTRY.EFFECTS                  2032 2039
    PAGE 262                                        2088 2114 2122
    PAGE 263                                        2163
    PAGE 264                                        2196 2240
    PAGE 267                                        2396 2399 2403
    PAGE 269                                        2527 2539 2543
    PAGE 270                                        2550 2573
    PAGE 271                                        2639 2643
    PAGE 362  EVENT CFR.OPERATOR                    6522
    PAGE 363                                        6540 6544 6548 6551
    PAGE 387  EVENT PDB.OPERATOR                    7540
    PAGE 435  PROCESS ARTY.ASSESS                   9729
    PAGE 436                                        9774
```

```
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1458
    PAGE 470  PROCESS TARGET.REPORT                 1503
    PAGE 472                                        1646
    PAGE 473                                        1695
    PAGE 474                                        1740 1746 1759
    PAGE 481  PROCESS FIRE.MISSION                  2104
    PAGE 482                                        2148 2162 2175 2185
    PAGE 483                                        2211 2256
    PAGE 485                                        2329 2346 2365 2370 2371 2374
    PAGE 486                                        2402
    PAGE 590  ROUTINE TR.INPUT                      7086
    PAGE 611  ROUTINE OUTPUT.ATTRITION              7963 7964
    PAGE 656  ROUTINE AR.DETECTION                  9434
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9582
    PAGE 697  ''PROGRAM  OLDER.VERSION              1594
TR.RECVD.TIME
    PAGE   5  ''PROGRAM'' PREAMBLE                   222
    PAGE  38  ''SECTION FOR PROCESSES               2165
    PAGE  48  ''SECTION FOR DEFINITIONS             2753
    PAGE 171  ROUTINE COMPARE.TRS                   7822 7822
    PAGE 172  ROUTINE COPY                          7874 7874
    PAGE 217  ROUTINE REQUEST.FASCAM                9968
    PAGE 220  ROUTINE REQUEST.ILLUM                  149
    PAGE 225  ROUTINE REQUEST.SMOKE                  367
    PAGE 383  EVENT CFR.OPERATOR                    6558 6559
    PAGE 387  EVENT PDB.OPERATOR                    7546 7547
    PAGE 432  PROCESS AIR.OBSERVER                  9586
    PAGE 441  PROCESS FORWARD.OBSERVER               46
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1455
    PAGE 476  PROCESS TARGET.REPORT                 1877
    PAGE 590  ROUTINE TR.INPUT                      7091
    PAGE 654  PROCESS AIRBORNE.RADAR                9367
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9568
    PAGE 664  ''PROGRAM  OLDER.VERSION              9660
    PAGE 697                                        1603
    PAGE 707                                        2175
TR.REM.EFFECTS
    PAGE  38  ''SECTION FOR PROCESSES               2157
    PAGE  46  ''SECTION FOR DEFINITIONS             2644
    PAGE 172  ROUTINE COPY                          7866 7866
    PAGE 181  ROUTINE FASCAM.COMPUTATION            8266 8269 8269 8279
    PAGE 182  ROUTINE FA.BN.ASGN                    8313
    PAGE 184                                        8452 8456
    PAGE 199  ROUTINE ILLUM.COMPUTATION             9159 9167 9167 9174
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION       9813 9815
    PAGE 233  ROUTINE SMOKE.COMPUTATION              716  724  724  731
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS              1416
    PAGE 474  PROCESS TARGET.REPORT                 1761 1770
    PAGE 475                                        1774 1801 1812 1823
    PAGE 476                                        1855
    PAGE 697  ''PROGRAM  OLDER.VERSION              1595
    PAGE 705                                        2068
TR.REP.UNIT
    PAGE  38  ''SECTION FOR PROCESSES               2153
    PAGE  46  ''SECTION FOR DEFINITIONS             2645
    PAGE 157  ROUTINE AO.DETECTION                  7271
```

```
    PAGE 172  ROUTINE COPY                          7862 7862
    PAGE 181  ROUTINE FASCAM.COMPUTATION            8276
    PAGE 200  ROUTINE ILLUM.EFFECTS                 9192
    PAGE 217  ROUTINE REQUEST.FASCAM                9962
    PAGE 219  ROUTINE REQUEST.ILLUM                   95
    PAGE 220                                          97  100  102  115  117  122  124  142
    PAGE 224  ROUTINE REQUEST.SMOKE                  293  295  298  304  323  325  330  336
    PAGE 225                                         360
    PAGE 230  ROUTINE RPV.DETECTION                  594
    PAGE 233  ROUTINE SMOKE.COMPUTATION              701  702
    PAGE 234  ROUTINE SMOKE.EFFECTS                  749
    PAGE 235                                         827  829
    PAGE 236                                         859
    PAGE 256  ROUTINE FO.DETECTION                  1841
    PAGE 257                                        1869
    PAGE 267  ROUTINE BTRY.EFFECTS                  2405 2407
    PAGE 362  EVENT CFR.OPERATOR                    6509 6512
    PAGE 387  EVENT PDB.OPERATOR                    7528 7531
    PAGE 441  PROCESS FORWARD.OBSERVER                57   58
    PAGE 474  PROCESS TARGET.REPORT                 1763
    PAGE 484  PROCESS FIRE.MISSION                  2266
    PAGE 590  ROUTINE TR.INPUT                      7084
    PAGE 656  ROUTINE AR.DETECTION                  9428
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9575
    PAGE 697  ''PROGRAM  OLDER.VERSION              1591
    PAGE 705                                        2069
TR.REQ.EFFECTS
    PAGE  38  ''SECTION FOR PROCESSES               2158
    PAGE 172  ROUTINE COPY                          7867 7867
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION       9810
    PAGE 474  PROCESS TARGET.REPORT                 1769 1770
    PAGE 475                                        1773
    PAGE 697  ''PROGRAM  OLDER.VERSION              1596
TR.SENSOR.ID
    PAGE   3  PROGRAM REVISIONS                      150
    PAGE  38  ''SECTION FOR PROCESSES               2152
    PAGE 157  ROUTINE AO.DETECTION                  7269
    PAGE 171  ROUTINE COMPARE.TRS                   7812 7812
    PAGE 172  ROUTINE COPY                          7861 7861
    PAGE 230  ROUTINE RPV.DETECTION                  592
    PAGE 256  ROUTINE FO.DETECTION                  1840
    PAGE 257                                        1861
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS        2765
    PAGE 362  EVENT CFR.OPERATOR                    6507
    PAGE 387  EVENT PDB.OPERATOR                    7526
    PAGE 441  PROCESS FORWARD.OBSERVER                50   52
    PAGE 470  PROCESS TARGET.REPORT                 1516 1533
    PAGE 471                                        1542 1544 1580
    PAGE 472                                        1602 1626
    PAGE 473                                        1657
    PAGE 474                                        1727 1743
    PAGE 475                                        1793
    PAGE 476                                        1841
    PAGE 477                                        1904
    PAGE 480  PROCESS FIRE.MISSION                  2044
    PAGE 485                                        2349 2352
```

```
    PAGE 656  ROUTINE AR.DETECTION                9426
    PAGE 659  PROCESS PHOTO.IR.FLIGHT             9573
    PAGE 697  ''PROGRAM  OLDER.VERSION            1590
TR.SENSOR.TYPE
    PAGE  38  ''SECTION FOR PROCESSES             2151
    PAGE  47  ''SECTION FOR DEFINITIONS           2698
    PAGE 157  ROUTINE AO.DETECTION                7268
    PAGE 160  ROUTINE ATTRIT.SENSOR               7392
    PAGE 172  ROUTINE COPY                        7860 7860
    PAGE 230  ROUTINE RPV.DETECTION                591
    PAGE 256  ROUTINE FO.DETECTION                1839
    PAGE 362  EVENT CFR.OPERATOR                  6506
    PAGE 387  EVENT PDB.OPERATOR                  7525
    PAGE 474  PROCESS TARGET.REPORT               1741
    PAGE 476                                      1874
    PAGE 482  PROCESS FIRE.MISSION                2161
    PAGE 485                                      2345
    PAGE 590  ROUTINE TR.INPUT                    7083
    PAGE 656  ROUTINE AR.DETECTION                9425
    PAGE 659  PROCESS PHOTO.IR.FLIGHT             9572
    PAGE 697  ''PROGRAM  OLDER.VERSION            1589
    PAGE 706                                      2122
TR.START.TIME
    PAGE   5  ''PROGRAM'' PREAMBLE                 222
    PAGE 185  ROUTINE FA.BN.ASGN                  8494 8497
    PAGE 476  PROCESS TARGET.REPORT               1859
    PAGE 629  FUNCTION BTRY.AVAILABLE             8655
    PAGE 664  ''PROGRAM  OLDER.VERSION            9660
TR.TGT.UNIT
    PAGE  38  ''SECTION FOR PROCESSES             2154
    PAGE  46  ''SECTION FOR DEFINITIONS           2646
    PAGE 157  ROUTINE AO.DETECTION                7266 7267
    PAGE 168  ROUTINE CHK.COMP.TR                 7709 7709
    PAGE 169  ROUTINE CHK.FD.TR                   7741 7741
    PAGE 172  ROUTINE COPY                        7863 7863
    PAGE 180  ROUTINE EST.MIL.WORTH               8227
    PAGE 181  ROUTINE FASCAM.COMPUTATION          8256 8277
    PAGE 183  ROUTINE FA.BN.ASGN                  8400
    PAGE 184                                      8410 8410
    PAGE 188  ROUTINE FINAL.COVERAGE              8562
    PAGE 191                                      8757 8758
    PAGE 193                                      8874 8883
    PAGE 199  ROUTINE ILLUM.COMPUTATION           9147
    PAGE 200  ROUTINE ILLUM.EFFECTS               9193
    PAGE 208  ROUTINE PGM.MSN.ASGN                9504 9505
    PAGE 210                                      9620 9628 9629
    PAGE 216  ROUTINE REQUEST.FASCAM              9909 9917
    PAGE 217                                      9963
    PAGE 220  ROUTINE REQUEST.ILLUM                100  106  122  128  143
    PAGE 224  ROUTINE REQUEST.SMOKE                298  309  330  341
    PAGE 225                                       361
    PAGE 230  ROUTINE RPV.DETECTION                589  590
    PAGE 233  ROUTINE SMOKE.COMPUTATION            701  704
    PAGE 235  ROUTINE SMOKE.EFFECTS                827  829  841
    PAGE 236                                       859  861
    PAGE 239  ROUTINE TARGET.ANALYSIS              992
```

```
    PAGE 256  ROUTINE FO.DETECTION                    1842
    PAGE 257                                          1862
    PAGE 260  ROUTINE BTRY.EFFECTS                    2003
    PAGE 261                                          2029
    PAGE 267                                          2404 2406 2409 2410 2411
    PAGE 268                                          2456
    PAGE 272                                          2704
    PAGE 362  EVENT CFR.OPERATOR                      6505
    PAGE 387  EVENT PDB.OPERATOR                      7524
    PAGE 441  PROCESS FORWARD.OBSERVER                  56
    PAGE 471  PROCESS TARGET.REPORT                   1548
    PAGE 474                                          1754
    PAGE 476                                          1875 1876
    PAGE 482  PROCESS FIRE.MISSION                    2152
    PAGE 483                                          2226 2227
    PAGE 484                                          2266 2271 2272 2273 2281 2289
    PAGE 590  ROUTINE TR.INPUT                        7085
    PAGE 611  ROUTINE OUTPUT.ATTRITION                7956 7976
    PAGE 634  FUNCTION FEBA.BAND                      8786 8812 8821 8824
    PAGE 636  FUNCTION HE.WLA                         8875
    PAGE 638  FUNCTION ICM.WLA                        8997
    PAGE 656  ROUTINE AR.DETECTION                    9424
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                 9571
    PAGE 697  ''PROGRAM  OLDER.VERSION                1592
    PAGE 705                                          2070
TR.TOT.STATUS
    PAGE  38  ''SECTION FOR PROCESSES                 2168
    PAGE 172  ROUTINE COPY                            7877 7877
    PAGE 183  ROUTINE FA.BN.ASGN                      8400
    PAGE 184                                          8409
    PAGE 185                                          8495
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION           9034
    PAGE 197                                          9060
    PAGE 198                                          9110 9117
    PAGE 217  ROUTINE REQUEST.FASCAM                  9970
    PAGE 220  ROUTINE REQUEST.ILLUM                    151
    PAGE 225  ROUTINE REQUEST.SMOKE                    369
    PAGE 334  ROUTINE FINISH.COMPUTATION              5398
    PAGE 471  PROCESS TARGET.REPORT                   1551
    PAGE 473                                          1664
    PAGE 475                                          1821
    PAGE 476                                          1846 1866
    PAGE 636  FUNCTION HE.WLA                         8900
    PAGE 697  ''PROGRAM  OLDER.VERSION                1606
TT.FACTORS.INPUT
    PAGE 521  ROUTINE MAIN2                           4272 4274
    PAGE 571  ROUTINE TT.FACTORS.INPUT                6282
TT.IN.USE
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES        1793
    PAGE  68  ROUTINE GENERAL.BATTLE                  3528 3560
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                 3842 3852
    PAGE 398  EVENT START.BATTLE                      7920
    PAGE 691  ''PROGRAM  OLDER.VERSION                1232
TT.LOS.SCALE
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES         838
    PAGE  48  ''SECTION FOR DEFINITIONS               2755
```

```
    PAGE  97  ROUTINE LINE.OF.SIGHT                  4733 4752
    PAGE 109  ROUTINE NEW.SEGMENT                    5246
    PAGE 210  ROUTINE PGM.MSN.ASGN                   9617
    PAGE 479  PROCESS WITH.DRAW                      1978 2001
    PAGE 571  ROUTINE TT.FACTORS.INPUT               6310
    PAGE 674  ''PROGRAM  OLDER.VERSION                280
    PAGE 707                                         2177
TT.LOS.SHAPE
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        837
    PAGE  48  ''SECTION FOR DEFINITIONS              2754
    PAGE  97  ROUTINE LINE.OF.SIGHT                  4732 4751
    PAGE 109  ROUTINE NEW.SEGMENT                    5245
    PAGE 210  ROUTINE PGM.MSN.ASGN                   9616
    PAGE 479  PROCESS WITH.DRAW                      1977 2000
    PAGE 571  ROUTINE TT.FACTORS.INPUT               6308
    PAGE 674  ''PROGRAM  OLDER.VERSION                279
    PAGE 707                                         2176
TT.MOVING.LOS.BREAK
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        842
    PAGE  96  ROUTINE LINE.OF.SIGHT                  4701
    PAGE 101  ROUTINE LOS.CHECK                      4903
    PAGE 107  ROUTINE NEW.SEGMENT                    5134
    PAGE 571  ROUTINE TT.FACTORS.INPUT               6315
    PAGE 674  ''PROGRAM  OLDER.VERSION                284
TT.M.S.LOS.BREAK
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        843
    PAGE  96  ROUTINE LINE.OF.SIGHT                  4695 4699
    PAGE 101  ROUTINE LOS.CHECK                      4896 4901
    PAGE 107  ROUTINE NEW.SEGMENT                    5126 5131
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION          812
    PAGE 571  ROUTINE TT.FACTORS.INPUT               6317
    PAGE 674  ''PROGRAM  OLDER.VERSION                285
TT.NLOS.SCALE
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        840
    PAGE  48  ''SECTION FOR DEFINITIONS              2757
    PAGE  97  ROUTINE LINE.OF.SIGHT                  4736 4755
    PAGE 109  ROUTINE NEW.SEGMENT                    5242
    PAGE 479  PROCESS WITH.DRAW                      1974 2004
    PAGE 571  ROUTINE TT.FACTORS.INPUT               6311
    PAGE 674  ''PROGRAM  OLDER.VERSION                282
    PAGE 707                                         2179
TT.NLOS.SHAPE
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        839
    PAGE  48  ''SECTION FOR DEFINITIONS              2756
    PAGE  97  ROUTINE LINE.OF.SIGHT                  4735 4754
    PAGE 109  ROUTINE NEW.SEGMENT                    5241
    PAGE 479  PROCESS WITH.DRAW                      1973 2003
    PAGE 571  ROUTINE TT.FACTORS.INPUT               6309
    PAGE 674  ''PROGRAM  OLDER.VERSION                281
    PAGE 707                                         2178
TT.SIDE
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1794
    PAGE  60  ROUTINE CREATE.TEAMS                   3167
    PAGE  68  ROUTINE GENERAL.BATTLE                 3529 3561
    PAGE  69                                         3598
    PAGE  72  ROUTINE ORIENTATION                    3697
```

```
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                3843
    PAGE 691  ''PROGRAM  OLDER.VERSION               1233
TT.STATIONARY.LOS.BREAK
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES        841
    PAGE  98  ROUTINE LINE.OF.SIGHT                  4693
    PAGE 101  ROUTINE LOS.CHECK                      4894
    PAGE 107  ROUTINE NEW.SEGMENT                    5123
    PAGE 115  ROUTINE PROX.CHECK                     5528
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION          810
    PAGE 571  ROUTINE TT.FACTORS.INPUT               6313
    PAGE 674  ''PROGRAM  OLDER.VERSION                283
TT.TYPE
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1795
    PAGE  60  ROUTINE CREATE.TEAMS                   3168
    PAGE  68  ROUTINE GENERAL.BATTLE                 3530 3562
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                3844
    PAGE 691  ''PROGRAM  OLDER.VERSION               1234
TT.UNIT
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1792
    PAGE  68  ROUTINE GENERAL.BATTLE                 3531 3563
    PAGE  69                                         3599 3615
    PAGE 691  ''PROGRAM  OLDER.VERSION               1231
TUBES.AVAILABLE
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ           9323 9336 9357
    PAGE 204                                         9367
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS               1415
    PAGE 249                                         1447
TU.AC.PER.MSN
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES        927
    PAGE 284  ROUTINE CAS.EVAL                       3267
    PAGE 527  ROUTINE CAT.TU.INPUT                   4486
    PAGE 676  ''PROGRAM  OLDER.VERSION                369
TU.ATKING.AC
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES        926
    PAGE 284  ROUTINE CAS.EVAL                       3265
    PAGE 527  ROUTINE CAT.TU.INPUT                   4485
    PAGE 676  ''PROGRAM  OLDER.VERSION                368
TU.CAT
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES        929
    PAGE  66  ROUTINE FORM.TF.LIST                   3425
    PAGE  89  ROUTINE FA.BN.MOVEMENT                 4425
    PAGE 104  ROUTINE MINE.DELAY                     5029
    PAGE 171  ROUTINE COMPARE.TRS                    7824 7824
    PAGE 181  ROUTINE FASCAM.COMPUTATION             8256
    PAGE 196  ROUTINE HE.OR.ICM.COMPUTATION          9011
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9147
    PAGE 233  ROUTINE SMOKE.COMPUTATION               702  704
    PAGE 242  ROUTINE UNIT.ENVIR                     1164
    PAGE 252  ROUTINE MINE.EFFECTS                   1604
    PAGE 262  ROUTINE BTRY.EFFECTS                   2109 2125
    PAGE 263                                         2172
    PAGE 270                                         2582
    PAGE 272                                         2709
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                2905 2953
    PAGE 282  ROUTINE AC.DF.EFFECTS                  3135
    PAGE 384  EVENT OFF.LINE.ATTRITION               7435
```

```
    PAGE 476  PROCESS TARGET.REPORT                    1876
    PAGE 483  PROCESS FIRE.MISSION                     2227
    PAGE 484                                           2281
    PAGE 494  PROCESS SHOOT.OUT                        2814
    PAGE 506  PROCESS CAS.MISSION                      3495
    PAGE 527  ROUTINE CAT.TU.INPUT                     4476 4487
    PAGE 533  ROUTINE UNIT.INPUT                       4735 4736
    PAGE 539  ROUTINE READ.ORDERS                      5013
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                  7257 7277
    PAGE 613  ROUTINE FOR POSITION.OUT                 8012
    PAGE 636  FUNCTION HE.WLA                          8898
    PAGE 638  FUNCTION ICM.WLA                         8996
    PAGE 676  ''PROGRAM  OLDER.VERSION                  371
TU.CRITICAL.EQUIP.INDIC
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES         1782
    PAGE  46  ''SECTION FOR DEFINITIONS                2640
    PAGE 527  ROUTINE CAT.TU.INPUT                     4499 4500
    PAGE 691  ''PROGRAM  OLDER.VERSION                 1221
    PAGE 705                                           2063
TU.CRIT.NO
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES          934
    PAGE  45  ''SECTION FOR DEFINITIONS                2598
    PAGE  58  ROUTINE CREATE.FORCE                     3134
    PAGE 130  ROUTINE CHECK.DEAD                       6082 6100
    PAGE 133  ROUTINE CHECK.FORCE                      6172
    PAGE 138  ROUTINE CHECK.PROX                       6429
    PAGE 139                                           6446
    PAGE 141  ROUTINE CHECK.STREN                      6513 6517
    PAGE 142  ROUTINE DEAD.UNIT                        6542
    PAGE 145  ROUTINE FIN.BATTLE                       6712
    PAGE 324  ROUTINE DECIDE                           5055
    PAGE 398  EVENT START.BATTLE                       7955
    PAGE 421  PROCESS AC.ATK.TGT                       8971 8982
    PAGE 426                                           9255 9288
    PAGE 478  PROCESS WITH.DRAW                        1931
    PAGE 527  ROUTINE CAT.TU.INPUT                     4501
    PAGE 676  ''PROGRAM  OLDER.VERSION                  376
    PAGE 704                                           2019
TU.FREQ
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES          931
    PAGE  45  ''SECTION FOR DEFINITIONS                2596
    PAGE 231  ROUTINE SIZE.ESTIMATE                     648  673
    PAGE 240  ROUTINE TARGET.ANALYSIS                  1043
    PAGE 533  ROUTINE UNIT.INPUT                       4729
    PAGE 676  ''PROGRAM  OLDER.VERSION                  373
    PAGE 704                                           2017
TU.LEVEL
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES          928
    PAGE  47  ''SECTION FOR DEFINITIONS                2699
    PAGE 240  ROUTINE TARGET.ANALYSIS                  1082
    PAGE 263  ROUTINE BTRY.EFFECTS                     2139
    PAGE 272                                           2692
    PAGE 392  EVENT SET.DEBUG                          7640
    PAGE 527  ROUTINE CAT.TU.INPUT                     4475
    PAGE 533  ROUTINE UNIT.INPUT                       4701
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                  7277
```

```
    PAGE 611  ROUTINE OUTPUT.ATTRITION                7985
    PAGE 676  ''PROGRAM  OLDER.VERSION                 370
    PAGE 706                                          2123
TU.MF.FACTOR
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         939
    PAGE  46  ''SECTION FOR DEFINITIONS               2602
    PAGE 104  ROUTINE MINE.DELAY                      5006
    PAGE 527  ROUTINE CAT.TU.INPUT                    4484
    PAGE 676  ''PROGRAM  OLDER.VERSION                 381
    PAGE 704                                          2023
TU.MIL.WORTH
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         930
    PAGE  45  ''SECTION FOR DEFINITIONS               2595
    PAGE 167  ROUTINE CFR.DETECTION                   7692
    PAGE 180  ROUTINE EST.MIL.WORTH                   8227
    PAGE 206  ROUTINE PDB.DETECTION                   9458
    PAGE 527  ROUTINE CAT.TU.INPUT                    4477
    PAGE 676  ''PROGRAM  OLDER.VERSION                 372
    PAGE 704                                          2016
TU.MOV.RATE
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         935
    PAGE  45  ''SECTION FOR DEFINITIONS               2599
    PAGE  80  ROUTINE BLOCK.LOS                       4021 4028 4045
    PAGE  83  ROUTINE CHANGE.LOC                      4148 4154
    PAGE 106  ROUTINE MIN.MOVE                        5093
    PAGE 111  ROUTINE PRED.POS                        5332
    PAGE 210  ROUTINE PGM.MSN.ASGN                    9626 9654
    PAGE 394  EVENT START.ARTY.MOVEMENT               7730
    PAGE 403  EVENT START.MOVE                        8210 8216
    PAGE 527  ROUTINE CAT.TU.INPUT                    4478
    PAGE 676  ''PROGRAM  OLDER.VERSION                 377
    PAGE 704                                          2020
TU.NTE.ID
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES        1772
    PAGE 240  ROUTINE TARGET.ANALYSIS                 1050
    PAGE 528  ROUTINE CAT.TU.INPUT                    4511 4512
    PAGE 690  ''PROGRAM  OLDER.VERSION                1211
TU.NTE.LINK
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES        1771
    PAGE 528  ROUTINE CAT.TU.INPUT                    4510 4511 4512 4513 4516
    PAGE 690  ''PROGRAM  OLDER.VERSION                1210
TU.NTE.SET
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         942
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES        1774
    PAGE  49  ''SECTION FOR DEFINITIONS               2829
    PAGE 240  ROUTINE TARGET.ANALYSIS                 1048
    PAGE 528  ROUTINE CAT.TU.INPUT                    4516
    PAGE 676  ''PROGRAM  OLDER.VERSION                 384
    PAGE 690                                          1213
    PAGE 708                                          2250
TU.OPP.PRIORITY
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         938
    PAGE  46  ''SECTION FOR DEFINITIONS               2600
    PAGE 305  ROUTINE UNIT.PRIORITY                   4285
    PAGE 527  ROUTINE CAT.TU.INPUT                    4483
    PAGE 676  ''PROGRAM  OLDER.VERSION                 380
```

```
    PAGE 704                                        2021
TU.PRIN.TE
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES       933
    PAGE 272  ROUTINE BTRY.EFFECTS                  2692
    PAGE 527  ROUTINE CAT.TU.INPUT                  4479
    PAGE 636  FUNCTION HE.WLA                       8897
    PAGE 638  FUNCTION ICM.WLA                      8995
    PAGE 676  ''PROGRAM  OLDER.VERSION               375
TU.RADIUS
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES       932
    PAGE  45  ''SECTION FOR DEFINITIONS             2597
    PAGE 219  ROUTINE REQUEST.ILLUM                   44   48   51
    PAGE 223  ROUTINE REQUEST.SMOKE                  264  267  281
    PAGE 241  ROUTINE TARGET.ANALYSIS               1095
    PAGE 263  ROUTINE BTRY.EFFECTS                  2141
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2830
    PAGE 527  ROUTINE CAT.TU.INPUT                  4480
    PAGE 676  ''PROGRAM  OLDER.VERSION               374
    PAGE 704                                        2018
TU.SEQ.NO
    PAGE 527  ROUTINE CAT.TU.INPUT                  4474
TU.SIDE
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES       936
    PAGE 231  ROUTINE SIZE.ESTIMATE                  633
    PAGE 240  ROUTINE TARGET.ANALYSIS               1042
    PAGE 527  ROUTINE CAT.TU.INPUT                  4481
    PAGE 611  ROUTINE OUTPUT.ATTRITION              7933
    PAGE 676  ''PROGRAM  OLDER.VERSION               378
TU.SUP.PRIORITY
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES       937
    PAGE  46  ''SECTION FOR DEFINITIONS             2601
    PAGE 305  ROUTINE UNIT.PRIORITY                 4272
    PAGE 527  ROUTINE CAT.TU.INPUT                  4482
    PAGE 676  ''PROGRAM  OLDER.VERSION               379
    PAGE 704                                        2022
TU.TE.ID
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES      1781
    PAGE  46  ''SECTION FOR DEFINITIONS             2639
    PAGE 183  ROUTINE FA.BN.ASGN                    8367 8388
    PAGE 208  ROUTINE PGM.MSN.ASGN                  9535
    PAGE 231  ROUTINE SIZE.ESTIMATE                  639
    PAGE 480  PROCESS FIRE.MISSION                  2059
    PAGE 527  ROUTINE CAT.TU.INPUT                  4493 4494
    PAGE 690  ''PROGRAM  OLDER.VERSION              1220
    PAGE 705                                        2062
TU.TE.LINK
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES      1780
    PAGE 231  ROUTINE SIZE.ESTIMATE                  638  639  645
    PAGE 527  ROUTINE CAT.TU.INPUT                  4492 4493 4494 4495 4498 4499 4500 4501 4503
    PAGE 690  ''PROGRAM  OLDER.VERSION              1219
TU.TE.LIST
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES       941
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES      1785
    PAGE  49  ''SECTION FOR DEFINITIONS             2829
    PAGE 183  ROUTINE FA.BN.ASGN                    8366
    PAGE 208  ROUTINE PGM.MSN.ASGN                  9534
```

```
    PAGE 231  ROUTINE SIZE.ESTIMATE                    638
    PAGE 480  PROCESS FIRE.MISSION                    2058
    PAGE 527  ROUTINE CAT.TU.INPUT                    4503
    PAGE 676  ''PROGRAM  OLDER.VERSION                 383
    PAGE 691                                          1224
    PAGE 708                                          2250
TU.TE.QUANT
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES        1783
    PAGE 183  ROUTINE FA.BN.ASGN                      8384 8387 8388
    PAGE 209  ROUTINE PGM.MSN.ASGN                    9552
    PAGE 231  ROUTINE SIZE.ESTIMATE                    645
    PAGE 480  PROCESS FIRE.MISSION                    2078
    PAGE 481                                          2085 2092
    PAGE 527  ROUTINE CAT.TU.INPUT                    4498 4501
    PAGE 691  ''PROGRAM  OLDER.VERSION                1222
    PAGE 705                                          2064
TWO.
    PAGE  77  ROUTINE ADJUST                          3928
TW.AC.DET.TIME
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         958
    PAGE 420  PROCESS AC.ATK.TGT                      8934 8939 8940
    PAGE 493  PROCESS SHOOT.OUT                       2778
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4670
    PAGE 676  ''PROGRAM  OLDER.VERSION                 400
TW.BASIC.LOAD
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         966
    PAGE  46  ''SECTION FOR DEFINITIONS               2605
    PAGE 309  ROUTINE AD.SHOOT                        4422 4438
    PAGE 310                                          4495
    PAGE 421  PROCESS AC.ATK.TGT                      8998
    PAGE 446  PROCESS HC.ARRIVE.BATTLE                 288  292  294
    PAGE 487  PROCESS ASSESSMENT                      2464
    PAGE 489                                          2578
    PAGE 490                                          2616
    PAGE 491                                          2670
    PAGE 492                                          2723
    PAGE 499  PROCESS SHOOT.OUT                       3143
    PAGE 500                                          3176 3195
    PAGE 501                                          3217 3245
    PAGE 502                                          3298
    PAGE 516  PROCESS HELICOPTER.FIRE                 4011
    PAGE 517                                          4090
    PAGE 518                                          4149
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4660
    PAGE 676  ''PROGRAM  OLDER.VERSION                 408
    PAGE 704                                          2028
TW.FIRE.OTM.PTR
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         969
    PAGE  44  ''SECTION FOR DEFINITIONS               2496
    PAGE 150  ROUTINE PK.COMPUTE                      7007
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4663
    PAGE 677  ''PROGRAM  OLDER.VERSION                 411
    PAGE 703                                          1917
TW.HFOS
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         974
    PAGE  46  ''SECTION FOR DEFINITIONS               2610
```

```
    PAGE 341  ROUTINE PROB.TIME                          5652
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT                  4668
    PAGE 677  ''PROGRAM  OLDER.VERSION                    416
    PAGE 705                                             2033
TW.HFOV
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES            972
    PAGE  46  ''SECTION FOR DEFINITIONS                  2608
    PAGE 341  ROUTINE PROB.TIME                          5653
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT                  4674
    PAGE 677  ''PROGRAM  OLDER.VERSION                    414
    PAGE 704                                             2031
TW.MAX.RANGE
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES            964
    PAGE  46  ''SECTION FOR DEFINITIONS                  2603
    PAGE 123  ROUTINE TIME.TO.DETECT                     5826
    PAGE 149  ROUTINE PK.COMPUTE                         6947
    PAGE 281  ROUTINE AC.DF.EFFECTS                      3065 3070
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE                  4169 4170
    PAGE 308  ROUTINE AD.SHOOT                           4359 4365
    PAGE 352  EVENT AD.ENGAGEMENT                        6081
    PAGE 422  PROCESS AC.ATK.TGT                         9027
    PAGE 446  PROCESS HC.ARRIVE.BATTLE                    275  277
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION              914
    PAGE 459                                             1025
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT                  4672
    PAGE 676  ''PROGRAM  OLDER.VERSION                    406
    PAGE 704                                             2026
TW.MIN.RANGE
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES            965
    PAGE  46  ''SECTION FOR DEFINITIONS                  2604
    PAGE 123  ROUTINE TIME.TO.DETECT                     5825
    PAGE 149  ROUTINE PK.COMPUTE                         6948
    PAGE 281  ROUTINE AC.DF.EFFECTS                      3064 3070 3071
    PAGE 308  ROUTINE AD.SHOOT                           4358 4365 4366
    PAGE 422  PROCESS AC.ATK.TGT                         9026
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION              912
    PAGE 459                                             1023
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT                  4673
    PAGE 676  ''PROGRAM  OLDER.VERSION                    407
    PAGE 704                                             2027
TW.NAME
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES            959
    PAGE  47  ''SECTION FOR DEFINITIONS                  2700
    PAGE 227  ROUTINE REQUEST.WD.FASCAM                   470
    PAGE 295  ROUTINE END.CAS.MISSION                    3839
    PAGE 309  ROUTINE AD.SHOOT                           4425 4441
    PAGE 382  EVENT OFF.LINE.ATTRITION                   7330
    PAGE 422  PROCESS AC.ATK.TGT                         9023
    PAGE 487  PROCESS ASSESSMENT                         2467
    PAGE 489                                             2581
    PAGE 490                                             2619
    PAGE 491                                             2673
    PAGE 492                                             2726
    PAGE 499  PROCESS SHOOT.OUT                          3146
    PAGE 500                                             3179 3198
    PAGE 501                                             3220 3248
```

```
    PAGE 502                                          3288 3301
    PAGE 512  PROCESS HELICOPTER.FIRE                 3809
    PAGE 514                                          3909
    PAGE 515                                          3972 3997
    PAGE 516                                          4014
    PAGE 517                                          4076 4093
    PAGE 518                                          4135 4152
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4652
    PAGE 535  ROUTINE UNIT.INPUT                      4823
    PAGE 593  ROUTINE AMMO.RPT                        7233
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                 7325
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES             8542
    PAGE 625                                          8572 8602
    PAGE 626                                          8616
    PAGE 676  ''PROGRAM  OLDER.VERSION                 401
    PAGE 706                                          2124
TW.NITE.FAC
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         968
    PAGE  48  ''SECTION FOR DEFINITIONS               2759
    PAGE 150  ROUTINE PK.COMPUTE                      7002
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4662
    PAGE 677  ''PROGRAM  OLDER.VERSION                 410
    PAGE 707                                          2181
TW.NO.SENSORS
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         961
    PAGE 123  ROUTINE TIME.TO.DETECT                  5824 5830
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION           911  922
    PAGE 459                                          1022 1029
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4655
    PAGE 676  ''PROGRAM  OLDER.VERSION                 403
TW.PK.PTR
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         967
    PAGE  44  ''SECTION FOR DEFINITIONS               2493
    PAGE 149  ROUTINE PK.COMPUTE                      6950
    PAGE 281  ROUTINE AC.DF.EFFECTS                   3062
    PAGE 307  ROUTINE AD.SHOOT                        4351
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4661
    PAGE 677  ''PROGRAM  OLDER.VERSION                 409
    PAGE 702                                          1914
TW.RATE.OF.FIRE
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         960
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                 2815
    PAGE 280  ROUTINE AC.DF.EFFECTS                   3049 3051
    PAGE 281                                          3055 3058 3108
    PAGE 421  PROCESS AC.ATK.TGT                      8999 9010
    PAGE 502  PROCESS SHOOT.OUT                       3285 3292
    PAGE 514  PROCESS HELICOPTER.FIRE                 3894
    PAGE 517                                          4115
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4653
    PAGE 676  ''PROGRAM  OLDER.VERSION                 402
    PAGE 704                                          2024
TW.RND.WEIGHT
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         963
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                 2817
    PAGE 280  ROUTINE AC.DF.EFFECTS                   3051
    PAGE 310  ROUTINE AD.SHOOT                        4471 4483
```

```
    PAGE 384  EVENT OFF.LINE.ATTRITION                7419
    PAGE 499  PROCESS SHOOT.OUT                       3105 3112
    PAGE 514  PROCESS HELICOPTER.FIRE                 3924
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4671
    PAGE 676  ''PROGRAM  OLDER.VERSION                 405
TW.ROF.AIR
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         957
    PAGE 502  PROCESS SHOOT.OUT                       3282
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4654
    PAGE 676  ''PROGRAM  OLDER.VERSION                 399
TW.ROUND.VELOCITY
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         962
    PAGE 310  ROUTINE AD.SHOOT                        4501
    PAGE 499  PROCESS SHOOT.OUT                       3138
    PAGE 515  PROCESS HELICOPTER.FIRE                 3949
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4656
    PAGE 676  ''PROGRAM  OLDER.VERSION                 404
    PAGE 704                                          2025
TW.RW
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4642 4657 4671
TW.SEQ.NO
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4651
TW.SPECTRUM
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         971
    PAGE  46  ''SECTION FOR DEFINITIONS               2607
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4665
    PAGE 677  ''PROGRAM  OLDER.VERSION                 413
    PAGE 704                                          2030
TW.TYPE.OF.SENSOR
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES         970
    PAGE  46  ''SECTION FOR DEFINITIONS               2606
    PAGE 120  ROUTINE SEARCH                          5726
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4664
    PAGE 677  ''PROGRAM  OLDER.VERSION                 412
    PAGE 704                                          2029
TW.VFOS
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         975
    PAGE  46  ''SECTION FOR DEFINITIONS               2611
    PAGE 341  ROUTINE PROB.TIME                       5652
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4669
    PAGE 677  ''PROGRAM  OLDER.VERSION                 417
    PAGE 705                                          2034
TW.VFOV
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         973
    PAGE  46  ''SECTION FOR DEFINITIONS               2609
    PAGE 341  ROUTINE PROB.TIME                       5653
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT               4675
    PAGE 677  ''PROGRAM  OLDER.VERSION                 415
    PAGE 704                                          2032
TYPE.
    PAGE  95  ROUTINE INIT.REINF                      4647 4657 4661
    PAGE 403  EVENT START.MOVE                        8182 8194 8201 8201
    PAGE 404                                          8270
    PAGE 405                                          8326
    PAGE 406                                          8362
```

```
TYPE.BATTLE.FIELD
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES          854
    PAGE  60  ROUTINE CREATE.TEAMS                     3169
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                  3827 3839 3841
    PAGE 397  EVENT START.BATTLE                       7863 7864 7865 7866 7867 7868 7869 7870 7871 7872 7873 7882 7883 7884 7885
                                                       7886 7887 7888 7889 7890 7891 7892 7893 7894 7902 7903
    PAGE 398                                           7918 7926 7934
    PAGE 569  ROUTINE TBF.INPUT                        6206 6220 6222
    PAGE 618  ROUTINE SNAP.R                           8247
    PAGE 675  ''PROGRAM  OLDER.VERSION                  296
    PAGE 712                                           2472
TYPE.BTRY
    PAGE   7  ''SECTION FOR PERMANENT_ENTITIES          371
    PAGE   9                                            501
    PAGE  10                                            535
    PAGE  16                                            879  908
    PAGE 473  PROCESS TARGET.REPORT                    1668 1671 1674 1679 1680
    PAGE 548  ROUTINE TB.INPUT                         5320 5322 5325 5326 5336 5337 5338 5339 5340 5341 5342 5343 5344 5345 5346
                                                       5347 5348 5349 5350 5351 5352 5353 5354 5355 5355 5356 5356 5357 5357 5358
                                                       5358
    PAGE 549                                           5372 5381
    PAGE 555  ROUTINE MUNS.INPUT                       5617 5618 5625 5626 5657 5658
    PAGE 556                                           5666 5687 5688 5692 5693 5701
    PAGE 558  ROUTINE HE.LA.INPUT                      5759 5763 5766 5767 5788 5794 5797 5798
    PAGE 560  ROUTINE  RUL.EN.INPUT                    5842
    PAGE 592  ROUTINE AMMO.RPT                         7135
    PAGE 598  ROUTINE ANALYSIS.OUTPUT                  7413
    PAGE 618  ROUTINE SNAP.R                           8248
    PAGE 666  ''PROGRAM  OLDER.VERSION                 9813
    PAGE 668                                           9942
    PAGE 669                                           9976
    PAGE 675                                            321  350
    PAGE 712                                           2473
TYPE.EQUIPMENT
    PAGE   5  ''SECTION FOR PERMANENT_ENTITIES          264
    PAGE  10                                            522
    PAGE  12                                            657  679
    PAGE  13                                            693
    PAGE  16                                            911
    PAGE  17                                            922
    PAGE 250  ROUTINE MINE.EFFECTS                     1497
    PAGE 429  PROCESS AIR.OBSERVER                     9411
    PAGE 530  ROUTINE EQ.TE.INPUT                      4580 4581
    PAGE 538  ROUTINE MFO.INPUT                        4970
    PAGE 557  ROUTINE SUBM.INPUT                       5741 5742 5743 5745
    PAGE 558  ROUTINE HE.LA.INPUT                      5791 5792 5793
    PAGE 559                                           5811 5812 5814
    PAGE 582  ROUTINE TACAIR.INPUT                     6778 6779 6785
    PAGE 588  ROUTINE AC.MUNS.INPUT                    7040 7041 7042 7044
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                  7291
    PAGE 618  ROUTINE SNAP.R                           8249
    PAGE 664  ''PROGRAM  OLDER.VERSION                 9706
    PAGE 669                                           9963
    PAGE 671                                             99
    PAGE 672                                            122  136
    PAGE 676                                            353  364
```

```
    PAGE 712                                        2474
TYPE.MOVE
    PAGE  28  ''SECTION FOR TEMPORARY_ENTITIES      1596
    PAGE  47  ''SECTION FOR DEFINITIONS             2701
    PAGE 415  EVENT ACT.MOVCOR                      8761
    PAGE 546  ROUTINE ORD.MOVCOR                    5237
    PAGE 687  ''PROGRAM  OLDER.VERSION              1035
    PAGE 706                                        2125
TYPE.ORDER
    PAGE 540  ROUTINE READ.ORDERS                   5043
TYPE.ORDER.
    PAGE 539  ROUTINE READ.ORDERS                   4996
    PAGE 540                                        5041 5049 5052
    PAGE 541                                        5099
TYPE.RANGE
    PAGE  71  ROUTINE ORIENTATION                   3648
    PAGE  72                                        3724 3727 3729 3730
    PAGE  73                                        3796
TYPE.TEAM
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES      1791
    PAGE  60  ROUTINE CREATE.TEAMS                  3165 3167 3168 3169 3179
    PAGE  67  ROUTINE GENERAL.BATTLE                3461 3465 3466 3468 3471 3472
    PAGE  68                                        3527 3528 3529 3530 3531 3532 3539 3548 3559 3560 3561 3562 3563
    PAGE  69                                        3564 3571 3580 3596 3598 3599 3601 3608 3615 3617
    PAGE  70                                        3624
    PAGE  72  ROUTINE ORIENTATION                   3695 3697 3701 3702 3704 3705 3711 3712 3714 3715
    PAGE  75  ROUTINE UNIT.ASSIGNMENT               3847
    PAGE 395  EVENT START.BATTLE                    7757
    PAGE 398                                        7918 7920
    PAGE 691  ''PROGRAM  OLDER.VERSION              1230
TYPE.UN
    PAGE 231  ROUTINE SIZE.ESTIMATE                  625  633  638  648  673
    PAGE 239  ROUTINE TARGET.ANALYSIS                984
    PAGE 240                                        1041 1042 1043 1048 1066 1069
TYPE.UNIT
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES       392
    PAGE  17                                         925
    PAGE 527  ROUTINE CAT.TU.INPUT                  4471 4472 4475 4476 4477 4478 4479 4480 4481 4482 4483 4484 4485 4486 4487
                                                    4487 4501 4503
    PAGE 528                                        4516
    PAGE 533  ROUTINE UNIT.INPUT                    4699 4701 4702
    PAGE 560  ROUTINE  RUL.EN.INPUT                 5848
    PAGE 595  ROUTINE ANALYSIS.OUTPUT               7272
    PAGE 618  ROUTINE SNAP.R                        8254
    PAGE 667  ''PROGRAM  OLDER.VERSION              9834
    PAGE 676                                         367
    PAGE 712                                        2475
TYPE.U.PTR
    PAGE 130  ROUTINE CHECK.DEAD                    6099 6100
TYPE.WEAPON
    PAGE  15  ''SECTION FOR PERMANENT_ENTITIES       813
    PAGE  17                                         956
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES      1876
    PAGE 227  ROUTINE REQUEST.WD.FASCAM              469
    PAGE 382  EVENT OFF.LINE.ATTRITION              7329
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT             4647 4649
```

```
    PAGE 535  ROUTINE UNIT.INPUT                    4821 4823 4824
    PAGE 593  ROUTINE AMMO.RPT                      7231
    PAGE 596  ROUTINE ANALYSIS.OUTPUT               7320
    PAGE 618  ROUTINE SNAP.R                        8255
    PAGE 674  ''PROGRAM  OLDER.VERSION               255
    PAGE 676                                         398
    PAGE 692                                        1315
    PAGE 712                                        2476
TYPE.WEAPON.INPUT
    PAGE 520  ROUTINE MAIN2                         4208 4210
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT             4638
TYP.MOV.
    PAGE 408  EVENT UPDATE.LOC                      8410 8428 8430
    PAGE 409                                        8488
    PAGE 411                                        8591
T.
    PAGE 664  ''PROGRAM  OLDER.VERSION              9671 9684
T.DELAY
    PAGE 373  EVENT GET.NX.ORD                      6889
    PAGE 374                                        6938 6958 6978 6982
T.EQPS
    PAGE 260  ROUTINE BTRY.EFFECTS                  1987
    PAGE 268                                        2455 2460 2461 2462 2464 2467 2469
T.FAC
    PAGE 260  ROUTINE BTRY.EFFECTS                  1991
    PAGE 267                                        2373 2375 2378
T.ST.DEV
    PAGE 242  ROUTINE UNIT.ENVIR                    1139
    PAGE 243                                        1209
    PAGE 244                                        1264
T.TYPE
    PAGE 115  ROUTINE PROX.CHECK                    5527 5528
UE.CRITICAL.EQUIP.INDIC
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES      1810
    PAGE 252  ROUTINE MINE.EFFECTS                  1575
    PAGE 270  ROUTINE BTRY.EFFECTS                  2568
    PAGE 278  ROUTINE AC.BOMB.EFFECTS               2949
    PAGE 282  ROUTINE AC.DF.EFFECTS                 3132
    PAGE 283  ROUTINE CAS.EVAL                      3197
    PAGE 324  ROUTINE DECIDE                        5042
    PAGE 384  EVENT OFF.LINE.ATTRITION              7400
    PAGE 424  PROCESS AC.ATK.TGT                    9149
    PAGE 436  PROCESS ARTY.ASSESS                   9751 9795
    PAGE 437                                        9833
    PAGE 465  PROCESS MINE.ASSESS                   1291 1311
    PAGE 492  PROCESS ASSESSMENT                    2737
    PAGE 516  PROCESS HELICOPTER.FIRE               4056
    PAGE 535  ROUTINE UNIT.INPUT                    4798 4805
    PAGE 537                                        4919
    PAGE 691  ''PROGRAM  OLDER.VERSION              1249
UE.ID
    PAGE  26  ''SECTION FOR TEMPORARY_ENTITIES      1469
    PAGE  32                                        1809
    PAGE  46  ''SECTION FOR DEFINITIONS             2641
    PAGE 123  ROUTINE TIME.TO.DETECT                5836
    PAGE 149  ROUTINE PK.COMPUTE                    6940 6944 6945
```

```
PAGE 154  ROUTINE AO.DETECTION                  7126
PAGE 155                                        7143
PAGE 157                                        7259
PAGE 183  ROUTINE FA.BN.ASGN                    8376 8387 8388
PAGE 208  ROUTINE PGM.MSN.ASGN                  9544
PAGE 212  ROUTINE PIR.DETECTION                 9745
PAGE 227  ROUTINE REQUEST.WD.FASCAM              457
PAGE 229  ROUTINE RPV.DETECTION                  558
PAGE 251  ROUTINE MINE.EFFECTS                  1559 1560
PAGE 252                                        1584 1594
PAGE 253                                        1637 1645 1653
PAGE 255  ROUTINE FO.DETECTION                  1767
PAGE 264  ROUTINE BTRY.EFFECTS                  2209 2226 2227 2232
PAGE 268                                        2444 2447 2457
PAGE 269                                        2518
PAGE 270                                        2552 2563
PAGE 271                                        2628 2649 2649
PAGE 272                                        2660 2660
PAGE 277  ROUTINE AC.BOMB.EFFECTS               2890
PAGE 279                                        2973 2979
PAGE 280  ROUTINE AC.DF.EFFECTS                 3021 3046
PAGE 283  ROUTINE CAS.EVAL                      3194
PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS         3298 3328
PAGE 286                                        3349
PAGE 289  ROUTINE EMPLOY.HELICOPTERS            3499 3502 3508 3511
PAGE 295  ROUTINE END.CAS.MISSION               3823 3828
PAGE 297  ROUTINE FARRP.CHECK                   3877 3878 3884 3885
PAGE 298  ROUTINE HC.COMPUTE.TIMES              3960 3960 3962
PAGE 302  ROUTINE HEL.RANGE.COMPUTE             4161
PAGE 304  ROUTINE REPLACE.HC                    4202
PAGE 307  ROUTINE AD.SHOOT                      4331 4337
PAGE 352  EVENT AD.ENGAGEMENT                   6073
PAGE 370  EVENT ENGAGEMENT                      6743
PAGE 376  EVENT HELO.ENGAGEMENT                 7079
PAGE 377                                        7098 7115
PAGE 378                                        7155 7184 7192 7203
PAGE 382  EVENT OFF.LINE.ATTRITION              7320
PAGE 384                                        7425
PAGE 392  EVENT SET.DEBUG                       7644
PAGE 420  PROCESS AC.ATK.TGT                    8926 8927 8950
PAGE 435  PROCESS ARTY.ASSESS                   9712
PAGE 437                                        9843
PAGE 445  PROCESS HC.ARRIVE.BATTLE               266
PAGE 452  PROCESS HC.RETURN.FARRP                596  598  604  605
PAGE 457  PROCESS HEL.TARGET.ACQUISITION         900
PAGE 458                                         928
PAGE 461                                        1106
PAGE 463  PROCESS HOW.REPAIR                    1197
PAGE 464  PROCESS MINE.ASSESS                   1251
PAGE 465                                        1322
PAGE 466                                        1339
PAGE 480  PROCESS FIRE.MISSION                  2069
PAGE 487  PROCESS ASSESSMENT                    2461 2462 2465 2466 2468
PAGE 488                                        2469 2483
PAGE 489                                        2550 2579 2580 2582 2583
PAGE 490                                        2596 2597 2613 2614 2617 2618 2620 2621
```

```
    PAGE 491                                        2658 2659 2671 2672 2674 2675
    PAGE 492                                        2705 2706 2715 2716 2724 2725 2727 2728
    PAGE 494  PROCESS SHOOT.OUT                     2846
    PAGE 495                                        2891
    PAGE 497                                        3021 3037
    PAGE 498                                        3082
    PAGE 499                                        3111 3114 3118 3121 3128 3144 3145 3147 3148
    PAGE 500                                        3177 3178 3180 3181 3196 3197 3199 3200
    PAGE 501                                        3213 3218 3219 3221 3222 3246 3247 3249 3250
    PAGE 502                                        3299 3300 3302 3303
    PAGE 504  PROCESS CAS.MISSION                   3375
    PAGE 508                                        3591
    PAGE 510  PROCESS HELICOPTER.FIRE               3680
    PAGE 511                                        3760 3761
    PAGE 514                                        3910 3920 3930 3935 3941
    PAGE 515                                        3974 3999
    PAGE 516                                        4012 4013 4015 4016 4061 4062
    PAGE 517                                        4078 4094 4095 4110
    PAGE 518                                        4137 4153 4154
    PAGE 534  ROUTINE UNIT.INPUT                    4771 4787 4791
    PAGE 535                                        4801
    PAGE 537                                        4918 4921
    PAGE 551  ROUTINE BTRY.INPUT                    5463
    PAGE 610  ROUTINE KV.SCOREBOARD                 7908 7909
    PAGE 611  ROUTINE OUTPUT.ATTRITION              7941 7957
    PAGE 656  ROUTINE AR.DETECTION                  9404
    PAGE 685  ''PROGRAM  OLDER.VERSION               908
    PAGE 691                                        1248
    PAGE 705                                        2065
UE.LINK
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES      1808
    PAGE 108  ROUTINE NEW.SEGMENT                   5208 5211 5215
    PAGE 109                                        5220 5225 5228 5232 5237
    PAGE 282  ROUTINE AC.DF.EFFECTS                 3157 3158
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS         3326
    PAGE 329  ROUTINE EMPTY                         5247 5249 5253 5256 5258 5260
    PAGE 398  EVENT START.BATTLE                    7968
    PAGE 419  PROCESS AC.ATK.TGT                    8889
    PAGE 423                                        9088 9120
    PAGE 425                                        9196 9227
    PAGE 426                                        9271
    PAGE 445  PROCESS HC.ARRIVE.BATTLE               265  266
    PAGE 446                                         274  285
    PAGE 459  PROCESS HEL.TARGET.ACQUISITION        1016 1018 1020 1027
    PAGE 496  PROCESS SHOOT.OUT                     2924 2926 2931
    PAGE 534  ROUTINE UNIT.INPUT                    4760 4763
    PAGE 537                                        4916 4918 4919 4920 4921 4923
    PAGE 588  ROUTINE AC.MUNS.INPUT                 7052 7054 7055 7062 7063
    PAGE 619  ROUTINE SNAP.R                        8269 8272 8274
    PAGE 691  ''PROGRAM  OLDER.VERSION              1247
UE.LINK.
    PAGE 619  ROUTINE SNAP.R                        8265
UE.QUANT
    PAGE   3  PROGRAM REVISIONS                      169
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES      1811
    PAGE  44  ''SECTION FOR DEFINITIONS             2502
```

PAGE 123	ROUTINE TIME.TO.DETECT	5817 5820 5828
PAGE 154	ROUTINE AO.DETECTION	7124
PAGE 157		7257
PAGE 183	ROUTINE FA.BN.ASGN	8384 8387 8388
PAGE 209	ROUTINE PGM.MSN.ASGN	9552
PAGE 212	ROUTINE PIR.DETECTION	9742 9744
PAGE 227	ROUTINE REQUEST.WD.FASCAM	458 461
PAGE 229	ROUTINE RPV.DETECTION	555 557
PAGE 251	ROUTINE MINE.EFFECTS	1557 1563 1572 1573
PAGE 252		1593
PAGE 253		1647
PAGE 255	ROUTINE FO.DETECTION	1765 1775
PAGE 264	ROUTINE BTRY.EFFECTS	2211 2224 2243
PAGE 268		2460 2464 2467 2469
PAGE 269		2486 2496 2541
PAGE 270		2557 2557
PAGE 271		2644 2652
PAGE 272		2663
PAGE 277	ROUTINE AC.BOMB.EFFECTS	2887
PAGE 278		2920 2921 2929 2932 2945
PAGE 280	ROUTINE AC.DF.EFFECTS	3020
PAGE 281		3111
PAGE 282		3116 3130
PAGE 283	ROUTINE CAS.EVAL	3195 3198 3199
PAGE 285	ROUTINE CHECK.CAS.CONSTRAINTS	3305 3307 3318 3321 3324
PAGE 286		3343 3361 3361 3373 3373 3374 3375 3379 3380 3386
PAGE 287		3405
PAGE 289	ROUTINE EMPLOY.HELICOPTERS	3494 3500 3501 3509 3510
PAGE 304	ROUTINE REPLACE.HC	4203 4207 4209
PAGE 308	ROUTINE AD.SHOOT	4411
PAGE 310		4494
PAGE 369	EVENT ENGAGEMENT	6715 6719 6731
PAGE 370		6742
PAGE 377	EVENT HELO.ENGAGEMENT	7090
PAGE 378		7177
PAGE 379		7211
PAGE 383	EVENT OFF.LINE.ATTRITION	7359 7366 7367 7369 7370
PAGE 384		7399
PAGE 392	EVENT SET.DEBUG	7644
PAGE 424	PROCESS AC.ATK.TGT	9150
PAGE 435	PROCESS ARTY.ASSESS	9717
PAGE 436		9744 9766 9776 9788
PAGE 437		9826
PAGE 452	PROCESS HC.RETURN.FARRP	597
PAGE 457	PROCESS HEL.TARGET.ACQUISITION	919
PAGE 459		1018 1027
PAGE 463	PROCESS HOW.REPAIR	1202 1206
PAGE 464	PROCESS MINE.ASSESS	1252 1253 1256 1260
PAGE 465		1277 1289 1308 1309 1325
PAGE 480	PROCESS FIRE.MISSION	2078
PAGE 481		2085 2092
PAGE 491	PROCESS ASSESSMENT	2668
PAGE 492		2721
PAGE 493	PROCESS SHOOT.OUT	2788
PAGE 494		2863
PAGE 498		3077

```
    PAGE 501                                        3233
    PAGE 507  PROCESS CAS.MISSION                   3566 3570 3578 3579
    PAGE 511  PROCESS HELICOPTER.FIRE               3729
    PAGE 512                                        3796
    PAGE 514                                        3903
    PAGE 515                                        3991
    PAGE 516                                        4030 4037 4041 4055
    PAGE 535  ROUTINE UNIT.INPUT                    4797 4803 4807
    PAGE 537                                        4920
  * PAGE 551  ROUTINE BTRY.INPUT                    5485 5488
    PAGE 596  ROUTINE ANALYSIS.OUTPUT               7317
    PAGE 610  ROUTINE KV.SCOREBOARD                 7909
    PAGE 611  ROUTINE OUTPUT.ATTRITION              7984
    PAGE 656  ROUTINE AR.DETECTION                  9402 9417
    PAGE 691  ''PROGRAM  OLDER.VERSION              1250
    PAGE 703                                        1923
UE.TARGET.LIST
    PAGE  25  ''SECTION FOR TEMPORARY_ENTITIES      1379
    PAGE  32                                        1815
    PAGE  81  ROUTINE BLOCK.LOS                     4095 4102 4109 4116
    PAGE 108  ROUTINE NEW.SEGMENT                   5211
    PAGE 109                                        5220 5228 5237
    PAGE 293  ROUTINE END.CAS.MISSION               3719 3722
    PAGE 300  ROUTINE HC.DISENGAGE                  4056 4059
    PAGE 301                                        4090 4095
    PAGE 312  ROUTINE INTER.HELO                    4597 4602
    PAGE 329  ROUTINE EMPTY                         5249 5253 5256
    PAGE 369  EVENT ENGAGEMENT                      6724
    PAGE 377  EVENT HELO.ENGAGEMENT                 7095
    PAGE 378                                        7183 7199
    PAGE 419  PROCESS AC.ATK.TGT                    8892
    PAGE 423                                        9091 9123
    PAGE 425                                        9199 9230
    PAGE 426                                        9274
    PAGE 461  PROCESS HEL.TARGET.ACQUISITION        1102 1108 1116 1118
    PAGE 489  PROCESS ASSESSMENT                    2567 2571
    PAGE 491                                        2647 2650
    PAGE 494  PROCESS SHOOT.OUT                     2807 2813 2821
    PAGE 495                                        2875 2877
    PAGE 496                                        2977
    PAGE 497                                        2994
    PAGE 498                                        3052 3062 3070
    PAGE 501                                        3214 3234 3240
    PAGE 503                                        3335
    PAGE 507  PROCESS CAS.MISSION                   3573
    PAGE 511  PROCESS HELICOPTER.FIRE               3722 3727 3735 3737
    PAGE 513                                        3847 3858
    PAGE 514                                        3902
    PAGE 619  ROUTINE SNAP.R                        8272
    PAGE 684  ''PROGRAM  OLDER.VERSION               818
    PAGE 691                                        1254
UE.TARGET.LISTS
    PAGE 300  ROUTINE HC.DISENGAGE                  4051
    PAGE 312  ROUTINE INTER.HELO                    4591
UE.WEAPON.SET
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES      1817
```

```
    PAGE  33                                    1882
    PAGE 123  ROUTINE TIME.TO.DETECT            5822
    PAGE 149  ROUTINE PK.COMPUTE                6943
    PAGE 295  ROUTINE END.CAS.MISSION           3835
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE         4168
    PAGE 377  EVENT HELO.ENGAGEMENT             7117
    PAGE 420  PROCESS AC.ATK.TGT                8938
    PAGE 421                                    8979
    PAGE 446  PROCESS HC.ARRIVE.BATTLE           274  285
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION     908
    PAGE 459                                    1020
    PAGE 535  ROUTINE UNIT.INPUT                4848
    PAGE 588  ROUTINE AC.MUNS.INPUT             7055 7063
    PAGE 596  ROUTINE ANALYSIS.OUTPUT           7316
    PAGE 611  ROUTINE OUTPUT.ATTRITION          7944
    PAGE 691  ''PROGRAM  OLDER.VERSION          1256
    PAGE 692                                    1321
UL.ORDER
    PAGE  41  ''SECTION FOR EVENTS              2336
    PAGE 700  ''PROGRAM  OLDER.VERSION          1773
UL.RATE.MOVE
    PAGE  41  ''SECTION FOR EVENTS              2334
    PAGE 700  ''PROGRAM  OLDER.VERSION          1771
UL.TYP.MOV
    PAGE  41  ''SECTION FOR EVENTS              2335
    PAGE  47  ''SECTION FOR DEFINITIONS         2702
    PAGE 700  ''PROGRAM  OLDER.VERSION          1772
    PAGE 706                                    2126
UL.UNIT
    PAGE  41  ''SECTION FOR EVENTS              2331
    PAGE 104  ROUTINE MINE.DELAY                5047
    PAGE 112  ROUTINE PREPARE.LIST              5386
    PAGE 630  FUNCTION COLLISION                8685
    PAGE 700  ''PROGRAM  OLDER.VERSION          1768
UL.XCOR
    PAGE  41  ''SECTION FOR EVENTS              2332
    PAGE  44  ''SECTION FOR DEFINITIONS         2513
    PAGE 700  ''PROGRAM  OLDER.VERSION          1769
    PAGE 703                                    1934
UL.YCOR
    PAGE  41  ''SECTION FOR EVENTS              2333
    PAGE  44  ''SECTION FOR DEFINITIONS         2514
    PAGE 700  ''PROGRAM  OLDER.VERSION          1770
    PAGE 703                                    1935
UNIFORM.F
    PAGE 115  ROUTINE PROX.CHECK                5532
    PAGE 206  ROUTINE PDB.DETECTION             9474
    PAGE 243  ROUTINE UNIT.ENVIR                1220
    PAGE 251  ROUTINE MINE.EFFECTS              1561
    PAGE 278  ROUTINE AC.BOMB.EFFECTS           2918
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE         4154 4172 4175
    PAGE 310  ROUTINE AD.SHOOT                  4502
    PAGE 362  EVENT CFR.OPERATOR                6492
    PAGE 371  EVENT FEBA.SORTIE                 6827
    PAGE 372                                    6869 6873
    PAGE 387  EVENT PDB.OPERATOR                7516
```

```
    PAGE 422  PROCESS AC.ATK.TGT                    9025
    PAGE 428  PROCESS AIR.OBSERVER                  9381
    PAGE 429                                        9397
    PAGE 432                                        9562
    PAGE 440  PROCESS FORWARD.OBSERVER                26
    PAGE 447  PROCESS HC.ARRIVE.BATTLE               366
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION         729  731  735  737
    PAGE 461                                        1136 1139
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1390
    PAGE 468                                        1453
    PAGE 475  PROCESS TARGET.REPORT                 1808
    PAGE 482  PROCESS FIRE.MISSION                  2149
    PAGE 504  PROCESS CAS.MISSION                   3397
    PAGE 568  ROUTINE SENSOR.INPUT                  6190
    PAGE 585  ROUTINE TACAIR.INPUT                  6951
    PAGE 653  PROCESS AIRBORNE.RADAR                9293
    PAGE 654                                        9364
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9554 9566
UNIT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           47
    PAGE   2  PROGRAM REVISIONS                      109
    PAGE  17  ''SECTION FOR PERMANENT_ENTITIES       935
    PAGE  18                                         977  981  986  987 1000
    PAGE  19                                        1066 1066
    PAGE  38  ''SECTION FOR PROCESSES               2153 2154 2164
    PAGE  56  ROUTINE MAIN1                         3043
    PAGE  58  ROUTINE CREATE.FORCE                  3137 3139
    PAGE  61  ROUTINE FEBA.INITIAL                  3187 3209 3210 3211 3212 3214 3216 3216 3216 3217 3220 3222 3222 3222 3223
                                                    3225 3226 3226 3227 3228 3229 3232 3233 3236 3236 3237
    PAGE  62                                        3255
    PAGE  65  ROUTINE FILE.KAD.SENSOR               3371
    PAGE  66  ROUTINE FORM.TF.LIST                  3420 3420 3422 3423 3425 3426 3427 3427 3430 3431 3432 3433 3434 3435 3438
                                                    3439 3440 3440 3441 3441 3442 3443
    PAGE  67  ROUTINE GENERAL.BATTLE                3457
    PAGE  68                                        3525 3526 3557 3558
    PAGE  77  ROUTINE ADJUST                        3924 3927 3933
    PAGE  83  ROUTINE CHANGE.LOC                    4126 4135 4136 4138 4140 4141 4143 4144 4147 4148 4149 4153 4154 4158 4159
                                                    4162 4167 4167 4169 4170 4171 4175 4177 4177
    PAGE  84                                        4182 4189 4194 4198 4198 4199 4199 4200 4201 4203 4207 4208 4210 4218 4223
                                                    4226 4231 4233 4234 4236 4237
    PAGE  85                                        4242 4246 4253 4258 4262 4263 4264 4267 4271 4272 4274 4280 4281 4282 4285
                                                    4289
    PAGE  86                                        4300 4303
    PAGE  87  ROUTINE END.MOVE                      4314 4315
    PAGE  88                                        4390
    PAGE  90  ROUTINE FA.BN.MOVEMENT                4469 4472 4488 4491
    PAGE  91                                        4547 4550 4561 4564
    PAGE  92                                        4578 4581
    PAGE  93  ROUTINE INITIAL.DETECT                4600 4603 4606
    PAGE  94  ROUTINE INITIAL.MOVE                  4625 4627 4628 4632
    PAGE  95  ROUTINE INIT.REINF                    4641 4643
    PAGE 100  ROUTINE LOCATE.SECTOR                 4852 4859
    PAGE 101  ROUTINE LOS.CHECK                     4873 4877 4878 4887 4890 4891 4894 4896 4901 4903 4908 4914 4920
    PAGE 102                                        4929
    PAGE 103  ROUTINE MINE.DELAY                    4962 4962
    PAGE 104                                        5015 5026 5040
```

```
PAGE 106  ROUTINE MIN.MOVE                    5062 5072 5078 5079 5083 5089 5093 5094 5097
PAGE 107  ROUTINE NEW.SEGMENT                 5104 5115 5121 5124 5127 5132 5135 5146 5147 5156
PAGE 108                                      5165 5173 5176 5181 5193 5197 5201 5212
PAGE 109                                      5225 5241 5242 5245 5246 5248 5254
PAGE 110  ROUTINE POSITION                    5266 5276 5277 5279 5280 5283 5287 5292
PAGE 111  ROUTINE PRED.POS                    5301 5311 5312 5318 5319 5324 5326 5332 5333 5334
PAGE 112  ROUTINE PREPARE.LIST                5373
PAGE 113                                      5406
PAGE 114  ROUTINE PREP.WITHDRAW               5464 5469 5470 5471 5475 5476 5477 5479 5480 5482 5491
PAGE 115  ROUTINE PROX.CHECK                  5503 5504 5508 5548
PAGE 117  ROUTINE PROX.POS                    5570 5572
PAGE 118  ROUTINE REIN.ARRIVE                 5603 5604 5605 5607
PAGE 119  ROUTINE RESET.FEBA.SECTOR           5654 5675 5676 5677 5678 5679 5683 5686 5687 5689 5693 5697
PAGE 122  ROUTINE SEGMENT.ADJUST              5775 5781 5787
PAGE 125  ROUTINE WHAT.NEXT                   5880
PAGE 126                                      5952 5970
PAGE 130  ROUTINE CHECK.DEAD                  6084 6101 6107
PAGE 132  ROUTINE CHECK.ENGAGEMENT            6133 6137 6144 6148
PAGE 133  ROUTINE CHECK.FORCE                 6169 6171 6172 6174 6187 6188 6189 6190 6194 6197
PAGE 134  ROUTINE CHECK.FOR.MINES             6235
PAGE 135                                      6289
PAGE 137  ROUTINE CHECK.LIST                  6335 6361
PAGE 138  ROUTINE CHECK.PROX                  6377 6378 6379
PAGE 142  ROUTINE DEAD.UNIT                   6527 6528 6528 6530 6535 6539 6545 6556 6556 6558
PAGE 143                                      6587 6629 6629
PAGE 144                                      6639 6640 6641 6657 6680
PAGE 146  ROUTINE INTER.BATTLE                6742 6774
PAGE 154  ROUTINE AO.DETECTION                7115
PAGE 159  ROUTINE ATTRIT.SENSOR               7318 7322 7323 7334 7335 7342 7344
PAGE 162                                      7498 7501 7503
PAGE 163  ROUTINE BTRY.FM.DEQ                 7515 7532 7535
PAGE 164  ROUTINE BTRY.FM.ENQ                 7557 7560
PAGE 173  ROUTINE DUST.EFFECTS                7901 7936
PAGE 174                                      7962
PAGE 180  ROUTINE EST.MIL.WORTH               8228 8230
PAGE 182  ROUTINE FA.BN.ASGN                  8295
PAGE 183                                      8371 8380
PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ        9341
PAGE 208  ROUTINE PGM.MSN.ASGN                9539
PAGE 209                                      9548
PAGE 212  ROUTINE PIR.DETECTION               9724
PAGE 217  ROUTINE REQUEST.FASCAM              9945
PAGE 219  ROUTINE REQUEST.ILLUM                 75
PAGE 221                                        166
PAGE 223  ROUTINE REQUEST.SMOKE                 248  261
PAGE 225                                        385
PAGE 226  ROUTINE REQUEST.WD.FASCAM             396  421
PAGE 229  ROUTINE RPV.DETECTION                 537  539
PAGE 230                                        604
PAGE 234  ROUTINE SMOKE.EFFECTS                 787
PAGE 238  ROUTINE SWITCH.FO                     923  927  929
PAGE 239  ROUTINE TARGET.ANALYSIS               979  980  997
PAGE 240                                       1071 1075 1084
PAGE 241                                       1091 1091
PAGE 242  ROUTINE UNIT.ENVIR                   1122 1124 1125 1125 1130 1131 1132 1147 1163
PAGE 243                                       1205
```

```
PAGE 244                                    1262
PAGE 248  ROUTINE WEIGHTED.VOLLEYS          1381
PAGE 251  ROUTINE MINE.EFFECTS              1541 1551
PAGE 252                                    1590 1600
PAGE 253                                    1648 1666
PAGE 254  ROUTINE FO.DETECTION              1697
PAGE 257                                    1863 1871
PAGE 260  ROUTINE BTRY.EFFECTS              1985 2003 2011 2013 2018
PAGE 261                                    2026 2029 2042 2063 2064 2065 2066 2068 2077
PAGE 262                                    2081 2086 2087 2094 2108 2118 2137
PAGE 263                                    2139 2141 2142 2159
PAGE 264                                    2208 2223 2243 2253
PAGE 265                                    2260 2266 2283 2290 2294 2306 2309
PAGE 266                                    2319 2328 2336
PAGE 267                                    2393
PAGE 268                                    2458
PAGE 270                                    2561 2568 2569 2582 2585 2587 2598 2599 2600
PAGE 271                                    2621 2622 2627 2656
PAGE 272                                    2667 2674 2679 2680 2685 2686 2693 2697 2706 2708 2709 2716 2717
PAGE 273                                    2718 2722 2724
PAGE 280  ROUTINE AC.DF.EFFECTS             3028
PAGE 286  ROUTINE CHECK.CAS.CONSTRAINTS     3395
PAGE 293  ROUTINE END.CAS.MISSION           3704
PAGE 295                                    3804 3814 3818
PAGE 298  ROUTINE HC.COMPUTE.TIMES          3941 3943 3944
PAGE 300  ROUTINE HC.DISENGAGE              4051
PAGE 302  ROUTINE HEL.RANGE.COMPUTE         4128
PAGE 305  ROUTINE UNIT.PRIORITY             4270 4272 4283 4285
PAGE 309  ROUTINE AD.SHOOT                  4433 4449
PAGE 312  ROUTINE INTER.HELO                4591
PAGE 321  ROUTINE COMPUTE.D                 4944 4953 4954 4958 4959
PAGE 322  ROUTINE COMPUTE.WD                4966 4974 4975 4976 4983 4984 4990 4992 4993
PAGE 324  ROUTINE DECIDE                    5068
PAGE 325  ROUTINE DEQ.FEBA.SET              5079 5086 5087 5089 5090 5091 5092 5093 5093 5096 5097 5103 5106
PAGE 328  ROUTINE EMPTY                     5176
PAGE 329                                    5235 5237 5238 5239 5241 5243 5247 5269 5271 5275 5277 5280
PAGE 330                                    5293
PAGE 331  ROUTINE ENQ.FEBA.SET              5302 5308 5309 5311 5312 5313 5314 5317 5318 5319 5322 5323 5325 5327 5335
PAGE 343  ROUTINE SEARCH.COVERAGE           5708 5717
PAGE 345  ROUTINE TERM.CHECK                5782 5783 5784
PAGE 347  EVENT ACT.REINF                   5810 5811 5813 5814 5816 5817 5823 5829 5834 5835 5836 5837 5838 5842 5847
                                            5858
PAGE 348                                    5861 5866 5867 5867 5873 5876 5884
PAGE 352  EVENT AD.ENGAGEMENT               6077
PAGE 355  EVENT ARTY.OCCUPATION             6238 6241 6244 6245
PAGE 356  EVENT BTL.ENDED                   6305 6305
PAGE 357                                    6313 6313 6321 6321
PAGE 360  EVENT CFR.ON                      6455
PAGE 362  EVENT CFR.OPERATOR                6514
PAGE 369  EVENT ENGAGEMENT                  6695 6696
PAGE 373  EVENT GET.NX.ORD                  6904
PAGE 376  EVENT HELO.ENGAGEMENT             7055 7055
PAGE 378                                    7165
PAGE 381  EVENT MOVE                        7269 7276
PAGE 382  EVENT OFF.LINE.ATTRITION          7287 7289 7293 7306 7315 7324 7325 7334
PAGE 383                                    7361 7368
```

PAGE 385		7461
PAGE 387	EVENT PDB.OPERATOR	7532
PAGE 392	EVENT SET.DEBUG	7637 7638 7640 7640 7642 7643 7645 7646
PAGE 393	EVENT START.ARTY.MOVEMENT	7707
PAGE 394		7723 7726
PAGE 395	EVENT START.BATTLE	7756 7789
PAGE 398		7939 7946 7948 7949 7950 7952 7954 7955 7958 7959 7959 7961 7964 7965 7973 7974
PAGE 399		7978 7979 7980 7981 7983 7984 7986 7987 7988 7989 7995 7996 7997 8000 8000 8001 8001 8002 8002
PAGE 403	EVENT START.MOVE	8186 8189 8190 8209 8212
PAGE 404		8273
PAGE 405		8319
PAGE 407	EVENT STOP.ARTY.MOVEMENT	8388 8391
PAGE 408	EVENT UPDATE.LOC	8418 8421 8454
PAGE 414	EVENT ACT.DEF	8720
PAGE 415	EVENT ACT.MOVCOR	8742 8743 8745
PAGE 416	EVENT ACT.MOVDIS	8773
PAGE 417	EVENT DYNAMIC.ANALYSIS.REPORT	8810
PAGE 421	PROCESS AC.ATK.TGT	8969 9005 9014
PAGE 422		9033 9036 9071
PAGE 435	PROCESS ARTY.ASSESS	9691 9700 9721
PAGE 436		9750 9753 9794 9797
PAGE 437		9832 9834 9841
PAGE 441	PROCESS FORWARD.OBSERVER	54
PAGE 444	PROCESS HC.ARRIVE.BATTLE	168 171 172 175 176 195 203 205 206
PAGE 445		209 210
PAGE 447		326
PAGE 448		412 413 414 415
PAGE 452	PROCESS HC.RETURN.FARRP	582 591
PAGE 456	PROCESS HEL.TARGET.ACQUISITION	833
PAGE 457		895
PAGE 459		1012
PAGE 464	PROCESS MINE.ASSESS	1223 1241
PAGE 465		1326
PAGE 466		1351
PAGE 467	PROCESS REMOTE.PILOT.VEHICLE	1365 1376 1377 1382
PAGE 471	PROCESS TARGET.REPORT	1568
PAGE 474		1714
PAGE 476		1873 1884
PAGE 478	PROCESS WITH.DRAW	1922 1938 1959 1961 1964 1969
PAGE 479		1982 1991 1992
PAGE 480	PROCESS FIRE.MISSION	2031 2063 2073
PAGE 481		2137
PAGE 482		2156
PAGE 484		2268 2292
PAGE 488	PROCESS ASSESSMENT	2475
PAGE 490		2589 2627
PAGE 491		2681
PAGE 492		2734
PAGE 499	PROCESS SHOOT.OUT	3130 3154
PAGE 500		3187 3206
PAGE 501		3228 3256
PAGE 502		3309
PAGE 506	PROCESS CAS.MISSION	3491
PAGE 514	PROCESS HELICOPTER.FIRE	3943

```
PAGE 515                                    3983
PAGE 516                                    4008 4023
PAGE 517                                    4087 4102
PAGE 518                                    4146 4161
PAGE 533  ROUTINE UNIT.INPUT                4687 4708 4714
PAGE 534                                    4751 4755 4756
PAGE 535                                    4851 4852
PAGE 536                                    4857 4869 4876 4897 4900 4903 4903 4903 4903 4904 4906 4907 4908 4909 4910
PAGE 537                                    4912 4912 4912 4913 4916 4925 4926
PAGE 539  ROUTINE READ.ORDERS               4986 4986 5007 5015 5020 5029
PAGE 540                                    5045 5088
PAGE 542  ROUTINE ORD.DEF                   5123 5125 5126
PAGE 543  ROUTINE ORD.ATK                   5147
PAGE 544  ROUTINE ORD.REINF                 5166 5167 5168 5169
PAGE 545  ROUTINE ORD.MOVDIS                5190
PAGE 546  ROUTINE ORD.MOVCOR                5219 5220 5221 5223 5225
PAGE 547  ROUTINE P.E.M.INPUT               5278 5292
PAGE 551  ROUTINE BTRY.INPUT                5469 5480
PAGE 553  ROUTINE FBN.FD.INPUT              5588 5588
PAGE 565  ROUTINE SENSOR.INPUT              6035 6035 6064
PAGE 567                                    6147 6147 6154 6157 6160
PAGE 573  ROUTINE FARRP.INPUT               6413 6414 6421 6421 6427
PAGE 578  ROUTINE MINE.INPUT                6601 6637
PAGE 582  ROUTINE TACAIR.INPUT              6756 6757 6764
PAGE 585                                    6925
PAGE 592  ROUTINE AMMO.RPT                  7141 7143 7175
PAGE 593                                    7178 7180 7192 7194 7196 7209 7211 7213
PAGE 595  ROUTINE ANALYSIS.OUTPUT           7254 7256 7257 7259 7281 7283 7286 7286 7286
PAGE 596                                    7314 7315
PAGE 598                                    7424
PAGE 606  ROUTINE KV.PRINT                  7735
PAGE 607                                    7788
PAGE 609  ROUTINE KV.SCOREBOARD             7859 7871 7872 7873 7874 7875 7876 7890 7892 7895 7899
PAGE 610                                    7904 7912
PAGE 611  ROUTINE OUTPUT.ATTRITION          7930 7933 7938 7939 7941 7948 7955
PAGE 612                                    7990
PAGE 613  ROUTINE FOR POSITION.OUT          7997 7999 8002 8004 8004 8005 8005 8006 8009 8011 8012 8016
PAGE 618  ROUTINE SNAP.R                    8256
PAGE 619                                    8263 8281 8314
PAGE 622  ROUTINE TACAIR.DATA.REPORT        8443
PAGE 624  ROUTINE OUTPUT.EXPENDITURES       8494 8529
PAGE 625                                    8591
PAGE 629  FUNCTION BTRY.AVAILABLE           8648
PAGE 632  FUNCTION EST.RANGE                8744 8749 8752 8753 8756 8757
PAGE 635  FUNCTION FEBA.BAND                8835
PAGE 636  FUNCTION HE.WLA                   8876
PAGE 638  FUNCTION ICM.WLA                  8999
PAGE 647  ROUTINE OPEN.INPUT.OUTPUT.FILES   9182 9184 9185 9187
PAGE 648  ROUTINE PERFORM.INSTRUMENTATION   9201
PAGE 653  PROCESS AIRBORNE.RADAR            9272 9276
PAGE 656  ROUTINE AR.DETECTION              9437
PAGE 658  PROCESS PHOTO.IR.FLIGHT           9481 9491 9492
PAGE 659                                    9584
PAGE 660  FUNCTION STAY.TIME                9592 9596 9599 9602 9606
PAGE 676  ''PROGRAM  OLDER.VERSION           377
PAGE 677                                     419  423  428  429  442
```

```
    PAGE 678                                    508  508
    PAGE 697                                   1591 1592 1602
    PAGE 712                                   2477
UNITS.
    PAGE 130  ROUTINE CHECK.DEAD               6067 6077 6078 6079 6081 6082 6085 6091 6092 6094 6097 6097 6099 6102 6103
                                               6104 6106 6110 6117 6122
    PAGE 131                                   6125 6129
    PAGE 182  ROUTINE FA.BN.ASGN               8297
    PAGE 506  PROCESS CAS.MISSION              3480
    PAGE 510  PROCESS HELICOPTER.FIRE          3664
    PAGE 546  ROUTINE ORD.MOVCOR               5212
UNIT.
    PAGE  71  ROUTINE ORIENTATION              3668 3670 3671 3672 3675 3676
    PAGE  77  ROUTINE ADJUST                   3910
    PAGE  83  ROUTINE CHANGE.LOC               4137
    PAGE  95  ROUTINE INIT.REINF               4642
    PAGE 115  ROUTINE PROX.CHECK               5512
    PAGE 117  ROUTINE PROX.POS                 5571
    PAGE 118  ROUTINE REIN.ARRIVE              5604 5606
    PAGE 142  ROUTINE DEAD.UNIT                6526
    PAGE 183  ROUTINE FA.BN.ASGN               8363
    PAGE 208  ROUTINE PGM.MSN.ASGN             9524
    PAGE 347  EVENT ACT.REINF                  5812 5820
    PAGE 376  EVENT HELO.ENGAGEMENT            7046
    PAGE 408  EVENT UPDATE.LOC                 8414 8422
    PAGE 420  PROCESS AC.ATK.TGT               8934
    PAGE 422                                   9037
    PAGE 443  PROCESS HC.ARRIVE.BATTLE          103
    PAGE 506  PROCESS CAS.MISSION              3492 3513
    PAGE 619  ROUTINE SNAP.R                   8265 8266 8267 8268 8269
    PAGE 629  FUNCTION BTRY.AVAILABLE          8650
    PAGE 634  FUNCTION FEBA.BAND               8791 8809
UNIT.A
    PAGE 342  ROUTINE RANGE.COMPUTE            5663 5668 5670 5671
UNIT.AREA
    PAGE 343  ROUTINE SEARCH.COVERAGE          5707 5718 5730
UNIT.B
    PAGE 342  ROUTINE RANGE.COMPUTE            5663 5668 5670 5671
UNIT.ENVIR
    PAGE 154  ROUTINE AO.DETECTION             7118
    PAGE 212  ROUTINE PIR.DETECTION            9725 9728
    PAGE 229  ROUTINE RPV.DETECTION             538  541
    PAGE 242  ROUTINE UNIT.ENVIR               1111
    PAGE 261  ROUTINE BTRY.EFFECTS             2040
    PAGE 276  ROUTINE AC.BOMB.EFFECTS          2822
UNIT.HQ
    PAGE 347  EVENT ACT.REINF                  5831 5833 5834
UNIT.INPUT
    PAGE 520  ROUTINE MAIN2                    4215 4217
    PAGE 533  ROUTINE UNIT.INPUT               4680
UNIT.LIST
    PAGE  66  ROUTINE FORM.TF.LIST             3440 3441
UNIT.LIST.
    PAGE  66  ROUTINE FORM.TF.LIST             3416 3438 3439 3443
UNIT.M.FACTOR
    PAGE 242  ROUTINE UNIT.ENVIR               1136 1155 1156 1157 1160
```

```
    PAGE 243                                          1215
    PAGE 244                                          1231 1242
UNIT.NO
    PAGE 539  ROUTINE READ.ORDERS                     5036 5037
    PAGE 540                                          5061 5084
UNIT.NOS
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         979
    PAGE  61  ROUTINE FEBA.INITIAL                    3216 3222 3226 3236
    PAGE  66  ROUTINE FORM.TF.LIST                    3426 3433 3433
    PAGE  88  ROUTINE END.MOVE                        4388
    PAGE 106  ROUTINE MIN.MOVE                        5078
    PAGE 107  ROUTINE NEW.SEGMENT                     5147 5148
    PAGE 115  ROUTINE PROX.CHECK                      5545
    PAGE 126  ROUTINE WHAT.NEXT                       5951 5969
    PAGE 130  ROUTINE CHECK.DEAD                      6106
    PAGE 142  ROUTINE DEAD.UNIT                       6544 6554 6554 6557
    PAGE 143                                          6586 6628 6628
    PAGE 146  ROUTINE INTER.BATTLE                    6772
    PAGE 154  ROUTINE AO.DETECTION                    7114
    PAGE 159  ROUTINE ATTRIT.SENSOR                   7334 7342
    PAGE 162                                          7501
    PAGE 239  ROUTINE TARGET.ANALYSIS                  996
    PAGE 240                                          1082
    PAGE 241                                          1090
    PAGE 242  ROUTINE UNIT.ENVIR                      1146
    PAGE 271  ROUTINE BTRY.EFFECTS                    2621
    PAGE 286  ROUTINE CHECK.CAS.CONSTRAINTS           3393
    PAGE 288                                          3466 3468
    PAGE 295  ROUTINE END.CAS.MISSION                 3795
    PAGE 303  ROUTINE HEL.RANGE.COMPUTE               4181
    PAGE 324  ROUTINE DECIDE                          5065
    PAGE 325  ROUTINE DEQ.FEBA.SET                    5086
    PAGE 331  ROUTINE ENQ.FEBA.SET                    5308
    PAGE 347  EVENT A  REINF                          5828 5835 5835
    PAGE 348                                          5866 5866 5883
    PAGE 356  EVENT BTL.ENDED                         6302 6303 6310
    PAGE 357                                          6311 6318 6319
    PAGE 369  EVENT ENGAGEMENT                        6690 6691
    PAGE 373  EVENT GET.NX.ORD                        6902
    PAGE 376  EVENT HELO.ENGAGEMENT                   7052 7053
    PAGE 382  EVENT OFF.LINE.ATTRITION                7289
    PAGE 392  EVENT SET.DEBUG                         7640
    PAGE 404  EVENT START.MOVE                        8270
    PAGE 408  EVENT UPDATE.LOC                        8453
    PAGE 441  PROCESS FORWARD.OBSERVER                  57
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION           820  821
    PAGE 457                                           894  896
    PAGE 459                                          1011 1013
    PAGE 478  PROCESS WITH.DRAW                       1934
    PAGE 481  PROCESS FIRE.MISSION                    2135
    PAGE 494  PROCESS SHOOT.OUT                       2817
    PAGE 533  ROUTINE UNIT.INPUT                      4692 4713 4721 4721 4722 4723
    PAGE 534                                          4743 4744 4750 4792 4793
    PAGE 535                                          4829 4830
    PAGE 536                                          4860 4864 4865
    PAGE 537                                          4912
```

```
    PAGE 539  ROUTINE READ.ORDERS                    5002
    PAGE 540                                         5087
    PAGE 551  ROUTINE BTRY.INPUT                     5446 5468 5479
    PAGE 565  ROUTINE SENSOR.INPUT                   6030
    PAGE 567                                         6142
    PAGE 573  ROUTINE FARRP.INPUT                    6421 6434
    PAGE 582  ROUTINE TACAIR.INPUT                   6757
    PAGE 585                                         6926
    PAGE 588  ROUTINE AC.MUNS.INPUT                  7048 7049
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                7286
    PAGE 609  ROUTINE KV.SCOREBOARD                  7895
    PAGE 613  ROUTINE FOR POSITION.OUT               8004 8009
    PAGE 623  ROUTINE TACAIR.DATA.REPORT             8450
    PAGE 632  FUNCTION EST.RANGE                     8756
    PAGE 677  ''PROGRAM  OLDER.VERSION                421
UNIT.NOS.
    PAGE 398  EVENT START.BATTLE                     7958
UNIT.NO.
    PAGE 539  ROUTINE READ.ORDERS                    4998 4999 5002 5006 5011 5012 5013 5018 5026 5033
    PAGE 540                                         5098
    PAGE 541                                         5102 5103
UNIT.NUMBER
    PAGE 609  ROUTINE KV.SCOREBOARD                  7894 7895
UNIT.PRIORITY
    PAGE 305  ROUTINE UNIT.PRIORITY                  4234 4260 4291
    PAGE 396  EVENT START.BATTLE                     7843 7850
UNIT.REIN
    PAGE 118  ROUTINE REIN.ARRIVE                    5621
UNIT.REIN.
    PAGE 118  ROUTINE REIN.ARRIVE                    5614 5616 5624 5625 5633
UNIT.SEC.
    PAGE 347  EVENT ACT.REINF                        5844 5848
UNIT.SEQ.NO
    PAGE 533  ROUTINE UNIT.INPUT                     4691 4712
UNIT.SET
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES        772
    PAGE  18                                         1015
    PAGE  49  ''SECTION FOR DEFINITIONS              2830
    PAGE  61  ROUTINE FEBA.INITIAL                   3209
    PAGE 119  ROUTINE RESET.FEBA.SECTOR              5675
    PAGE 430  PROCESS AIR.OBSERVER                   9479
    PAGE 439  PROCESS FORWARD.OBSERVER               9915
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE           1409
    PAGE 534  ROUTINE UNIT.INPUT                     4756
    PAGE 653  PROCESS AIRBORNE.RADAR                 9316
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                9517
    PAGE 673  ''PROGRAM  OLDER.VERSION                214
    PAGE 677                                          457
    PAGE 708                                         2251
UNKNOWN.
    PAGE 316  ROUTINE FLIGHT.PATH                    4779
UNSUC.REINF.OP
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES       1709
    PAGE 348  EVENT ACT.REINF                        5881
    PAGE 544  ROUTINE ORD.REINF                      5175
    PAGE 689  ''PROGRAM  OLDER.VERSION               1148
```

```
UNT.
     PAGE  66  ROUTINE FORM.TF.LIST                3416 3423 3433
     PAGE  87  ROUTINE END.MOVE                    4309 4321 4328 4330 4344 4347 4350 4357 4359
     PAGE  88                                      4372 4383 4388
     PAGE 112  ROUTINE PREPARE.LIST                5371 5376 5380 5386 5393 5401 5403
     PAGE 113                                      5404 5405 5408 5409 5414 5419 5423 5425 5428
     PAGE 117  ROUTINE PROX.POS                    5565 5579 5590
     PAGE 118  ROUTINE REIN.ARRIVE                 5599 5620
     PAGE 137  ROUTINE CHECK.LIST                  6332 6345 6352 6353 6354 6357 6358 6360
     PAGE 138  ROUTINE CHECK.PROX                  6373 6392 6424 6426
     PAGE 139                                      6440
     PAGE 141  ROUTINE CHECK.STREN                 6503 6512 6513 6514
     PAGE 142  ROUTINE DEAD.UNIT                   6523 6537 6544 6547 6554 6562 6573 6574
     PAGE 143                                      6609 6617 6621 6622 6625 6628 6632 6633
     PAGE 144                                      6637 6639 6655 6656 6656 6657 6670 6671 6688
     PAGE 326  ROUTINE DESTROY.ORD                 5118 5122 5124
     PAGE 373  EVENT GET.NX.ORD                    6877 6890 6902 6914 6928
     PAGE 374                                      6934 6943 6950 6955 6966
     PAGE 403  EVENT START.MOVE                    8179 8209 8210 8216 8219 8220 8221 8223 8226 8227 8229 8233
     PAGE 404                                      8250 8255 8262 8270 8270 8271 8272 8280 8282 8283 8287 8288 8290
     PAGE 405                                      8293 8310 8312 8318 8322 8324 8329 8340 8347
     PAGE 406                                      8358
     PAGE 408  EVENT UPDATE.LOC                    8406 8435 8436 8440 8441 8448 8449 8453 8459
     PAGE 409                                      8478 8483 8491 8500 8502 8508 8515
     PAGE 410                                      8525 8528 8542 8543 8545 8548 8565 8567 8573
     PAGE 411                                      8582 8587
     PAGE 412  EVENT ACT.ATK                       8601 8629 8630 8630 8633 8634 8636
     PAGE 413                                      8659 8661
     PAGE 414  EVENT ACT.DEF                       8716 8726 8727 8728 8730
     PAGE 416  EVENT ACT.MOVDIS                    8768 8778 8791 8792 8795
     PAGE 546  ROUTINE ORD.MOVCOR                  5206 5248
UNT.CALLING
     PAGE  95  ROUTINE INIT.REINF                  4659
UNT.CALLING.
     PAGE  95  ROUTINE INIT.REINF                  4637 4652 4660 4661
UNT.RESPONDING
     PAGE  95  ROUTINE INIT.REINF                  4655
UNT.RESPONDING.
     PAGE  95  ROUTINE INIT.REINF                  4637 4656 4658 4659 4660
UNT.SECTOR
     PAGE 138  ROUTINE CHECK.PROX                  6408
     PAGE 409  EVENT UPDATE.LOC                    8473
UNT.SECTOR.
     PAGE 138  ROUTINE CHECK.PROX                  6373 6398 6400 6402 6403 6405 6407 6410
     PAGE 409  EVENT UPDATE.LOC                    8486 8492
UN.AD.AVAIL
     PAGE  18  ''SECTION FOR PERMANENT_ENTITIES     978
     PAGE 351  EVENT AD.ENGAGEMENT                 6045
     PAGE 354                                      6218
     PAGE 677  ''PROGRAM  OLDER.VERSION             420
UN.BATTLE
     PAGE 440  PROCESS FORWARD.OBSERVER            9976
UN.BATTLE.INDEX
     PAGE  18  ''SECTION FOR PERMANENT_ENTITIES     992
     PAGE  80  ROUTINE BLOCK.LOS                   4023
     PAGE  83  ROUTINE CHANGE.LOC                  4149
```

PAGE 101	ROUTINE LOS.CHECK	4877 4894 4896 4901 4903
PAGE 104	ROUTINE MINE.DELAY	5021
PAGE 106	ROUTINE MIN.MOVE	5083
PAGE 107	ROUTINE NEW.SEGMENT	5124 5127 5132 5135 5146
PAGE 108		5181
PAGE 109		5241 5242 5245 5246
PAGE 115	ROUTINE PROX.CHECK	5526 5544
PAGE 118	ROUTINE REIN.ARRIVE	5616 5621
PAGE 133	ROUTINE CHECK.FORCE	6159 6181 6200
PAGE 139	ROUTINE CHECK.PROX	6469 6470 6479
PAGE 149	ROUTINE PK.COMPUTE	6911
PAGE 173	ROUTINE DUST.EFFECTS	7902 7923 7929
PAGE 200	ROUTINE ILLUM.EFFECTS	9223 9230
PAGE 216	ROUTINE REQUEST.FASCAM	9932
PAGE 218	ROUTINE REQUEST.ILLUM	9999 6
PAGE 219		62 95 96
PAGE 220		115 116
PAGE 222	ROUTINE REQUEST.SMOKE	188 194
PAGE 223		235
PAGE 224		293 294 323 324
PAGE 226	ROUTINE REQUEST.WD.FASCAM	406 412
PAGE 234	ROUTINE SMOKE.EFFECTS	789
PAGE 235		813 820
PAGE 236		889
PAGE 260	ROUTINE BTRY.EFFECTS	2010
PAGE 261		2064
PAGE 262		2087
PAGE 287	ROUTINE CHECK.CAS.CONSTRAINTS	3408 3409
PAGE 293	ROUTINE END.CAS.MISSION	3728
PAGE 324	ROUTINE DECIDE	5064
PAGE 328	ROUTINE EMPTY	5184 5191 5198 5209 5218
PAGE 329		5226 5238
PAGE 338	ROUTINE HC.EMPTY	5567
PAGE 345	ROUTINE TERM.CHECK	5772 5792
PAGE 347	EVENT ACT.REINF	5838
PAGE 356	EVENT BTL.ENDED	6264 6267 6270 6296
PAGE 369	EVENT ENGAGEMENT	6689
PAGE 370		6745
PAGE 376	EVENT HELO.ENGAGEMENT	7075 7075
PAGE 385	EVENT OFF.LINE.ATTRITION	7460
PAGE 398	EVENT START.BATTLE	7973 7974
PAGE 420	PROCESS AC.ATK.TGT	8962
PAGE 440	PROCESS FORWARD.OBSERVER	9972 9980
PAGE 443	PROCESS HC.ARRIVE.BATTLE	113
PAGE 453	PROCESS HC.RETURN.FARRP	656
PAGE 478	PROCESS WITH.DRAW	1936 1953 1961
PAGE 479		1992
PAGE 488	PROCESS ASSESSMENT	2485 2487
PAGE 489		2534
PAGE 492		2741 2744
PAGE 493	PROCESS SHOOT.OUT	2796
PAGE 496		2963
PAGE 499		3125
PAGE 500		3168
PAGE 502		3323
PAGE 505	PROCESS CAS.MISSION	3452

```
    PAGE 508                                          3606 3607
    PAGE 514  PROCESS HELICOPTER.FIRE                 3939
    PAGE 515                                          3969 3970 3994 3995
    PAGE 517                                          4073 4074
    PAGE 518                                          4132 4133
    PAGE 611  ROUTINE OUTPUT.ATTRITION                7939
    PAGE 677  ''PROGRAM  OLDER.VERSION                 434
UN.BTRY.INDEX
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         993
    PAGE 252  ROUTINE MINE.EFFECTS                    1609
    PAGE 270  ROUTINE BTRY.EFFECTS                    2585
    PAGE 272                                          2717
    PAGE 273                                          2718 2722 2724
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                 2954
    PAGE 282  ROUTINE AC.DF.EFFECTS                   3136
    PAGE 384  EVENT OFF.LINE.ATTRITION                7436 7437
    PAGE 551  ROUTINE BTRY.INPUT                      5456
    PAGE 677  ''PROGRAM  OLDER.VERSION                 435
UN.COLOR
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         990
    PAGE  61  ROUTINE FEBA.INITIAL                    3227
    PAGE  65  ROUTINE FILE.KAD.SENSOR                 3366
    PAGE  71  ROUTINE ORIENTATION                     3670
    PAGE  89  ROUTINE FA.BN.MOVEMENT                  4411
    PAGE  90                                          4505
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4785
    PAGE 101  ROUTINE LOS.CHECK                       4878
    PAGE 103  ROUTINE MINE.DELAY                      4951 4972
    PAGE 115  ROUTINE PROX.CHECK                      5541
    PAGE 119  ROUTINE RESET.FEBA.SECTOR               5686
    PAGE 128  ROUTINE BTL.CHECK                       5996
    PAGE 130  ROUTINE CHECK.DEAD                      6081
    PAGE 133  ROUTINE CHECK.FORCE                     6160 6201
    PAGE 134  ROUTINE CHECK.FOR.MINES                 6245
    PAGE 138  ROUTINE CHECK.PROX                      6392
    PAGE 145  ROUTINE FIN.BATTLE                      6716
    PAGE 147  ROUTINE INTER.BATTLE                    6797 6817
    PAGE 149  ROUTINE PK.COMPUTE                      6913 6914
    PAGE 159  ROUTINE ATTRIT.SENSOR                   7323
    PAGE 165  ROUTINE CFR.DEGRADE                     7585
    PAGE 173  ROUTINE DUST.EFFECTS                    7924
    PAGE 183  ROUTINE FA.BN.ASGN                      8365
    PAGE 200  ROUTINE ILLUM.EFFECTS                   9209 9218 9224
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ            9325
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9533
    PAGE 212  ROUTINE PIR.DETECTION                   9721
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM              9859
    PAGE 216  ROUTINE REQUEST.FASCAM                  9933
    PAGE 217                                          9946 9946
    PAGE 218  ROUTINE REQUEST.ILLUM                      0
    PAGE 219                                            63   76   76
    PAGE 220                                            97   97  117  118
    PAGE 222  ROUTINE REQUEST.SMOKE                    189
    PAGE 223                                           236  249  249
    PAGE 224                                           295  295  325  326
    PAGE 226  ROUTINE REQUEST.WD.FASCAM                407  419
```

PAGE 227		473
PAGE 229	ROUTINE RPV.DETECTION	534
PAGE 234	ROUTINE SMOKE.EFFECTS	773 782
PAGE 235		814
PAGE 239	ROUTINE TARGET.ANALYSIS	1005
PAGE 246	ROUTINE VOLLEY	1322
PAGE 248	ROUTINE WEIGHTED.VOLLEYS	1423
PAGE 250	ROUTINE MINE.EFFECTS	1482
PAGE 254	ROUTINE FO.DETECTION	1700
PAGE 262	ROUTINE BTRY.EFFECTS	2099
PAGE 271		2646
PAGE 272		2671
PAGE 287	ROUTINE CHECK.CAS.CONSTRAINTS	3411
PAGE 298	ROUTINE HC.COMPUTE.TIMES	3932
PAGE 299		3973
PAGE 302	ROUTINE HEL.RANGE.COMPUTE	4171
PAGE 308	ROUTINE AD.SHOOT	4399
PAGE 309		4429 4445 4452
PAGE 310		4472 4473 4474
PAGE 314	ROUTINE FLIGHT.PATH	4649
PAGE 324	ROUTINE DECIDE	5056
PAGE 325	ROUTINE DEQ.FEBA.SET	5090
PAGE 328	ROUTINE EMPTY	5200
PAGE 331	ROUTINE ENQ.FEBA.SET	5311
PAGE 345	ROUTINE TERM.CHECK	5773
PAGE 350	EVENT AD.ENGAGEMENT	5961 6002
PAGE 358	EVENT CFR.ACTIVATION	6356
PAGE 360	EVENT CFR.ON	6428
PAGE 370	EVENT ENGAGEMENT	6763 6769
PAGE 375	EVENT HC.DEPART.BATTLE	7006
PAGE 380	EVENT INIT.PREPLAN.CAS	7239
PAGE 382	EVENT OFF.LINE.ATTRITION	7296
PAGE 386	EVENT PDB.ACTIVATION	7483
PAGE 395	EVENT START.BATTLE	7770 7770
PAGE 398		7949 7959
PAGE 403	EVENT START.MOVE	8219
PAGE 410	EVENT UPDATE.LOC	8528
PAGE 412	EVENT ACT.ATK	8633
PAGE 416	EVENT ACT.MOVDIS	8778
PAGE 428	PROCESS AIR.OBSERVER	9361
PAGE 435	PROCESS ARTY.ASSESS	9699
PAGE 438	PROCESS FORWARD.OBSERVER	9881
PAGE 443	PROCESS HC.ARRIVE.BATTLE	115
PAGE 450	PROCESS HC.RETURN.FARRP	475
PAGE 454	PROCESS HEL.TARGET.ACQUISITION	727
PAGE 461		1135
PAGE 464	PROCESS MINE.ASSESS	1230
PAGE 467	PROCESS REMOTE.PILOT.VEHICLE	1377
PAGE 474	PROCESS TARGET.REPORT	1763
PAGE 476		1875
PAGE 478	PROCESS WITH.DRAW	1947 1954
PAGE 480	PROCESS FIRE.MISSION	2057
PAGE 487	PROCESS ASSESSMENT	2432
PAGE 488		2471 2484
PAGE 489		2535 2553
PAGE 490		2585 2604 2612 2623

```
    PAGE 491                                          2677
    PAGE 492                                          2708 2730
    PAGE 494  PROCESS SHOOT.OUT                       2838
    PAGE 497                                          3014
    PAGE 499                                          3104 3113 3116 3150
    PAGE 500                                          3183 3202
    PAGE 501                                          3224 3252
    PAGE 502                                          3305
    PAGE 506  PROCESS CAS.MISSION                     3494
    PAGE 508                                          3608
    PAGE 510  PROCESS HELICOPTER.FIRE                 3695
    PAGE 514                                          3917 3919 3928
    PAGE 515                                          3968 3993
    PAGE 516                                          4018
    PAGE 517                                          4072 4097
    PAGE 518                                          4131 4156
    PAGE 533  ROUTINE UNIT.INPUT                      4697 4730 4734
    PAGE 534                                          4776
    PAGE 535                                          4803
    PAGE 537                                          4912
    PAGE 565  ROUTINE SENSOR.INPUT                    6041
    PAGE 566                                          6090
    PAGE 573  ROUTINE FARRP.INPUT                     6430 6435
    PAGE 585  ROUTINE TACAIR.INPUT                    6956
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                 7259
    PAGE 613  ROUTINE FOR POSITION.OUT                8011
    PAGE 634  FUNCTION FEBA.BAND                      8786
    PAGE 653  PROCESS AIRBORNE.RADAR                  9276
    PAGE 656  ROUTINE AR.DETECTION                    9400
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                 9492
    PAGE 660  FUNCTION STAY.TIME                      9599
    PAGE 677  ''PROGRAM  OLDER.VERSION                 432
UN.DELAY
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         996
    PAGE  44  ''SECTION FOR DEFINITIONS               2510
    PAGE  83  ROUTINE CHANGE.LOC                      4136 4138 4140 4141 4144 4157
    PAGE 104  ROUTINE MINE.DELAY                      5023 5023
    PAGE 106  ROUTINE MIN.MOVE                        5094
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM              9862
    PAGE 226  ROUTINE REQUEST.WD.FASCAM                429
    PAGE 413  EVENT ACT.ATK                           8688 8694
    PAGE 677  ''PROGRAM  OLDER.VERSION                 438
    PAGE 703                                          1931
UN.DISTANCE
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                 205  207  208
UN.ENGAGED.INDEX
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         980
    PAGE 263  ROUTINE BTRY.EFFECTS                    2142
    PAGE 398  EVENT START.BATTLE                      7948
    PAGE 609  ROUTINE KV.SCOREBOARD                   7899
    PAGE 677  ''PROGRAM  OLDER.VERSION                 422
UN.ENVIR.FRACT
    PAGE 212  ROUTINE PIR.DETECTION                   9710 9711 9732 9734 9736 9749 9751
    PAGE 213                                          9789
    PAGE 229  ROUTINE RPV.DETECTION                    522  523  545  547  549  562  564
    PAGE 230                                           621
```

```
    PAGE 260  ROUTINE BTRY.EFFECTS                    1996 1998
    PAGE 261                                          2050 2053 2056
    PAGE 265                                          2258 2264 2272 2281 2308
    PAGE 266                                          2350 2351 2358 2359 2366 2367
    PAGE 273                                          2729
UN.EQUIP.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         998
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES        1813
    PAGE  49  ''SECTION FOR DEFINITIONS               2829
    PAGE  81  ROUTINE BLOCK.LOS                       4093 4107
    PAGE 108  ROUTINE NEW.SEGMENT                     5208
    PAGE 109                                          5225
    PAGE 123  ROUTINE TIME.TO.DETECT                  5816 5819
    PAGE 154  ROUTINE AO.DETECTION                    7123
    PAGE 183  ROUTINE FA.BN.ASGN                      8375
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9543
    PAGE 212  ROUTINE PIR.DETECTION                   9741
    PAGE 227  ROUTINE REQUEST.WD.FASCAM                456
    PAGE 229  ROUTINE RPV.DETECTION                    554
    PAGE 251  ROUTINE MINE.EFFECTS                    1556
    PAGE 255  ROUTINE FO.DETECTION                    1764
    PAGE 264  ROUTINE BTRY.EFFECTS                    2208 2223
    PAGE 268                                          2456
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                 2884
    PAGE 282  ROUTINE AC.DF.EFFECTS                   3157
    PAGE 283  ROUTINE CAS.EVAL                        3192
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS           3297 3327
    PAGE 286                                          3347
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS              3495
    PAGE 293  ROUTINE END.CAS.MISSION                 3717
    PAGE 295                                          3822 3827
    PAGE 297  ROUTINE FARRP.CHECK                     3873
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                3959
    PAGE 300  ROUTINE HC.DISENGAGE                    4052
    PAGE 301                                          4086
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE               4160
    PAGE 304  ROUTINE REPLACE.HC                      4200
    PAGE 307  ROUTINE AD.SHOOT                        4330
    PAGE 312  ROUTINE INTER.HELO                      4592
    PAGE 329  ROUTINE EMPTY                           5247
    PAGE 352  EVENT AD.ENGAGEMENT                     6072
    PAGE 369  EVENT ENGAGEMENT                        6714 6718
    PAGE 376  EVENT HELO.ENGAGEMENT                   7076
    PAGE 377                                          7089 7114
    PAGE 378                                          7154 7176 7191
    PAGE 382  EVENT OFF.LINE.ATTRITION                7319
    PAGE 392  EVENT SET.DEBUG                         7643
    PAGE 398  EVENT START.BATTLE                      7966
    PAGE 419  PROCESS AC.ATK.TGT                      8887
    PAGE 420                                          8925 8937 8949
    PAGE 423                                          9086 9102 9118
    PAGE 424                                          9148 9194
    PAGE 425                                          9209 9225 9241
    PAGE 426                                          9269
    PAGE 445  PROCESS HC.ARRIVE.BATTLE                 265
    PAGE 452  PROCESS HC.RETURN.FARRP                  592
```

```
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION          899  918
    PAGE 459                                         1017
    PAGE 461                                         1100 1114
    PAGE 463  PROCESS HOW.REPAIR                     1196
    PAGE 480  PROCESS FIRE.MISSION                   2068
    PAGE 489  PROCESS ASSESSMENT                     2565
    PAGE 491                                         2645
    PAGE 496  PROCESS SHOOT.OUT                      2924
    PAGE 504  PROCESS CAS.MISSION                    3374
    PAGE 507                                         3565
    PAGE 510  PROCESS HELICOPTER.FIRE                3679
    PAGE 535  ROUTINE UNIT.INPUT                     4809
    PAGE 537                                         4916
    PAGE 551  ROUTINE BTRY.INPUT                     5462
    PAGE 588  ROUTINE AC.MUNS.INPUT                  7052 7062
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                7315
    PAGE 610  ROUTINE KV.SCOREBOARD                  7904
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7941 7956
    PAGE 619  ROUTINE SNAP.R                         8265 8269
    PAGE 656  ROUTINE AR.DETECTION                   9401
    PAGE 677  ''PROGRAM  OLDER.VERSION                440
    PAGE 691                                         1252
    PAGE 708                                         2250
UN.FASCAM.RECVD
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES        995
    PAGE  44  ''SECTION FOR DEFINITIONS              2509
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM             9861
    PAGE 217  ROUTINE REQUEST.FASCAM                 9971
    PAGE 226  ROUTINE REQUEST.WD.FASCAM               430
    PAGE 227                                          497
    PAGE 413  EVENT ACT.ATK                          8689 8695
    PAGE 677  ''PROGRAM  OLDER.VERSION                437
    PAGE 703                                         1930
UN.FWD.ADV
    PAGE 119  ROUTINE RESET.FEBA.SECTOR              5687 5689 5692 5696
UN.GROUP
    PAGE 270  ROUTINE BTRY.EFFECTS                   2582 2583
UN.HC.LOS.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES       1003
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES       1862
    PAGE 300  ROUTINE HC.DISENGAGE                   4035 4038
    PAGE 312  ROUTINE INTER.HELO                     4575 4579
    PAGE 377  EVENT HELO.ENGAGEMENT                  7127
    PAGE 378                                         7170
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION          842  852
    PAGE 459                                          991
    PAGE 460                                         1077
    PAGE 461                                         1127 1130
    PAGE 489  PROCESS ASSESSMENT                     2542 2546
    PAGE 498  PROCESS SHOOT.OUT                      3042
    PAGE 677  ''PROGRAM  OLDER.VERSION                445
    PAGE 692                                         1301
UN.LAST.ARTY.ENG
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES        985
    PAGE 182  ROUTINE FA.BN.ASGN                     8341
    PAGE 183                                         8345
```

```
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9518
    PAGE 266  ROUTINE BTRY.EFFECTS                    2336
    PAGE 270                                          2587
    PAGE 272                                          2706
    PAGE 481  PROCESS FIRE.MISSION                    2127
    PAGE 482                                          2143
    PAGE 677  ''PROGRAM  OLDER.VERSION                 427
UN.LOS.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES        1002
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES        1860
    PAGE  80  ROUTINE BLOCK.LOS                       4056 4060 4062
    PAGE  81                                          4066
    PAGE  93  ROUTINE INITIAL.DETECT                  4603
    PAGE  96  ROUTINE LINE.OF.SIGHT                   4718
    PAGE  97                                          4722
    PAGE 101  ROUTINE LOS.CHECK                       4886 4916 4920
    PAGE 108  ROUTINE NEW.SEGMENT                     5193 5197 5200 5204
    PAGE 109                                          5250 5254
    PAGE 132  ROUTINE CHECK.ENGAGEMENT                6137 6144
    PAGE 201  ROUTINE ILLUM.EFFECTS                   9277
    PAGE 255  ROUTINE FO.DETECTION                    1729
    PAGE 329  ROUTINE EMPTY                           5241 5243
    PAGE 369  EVENT ENGAGEMENT                        6706
    PAGE 478  PROCESS WITH.DRAW                       1968
    PAGE 487  PROCESS ASSESSMENT                      2433
    PAGE 498  PROCESS SHOOT.OUT                       3057
    PAGE 619  ROUTINE SNAP.R                          8267
    PAGE 677  ''PROGRAM  OLDER.VERSION                 444
    PAGE 692                                          1299
UN.MIL.WORTH
    PAGE 180  ROUTINE EST.MIL.WORTH                   8227 8236
UN.MISSION
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         988
    PAGE  52  ''SECTION FOR SUBSTITUTIONS             2952
    PAGE  61  ROUTINE FEBA.INITIAL                    3210
    PAGE  95  ROUTINE INIT.REINF                      4659 4659
    PAGE 113  ROUTINE PREPARE.LIST                    5425 5430
    PAGE 115  ROUTINE PROX.CHECK                      5529 5534 5541
    PAGE 119  ROUTINE RESET.FEBA.SECTOR               5676
    PAGE 144  ROUTINE DEAD.UNIT                       6670
    PAGE 147  ROUTINE INTER.BATTLE                    6851 6852
    PAGE 148                                          6888 6889
    PAGE 200  ROUTINE ILLUM.EFFECTS                   9210 9219 9237
    PAGE 201                                          9244 9251
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM              9843 9849 9859
    PAGE 226  ROUTINE REQUEST.WD.FASCAM                419  420
    PAGE 234  ROUTINE SMOKE.EFFECTS                    774  783
    PAGE 235                                           828  830  831
    PAGE 262  ROUTINE BTRY.EFFECTS                    2094
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                 2906
    PAGE 331  ROUTINE ENQ.FEBA.SET                    5313
    PAGE 370  EVENT ENGAGEMENT                        6754 6764 6770
    PAGE 398  EVENT START BATTLE                      7950 7952 7959
    PAGE 399                                          7978 7979 7980 7986 7987 7988
    PAGE 413  EVENT ACT.ATK                           8706 8707
    PAGE 414  EVENT ACT.DEF                           8726 8730 8730
```

```
    PAGE 415  EVENT ACT.MOVCOR                          8750 8753 8753
    PAGE 478  PROCESS WITH.DRAW                         1948
    PAGE 636  FUNCTION HE.WLA                           8875
    PAGE 638  FUNCTION ICM.WLA                          8998
    PAGE 677  ''PROGRAM  OLDER.VERSION                   430
    PAGE 710                                            2359
UN.PARENT
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES           987
    PAGE 125  ROUTINE WHAT.NEXT                         5893 5908
    PAGE 143  ROUTINE DEAD.UNIT                         6622 6632 6632 6634 6635
    PAGE 144                                            6640 6655 6656 6657
    PAGE 147  ROUTINE INTER.BATTLE                      6823
    PAGE 148                                            6860
    PAGE 347  EVENT ACT.REINF                           5831
    PAGE 533  ROUTINE UNIT.INPUT                        4696
    PAGE 536                                            4858 4860 4864 4868 4869
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                   7286
    PAGE 613  ROUTINE FOR POSITION.OUT                  8006 8009
    PAGE 677  ''PROGRAM  OLDER.VERSION                   429
UN.PATH
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          1000
    PAGE  30  ''SECTION FOR TEMPORARY_ENTITIES          1701
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                   3867
    PAGE  77  ROUTINE ADJUST                            3914
    PAGE  85  ROUTINE CHANGE.LOC                        4280
    PAGE 111  ROUTINE PRED.POS                          5324
    PAGE 321  ROUTINE COMPUTE.D                         4953
    PAGE 322  ROUTINE COMPUTE.WD                        4974 4983
    PAGE 329  ROUTINE EMPTY                             5269 5271
    PAGE 399  EVENT START.BATTLE                        7989
    PAGE 619  ROUTINE SNAP.R                            8266
    PAGE 677  ''PROGRAM  OLDER.VERSION                   442
    PAGE 689                                            1140
UN.PCT.OPEN
    PAGE 154  ROUTINE AO.DETECTION                      7096 7118 7120
    PAGE 212  ROUTINE PIR.DETECTION                     9709 9728 9732
    PAGE 229  ROUTINE RPV.DETECTION                      521  541  545
    PAGE 242  ROUTINE UNIT.ENVIR                        1115 1138
    PAGE 243                                            1214 1216 1217 1218 1220 1221 1223
    PAGE 244                                            1241 1243 1244 1246 1247 1250 1251 1253 1257 1283
    PAGE 245                                            1285 1291
    PAGE 260  ROUTINE BTRY.EFFECTS                      1995
    PAGE 261                                            2044 2050
    PAGE 271                                            2651
    PAGE 272                                            2662 2693
UN.PCT.TOWN
    PAGE 154  ROUTINE AO.DETECTION                      7096 7118
    PAGE 212  ROUTINE PIR.DETECTION                     9709 9728 9736
    PAGE 229  ROUTINE RPV.DETECTION                      521  541  549
    PAGE 242  ROUTINE UNIT.ENVIR                        1117 1138
    PAGE 243                                            1221 1225
    PAGE 244                                            1251 1254 1258 1264 1265 1266 1268 1269 1281 1282
    PAGE 245                                            1285 1293
    PAGE 260  ROUTINE BTRY.EFFECTS                      1995
    PAGE 261                                            2046 2056
    PAGE 271                                            2651
```

```
    PAGE 272                                          2662 2693
UN.PCT.WOOD
    PAGE 154  ROUTINE AO.DETECTION                    7096 7118
    PAGE 212  ROUTINE PIR.DETECTION                   9709 9728 9734
    PAGE 229  ROUTINE RPV.DETECTION                    521  541  547
    PAGE 242  ROUTINE UNIT.ENVIR                      1116 1138
    PAGE 243                                          1220 1221 1224
    PAGE 244                                          1230 1232 1233 1235 1236 1239 1250 1251 1253 1257 1273 1274 1275 1277 1278
                                                      1281 1282
    PAGE 245                                          1285 1292
    PAGE 260  ROUTINE BTRY.EFFECTS                    1995
    PAGE 261                                          2045 2053
    PAGE 271                                          2651
    PAGE 272                                          2662 2693
UN.POSITION.INDEX
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         991
    PAGE  78  ROUTINE ADJUST                          3939 3971 3987 3992
    PAGE  79                                          3999
    PAGE  83  ROUTINE CHANGE.LOC                      4167 4169 4177
    PAGE  84                                          4198 4199 4231 4236 4237
    PAGE  85                                          4262 4263 4285
    PAGE 111  ROUTINE PRED.POS                        5326
    PAGE 115  ROUTINE PROX.CHECK                      5523
    PAGE 133  ROUTINE CHECK.FORCE                     6190
    PAGE 321  ROUTINE COMPUTE.D                       4954
    PAGE 322  ROUTINE COMPUTE.WD                      4984
    PAGE 324  ROUTINE DECIDE                          5049
    PAGE 329  ROUTINE EMPTY                           5237
    PAGE 345  ROUTINE TERM.CHECK                      5783
    PAGE 399  EVENT START.BATTLE                      7997 8002
    PAGE 677  ''PROGRAM  OLDER.VERSION                 433
UN.PTR
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         994
    PAGE  61  ROUTINE FEBA.INITIAL                    3211 3212
    PAGE  66  ROUTINE FORM.TF.LIST                    3430 3431 3432 3438 3438 3439 3440 3441
    PAGE  87  ROUTINE END.MOVE                        4321
    PAGE  95  ROUTINE INIT.REINF                      4655 4656 4658
    PAGE 104  ROUTINE MINE.DELAY                      5041 5042
    PAGE 112  ROUTINE PREPARE.LIST                    5370 5375
    PAGE 113                                          5408 5409 5419 5423 5428
    PAGE 118  ROUTINE REIN.ARRIVE                     5624 5625
    PAGE 119  ROUTINE RESET.FEBA.SECTOR               5677 5678
    PAGE 125  ROUTINE WHAT.NEXT                       5892 5897 5900 5903 5904 5905 5909 5912 5916 5918 5930
    PAGE 126                                          5931 5933 5954 5956
    PAGE 130  ROUTINE CHECK.DEAD                      6085 6102
    PAGE 133  ROUTINE CHECK.FORCE                     6171 6174
    PAGE 137  ROUTINE CHECK.LIST                      6358
    PAGE 138  ROUTINE CHECK.PROX                      6424 6428
    PAGE 139                                          6440 6445
    PAGE 141  ROUTINE CHECK.STREN                     6512 6514
    PAGE 142  ROUTINE DEAD.UNIT                       6537 6552 6574
    PAGE 143                                          6581
    PAGE 144                                          6643
    PAGE 145  ROUTINE FIN.BATTLE                      6711
    PAGE 146  ROUTINE INTER.BATTLE                    6757
    PAGE 147                                          6806 6815 6819 6824 6826 6831 6832 6833 6835
```

```
    PAGE 148                                        6856 6861 6863 6868 6869 6870 6872
    PAGE 251  ROUTINE MINE.EFFECTS                  1574
    PAGE 252                                        1576
    PAGE 270  ROUTINE BTRY.EFFECTS                  2568 2569
    PAGE 278  ROUTINE AC.BOMB.EFFECTS               2948 2950
    PAGE 282  ROUTINE AC.DF.EFFECTS                 3131 3133
    PAGE 324  ROUTINE DECIDE                        5054
    PAGE 326  ROUTINE DESTROY.ORD                   5122 5124
    PAGE 347  EVENT ACT.REINF                       5836 5837
    PAGE 373  EVENT GET.NX.ORD                      6890
    PAGE 384  EVENT OFF.LINE.ATTRITION              7400 7401
    PAGE 398  EVENT START.BATTLE                    7954
    PAGE 403  EVENT START.MOVE                      8233
    PAGE 404                                        8287 8290
    PAGE 405                                        8293 8322
    PAGE 408  EVENT UPDATE.LOC                      8459
    PAGE 410                                        8542 8545 8548
    PAGE 412  EVENT ACT.ATK                         8629 8630 8636
    PAGE 413                                        8661
    PAGE 414  EVENT ACT.DEF                         8728
    PAGE 415  EVENT ACT.MOVCOR                      8751 8755
    PAGE 436  PROCESS ARTY.ASSESS                   9750 9753 9794 9797
    PAGE 437                                        9832 9834
    PAGE 465  PROCESS MINE.ASSESS                   1290 1292 1310 1312
    PAGE 478  PROCESS WITH.DRAW                     1930
    PAGE 492  PROCESS ASSESSMENT                    2736 2738
    PAGE 516  PROCESS HELICOPTER.FIRE               4059
    PAGE 534  ROUTINE UNIT.INPUT                    4739 4740 4741 4742 4742
    PAGE 535                                        4807
    PAGE 539  ROUTINE READ.ORDERS                   5018 5026
    PAGE 540                                        5098
    PAGE 541                                        5102
    PAGE 630  FUNCTION COLLISION                    8694 8695 8696
    PAGE 677  ''PROGRAM  OLDER.VERSION               436
UN.RADIUS
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES       984
    PAGE  44  ''SECTION FOR DEFINITIONS             2508
    PAGE  87  ROUTINE END.MOVE                      4332 4347
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA            4784
    PAGE 130  ROUTINE CHECK.DEAD                    6092 6103
    PAGE 188  ROUTINE FINAL.COVERAGE                8562
    PAGE 403  EVENT START.MOVE                      8209
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1427
    PAGE 533  ROUTINE UNIT.INPUT                    4698
    PAGE 537                                        4912
    PAGE 566  ROUTINE SENSOR.INPUT                  6099
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9537
    PAGE 677  ''PROGRAM  OLDER.VERSION               426
    PAGE 703                                        1929
UN.SEGMENT.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES      1001
    PAGE  31  ''SECTION FOR TEMPORARY_ENTITIES      1742
    PAGE  80  ROUTINE BLOCK.LOS                     4017 4041
    PAGE  84  ROUTINE CHANGE.LOC                    4208 4210
    PAGE  85                                        4272 4274
    PAGE  97  ROUTINE LINE.OF.SIGHT                 4744 4763
```

```
    PAGE 106  ROUTINE MIN.MOVE                          5072
    PAGE 122  ROUTINE SEGMENT.ADJUST                    5781
    PAGE 329  ROUTINE EMPTY                             5275 5277
    PAGE 479  PROCESS WITH.DRAW                         1986 1989
    PAGE 619  ROUTINE SNAP.R                            8268
    PAGE 677  ''PROGRAM  OLDER.VERSION                   443
    PAGE 690                                            1181
UN.SENSOR.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          1004
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1846
    PAGE 162  ROUTINE ATTRIT.SENSOR                     7498
    PAGE 216  ROUTINE REQUEST.FASCAM                    9925 9939
    PAGE 217                                            9947
    PAGE 219  ROUTINE REQUEST.ILLUM                       55   69   77
    PAGE 222  ROUTINE REQUEST.SMOKE                      228
    PAGE 223                                             242  250
    PAGE 238  ROUTINE SWITCH.FO                          929
    PAGE 253  ROUTINE MINE.EFFECTS                      1652
    PAGE 271  ROUTINE BTRY.EFFECTS                      2627
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                   2972
    PAGE 384  EVENT OFF.LINE.ATTRITION                  7424
    PAGE 419  PROCESS AC.ATK.TGT                        8903
    PAGE 566  ROUTINE SENSOR.INPUT                      6075
    PAGE 567                                            6165
    PAGE 619  ROUTINE SNAP.R                            8283
    PAGE 677  ''PROGRAM  OLDER.VERSION                   446
    PAGE 692                                            1285
UN.SENSOR.TYPE
    PAGE   3  PROGRAM REVISIONS                          124
UN.SEQ.NO
    PAGE 142  ROUTINE DEAD.UNIT                         6549
UN.STATUS
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES           989
    PAGE  51  ''SECTION FOR SUBSTITUTIONS               2930
    PAGE  78  ROUTINE ADJUST                            3940
    PAGE  79                                            3995
    PAGE  80  ROUTINE BLOCK.LOS                         4027
    PAGE  83  ROUTINE CHANGE.LOC                        4153 4158 4159
    PAGE  84                                            4200 4201 4203 4223
    PAGE  85                                            4264
    PAGE  94  ROUTINE INITIAL.MOVE                      4627
    PAGE  96  ROUTINE LINE.OF.SIGHT                     4691 4692 4698
    PAGE  97                                            4727 4746
    PAGE 101  ROUTINE LOS.CHECK                         4890 4891 4892 4893 4899 4900
    PAGE 103  ROUTINE MINE.DELAY                        4957 4957 4961
    PAGE 106  ROUTINE MIN.MOVE                          5089
    PAGE 107  ROUTINE NEW.SEGMENT                       5121 5122 5130
    PAGE 111  ROUTINE PRED.POS                          5311 5312 5318 5319
    PAGE 114  ROUTINE PREP.WITHDRAW                     5469 5470 5471 5475 5477 5479 5480
    PAGE 115  ROUTINE PROX.CHECK                        5517 5518 5519
    PAGE 133  ROUTINE CHECK.FORCE                       6188 6189
    PAGE 144  ROUTINE DEAD.UNIT                         6671
    PAGE 149  ROUTINE PK.COMPUTE                        6915 6916 6922 6926 6930 6931
    PAGE 173  ROUTINE DUST.EFFECTS                      7905 7906 7907
    PAGE 174                                            7969 7970 7971 7985 7986 7987
    PAGE 200  ROUTINE ILLUM.EFFECTS                     9202 9203
```

```
    PAGE 208  ROUTINE PGM.MSN.ASGN                  9504 9505
    PAGE 210                                        9628 9629
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM            9844 9845 9850 9851
    PAGE 226  ROUTINE REQUEST.WD.FASCAM              427  428
    PAGE 234  ROUTINE SMOKE.EFFECTS                  765  766  793  794  795
    PAGE 236                                         893  894  895
    PAGE 242  ROUTINE UNIT.ENVIR                    1151 1152 1153 1159
    PAGE 252  ROUTINE MINE.EFFECTS                  1606 1607
    PAGE 254  ROUTINE FO.DETECTION                  1703 1704 1705
    PAGE 256                                        1815 1816 1843 1844
    PAGE 257                                        1845 1896 1896
    PAGE 258                                        1937 1938
    PAGE 261  ROUTINE BTRY.EFFECTS                  2065 2066
    PAGE 267                                        2409 2410 2411
    PAGE 270                                        2598
    PAGE 272                                        2708
    PAGE 324  ROUTINE DECIDE                        5046 5047 5048 5050
    PAGE 345  ROUTINE TERM.CHECK                    5784
    PAGE 378  EVENT HELO.ENGAGEMENT                 7190
    PAGE 384  EVENT OFF.LINE.ATTRITION              7451
    PAGE 394  EVENT START.ARTY.MOVEMENT             7729
    PAGE 399  EVENT START.BATTLE                    7981 7983
    PAGE 403  EVENT START.MOVE                      8221 8223 8227 8229
    PAGE 405                                        8324 8324
    PAGE 407  EVENT STOP.ARTY.MOVEMENT              8394
    PAGE 414  EVENT ACT.DEF                         8727 8731
    PAGE 448  PROCESS HC.ARRIVE.BATTLE               413  414  415
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION         809
    PAGE 478  PROCESS WITH.DRAW                     1951
    PAGE 479                                        1988 2020
    PAGE 483  PROCESS FIRE.MISSION                  2226
    PAGE 533  ROUTINE UNIT.INPUT                    4733
    PAGE 611  ROUTINE OUTPUT.ATTRITION              7977 7977
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9520 9521
    PAGE 677  ''PROGRAM  OLDER.VERSION               431
    PAGE 710                                        2337
UN.SUB.LIST
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES       999 1008
    PAGE  49  ''SECTION FOR DEFINITIONS             2830
    PAGE  66  ROUTINE FORM.TF.LIST                  3423 3442
    PAGE  68  ROUTINE GENERAL.BATTLE                3534 3541
    PAGE  69                                        3566 3573 3599 3615
    PAGE 142  ROUTINE DEAD.UNIT                     6567 6573
    PAGE 143                                        6621 6633 6635
    PAGE 144                                        6637 6639 6641 6656
    PAGE 347  EVENT ACT.REINF                       5834
    PAGE 536  ROUTINE UNIT.INPUT                    4869
    PAGE 539  ROUTINE READ.ORDERS                   5033
    PAGE 677  ''PROGRAM  OLDER.VERSION               441  450
    PAGE 708                                        2251
UN.TIME.LAST.MOVE
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES       986
    PAGE  48  ''SECTION FOR DEFINITIONS             2758
    PAGE  83  ROUTINE CHANGE.LOC                    4135 4147
    PAGE  89  ROUTINE FA.BN.MOVEMENT                4432
    PAGE 114  ROUTINE PREP.WITHDRAW                 5476
```

```
    PAGE 252  ROUTINE MINE.EFFECTS                    1627
    PAGE 270  ROUTINE BTRY.EFFECTS                    2599
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                 2965
    PAGE 282  ROUTINE AC.DF.EFFECTS                   3145
    PAGE 329  ROUTINE EMPTY                           5239
    PAGE 355  EVENT ARTY.OCCUPATION                   6245
    PAGE 384  EVENT OFF.LINE.ATTRITION                7452
    PAGE 399  EVENT START.BATTLE                      7984
    PAGE 677  ''PROGRAM  OLDER.VERSION                 428
    PAGE 707                                          2180
UN.TYPE
    PAGE 230  ROUTINE RPV.DETECTION                    603
    PAGE 656  ROUTINE AR.DETECTION                    9436
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                 9584
UN.TYPE.UNIT
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         981
    PAGE  44  ''SECTION FOR DEFINITIONS               2503
    PAGE  58  ROUTINE CREATE.FORCE                    3134
    PAGE  66  ROUTINE FORM.TF.LIST                    3425
    PAGE  68  ROUTINE GENERAL.BATTLE                  3530 3562
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                 3844 3848
    PAGE  80  ROUTINE BLOCK.LOS                       4021 4028 4045
    PAGE  83  ROUTINE CHANGE.LOC                      4148 4154
    PAGE  89  ROUTINE FA.BN.MOVEMENT                  4425
    PAGE 104  ROUTINE MINE.DELAY                      5006 5029
    PAGE 106  ROUTINE MIN.MOVE                        5093
    PAGE 111  ROUTINE PRED.POS                        5332
    PAGE 130  ROUTINE CHECK.DEAD                      6082 6099
    PAGE 133  ROUTINE CHECK.FORCE                     6172
    PAGE 138  ROUTINE CHECK.PROX                      6429
    PAGE 139                                          6446
    PAGE 141  ROUTINE CHECK.STREN                     6513 6517
    PAGE 142  ROUTINE DEAD.UNIT                       6542
    PAGE 145  ROUTINE FIN.BATTLE                      6712
    PAGE 167  ROUTINE CFR.DETECTION                   7692
    PAGE 180  ROUTINE EST.MIL.WORTH                   8227
    PAGE 181  ROUTINE FASCAM.COMPUTATION              8256
    PAGE 183  ROUTINE FA.BN.ASGN                      8366 8370
    PAGE 199  ROUTINE ILLUM.COMPUTATION               9147
    PAGE 206  ROUTINE PDB.DETECTION                   9458
    PAGE 208  ROUTINE PGM.MSN.ASGN                    9534 9538
    PAGE 210                                          9620
    PAGE 219  ROUTINE REQUEST.ILLUM                    44   48   51
    PAGE 223  ROUTINE REQUEST.SMOKE                   264  267  281
    PAGE 233  ROUTINE SMOKE.COMPUTATION               702  704
    PAGE 240  ROUTINE TARGET.ANALYSIS                 1082
    PAGE 242  ROUTINE UNIT.ENVIR                      1164
    PAGE 252  ROUTINE MINE.EFFECTS                    1604
    PAGE 262  ROUTINE BTRY.EFFECTS                    2108
    PAGE 263                                          2139 2141
    PAGE 270                                          2582
    PAGE 272                                          2709
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                 2830
    PAGE 278                                          2905 2953
    PAGE 282  ROUTINE AC.DF.EFFECTS                   3135
    PAGE 284  ROUTINE CAS.EVAL                        3265 3267
```

```
    PAGE 305  ROUTINE UNIT.PRIORITY                4272 4285
    PAGE 324  ROUTINE DECIDE                       5055
    PAGE 384  EVENT OFF.LINE.ATTRITION             7435
    PAGE 392  EVENT SET.DEBUG                      7640
    PAGE 394  EVENT START.ARTY.MOVEMENT            7730
    PAGE 398  EVENT START.BATTLE                   7955
    PAGE 403  EVENT START.MOVE                     8210 8216
    PAGE 421  PROCESS AC.ATK.TGT                   8971 8982
    PAGE 426                                       9255 9288
    PAGE 476  PROCESS TARGET.REPORT                1876
    PAGE 478  PROCESS WITH.DRAW                    1931
    PAGE 480  PROCESS FIRE.MISSION                 2058 2062
    PAGE 483                                       2227
    PAGE 484                                       2281
    PAGE 494  PROCESS SHOOT.OUT                    2814
    PAGE 506  PROCESS CAS.MISSION                  3495
    PAGE 533  ROUTINE UNIT.INPUT                   4702 4706 4729 4735 4736
    PAGE 537                                       4913
    PAGE 539  ROUTINE READ.ORDERS                  5013
    PAGE 595  ROUTINE ANALYSIS.OUTPUT              7257 7286
    PAGE 611  ROUTINE OUTPUT.ATTRITION             7933 7985
    PAGE 613  ROUTINE FOR POSITION.OUT             8004 8012
    PAGE 677  ''PROGRAM  OLDER.VERSION              423
    PAGE 703                                       1924
    PAGE 714  ROUTINE PLAT.COUNT                   2522 2523 2526 2527 2530 2531 2534 2535 2538 2539
UN.X.COORD
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES      982
    PAGE  44  ''SECTION FOR DEFINITIONS            2504
    PAGE  49                                       2823
    PAGE  61  ROUTINE FEBA.INITIAL                 3228 3232
    PAGE  65  ROUTINE FILE.KAD.SENSOR              3383 3384
    PAGE  66  ROUTINE FORM.TF.LIST                 3440 3440
    PAGE  67  ROUTINE GENERAL.BATTLE               3485 3493
    PAGE  71  ROUTINE ORIENTATION                  3671 3675
    PAGE  77  ROUTINE ADJUST                       3912
    PAGE  78                                       3937 3953 3967 3985 3993
    PAGE  83  ROUTINE CHANGE.LOC                   4170
    PAGE  84                                       4233
    PAGE  85                                       4281
    PAGE  87  ROUTINE END.MOVE                     4328 4328
    PAGE  90  ROUTINE FA.BN.MOVEMENT               4455 4455
    PAGE  91                                       4535 4535
    PAGE  95  ROUTINE INIT.REINF                   4660
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA           4810
    PAGE 103  ROUTINE MINE.DELAY                   4975
    PAGE 104                                       4994
    PAGE 110  ROUTINE POSITION                     5276 5279
    PAGE 111  ROUTINE PRED.POS                     5333
    PAGE 119  ROUTINE RESET.FEBA.SECTOR            5687 5689
    PAGE 134  ROUTINE CHECK.FOR.MINES              6229 6230 6236 6238
    PAGE 157  ROUTINE AO.DETECTION                 7274
    PAGE 165  ROUTINE CFR.DEGRADE                  7586 7589 7601
    PAGE 166  ROUTINE CFR.DETECTION                7636 7636
    PAGE 173  ROUTINE DUST.EFFECTS                 7939 7939
    PAGE 174                                       7942 7942 7946 7946 7948
    PAGE 191  ROUTINE FINAL.COVERAGE               8757
```

PAGE 216	ROUTINE REQUEST.FASCAM	9902
PAGE 218	ROUTINE REQUEST.ILLUM	16 35
PAGE 222	ROUTINE REQUEST.SMOKE	203 223
PAGE 227	ROUTINE REQUEST.WD.FASCAM	465
PAGE 230	ROUTINE RPV.DETECTION	597
PAGE 239	ROUTINE TARGET.ANALYSIS	1020 1021 1024 1025
PAGE 254	ROUTINE FO.DETECTION	1701 1701
PAGE 257		1877
PAGE 261	ROUTINE BTRY.EFFECTS	2027
PAGE 267		2404 2405
PAGE 279	ROUTINE AC.BOMB.EFFECTS	2987
PAGE 287	ROUTINE CHECK.CAS.CONSTRAINTS	3441 3442
PAGE 298	ROUTINE HC.COMPUTE.TIMES	3943 3951
PAGE 302	ROUTINE HEL.RANGE.COMPUTE	4135
PAGE 314	ROUTINE FLIGHT.PATH	4657 4661 4666
PAGE 316		4790
PAGE 321	ROUTINE COMPUTE.D	4958
PAGE 322	ROUTINE COMPUTE.WD	4975 4992
PAGE 331	ROUTINE ENQ.FEBA.SET	5317 5319
PAGE 342	ROUTINE RANGE.COMPUTE	5670 5670
PAGE 343	ROUTINE SEARCH.COVERAGE	5687 5689 5694 5696
PAGE 349	EVENT AD.ENGAGEMENT	5918 5936 5945 5946
PAGE 350		5956
PAGE 351		6065
PAGE 352		6088 6099 6104 6106
PAGE 354		6211
PAGE 362	EVENT CFR.OPERATOR	6518
PAGE 371	EVENT FEBA.SORTIE	6817
PAGE 372		6858
PAGE 387	EVENT PDB.OPERATOR	7536
PAGE 393	EVENT START.ARTY.MOVEMENT	7675 7705
PAGE 394		7739
PAGE 399	EVENT START.BATTLE	7995 8000
PAGE 403	EVENT START.MOVE	8220 8226
PAGE 404		8270 8282
PAGE 407	EVENT STOP.ARTY.MOVEMENT	8396
PAGE 408	EVENT UPDATE.LOC	8431 8435 8440 8448 8461
PAGE 416	EVENT ACT.MOVDIS	8792
PAGE 428	PROCESS AIR.OBSERVER	9382
PAGE 430		9481 9485
PAGE 441	PROCESS FORWARD.OBSERVER	58
PAGE 444	PROCESS HC.ARRIVE.BATTLE	171 175 205
PAGE 445		209
PAGE 467	PROCESS REMOTE.PILOT.VEHICLE	1393 1411
PAGE 468		1415
PAGE 484	PROCESS FIRE.MISSION	2272
PAGE 506	PROCESS CAS.MISSION	3515 3516 3521 3522
PAGE 507		3526 3527
PAGE 533	ROUTINE UNIT.INPUT	4694 4731 4731
PAGE 585	ROUTINE TACAIR.INPUT	6961 6962
PAGE 613	ROUTINE FOR POSITION.OUT	8005
PAGE 628	FUNCTION ACT.RANGE	8639 8639
PAGE 632	FUNCTION EST.RANGE	8752
PAGE 634	FUNCTION FEBA.BAND	8803 8806 8821 8821 8824 8824
PAGE 653	PROCESS AIRBORNE.RADAR	9295
PAGE 656	ROUTINE AR.DETECTION	9430

```
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                  9524 9528
    PAGE 659                                           9578
    PAGE 677  ''PROGRAM  OLDER.VERSION                  424
    PAGE 703                                           1925
    PAGE 708                                           2244
UN.X.GRID
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          982
    PAGE  44  ''SECTION FOR DEFINITIONS                2506
    PAGE 439  PROCESS FORWARD.OBSERVER                 9917
    PAGE 654  PROCESS AIRBORNE.RADAR                   9322 9322 9327
    PAGE 677  ''PROGRAM  OLDER.VERSION                  424
    PAGE 703                                           1927
UN.Y.COORD
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES          983
    PAGE  44  ''SECTION FOR DEFINITIONS                2505
    PAGE  49                                           2825
    PAGE  61  ROUTINE FEBA.INITIAL                     3214 3216 3217 3220 3222 3223 3225
    PAGE  65  ROUTINE FILE.KAD.SENSOR                  3385 3386
    PAGE  66  ROUTINE FORM.TF.LIST                     3441 3441
    PAGE  67  ROUTINE GENERAL.BATTLE                   3486 3494
    PAGE  71  ROUTINE ORIENTATION                      3672 3676
    PAGE  77  ROUTINE ADJUST                           3913
    PAGE  78                                           3938 3955 3969 3986 3994
    PAGE  83  ROUTINE CHANGE.LOC                       4171
    PAGE  84                                           4234
    PAGE  85                                           4282
    PAGE  87  ROUTINE END.MOVE                         4330 4330
    PAGE  89  ROUTINE FA.BN.MOVEMENT                   4436
    PAGE  91                                           4516
    PAGE  95  ROUTINE INIT.REINF                       4661
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA               4819
    PAGE 103  ROUTINE MINE.DELAY                       4976
    PAGE 110  ROUTINE POSITION                         5277 5280
    PAGE 111  ROUTINE PRED.POS                         5334
    PAGE 119  ROUTINE RESET.FEBA.SECTOR                5683
    PAGE 134  ROUTINE CHECK.FOR.MINES                  6231 6232 6237
    PAGE 147  ROUTINE INTER.BATTLE                     6825 6825
    PAGE 148                                           6862 6862
    PAGE 157  ROUTINE AO.DETECTION                     7276
    PAGE 165  ROUTINE CFR.DEGRADE                      7592 7593 7602
    PAGE 166  ROUTINE CFR.DETECTION                    7637 7637
    PAGE 174  ROUTINE DUST.EFFECTS                     7940 7940 7945 7945 7948 7952 7952
    PAGE 191  ROUTINE FINAL.COVERAGE                   8758
    PAGE 216  ROUTINE REQUEST.FASCAM                   9903
    PAGE 218  ROUTINE REQUEST.ILLUM                      17   36
    PAGE 222  ROUTINE REQUEST.SMOKE                     204  224
    PAGE 227  ROUTINE REQUEST.WD.FASCAM                 466
    PAGE 230  ROUTINE RPV.DETECTION                     599
    PAGE 239  ROUTINE TARGET.ANALYSIS                  1002
    PAGE 246  ROUTINE VOLLEY                           1319
    PAGE 254  ROUTINE FO.DETECTION                     1702 1702
    PAGE 257                                           1879
    PAGE 261  ROUTINE BTRY.EFFECTS                     2028
    PAGE 267                                           2406 2407
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                  2988
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS            3443 3444
```

```
    PAGE 298  ROUTINE HC.COMPUTE.TIMES                3944 3952
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE               4136
    PAGE 314  ROUTINE FLIGHT.PATH                     4658 4663 4667
    PAGE 316                                          4791
    PAGE 321  ROUTINE COMPUTE.D                       4959
    PAGE 322  ROUTINE COMPUTE.WD                      4976 4993
    PAGE 325  ROUTINE DEQ.FEBA.SET                    5089
    PAGE 331  ROUTINE ENQ.FEBA.SET                    5312
    PAGE 342  ROUTINE RANGE.COMPUTE                   5671 5671
    PAGE 343  ROUTINE SEARCH.COVERAGE                 5687 5689 5694 5696
    PAGE 347  EVENT ACT.REINF                         5842
    PAGE 349  EVENT AD.ENGAGEMENT                     5919 5931 5936 5947 5948
    PAGE 350                                          5957
    PAGE 351                                          6067
    PAGE 352                                          6100 6101 6105 6107 6107
    PAGE 354                                          6213
    PAGE 360  EVENT CFR.ON                            6431
    PAGE 362  EVENT CFR.OPERATOR                      6520
    PAGE 371  EVENT FEBA.SORTIE                       6796
    PAGE 372                                          6839
    PAGE 387  EVENT PDB.OPERATOR                      7538
    PAGE 393  EVENT START.ARTY.MOVEMENT               7680 7705
    PAGE 399  EVENT START.BATTLE                      7996 8001
    PAGE 404  EVENT START.MOVE                        8271 8283
    PAGE 408  EVENT UPDATE.LOC                        8432 8436 8441 8449
    PAGE 409                                          8462
    PAGE 416  EVENT ACT.MOVDIS                        8791
    PAGE 428  PROCESS AIR.OBSERVER                    9383
    PAGE 430                                          9481 9485
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                 172  176  206
    PAGE 445                                           210
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE            1394 1411
    PAGE 468                                          1415
    PAGE 484  PROCESS FIRE.MISSION                    2273
    PAGE 506  PROCESS CAS.MISSION                     3517 3518 3523
    PAGE 507                                          3524 3528 3529
    PAGE 533  ROUTINE UNIT.INPUT                      4695 4732 4732
    PAGE 585  ROUTINE TACAIR.INPUT                    6963 6963
    PAGE 613  ROUTINE FOR POSITION.OUT                8005
    PAGE 628  FUNCTION ACT.RANGE                      8640 8640
    PAGE 632  FUNCTION EST.RANGE                      8753
    PAGE 634  FUNCTION FEBA.BAND                      8812
    PAGE 653  PROCESS AIRBORNE.RADAR                  9296
    PAGE 654                                          9335
    PAGE 656  ROUTINE AR.DETECTION                    9432
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                 9524 9528
    PAGE 659                                          9580
    PAGE 660  FUNCTION STAY.TIME                      9602
    PAGE 677  ''PROGRAM  OLDER.VERSION                 425
    PAGE 703                                          1926
    PAGE 708                                          2246
UN.Y.GRID
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         983
    PAGE  44  ''SECTION FOR DEFINITIONS               2507
    PAGE 439  PROCESS FORWARD.OBSERVER                9918
    PAGE 677  ''PROGRAM  OLDER.VERSION                 425
```

```
    PAGE 703                                          1928
UPDATE.LOC
    PAGE  41  ''SECTION FOR EVENTS                    2330
    PAGE 112  ROUTINE PREPARE.LIST                    5389 5390
    PAGE 406  EVENT START.MOVE                        8356
    PAGE 408  EVENT UPDATE.LOC                        8404
    PAGE 410                                          8575
    PAGE 411                                          8585
    PAGE 602  ROUTINE BETWEEN.ROUTINE                 7648 7649
    PAGE 617  ROUTINE SNAP2                           8195
    PAGE 630  FUNCTION COLLISION                      8684 8685 8688 8689
    PAGE 700  ''PROGRAM  OLDER.VERSION                1767
UPDATE.LOCATION.
    PAGE 325  ROUTINE DEQ.FEBA.SET                    5098
UP.L
    PAGE 112  ROUTINE PREPARE.LIST                    5385 5386 5389 5390
USED.
    PAGE 325  ROUTINE DEQ.FEBA.SET                    5099
USED.DBANK.V
    PAGE  57  ROUTINE MAIN3                           3082
    PAGE 368  EVENT END.SIMULATION                    6666
    PAGE 392  EVENT SET.DEBUG                         7631
    PAGE 520  ROUTINE MAIN2                           4194 4197 4200 4203 4206 4209 4216 4219 4222 4225 4228 4231 4234 4237 4240
    PAGE 521                                          4243 4246 4249 4252 4255 4258 4261 4264 4267 4270 4273 4276 4279 4282 4285
                                                      4288 4291 4296
    PAGE 522                                          4301 4304
    PAGE 599  ROUTINE BETWEEN.ROUTINE                 7437
    PAGE 618  ROUTINE SNAP.R                          8207
US.EQ.ID
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES        1844
    PAGE  45  ''SECTION FOR DEFINITIONS               2571
    PAGE 253  ROUTINE MINE.EFFECTS                    1653
    PAGE 271  ROUTINE BTRY.EFFECTS                    2628
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                 2973
    PAGE 307  ROUTINE AD.SHOOT                        4341
    PAGE 384  EVENT OFF.LINE.ATTRITION                7425
    PAGE 566  ROUTINE SENSOR.INPUT                    6084 6097 6119
    PAGE 567                                          6129
    PAGE 692  ''PROGRAM  OLDER.VERSION                1283
    PAGE 704                                          1992
US.FDC
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES        1841
    PAGE 157  ROUTINE AO.DETECTION                    7270 7301
    PAGE 216  ROUTINE REQUEST.FASCAM                  9929
    PAGE 217                                          9943 9953
    PAGE 219  ROUTINE REQUEST.ILLUM                     59   73   83
    PAGE 223  ROUTINE REQUEST.SMOKE                    232  246  256
    PAGE 230  ROUTINE RPV.DETECTION                    593
    PAGE 256  ROUTINE FO.DETECTION                    1835
    PAGE 257                                          1901
    PAGE 258                                          1946
    PAGE 362  EVENT CFR.OPERATOR                      6508
    PAGE 363                                          6564 6572
    PAGE 387  EVENT PDB.OPERATOR                      7527 7549 7552
    PAGE 566  ROUTINE SENSOR.INPUT                    6079
    PAGE 656  ROUTINE AR.DETECTION                    9427
```

```
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                9574
    PAGE 692  ''PROGRAM  OLDER.VERSION               1280
US.ID
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1842
    PAGE 159  ROUTINE ATTRIT.SENSOR                  7324 7352
    PAGE 238  ROUTINE SWITCH.FO                       932
    PAGE 371  EVENT FEBA.SORTIE                      6792 6809
    PAGE 372                                         6835 6850
    PAGE 429  PROCESS AIR.OBSERVER                   9388
    PAGE 566  ROUTINE SENSOR.INPUT                   6114 6122
    PAGE 567                                         6132 6172
    PAGE 619  ROUTINE SNAP.R                         8287 8289 8293 8298
    PAGE 692  ''PROGRAM  OLDER.VERSION               1281
US.LINK
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1837
    PAGE 162  ROUTINE ATTRIT.SENSOR                  7508
    PAGE 279  ROUTINE AC.BOMB.EFFECTS                2972 2973 2976
    PAGE 371  EVENT FEBA.SORTIE                      6790 6792 6796 6808 6809 6810
    PAGE 372                                         6833 6835 6835 6839 6849 6850 6851
    PAGE 428  PROCESS AIR.OBSERVER                   9387
    PAGE 429                                         9388 9404 9425
    PAGE 431                                         9530 9549
    PAGE 432                                         9580 9613
    PAGE 433                                         9631 9649 9668
    PAGE 440  PROCESS FORWARD.OBSERVER               9994
    PAGE 441                                           40   82
    PAGE 566  ROUTINE SENSOR.INPUT                   6074
    PAGE 619  ROUTINE SNAP.R                         8283 8285 8286 8287 8289 8291 8292 8293 8295 8296 8298
    PAGE 691  ''PROGRAM  OLDER.VERSION               1276
US.LINKS
    PAGE 260  ROUTINE BTRY.EFFECTS                   1971
US.MODEL
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1839
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA             4797
    PAGE 166  ROUTINE CFR.DETECTION                  7628
    PAGE 206  ROUTINE PDB.DETECTION                  9440 9469
    PAGE 212  ROUTINE PIR.DETECTION                  9719
    PAGE 229  ROUTINE RPV.DETECTION                   532
    PAGE 254  ROUTINE FO.DETECTION                   1689
    PAGE 358  EVENT CFR.ACTIVATION                   6369
    PAGE 359  EVENT CFR.OFF                          6393
    PAGE 360  EVENT CFR.ON                           6421
    PAGE 386  EVENT PDB.ACTIVATION                   7495
    PAGE 420  PROCESS AC.ATK.TGT                     8910
    PAGE 428  PROCESS AIR.OBSERVER                   9362
    PAGE 439  PROCESS FORWARD.OBSERVER               9928 9930
    PAGE 441                                           53
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE           1378
    PAGE 566  ROUTINE SENSOR.INPUT                   6077
    PAGE 653  PROCESS AIRBORNE.RADAR                 9277
    PAGE 656  ROUTINE AR.DETECTION                   9397
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                9493
    PAGE 691  ''PROGRAM  OLDER.VERSION               1278
US.SENSOR.TYPE
    PAGE   3  PROGRAM REVISIONS                       125
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES       1838
```

```
    PAGE 159  ROUTINE ATTRIT.SENSOR                    7343 7347 7351
    PAGE 160                                           7378 7416 7417
    PAGE 161                                           7439 7440
    PAGE 162                                           7487 7506 7507
    PAGE 206  ROUTINE PDB.DETECTION                    9441
    PAGE 212  ROUTINE PIR.DETECTION                    9720
    PAGE 216  ROUTINE REQUEST.FASCAM                   9926 9940
    PAGE 217                                           9948
    PAGE 219  ROUTINE REQUEST.ILLUM                      56   70   78
    PAGE 222  ROUTINE REQUEST.SMOKE                     229
    PAGE 223                                            243  251
    PAGE 229  ROUTINE RPV.DETECTION                     533
    PAGE 238  ROUTINE SWITCH.FO                         930
    PAGE 254  ROUTINE FO.DETECTION                     1699
    PAGE 358  EVENT CFR.ACTIVATION                     6365
    PAGE 362  EVENT CFR.OPERATOR                       6489
    PAGE 386  EVENT PDB.ACTIVATION                     7492
    PAGE 387  EVENT PDB.OPERATOR                       7513
    PAGE 419  PROCESS AC.ATK.TGT                       8904
    PAGE 428  PROCESS AIR.OBSERVER                     9363
    PAGE 440  PROCESS FORWARD.OBSERVER                   23   24
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE             1380
    PAGE 566  ROUTINE SENSOR.INPUT                     6076
    PAGE 567                                           6166 6169
    PAGE 619  ROUTINE SNAP.R                           8286 8291 8292 8295 8296
    PAGE 653  PROCESS AIRBORNE.RADAR                   9278
    PAGE 656  ROUTINE AR.DETECTION                     9399
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                  9494
    PAGE 691  ''PROGRAM  OLDER.VERSION                 1277
US.STATUS
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES         1843
    PAGE 159  ROUTINE ATTRIT.SENSOR                    7357
    PAGE 160                                           7408
    PAGE 206  ROUTINE PDB.DETECTION                    9471
    PAGE 238  ROUTINE SWITCH.FO                         944  960  963
    PAGE 254  ROUTINE FO.DETECTION                     1718
    PAGE 358  EVENT CFR.ACTIVATION                     6371
    PAGE 359  EVENT CFR.OFF                            6389 6400
    PAGE 360  EVENT CFR.ON                             6417
    PAGE 361                                           6472
    PAGE 371  EVENT FEBA.SORTIE                        6810
    PAGE 372                                           6835 6851
    PAGE 428  PROCESS AIR.OBSERVER                     9387
    PAGE 429                                           9398 9424
    PAGE 431                                           9519 9537
    PAGE 432                                           9568 9602
    PAGE 433                                           9620 9638 9657
    PAGE 434                                           9683
    PAGE 438  PROCESS FORWARD.OBSERVER                 9891
    PAGE 439                                           9931 9965
    PAGE 440                                           9993
    PAGE 441                                             39   81
    PAGE 469  PROCESS REMOTE.PILOT.VEHICLE             1480
    PAGE 482  PROCESS FIRE.MISSION                     2165
    PAGE 566  ROUTINE SENSOR.INPUT                     6123
    PAGE 692  ''PROGRAM  OLDER.VERSION                 1282
```

```
US.UNIT
    PAGE  32  ''SECTION FOR TEMPORARY_ENTITIES          1840
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA                4783
    PAGE 103  ROUTINE MINE.DELAY                        4972 4974
    PAGE 154  ROUTINE AO.DETECTION                      7109
    PAGE 159  ROUTINE ATTRIT.SENSOR                     7322
    PAGE 162                                            7488 7489
    PAGE 166  ROUTINE CFR.DETECTION                     7629
    PAGE 205  ROUTINE NOISE.DEGRADE                     9399 9411
    PAGE 212  ROUTINE PIR.DETECTION                     9718
    PAGE 229  ROUTINE RPV.DETECTION                      530
    PAGE 254  ROUTINE FO.DETECTION                      1687 1696
    PAGE 358  EVENT CFR.ACTIVATION                      6366
    PAGE 360  EVENT CFR.ON                              6427
    PAGE 362  EVENT CFR.OPERATOR                        6509
    PAGE 371  EVENT FEBA.SORTIE                         6796
    PAGE 372                                            6839
    PAGE 386  EVENT PDB.ACTIVATION                      7494
    PAGE 387  EVENT PDB.OPERATOR                        7528
    PAGE 428  PROCESS AIR.OBSERVER                      9360 9361
    PAGE 438  PROCESS FORWARD.OBSERVER                  9880
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE              1376
    PAGE 566  ROUTINE SENSOR.INPUT                      6078
    PAGE 656  ROUTINE AR.DETECTION                      9396
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                   9491
    PAGE 659                                            9575
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1279
U.FRACT
    PAGE 133  ROUTINE CHECK.FORCE                       6171 6173
VALUES.
    PAGE 212  ROUTINE PIR.DETECTION                     9725
    PAGE 308  ROUTINE AD.SHOOT                          4364
    PAGE 420  PROCESS AC.ATK.TGT                        8935
VALUE.
    PAGE 546  ROUTINE ORD.MOVCOR                        5226
VARIABLE.
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                   105
VISIBILITY
    PAGE   1  ROUTINE FOR CROSS_REFERENCING              54
    PAGE  43  ''SECTION FOR DEFINITIONS                 2428
    PAGE 173  ROUTINE DUST.EFFECTS                      7892
    PAGE 193  ROUTINE FINAL.COVERAGE                    8875
    PAGE 256  ROUTINE FO.DETECTION                      1802
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                   2838
    PAGE 281  ROUTINE AC.DF.EFFECTS                     3104
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS             3417
    PAGE 323  ROUTINE CONTRAST.TO.FREQ                  5010 5011
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION           5747
    PAGE 350  EVENT AD.ENGAGEMENT                       6002
    PAGE 366  EVENT CHANGE.WEATHER                      6623 6627
    PAGE 581  ROUTINE VIS.INPUT                         6724
    PAGE 621  ROUTINE TACAIR.DATA.REPORT                8367 8368
    PAGE 701  ''PROGRAM  OLDER.VERSION                  1858
VISIBLE.HC
    PAGE 378  EVENT HELO.ENGAGEMENT                     7170 7171 7173
    PAGE 489  PROCESS ASSESSMENT                        2542 2543 2546 2547
```

```
VISIBLE.UNIT
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1855
    PAGE  80  ROUTINE BLOCK.LOS                         4061
    PAGE  81                                            4067
    PAGE  93  ROUTINE INITIAL.DETECT                    4603 4607
    PAGE  96  ROUTINE LINE.OF.SIGHT                     4715 4716 4717 4718 4719 4720
    PAGE  97                                            4721 4722
    PAGE 101  ROUTINE LOS.CHECK                         4886 4887 4913 4914 4915 4916 4917 4918 4919 4920
    PAGE 108  ROUTINE NEW.SEGMENT                       5193 5194 5197 5198 5200 5201 5204 5205
    PAGE 109                                            5247 5248 5249 5250 5251 5252 5253 5254
    PAGE 132  ROUTINE CHECK.ENGAGEMENT                  6137 6138 6144 6149
    PAGE 255  ROUTINE FO.DETECTION                      1729 1730
    PAGE 300  ROUTINE HC.DISENGAGE                      4025 4028 4030 4031 4035 4036 4038 4040
    PAGE 311  ROUTINE INTER.HELO                        4564 4567 4569
    PAGE 312                                            4570 4574 4576 4578 4580
    PAGE 329  ROUTINE EMPTY                             5243 5244
    PAGE 369  EVENT ENGAGEMENT                          6706 6707 6713
    PAGE 376  EVENT HELO.ENGAGEMENT                     7068 7069 7071
    PAGE 377                                            7123 7124 7125 7126 7128 7129 7130 7131 7132
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION             827  830  834  835  837  838  841  844  847  848  850  851  864
    PAGE 457                                             866  867
    PAGE 459                                             991  994
    PAGE 461                                            1131
    PAGE 478  PROCESS WITH.DRAW                         1968 1969
    PAGE 487  PROCESS ASSESSMENT                        2433 2434 2439
    PAGE 488                                            2526
    PAGE 489                                            2529 2531 2547
    PAGE 498  PROCESS SHOOT.OUT                         3042 3043 3057 3058
    PAGE 619  ROUTINE SNAP.R                            8267
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1294
    PAGE 713                                            2501
VISIBLE.UNITS
    PAGE 300  ROUTINE HC.DISENGAGE                      4011
VIS.FAC
    PAGE 208  ROUTINE PGM.MSN.ASGN                      9497
    PAGE 210                                            9622 9624 9626
    PAGE 393  EVENT START.ARTY.MOVEMENT                 7665
    PAGE 394                                            7713 7715 7731
    PAGE 403  EVENT START.MOVE                          8193 8204 8206 8210 8216
VIS.INPUT
    PAGE 521  ROUTINE MAIN2                             4290 4292
    PAGE 581  ROUTINE VIS.INPUT                         6711
VOLLEY.RAD.OF.EFFECTS
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS                  1425 1432 1445 1449 1451
VU.POINTER
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1856
    PAGE  80  ROUTINE BLOCK.LOS                         4057
    PAGE  81                                            4063
    PAGE  93  ROUTINE INITIAL.DETECT                    4607
    PAGE  96  ROUTINE LINE.OF.SIGHT                     4716 4720
    PAGE 101  ROUTINE LOS.CHECK                         4887 4914 4918
    PAGE 108  ROUTINE NEW.SEGMENT                       5194 5201
    PAGE 109                                            5248 5252
    PAGE 132  ROUTINE CHECK.ENGAGEMENT                  6149
    PAGE 201  ROUTINE ILLUM.EFFECTS                     9278
    PAGE 255  ROUTINE FO.DETECTION                      1730
```

```
    PAGE 300  ROUTINE HC.DISENGAGE                    4036
    PAGE 312  ROUTINE INTER.HELO                      4576
    PAGE 369  EVENT ENGAGEMENT                        6707
    PAGE 376  EVENT HELO.ENGAGEMENT                   7069
    PAGE 377                                          7124 7129
    PAGE 378                                          7171
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION           830  835  844  848
    PAGE 457                                           866
    PAGE 459                                           994
    PAGE 461                                          1127 1128 1130
    PAGE 478  PROCESS WITH.DRAW                       1969
    PAGE 487  PROCESS ASSESSMENT                      2434
    PAGE 489                                          2543
    PAGE 498  PROCESS SHOOT.OUT                       3043 3058
    PAGE 692  ''PROGRAM  OLDER.VERSION                1295
VU.PREV.ENG
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES        1858
    PAGE 376  EVENT HELO.ENGAGEMENT                   7071
    PAGE 377                                          7131
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION           867
    PAGE 692  ''PROGRAM  OLDER.VERSION                1297
VU.STATUS
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES        1857
    PAGE  96  ROUTINE LINE.OF.SIGHT                   4717
    PAGE  97                                          4721
    PAGE 101  ROUTINE LOS.CHECK                       4915 4919
    PAGE 109  ROUTINE NEW.SEGMENT                     5249 5253
    PAGE 132  ROUTINE CHECK.ENGAGEMENT                6138
    PAGE 201  ROUTINE ILLUM.EFFECTS                   9279
    PAGE 369  EVENT ENGAGEMENT                        6713
    PAGE 377  EVENT HELO.ENGAGEMENT                   7125 7130
    PAGE 378                                          7173
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION           837  850
    PAGE 487  PROCESS ASSESSMENT                      2439
    PAGE 692  ''PROGRAM  OLDER.VERSION                1296
WAIT.TIME
    PAGE 453  PROCESS HC.RETURN.FARRP                  672  675  677  682
WD.DESTRUCT.INDIC
    PAGE  39  ''SECTION FOR PROCESSES                 2197
    PAGE 328  ROUTINE EMPTY                           5213
    PAGE 478  PROCESS WITH.DRAW                       1926
    PAGE 698  ''PROGRAM  OLDER.VERSION                1635
WD.DIST
    PAGE  11  ''SECTION FOR PERMANENT_ENTITIES         587
    PAGE 115  ROUTINE PROX.CHECK                      5541
    PAGE 570  ROUTINE DECISION.INPUT                  6276
    PAGE 670  ''PROGRAM  OLDER.VERSION                  28
WD.NAME
    PAGE  47  ''SECTION FOR DEFINITIONS               2703
    PAGE 706  ''PROGRAM  OLDER.VERSION                2127
WD.UNIT
    PAGE  39  ''SECTION FOR PROCESSES                 2196
    PAGE 328  ROUTINE EMPTY                           5209
    PAGE 698  ''PROGRAM  OLDER.VERSION                1634
WEAPON
    PAGE   1  ROUTINE FOR CROSS_REFERENCING             50
```

```
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1875
    PAGE  49  ''SECTION FOR DEFINITIONS                 2781
    PAGE 149  ROUTINE PK.COMPUTE                        6943 6947 6948 6950
    PAGE 150                                            7002 7007
    PAGE 151                                            7056
    PAGE 351  EVENT AD.ENGAGEMENT                       6070
    PAGE 382  EVENT OFF.LINE.ATTRITION                  7288 7335 7335
    PAGE 383                                            7341 7341 7351 7351
    PAGE 420  PROCESS AC.ATK.TGT                        8945
    PAGE 426                                            9282
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                   103
    PAGE 445                                             244
    PAGE 447                                             355
    PAGE 493  PROCESS SHOOT.OUT                         2763
    PAGE 495                                            2920
    PAGE 496                                            2943
    PAGE 514  PROCESS HELICOPTER.FIRE                   3911
    PAGE 530  ROUTINE EQ.TE.INPUT                       4627
    PAGE 535  ROUTINE UNIT.INPUT                        4814 4817 4830 4838 4840
    PAGE 588  ROUTINE AC.MUNS.INPUT                     7055 7063 7064
    PAGE 593  ROUTINE AMMO.RPT                          7230
    PAGE 596  ROUTINE ANALYSIS.OUTPUT                   7316 7317 7317
    PAGE 611  ROUTINE OUTPUT.ATTRITION                  7944 7945 7945
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1314
    PAGE 707                                            2203
    PAGE 713                                            2502
WEAPON.
    PAGE 420  PROCESS AC.ATK.TGT                        8948
WEIBULL.F
    PAGE  97  ROUTINE LINE.OF.SIGHT                     4738 4739 4757 4758
    PAGE 109  ROUTINE NEW.SEGMENT                       5256 5257
    PAGE 210  ROUTINE PGM.MSN.ASGN                      9618 9619
    PAGE 479  PROCESS WITH.DRAW                         1983 1984 2006 2007
    PAGE 642  ROUTINE WEIBULL.F                         9062
WEIGHTED.VOLS
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS                  1374 1388 1396
    PAGE 249                                            1441 1451 1455
WEIGHT.VOLS
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS                  1410 1419
WHAT.NEXT
    PAGE 125  ROUTINE WHAT.NEXT                         5873
    PAGE 356  EVENT BTL.ENDED                           6258
    PAGE 357                                            6333 6336
WIDTH.ADJ.FAT
    PAGE 191  ROUTINE FINAL.COVERAGE                    8735 8736 8740
WIDTH.DAM
    PAGE 191  ROUTINE FINAL.COVERAGE                    8740 8741 8746 8747 8760 8767 8768
    PAGE 193                                            8842 8843 8857 8864
WIDTH.TGT
    PAGE 178  ROUTINE EST.COVERAGE                      8150
    PAGE 179                                            8198 8201 8204 8210
    PAGE 190  ROUTINE FINAL.COVERAGE                    8699
    PAGE 191                                            8760 8769 8770
    PAGE 192                                            8802 8825
    PAGE 193                                            8842 8843 8848 8856
```

```
WIDTH.VOL
    PAGE 191  ROUTINE FINAL.COVERAGE                    8738 8740
WIDTH.1RND.PAT
    PAGE 178  ROUTINE EST.COVERAGE                      8156 8165 8168 8170
    PAGE 190  ROUTINE FINAL.COVERAGE                    8705 8713 8714
    PAGE 191                                            8723 8735
    PAGE 192                                            8824
    PAGE 193                                            8855
WINNER.
    PAGE   2  PROGRAM REVISIONS                          87
    PAGE 356  EVENT BTL.ENDED                           6253 6297 6298 6306
WIN.
    PAGE 125  ROUTINE WHAT.NEXT                         5875
WITHDRAWING.
    PAGE 115  ROUTINE PROX.CHECK                        5506
WITHIN.
    PAGE 325  ROUTINE DEQ.FEBA.SET                      5096
WITH.DRAW
    PAGE  39  ''SECTION FOR PROCESSES                   2195
    PAGE 115  ROUTINE PROX.CHECK                        5553
    PAGE 133  ROUTINE CHECK.FORCE                       6195
    PAGE 324  ROUTINE DECIDE                            5061
    PAGE 328  ROUTINE EMPTY                             5208 5209 5211 5212 5213 5214
    PAGE 478  PROCESS WITH.DRAW                         1914 1920 1926
    PAGE 600  ROUTINE BETWEEN.ROUTINE                   7520 7521
    PAGE 616  ROUTINE SNAP2                             8160
    PAGE 698  ''PROGRAM  OLDER.VERSION                  1633
WITH.DRAW.FORCE
    PAGE 145  ROUTINE FIN.BATTLE                        6693 6704
WOODS.
    PAGE 242  ROUTINE UNIT.ENVIR                        1123
WOOD.PCT
    PAGE 242  ROUTINE UNIT.ENVIR                        1137 1167
    PAGE 243                                            1178 1185 1189 1194 1197 1208
    PAGE 244                                            1230 1239 1273
WPN.AC.MUNS
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1877
    PAGE  44  ''SECTION FOR DEFINITIONS                 2488
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                   2808
    PAGE 279                                            2983
    PAGE 295  ROUTINE END.CAS.MISSION                   3841 3842
    PAGE 421  PROCESS AC.ATK.TGT                        8989
    PAGE 423                                            9133
    PAGE 535  ROUTINE UNIT.INPUT                        4840 4842 4843 4844
    PAGE 588  ROUTINE AC.MUNS.INPUT                     7059 7064
    PAGE 692  ''PROGRAM  OLDER.VERSION                  1316
    PAGE 702                                            1909
WPN.ID
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES          1876
    PAGE 123  ROUTINE TIME.TO.DETECT                    5824 5825 5826 5830 5833
    PAGE 149  ROUTINE PK.COMPUTE                        6947 6948 6950
    PAGE 150                                            7002 7007
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                   2809
    PAGE 280  ROUTINE AC.DF.EFFECTS                     3043
    PAGE 295  ROUTINE END.CAS.MISSION                   3838 3839
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE                 4169 4170
```

```
    PAGE 307  ROUTINE AD.SHOOT                      4339
    PAGE 309                                        4422 4425 4438 4441
    PAGE 352  EVENT AD.ENGAGEMENT                   6081
    PAGE 420  PROCESS AC.ATK.TGT                    8939 8940
    PAGE 421                                        8998 8999 9010
    PAGE 422                                        9023 9026 9027
    PAGE 446  PROCESS HC.ARRIVE.BATTLE               275  277  288  292
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION         911  912  914  922
    PAGE 458                                         925
    PAGE 459                                        1022 1023 1025 1029 1032
    PAGE 487  PROCESS ASSESSMENT                    2464 2467
    PAGE 489                                        2578 2581
    PAGE 490                                        2616 2619
    PAGE 491                                        2670 2673
    PAGE 492                                        2723 2726
    PAGE 493  PROCESS SHOOT.OUT                     2778
    PAGE 499                                        3104 3105 3112 3117 3120 3127 3138 3143 3146
    PAGE 500                                        3176 3179 3195 3198
    PAGE 501                                        3217 3220 3245 3248
    PAGE 502                                        3282 3285 3288 3292 3298 3301
    PAGE 512  PROCESS HELICOPTER.FIRE               3809
    PAGE 514                                        3894 3909 3918 3924 3929 3934 3940
    PAGE 515                                        3949 3972 3997
    PAGE 516                                        4011 4014
    PAGE 517                                        4076 4090 4093 4115
    PAGE 518                                        4135 4149 4152
    PAGE 535  ROUTINE UNIT.INPUT                    4824 4828
    PAGE 596  ROUTINE ANALYSIS.OUTPUT               7317
    PAGE 611  ROUTINE OUTPUT.ATTRITION              7945
    PAGE 692  ''PROGRAM  OLDER.VERSION              1315
WPN.QUANT
    PAGE 595  ROUTINE ANALYSIS.OUTPUT               7243 7247
    PAGE 596                                        7317 7325 7328
WPN.QUANTITY
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES      1879
    PAGE 123  ROUTINE TIME.TO.DETECT                5823 5829
    PAGE 280  ROUTINE AC.DF.EFFECTS                 3049 3052
    PAGE 281                                        3055 3058 3109
    PAGE 308  ROUTINE AD.SHOOT                      4411
    PAGE 310                                        4495
    PAGE 421  PROCESS AC.ATK.TGT                    8995 8999 9009
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION         909  921
    PAGE 459                                        1021 1028
    PAGE 535  ROUTINE UNIT.INPUT                    4835
    PAGE 596  ROUTINE ANALYSIS.OUTPUT               7317
    PAGE 611  ROUTINE OUTPUT.ATTRITION              7945
    PAGE 692  ''PROGRAM  OLDER.VERSION              1318
WPN.RDS.REMAIN
    PAGE 446  PROCESS HC.ARRIVE.BATTLE               294
WPN.ROUNDS.REMAINING
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES      1880
    PAGE 309  ROUTINE AD.SHOOT                      4413
    PAGE 310                                        4492 4494
    PAGE 446  PROCESS HC.ARRIVE.BATTLE               287  291
    PAGE 512  PROCESS HELICOPTER.FIRE               3794
    PAGE 514                                        3891 3901 3927
```

```
    PAGE 692  ''PROGRAM  OLDER.VERSION                1319
WRITE.V
    PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT           8809
    PAGE 619  ROUTINE SNAP.R                          8313
WS.NAME
    PAGE  47  ''SECTION FOR DEFINITIONS               2704
    PAGE 706  ''PROGRAM  OLDER.VERSION                2128
WT.VOL
    PAGE 176  ROUTINE EST.COVERAGE                    8005
W.
    PAGE 664  ''PROGRAM  OLDER.VERSION                9679 9680
W.ST.DEV
    PAGE 242  ROUTINE UNIT.ENVIR                      1139
    PAGE 243                                          1208
    PAGE 244                                          1230 1273
W.TGT
    PAGE 192  ROUTINE FINAL.COVERAGE                  8829
    PAGE 193                                          8861
W.1RND
    PAGE 192  ROUTINE FINAL.COVERAGE                  8828
    PAGE 193                                          8860
XCOR.
    PAGE 403  EVENT START.MOVE                        8180 8220 8226
    PAGE 404                                          8263
    PAGE 406                                          8359
    PAGE 408  EVENT UPDATE.LOC                        8407 8431 8435 8443
    PAGE 410                                          8573
    PAGE 411                                          8588
    PAGE 416  EVENT ACT.MOVDIS                        8792 8796
XDIF.
    PAGE 408  EVENT UPDATE.LOC                        8435 8437 8440
XMIT.TIME
    PAGE 362  EVENT CFR.OPERATOR                      6485 6492 6498
    PAGE 387  EVENT PDB.OPERATOR                      7511 7516 7518
XMT.TIME
    PAGE 428  PROCESS AIR.OBSERVER                    9349
    PAGE 432                                          9562 9564 9565 9567
XX.BLUE
    PAGE  72  ROUTINE ORIENTATION                     3692 3701 3701 3704 3704 3719 3719 3727 3738
    PAGE  73                                          3760 3774 3779 3803
    PAGE  74                                          3808
XX.RED
    PAGE  72  ROUTINE ORIENTATION                     3692 3711 3711 3714 3714 3721 3721 3727 3738
    PAGE  73                                          3760 3765 3779
    PAGE  74                                          3813 3818
X.BLUE
    PAGE  67  ROUTINE GENERAL.BATTLE                  3485 3488 3488 3498
    PAGE  71  ROUTINE ORIENTATION                     3638 3662 3671 3681 3681 3688
    PAGE  72                                          3736
    PAGE  73                                          3761 3775 3780 3799
    PAGE  74                                          3809
    PAGE 397  EVENT START.BATTLE                      7911
    PAGE 399                                          8017
    PAGE 400                                          8043
X.DAM
    PAGE 191  ROUTINE FINAL.COVERAGE                  8755 8759
```

```
    PAGE 192                                        8784 8785
X.FAC
    PAGE 260  ROUTINE BTRY.EFFECTS                  1991
    PAGE 267                                        2374 2378 2381 2382
X.PERPENDICULAR
    PAGE 428  PROCESS AIR.OBSERVER                  9344
    PAGE 430                                        9481 9483 9484 9489 9491
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1367 1411 1413
    PAGE 468                                        1414 1419 1421
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9483 9524 9526 9527 9529 9531
X.RED
    PAGE  67  ROUTINE GENERAL.BATTLE                3493 3496 3496 3498
    PAGE  71  ROUTINE ORIENTATION                   3640 3675 3683 3683 3688
    PAGE  72                                        3736
    PAGE  73                                        3761 3766 3780 3799
    PAGE  74                                        3819
    PAGE 397  EVENT START.BATTLE                    7913
    PAGE 399                                        8024 8032
X.TGT
    PAGE 191  ROUTINE FINAL.COVERAGE                8757 8759
    PAGE 192                                        8786 8787
    PAGE 193                                        8879
X.0
    PAGE  77  ROUTINE ADJUST                        3882 3889 3894 3912 3919
    PAGE  78                                        3954 3963 3964 3968
X.1
    PAGE  78  ROUTINE ADJUST                        3947 3951 3953 3967 3978 3979
X.2
    PAGE  78  ROUTINE ADJUST                        3949 3951 3954 3961 3968 3977 3978 3979 3983
X.3
    PAGE  78  ROUTINE ADJUST                        3954 3956 3968 3970
YCOR.
    PAGE 403  EVENT START.MOVE                      8181
    PAGE 404                                        8264
    PAGE 406                                        8360
    PAGE 408  EVENT UPDATE.LOC                      8408 8432 8436 8444
    PAGE 410                                        8573
    PAGE 411                                        8589
    PAGE 416  EVENT ACT.MOVDIS                      8791 8797
YDIF.
    PAGE 408  EVENT UPDATE.LOC                      8436 8437 8441
YY.BLUE
    PAGE  72  ROUTINE ORIENTATION                   3692 3702 3702 3705 3705 3720 3720 3727 3737
    PAGE  73                                        3759 3773 3778 3804
    PAGE  74                                        3808
YY.RED
    PAGE  72  ROUTINE ORIENTATION                   3692 3712 3712 3715 3715 3722 3722 3727 3737
    PAGE  73                                        3759 3764 3778
    PAGE  74                                        3814 3818
Y.BLUE
    PAGE  67  ROUTINE GENERAL.BATTLE                3486 3489 3489 3499
    PAGE  71  ROUTINE ORIENTATION                   3639 3662 3672 3682 3682 3688
    PAGE  72                                        3735
    PAGE  73                                        3762 3776 3781 3799
    PAGE  74                                        3810
    PAGE 397  EVENT START.BATTLE                    7912
```

```
    PAGE 399                                        8018
    PAGE 400                                        8044
Y.COORD
    PAGE 100  ROUTINE LOCATE.SECTOR                 4832 4846 4850 4856 4866 4867
Y.DAM
    PAGE 191  ROUTINE FINAL.COVERAGE                8756 8760 8767 8768
Y.FAC
    PAGE 260  ROUTINE BTRY.EFFECTS                  1991
    PAGE 267                                        2376 2378 2381 2383
Y.PERPENDICULAR
    PAGE 428  PROCESS AIR.OBSERVER                  9344
    PAGE 430                                        9483 9485 9490 9492
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE          1367 1413
    PAGE 468                                        1415 1420 1422
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9483 9526 9528 9530
    PAGE 659                                        9532
Y.RED
    PAGE  67  ROUTINE GENERAL.BATTLE                3494 3497 3497 3499
    PAGE  71  ROUTINE ORIENTATION                   3641 3676 3684 3684 3688
    PAGE  72                                        3735
    PAGE  73                                        3762 3767 3781 3799
    PAGE  74                                        3820
    PAGE 397  EVENT START.BATTLE                    7914
    PAGE 399                                        8025
    PAGE 400                                        8033
Y.TGT
    PAGE 191  ROUTINE FINAL.COVERAGE                8758 8760 8769 8770
    PAGE 193                                        8880
Y.0
    PAGE  77  ROUTINE ADJUST                        3883 3889 3894 3913 3920
    PAGE  78                                        3955 3961 3969
Y.1
    PAGE  78  ROUTINE ADJUST                        3946 3950 3980 3981
Y.2
    PAGE  78  ROUTINE ADJUST                        3948 3950 3961 3977 3980 3981 3983
Y.3
    PAGE  78  ROUTINE ADJUST                        3961 3963 3964
ZERO.REP.TMS
    PAGE 176  ROUTINE EST.COVERAGE                  8019
    PAGE 179                                        8183 8186
ZTIME.F
    PAGE  57  ROUTINE MAIN3                         3082
    PAGE 368  EVENT END.SIMULATION                  6666
    PAGE 392  EVENT SET.DEBUG                       7631
    PAGE 520  ROUTINE MAIN2                         4194
    PAGE 599  ROUTINE BETWEEN.ROUTINE               7437 7440
    PAGE 618  ROUTINE SNAP.R                        8207
Z.FAC
    PAGE 260  ROUTINE BTRY.EFFECTS                  1991
    PAGE 267                                        2380 2384
####################.C
    PAGE 352  EVENT AD.ENGAGEMENT                   6111 6121
00.0
    PAGE 387  EVENT PDB.OPERATOR                    7536 7538
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9578 9580
```

```
0.
    PAGE 149  ROUTINE PK.COMPUTE                    6941 6949
    PAGE 156  ROUTINE AO.DETECTION                  7232
    PAGE 166  ROUTINE CFR.DETECTION                 7647 7662
    PAGE 192  ROUTINE FINAL.COVERAGE                8805 8814 8815
    PAGE 265  ROUTINE BTRY.EFFECTS                  2272
    PAGE 267                                        2375
    PAGE 273                                        2719
    PAGE 281  ROUTINE AC.DF.EFFECTS                 3076
    PAGE 308  ROUTINE AD.SHOOT                      4371
    PAGE 351  EVENT AD.ENGAGEMENT                   6043
    PAGE 362  EVENT CFR.OPERATOR                    6519 6521
    PAGE 422  PROCESS AC.ATK.TGT                    9067
    PAGE 423                                        9101
    PAGE 432  PROCESS AIR.OBSERVER                  9564 9565
    PAGE 546  ROUTINE ORD.MOVCOR                    5227
    PAGE 568  ROUTINE SENSOR.INPUT                  6190
    PAGE 636  FUNCTION HE.WLA                       8893
0.0
    PAGE  51  ''SECTION FOR SUBSTITUTIONS           2901
    PAGE 104  ROUTINE MINE.DELAY                    5003
    PAGE 157  ROUTINE AO.DETECTION                  7248 7275 7277
    PAGE 166  ROUTINE CFR.DETECTION                 7638 7639 7645 7646 7653
    PAGE 191  ROUTINE FINAL.COVERAGE                8729 8730 8731 8762 8763
    PAGE 192                                        8831
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ          9324
    PAGE 206  ROUTINE PDB.DETECTION                 9474
    PAGE 210  ROUTINE PGM.MSN.ASGN                  9628
    PAGE 212  ROUTINE PIR.DETECTION                 9753 9753
    PAGE 229  ROUTINE RPV.DETECTION                  566  566
    PAGE 230                                         598  600
    PAGE 231  ROUTINE SIZE.ESTIMATE                  649  657  661  667  678  679
    PAGE 232                                         684
    PAGE 242  ROUTINE UNIT.ENVIR                    1156 1157
    PAGE 243                                        1217 1218
    PAGE 244                                        1235 1236 1246 1247 1268 1269 1277 1278
    PAGE 255  ROUTINE FO.DETECTION                  1770
    PAGE 257                                        1875 1876
    PAGE 264  ROUTINE BTRY.EFFECTS                  2228 2233 2234 2235 2236 2237 2238 2239
    PAGE 265                                        2278
    PAGE 266                                        2348 2356 2364
    PAGE 269                                        2540
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2843 2844
    PAGE 277                                        2892
    PAGE 293  ROUTINE END.CAS.MISSION               3708
    PAGE 318  ROUTINE FLIGHT.PATH                   4890 4899
    PAGE 323  ROUTINE CONTRAST.TO.FREQ              5021 5028
    PAGE 335  ROUTINE FRAC.COMPUTE                  5422
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION       5749
    PAGE 360  EVENT CFR.ON                          6446 6454 6459
    PAGE 361                                        6464 6465
    PAGE 424  PROCESS AC.ATK.TGT                    9184
    PAGE 430  PROCESS AIR.OBSERVER                  9503
    PAGE 446  PROCESS HC.ARRIVE.BATTLE               320
    PAGE 447                                         348
    PAGE 451  PROCESS HC.RETURN.FARRP                525  539
```

```
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1428 1438
    PAGE 481  PROCESS FIRE.MISSION                  2109
    PAGE 482                                        2153
    PAGE 494  PROCESS SHOOT.OUT                     2828 2829
    PAGE 505  PROCESS CAS.MISSION                   3419
    PAGE 507                                        3541
    PAGE 508                                        3626
    PAGE 581  ROUTINE VIS.INPUT                     6723
    PAGE 584  ROUTINE TACAIR.INPUT                  6876
    PAGE 585                                        6966
    PAGE 636  FUNCTION HE.WLA                       8880 8886
    PAGE 638  FUNCTION ICM.WLA                      8974 8982
    PAGE 640  ROUTINE EXPONENTIAL.F                 9030 9034
    PAGE 641  ROUTINE NORMAL.F                      9055
    PAGE 645  ROUTINE LINE.CIRCLE                   9145
    PAGE 646  ROUTINE MRT.TO.FREQ                   9166 9171 9172
    PAGE 656  ROUTINE AR.DETECTION                  9407 9431 9433
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9538
    PAGE 709  ''PROGRAM  OLDER.VERSION              2309
0.00
    PAGE 242  ROUTINE UNIT.ENVIR                    1129
    PAGE 243                                        1220 1224 1225
    PAGE 244                                        1254 1258 1283
    PAGE 642  ROUTINE WEIBULL.F                     9066
0.000
    PAGE   2  PROGRAM REVISIONS                       74
0.000001
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS              1439
0.00001
    PAGE 278  ROUTINE AC.BOMB.EFFECTS               2923
0.0001
    PAGE 265  ROUTINE BTRY.EFFECTS                  2277
    PAGE 266                                        2347 2355 2363
0.01
    PAGE 231  ROUTINE SIZE.ESTIMATE                  658
0.016
    PAGE 263  ROUTINE BTRY.EFFECTS                  2154
0.025
    PAGE 323  ROUTINE CONTRAST.TO.FREQ              5018
0.0445
    PAGE 646  ROUTINE MRT.TO.FREQ                   9175
0.050
    PAGE   2  PROGRAM REVISIONS                       77
0.0515
    PAGE 323  ROUTINE CONTRAST.TO.FREQ              5019
0.0895
    PAGE 205  ROUTINE NOISE.DEGRADE                 9423
0.0968
    PAGE 323  ROUTINE CONTRAST.TO.FREQ              5026
0.10
    PAGE 243  ROUTINE UNIT.ENVIR                    1207 1208 1209
0.14
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION       5754
0.1473
    PAGE 646  ROUTINE MRT.TO.FREQ                   9175
```

```
0.15
    PAGE 243  ROUTINE UNIT.ENVIR                    1202
0.1566
    PAGE 205  ROUTINE NOISE.DEGRADE                 9418
0.17
    PAGE 243  ROUTINE UNIT.ENVIR                    1177 1185
0.2
    PAGE 165  ROUTINE CFR.DEGRADE                   7608
0.20
    PAGE 242  ROUTINE UNIT.ENVIR                    1155
0.212
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION       5756
0.2251
    PAGE 323  ROUTINE CONTRAST.TO.FREQ              5016
0.307
    PAGE 178  ROUTINE EST.COVERAGE                  8163 8165 8165
    PAGE 191  ROUTINE FINAL.COVERAGE                8722 8723
0.33
    PAGE 243  ROUTINE UNIT.ENVIR                    1189 1193
0.333
    PAGE 115  ROUTINE PROX.CHECK                    5531
0.4
    PAGE 165  ROUTINE CFR.DEGRADE                   7611 7613
0.50
    PAGE 243  ROUTINE UNIT.ENVIR                    1196 1197
0.5687
    PAGE 323  ROUTINE CONTRAST.TO.FREQ              5026
0.573
    PAGE 178  ROUTINE EST.COVERAGE                  8144 8145 8146 8147
    PAGE 190  ROUTINE FINAL.COVERAGE                8694 8695 8696 8696 8697 8697
0.667
    PAGE 115  ROUTINE PROX.CHECK                    5530
0.67
    PAGE 243  ROUTINE UNIT.ENVIR                    1188 1194
0.7
    PAGE 340  ROUTINE PROB.INF                      5623
0.80
    PAGE 428  PROCESS AIR.OBSERVER                  9380
0.83
    PAGE 243  ROUTINE UNIT.ENVIR                    1178 1184
0.980
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION       5754
0.99
    PAGE 231  ROUTINE SIZE.ESTIMATE                  655
0.995
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS              1420
10000.
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION       9809 9815 9818
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS              1416
    PAGE 638  FUNCTION ICM.WLA                      9004
1000.
    PAGE 156  ROUTINE AO.DETECTION                  7229 7233 7236
    PAGE 178  ROUTINE EST.COVERAGE                  8152 8156
    PAGE 190  ROUTINE FINAL.COVERAGE                8702
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION       5746 5747 5759
    PAGE 356  EVENT BTL.ENDED                       6266
```

```
    PAGE 395  EVENT START.BATTLE                    7796
    PAGE 484  PROCESS FIRE.MISSION                  2259
    PAGE 485                                        2321
    PAGE 638  FUNCTION ICM.WLA                      8986
100.
    PAGE  63  ROUTINE FILE.FD.SCHD                  3270
    PAGE 138  ROUTINE CHECK.PROX                    6429
    PAGE 139                                        6446
    PAGE 157  ROUTINE AO.DETECTION                  7286
    PAGE 166  ROUTINE CFR.DETECTION                 7660 7661
    PAGE 167                                        7678 7691
    PAGE 190  ROUTINE FINAL.COVERAGE                8701 8712 8716
    PAGE 194  ROUTINE FIND.START.TIME               8906 8911 8928
    PAGE 195                                        8987
    PAGE 206  ROUTINE PDB.DETECTION                 9449 9457
    PAGE 211  ROUTINE PGM.MSN.ASGN                  9677
    PAGE 213  ROUTINE PIR.DETECTION                 9783
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION       9810
    PAGE 230  ROUTINE RPV.DETECTION                  614
    PAGE 231  ROUTINE SIZE.ESTIMATE                  653
    PAGE 242  ROUTINE UNIT.ENVIR                    1166 1167 1168
    PAGE 255  ROUTINE FO.DETECTION                  1752 1769
    PAGE 256                                        1793
    PAGE 262  ROUTINE BTRY.EFFECTS                  2130
    PAGE 263                                        2178 2192
    PAGE 265                                        2265
    PAGE 266                                        2315 2316 2324 2325 2333 2334 2335 2337 2338 2339
    PAGE 281  ROUTINE AC.DF.EFFECTS                 3077 3098
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE             4154
    PAGE 308  ROUTINE AD.SHOOT                      4392
    PAGE 361  EVENT CFR.ON                          6468 6470
    PAGE 473  PROCESS TARGET.REPORT                 1691
    PAGE 504  PROCESS CAS.MISSION                   3402
    PAGE 548  ROUTINE TB.INPUT                      5339
    PAGE 556  ROUTINE MUNS.INPUT                    5683
    PAGE 593  ROUTINE AMMO.RPT                      7182 7198 7215
    PAGE 606  ROUTINE KV.PRINT                      7731
    PAGE 609  ROUTINE KV.SCOREBOARD                 7875
    PAGE 629  FUNCTION BTRY.AVAILABLE               8656
    PAGE 636  FUNCTION HE.WLA                       8910
    PAGE 637                                        8917 8936 8945 8948 8950
    PAGE 638  FUNCTION ICM.WLA                      9010
    PAGE 639                                        9018 9019
    PAGE 656  ROUTINE AR.DETECTION                  9441
100.0
    PAGE 481  PROCESS FIRE.MISSION                  2095
100.00
    PAGE 245  ROUTINE UNIT.ENVIR                    1291 1292 1293
10.
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES       722  723
    PAGE  16                                         889  890
    PAGE 150  ROUTINE PK.COMPUTE                    6971
    PAGE 185  ROUTINE FA.BN.ASGN                    8505
    PAGE 190  ROUTINE FINAL.COVERAGE                8707 8709 8720
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ          9355
    PAGE 206  ROUTINE PDB.DETECTION                 9473 9474
```

```
    PAGE 265  ROUTINE BTRY.EFFECTS                    2259 2264
    PAGE 266                                          2326 2336
    PAGE 308  ROUTINE AD.SHOOT                        4365
    PAGE 343  ROUTINE SEARCH.COVERAGE                 5693 5695
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION         5761
    PAGE 359  EVENT CFR.OFF                           6394
    PAGE 360  EVENT CFR.ON                            6426
    PAGE 362  EVENT CFR.OPERATOR                      6493
    PAGE 387  EVENT PDB.OPERATOR                      7517
    PAGE 432  PROCESS AIR.OBSERVER                    9563
    PAGE 440  PROCESS FORWARD.OBSERVER                  27
    PAGE 463  PROCESS HOW.REPAIR                      1190 1204
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE            1414 1454
    PAGE 475  PROCESS TARGET.REPORT                   1809
    PAGE 481  PROCESS FIRE.MISSION                    2128 2133
    PAGE 482                                          2144
    PAGE 517  PROCESS HELICOPTER.FIRE                 4115
    PAGE 523  ROUTINE SYS.INPUT                       4365 4366 4367 4368
    PAGE 527  ROUTINE CAT.TU.INPUT                    4466
    PAGE 533  ROUTINE UNIT.INPUT                      4731 4732
    PAGE 548  ROUTINE TB.INPUT                        5344 5345 5347 5348
    PAGE 578  ROUTINE MINE.INPUT                      6625 6627
    PAGE 590  ROUTINE TR.INPUT                        7093 7094
    PAGE 637  FUNCTION HE.WLA                         8934
    PAGE 638  FUNCTION ICM.WLA                        9006 9014
    PAGE 654  PROCESS AIRBORNE.RADAR                  9365
    PAGE 658  PROCESS PHOTO.IR FLIGHT                 9527
    PAGE 659                                          9567
    PAGE 672  ''PROGRAM  OLDER.VERSION                 165  166
    PAGE 675                                           331  332
10.0
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4800
    PAGE 473  PROCESS TARGET.REPORT                   1682
10.1
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION           878
12.
    PAGE 428  PROCESS AIR.OBSERVER                    9386
15000.
    PAGE 205  ROUTINE NOISE.DEGRADE                   9399 9412
15.
    PAGE 165  ROUTINE CFR.DEGRADE                     7611
15.0
    PAGE 246  ROUTINE VOLLEY                          1362
15.24
    PAGE 429  PROCESS AIR.OBSERVER                    9418
16.
    PAGE  97  ROUTINE LINE.OF.SIGHT                   4740 4759
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4784
    PAGE 109  ROUTINE NEW.SEGMENT                     5258
    PAGE 115  ROUTINE PROX.CHECK                      5541
    PAGE 157  ROUTINE AO.DETECTION                    7275 7277
    PAGE 178  ROUTINE EST.COVERAGE                    8152 8156
    PAGE 190  ROUTINE FINAL.COVERAGE                  8702
    PAGE 205  ROUTINE NOISE.DEGRADE                   9399 9412
    PAGE 210  ROUTINE PGM.MSN.ASGN                    9619
    PAGE 230  ROUTINE RPV.DETECTION                    598  600
```

```
    PAGE 246  ROUTINE VOLLEY                          1343
    PAGE 257  ROUTINE FO.DETECTION                    1878 1880
    PAGE 310  ROUTINE AD.SHOOT                        4501
    PAGE 360  EVENT CFR.ON                            6456
    PAGE 362  EVENT CFR.OPERATOR                      6519 6521
    PAGE 386  EVENT PDB.ACTIVATION                    7494
    PAGE 387  EVENT PDB.OPERATOR                      7537 7539
    PAGE 479  PROCESS WITH.DRAW                       1985 2008
    PAGE 523  ROUTINE SYS.INPUT                       4365 4366 4367 4368
    PAGE 527  ROUTINE CAT.TU.INPUT                    4466
    PAGE 533  ROUTINE UNIT.INPUT                      4731 4732
    PAGE 548  ROUTINE TB.INPUT                        5355 5356 5357 5358
    PAGE 556  ROUTINE MUNS.INPUT                      5670
    PAGE 562  ROUTINE MCFR.INPUT                      5899
    PAGE 576  ROUTINE ILLUM.INPUT                     6516
    PAGE 577  ROUTINE MINE.INPUT                      6558 6559 6570 6584 6585
    PAGE 578                                          6626 6628
    PAGE 580  ROUTINE SMOKE.INPUT                     6689
    PAGE 582  ROUTINE TACAIR.INPUT                    6772 6773 6774
    PAGE 583                                          6829 6830 6834 6835 6836 6840 6841 6842 6846 6847 6848
    PAGE 584                                          6852
    PAGE 590  ROUTINE TR.INPUT                        7093 7094
    PAGE 638  FUNCTION ICM.WLA                        8986
    PAGE 656  ROUTINE AR.DETECTION                    9431 9433
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                 9579 9581
16.0
    PAGE 267  ROUTINE BTRY.EFFECTS                    2408 2416
    PAGE 538  ROUTINE MFO.INPUT                       4957
190
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE               4154
1.
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4798
    PAGE 104  ROUTINE MINE.DELAY                      5004
    PAGE 106  ROUTINE MIN.MOVE                        5084
    PAGE 150  ROUTINE PK.COMPUTE                      6983
    PAGE 156  ROUTINE AO.DETECTION                    7234
    PAGE 165  ROUTINE CFR.DEGRADE                     7597
    PAGE 181  ROUTINE FASCAM.COMPUTATION              8270
    PAGE 190  ROUTINE FINAL.COVERAGE                  8711
    PAGE 192                                          8824
    PAGE 193                                          8855
    PAGE 199  ROUTINE ILLUM.COMPUTATION               9168
    PAGE 205  ROUTINE NOISE.DEGRADE                   9406 9425 9425 9425
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION         9811
    PAGE 233  ROUTINE SMOKE.COMPUTATION                725
    PAGE 255  ROUTINE FO.DETECTION                    1754 1754
    PAGE 281  ROUTINE AC.DF.EFFECTS                   3108 3108
    PAGE 284  ROUTINE CAS.EVAL                        3236 3248
    PAGE 309  ROUTINE AD.SHOOT                        4415 4415
    PAGE 310                                          4502
    PAGE 343  ROUTINE SEARCH.COVERAGE                 5688 5690
    PAGE 359  EVENT CFR.OFF                           6394
    PAGE 362  EVENT CFR.OPERATOR                      6519 6521
    PAGE 430  PROCESS AIR.OBSERVER                    9482
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE            1412
    PAGE 484  PROCESS FIRE.MISSION                    2259
```

```
    PAGE 485                                        2321
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9525
    PAGE 661  ROUTINE JOHNSON.CRITERIA              9622
1.0
    PAGE   2  PROGRAM REVISIONS                      105
    PAGE   3                                         173
    PAGE  77  ROUTINE ADJUST                        3902
    PAGE 155  ROUTINE AO.DETECTION                  7149
    PAGE 157                                        7249 7275 7277
    PAGE 165  ROUTINE CFR.DEGRADE                   7613
    PAGE 170  ROUTINE COMBINE.TRS                   7773 7774
    PAGE 179  ROUTINE EST.COVERAGE                  8179 8180 8184 8206 8208 8210 8211
    PAGE 180  ROUTINE EST.MIL.WORTH                 8232
    PAGE 192  ROUTINE FINAL.COVERAGE                8811 8812 8817
    PAGE 193                                        8839 8840 8849 8850 8853 8869
    PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ          9345
    PAGE 204                                        9359 9361 9369 9371
    PAGE 205  ROUTINE NOISE.DEGRADE                 9394
    PAGE 210  ROUTINE PGM.MSN.ASGN                  9622
    PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION       9816
    PAGE 230  ROUTINE RPV.DETECTION                  598  600
    PAGE 231  ROUTINE SIZE.ESTIMATE                  634  654  660
    PAGE 243  ROUTINE UNIT.ENVIR                    1220 1221
    PAGE 244                                        1232 1233 1243 1244 1265 1266 1274 1275
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS              1446
    PAGE 257  ROUTINE FO.DETECTION                  1875 1876
    PAGE 264  ROUTINE BTRY.EFFECTS                  2199
    PAGE 265                                        2281
    PAGE 266                                        2350 2351 2358 2359 2366 2367
    PAGE 267                                        2374 2376 2378 2378 2380 2380 2380 2382 2382 2382 2383 2383 2384 2384
    PAGE 278  ROUTINE AC.BOMB.EFFECTS               2903 2913
    PAGE 323  ROUTINE CONTRAST.TO.FREQ              5012 5013
    PAGE 335  ROUTINE FRAC.COMPUTE                  5421
    PAGE 340  ROUTINE PROB.INF                      5618 5625 5628
    PAGE 341  ROUTINE PROB.TIME                     5656
    PAGE 350  EVENT AD.ENGAGEMENT                   6005 6005
    PAGE 384  EVENT OFF.LINE.ATTRITION              7433
    PAGE 387  EVENT PDB.OPERATOR                    7536 7538
    PAGE 481  PROCESS FIRE.MISSION                  2095
    PAGE 486                                        2403
    PAGE 638  FUNCTION ICM.WLA                      9008
    PAGE 639                                        9016
    PAGE 656  ROUTINE AR.DETECTION                  9431 9433
    PAGE 659  PROCESS PHOTO.IR.FLIGHT               9578 9580
1.00
    PAGE 157  ROUTINE AO.DETECTION                  7251
    PAGE 242  ROUTINE UNIT.ENVIR                    1129 1160
    PAGE 243                                        1216 1223
    PAGE 244                                        1250 1251 1253 1257 1281 1282
    PAGE 245                                        1285
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS              1417
    PAGE 393  EVENT START.ARTY.MOVEMENT             7711
    PAGE 394                                        7715
    PAGE 428  PROCESS AIR.OBSERVER                  9373
1.024
    PAGE 654  PROCESS AIRBORNE.RADAR                9328
```

```
1.1774
    PAGE 157  ROUTINE AO.DETECTION                    7275 7277
    PAGE 230  ROUTINE RPV.DETECTION                    598  600
    PAGE 257  ROUTINE FO.DETECTION                    1878 1880
    PAGE 362  EVENT CFR.OPERATOR                      6519 6521
    PAGE 387  EVENT PDB.OPERATOR                      7537 7539
    PAGE 656  ROUTINE AR.DETECTION                    9431 9433
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                 9579 9581
1.30
    PAGE 242  ROUTINE UNIT.ENVIR                      1155
1.53
    PAGE 165  ROUTINE CFR.DEGRADE                     7611
1.6
    PAGE 254  ROUTINE FO.DETECTION                    1719
    PAGE 404  EVENT START.MOVE                        8270 8271 8271 8271
    PAGE 421  PROCESS AC.ATK.TGT                      8991
1.851
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION         5755
2000.
    PAGE 609  ROUTINE KV.SCOREBOARD                   7875
20.
    PAGE 483  PROCESS FIRE.MISSION                    2257
20.0
    PAGE 340  ROUTINE PROB.INF                        5624
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION         5752
24.
    PAGE 567  ROUTINE SENSOR.INPUT                    6182
24.0
    PAGE 365  EVENT CHANGE.LITE                       6601 6607
    PAGE 523  ROUTINE SYS.INPUT                       4340
24.000
    PAGE   4  PROGRAM REVISIONS                        189
2.
    PAGE  97  ROUTINE LINE.OF.SIGHT                   4740 4759
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4795 4806
    PAGE 109  ROUTINE NEW.SEGMENT                     5258
    PAGE 134  ROUTINE CHECK.FOR.MINES                 6233 6234
    PAGE 155  ROUTINE AO.DETECTION                    7133 7139
    PAGE 156                                          7227 7229 7231 7233 7236 7237 7239
    PAGE 165  ROUTINE CFR.DEGRADE                     7584
    PAGE 166  ROUTINE CFR.DETECTION                   7640 7642 7663 7665 7666 7667 7669
    PAGE 178  ROUTINE EST.COVERAGE                    8140 8163 8165
    PAGE 190  ROUTINE FINAL.COVERAGE                  8690
    PAGE 205  ROUTINE NOISE.DEGRADE                   9425
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                 2856 2858
    PAGE 281  ROUTINE AC.DF.EFFECTS                   3077 3098
    PAGE 308  ROUTINE AD.SHOOT                        4392
    PAGE 343  ROUTINE SEARCH.COVERAGE                 5694 5696
    PAGE 352  EVENT AD.ENGAGEMENT                     6124 6125
    PAGE 360  EVENT CFR.ON                            6458 6462
    PAGE 361                                          6463 6467 6467 6468
    PAGE 371  EVENT FEBA.SORTIE                       6784
    PAGE 429  PROCESS AIR.OBSERVER                    9419
    PAGE 431                                          9507 9515 9516
    PAGE 432                                          9594 9599 9600 9601
    PAGE 433                                          9637
```

```
    PAGE 446  PROCESS HC.ARRIVE.BATTLE                307  321
    PAGE 447                                          335  349
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE           1415
    PAGE 479  PROCESS WITH.DRAW                      1985 2008
    PAGE 482  PROCESS FIRE.MISSION                   2167
    PAGE 507  PROCESS CAS.MISSION                    3532 3534
    PAGE 525  ROUTINE PK.INPUT                       4400
    PAGE 568  ROUTINE SENSOR.INPUT                   6190
    PAGE 584  ROUTINE TACAIR.INPUT                   6867 6869
    PAGE 645  ROUTINE LINE.CIRCLE                    9141 9141 9143 9147 9148
    PAGE 654  PROCESS AIRBORNE.RADAR                 9374
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                9528
2.0
    PAGE 166  ROUTINE CFR.DETECTION                  7654
    PAGE 191  ROUTINE FINAL.COVERAGE                 8722
    PAGE 335  ROUTINE FRAC.COMPUTE                   5421 5426 5426 5429 5430 5431
    PAGE 360  EVENT CFR.ON                           6460
    PAGE 361                                         6470
2.00
    PAGE 244  ROUTINE UNIT.ENVIR                     1231
2.105
    PAGE 205  ROUTINE NOISE.DEGRADE                  9423
2.3263
    PAGE 323  ROUTINE CONTRAST.TO.FREQ               5019
2.5
    PAGE 513  PROCESS HELICOPTER.FIRE                3883
2.6
    PAGE 514  PROCESS HELICOPTER.FIRE                3911
2.7
    PAGE 340  ROUTINE PROB.INF                       5623
2.8668
    PAGE 323  ROUTINE CONTRAST.TO.FREQ               5016
2.96
    PAGE 179  ROUTINE EST.COVERAGE                   8191 8194 8197 8198 8201 8204
    PAGE 192  ROUTINE FINAL.COVERAGE                 8832 8833
    PAGE 193                                         8838 8842 8843 8848
30.0
    PAGE 340  ROUTINE PROB.INF                       5619
30.48
    PAGE 155  ROUTINE AO.DETECTION                   7138 7139
31.25
    PAGE 404  EVENT START.MOVE                       8284
    PAGE 408  EVENT UPDATE.LOC                       8438 8438 8439
    PAGE 410                                         8539
350.
    PAGE 246  ROUTINE VOLLEY                         1343
35.
    PAGE 165  ROUTINE CFR.DEGRADE                    7613
3600.
    PAGE 310  ROUTINE AD.SHOOT                       4501 4502
    PAGE 351  EVENT AD.ENGAGEMENT                    6039 6046
    PAGE 422  PROCESS AC.ATK.TGT                     9044
    PAGE 485  PROCESS FIRE.MISSION                   2350 2366
    PAGE 584  ROUTINE TACAIR.INPUT                   6852
3600.0
    PAGE  51  ''SECTION FOR SUBSTITUTIONS            2897
```

```
    PAGE 709  ''PROGRAM  OLDER.VERSION                2305
360.
    PAGE 166  ROUTINE CFR.DETECTION                   7659
    PAGE 360  EVENT CFR.ON                            6462
3.
    PAGE  51  ''SECTION FOR SUBSTITUTIONS             2900
    PAGE 106  ROUTINE MIN.MOVE                        5090
    PAGE 166  ROUTINE CFR.DETECTION                   7642
    PAGE 185  ROUTINE FA.BN.ASGN                      8503
    PAGE 211  ROUTINE PGM.MSN.ASGN                    9677
    PAGE 269  ROUTINE BTRY.EFFECTS                    2498
    PAGE 425  PROCESS AC.ATK.TGT                      9238
    PAGE 709  ''PROGRAM  OLDER.VERSION                2308
3.0
    PAGE  83  ROUTINE CHANGE.LOC                      4154
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4802
    PAGE 205  ROUTINE NOISE.DEGRADE                   9421
    PAGE 257  ROUTINE FO.DETECTION                    1853
3.00
    PAGE 403  EVENT START.MOVE                        8198
3.4
    PAGE 341  ROUTINE PROB.TIME                       5654
3.5
    PAGE 511  PROCESS HELICOPTER.FIRE                 3766
3.912
    PAGE 323  ROUTINE CONTRAST.TO.FREQ                5010
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION         5747
4.
    PAGE 352  EVENT AD.ENGAGEMENT                     6109
    PAGE 645  ROUTINE LINE.CIRCLE                     9144
4.0
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4802
4.130
    PAGE   2  PROGRAM REVISIONS                         80
4.132
    PAGE   2  PROGRAM REVISIONS                         84
4.134
    PAGE   2  PROGRAM REVISIONS                         87
4.164
    PAGE   2  PROGRAM REVISIONS                         90
4.198
    PAGE   2  PROGRAM REVISIONS                         94
4.273
    PAGE   2  PROGRAM REVISIONS                         98
4.319
    PAGE   2  PROGRAM REVISIONS                        102
4.424
    PAGE   2  PROGRAM REVISIONS                        105  109
4.533
    PAGE   2  PROGRAM REVISIONS                        117
4.546
    PAGE   2  PROGRAM REVISIONS                        113
4.610
    PAGE   3  PROGRAM REVISIONS                        124
4.611
    PAGE   3  PROGRAM REVISIONS                        127
```

```
4.641
    PAGE   3  PROGRAM REVISIONS                    137
4.841
    PAGE   3  PROGRAM REVISIONS                    130
4.875
    PAGE   3  PROGRAM REVISIONS                    133
4.913
    PAGE   3  PROGRAM REVISIONS                    141
5000.
    PAGE 360  EVENT CFR.ON                        6456
500.0
    PAGE 267  ROUTINE BTRY.EFFECTS                2423
5.
    PAGE 104  ROUTINE MINE.DELAY                  5004
    PAGE 155  ROUTINE AO.DETECTION                7151
    PAGE 178  ROUTINE EST.COVERAGE                8142
    PAGE 190  ROUTINE FINAL.COVERAGE              8692
    PAGE 310  ROUTINE AD.SHOOT                    4502
    PAGE 429  PROCESS AIR.OBSERVER                9415
5.0
    PAGE 155  ROUTINE AO.DETECTION                7148
    PAGE 341  ROUTINE PROB.TIME                   5657
5.049
    PAGE   3  PROGRAM REVISIONS                    145
5.166
    PAGE   3  PROGRAM REVISIONS                    149
5.352
    PAGE   3  PROGRAM REVISIONS                    172
5.608
    PAGE   3  PROGRAM REVISIONS                    153
5.65
    PAGE 178  ROUTINE EST.COVERAGE                8171
5.67
    PAGE 178  ROUTINE EST.COVERAGE                8171
600.
    PAGE 482  PROCESS FIRE.MISSION                2150
60.
    PAGE  89  ROUTINE FA.BN.MOVEMENT              4432
    PAGE 211  ROUTINE PGM.MSN.ASGN                9692
    PAGE 310  ROUTINE AD.SHOOT                    4493
    PAGE 421  PROCESS AC.ATK.TGT                  8991
    PAGE 428  PROCESS AIR.OBSERVER                9386 9386
    PAGE 429                                      9410
    PAGE 431                                      9515 9516
    PAGE 432                                      9598 9600 9601
    PAGE 465  PROCESS MINE.ASSESS                 1301
    PAGE 567  ROUTINE SENSOR.INPUT                6182
    PAGE 585  ROUTINE TACAIR.INPUT                6953
60.0
    PAGE  51  ''SECTION FOR SUBSTITUTIONS         2896
    PAGE 709  ''PROGRAM  OLDER.VERSION            2304
62.5
    PAGE  92  ROUTINE FA.BN.MOVEMENT              4571 4573
64.
    PAGE 653  PROCESS AIRBORNE.RADAR              9280 9281 9282
```

```
64.0
    PAGE 340  ROUTINE PROB.INF                        5624
6.
    PAGE 473  PROCESS TARGET.REPORT                   1671
    PAGE 629  FUNCTION BTRY.AVAILABLE                 8663 8667
6.0
    PAGE 473  PROCESS TARGET.REPORT                   1682
6.2
    PAGE 614  ROUTINE SNAP2                           8023
6.363
    PAGE   3  PROGRAM REVISIONS                        161
762.
    PAGE 429  PROCESS AIR.OBSERVER                    9419
78.
    PAGE 542  ROUTINE ORD.DEF                         5122
7.276
    PAGE   3  PROGRAM REVISIONS                        157
7.82
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION         5751
8.0
    PAGE  98  ROUTINE LOCATE.SEARCH.AREA              4791 4791
8.165
    PAGE   3  PROGRAM REVISIONS                        165
8.799
    PAGE   4  PROGRAM REVISIONS                        207
9.
    PAGE 211  ROUTINE PGM.MSN.ASGN                    9691
9.000
    PAGE   3  PROGRAM REVISIONS                        175
9.687
    PAGE   4  PROGRAM REVISIONS                        181
9.810
    PAGE   4  PROGRAM REVISIONS                        185
9.872
    PAGE   4  PROGRAM REVISIONS                        197
9.879
    PAGE   3  PROGRAM REVISIONS                        168
$$$ARTY.ASSESS
    PAGE 437  PROCESS ARTY.ASSESS                     9820

    PAGE  57  ROUTINE MAIN3                           3083
    PAGE  88  ROUTINE END.MOVE                        4390
    PAGE  89  ROUTINE FA.BN.MOVEMENT                  4405
    PAGE  90                                          4471 4490
    PAGE  91                                          4549 4563
    PAGE  92                                          4580
    PAGE 107  ROUTINE NEW.SEGMENT                     5151
    PAGE 108                                          5184
    PAGE 113  ROUTINE PREPARE.LIST                    5444
    PAGE 115  ROUTINE PROX.CHECK                      5548
    PAGE 123  ROUTINE TIME.TO.DETECT                  5849
    PAGE 133  ROUTINE CHECK.FORCE                     6185
    PAGE 142  ROUTINE DEAD.UNIT                       6545
    PAGE 146  ROUTINE INTER.BATTLE                    6774 6795
    PAGE 154  ROUTINE AO.DETECTION                    7113
    PAGE 155                                          7145 7146 7146 7146
```

```
PAGE 156                                    7244 7244 7244 7246 7246
PAGE 157                                    7254
PAGE 159  ROUTINE ATTRIT.SENSOR             7329
PAGE 163  ROUTINE BTRY.FM.DEQ               7534
PAGE 164  ROUTINE BTRY.FM.ENQ               7559
PAGE 179  ROUTINE EST.COVERAGE              8215
PAGE 181  ROUTINE FASCAM.COMPUTATION        8280
PAGE 182  ROUTINE FA.BN.ASGN                8300
PAGE 183                                    8348 8348 8349 8355
PAGE 184                                    8417 8441
PAGE 192  ROUTINE FINAL.COVERAGE            8826 8826 8826 8827 8827 8827 8828 8828 8828 8829 8829
PAGE 193                                    8858 8858 8858 8859 8859 8859 8860 8860 8860 8861 8861 8862 8862 8862 8862
                                            8862 8863 8863 8863 8863 8864 8864 8866
PAGE 199  ROUTINE ILLUM.COMPUTATION         9175
PAGE 200  ROUTINE ILLUM.EFFECTS             9212
PAGE 201                                    9268
PAGE 203  ROUTINE MARGINAL.EFFECTS.ADJ      9316 9316 9341
PAGE 209  ROUTINE PGM.MSN.ASGN              9597
PAGE 210                                    9655
PAGE 211                                    9696
PAGE 214  ROUTINE REM.EFFECTS.COMPUTATION   9806 9823 9823 9823
PAGE 221  ROUTINE REQUEST.ILLUM              156  166
PAGE 225  ROUTINE REQUEST.SMOKE              375  385
PAGE 233  ROUTINE SMOKE.COMPUTATION          732
PAGE 234  ROUTINE SMOKE.EFFECTS              757  776
PAGE 235                                     853
PAGE 236                                     881
PAGE 242  ROUTINE UNIT.ENVIR                1124
PAGE 245                                    1298 1299 1300
PAGE 248  ROUTINE WEIGHTED.VOLLEYS          1419
PAGE 249                                    1451 1452 1452
PAGE 254  ROUTINE FO.DETECTION              1693 1697 1723
PAGE 255                                    1759
PAGE 256                                    1819
PAGE 258                                    1945
PAGE 263  ROUTINE BTRY.EFFECTS              2148
PAGE 265                                    2286 2286 2286
PAGE 268                                    2439 2439
PAGE 269                                    2529
PAGE 276  ROUTINE AC.BOMB.EFFECTS           2796
PAGE 280  ROUTINE AC.DF.EFFECTS             3007
PAGE 281                                    3101
PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS     3290
PAGE 287                                    3396
PAGE 288                                    3467
PAGE 291  ROUTINE EMPLOY.HELICOPTERS        3613
PAGE 292                                    3667
PAGE 295  ROUTINE END.CAS.MISSION           3804 3805 3806 3807 3808 3809
PAGE 307  ROUTINE AD.SHOOT                  4317
PAGE 308                                    4395
PAGE 327  ROUTINE DQ.CMSN.QUEUE             5155
PAGE 334  ROUTINE FINISH.COMPUTATION        5402
PAGE 347  EVENT ACT.REINF                   5819 5829
PAGE 349  EVENT AD.ENGAGEMENT               5911
PAGE 351                                    6049 6049
PAGE 355  EVENT ARTY.OCCUPATION             6240
```

```
    PAGE 356  EVENT BTL.ENDED                          6304
    PAGE 357                                           6312 6320
    PAGE 366  EVENT CHANGE.WEATHER                     6629
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE                6652
    PAGE 368  EVENT END.SIMULATION                     6667 6668 6670
    PAGE 369  EVENT ENGAGEMENT                         6694
    PAGE 371  EVENT FEBA.SORTIE                        6786
    PAGE 373  EVENT GET.NX.ORD                         6904
    PAGE 376  EVENT HELO.ENGAGEMENT                    7041 7055
    PAGE 380  EVENT INIT.PREPLAN.CAS                   7256
    PAGE 382  EVENT OFF.LINE.ATTRITION                 7287 7293 7306 7315 7324 7334
    PAGE 393  EVENT START.ARTY.MOVEMENT                7707
    PAGE 394                                           7725
    PAGE 398  EVENT START.BATTLE                       7942
    PAGE 404  EVENT START.MOVE                         8273
    PAGE 407  EVENT STOP.ARTY.MOVEMENT                 8390
    PAGE 408  EVENT UPDATE.LOC                         8454
    PAGE 412  EVENT ACT.ATK                            8627
    PAGE 419  PROCESS AC.ATK.TGT                       8883
    PAGE 423                                           9084 9116
    PAGE 424                                           9192
    PAGE 425                                           9223
    PAGE 426                                           9267
    PAGE 439  PROCESS FORWARD.OBSERVER                 9954
    PAGE 441                                             59
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                  151
    PAGE 448                                            399
    PAGE 451  PROCESS HC.RETURN.FARRP                   522  522
    PAGE 458  PROCESS HEL.TARGET.ACQUISITION            963
    PAGE 460                                           1055
    PAGE 464  PROCESS MINE.ASSESS                      1241
    PAGE 466                                           1351
    PAGE 470  PROCESS TARGET.REPORT                    1505
    PAGE 472                                           1648
    PAGE 476                                           1878 1878
    PAGE 478  PROCESS WITH.DRAW                        1939
    PAGE 480  PROCESS FIRE.MISSION                     2083
    PAGE 481                                           2089 2099 2106 2137
    PAGE 504  PROCESS CAS.MISSION                      3368
    PAGE 506                                           3476
    PAGE 508                                           3602
    PAGE 509                                           3645
    PAGE 512  PROCESS HELICOPTER.FIRE                  3816 3816
    PAGE 520  ROUTINE MAIN2                            4195
    PAGE 542  ROUTINE ORD.DEF                          5127
    PAGE 543  ROUTINE ORD.ATK                          5148
    PAGE 573  ROUTINE FARRP.INPUT                      6399 6399 6400 6400 6401 6401 6402 6402 6403 6403 6404 6404 6405 6405 6406
                                                       6406 6407 6407 6408 6408
    PAGE 604  ROUTINE ERROR.STOP                       7672
    PAGE 608  ROUTINE KV.PRINT                         7837 7837 7837 7837 7837 7837 7837 7837 7837 7838 7838 7838 7838 7838 7838
                                                       7838
    PAGE 615  ROUTINE SNAP2                            8085
    PAGE 645  ROUTINE LINE.CIRCLE                      9135
.A
    PAGE 173  ROUTINE DUST.EFFECTS                     7895
    PAGE 174                                           7945 7948 7949 7950
```

```
    PAGE 349  EVENT AD.ENGAGEMENT                    5904
    PAGE 352                                         6098 6100 6101 6103 6109 6124 6125
    PAGE 645  ROUTINE LINE.CIRCLE                    9138 9140 9144 9147 9148
.AAT
    PAGE 490  PROCESS ASSESSMENT                     2606 2608 2611 2629 2630 2631 2634
    PAGE 491                                         2648
.AATT
    PAGE 280  ROUTINE AC.DF.EFFECTS                  3018 3020 3021 3023 3034 3035 3045
.AA.LINK
    PAGE 435  PROCESS ARTY.ASSESS                    9711 9712 9716 9717 9720 9722 9732 9734
    PAGE 436                                         9744 9752 9764 9766 9776 9776 9780 9788 9796
    PAGE 437                                         9812 9826 9833 9843 9849
.ABSORPTION.XMIS
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION        5743 5746 5760
.AC
    PAGE 380  EVENT INIT.PREPLAN.CAS                 7232 7248
    PAGE 583  ROUTINE TACAIR.INPUT                   6794 6806 6809 6818 6820 6823 6824 6825 6826 6827 6828 6829 6830 6834 6835
                                                     6836 6840 6841 6842 6846 6847 6848 6851 6851
    PAGE 584                                         6855 6855 6856 6856 6857 6857 6858 6858 6859 6859 6860 6860 6863 6863 6864
                                                     6864 6867 6869 6874 6876 6885 6885 6886 6886 6887 6887 6888 6888 6889 6889
                                                     6890 6890 6891 6891 6892 6892 6893 6893 6894 6894 6899 5901 6903 6909
    PAGE 585                                         6941 6942 6951 6952 6964
    PAGE 586                                         6971
    PAGE 622  ROUTINE TACAIR.DATA.REPORT             8408 8410 8413 8415 8416 8418 8419 8420 8421 8422 8423 8424 8425 8426 8427
                                                     8428 8429 8430 8434 8434 8434 8435 8435 8435 8436 8436 8436
.ACQUIS.TIME
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          708  731  737
    PAGE 460                                         1060
.ACQ.TIME
    PAGE 493  PROCESS SHOOT.OUT                      2765 2778 2780 2783
.AC.ATK.TGT
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2789 2795 2807 2810
    PAGE 280  ROUTINE AC.DF.EFFECTS                  2998 3006 3018 3034 3044 3047 3048
    PAGE 281                                         3064 3065 3071 3106
    PAGE 282                                         3155
    PAGE 419  PROCESS AC.ATK.TGT                     8851 8852 8853 8854 8855 8856 8860 8861 8862 8863 8864 8865 8866 8867 8868
                                                     8869 8870 8871 8872 8873 8874 8875 8876 8877 8878 8879 8882 8884 8897
    PAGE 420                                         8930
    PAGE 422                                         9054
    PAGE 423                                         9083 9115 9136
    PAGE 424                                         9143 9172 9191
    PAGE 425                                         9222 9244
    PAGE 426                                         9266 9285 9295
    PAGE 427                                         9311 9327
    PAGE 490  PROCESS ASSESSMENT                     2607 2608
.AC.DET.TIME
    PAGE 419  PROCESS AC.ATK.TGT                     8860
    PAGE 420                                         8936 8939 8940 8941 8942
    PAGE 422                                         9076
.AC.LINK
    PAGE 295  ROUTINE END.CAS.MISSION                3827 3828 3835
    PAGE 420  PROCESS AC.ATK.TGT                     8949 8950
    PAGE 421                                         8979
    PAGE 504  PROCESS CAS.MISSION                    3374 3375 3382
    PAGE 507                                         3575
```

```
.AC.TYPE
    PAGE 314  ROUTINE FLIGHT.PATH                4637
    PAGE 316                                     4780
    PAGE 317                                     4826 4828
    PAGE 419  PROCESS AC.ATK.TGT                 8857
    PAGE 420                                     8913 8913 8914 8914 8915 8915 8916 8916 8917 8917 8918 8918 8919 8919 8950
    PAGE 421                                     8964 8964 8976 8977 8991 9003 9012
    PAGE 422                                     9028 9029 9030 9031 9077
    PAGE 424                                     9158 9159
    PAGE 426                                     9260
    PAGE 504  PROCESS CAS.MISSION                3355 3375 3397 3398 3402
    PAGE 505                                     3451 3457 3458 3458 3460
    PAGE 506                                     3515 3517 3519 3521 3523
    PAGE 507                                     3525 3526 3528 3548 3554
.AD
    PAGE 162  ROUTINE ATTRIT.SENSOR              7488 7489 7495
    PAGE 316  ROUTINE FLIGHT.PATH                4788 4790 4791 4792 4793 4794 4802
    PAGE 317                                     4823 4830 4841
    PAGE 318                                     4876 4904
    PAGE 645  ROUTINE LINE.CIRCLE                9123
.ADS
    PAGE 308  ROUTINE AD.SHOOT                   4399 4400 4407
    PAGE 350  EVENT AD.ENGAGEMENT                5961 5962 5969 5974 5975 5992 5997 6001 6003 6005
    PAGE 351                                     6039 6046
    PAGE 354                                     6222
.AD.UNIT
    PAGE  65  ROUTINE FILE.KAD.SENSOR            3360 3366 3368 3379 3383 3384 3385 3386
    PAGE 307  ROUTINE AD.SHOOT                   4301 4318 4330
    PAGE 308                                     4399 4400
    PAGE 309                                     4429 4445 4452
    PAGE 310                                     4472 4473 4474 4486
    PAGE 349  EVENT AD.ENGAGEMENT                5917 5918 5919 5931 5936 5936 5945 5946 5947 5948
    PAGE 350                                     5956 5957 5961 5962 6002
    PAGE 351                                     6045 6065 6067
    PAGE 352                                     6072 6076 6088 6099 6100 6101 6104 6105 6106 6107 6107
    PAGE 354                                     6193 6211 6213 6218 6221
    PAGE 419  PROCESS AC.ATK.TGT                 8853 8899 8902 8903
    PAGE 422                                     9032 9053
    PAGE 424                                     9160 9171
    PAGE 506  PROCESS CAS.MISSION                3509 3512 3516 3518 3522
    PAGE 507                                     3524 3527 3529 3560
.AD.UNIT.
    PAGE 419  PROCESS AC.ATK.TGT                 8900
.AFLD
    PAGE 293  ROUTINE END.CAS.MISSION            3683
    PAGE 295                                     3827
.AIRFIELD
    PAGE 582  ROUTINE TACAIR.INPUT               6755 6757
    PAGE 585                                     6957 6959 6961 6963
.ALPHA
    PAGE 293  ROUTINE END.CAS.MISSION            3685
    PAGE 294                                     3764 3766
    PAGE 295                                     3796
    PAGE 577  ROUTINE MINE.INPUT                 6540
    PAGE 578                                     6606 6608 6646
```

```
.ALT
    PAGE 587  ROUTINE MADS.INPUT                    7000 7003
.ANGLE.P1.TO.ADU
    PAGE 504  PROCESS CAS.MISSION                   3362
    PAGE 507                                        3532 3534 3539 3541 3544 3549
.AREA
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2800
    PAGE 277                                        2877 2879 2901
    PAGE 278                                        2911
.AS
    PAGE 425  PROCESS AC.ATK.TGT                    9247 9248 9250 9251 9252
.ASP.STATUS
    PAGE 504  PROCESS CAS.MISSION                   3359
    PAGE 505                                        3415
    PAGE 508                                        3611
.ASSESS
    PAGE  81  ROUTINE BLOCK.LOS                     4080 4081 4082 4083 4084 4086 4087 4088 4089
    PAGE 426  PROCESS AC.ATK.TGT                    9298 9299 9301 9302 9303
    PAGE 490  PROCESS ASSESSMENT                    2637 2638 2640 2641 2642
.ASSESSMENT
    PAGE 487  PROCESS ASSESSMENT                    2423 2427 2454
.ATK
    PAGE  59  ROUTINE CREATE.FORCE                  3147 3148 3150 3151
.ATK.HELICOPTER
    PAGE 376  EVENT HELO.ENGAGEMENT                 7034 7051
    PAGE 377                                        7143
    PAGE 378                                        7151
    PAGE 379                                        7218 7219
    PAGE 510  PROCESS HELICOPTER.FIRE               3660 3680 3702 3703 3704
    PAGE 511                                        3720 3721 3742 3754
    PAGE 512                                        3829 3831
    PAGE 513                                        3873 3881
    PAGE 514                                        3909
    PAGE 515                                        3956 3958 3971 3996
    PAGE 517                                        4075 4091 4092 4109
    PAGE 518                                        4134 4150 4151 4171
.ATK.HELO.UNIT
    PAGE 376  EVENT HELO.ENGAGEMENT                 7048 7052 7075 7076
    PAGE 377                                        7114
    PAGE 378                                        7154 7189 7190 7191 7202
.ATK.TIME
    PAGE 582  ROUTINE TACAIR.INPUT                  6742
    PAGE 585                                        6921 6950 6950 6965 6966 6966
    PAGE 586                                        6973
.ATTACKER
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM            9847 9853 9855 9861 9862 9868 9876
.ATTACK.COUNT
    PAGE 455  PROCESS HEL.TARGET.ACQUISITION         771  779  790  794
.AU
    PAGE 134  ROUTINE CHECK.FOR.MINES               6248 6249
    PAGE 143  ROUTINE DEAD.UNIT                     6608 6609 6611
.B
    PAGE 314  ROUTINE FLIGHT.PATH                   4638 4674
    PAGE 315                                        4688 4689
    PAGE 316                                        4787 4798
    PAGE 317                                        4850
```

```
    PAGE 318                                    4885 4913
    PAGE 349  EVENT AD.ENGAGEMENT               5904
    PAGE 352                                    6104 6109 6124 6125
    PAGE 645  ROUTINE LINE.CIRCLE               9138 9141 9144 9147 9148
.BATTLE
    PAGE 145  ROUTINE FIN.BATTLE                6693 6699 6700 6702
    PAGE 283  ROUTINE CAS.EVAL                  3169 3182 3220
    PAGE 452  PROCESS HC.RETURN.FARRP            610  611  615  620
.BATTLE.ENDED
    PAGE 464  PROCESS MINE.ASSESS               1220
.BATTLE.STATUS
    PAGE 395  EVENT START.BATTLE                7755 7782 7784 7786
.BLIND
    PAGE 419  PROCESS AC.ATK.TGT                8861
    PAGE 421                                    8963 8966 8987 8996 8998 9009
    PAGE 422                                    9022
    PAGE 423                                    9132
    PAGE 424                                    9153 9154
.BLUE.CEQ
    PAGE 283  ROUTINE CAS.EVAL                  3209
    PAGE 284                                    3253
.BLUE.FORCE
    PAGE 283  ROUTINE CAS.EVAL                  3211
    PAGE 284                                    3229
.BLUE.TGT
    PAGE 283  ROUTINE CAS.EVAL                  3210
    PAGE 284                                    3254
.BLUE.TKS
    PAGE 283  ROUTINE CAS.EVAL                  3208
    PAGE 284                                    3233 3239 3244 3251
.BLUE.UNITS
    PAGE 145  ROUTINE FIN.BATTLE                6697 6699 6736
.BN
    PAGE 258  ROUTINE FO.DETECTION              1907 1909
    PAGE 362  EVENT CFR.OPERATOR                6532 6534
.BTL
    PAGE 311  ROUTINE INTER.HELO                4514 4525 4526 4527 4528 4529 4530 4531 4532 4537 4539
.BTRY
    PAGE   3  PROGRAM REVISIONS                  131
    PAGE  89  ROUTINE FA.BN.MOVEMENT            4396 4401 4403 4422 4424 4429 4429 4433 4436
    PAGE  90                                    4455 4456 4457 4460 4501 4502 4505
    PAGE  91                                    4516 4535 4536 4538 4539 4539 4541 4542 4543 4543 4544 4545 4548 4548 4552
                                                4555 4556 4557 4557 4558 4559 4562 4562 4566
    PAGE  92                                    4569 4570 4575 4576 4579 4579 4583
    PAGE 258  ROUTINE FO.DETECTION              1909 1910
    PAGE 355  EVENT ARTY.OCCUPATION             6228 6233 6234 6239 6239 6243 6244 6248 6249 6250
    PAGE 362  EVENT CFR.OPERATOR                6534
    PAGE 363                                    6535
    PAGE 480  PROCESS FIRE.MISSION              2043 2057 2058 2062 2068 2072 2080 2082
    PAGE 481                                    2086 2088 2095 2096 2098 2110 2117 2121 2128 2129 2135 2135
    PAGE 482                                    2144 2149 2150
    PAGE 483                                    2217 2229 2243 2244 2248 2249 2254
    PAGE 484                                    2284 2287 2290
    PAGE 485                                    2330 2332 2335
    PAGE 486                                    2386 2398 2399
    PAGE 611  ROUTINE OUTPUT.ATTRITION          7970 7983
```

```
.C
    PAGE 173  ROUTINE DUST.EFFECTS                  7895
    PAGE 174                                        7948 7949
    PAGE 349  EVENT AD.ENGAGEMENT                   5904
    PAGE 352                                        6106 6109 6109 6110 6112 6120 6123 6123 6124 6125
    PAGE 645  ROUTINE LINE.CIRCLE                   9138 9142 9144
.CALLING.PROCESS
    PAGE 307  ROUTINE AD.SHOOT                      4302 4318 4321 4322 4323 4325 4326 4327 4341 4344 4348
    PAGE 309                                        4418 4456 4457 4458 4461 4462 4463 4464
    PAGE 349  EVENT AD.ENGAGEMENT                   5900 5913 5915
    PAGE 350                                        6009
    PAGE 351                                        6019
    PAGE 354                                        6194
.CAS
    PAGE  58  ROUTINE CREATE.FORCE                  3117 3118 3125 3126 3127
    PAGE 284  ROUTINE CAS.EVAL                      3259 3261 3262 3263 3264 3266 3268 3269 3270 3274
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS         3282 3287 3289 3291 3296 3298 3305 3307 3311 3315 3318 3321
    PAGE 286                                        3343 3374 3377 3378 3378 3380 3384 3386 3391 3392 3393
    PAGE 287                                        3398 3403 3405 3408 3410 3411 3421 3429 3441 3443 3450
    PAGE 288                                        3455 3457 3458 3460 3462 3463 3465 3466
    PAGE 293  ROUTINE END.CAS.MISSION               3705 3706 3706 3707 3708
    PAGE 327  ROUTINE DQ.CMSN.QUEUE                 5158 5161
    PAGE 380  EVENT INIT.PREPLAN.CAS                7238 7245 7246 7247 7248 7249 7250 7251 7252 7263
.CASUALTIES
    PAGE 251  ROUTINE MINE.EFFECTS                  1561 1563 1563 1569 1572 1573
    PAGE 252                                        1576 1579 1582 1583 1593 1605 1612 1616
    PAGE 253                                        1638 1646
    PAGE 464  PROCESS MINE.ASSESS                   1249 1252 1252 1253 1256
    PAGE 465                                        1276 1284 1308 1324
.CAS.MISSION
    PAGE 314  ROUTINE FLIGHT.PATH                   4632 4636 4637 4649 4661 4662 4663 4664 4666 4667 4668 4678
    PAGE 315                                        4719 4745
    PAGE 316                                        4749 4759 4771 4774 4781
    PAGE 419  PROCESS AC.ATK.TGT                    8852 8857 8858 8885 8885 8886
    PAGE 420                                        8949
    PAGE 421                                        8977 8983
    PAGE 422                                        9039 9040 9041 9042
    PAGE 423                                        9085 9085 9117 9117
    PAGE 424                                        9161 9193 9193
    PAGE 425                                        9224 9224
    PAGE 426                                        9256 9268 9268 9286 9289 9307
    PAGE 427                                        9318 9323 9324
    PAGE 504  PROCESS CAS.MISSION                   3353 3354 3355 3356 3357 3358 3359 3360 3361 3367 3369 3372 3384 3407
    PAGE 505                                        3409 3410 3418 3421 3423 3442 3448 3463 3465
    PAGE 506                                        3466 3477 3483 3485 3486 3514 3520
    PAGE 507                                        3548 3559 3577
    PAGE 508                                        3595 3603 3617 3619 3620 3627 3634
    PAGE 509                                        3646 3650 3653 3654
.CAT
    PAGE 181  ROUTINE FASCAM.COMPUTATION            8256 8268
    PAGE 199  ROUTINE ILLUM.COMPUTATION             9147 9165 9173
    PAGE 233  ROUTINE SMOKE.COMPUTATION              702  704  722  730
.CEQ
    PAGE 284  ROUTINE CAS.EVAL                      3241 3253 3256
.CE.COUNT
    PAGE 283  ROUTINE CAS.EVAL                      3188 3198 3209 3214
```

```
.CHOSEN BTRY
    PAGE 209  ROUTINE PGM.MSN.ASGN                  9563 9568 9573 9581
.CLOSEST.FO
    PAGE 103  ROUTINE MINE.DELAY                    4968 4980 4983 4985
.CMSN
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2810 2811 2812 2813 2837 2840
    PAGE 293  ROUTINE END.CAS.MISSION               3678 3683 3684 3689 3696 3701 3720 3730
    PAGE 294                                        3762 3763 3769 3769
    PAGE 295                                        3794 3794 3795 3796 3797 3798 3799 3799 3800 3801 3801 3815 3815 3816 3816
                                                    3828
.COL
    PAGE 315  ROUTINE FLIGHT.PATH                   4692 4693 4695 4696 4697 4698 4700 4701 4702 4703 4705 4710 4713 4715 4720
                                                    4727
    PAGE 316                                        4746 4750 4753 4754
.COLUMN
    PAGE 277  ROUTINE AC.BOMB.EFFECTS               2886
    PAGE 279                                        2980
    PAGE 282  ROUTINE AC.DF.EFFECTS                 3156 3159 3160
.CONCEAL
    PAGE 223  ROUTINE REQUEST.SMOKE                  273  276  279
.COPY.FROM
    PAGE 172  ROUTINE COPY                          7845 7857 7858 7859 7860 7861 7862 7863 7864 7865 7866 7867 7868 7869 7870
                                                    7871 7872 7873 7874 7875 7876 7877 7878
.COPY.TO
    PAGE 172  ROUTINE COPY                          7846 7857 7858 7859 7860 7861 7862 7863 7864 7865 7866 7867 7868 7869 7870
                                                    7871 7872 7873 7874 7875 7876 7877 7878
.COUNT
    PAGE 123  ROUTINE TIME.TO.DETECT                5828 5837
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS         3447 3450
    PAGE 370  EVENT ENGAGEMENT                      6737 6742 6751
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION         921
    PAGE 458                                         929
    PAGE 459                                        1027 1036
    PAGE 622  ROUTINE TACAIR.DATA.REPORT            8447 8448
.COUNTER
    PAGE 404  EVENT START.MOVE                      8292
    PAGE 405                                        8294 8295
    PAGE 410  EVENT UPDATE.LOC                      8547 8549 8550
.COVER
    PAGE 236  ROUTINE SMOKE.EFFECTS                  860  861  865  873
.COVERAGE
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2799
    PAGE 277                                        2853 2861 2864 2869 2872 2901
    PAGE 278                                        2911
.CS
    PAGE 314  ROUTINE FLIGHT.PATH                   4677 4678 4679 4680
    PAGE 315                                        4718 4719 4720 4721 4730 4744 4745
    PAGE 316                                        4746 4747 4748 4749 4750 4751 4759 4761 4762 4763 4765 4765 4766 4766 4768
                                                    4769 4769 4770 4770 4773 4774
.CTNAME
    PAGE 533  ROUTINE UNIT.INPUT                    4684 4736
    PAGE 535                                        4810 4811 4836
.CUM1
    PAGE 511  PROCESS HELICOPTER.FIRE               3725
    PAGE 512                                        3801 3801 3803 3814 3817 3822
    PAGE 513                                        3846
```

```
.CUM2
    PAGE 511  PROCESS HELICOPTER.FIRE                3726
    PAGE 512                                         3802 3802 3804 3814 3817 3821
    PAGE 513                                         3857
.DEFENDER
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM             9846 9852 9855 9859 9859 9867 9875
.DELAY
    PAGE 103  ROUTINE MINE.DELAY                     4940
    PAGE 104                                         5004 5006 5006 5010 5023 5037
    PAGE 105                                         5051
    PAGE 218  ROUTINE REQUEST.ILLUM                  9985
    PAGE 219                                           89
    PAGE 220                                          138
    PAGE 222  ROUTINE REQUEST.SMOKE                   176
    PAGE 223                                          287
    PAGE 225                                          356  374
    PAGE 250  ROUTINE MINE.EFFECTS                   1469
    PAGE 251                                         1526 1550
    PAGE 253                                         1665
    PAGE 393  EVENT START.ARTY.MOVEMENT              7698 7699
    PAGE 405  EVENT START.MOVE                       8307 8308
    PAGE 410  EVENT UPDATE.LOC                       8562 8563
    PAGE 484  PROCESS FIRE.MISSION                   2279
.DELTA
    PAGE 280  ROUTINE AC.DF.EFFECTS                  3010
    PAGE 281                                         3070 3072 3075 3076 3098
    PAGE 307  ROUTINE AD.SHOOT                       4311
    PAGE 308                                         4365 4366 4369 4371 4392
    PAGE 530  ROUTINE EQ.TE.INPUT                    4577 4588 4590
.DELTAX
    PAGE 103  ROUTINE MINE.DELAY                     4975 4977
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2843
    PAGE 277                                         2845
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS          3441 3445
    PAGE 349  EVENT AD.ENGAGEMENT                    5927 5929 5933
    PAGE 351                                         6061
    PAGE 352                                         6086
    PAGE 404  EVENT START.MOVE                       8282 8284
    PAGE 410  EVENT UPDATE.LOC                       8536 8538
    PAGE 507  PROCESS CAS.MISSION                    3526 3530 3538 3545
    PAGE 584  ROUTINE TACAIR.INPUT                   6863 6865 6873 6880
.DELTAY
    PAGE 103  ROUTINE MINE.DELAY                     4976 4977
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2844
    PAGE 277                                         2845
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS          3443 3445
    PAGE 349  EVENT AD.ENGAGEMENT                    5928 5933
    PAGE 351                                         6062
    PAGE 404  EVENT START.MOVE                       8283 8284
    PAGE 410  EVENT UPDATE.LOC                       8537 8538
    PAGE 507  PROCESS CAS.MISSION                    3528 3531 3537 3544
    PAGE 584  ROUTINE TACAIR.INPUT                   6864 6866 6872 6879
.DELTA.CONTRAST
    PAGE 323  ROUTINE CONTRAST.TO.FREQ               5007 5012 5015 5016 5018 5019 5025 5026
.DELTA.T.PRIME
    PAGE 120  ROUTINE SEARCH                         5740 5744
```

```
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION          5738 5742 5761
    PAGE 646  ROUTINE MRT.TO.FREQ                      9157 9164 9166 9171 9174 9175
.DETECTED.TE
    PAGE 120  ROUTINE SEARCH                           5710 5717 5737 5759
    PAGE 340  ROUTINE PROB.INF                         5606 5622
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION          5734 5761
.DETECTED.UNIT
    PAGE 123  ROUTINE TIME.TO.DETECT                   5797 5805 5811 5816 5847
    PAGE 124                                           5854 5861 5867
    PAGE 369  EVENT ENGAGEMENT                         6681 6689 6691 6707 6718 6722
    PAGE 370                                           6745 6758
.DETECTING.UNIT
    PAGE 123  ROUTINE TIME.TO.DETECT                   5796 5806 5811 5815 5847
    PAGE 124                                           5853 5860 5866
    PAGE 369  EVENT ENGAGEMENT                         6680 6690 6700 6706 6714 6735
    PAGE 370                                           6754 6757 6762 6763 6764 6768 6769 6770
.DETECTING.WPN
    PAGE 120  ROUTINE SEARCH                           5707 5717 5726
    PAGE 121                                           5765
    PAGE 341  ROUTINE PROB.TIME                        5636 5652 5652 5653 5653
.DETECT.TIME
    PAGE 123  ROUTINE TIME.TO.DETECT                   5801 5839 5841
    PAGE 419  PROCESS AC.ATK.TGT                       8862
    PAGE 422                                           9075 9078
    PAGE 423                                           9080 9099 9101 9131
    PAGE 424                                           9184 9188
    PAGE 425                                           9207
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION            709
    PAGE 458                                            931  933
    PAGE 459                                           1038
    PAGE 460                                           1040
.DET.TIME
    PAGE 349  EVENT AD.ENGAGEMENT                      5905
    PAGE 351                                           6042 6043 6043 6044 6044 6046 6048
.DFFB
    PAGE 181  ROUTINE FASCAM.COMPUTATION               8255 8268
    PAGE 199  ROUTINE ILLUM.COMPUTATION                9146 9165 9173
    PAGE 233  ROUTINE SMOKE.COMPUTATION                 700  722  730
.DIS
    PAGE 103  ROUTINE MINE.DELAY                       4977 4978 4979
    PAGE 174  ROUTINE DUST.EFFECTS                     7949 7952 7954
    PAGE 410  EVENT UPDATE.LOC                         8538 8539
DIST
    PAGE 353  EVENT AD.ENGAGEMENT                      6160 6163
DISTANCE
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS            3445 3446
    PAGE 584  ROUTINE TACAIR.INPUT                     6885 6894
    PAGE 585                                           6961 6964
DIST.TO.FP
    PAGE 419  PROCESS AC.ATK.TGT                       8863
    PAGE 422                                           9030 9031 9039 9040
    PAGE 424                                           9157
    PAGE P3
    PAGE 419  PROCESS AC.ATK.TGT                       8864
    PAGE 424                                           9157 9158
```

```
.DUMMY.WPN
    PAGE 510  PROCESS HELICOPTER.FIRE                3668
    PAGE 512                                         3790
.DUPLICATE
    PAGE 219  ROUTINE REQUEST.ILLUM                    88
    PAGE 220                                           98  103  107  119  125  129  135
    PAGE 223  ROUTINE REQUEST.SMOKE                   286
    PAGE 224                                          296  301  305  310  314  327  333  337  342
    PAGE 225                                          346  353
.DURATION
    PAGE  80  ROUTINE BLOCK.LOS                      4009 4030 4045
    PAGE 173  ROUTINE DUST.EFFECTS                   7888 7897
    PAGE 174                                         7959 7961
    PAGE 235  ROUTINE SMOKE.EFFECTS                   811  847  852
    PAGE 236                                          875  880
.DUST.RAD
    PAGE 555  ROUTINE MUNS.INPUT                     5646 5648
.E
    PAGE 340  ROUTINE PROB.INF                       5613 5623 5624 5627 5628
.EN
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                2895 2900
    PAGE 278                                         2903 2910 2913
.ENEMY
    PAGE  58  ROUTINE CREATE.FORCE                   3099 3101 3117
    PAGE  80  ROUTINE BLOCK.LOS                      4008 4018 4041 4045 4057 4062
    PAGE  81                                         4066 4072 4073 4082 4083 4107
    PAGE 173  ROUTINE DUST.EFFECTS                   7938 7939
    PAGE 174                                         7940 7942 7945 7946 7952 7958 7961 7968 7969 7970 7971 7974 7983
    PAGE 201  ROUTINE ILLUM.EFFECTS                  9274 9275 9278 9285
    PAGE 218  ROUTINE REQUEST.ILLUM                  9992   19   22   27   31
    PAGE 219                                           40   42   44
    PAGE 222  ROUTINE REQUEST.SMOKE                   207  210  215  220
    PAGE 226  ROUTINE REQUEST.WD.FASCAM               422  426  427  428  429  430  435  440
    PAGE 227                                          481  486  491  500
    PAGE 235  ROUTINE SMOKE.EFFECTS                   840  841  846  852
    PAGE 236                                          869  874  879
    PAGE 283  ROUTINE CAS.EVAL                       3223 3224
    PAGE 314  ROUTINE FLIGHT.PATH                    4649 4683
    PAGE 316                                         4788
.ENEMY.FORCE
    PAGE 173  ROUTINE DUST.EFFECTS                   7929 7930 7938
    PAGE 174                                         7968
    PAGE 200  ROUTINE ILLUM.EFFECTS                  9229 9231
    PAGE 201                                         9239 9247 9254
    PAGE 218  ROUTINE REQUEST.ILLUM                  9992    5    7   22
    PAGE 219                                           42
    PAGE 222  ROUTINE REQUEST.SMOKE                   194  195  210
    PAGE 223                                          276
    PAGE 226  ROUTINE REQUEST.WD.FASCAM               412  413  426
    PAGE 227                                          481  494  494  496
    PAGE 235  ROUTINE SMOKE.EFFECTS                   819  821  840
    PAGE 236                                          866
.ENEMY.UNIT
    PAGE 376  EVENT HELO.ENGAGEMENT                  7035 7053 7060 7069 7075
    PAGE 377                                         7089 7093 7097 7100 7127 7129 7140 7144
    PAGE 378                                         7150 7151 7170 7176
```

```
    PAGE 379                                        7215 7219
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION        1055
.ENEMY.UNITS
    PAGE 300  ROUTINE HC.DISENGAGE                  4007 4015 4017 4020 4025 4035 4039 4057
    PAGE 301                                        4083 4086
    PAGE 311  ROUTINE INTER.HELO                    4521 4555 4556 4559 4564
    PAGE 312                                        4575 4579 4598
    PAGE 375  EVENT HC.DEPART.BATTLE                6992 7001 7015
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION         695  707  712
    PAGE 455                                         807
    PAGE 456                                         809  819  820  829  836  842  852
    PAGE 459                                         991 1002 1017
    PAGE 460                                        1048 1052
    PAGE 461                                        1100
    PAGE 462                                        1158 1162 1163
.ENG
    PAGE  81  ROUTINE BLOCK.LOS                     4070 4071 4072 4073 4074 4076 4077
.ENVIR.FRAC
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2802 2805 2826 2827 2828
    PAGE 278                                        2903 2913
    PAGE 279                                        2992
.EQ
    PAGE 227  ROUTINE REQUEST.WD.FASCAM              456  457  458  461
    PAGE 251  ROUTINE MINE.EFFECTS                  1556 1557 1559 1560 1563 1568 1572 1573
    PAGE 252                                        1575 1584 1593 1594
    PAGE 253                                        1645 1647 1653
    PAGE 352  EVENT AD.ENGAGEMENT                   6072 6073 6081
    PAGE 465  PROCESS MINE.ASSESS                   1329 1330
    PAGE 466                                        1333 1338
.EQUIP
    PAGE 382  EVENT OFF.LINE.ATTRITION              7319 7320
    PAGE 383                                        7359 7366 7367 7369 7370 7372 7374
    PAGE 384                                        7399 7400 7403 7425
    PAGE 534  ROUTINE UNIT.INPUT                    4783 4784 4787
.EQ.NAME
    PAGE 621  ROUTINE TACAIR.DATA.REPORT            8343
    PAGE 622                                        8411 8413 8417
.ERROR
    PAGE  66  ROUTINE FORM.TF.LIST                  3416 3428 3436 3443
    PAGE 550  ROUTINE BTRY.INPUT                    5396
    PAGE 551                                        5467 5478
    PAGE 552                                        5530
.EXTINCTION.COEF
    PAGE 323  ROUTINE CONTRAST.TO.FREQ              5007 5010 5013
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION       5742 5749 5752 5754 5759
.FARRP
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE             4146 4160 4171
    PAGE 311  ROUTINE INTER.HELO                    4537 4539 4541 4547 4559
    PAGE 312                                        4592
    PAGE 391  EVENT SEND.TEAM                       7598 7610 7617
    PAGE 401  EVENT START.BATTLE                    8144 8146
    PAGE 402                                        8149 8151 8154 8155 8160 8167
.FASCAM
    PAGE 181  ROUTINE FASCAM.COMPUTATION            8263 8264 8279
.FA.BN
    PAGE  89  ROUTINE FA.BN.MOVEMENT                4396 4401 4403 4409 4410 4422
```

```
    PAGE 390  EVENT SCHEDULE.ARTY.MOVEMENT          7587 7591
.FDC
    PAGE 216  ROUTINE REQUEST.FASCAM                9929
    PAGE 217                                        9943 9953 9961
    PAGE 219  ROUTINE REQUEST.ILLUM                   59   73   83
    PAGE 220                                        141
    PAGE 223  ROUTINE REQUEST.SMOKE                 232  246  256
    PAGE 225                                        359
    PAGE 257  ROUTINE FO.DETECTION                  1902
    PAGE 258                                        1903 1905 1907 1915 1916 1921 1921
    PAGE 362  EVENT CFR.OPERATOR                    6527 6528 6530 6532
    PAGE 363                                        6538 6539 6545 6545
.FIRE
    PAGE 280  ROUTINE AC.DF.EFFECTS                 3013
.FIRER
    PAGE 383  EVENT OFF.LINE.ATTRITION              7355 7357
    PAGE 385                                        7460
.FIRER.EQ
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2811 2818
    PAGE 279                                        2979
    PAGE 280  ROUTINE AC.DF.EFFECTS                 3047 3053
    PAGE 282                                        3154
.FIRER.EQUIP
    PAGE 369  EVENT ENGAGEMENT                      6714 6715 6716 6724 6730 6731 6734
    PAGE 370                                        6736 6742 6743 6744
.FIRE.TAB
    PAGE 511  PROCESS HELICOPTER.FIRE               3727 3729 3731 3735 3736 3755 3761 3764
    PAGE 512                                        3775 3778 3783 3786 3791 3792 3793 3796 3803 3804
.FIRING.EQ
    PAGE 307  ROUTINE AD.SHOOT                      4337
    PAGE 309                                        4423 4424 4439 4440 4453
    PAGE 310                                        4472
.FIRING.EQUIP
    PAGE 378  EVENT HELO.ENGAGEMENT                 7176 7177 7178 7183 7199
    PAGE 379                                        7210 7211 7214 7216
.FIRING.HELO
    PAGE 376  EVENT HELO.ENGAGEMENT                 7076 7079
    PAGE 377                                        7095 7113 7115 7117
    PAGE 378                                        7154 7155 7161
.FIRING.RANGE
    PAGE 419  PROCESS AC.ATK.TGT                    8865
    PAGE 422                                        9025 9028 9028 9029 9029 9030 9077
.FIRING.TABLE
    PAGE 300  ROUTINE HC.DISENGAGE                  4056 4057 4059 4060
    PAGE 312  ROUTINE INTER.HELO                    4596 4599 4601 4603
.FIRING.TIME
    PAGE 307  ROUTINE AD.SHOOT                      4300 4311 4318
    PAGE 310                                        4491
    PAGE 349  EVENT AD.ENGAGEMENT                   5905
    PAGE 351                                        6045 6051 6064 6066
    PAGE 353                                        6164 6166 6168 6169 6182 6185
    PAGE 354                                        6187 6192 6198 6210 6212 6218
    PAGE 419  PROCESS AC.ATK.TGT                    8866
    PAGE 422                                        9023 9044 9047 9052 9058 9064 9065 9067
    PAGE 424                                        9165 9170 9176
```

```
FIRING.TIME.
    PAGE 351  EVENT AD.ENGAGEMENT                    6059
.FLAG
    PAGE 173  ROUTINE DUST.EFFECTS                   7937
    PAGE 174                                         7960 7966
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS          3418 3422 3427 3431 3435 3451
    PAGE 288                                         3455 3456 3466
    PAGE 293  ROUTINE END.CAS.MISSION                3713 3715 3728 3732
    PAGE 465  PROCESS MINE.ASSESS                    1274 1278 1280 1286
.FLIGHT.TIME
    PAGE 504  PROCESS CAS.MISSION                    3361
    PAGE 505                                         3425 3451
    PAGE 506                                         3472
    PAGE 509                                         3641
.FLY
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          706
.FM
    PAGE 216  ROUTINE REQUEST.FASCAM                 9915 9916 9917
    PAGE 218  ROUTINE REQUEST.ILLUM                  9992
    PAGE 220                                          113  114  115  117  121
    PAGE 224  ROUTINE REQUEST.SMOKE                   321  322  323  325  329
    PAGE 270  ROUTINE BTRY.EFFECTS                   2590 2592 2595 2595
    PAGE 476  PROCESS TARGET.REPORT                  1879 1881 1881 1881 1881 1882 1882
    PAGE 577  ROUTINE MINE.INPUT                     6565 6567 6570
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                7373 7375 7378 7378
.FM.N.TUBES
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                7244 7250
    PAGE 596                                         7347
    PAGE 597                                         7378 7397
.FO
    PAGE 103  ROUTINE MINE.DELAY                     4971 4972 4974
.FORCE
    PAGE 145  ROUTINE FIN.BATTLE                     6702 6704 6705 6709 6722 6723 6723 6724 6726
    PAGE 173  ROUTINE DUST.EFFECTS                   7902 7904 7923 7924 7930
    PAGE 200  ROUTINE ILLUM.EFFECTS                  9223 9224 9231
    PAGE 201                                         9240 9246 9255
    PAGE 216  ROUTINE REQUEST.FASCAM                 9931 9933 9938
    PAGE 218  ROUTINE REQUEST.ILLUM                  9992 9999    0    7
    PAGE 219                                           61   63   68
    PAGE 222  ROUTINE REQUEST.SMOKE                   188  189  195
    PAGE 223                                          234  236  241  273
    PAGE 226  ROUTINE REQUEST.WD.FASCAM               406  407  413
    PAGE 234  ROUTINE SMOKE.EFFECTS                   789  792
    PAGE 235                                          813  814  821  837
    PAGE 236                                          862  889  892
    PAGE 283  ROUTINE CAS.EVAL                       3182 3184 3184 3189 3207 3211 3216 3220 3221 3223
    PAGE 400  EVENT START.BATTLE                     8051 8053
    PAGE 444  PROCESS HC.ARRIVE.BATTLE                166  168  170  200  201  203
    PAGE 489  PROCESS ASSESSMENT                     2533 2535 2540 2563
.FO.UNIT
    PAGE 103  ROUTINE MINE.DELAY                     4974 4975 4976 4980 4983 4988 4989
    PAGE 104                                         4994
.FP
    PAGE  65  ROUTINE FILE.KAD.SENSOR                3382 3383 3384 3385 3386 3389 3392 3392 3393 3393 3394 3395 3396 3396 3397
                                                     3397 3398 3399 3402 3405 3405 3406 3406 3407 3407 3408 3408
```

```
.FPO
    PAGE 314  ROUTINE FLIGHT.PATH                    4683 4686
    PAGE 315                                         4688 4689 4690 4690 4691 4691 4693 4695 4697 4700 4702 4715 4716 4716 4724
                                                     4725 4731 4732 4733 4735 4738 4740
.FRAC.CAS
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2801
    PAGE 277                                         2892 2902 2902
    PAGE 278                                         2912 2912 2921 2923 2930 2935 2958
    PAGE 279                                         2977
.FRAC.CASUALTIES
    PAGE 250  ROUTINE MINE.EFFECTS                   1476
    PAGE 251                                         1572
    PAGE 252                                         1585 1595 1618
    PAGE 253                                         1658
    PAGE 464  PROCESS MINE.ASSESS                    1227 1256
    PAGE 465                                         1287
.FRAC.TARGET.VISIBLE
    PAGE 120  ROUTINE SEARCH                         5709 5758
    PAGE 340  ROUTINE PROB.INF                       5605 5614 5621
.FREQUENCY
    PAGE 120  ROUTINE SEARCH                         5732 5746 5756
    PAGE 323  ROUTINE CONTRAST.TO.FREQ               5003 5007 5016 5019 5021 5026 5028
    PAGE 340  ROUTINE PROB.INF                       5603 5613 5621
    PAGE 646  ROUTINE MRT.TO.FREQ                    9160 9164 9172 9174
.FT
    PAGE  81  ROUTINE BLOCK.LOS                      4095 4098 4102 4103 4109 4112 4116 4117
    PAGE 293  ROUTINE END.CAS.MISSION                3719 3720 3722 3723
    PAGE 419  PROCESS AC.ATK.TGT                     8892 8893
    PAGE 423                                         9091 9092 9123 9124
    PAGE 425                                         9199 9200 9230 9231
    PAGE 426                                         9274 9275
    PAGE 461  PROCESS HEL.TARGET.ACQUISITION         1102 1106 1108 1109 1116 1118 1119
    PAGE 507  PROCESS CAS.MISSION                    3572 3573 3574 3575 3576 3577
.F.LINK
    PAGE 483  PROCESS FIRE.MISSION                   2229 2230 2231 2234 2241 2250
    PAGE 484                                         2264
.GAP.XMIS
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION        5743 5759 5760
.GROUND.UNIT
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4122 4135 4136
    PAGE 303                                         4181 4183
.GROUPING
    PAGE 252  ROUTINE MINE.EFFECTS                   1604 1605
    PAGE 533  ROUTINE UNIT.INPUT                     4735 4737
    PAGE 534                                         4750 4756
    PAGE 535                                         4806 4810
.HAB
    PAGE 402  EVENT START.BATTLE                     8159 8160 8161
.HAND.OFF.TIME
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          708  728  734
    PAGE 458                                          957
.HC
    PAGE 311  ROUTINE INTER.HELO                     4565 4567 4569
    PAGE 312                                         4576
    PAGE 376  EVENT HELO.ENGAGEMENT                  7066 7068 7078 7079 7085
    PAGE 377                                         7109 7110 7112 7115 7121 7122 7124 7133 7139
```

```
    PAGE 378                                    7149 7155 7167 7168 7171
    PAGE 460  PROCESS HEL.TARGET.ACQUISITION    1064 1066
    PAGE 461                                    1151 1152
    PAGE 488  PROCESS ASSESSMENT                2489 2490 2493 2494 2498 2499
    PAGE 489                                    2558 2559 2560 2560
    PAGE 498  PROCESS SHOOT.OUT                 3039 3043
.HC.TEAM
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION     894  713  714  717  718  718  722  727  741  746
    PAGE 455                                     755  758  772  788  789  789  800
    PAGE 456                                     861
    PAGE 457                                     870  871  875  877  891  892  893  899
    PAGE 458                                     940  944  952  953  959  961  970  971  976  978
    PAGE 459                                     993 1008 1009 1010
    PAGE 460                                    1045 1046 1062 1064 1070 1082 1088 1094
    PAGE 461                                    1104 1114 1122 1135 1142 1146 1151
    PAGE 462                                    1156 1157 1161 1161 1163
.HEIGHT
    PAGE 530  ROUTINE EQ.TE.INPUT               4577 4589 4591
.HELICOPTER
    PAGE 300  ROUTINE HC.DISENGAGE              4026 4028 4030 4036
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE         4121 4133 4134 4146 4147 4148 4152 4158
    PAGE 303                                    4181 4183
    PAGE 378  EVENT HELO.ENGAGEMENT             7180 7181 7185 7192
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION     746  748  749
    PAGE 455                                     750  758  759  760  764  772  774  776  777  780  781  782
    PAGE 457                                     898  900  908
    PAGE 458                                     943  945  947  954  960
    PAGE 460                                    1055
    PAGE 461                                    1104 1105 1122 1124 1125 1128
.HELO
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE         4148 4149 4158 4161
    PAGE 445  PROCESS HC.ARRIVE.BATTLE           238  239  245  246  266
    PAGE 448                                     390  392  393  396  396  396
.HELO.LINK
    PAGE 510  PROCESS HELICOPTER.FIRE           3679 3680 3691
    PAGE 511                                    3716 3722 3727 3730 3735 3737 3739 3760 3766
    PAGE 512                                    3774 3784
    PAGE 513                                    3839 3847 3858
    PAGE 514                                    3902 3921
    PAGE 515                                    3966
    PAGE 516                                    4012 4013 4061
    PAGE 518                                    4123 4168
.HEL.FIRE
    PAGE 511  PROCESS HELICOPTER.FIRE           3730 3731 3733
.HFOV
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT         4642 4666 4674
.HIGHX
    PAGE 349  EVENT AD.ENGAGEMENT               5923 5943
    PAGE 353                                    6131 6133 6146
.HIGHY
    PAGE 135  ROUTINE CHECK.FOR.MINES           6314 6317
    PAGE 349  EVENT AD.ENGAGEMENT               5925 5944
    PAGE 353                                    6132 6134 6147
.HIGH.POINT
    PAGE 281  ROUTINE AC.DF.EFFECTS             3074 3090
    PAGE 308  ROUTINE AD.SHOOT                  4368 4384
```

```
.HIT
    PAGE 135  ROUTINE CHECK.FOR.MINES                6274 6278 6325
.HIT.CHANCE
    PAGE 510  PROCESS HELICOPTER.FIRE                3674
    PAGE 515                                         3988 3989
.HOW
    PAGE 252  ROUTINE MINE.EFFECTS                   1615 1620 1621
    PAGE 384  EVENT OFF.LINE.ATTRITION               7440 7445 7446
.HOW.LOSS
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                2957 2959
.HPK
    PAGE 281  ROUTINE AC.DF.EFFECTS                  3096 3098
    PAGE 308  ROUTINE AD.SHOOT                       4390 4392
    PAGE 310                                         4507
.I
    PAGE 252  ROUTINE MINE.EFFECTS                   1583 1593
    PAGE 278  ROUTINE AC.BOMB.EFFECTS                2959
    PAGE 295  ROUTINE END.CAS.MISSION                3825 3834 3837 3846
    PAGE 312  ROUTINE INTER.HELO                     4572 4575 4579 4594 4598
    PAGE 413  EVENT ACT.ATK                          8686 8688 8689 8692 8694 8695
    PAGE 455  PROCESS HEL.TARGET.ACQUISITION          803  807
    PAGE 456                                          809  819  820  829  836  842  852
    PAGE 461                                         1098 1100
    PAGE 464  PROCESS MINE.ASSESS                    1260
    PAGE 507  PROCESS CAS.MISSION                    3555 3564
    PAGE 516  PROCESS HELICOPTER.FIRE                4034
    PAGE 578  ROUTINE MINE.INPUT                     6619 6623 6624 6626 6628
    PAGE 585  ROUTINE TACAIR.INPUT                   6918
    PAGE 587  ROUTINE MADS.INPUT                     6996
.ID
    PAGE 592  ROUTINE AMMO.RPT                       7126 7150 7152 7156 7160 7164 7168 7171 7174 7175
.IL.N.TUBES
    PAGE 595  ROUTINE ANALYSIS.OUTPUT                7244 7251
    PAGE 596                                         7350
    PAGE 597                                         7386 7397
.IM
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9154 9155 9158 9173
    PAGE 576  ROUTINE ILLUM.INPUT                    6509 6511 6512 6514 6515 6516
    PAGE 597  ROUTINE ANALYSIS.OUTPUT                7381 7383 7386 7386
.INCREMENT
    PAGE 239  ROUTINE TARGET.ANALYSIS                 990 1008 1011 1013
    PAGE 360  EVENT CFR.ON                           6415 6435 6438 6440
.INTERCEPT
    PAGE 349  EVENT AD.ENGAGEMENT                    5904 5934 5937 5938
    PAGE 352                                         6104 6106 6107 6126 6127
.INTERSECT
    PAGE 349  EVENT AD.ENGAGEMENT                    5899 5912 5917 5923 5923 5924 5924 5925 5925 5926 5926 5927 5927 5928 5928
                                                     5930 5934 5935 5945 5946 5947 5948 5949 5950 5952 5953
    PAGE 350                                         5987
    PAGE 351                                         6036 6036 6037 6037 6064 6066 6068
    PAGE 352                                         6084 6087
    PAGE 353                                         6148 6149 6150 6151 6160 6161
    PAGE 354                                         6210 6212 6214
    PAGE 505  PROCESS CAS.MISSION                    3437 3441 3444
    PAGE 508                                         3629 3633 3636
```

```
.INTERVAL.BW.ATK
    PAGE 504  PROCESS CAS.MISSION                    3362
    PAGE 507                                         3554 3564
.I1
    PAGE 250  ROUTINE MINE.EFFECTS                   1502 1505 1508
    PAGE 534  ROUTINE UNIT.INPUT                     4777 4780 4783
.I2
    PAGE 250  ROUTINE MINE.EFFECTS                   1503 1506 1508
    PAGE 534  ROUTINE UNIT.INPUT                     4778 4781 4783
.J
    PAGE 295  ROUTINE END.CAS.MISSION                3845 3846
    PAGE 459  PROCESS HEL.TARGET.ACQUISITION          989  991 1002 1017
    PAGE 460                                         1048 1052
.KAS
    PAGE  65  ROUTINE FILE.KAD.SENSOR                3367 3368 3377 3378 3379
    PAGE 317  ROUTINE FLIGHT.PATH                    4822 4823
.KILLED
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                2891
    PAGE 278                                         2918 2920 2920 2921 2929 2932 2943 2950
    PAGE 279                                         2978 2980
    PAGE 282  ROUTINE AC.DF.EFFECTS                  3114 3126 3129
    PAGE 293  ROUTINE END.CAS.MISSION                3727
    PAGE 294                                         3737 3745 3747
.KILLED.IND
    PAGE 419  PROCESS AC.ATK.TGT                     8856
    PAGE 421                                         8980 9020
    PAGE 422                                         9060
    PAGE 423                                         9095 9127
    PAGE 424                                         9178
    PAGE 425                                         9203 9234
    PAGE 426                                         9278 9306
.KILLER
    PAGE 250  ROUTINE MINE.EFFECTS                   1512
    PAGE 251                                         1535
    PAGE 252                                         1578 1580
    PAGE 382  EVENT OFF.LINE.ATTRITION               7310 7311
    PAGE 383                                         7357
    PAGE 384                                         7406 7407 7418 7420
.KILLER.COLOR
    PAGE 103  ROUTINE MINE.DELAY                     4952 4954 4972
    PAGE 104                                         4997 5002
    PAGE 250  ROUTINE MINE.EFFECTS                   1483 1485 1501
    PAGE 252                                         1580 1585 1595
    PAGE 464  PROCESS MINE.ASSESS                    1231 1233
    PAGE 465                                         1296 1316
.KILLER.EQ
    PAGE 250  ROUTINE MINE.EFFECTS                   1508 1509 1512
    PAGE 253                                         1633 1637
.KILL.EM
    PAGE 369  EVENT ENGAGEMENT                       6733 6734 6735
    PAGE 370                                         6736
.KV
    PAGE 514  PROCESS HELICOPTER.FIRE                3920 3922 3925
.KV.SIDE
    PAGE 514  PROCESS HELICOPTER.FIRE                3919 3925
    PAGE 517                                         4069
```

```
.LEFTX
    PAGE 135  ROUTINE CHECK.FOR.MINES                6308 6316
.LINK
    PAGE 123  ROUTINE TIME.TO.DETECT                 5819 5820 5822 5828
    PAGE 181  ROUTINE FASCAM.COMPUTATION             8258 8259 8260 8263
    PAGE 199  ROUTINE ILLUM.COMPUTATION              9149 9150 9151 9154
    PAGE 216  ROUTINE REQUEST.FASCAM                 9925 9926 9929 9939 9940
    PAGE 217                                         9943 9947 9948 9953
    PAGE 218  ROUTINE REQUEST.ILLUM                  9992
    PAGE 219                                           55   56   59   69   70   73   77   78   83
    PAGE 222  ROUTINE REQUEST.SMOKE                   228  229
    PAGE 223                                          232  242  243  246  250  251  256
    PAGE 233  ROUTINE SMOKE.COMPUTATION               707  708  709  712
    PAGE 253  ROUTINE MINE.EFFECTS                   1652 1653 1657
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                2884 2887 2890
    PAGE 278                                         2920 2921 2927 2929 2932 2934 2945 2949
    PAGE 279                                         2973 2979
    PAGE 283  ROUTINE CAS.EVAL                       3192 3194 3195 3197 3198 3199
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS          3297 3298 3305 3307 3318 3321 3324
    PAGE 286                                         3373 3374 3375
    PAGE 287                                         3402 3405
    PAGE 293  ROUTINE END.CAS.MISSION                3691 3693 3694 3717 3719 3722 3725
    PAGE 294                                         3743
    PAGE 295                                         3822 3823
    PAGE 307  ROUTINE AD.SHOOT                       4330 4331 4337 4338
    PAGE 308                                         4411
    PAGE 310                                         4494
    PAGE 312  ROUTINE INTER.HELO                     4592 4597 4602
    PAGE 384  EVENT OFF.LINE.ATTRITION               7424 7425 7433
    PAGE 419  PROCESS AC.ATK.TGT                     8887 8889 8890 8892 8903 8904
    PAGE 420                                         8910 8925 8926 8927 8931 8937 8938
    PAGE 423                                         9086 9088 9089 9091 9102 9103 9118 9120 9121 9123
    PAGE 424                                         9148 9149 9150 9194
    PAGE 425                                         9196 9197 9199 9209 9210 9225 9227 9228 9230 9241 9242
    PAGE 426                                         9269 9271 9272 9274
    PAGE 427                                         9311 9313 9314
    PAGE 464  PROCESS MINE.ASSESS                    1247 1248 1249 1250
    PAGE 507  PROCESS CAS.MISSION                    3565 3566 3567 3570 3570 3573 3578 3578 3579 3579
    PAGE 508                                         3582 3583 3589 3591
.LITED
    PAGE 201  ROUTINE ILLUM.EFFECTS                  9239 9246 9254 9274
.LIT.UNIT
    PAGE 201  ROUTINE ILLUM.EFFECTS                  9241 9248 9256 9275 9275
.LOWX
    PAGE 349  EVENT AD.ENGAGEMENT                    5924 5943
    PAGE 353                                         6131 6133 6146
.LOWY
    PAGE 135  ROUTINE CHECK.FOR.MINES                6312 6317
    PAGE 349  EVENT AD.ENGAGEMENT                    5926 5944
    PAGE 353                                         6132 6134 6147
.LOW.POINT
    PAGE 281  ROUTINE AC.DF.EFFECTS                  3073 3074 3075 3076 3077 3081
    PAGE 308  ROUTINE AD.SHOOT                       4367 4368 4369 4371 4372 4375
.LPK
    PAGE 281  ROUTINE AC.DF.EFFECTS                  3087 3098 3098
    PAGE 308  ROUTINE AD.SHOOT                       4381 4392 4392
```

```
    PAGE 310                                    4507
.M
    PAGE 582  ROUTINE TACAIR.INPUT              6775 6785 6786 6787 6788
    PAGE 621  ROUTINE TACAIR.DATA.REPORT        8382 8384 8385 8386 8387 8388
    PAGE 645  ROUTINE LINE.CIRCLE               9118 9138 9140 9141 9149 9150
.MA
    PAGE 251  ROUTINE MINE.EFFECTS              1532 1537 1540 1546 1565 1567
    PAGE 464  PROCESS MINE.ASSESS               1229 1238 1245 1247 1257 1272
    PAGE 465                                    1320
    PAGE 466                                    1342 1348
.MADS
    PAGE 308  ROUTINE AD.SHOOT                  4407 4408
    PAGE 310                                    4493 4497 4497 4498
    PAGE 587  ROUTINE MADS.INPUT                6985 6988 6989 6990 6991 6992 6993 6994 7004
    PAGE 623  ROUTINE TACAIR.DATA.REPORT        8468 8470 8471 8472 8473 8474 8475 8476 8477 8480
.MADS.NAME
    PAGE 314  ROUTINE FLIGHT.PATH               4639
    PAGE 316                                    4792
    PAGE 349  EVENT AD.ENGAGEMENT               5907
    PAGE 350                                    5974
.MAL
    PAGE 251  ROUTINE MINE.EFFECTS              1566 1567 1568 1569
.MAX.CE
    PAGE 283  ROUTINE CAS.EVAL                  3185 3202 3203
.MAX.UNIT
    PAGE 283  ROUTINE CAS.EVAL                  3186 3204 3210 3215
.MAX.WPN.RANGE
    PAGE 352  EVENT AD.ENGAGEMENT               6081 6082 6083
    PAGE 353                                    6171 6178 6181
.MF
    PAGE 134  ROUTINE CHECK.FOR.MINES           6244 6245 6248 6254 6259 6260 6261 6262 6267 6267 6268 6268
    PAGE 135                                    6276 6321
    PAGE 143  ROUTINE DEAD.UNIT                 6606 6608
    PAGE 410  EVENT UPDATE.LOC                  8527 8528 8530 8531
.MFB
    PAGE 104  ROUTINE MINE.DELAY                4996 4997 5000 5002
.MFB.DELAY
    PAGE 103  ROUTINE MINE.DELAY                4948
    PAGE 104                                    5002 5003 5004
.MFP
    PAGE 135  ROUTINE CHECK.FOR.MINES           6276 6277 6280 6280 6281 6282 6283 6284 6296 6296 6298 6304 6305 6308 6309
                                                6310 6311 6312 6313 6314 6315
.MINE.DIST
    PAGE 104  ROUTINE MINE.DELAY                4994 4997
.MINE.FLAG
    PAGE 134  ROUTINE CHECK.FOR.MINES           6219 6228 6255
    PAGE 250  ROUTINE MINE.EFFECTS              1466
    PAGE 251                                    1555
    PAGE 393  EVENT START.ARTY.MOVEMENT         7684 7698
    PAGE 404  EVENT START.MOVE                  8267
    PAGE 405                                    8305
    PAGE 410  EVENT UPDATE.LOC                  8532 8560
.MINE.KV.ID
    PAGE 464  PROCESS MINE.ASSESS               1219
    PAGE 465                                    1294 1297 1314 1316 1330
```

```
.MIN.ALT
    PAGE 583  ROUTINE TACAIR.INPUT                    6821 6829
.MIN.DIS
    PAGE 103  ROUTINE MINE.DELAY                      4969 4978 4979
    PAGE 218  ROUTINE REQUEST.ILLUM                     20   30   32
    PAGE 222  ROUTINE REQUEST.SMOKE                    208  218  219
    PAGE 226  ROUTINE REQUEST.WD.FASCAM                423  438  439  444  445
    PAGE 227                                           454  479  489  490  498  499
.MIN.RANGE
    PAGE 506  PROCESS CAS.MISSION                     3493 3505 3508
.MIN.UNIT
    PAGE 226  ROUTINE REQUEST.WD.FASCAM                424  440  443
    PAGE 227                                           449  454  464  465  466  491  503
.MISSION
    PAGE 284  ROUTINE CAS.EVAL                        3226 3229 3256 3257 3258
    PAGE 622  ROUTINE TACAIR.DATA.REPORT              8445
    PAGE 623                                          8449 8450 8451 8452
.MO
    PAGE  87  ROUTINE END.MOVE                        4357 4359 4360
    PAGE 135  ROUTINE CHECK.FOR.MINES                 6320 6321 6322 6323 6324
    PAGE 393  EVENT START.ARTY.MOVEMENT               7686 7688 7692 7693 7700
    PAGE 404  EVENT START.MOVE                        8280 8282 8283
    PAGE 405                                          8301 8302 8311 8312 8313
    PAGE 409  EVENT UPDATE.LOC                        8500 8502 8503
    PAGE 410                                          8525 8530 8536 8537 8556 8557 8566 8567 8568
    PAGE 630  FUNCTION COLLISION                      8706 8708 8709
.MODEL.ADS
    PAGE 419  PROCESS AC.ATK.TGT                      8867
    PAGE 420                                          8910
    PAGE 422                                          9044
.MOVCOR
    PAGE  58  ROUTINE CREATE.FORCE                    3141 3142 3144 3145
.MOVE
    PAGE  92  ROUTINE FA.BN.MOVEMENT                  4571 4573 4575
    PAGE 173  ROUTINE DUST.EFFECTS                    7909 7910 7915 7916
    PAGE 174                                          7973 7974 7979 7980 7988 7989 7994 7995
    PAGE 235  ROUTINE SMOKE.EFFECTS                    797  798  803  804
    PAGE 236                                           897  898  903  904
    PAGE 393  EVENT START.ARTY.MOVEMENT               7662
    PAGE 394                                          7728
.MOVE.RATE
    PAGE  80  ROUTINE BLOCK.LOS                       4021 4025 4025 4028 4030
.MSN
    PAGE 181  ROUTINE FASCAM.COMPUTATION              8244 8252 8253 8260 8264 8265 8267 8270 8278 8278
    PAGE 199  ROUTINE ILLUM.COMPUTATION               9135 9143 9144 9151 9155 9156 9162 9164 9168 9172 9172 9173
    PAGE 200  ROUTINE ILLUM.EFFECTS                   9183 9191 9207 9211 9220
    PAGE 201                                          9266 9291
    PAGE 233  ROUTINE SMOKE.COMPUTATION                689  697  698  [illegible]  713  714  719  721  725  729  729  730
    PAGE 234  ROUTINE SMOKE.EFFECTS                    740  748  754  7[illegible]  784
    PAGE 235                                           811  850
    PAGE 236                                           878
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS           3437 3438 3439  [illegible]42 344[illegible]47
.MU
    PAGE 349  EVENT AD.ENGAGEMENT                     5904
    PAGE 351                                          6039 6040 6042
    PAGE 405  EVENT START.MOVE                        8293 8296
```

```
    PAGE 410  EVENT UPDATE.LOC                    8548 8551
.MUN
    PAGE 276  ROUTINE AC.BOMB.EFFECTS             2808 2820 2831 2832 2837
    PAGE 277                                      2877 2879 2879 2893 2900 2901
    PAGE 278                                      2910 2911 2928 2942
.N
    PAGE 282  ROUTINE AC.DF.EFFECTS               3117 3119 3119 3120
    PAGE 340  ROUTINE PROB.INF                    5613 5619 5621 5623 5624 5627 5628
    PAGE 507  PROCESS CAS.MISSION                 3579
    PAGE 516  PROCESS HELICOPTER.FIRE             4043 4045 4047
.NAME
    PAGE 250  ROUTINE MINE.EFFECTS                1477 1488 1491 1494 1509
    PAGE 252                                      1584 1594
    PAGE 253                                      1639
    PAGE 510  PROCESS HELICOPTER.FIRE             3675
    PAGE 512                                      3807 3809 3811
    PAGE 582  ROUTINE TACAIR.INPUT                6743 6777 6779
    PAGE 583                                      6797 6799 6807 6808 6812
    PAGE 585                                      6923 6934
.NEW.BATTLE
    PAGE 395  EVENT START.BATTLE                  7765 7767 7781 7792
    PAGE 396                                      7841
    PAGE 400                                      8090
.NEW.SECTOR
    PAGE 395  EVENT START.BATTLE                  7757
    PAGE 400                                      8065
.NEXT.FIRING
    PAGE 307  ROUTINE AD.SHOOT                    4307 4311
    PAGE 310                                      4491 4493 4498 4503 4506
.NFOV
    PAGE 341  ROUTINE PROB.TIME                   5645 5652 5654
.NLOS
    PAGE  80  ROUTINE BLOCK.LOS                   4030 4032 4032 4034 4037 4045 4047 4047 4049 4052
.NMINES
    PAGE 393  EVENT START.ARTY.MOVEMENT           7685 7703 7706
.NORM.ALT
    PAGE 583  ROUTINE TACAIR.INPUT                6822 6830
.NO.ATTACK
    PAGE 445  PROCESS HC.ARRIVE.BATTLE             241
    PAGE 446                                       287  293
.NO.BARS
    PAGE 120  ROUTINE SEARCH                      5751 5755
    PAGE 340  ROUTINE PROB.INF                    5602 5614 5623 5627 5628
    PAGE 661  ROUTINE JOHNSON.CRITERIA            9617 9622
.NO.ENEMY.UNITS
    PAGE 311  ROUTINE INTER.HELO                  4556
    PAGE 312                                      4572 4594
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION       712
    PAGE 455                                       803
    PAGE 459                                       989
    PAGE 461                                      1098
.NO.FLY
    PAGE 582  ROUTINE TACAIR.INPUT                6769 6773
.NO.IN.SET
    PAGE 487  PROCESS ASSESSMENT                  2421
    PAGE 491                                      2687 2688
```

```
.NO.UNITS
    PAGE 413  EVENT ACT.ATK                          8685 8686 8691 8692
.NO.VICTIMS
    PAGE 516  PROCESS HELICOPTER.FIRE                4029 4032 4034
.NR.ABORTED
    PAGE 504  PROCESS CAS.MISSION                    3358 3401 3403
.NR.AC
    PAGE 504  PROCESS CAS.MISSION                    3356 3389 3394 3401 3403
.NR.SURV.AC
    PAGE 504  PROCESS CAS.MISSION                    3357 3403 3404
    PAGE 506                                         3479
    PAGE 507                                         3554 3555
    PAGE 508                                         3614
.NUM.AC
    PAGE 351  EVENT AD.ENGAGEMENT                    6013 6018 6024
    PAGE 380  EVENT INIT.PREPLAN.CAS                 7233 7249
    PAGE 585  ROUTINE TACAIR.INPUT                   6924
    PAGE 586                                         6972
.NUM.DET
    PAGE 351  EVENT AD.ENGAGEMENT                    6024 6026 6029
    PAGE 353                                         6177 6180 6185
    PAGE 354                                         6207
.NUM.FIRES.PER.PASS
    PAGE 280  ROUTINE AC.DF.EFFECTS                  3001 3013
    PAGE 419  PROCESS AC.ATK.TGT                     8868
    PAGE 421                                         8990 8992 8993 9000 9004
    PAGE 424                                         9139
.NUM.IN.QUEUE
    PAGE 183  ROUTINE FA.BN.ASGN                     8394 8396
    PAGE 208  ROUTINE PGM.MSN.ASGN                   9525 9528
.NUM.KILLED
    PAGE 464  PROCESS MINE.ASSESS                    1273
    PAGE 465                                         1276 1284 1288 1308 1309 1312 1315 1323 1328
    PAGE 466                                         1340
.NUM.MSNS
    PAGE 585  ROUTINE TACAIR.INPUT                   6914 6915 6918
.NUM.PASSES
    PAGE 419  PROCESS AC.ATK.TGT                     8869
    PAGE 421                                         8996 8998 9004 9009 9013 9017
    PAGE 422                                         9022
.NUM.RH
    PAGE 587  ROUTINE MADS.INPUT                     6995 6996
.NUM.SEARCHES
    PAGE 120  ROUTINE SEARCH                         5711 5717 5720
    PAGE 121                                         5767
    PAGE 341  ROUTINE PROB.TIME                      5638 5654
.NUM.WPNS
    PAGE 308  ROUTINE AD.SHOOT                       4409 4411
    PAGE 309                                         4413 4415
    PAGE 310                                         4471 4473 4475 4478 4492 4506
.NX.ORDER
    PAGE 630  FUNCTION COLLISION                     8695 8696 8701
.OBS
    PAGE  65  ROUTINE FILE.KAD.SENSOR                3391 3392 3393 3394 3394 3395 3395 3396 3397 3398 3398 3399 3399 3405 3406
                                                     3407 3408 3409 3410
```

```
.ORDER
    PAGE 113  ROUTINE PREPARE.LIST                5408 5409 5413 5414 5418
    PAGE 630  FUNCTION COLLISION                  8679 8694
.ORDINATE
    PAGE 280  ROUTINE AC.DF.EFFECTS               3010
    PAGE 281                                      3071 3073 3075
    PAGE 307  ROUTINE AD.SHOOT                    4311
    PAGE 308                                      4366 4367 4369
.OTHER.FORCE
    PAGE 236  ROUTINE SMOKE.EFFECTS                862  866  869
.P
    PAGE 277  ROUTINE AC.BOMB.EFFECTS             2898 2900
    PAGE 278                                      2906
.PASS
    PAGE 421  PROCESS AC.ATK.TGT                  9017
    PAGE 422                                      9021
.PD
    PAGE 349  EVENT AD.ENGAGEMENT                 5904
    PAGE 350                                      6001 6003 6003 6005 6005
    PAGE 351                                      6024
    PAGE 353                                      6177
.PK
    PAGE 280  ROUTINE AC.DF.EFFECTS               3010
    PAGE 281                                      3077 3098 3099 3100 3105 3105 3108 3108 3110
    PAGE 307  ROUTINE AD.SHOOT                    4311
    PAGE 308                                      4372 4392 4393 4394
    PAGE 309                                      4415 4415 4416 4429 4445
    PAGE 310                                      4506
    PAGE 510  PROCESS HELICOPTER.FIRE             3669
    PAGE 512                                      3780 3792 3795 3812 3816
.PK.BAR
    PAGE 510  PROCESS HELICOPTER.FIRE             3669
    PAGE 512                                      3789 3793 3800 3812 3816
.PK.PTR
    PAGE 281  ROUTINE AC.DF.EFFECTS               3062 3063 3077 3087 3096
    PAGE 307  ROUTINE AD.SHOOT                    4351 4353
    PAGE 308                                      4372 4381 4390
    PAGE 310                                      4507
.POINT
    PAGE 579  ROUTINE MINE.INPUT                  6653 6656 6657
.POOR.FLY
    PAGE 582  ROUTINE TACAIR.INPUT                6770 6774
.PREP
    PAGE 504  PROCESS CAS.MISSION                 3397 3399
.PROBABILITY.INF
    PAGE 120  ROUTINE SEARCH                      5761
    PAGE 121                                      5766
    PAGE 340  ROUTINE PROB.INF                    5609 5613 5618 5625 5627
.PROB.INF
    PAGE 341  ROUTINE PROB.TIME                   5637 5645 5651 5654 5656
.PROB.TOTAL
    PAGE 341  ROUTINE PROB.TIME                   5645 5650 5651 5656
.PROCESS
    PAGE 350  EVENT AD.ENGAGEMENT                 6008 6009
    PAGE 351                                      6013 6014
    PAGE 505  PROCESS CAS.MISSION                 3430 3432 3434
```

```
.PROJ.AREA
    PAGE 280  ROUTINE AC.DF.EFFECTS                    3020 3022
.P1.TO.P2
    PAGE 419  PROCESS AC.ATK.TGT                       8870
    PAGE 420                                           8913
    PAGE 421                                           8990
    PAGE 424                                           9157
.P2.TO.P3
    PAGE 419  PROCESS AC.ATK.TGT                       8871
    PAGE 420                                           8914
    PAGE 424                                           9157
.P3.TO.P1
    PAGE 419  PROCESS AC.ATK.TGT                       8872
    PAGE 420                                           8917
    PAGE 426                                           9260
.QUANT
    PAGE 493  PROCESS SHOOT.OUT                        2766
    PAGE 494                                           2860 2863
    PAGE 495                                           2865 2869
    PAGE 496                                           2944
.R
    PAGE 645  ROUTINE LINE.CIRCLE                      9122 9142
.RAD
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                  2800
    PAGE 277                                           2846 2847 2852 2856 2858
.RADIUS
    PAGE 173  ROUTINE DUST.EFFECTS                     7887 7897
    PAGE 174                                           7954
    PAGE 218  ROUTINE REQUEST.ILLUM                    9992
    PAGE 219                                             44   48   51
    PAGE 220                                           148
    PAGE 223  ROUTINE REQUEST.SMOKE                    264  267  281
    PAGE 225                                           366
    PAGE 582  ROUTINE TACAIR.INPUT                     6768 6772
.RAD.SQ
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                  2800
    PAGE 277                                           2845 2846 2855 2857 2859
.RANGE
    PAGE  65  ROUTINE FILE.KAD.SENSOR                  3381 3383 3384 3385 3386
    PAGE 120  ROUTINE SEARCH                           5717 5725 5730 5738 5757
    PAGE 123  ROUTINE TIME.TO.DETECT                   5808 5809 5810 5825 5826 5834 5848
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM               9870 9871 9872
    PAGE 218  ROUTINE REQUEST.ILLUM                      29   30   32
    PAGE 222  ROUTINE REQUEST.SMOKE                    217  218  219
    PAGE 226  ROUTINE REQUEST.WD.FASCAM                437  438  439
    PAGE 227                                           488  489  490
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE                4124 4137 4154 4172 4175
    PAGE 303                                           4180
    PAGE 307  ROUTINE AD.SHOOT                         4299 4318
    PAGE 308                                           4358 4359 4366
    PAGE 309                                           4428 4444
    PAGE 310                                           4501
    PAGE 316  ROUTINE FLIGHT.PATH                      4794 4801
    PAGE 323  ROUTINE CONTRAST.TO.FREQ                 5000 5013
    PAGE 340  ROUTINE PROB.INF                         5604 5617 5622
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION          5735 5746 5759
```

```
    PAGE 350  EVENT AD.ENGAGEMENT                5956 5958 5976 5984
    PAGE 351                                     6063
    PAGE 352                                     6082 6083 6088 6098 6108
    PAGE 353                                     6171 6177 6181 6185
    PAGE 354                                     6191 6209
    PAGE 419  PROCESS AC.ATK.TGT                 8873
    PAGE 422                                     9022 9042 9051
    PAGE 424                                     9161 9169
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION      808  815  824
    PAGE 457                                      887  888  889  912  914
    PAGE 458                                      926
    PAGE 459                                     1004 1005 1006 1023 1025 1033
    PAGE 506  PROCESS CAS.MISSION                3504 3505 3506 3508
    PAGE 576  ROUTINE ILLUM.INPUT                6513 6516
    PAGE 577  ROUTINE MINE.INPUT                 6568 6570
    PAGE 580  ROUTINE SMOKE.INPUT                6686 6689
    PAGE 587  ROUTINE MADS.INPUT                 6999 7002
.RANGE.AT.FP
    PAGE 419  PROCESS AC.ATK.TGT                 8874
    PAGE 422                                     9038 9043
    PAGE 424                                     9162
.RANGE.HDM
    PAGE 120  ROUTINE SEARCH                     5708 5717 5725
.RATIO
    PAGE 283  ROUTINE CAS.EVAL                   3173
    PAGE 284                                     3234 3236 3239 3246 3248 3251 3257 3258
.READY
    PAGE 355  EVENT ARTY.OCCUPATION              6236 6243
.RED.CEQ
    PAGE 283  ROUTINE CAS.EVAL                   3214
    PAGE 284                                     3241
.RED.FORCE
    PAGE 283  ROUTINE CAS.EVAL                   3216
    PAGE 284                                     3226
.RED.TGT
    PAGE 283  ROUTINE CAS.EVAL                   3215
    PAGE 284                                     3242
.RED.TKS
    PAGE 283  ROUTINE CAS.EVAL                   3213
    PAGE 284                                     3232 3239 3245 3251
.RED.UNITS
    PAGE 145  ROUTINE FIN.BATTLE                 6698 6700 6736
.REPEAT
    PAGE 558  ROUTINE HE.LA.INPUT                5765 5774 5779 5796 5805
    PAGE 559                                     5810
.REQUESTOR
    PAGE 200  ROUTINE ILLUM.EFFECTS              9192 9195 9202 9203 9208 9209 9210 9217 9218 9219 9223 9224 9230 9237 9237
    PAGE 201                                     9244 9244 9248 9251 9252 9266
    PAGE 216  ROUTINE REQUEST.FASCAM             9887 9925 9932 9933
    PAGE 217                                     9946 9962 9974
    PAGE 218  ROUTINE REQUEST.ILLUM              9983 9999    0    6   15   16   17   28
    PAGE 219                                       48   55   62   63   76   96
    PAGE 220                                       97  102  116  118  124  142  154
    PAGE 221                                      164
    PAGE 222  ROUTINE REQUEST.SMOKE               174  188  189  194  203  204  205  214  228
    PAGE 223                                      235  236  249  267
```

```
    PAGE 224                                        294  295  304  324  326  336
    PAGE 225                                        360  372  383
    PAGE 234  ROUTINE SMOKE.EFFECTS                 749  750  756  765  766  772  773  774  781  782  783  789
    PAGE 235                                        813  814  820  828  830  831
    PAGE 236                                        865  889
.REQ.VOLS
    PAGE 181  ROUTINE FASCAM.COMPUTATION           8266 8267 8270
    PAGE 199  ROUTINE ILLUM.COMPUTATION            9158 9160 9161 9164 9168
    PAGE 233  ROUTINE SMOKE.COMPUTATION             715  717  718  721  725
.RESULT
    PAGE 307  ROUTINE AD.SHOOT                     4306 4354
    PAGE 308                                       4360
    PAGE 309                                       4417 4436
    PAGE 310                                       4506
    PAGE 353  EVENT AD.ENGAGEMENT                  6185
    PAGE 354                                       6197 6199 6203 6203 6206
    PAGE 419  PROCESS AC.ATK.TGT                   8875
    PAGE 422                                       9022 9045 9046 9046 9047 9057 9059 9078
    PAGE 424                                       9163 9164 9164 9175 9177 9183
.RETURN.TIME
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION        709  741  743
.RH
    PAGE 350  EVENT AD.ENGAGEMENT                  5975 5976 5984 5987 6001
    PAGE 587  ROUTINE MADS.INPUT                   6998 7001 7002 7003 7004
    PAGE 623  ROUTINE TACAIR.DATA.REPORT           8480 8481 8482 8483
.RIGHTX
    PAGE 135  ROUTINE CHECK.FOR.MINES              6310 6316
.RNG
    PAGE 456  PROCESS HEL.TARGET.ACQUISITION        810  812  824
    PAGE 513  PROCESS HELICOPTER.FIRE              3876
    PAGE 515                                       3949
.ROOT
    PAGE 314  ROUTINE FLIGHT.PATH                  4638
    PAGE 317                                       4808 4810
    PAGE 645  ROUTINE LINE.CIRCLE                  9130 9138 9144 9145 9146 9146 9147 9148
.ROUNDS
    PAGE 276  ROUTINE AC.BOMB.EFFECTS              2815 2816 2817
    PAGE 277                                       2900
    PAGE 278                                       2910
    PAGE 484  PROCESS FIRE.MISSION                 2284 2287 2290
    PAGE 510  PROCESS HELICOPTER.FIRE              3696 3698
    PAGE 513                                       3887
    PAGE 514                                       3890 3891 3891 3893 3895 3896 3910 3916 3923 3927 3932 3936
    PAGE 515                                       3973 3998
    PAGE 516                                       4029
    PAGE 517                                       4077
    PAGE 518                                       4136
.ROW
    PAGE 276  ROUTINE AC.BOMB.EFFECTS              2791
    PAGE 279                                       2980
    PAGE 280  ROUTINE AC.DF.EFFECTS                3000
    PAGE 282                                       3160
    PAGE 419  PROCESS AC.ATK.TGT                   8876
    PAGE 421                                       8985 8986
    PAGE 424                                       9138 9145
```

```
.RULE
    PAGE 200  ROUTINE ILLUM.EFFECTS                 9238
    PAGE 201                                        9245 9253 9266 9267
    PAGE 216  ROUTINE REQUEST.FASCAM                9891
    PAGE 217                                        9974
    PAGE 218  ROUTINE REQUEST.ILLUM                 9984 9995   13
    PAGE 219                                          39   47   50
    PAGE 220                                         106  128  154
    PAGE 221                                         163
    PAGE 222  ROUTINE REQUEST.SMOKE                  175  184  201  201
    PAGE 223                                         260  260  262  271
    PAGE 224                                         299  312  331  344
    PAGE 225                                         372  382
    PAGE 235  ROUTINE SMOKE.EFFECTS                  832  834  850
    PAGE 236                                         860  864  878
.R1
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2831 2833 2835
    PAGE 277                                        2847 2852 2853 2856
.R1.SQ
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2833
    PAGE 277                                        2855 2857 2859
.R2
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2832 2834 2835
    PAGE 277                                        2847 2852 2858 2860 2868
.R2.SQ
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2834
    PAGE 277                                        2855 2857 2859
.S
    PAGE 582  ROUTINE TACAIR.INPUT                  6752 6764 6765 6766 6767 6771 6772 6773 6774 6785 6786 6787 6788
    PAGE 621  ROUTINE TACAIR.DATA.REPORT            8376 8378 8381 8385 8386 8387 8388
.SCORE
    PAGE 280  ROUTINE AC.DF.EFFECTS                 3033 3035
.SCOREBOARD
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2803 2813
    PAGE 279                                        2980
    PAGE 280  ROUTINE AC.DF.EFFECTS                 3011
    PAGE 282                                        3155 3160
    PAGE 293  ROUTINE END.CAS.MISSION               3686
    PAGE 294                                        3762
    PAGE 295                                        3846
    PAGE 296                                        3850
    PAGE 504  PROCESS CAS.MISSION                   3364 3382 3384
.SCORE1
    PAGE 512  PROCESS HELICOPTER.FIRE               3795 3798 3800 3801 3813 3817
.SCORE2
    PAGE 512  PROCESS HELICOPTER.FIRE               3800 3802 3813 3817
.SCOUT.COUNT
    PAGE 455  PROCESS HEL.TARGET.ACQUISITION         770  775  790
.SECTOR
    PAGE  89  ROUTINE FA.BN.MOVEMENT                4401 4436 4437 4438 4439 4440 4440 4442 4443 4444 4444 4451
    PAGE  90                                        4453
    PAGE  91                                        4516 4517 4518 4519 4520 4520 4522 4523 4524 4524 4531 4533
.SEG
    PAGE  80  ROUTINE BLOCK.LOS                     4017 4018 4033 4034 4034 4036 4037 4041 4042 4048 4049 4049 4051 4052
    PAGE 293  ROUTINE END.CAS.MISSION               3689 3691 3693 3696 3697
    PAGE 316  ROUTINE FLIGHT.PATH                   4781 4783 4783 4784 4784 4785 4785 4786 4786 4787 4787
```

```
    PAGE 317                                        4819 4843 4857 4858
    PAGE 318                                        4867 4868 4870 4871 4874 4875 4877 4878 4895 4896 4906 4911 4915 4916 4919
    PAGE 319                                        4920 4923 4924
    PAGE 505  PROCESS CAS.MISSION                   3423 3425 3427 3437 3446
    PAGE 508                                        3627 3629 3638
.SEG.BEGIN.TIME
    PAGE 504  PROCESS CAS.MISSION                   3362
    PAGE 505                                        3419 3443 3446
    PAGE 508                                        3626 3635 3638
.SENSOR
    PAGE 506  PROCESS CAS.MISSION                   3494 3496 3497 3502 3507 3509
.SENS.TYPE
    PAGE  65  ROUTINE FILE.KAD.SENSOR               3361 3381
.SHOOT.TIME
    PAGE 353  EVENT AD.ENGAGEMENT                   6163 6168 6169 6177
.SHOT.MADE
    PAGE 354  EVENT AD.ENGAGEMENT                   6204 6217
.SI
    PAGE 317  ROUTINE FLIGHT.PATH                   4818 4819 4826 4828 4830 4834 4835 4836 4837 4839 4840 4843 4845 4853 4854
                                                    4856 4857 4858
    PAGE 318                                        4865 4866 4867 4868 4869 4874 4875 4878 4880 4888 4889 4890 4895 4896 4897
                                                    4898 4899 4902 4903 4906 4908 4912 4913 4915 4916 4919
    PAGE 319                                        4920 4922 4929
.SIDE
    PAGE  65  ROUTINE FILE.KAD.SENSOR               3366 3367 3378 3391 3402 3409
    PAGE  89  ROUTINE FA.BN.MOVEMENT                4401 4411 4412 4412 4415 4437 4439 4443 4450 4451
    PAGE  90                                        4453 4505 4506 4506 4509
    PAGE  91                                        4517 4519 4523 4530 4531 4533
    PAGE 145  ROUTINE FIN.BATTLE                    6716 6717
    PAGE 183  ROUTINE FA.BN.ASGN                    8365 8385
    PAGE 231  ROUTINE SIZE.ESTIMATE                  633  674
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2812 2816 2818 2838
    PAGE 279                                        2978
    PAGE 280  ROUTINE AC.DF.EFFECTS                 3048 3050 3052
    PAGE 281                                        3054 3104
    PAGE 282                                        3153
    PAGE 284  ROUTINE CAS.EVAL                      3225 3228 3231 3256 3257 3258 3260 3261 3263
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS         3287 3289 3291 3297 3327
    PAGE 286                                        3348 3393
    PAGE 287                                        3417 3429 3430 3438 3446 3450
    PAGE 288                                        3458 3463 3465
    PAGE 314  ROUTINE FLIGHT.PATH                   4636 4652 4657 4658
    PAGE 315                                        4694
    PAGE 317                                        4822 4833
    PAGE 318                                        4863 4893
    PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE             6636 6647 6648 6653 6653 6654
    PAGE 380  EVENT INIT.PREPLAN.CAS                7240 7242 7244 7245 7247 7254 7255
    PAGE 504  PROCESS CAS.MISSION                   3354 3370 3374 3388 3392 3393 3394
    PAGE 505                                        3410
    PAGE 506                                        3466 3486
    PAGE 507                                        3574
    PAGE 508                                        3620
    PAGE 509                                        3654
    PAGE 533  ROUTINE UNIT.INPUT                    4734
    PAGE 534                                        4750 4756
```

```
.SIGMA
    PAGE 193  ROUTINE FINAL.COVERAGE                8877 8879 8880
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2799 2837 2839 2839 2843 2844
    PAGE 349  EVENT AD.ENGAGEMENT                   5904
    PAGE 351                                        6040 6042
.SIGMA.55
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION       5742 5747 5748 5751 5755 5756
.SIGN
    PAGE 314  ROUTINE FLIGHT.PATH                   4653 4655
    PAGE 315                                        4695 4697 4700 4702 4720
    PAGE 316                                        4746 4750
.SLOPE
    PAGE 134  ROUTINE CHECK.FOR.MINES               6226 6237 6240
    PAGE 135                                        6288 6292 6293 6299 6303 6305 6306
    PAGE 314  ROUTINE FLIGHT.PATH                   4638 4673 4674
    PAGE 315                                        4688 4689
    PAGE 316                                        4785 4787 4797
    PAGE 317                                        4850
    PAGE 318                                        4885 4913
    PAGE 349  EVENT AD.ENGAGEMENT                   5904 5933 5935 5936 5937 5937 5938
    PAGE 352                                        6103 6104 6105 6126 6127
.SLOPE2
    PAGE 134  ROUTINE CHECK.FOR.MINES               6226
    PAGE 135                                        6281 6286 6288 6292 6295 6304 6305
.SM
    PAGE 580  ROUTINE SMOKE.INPUT                   6682 6684 6685 6687 6688 6689
    PAGE 597  ROUTINE ANALYSIS.OUTPUT               7389 7391 7394 7394
.SMOKE
    PAGE 233  ROUTINE SMOKE.COMPUTATION              712  713  715  730
.SM.N.TUBES
    PAGE 595  ROUTINE ANALYSIS.OUTPUT               7244 7252
    PAGE 596                                        7353
    PAGE 597                                        7394 7397
.SNAKE.BITE
    PAGE 377  EVENT HELO.ENGAGEMENT                 7137
    PAGE 378                                        7146 7147 7161
.SO
    PAGE  81  ROUTINE BLOCK.LOS                     4097 4098 4100 4111 4112 4114
    PAGE 278  ROUTINE AC.BOMB.EFFECTS               2934 2936 2938 2939 2940
    PAGE 282  ROUTINE AC.DF.EFFECTS                 3118 3121 3122 3123 3124
    PAGE 293  ROUTINE END.CAS.MISSION               3725 3729 3730 3731
    PAGE 294                                        3735 3741 3742 3743 3744 3749
    PAGE 383  EVENT OFF.LINE.ATTRITION              7374 7379 7380 7381 7382 7383
    PAGE 419  PROCESS AC.ATK.TGT                    8890 8891
    PAGE 423                                        9089 9090 9103 9104 9105 9107 9108 9121 9122
    PAGE 425                                        9197 9198 9210 9211 9212 9214 9215 9228 9229 9242 9243 9244 9246 9248
    PAGE 426                                        9272 9273 9293 9294 9295 9297 9299
    PAGE 464  PROCESS MINE.ASSESS                   1263 1264 1265 1266 1267
    PAGE 465                                        1282 1283 1299 1300 1301 1302
    PAGE 490  PROCESS ASSESSMENT                    2632 2633 2634 2636 2638
    PAGE 508  PROCESS CAS.MISSION                   3589 3590
    PAGE 516  PROCESS HELICOPTER.FIRE               4017 4044 4049 4050 4051 4052
    PAGE 517                                        4096
    PAGE 518                                        4155
.SOG
    PAGE 323  ROUTINE CONTRAST.TO.FREQ              5007 5011 5013
```

```
.SPAM
    PAGE 104  ROUTINE MINE.DELAY                    5030 5031 5036 5037 5038
.SPEED
    PAGE 316  ROUTINE FLIGHT.PATH                   4780
    PAGE 317                                        4859
    PAGE 318                                        4871
    PAGE 319                                        4924
.STATUS
    PAGE 293  ROUTINE END.CAS.MISSION               3679
    PAGE 294                                        3774 3777 3780 3784 3788
.SUBS
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS         3315 3316 3328 3335
    PAGE 286                                        3340 3345
.SUBSTITUTE
    PAGE 286  ROUTINE CHECK.CAS.CONSTRAINTS         3340 3363 3369 3377
.SUBSTITUTE.LINK
    PAGE 286  ROUTINE CHECK.CAS.CONSTRAINTS         3341 3364 3373 3379 3381 3386
    PAGE 287                                        3402
.SUBS.LINK
    PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS         3327 3328
    PAGE 286                                        3341 3343 3361
.SUBS.SUBS
    PAGE 286  ROUTINE CHECK.CAS.CONSTRAINTS         3345 3346 3350 3358 3363
.SUBS.SUBS.LINK
    PAGE 286  ROUTINE CHECK.CAS.CONSTRAINTS         3347 3349 3362 3364
.SUM.CRIT.EQ
    PAGE 419  PROCESS AC.ATK.TGT                    8877
    PAGE 421                                        8971 8982
    PAGE 422                                        9021
    PAGE 424                                        9147 9150
    PAGE 426                                        9255 9287
.T
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2799 2835 2836
    PAGE 277                                        2862 2865 2870 2873
.TACAIR.FLAG
    PAGE 521  ROUTINE MAIN2                         4295 4298
    PAGE 582  ROUTINE TACAIR.INPUT                  6738 6745 6746
.TANK.TE
    PAGE 283  ROUTINE CAS.EVAL                      3184 3194
.TARGET
    PAGE 181  ROUTINE FASCAM.COMPUTATION            8252 8255 8256 8266 8269 8269 8276 8277 8278 8279
    PAGE 199  ROUTINE ILLUM.COMPUTATION             9143 9146 9147 9158 9159 9167 9167 9172 9174 9174
    PAGE 200  ROUTINE ILLUM.EFFECTS                 9193 9237
    PAGE 201                                        9244 9252 9256
    PAGE 216  ROUTINE REQUEST.FASCAM                9888 9902 9903 9909 9917
    PAGE 217                                        9963 9971 9974
    PAGE 218  ROUTINE REQUEST.ILLUM                   15   31   35   36
    PAGE 219                                          51
    PAGE 220                                         106  128  143
    PAGE 221                                         155  165
    PAGE 222  ROUTINE REQUEST.SMOKE                  205  220  223  224
    PAGE 223                                         264
    PAGE 224                                         309  341
    PAGE 225                                         361  373  384
    PAGE 233  ROUTINE SMOKE.COMPUTATION              697  700  701  701  702  704  715  716  724  724  729  731  731
    PAGE 284  ROUTINE CAS.EVAL                      3242 3254 3265 3267 3268
```

```
    PAGE 380  EVENT INIT.PREPLAN.CAS                    7231 7239 7250 7255
.TARGET.EQ
    PAGE 307  ROUTINE AD.SHOOT                          4341 4344 4347 4352
    PAGE 309                                            4426 4427 4442 4443 4453
    PAGE 310                                            4476 4479 4507
.TARGET.EQUIP
    PAGE 369  EVENT ENGAGEMENT                          6718 6719 6723
    PAGE 513  PROCESS HELICOPTER.FIRE                   3869
    PAGE 514                                            3903 3910 3930 3935 3941
    PAGE 515                                            3974 3991 3999
    PAGE 516                                            4015 4016 4030 4035 4037 4041 4044 4055 4056 4062
    PAGE 517                                            4078 4094 4095 4110
    PAGE 518                                            4137 4153 4154
.TARGET.UNIT
    PAGE 300  ROUTINE HC.DISENGAGE                      4046 4048
    PAGE 312  ROUTINE INTER.HELO                        4586 4588
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION             866  873  876  885  894  918
    PAGE 458                                             939  940  955  960  973  977
    PAGE 459                                            1011
    PAGE 513  PROCESS HELICOPTER.FIRE                   3868 3874
    PAGE 516                                            4059
    PAGE 517                                            4108
.TB
    PAGE 181  ROUTINE FASCAM.COMPUTATION                8253 8258 8268 8279
    PAGE 199  ROUTINE ILLUM.COMPUTATION                 9144 9149 9165 9173
    PAGE 233  ROUTINE SMOKE.COMPUTATION                  698  707  722  730
.TE
    PAGE 250  ROUTINE MINE.EFFECTS                      1497 1498
    PAGE 252                                            1584 1594
    PAGE 277  ROUTINE AC.BOMB.EFFECTS                   2890 2897
    PAGE 278                                            2910 2918 2919 2956
    PAGE 530  ROUTINE EQ.TE.INPUT                       4581 4583 4584 4585 4586 4587 4590 4591
.TEAM
    PAGE 311  ROUTINE INTER.HELO                        4552 4559 4565
    PAGE 312                                            4586 4588 4610 4618
    PAGE 391  EVENT SEND.TEAM                           7599 7607 7610 7610 7618 7619
    PAGE 401  EVENT START.BATTLE                        8145 8147
    PAGE 402                                            8150 8152 8156 8158 8161 8164 8168
    PAGE 488  PROCESS ASSESSMENT                        2485 2487 2489 2493
    PAGE 489                                            2558
.TERRAIN
    PAGE  96  ROUTINE LINE.OF.SIGHT                     4667 4693 4695 4699 4701 4711
    PAGE  97                                            4732 4733 4735 4736 4751 4752 4754 4755
    PAGE 210  ROUTINE PGM.MSN.ASGN                      9615 9616 9617
    PAGE 261  ROUTINE BTRY.EFFECTS                      2038
    PAGE 267                                            2416
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION             713
    PAGE 456                                             810  812  825
.TERRAIN.TYPE
    PAGE 478  PROCESS WITH.DRAW                         1961
    PAGE 479                                            1973 1974 1977 1978 1992 2000 2001 2003 2004
.TF.LEADER
    PAGE 104  ROUTINE MINE.DELAY                        5042 5044 5047
.TGT
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                   2807 2824 2830
    PAGE 277                                            2884
```

```
    PAGE 278                                2905 2906 2948 2950 2953 2954
    PAGE 279                                2965 2972 2986 2987 2988
    PAGE 280  ROUTINE AC.DF.EFFECTS         3044
    PAGE 282                                3131 3133 3135 3136 3145 3157
    PAGE 293  ROUTINE END.CAS.MISSION       3684 3717 3728
    PAGE 295                                3795 3822 3825 3845
    PAGE 404  EVENT START.MOVE              8288 8290
    PAGE 405                                8294 8296 8300 8318
    PAGE 410  EVENT UPDATE.LOC              8543 8545 8549 8551 8555 8574
    PAGE 419  PROCESS AC.ATK.TGT            8858 8887
    PAGE 420                                8925 8937 8962
    PAGE 421                                8971 8982 9003 9012
    PAGE 423                                9086 9102 9118
    PAGE 424                                9148 9194
    PAGE 425                                9209 9225 9241
    PAGE 426                                9255 9269 9288
    PAGE 504  PROCESS CAS.MISSION           3360 3383
    PAGE 505                                3452
    PAGE 506                                3494 3497 3501 3515 3517 3521 3523
    PAGE 507                                3526 3528 3565
    PAGE 508                                3584 3606 3607 3608
    PAGE 585  ROUTINE TACAIR.INPUT          6925 6926 6956 6962 6963
    PAGE 586                                6970
.TGT.BTRY
    PAGE 252  ROUTINE MINE.EFFECTS          1609 1610 1613 1615 1620 1625 1626 1631
    PAGE 278  ROUTINE AC.BOMB.EFFECTS       2954 2955 2956 2957
    PAGE 279                                2961 2964 2964 2969
    PAGE 282  ROUTINE AC.DF.EFFECTS         3136 3137 3138 3140 3141 3143 3144 3149
.TGT.EQ
    PAGE 280  ROUTINE AC.DF.EFFECTS         3046
    PAGE 281                                3056 3059 3062
    PAGE 282                                3139 3154
.TGT.EQUIP
    PAGE 511  PROCESS HELICOPTER.FIRE       3767
.TGT.LINK
    PAGE 280  ROUTINE AC.DF.EFFECTS         3045 3046
    PAGE 281                                3111
    PAGE 282                                3113 3116 3118 3119 3130 3132 3158
.TGT.NOS
    PAGE 585  ROUTINE TACAIR.INPUT          6922 6926
.TGT.RADIUS
    PAGE 276  ROUTINE AC.BOMB.EFFECTS       2830 2831 2832
    PAGE 277                                2853 2860 2868
.TGT.RANGE
    PAGE 419  PROCESS AC.ATK.TGT            8854
    PAGE 422                                9077
    PAGE 424                                9159 9159
.TGT.SCORE
    PAGE 510  PROCESS HELICOPTER.FIRE       3673
    PAGE 513                                3846 3848 3857 3859
.THETA1
    PAGE 276  ROUTINE AC.BOMB.EFFECTS       2799
    PAGE 277                                2855 2861 2862 2864 2865 2869 2870 2872 2873
.THETA2
    PAGE 276  ROUTINE AC.BOMB.EFFECTS       2799
    PAGE 277                                2857 2861 2862 2864 2865 2869 2870 2872 2873
```

```
.THE.KILLER
    PAGE 516  PROCESS HELICOPTER.FIRE                4061
    PAGE 517                                         4064 4069
.THE.VICTIM
    PAGE 516  PROCESS HELICOPTER.FIRE                4062
    PAGE 517                                         4065 4070
.TIME
    PAGE 120  ROUTINE SEARCH                         5713 5721
    PAGE 121                                         5769
    PAGE 510  PROCESS HELICOPTER.FIRE                3672
    PAGE 515                                         3949 3951
.TIME.IN.RANGE
    PAGE 353  EVENT AD.ENGAGEMENT                    6162 6164
.TIME.TO.ACQUIRE
    PAGE 341  ROUTINE PROB.TIME                      5646 5655 5657
.TIME.TO.DET
    PAGE 123  ROUTINE TIME.TO.DETECT                 5801 5814 5840 5840 5848
    PAGE 124                                         5862 5868
.TIME.TO.DETECT
    PAGE 341  ROUTINE PROB.TIME                      5641 5645 5648 5657
    PAGE 454  PROCESS HEL.TARGET.ACQUISITION          708
    PAGE 457                                          881
    PAGE 458                                          932  933  956  961  974
    PAGE 459                                         1015 1039 1039
    PAGE 460                                         1049 1053
.TIME.TO.LEAVE
    PAGE 419  PROCESS AC.ATK.TGT                     8878
    PAGE 421                                         8975 8981 9020
    PAGE 426                                         9287
.TK.COUNT
    PAGE 283  ROUTINE CAS.EVAL                       3187 3195 3208 3213
.TOF
    PAGE 307  ROUTINE AD.SHOOT                       4311
    PAGE 310                                         4501 4503
.TOTAL.AREA
    PAGE 280  ROUTINE AC.DF.EFFECTS                  3022 3023 3026 3027 3033
.TOTAL.DELAY
    PAGE 393  EVENT START.ARTY.MOVEMENT              7699 7706
    PAGE 394                                         7732
    PAGE 405  EVENT START.MOVE                       8308 8317 8318
    PAGE 406                                         8365
    PAGE 410  EVENT UPDATE.LOC                       8563 8572 8574
    PAGE 411                                         8594
.TOTAL.DISTANCE
    PAGE 353  EVENT AD.ENGAGEMENT                    6159 6162
.TOTAL.TIME
    PAGE 505  PROCESS CAS.MISSION                    3451 3456 3456 3460
.TOTAL.XMIS
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION        5743 5760 5761
.TOT.HOW.LOSS
    PAGE 252  ROUTINE MINE.EFFECTS                   1611 1612 1616 1619
.TR
    PAGE 200  ROUTINE ILLUM.EFFECTS                  9191 9192 9193
    PAGE 201                                         9266 9292
    PAGE 216  ROUTINE REQUEST.FASCAM                 9907 9908 9909
    PAGE 217                                         9958 9960 9961 9962 9963 9964 9965 9966 9967 9968 9969 9970 9974
```

```
    PAGE 218  ROUTINE REQUEST.ILLUM                  9993
    PAGE 219                                           92   93   94   95
    PAGE 220                                           97  100  100  102  106  121  122  122  124  128  137  140  141  142  143
                                                      144  145  146  147  148  149  150  151  154
    PAGE 224  ROUTINE REQUEST.SMOKE                   290  291  292  293  295  298  298  304  309  329  330  330  336  341
    PAGE 225                                          355  358  359  360  361  362  363  364  365  366  367  368  369  372
    PAGE 234  ROUTINE SMOKE.EFFECTS                   748  749  755
    PAGE 235                                          827  827  829  829  841  851
    PAGE 236                                          859  859  861  878
.TRAVEL.DIST
    PAGE 351  EVENT AD.ENGAGEMENT                    6035 6038 6061 6062
.TRAVEL.TIME
    PAGE 349  EVENT AD.ENGAGEMENT                    5904
    PAGE 351                                         6038 6044 6048 6051
    PAGE 353                                         6177 6182
.TW
    PAGE 227  ROUTINE REQUEST.WD.FASCAM               469  470  473
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT              4649 4652 4653 4654 4655 4656 4660 4661 4662 4663 4664 4665 4668 4669 4670
                                                     4671 4672 4673 4674 4675
.TYPE
    PAGE 103  ROUTINE MINE.DELAY                     4937 4985
    PAGE 104                                         5014 5018
    PAGE 250  ROUTINE MINE.EFFECTS                   1464 1487 1490 1493 1514
    PAGE 251                                         1524
.TYPE.MISSION
    PAGE 484  PROCESS FIRE.MISSION                   2283 2286 2291
.TYPE.PROCESS
    PAGE 307  ROUTINE AD.SHOOT                       4303 4313 4319 4320 4340 4343 4346
    PAGE 309                                         4455 4460
    PAGE 349  EVENT AD.ENGAGEMENT                    5907
    PAGE 351                                         6015 6020
    PAGE 354                                         6195
.TYPE.WPN
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2809 2815 2816 2817
    PAGE 280  ROUTINE AC.DF.EFFECTS                  3043 3049 3050 3051 3051
    PAGE 281                                         3055 3056 3058 3059 3062 3064 3065 3070 3070 3071 3108
    PAGE 307  ROUTINE AD.SHOOT                       4339 4351
    PAGE 308                                         4358 4359 4365 4365 4366
    PAGE 310                                         4471 4473 4476 4479 4483 4495 4501 4506
.T.AVERAGE
    PAGE 341  ROUTINE PROB.TIME                      5646 5654 5655
.T.SQ
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2799 2836
    PAGE 277                                         2861 2864 2869 2872
.UEL
    PAGE 123  ROUTINE TIME.TO.DETECT                 5816 5817 5836
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4160 4161 4168
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION          917  919
    PAGE 458                                          928
    PAGE 461                                         1100 1102 1108 1114 1116 1118
    PAGE 611  ROUTINE OUTPUT.ATTRITION               7956 7957 7984
.UE.LINK
    PAGE  81  ROUTINE BLOCK.LOS                      4093 4095 4097 4102 4107 4109 4111 4116
    PAGE 464  PROCESS MINE.ASSESS                    1248 1251 1252 1253 1256 1257 1260 1264 1267
    PAGE 465                                         1277 1282 1289 1291 1308 1309 1311 1322 1325
    PAGE 466                                         1339
```

```
.UL
    PAGE 104  ROUTINE MINE.DELAY                    5046 5047
    PAGE 105                                        5050 5051 5052
.UN
    PAGE 173  ROUTINE DUST.EFFECTS                  7904 7905 7906 7907 7910 7919
.UNIT
    PAGE  80  ROUTINE BLOCK.LOS                     4007 4017 4021 4023 4027 4028 4042 4056 4060
    PAGE  81                                        4063 4071 4074 4081 4084 4093
    PAGE  89  ROUTINE FA.BN.MOVEMENT                4410 4411 4424 4425 4432
    PAGE 103  ROUTINE MINE.DELAY                    4934 4951 4957 4957 4960 4961
    PAGE 104                                        5006 5011 5021 5023 5023 5029 5031 5041 5042 5044
    PAGE 134  ROUTINE CHECK.FOR.MINES               6213 6229 6230 6231 6232 6236 6237 6238 6245 6249
    PAGE 135                                        6324
    PAGE 138  ROUTINE CHECK.PROX                    6422 6426 6428 6429
    PAGE 139                                        6432 6434 6443 6445 6446 6449 6451
    PAGE 145  ROUTINE FIN.BATTLE                    6709 6711 6712 6716
    PAGE 147  ROUTINE INTER.BATTLE                  6804 6806 6807
    PAGE 173  ROUTINE DUST.EFFECTS                  7884 7897 7902 7923 7924 7929 7939
    PAGE 174                                        7940 7942 7945 7946 7948 7948 7952 7957 7961 7985 7986 7987 7989
    PAGE 175                                        7998
    PAGE 201  ROUTINE ILLUM.EFFECTS                 9271 9277 9284
    PAGE 216  ROUTINE REQUEST.FASCAM                9938 9939
    PAGE 217                                        9945 9946 9947
    PAGE 218  ROUTINE REQUEST.ILLUM                 9992
    PAGE 219                                          68   69   75   76   77
    PAGE 223  ROUTINE REQUEST.SMOKE                  241  242  248  249  250  279  281
    PAGE 226  ROUTINE REQUEST.WD.FASCAM              393  406  407  412  419  419  420  434  448
    PAGE 227                                         456  473  485  497  504
    PAGE 234  ROUTINE SMOKE.EFFECTS                  792  793  794  795
    PAGE 235                                         798  807  837  845  852
    PAGE 236                                         879  892  893  894  895  898  907
    PAGE 250  ROUTINE MINE.EFFECTS                  1461 1482
    PAGE 251                                        1521 1529 1534 1540 1549 1556 1574
    PAGE 252                                        1576 1604 1606 1607 1609 1627
    PAGE 253                                        1636 1644 1652 1665
    PAGE 283  ROUTINE CAS.EVAL                      3189 3192 3204
    PAGE 316  ROUTINE FLIGHT.PATH                   4793
    PAGE 382  EVENT OFF.LINE.ATTRITION              7289 7289 7296 7319
    PAGE 383                                        7372
    PAGE 384                                        7400 7401 7402 7403 7424 7435 7436 7437 7451 7452
    PAGE 385                                        7460
    PAGE 400  EVENT START.BATTLE                    8053 8057 8062
    PAGE 439  PROCESS FORWARD.OBSERVER              9915 9917 9918 9920 9923 9935 9938 9941
    PAGE 464  PROCESS MINE.ASSESS                   1218 1230 1239 1265
    PAGE 465                                        1290 1292 1310 1312 1321
    PAGE 466                                        1337 1349
    PAGE 489  PROCESS ASSESSMENT                    2540 2542 2546
    PAGE 573  ROUTINE FARRP.INPUT                   6420 6421
    PAGE 630  FUNCTION COLLISION                    8679 8685 8694 8695 8696 8706 8708
.UNITA
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM            9830 9843 9846 9850 9851 9853
.UNITB
    PAGE 215  ROUTINE REQUEST.DEF.FASCAM            9831 9844 9845 9847 9849 9852
.UNIT.CE
    PAGE 283  ROUTINE CAS.EVAL                      3191 3199 3202 3203
```

```
.UNIT.NOS
    PAGE 382  EVENT OFF.LINE.ATTRITION              7284 7285 7289 7292 7305 7314 7323 7333
.UNLIT
    PAGE 201  ROUTINE ILLUM.EFFECTS                 9240 9247 9255 9259 9271
.UNT
    PAGE 611  ROUTINE OUTPUT.ATTRITION              7976 7977 7977 7985
.UPPER1
    PAGE 577  ROUTINE MINE.INPUT                    6580 6584
.UPPER2
    PAGE 577  ROUTINE MINE.INPUT                    6582 6585
.VELOCITY
    PAGE 351  EVENT AD.ENGAGEMENT                   6014 6019 6038 6061 6062
    PAGE 353                                        6162 6163
.VFOV
    PAGE 532  ROUTINE TYPE.WEAPON.INPUT             4642 4667 4675
.VH
    PAGE 461  PROCESS HEL.TARGET.ACQUISITION        1127 1128 1130 1131
.VICTIM
    PAGE 251  ROUTINE MINE.EFFECTS                  1560
    PAGE 252                                        1578 1580
    PAGE 282  ROUTINE AC.DF.EFFECTS                 3116 3119
    PAGE 382  EVENT OFF.LINE.ATTRITION              7301 7302 7320
    PAGE 384                                        7406 7407 7412 7414 7438
    PAGE 385                                        7460
    PAGE 464  PROCESS MINE.ASSESS                   1251
    PAGE 465                                        1294 1297 1314 1316
    PAGE 516  PROCESS HELICOPTER.FIRE               4041 4045
.VIS
    PAGE 581  ROUTINE VIS.INPUT                     6720 6722 6722 6724 6728
.VISIBILITY
    PAGE 366  EVENT CHANGE.WEATHER                  6618 6623
.VIS.IND
    PAGE 419  PROCESS AC.ATK.TGT                    8855
    PAGE 423                                        9100
    PAGE 425                                        9208 9240
    PAGE 426                                        9310
.VU
    PAGE  80  ROUTINE BLOCK.LOS                     4056 4057 4060 4061 4062
    PAGE  81                                        4063 4066 4067
    PAGE 201  ROUTINE ILLUM.EFFECTS                 9277 9278 9279
    PAGE 461  PROCESS HEL.TARGET.ACQUISITION        1125 1127 1130
.WAIT.TIME
    PAGE 419  PROCESS AC.ATK.TGT                    8879
    PAGE 422                                        9021 9031 9047 9061 9061 9064 9065 9078
    PAGE 423                                        9099 9112 9131
    PAGE 424                                        9158 9165 9179 9179 9184
    PAGE 425                                        9207 9219
    PAGE 426                                        9260 9263
.WEAPON
    PAGE 123  ROUTINE TIME.TO.DETECT                5822 5823 5824 5825 5826 5829 5830 5833
    PAGE 307  ROUTINE AD.SHOOT                      4338 4339
    PAGE 308                                        4411
    PAGE 309                                        4413 4422 4425 4438 4441
    PAGE 310                                        4492 4494 4495
    PAGE 420  PROCESS AC.ATK.TGT                    8938 8939 8940
    PAGE 446  PROCESS HC.ARRIVE.BATTLE               274  275  277  285  287  288  291  292
```

```
    PAGE 457  PROCESS HEL.TARGET.ACQUISITION          907  909  911  912  914  921  922
    PAGE 458                                          925
    PAGE 459                                         1020 1021 1022 1023 1025 1028 1029 1032
.WPN
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2790 2808 2809
    PAGE 279                                         2983
    PAGE 280  ROUTINE AC.DF.EFFECTS                  2999 3043 3049 3052
    PAGE 281                                         3055 3058 3109
    PAGE 295  ROUTINE END.CAS.MISSION                3835 3838 3839 3841 3842
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4168 4169 4170
    PAGE 377  EVENT HELO.ENGAGEMENT                  7117 7118
    PAGE 421  PROCESS AC.ATK.TGT                     8979 8989 8995 8998 8999 8999 9009 9010 9020
    PAGE 422                                         9023 9026 9027
    PAGE 423                                         9133
    PAGE 424                                         9137 9144
    PAGE 513  PROCESS HELICOPTER.FIRE                3870 3880 3886 3889
    PAGE 514                                         3891 3894 3901 3909 3910 3915 3918 3924 3927 3929 3934 3940
    PAGE 515                                         3949 3972 3997
    PAGE 516                                         4011 4014 4028
    PAGE 517                                         4076 4090 4093 4115
    PAGE 518                                         4135 4149 4152
.WPN.STATUS
    PAGE 535  ROUTINE UNIT.INPUT                     4834
.X
    PAGE 276  ROUTINE AC.BOMB.EFFECTS                2799
    PAGE 277                                         2900
    PAGE 278                                         2903 2910 2913
    PAGE 349  EVENT AD.ENGAGEMENT                    5930 5936 5938 5943 5943 5949 5952
    PAGE 350                                         5956
    PAGE 352                                         6098 6126 6127
    PAGE 353                                         6135 6152 6155 6160
    PAGE 583  ROUTINE TACAIR.INPUT                   6831 6834 6837 6840 6843 6846
.XAIM
    PAGE 218  ROUTINE REQUEST.ILLUM                    16   35
    PAGE 220                                         146
    PAGE 222  ROUTINE REQUEST.SMOKE                  203  223
    PAGE 225                                         364
.XCOR
    PAGE 103  ROUTINE MINE.DELAY                     4935 4975
    PAGE 104                                         4990 4994 5012
    PAGE 216  ROUTINE REQUEST.FASCAM                 9889 9901 9902
    PAGE 217                                         9966
    PAGE 250  ROUTINE MINE.EFFECTS                   1462
    PAGE 251                                         1522
.XEND
    PAGE 134  ROUTINE CHECK.FOR.MINES                6214 6229 6230 6236 6238
    PAGE 135                                         6294 6299 6303 6306
    PAGE 393  EVENT START.ARTY.MOVEMENT              7675 7679 7706
.XHIGH
    PAGE 578  ROUTINE MINE.INPUT                     6615 6630 6630 6639 6648
.XIMPACT
    PAGE   3  PROGRAM REVISIONS                       161
    PAGE 173  ROUTINE DUST.EFFECTS                   7885 7895
    PAGE 174                                         7949
    PAGE 193  ROUTINE FINAL.COVERAGE                 8879 8884
```

```
.XINTER
    PAGE 135  ROUTINE CHECK.FOR.MINES                6294 6295 6298 6299 6303 6306 6316 6316 6318 6318 6322
.XLEFT
    PAGE 134  ROUTINE CHECK.FOR.MINES                6229 6233 6260
    PAGE 135                                         6318
.XLOW
    PAGE 578  ROUTINE MINE.INPUT                     6614 6629 6629 6638 6647
.XMAX
    PAGE 316  ROUTINE FLIGHT.PATH                    4784
    PAGE 317                                         4811 4831
.XMID
    PAGE 134  ROUTINE CHECK.FOR.MINES                6233 6267 6268
.XMIN
    PAGE 316  ROUTINE FLIGHT.PATH                    4783
    PAGE 317                                         4812 4831
    PAGE 318                                         4862
.XMINEFIELD
    PAGE 577  ROUTINE MINE.INPUT                     6541 6543
    PAGE 578                                         6623 6626 6634 6635
.XOBJ
    PAGE 314  ROUTINE FLIGHT.PATH                    4661 4666 4673
    PAGE 315                                         4716
    PAGE 316                                         4762
.XRIGHT
    PAGE 134  ROUTINE CHECK.FOR.MINES                6230 6233 6259
    PAGE 135                                         6318
.XSTART
    PAGE 314  ROUTINE FLIGHT.PATH                    4657 4673 4674 4679
.XVEL
    PAGE 349  EVENT AD.ENGAGEMENT                    5905
    PAGE 351                                         6061 6064
    PAGE 354                                         6210
.X0
    PAGE 316  ROUTINE FLIGHT.PATH                    4790 4799
    PAGE 317                                         4842 4848 4849
    PAGE 318                                         4877 4883 4884 4905 4911 4912 4913
    PAGE 349  EVENT AD.ENGAGEMENT                    5918
    PAGE 645  ROUTINE LINE.CIRCLE                    9120 9141 9142
.X1
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4133 4137
    PAGE 303                                         4180
    PAGE 317  ROUTINE FLIGHT.PATH                    4804 4811 4831 4834 4848 4849 4850 4853
    PAGE 318                                         4862 4865 4870 4883 4884 4885 4888
    PAGE 319                                         4928
    PAGE 352  EVENT AD.ENGAGEMENT                    6087 6088 6097 6124
    PAGE 353                                         6131 6131 6149 6155 6159
    PAGE 645  ROUTINE LINE.CIRCLE                    9126 9147 9149
.X2
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4135 4137
    PAGE 303                                         4180
    PAGE 317  ROUTINE FLIGHT.PATH                    4806 4812 4831 4836 4839 4842
    PAGE 318                                         4897 4902 4905
    PAGE 319                                         4923
    PAGE 352  EVENT AD.ENGAGEMENT                    6097 6125
    PAGE 353                                         6133 6135 6146 6148 6152 6159
    PAGE 645  ROUTINE LINE.CIRCLE                    9128 9148 9150
```

```
.Y
    PAGE 349  EVENT AD.ENGAGEMENT                5931 5938 5944 5944 5950 5953
    PAGE 350                                     5957
    PAGE 353                                     6136 6153 6156 6161
    PAGE 583  ROUTINE TACAIR.INPUT               6832 6835 6838 6841 6844 6847
.YAIM
    PAGE 218  ROUTINE REQUEST.ILLUM               17   36
    PAGE 220                                     147
    PAGE 222  ROUTINE REQUEST.SMOKE              204  224
    PAGE 225                                     365
.YCOR
    PAGE 103  ROUTINE MINE.DELAY                 4936 4976
    PAGE 104                                     4991 5013
    PAGE 216  ROUTINE REQUEST.FASCAM             9890 9901 9903
    PAGE 217                                     9967
    PAGE 250  ROUTINE MINE.EFFECTS               1463
    PAGE 251                                     1523
.YEND
    PAGE 134  ROUTINE CHECK.FOR.MINES            6215 6231 6232 6237
    PAGE 135                                     6300 6305 6306
.YHIGH
    PAGE 134  ROUTINE CHECK.FOR.MINES            6232 6234 6261
    PAGE 135                                     6319
    PAGE 578  ROUTINE MINE.INPUT                 6617 6632 6632 6641
    PAGE 579                                     6650
.YI
    PAGE 645  ROUTINE LINE.CIRCLE                9119 9138 9141 9142 9143 9149 9150
.YIMPACT
    PAGE   3  PROGRAM REVISIONS                   161
    PAGE 173  ROUTINE DUST.EFFECTS               7886 7895
    PAGE 174                                     7949
    PAGE 193  ROUTINE FINAL.COVERAGE             8880 8885
.YINTER
    PAGE 135  ROUTINE CHECK.FOR.MINES            6295 6299 6306 6317 6317 6319 6319 6323
.YLOW
    PAGE 134  ROUTINE CHECK.FOR.MINES            6231 6234 6262
    PAGE 135                                     6319
    PAGE 578  ROUTINE MINE.INPUT                 6616 6631 6631 6640
    PAGE 579                                     6649
.YMID
    PAGE 134  ROUTINE CHECK.FOR.MINES            6234 6267 6268
.YMINEFIELD
    PAGE 577  ROUTINE MINE.INPUT                 6542 6544
    PAGE 578                                     6624 6628 6635 6635
.YOBJ
    PAGE 314  ROUTINE FLIGHT.PATH                4663 4666 4673
    PAGE 315                                     4721
    PAGE 316                                     4763
.YSTART
    PAGE 314  ROUTINE FLIGHT.PATH                4658 4673 4674 4680
.YVEL
    PAGE 349  EVENT AD.ENGAGEMENT                5905
    PAGE 351                                     6062 6066
    PAGE 354                                     6212
.Y0
    PAGE 316  ROUTINE FLIGHT.PATH                4791 4800
```

```
    PAGE 349  EVENT AD.ENGAGEMENT                    5919
    PAGE 645  ROUTINE LINE.CIRCLE                    9121 9141 9142 9143
.Y1
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4134 4137
    PAGE 303                                         4180
    PAGE 315  ROUTINE FLIGHT.PATH                    4688 4690 4696 4703 4727 4733 4734 4735 4738 4739 4740
    PAGE 316                                         4747 4751
    PAGE 317                                         4805 4835 4850 4854
    PAGE 318                                         4866 4871 4885 4889
    PAGE 352  EVENT AD.ENGAGEMENT                    6100 6126
    PAGE 353                                         6132 6132 6151 6156 6159
    PAGE 645  ROUTINE LINE.CIRCLE                    9127 9149
.Y2
    PAGE 302  ROUTINE HEL.RANGE.COMPUTE              4136 4137
    PAGE 303                                         4180
    PAGE 315  ROUTINE FLIGHT.PATH                    4689 4691 4698 4701 4730 4731 4732
    PAGE 317                                         4807 4837 4840
    PAGE 318                                         4898 4903
    PAGE 319                                         4924
    PAGE 352  EVENT AD.ENGAGEMENT                    6101 6127
    PAGE 353                                         6134 6136 6147 6150 6153 6159
    PAGE 645  ROUTINE LINE.CIRCLE                    9129 9150
.Z
    PAGE 583  ROUTINE TACAIR.INPUT                   6833 6836 6839 6842 6845 6848
.0001
    PAGE 657  FUNCTION AR.PROB.DETECT                9449
.001
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS             3566
    PAGE 291                                         3601
    PAGE 292                                         3652
    PAGE 510  PROCESS HELICOPTER.FIRE                3709
.0037
    PAGE 254  ROUTINE FO.DETECTION                   1719
.0079
    PAGE 654  PROCESS AIRBORNE.RADAR                 9328
.01
    PAGE 443  PROCESS HC.ARRIVE.BATTLE                140
.02
    PAGE 150  ROUTINE PK.COMPUTE                     6999
    PAGE 308  ROUTINE AD.SHOOT                       4372
.0548
    PAGE 157  ROUTINE AO.DETECTION                   7251
.1
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION        5746
    PAGE 435  PROCESS ARTY.ASSESS                    9725
    PAGE 464  PROCESS MINE.ASSESS                    1269
    PAGE 478  PROCESS WITH.DRAW                      1924
    PAGE 487  PROCESS ASSESSMENT                     2451
.10
    PAGE 663  ROUTINE TIME.REQ                       9651
.101
    PAGE 493  PROCESS SHOOT.OUT                      2787
.16
    PAGE 556  ROUTINE MUNS.INPUT                     5705
.18
    PAGE 323  ROUTINE CONTRAST.TO.FREQ               5025
```

```
.2
    PAGE 493  PROCESS SHOOT.OUT                          2780
.25
    PAGE 590  ROUTINE TR.INPUT                           7092
.3
    PAGE 323  ROUTINE CONTRAST.TO.FREQ                   5012
.3333
    PAGE 323  ROUTINE CONTRAST.TO.FREQ                   5011
.35
    PAGE 323  ROUTINE CONTRAST.TO.FREQ                   5015
.5
    PAGE  67  ROUTINE GENERAL.BATTLE                     3500
    PAGE  71  ROUTINE ORIENTATION                        3688
    PAGE  72                                             3727
    PAGE  73                                             3759 3760 3761 3762 3778 3779 3780 3781 3790
    PAGE  74                                             3808 3818
    PAGE  75  ROUTINE UNIT.ASSIGNMENT                    3864
    PAGE  77  ROUTINE ADJUST                             3913 3920
    PAGE 191  ROUTINE FINAL.COVERAGE                     8759 8760 8767 8768 8769 8770
    PAGE 192                                             8784 8785 8786 8787
    PAGE 217  ROUTINE REQUEST.FASCAM                     9969
    PAGE 220  ROUTINE REQUEST.ILLUM                       150
    PAGE 225  ROUTINE REQUEST.SMOKE                       368
    PAGE 249  ROUTINE WEIGHTED.VOLLEYS                   1454
    PAGE 321  ROUTINE COMPUTE.D                          4960
    PAGE 322  ROUTINE COMPUTE.WD                         4994
    PAGE 363  EVENT CFR.OPERATOR                         6559
    PAGE 387  EVENT PDB.OPERATOR                         7547
    PAGE 432  PROCESS AIR.OBSERVER                       9587
    PAGE 441  PROCESS FORWARD.OBSERVER                     47
    PAGE 448  PROCESS HC.ARRIVE.BATTLE                    389
    PAGE 468  PROCESS REMOTE.PILOT.VEHICLE               1456
    PAGE 496  PROCESS SHOOT.OUT                          2955
    PAGE 654  PROCESS AIRBORNE.RADAR                     9368
    PAGE 659  PROCESS PHOTO.IR.FLIGHT                    9569
.50
    PAGE 428  PROCESS AIR.OBSERVER                       9368 9370
.54
    PAGE 254  ROUTINE FO.DETECTION                       1719
.575
    PAGE 654  PROCESS AIRBORNE.RADAR                     9328
.6667
    PAGE 323  ROUTINE CONTRAST.TO.FREQ                   5011
.667
    PAGE 350  EVENT AD.ENGAGEMENT                        5993
.675
    PAGE 193  ROUTINE FINAL.COVERAGE                     8877
.98
    PAGE 344  ROUTINE TEMPERATURE.ATTENUATION            5748
..FIRING.TABLE
    PAGE 513  PROCESS HELICOPTER.FIRE                    3847 3848 3854 3858 3859 3865 3868 3869 3870
    PAGE 515                                             3975 3989 4000
    PAGE 516                                             4018
    PAGE 517                                             4079 4097
    PAGE 518                                             4138 4156
```

```
..HELICOPTER
    PAGE 455  PROCESS HEL.TARGET.ACQUISITION          755  756  763  800  801  806
    PAGE 456                                          828  839  843  849  861  862  864
    PAGE 457                                          872  876  884  901  910
    PAGE 458                                          942  972  977
    PAGE 459                                          993  994 1001 1035
    PAGE 460                                         1047 1051
..ORDER
    PAGE 113  ROUTINE PREPARE.LIST                   5419 5420 5424 5426 5431
..PK
    PAGE 260  ROUTINE BTRY.EFFECTS                   1979
    PAGE 265                                         2281 2282 2285 2286
..RANGE
    PAGE 511  PROCESS HELICOPTER.FIRE                3757 3762 3768
    PAGE 512                                         3776 3785
    PAGE 515                                         3976 4001
    PAGE 516                                         4017
    PAGE 517                                         4080 4096
    PAGE 518                                         4139 4155
..TE
    PAGE 251  ROUTINE MINE.EFFECTS                   1559 1561 1562
    PAGE 252                                         1585 1595 1610
..TGT
    PAGE 493  PROCESS SHOOT.OUT                      2767
    PAGE 495                                         2921
    PAGE 496                                         2931 2935
..WEAPON
    PAGE 512  PROCESS HELICOPTER.FIRE                3781 3791 3794 3806 3809 3812 3816
..X
    PAGE 260  ROUTINE BTRY.EFFECTS                   1979
    PAGE 265                                         2280 2281
\ROUTINE.NAME?
    PAGE 660  FUNCTION STAY.TIME                     9600
```

```
APR79_%JN
    PAGE 107  ROUTINE NEW.SEGMENT                    5111
DEC_%RWF
    PAGE 431  PROCESS AIR.OBSERVER                   9539
JUL80%CACI
    PAGE 356  EVENT BTL.ENDED                        6278 6279 6281 6290 6291 6293
MAY80%CACI
    PAGE 356  EVENT BTL.ENDED                        6268 6269 6270 6271 6272 6273 6274 6275 6276 6277 6282 6283 6284 6285 6286
                                                     6287 6288 6289 6294 6295
SEP79_%RGR
    PAGE 428  PROCESS AIR.OBSERVER                   9361
%
    PAGE 133  ROUTINE CHECK.FORCE                    6185
    PAGE 142  ROUTINE DEAD.UNIT                      6545
    PAGE 324  ROUTINE DECIDE                         5068
    PAGE 398  EVENT START.BATTLE                     7961
    PAGE 478  PROCESS WITH.DRAW                      1939
    PAGE 608  ROUTINE KV.PRINT                       7837 7837 7837 7837 7837 7837 7837 7837 7837 7838 7838 7838 7838 7838 7838
                                                     7838
    PAGE 636  FUNCTION HE.WLA                        8892 8892
    PAGE 670  ''PROGRAM  OLDER.VERSION                 50   51   52   53   54   55   56
%AUG80_%HWJ
    PAGE 176  ROUTINE EST.COVERAGE                   8039
    PAGE 177                                         8095
    PAGE 188  ROUTINE FINAL.COVERAGE                 8582
    PAGE 189                                         8639
%JUL79_%RGR
    PAGE 521  ROUTINE MAIN2                          4260
%JUN79
    PAGE 176  ROUTINE EST.COVERAGE                   8005
%MAY83_%HWJ
    PAGE 271  ROUTINE BTRY.EFFECTS                   2636
%MAY83_%HWJ%
    PAGE 151  ROUTINE PK.COMPUTE                     7064
%MAY83_%JAF
    PAGE 487  PROCESS ASSESSMENT                     2463
    PAGE 489                                         2577
    PAGE 490                                         2615
    PAGE 491                                         2669
    PAGE 492                                         2722
    PAGE 499  PROCESS SHOOT.OUT                      3142
    PAGE 500                                         3175 3194
    PAGE 501                                         3216 3244
    PAGE 502                                         3297
    PAGE 516  PROCESS HELICOPTER.FIRE                4010
    PAGE 517                                         4089
    PAGE 518                                         4130 4148
%MAY83_%JAF%
    PAGE 309  ROUTINE AD.SHOOT                       4421 4437
%RGR
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES        730  731
    PAGE  18                                          984
    PAGE 180  ROUTINE EST.MIL.WORTH                  8230
    PAGE 325  ROUTINE DEQ.FEBA.SET                   5092
    PAGE 334  ROUTINE FINISH.COMPUTATION             5402
    PAGE 358  EVENT CFR.ACTIVATION                   6351
```

```
    PAGE 428  PROCESS AIR.OBSERVER                    9356 9358 9366 9373
    PAGE 672  ''PROGRAM  OLDER.VERSION                 173  174
    PAGE 677                                           426
%RWF
    PAGE 431  PROCESS AIR.OBSERVER                    9516
    PAGE 432                                          9600
%WARF
    PAGE 272  ROUTINE BTRY.EFFECTS                    2671
%030
    PAGE  14  ''SECTION FOR PERMANENT_ENTITIES         784
    PAGE 673  ''PROGRAM  OLDER.VERSION                 226
%1FEB79
    PAGE 163  ROUTINE BTRY.FM.DEQ                     7541
%10JAN80_%RGR
    PAGE 130  ROUTINE CHECK.DEAD                      6083 6099
%10JUL79_%JLR
    PAGE 336  ROUTINE GET.TERRAIN                     5439 5443
%10OCT80_%RWF
    PAGE 636  FUNCTION HE.WLA                         8912 8916
    PAGE 637                                          8924
%11FEB80_%RGR
    PAGE 334  ROUTINE FINISH.COMPUTATION              5399 5406
%11JAN80_%RGR
    PAGE 130  ROUTINE CHECK.DEAD                      6092 6103
%11JAN81_%RWF
    PAGE 156  ROUTINE AO.DETECTION                    7232
%11SEP79_%RGR
    PAGE 137  ROUTINE CHECK.LIST                      6356
%12MAY83_%HWJ
    PAGE 309  ROUTINE AD.SHOOT                        4420
%14FEB79_%JEN
    PAGE 487  PROCESS ASSESSMENT                      2461 2462
%14FEB80
    PAGE 107  ROUTINE NEW.SEGMENT                     5158
%14FEB80_%RGR
    PAGE 108  ROUTINE NEW.SEGMENT                     5162 5166 5170
    PAGE 355  EVENT ARTY.OCCUPATION                   6244 6245
%14JAN80_%RGR
    PAGE  61  ROUTINE FEBA.INITIAL                    3212
    PAGE 130  ROUTINE CHECK.DEAD                      6106
%14MAR80_%RGR
    PAGE 371  EVENT FEBA.SORTIE                       6785
%15JAN80_%RGR
    PAGE 212  ROUTINE PIR.DETECTION                   9723
    PAGE 229  ROUTINE RPV.DETECTION                    536
%15MAR
    PAGE  72  ROUTINE ORIENTATION                     3729 3730
%15MAR79
    PAGE  72  ROUTINE ORIENTATION                     3728
%15MAR79_%JN
    PAGE  71  ROUTINE ORIENTATION                     3634
%15MAY83_%HWJ
    PAGE 536  ROUTINE UNIT.INPUT                      4882
%16DEC80_%RWF
    PAGE 428  PROCESS AIR.OBSERVER                    9379 9380 9381 9383 9384 9385 9387
    PAGE 429                                          9394 9395 9396 9409 9410
```

```
%17DEC79_%RGR
    PAGE  61  ROUTINE FEBA.INITIAL                    3200
%18MAR80_%RGR
    PAGE 431  PROCESS AIR.OBSERVER                    9512
%18OCT79_%RGR
    PAGE  62  ROUTINE FEBA.INITIAL                    3243 3245
%19DEC80_%RWF
    PAGE 371  EVENT FEBA.SORTIE                       6805 6807
    PAGE 372                                          6839 6848 6868
%19MAR80_%RGR
    PAGE 115  ROUTINE PROX.CHECK                      5502
    PAGE 154  ROUTINE AO.DETECTION                    7097
%2JAN81_%RWF
    PAGE 428  PROCESS AIR.OBSERVER                    9347
    PAGE 431                                          9508 9543 9559
    PAGE 432                                          9574
%20MAR80
    PAGE 325  ROUTINE DEQ.FEBA.SET                    5092
%21JAN80_%RGR
    PAGE 213  ROUTINE PIR.DETECTION                   9789
    PAGE 230  ROUTINE RPV.DETECTION                    621
%21MAY80_%JLR
    PAGE 331  ROUTINE ENQ.FEBA.SET                    5322 5324
%21NOV80_%RWF
    PAGE 429  PROCESS AIR.OBSERVER                    9419
    PAGE 433                                          9656
%21OCT80_%RWF
    PAGE 429  PROCESS AIR.OBSERVER                    9421
%22DEC80
    PAGE 372  EVENT FEBA.SORTIE                       6837
%22DEC80_%RWF
    PAGE 371  EVENT FEBA.SORTIE                       6788 6789 6791 6792 6793 6794 6795 6796 6797 6798 6799 6800 6802 6804 6831
                                                      6832
    PAGE 372                                          6834 6835 6836 6838 6840 6841 6842 6843 6844 6845 6847
    PAGE 429  PROCESS AIR.OBSERVER                    9388
%22FEB79_%JEN
    PAGE 656  ROUTINE AR.DETECTION                    9398 9420
%22JAN80_%RGR
    PAGE 270  ROUTINE BTRY.EFFECTS                    2582
%23DEC80_%RWF
    PAGE 429  PROCESS AIR.OBSERVER                    9408
%24NOV80_%RWF
    PAGE 429  PROCESS AIR.OBSERVER                    9415
%26MAR78_%LM
    PAGE  71  ROUTINE ORIENTATION                     3648
%26OCT79_%PGR
    PAGE  13  ''SECTION FOR PERMANENT_ENTITIES         729
    PAGE 130  ROUTINE CHECK.DEAD                      6081 6090
    PAGE 145  ROUTINE FIN.BATTLE                      6716
    PAGE 672  ''PROGRAM  OLDER.VERSION                 172
%26OCT79_?%RGR?
    PAGE 324  ROUTINE DECIDE                          5056
%28OCT80_%HWJ
    PAGE 636  FUNCTION HE.WLA                         8876
    PAGE 638  FUNCTION ICM.WLA                        8999
```

```
%29DEC80_%RWF
    PAGE 428  PROCESS AIR.OBSERVER                    9371
    PAGE 429                                          9441
    PAGE 430                                          9459 9460 9461
    PAGE 431                                          9514 9517 9534 9535 9536 9558
    PAGE 432                                          9594 9595 9597 9619
    PAGE 433                                          9635
%30AUG_%RGR
    PAGE 137  ROUTINE CHECK.LIST                      6332 6358
%30DEC80_%RWF
    PAGE 154  ROUTINE AO.DETECTION                    7084 7100
    PAGE 157                                          7288
    PAGE 432  PROCESS AIR.OBSERVER                    9596
%31DEC80
    PAGE 432  PROCESS AIR.OBSERVER                    9589
%31DEC80_%RWF
    PAGE 428  PROCESS AIR.OBSERVER                    9386
    PAGE 431                                          9506 9507 9515 9519 9521 9522 9523 9524 9526 9527 9528
    PAGE 432                                          9568 9570 9571 9572 9573 9576 9577 9578 9584 9591 9598 9599 9604 9605 9606
                                                      9607 9609 9610 9611 9612
    PAGE 433                                          9622 9623 9624 9625 9627 9628 9629 9630 9640 9641 9642 9643 9645 9646 9647
                                                      9648 9659 9660 9661 9662 9664 9665 9666 9667 9672
%31DEC_%RWF
    PAGE 431  PROCESS AIR.OBSERVER                    9537 9540 9541 9542 9545 9546 9547
%31JAN79_GLM
    PAGE 163  ROUTINE BTRY.FM.DEQ                     7528
%31JAN79_%GLM
    PAGE 164  ROUTINE BTRY.FM.ENQ                     7544
    PAGE 168  ROUTINE CHK.COMP.TR                     7700
    PAGE 332  ROUTINE FDC.TR.DEQ                      5342 5349
%31MAR80_%RGR
    PAGE 412  EVENT ACT.ATK                           8624
%5DEC79_%RGR
    PAGE  18  ''SECTION FOR PERMANENT_ENTITIES         992  993
    PAGE 677  ''PROGRAM  OLDER.VERSION                 434  435
%5FEB80_%RGR
    PAGE 638  FUNCTION ICM.WLA                        8984
%5JAN81_%RWF
    PAGE 429  PROCESS AIR.OBSERVER                    9430 9431 9432 9434 9435 9436 9437 9438 9439
    PAGE 430                                          9453 9454 9455 9456 9462
%5MAR80_%RGR
    PAGE 428  PROCESS AIR.OBSERVER                    9357
%5MAY80_%JLR
    PAGE  61  ROUTINE FEBA.INITIAL                    3227 3235
%5NOV79_%RGR
    PAGE  58  ROUTINE CREATE.FORCE                    3135
%6JAN81
    PAGE 430  PROCESS AIR.OBSERVER                    9489 9491
%6JAN81_%RWF
    PAGE 430  PROCESS AIR.OBSERVER                    9490 9492
    PAGE 432                                          9601 9617
%6MAR80_%RGR
    PAGE 637  FUNCTION HE.WLA                         8941
%7MAR80_%RGR
    PAGE  61  ROUTINE FEBA.INITIAL                    3219
    PAGE  62                                          3252
```

```
    PAGE 100  ROUTINE LOCATE.SECTOR                  4862
    PAGE 371  EVENT FEBA.SORTIE                      6810
    PAGE 372                                         6851
%8APR80_%RGR
    PAGE 138  ROUTINE CHECK.PROX                     6390
%8DEC80_%RWF
    PAGE  33  ''SECTION FOR TEMPORARY_ENTITIES       1847
    PAGE 692  ''PROGRAM  OLDER.VERSION               1286
%8FEB79_%GLM
    PAGE 264  ROUTINE BTRY.EFFECTS                   2232
%8FEB80_%RGR
    PAGE 107  ROUTINE NEW.SEGMENT                    5124 5127
%8JAN81_%RWF
    PAGE 429  PROCESS AIR.OBSERVER                   9389 9391 9392 9393 9399 9401 9402 9403
    PAGE 431                                         9518
%8MAY80_%JLR
    PAGE  61  ROUTINE FEBA.INITIAL                   3220
%9JAN80_%RGR
    PAGE 331  ROUTINE ENQ.FEBA.SET                   5313
%9JAN81_%RWF
    PAGE 428  PROCESS AIR.OBSERVER                   9349
    PAGE 432                                         9563 9564 9565 9566 9567
%$#
    PAGE 183  ROUTINE FA.BN.ASGN                     8390
    PAGE 480  PROCESS FIRE.MISSION                   2083
    PAGE 481                                         2089 2099
```

```
CHG\01
    PAGE   2  PROGRAM REVISIONS                    74
    PAGE 419  PROCESS AC.ATK.TGT                 8851
    PAGE 428  PROCESS AIR.OBSERVER               9335
    PAGE 435  PROCESS ARTY.ASSESS                9690
    PAGE 438  PROCESS FORWARD.OBSERVER           9858
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE       1359
    PAGE 470  PROCESS TARGET.REPORT              1489
    PAGE 478  PROCESS WITH.DRAW                  1920
    PAGE 480  PROCESS FIRE.MISSION               2042
    PAGE 487  PROCESS ASSESSMENT                 2423
    PAGE 493  PROCESS SHOOT.OUT                  2753
    PAGE 504  PROCESS CAS.MISSION                3353
    PAGE 510  PROCESS HELICOPTER.FIRE            3670
CHG\02
    PAGE   2  PROGRAM REVISIONS                    77
    PAGE  90  ROUTINE FA.BN.MOVEMENT             4459
CHG\03
    PAGE   2  PROGRAM REVISIONS                    80
    PAGE 238  ROUTINE SWITCH.FO                   934  949
CHG\04
    PAGE   2  PROGRAM REVISIONS                    84
    PAGE 500  PROCESS SHOOT.OUT                  3161
CHG\05
    PAGE   2  PROGRAM REVISIONS                    87
    PAGE 356  EVENT BTL.ENDED                    6297
CHG\06
    PAGE   2  PROGRAM REVISIONS                    90
    PAGE 182  ROUTINE FA.BN.ASGN                 8318
CHG\07
    PAGE   2  PROGRAM REVISIONS                    94
    PAGE 486  PROCESS FIRE.MISSION               2387
CHG\08
    PAGE   2  PROGRAM REVISIONS                    98
    PAGE 254  ROUTINE FO.DETECTION               1689
CHG\09
    PAGE   2  PROGRAM REVISIONS                   102
    PAGE 178  ROUTINE EST.COVERAGE               8149
CHG\10
    PAGE   2  PROGRAM REVISIONS                   105
    PAGE 193  ROUTINE FINAL.COVERAGE             8854 8869
CHG\11
    PAGE   2  PROGRAM REVISIONS                   109
    PAGE  56  ROUTINE MAIN1                      3043
    PAGE 647  ROUTINE OPEN.INPUT.OUTPUT.FILES    9182 9187
CHG\12
    PAGE   2  PROGRAM REVISIONS                   113
    PAGE 481  PROCESS FIRE.MISSION               2112
    PAGE 485                                     2363
CHG\13
    PAGE   2  PROGRAM REVISIONS                   117
    PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION      9058
CHG\14
    PAGE   3  PROGRAM REVISIONS                   124
    PAGE 219  ROUTINE REQUEST.ILLUM               78
```

```
CHG\15
    PAGE   3  PROGRAM REVISIONS                        127
    PAGE 502  PROCESS SHOOT.OUT                       3316
CHG\16
    PAGE   3  PROGRAM REVISIONS                        130
    PAGE  91  ROUTINE FA.BN.MOVEMENT                  4538
CHG\17
    PAGE   3  PROGRAM REVISIONS                        133
    PAGE 572  ROUTINE FARRP.INPUT                     6334
    PAGE 582  ROUTINE TACAIR.INPUT                    6746
CHG\18
    PAGE   3  PROGRAM REVISIONS                        137
    PAGE 220  ROUTINE REQUEST.ILLUM                    111
CHG\19
    PAGE   3  PROGRAM REVISIONS                        141
    PAGE 184  ROUTINE FA.BN.ASGN                      8436 8457
    PAGE 185                                          8489
    PAGE 210  ROUTINE PGM.MSN.ASGN                    9658
    PAGE 274  ROUTINE CLEAN.UP.FIRE.MISSIONS          2736
CHG\20
    PAGE   3  PROGRAM REVISIONS                        145
    PAGE 220  ROUTINE REQUEST.ILLUM                    113
CHG\21
    PAGE   3  PROGRAM REVISIONS                        149
    PAGE 239  ROUTINE TARGET.ANALYSIS                  974  989
    PAGE 241                                          1096
    PAGE 471  PROCESS TARGET.REPORT                   1541 1542 1566
    PAGE 473                                          1712
    PAGE 576  ROUTINE ILLUM.INPUT                     6500
CHG\22
    PAGE   3  PROGRAM REVISIONS                        153
    PAGE 222  ROUTINE REQUEST.SMOKE                    184
    PAGE 224                                           319
    PAGE 580  ROUTINE SMOKE.INPUT                     6674
CHG\23
    PAGE   3  PROGRAM REVISIONS                        157
    PAGE 216  ROUTINE REQUEST.FASCAM                  9896
    PAGE 226  ROUTINE REQUEST.WD.FASCAM                402
    PAGE 577  ROUTINE MINE.INPUT                      6547
CHG\24
    PAGE   3  PROGRAM REVISIONS                        161
    PAGE 173  ROUTINE DUST.EFFECTS                    7895
    PAGE 193  ROUTINE FINAL.COVERAGE                  8884
CHG\25
    PAGE   3  PROGRAM REVISIONS                        165
    PAGE 258  ROUTINE FO.DETECTION                    1910
CHG\26
    PAGE   3  PROGRAM REVISIONS                        168
    PAGE 487  PROCESS ASSESSMENT                      2421
    PAGE 491                                          2687 2688
CHG\27
    PAGE   3  PROGRAM REVISIONS                        172
    PAGE 176  ROUTINE EST.COVERAGE                    8018
    PAGE 179                                          8182
CHG\28
    PAGE   3  PROGRAM REVISIONS                        175
```

```
    PAGE 160  ROUTINE ATTRIT.SENSOR                    7379 7383 7409 7413
CHG\29
    PAGE   4  PROGRAM REVISIONS                         181
    PAGE 239  ROUTINE TARGET.ANALYSIS                   990 1007
CHG\30
    PAGE   4  PROGRAM REVISIONS                         185
    PAGE 634  FUNCTION FEBA.BAND                       8797 8815
CHG\31
    PAGE   4  PROGRAM REVISIONS                         189
    PAGE 609  ROUTINE KV.SCOREBOARD                    7856
CHG\32
    PAGE   4  PROGRAM REVISIONS                         197
    PAGE 360  EVENT CFR.ON                             6434
CHG\33
    PAGE   4  PROGRAM REVISIONS                         207
    PAGE 470  PROCESS TARGET.REPORT                    1514
HUGH\CHECKOUT
    PAGE 502  PROCESS SHOOT.OUT                        3317
\ALPHA
    PAGE 581  ROUTINE VIS.INPUT                        6718
\ARG_MODE_ERROR
    PAGE 173  ROUTINE DUST.EFFECTS                     7895
\ARG_NO
    PAGE 356  EVENT BTL.ENDED                          6297
\CLEANUP
    PAGE 114  ROUTINE PREP.WITHDRAW                    5489
    PAGE 130  ROUTINE CHECK.DEAD                       6088
    PAGE 630  FUNCTION COLLISION                       8691
    PAGE 640  ROUTINE EXPONENTIAL.F                    9026
    PAGE 641  ROUTINE NORMAL.F                         9039
    PAGE 642  ROUTINE WEIBULL.F                        9062
    PAGE 652  ROUTINE GAMMA.F                          9236
\DEBUG
    PAGE 160  ROUTINE ATTRIT.SENSOR                    7379 7383 7409 7413
    PAGE 176  ROUTINE EST.COVERAGE                     8019
    PAGE 179                                           8182
    PAGE 184  ROUTINE FA.BN.ASGN                       8436 8457
    PAGE 185                                           8489
    PAGE 210  ROUTINE PGM.MSN.ASGN                     9658
    PAGE 239  ROUTINE TARGET.ANALYSIS                   974  989  990 1007
    PAGE 241                                           1096
    PAGE 271  ROUTINE BTRY.EFFECTS                     2646
    PAGE 360  EVENT CFR.ON                             6415 6434
    PAGE 441  PROCESS FORWARD.OBSERVER                   52
    PAGE 471  PROCESS TARGET.REPORT                    1540 1542 1566
    PAGE 473                                           1712
    PAGE 477                                           1906
    PAGE 487  PROCESS ASSESSMENT                       2421
    PAGE 491                                           2687 2688
    PAGE 536  ROUTINE UNIT.INPUT                       4884
    PAGE 609  ROUTINE KV.SCOREBOARD                    7856
    PAGE 634  FUNCTION FEBA.BAND                       8797 8815
\DF06
    PAGE 137  ROUTINE CHECK.LIST                       6332
\DYN_ANAL
    PAGE  41  ''SECTION FOR EVENTS                     2338
```

PAGE 50	''SECTION FOR DEFINITIONS	2844 2845
PAGE 53	''SECTION FOR SUBSTITUTIONS	3012
PAGE 55	''PROGRAM'' MAIN	3028 3029 3030
PAGE 56	ROUTINE MAIN1	3037
PAGE 57	ROUTINE MAIN3	3070
PAGE 58	ROUTINE CREATE.FORCE	3095
PAGE 60	ROUTINE CREATE.TEAMS	3161
PAGE 61	ROUTINE FEBA.INITIAL	3185
PAGE 63	ROUTINE FILE.FD.SCHD	3261
PAGE 65	ROUTINE FILE.KAD.SENSOR	3362
PAGE 66	ROUTINE FORM.TF.LIST	3417
PAGE 67	ROUTINE GENERAL.BATTLE	3452
PAGE 71	ROUTINE ORIENTATION	3645
PAGE 75	ROUTINE UNIT.ASSIGNMENT	3831
PAGE 77	ROUTINE ADJUST	3885
PAGE 80	ROUTINE BLOCK.LOS	4010
PAGE 83	ROUTINE CHANGE.LOC	4127
PAGE 87	ROUTINE END.MOVE	4311
PAGE 89	ROUTINE FA.BN.MOVEMENT	4398
PAGE 93	ROUTINE INITIAL.DETECT	4590
PAGE 94	ROUTINE INITIAL.MOVE	4616
PAGE 95	ROUTINE INIT.REINF	4639
PAGE 96	ROUTINE LINE.OF.SIGHT	4668
PAGE 98	ROUTINE LOCATE.SEARCH.AREA	4776
PAGE 100	ROUTINE LOCATE.SECTOR	4835
PAGE 101	ROUTINE LOS.CHECK	4875
PAGE 103	ROUTINE MINE.DELAY	4942
PAGE 106	ROUTINE MIN.MOVE	5064
PAGE 107	ROUTINE NEW.SEGMENT	5108
PAGE 110	ROUTINE POSITION	5271
PAGE 111	ROUTINE PRED.POS	5307
PAGE 112	ROUTINE PREPARE.LIST	5352
PAGE 114	ROUTINE PREP.WITHDRAW	5468
PAGE 115	ROUTINE PROX.CHECK	5501
PAGE 117	ROUTINE PROX.POS	5569
PAGE 118	ROUTINE REIN.ARRIVE	5602
PAGE 119	ROUTINE RESET.FEBA.SECTOR	5652
PAGE 120	ROUTINE SEARCH	5715
PAGE 122	ROUTINE SEGMENT.ADJUST	5778
PAGE 123	ROUTINE TIME.TO.DETECT	5799
PAGE 125	ROUTINE WHAT.NEXT	5878
PAGE 128	ROUTINE BTL.CHECK	5990
PAGE 130	ROUTINE CHECK.DEAD	6070
PAGE 132	ROUTINE CHECK.ENGAGEMENT	6135
PAGE 133	ROUTINE CHECK.FORCE	6157
PAGE 134	ROUTINE CHECK.FOR.MINES	6221
PAGE 137	ROUTINE CHECK.LIST	6334
PAGE 138	ROUTINE CHECK.PROX	6376
PAGE 141	ROUTINE CHECK.STREN	6505
PAGE 142	ROUTINE DEAD.UNIT	6525
PAGE 145	ROUTINE FIN.BATTLE	6695
PAGE 146	ROUTINE INTER.BATTLE	6741
PAGE 149	ROUTINE PK.COMPUTE	6904
PAGE 154	ROUTINE AO.DETECTION	7087
PAGE 159	ROUTINE ATTRIT.SENSOR	7316
PAGE 163	ROUTINE BTRY.FM.DEQ	7514

PAGE 164 ROUTINE BTRY.FM.ENQ 7545
PAGE 165 ROUTINE CFR.DEGRADE 7579
PAGE 166 ROUTINE CFR.DETECTION 7622
PAGE 168 ROUTINE CHK.COMP.TR 7702
PAGE 169 ROUTINE CHK.FD.TR 7730
PAGE 170 ROUTINE COMBINE.TRS 7769
PAGE 171 ROUTINE COMPARE.TRS 7805
PAGE 172 ROUTINE COPY 7848
PAGE 173 ROUTINE DUST.EFFECTS 7890
PAGE 176 ROUTINE EST.COVERAGE 8012
PAGE 180 ROUTINE EST.MIL.WORTH 8224
PAGE 181 ROUTINE FASCAM.COMPUTATION 8249
PAGE 182 ROUTINE FA.BN.ASGN 8292
PAGE 187 ROUTINE FD.EFFECTS.REQ 8528
PAGE 188 ROUTINE FINAL.COVERAGE 8551
PAGE 194 ROUTINE FIND.START.TIME 8898
PAGE 196 ROUTINE HE.OR.ICM.COMPUTATION 8996
PAGE 199 ROUTINE ILLUM.COMPUTATION 9137
PAGE 200 ROUTINE ILLUM.EFFECTS 9188
PAGE 203 ROUTINE MARGINAL.EFFECTS.ADJ 9308
PAGE 205 ROUTINE NOISE.DEGRADE 9386
PAGE 206 ROUTINE PDB.DETECTION 9434
PAGE 208 ROUTINE PGM.MSN.ASGN 9494
PAGE 212 ROUTINE PIR.DETECTION 9705
PAGE 214 ROUTINE REM.EFFECTS.COMPUTATION 9799
PAGE 215 ROUTINE REQUEST.DEF.FASCAM 9836
PAGE 216 ROUTINE REQUEST.FASCAM 9898
PAGE 218 ROUTINE REQUEST.ILLUM 9990
PAGE 222 ROUTINE REQUEST.SMOKE 181
PAGE 226 ROUTINE REQUEST.WD.FASCAM 399
PAGE 229 ROUTINE RPV.DETECTION 517
PAGE 231 ROUTINE SIZE.ESTIMATE 628
PAGE 233 ROUTINE SMOKE.COMPUTATION 691
PAGE 234 ROUTINE SMOKE.EFFECTS 742
PAGE 238 ROUTINE SWITCH.FO 918
PAGE 239 ROUTINE TARGET.ANALYSIS 976
PAGE 242 ROUTINE UNIT.ENVIR 1119
PAGE 246 ROUTINE VOLLEY 1311
PAGE 248 ROUTINE WEIGHTED.VOLLEYS 1378
PAGE 250 ROUTINE MINE.EFFECTS 1471
PAGE 254 ROUTINE FO.DETECTION 1679
PAGE 260 ROUTINE BTRY.EFFECTS 1968
PAGE 276 ROUTINE AC.BOMB.EFFECTS 2793
PAGE 280 ROUTINE AC.DF.EFFECTS 3004
PAGE 283 ROUTINE CAS.EVAL 3171
PAGE 285 ROUTINE CHECK.CAS.CONSTRAINTS 3284
PAGE 289 ROUTINE EMPLOY.HELICOPTERS 3478
PAGE 293 ROUTINE END.CAS.MISSION 3681
PAGE 297 ROUTINE FARRP.CHECK 3863
PAGE 298 ROUTINE HC.COMPUTE.TIMES 3923
PAGE 300 ROUTINE HC.DISENGAGE 4009
PAGE 302 ROUTINE HEL.RANGE.COMPUTE 4126
PAGE 304 ROUTINE REPLACE.HC 4192
PAGE 305 ROUTINE UNIT.PRIORITY 4241
PAGE 307 ROUTINE AD.SHOOT 4309
PAGE 311 ROUTINE INTER.HELO 4518

PAGE 314 ROUTINE FLIGHT.PATH 4634
PAGE 321 ROUTINE COMPUTE.D 4949
PAGE 322 ROUTINE COMPUTE.WD 4971
PAGE 323 ROUTINE CONTRAST.TO.FREQ 5005
PAGE 324 ROUTINE DECIDE 5039
PAGE 325 ROUTINE DEQ.FEBA.SET 5082
PAGE 326 ROUTINE DESTROY.ORD 5119
PAGE 327 ROUTINE DQ.CMSN.QUEUE 5144
PAGE 328 ROUTINE EMPTY 5171
PAGE 331 ROUTINE ENQ.FEBA.SET 5304
PAGE 332 ROUTINE FDC.TR.DEQ 5343
PAGE 333 ROUTINE FDC.TR.ENQ 5362
PAGE 334 ROUTINE FINISH.COMPUTATION 5383
PAGE 335 ROUTINE FRAC.COMPUTE 5414
PAGE 336 ROUTINE GET.TERRAIN 5436
PAGE 337 ROUTINE HC.EMPTY 5462
PAGE 340 ROUTINE PROB.INF 5611
PAGE 341 ROUTINE PROB.TIME 5643
PAGE 342 ROUTINE RANGE.COMPUTE 5666
PAGE 343 ROUTINE SEARCH.COVERAGE 5684
PAGE 344 ROUTINE TEMPERATURE.ATTENUATION 5740
PAGE 345 ROUTINE TERM.CHECK 5769
PAGE 347 EVENT ACT.REINF 5809
PAGE 349 EVENT AD.ENGAGEMENT 5902
PAGE 355 EVENT ARTY.OCCUPATION 6230
PAGE 356 EVENT BTL.ENDED 6255
PAGE 358 EVENT CFR.ACTIVATION 6345
PAGE 359 EVENT CFR.OFF 6385
PAGE 360 EVENT CFR.ON 6410
PAGE 362 EVENT CFR.OPERATOR 6482
PAGE 365 EVENT CHANGE.LITE 6597
PAGE 366 EVENT CHANGE.WEATHER 6620
PAGE 367 EVENT DQ.OLD.SORTIE.QUEUE 6638
PAGE 368 EVENT END.SIMULATION 6663
PAGE 369 EVENT ENGAGEMENT 6683
PAGE 371 EVENT FEBA.SORTIE 6781
PAGE 373 EVENT GET.NX.ORD 6882
PAGE 375 EVENT HC.DEPART.BATTLE 6994
PAGE 376 EVENT HELO.ENGAGEMENT 7037
PAGE 380 EVENT INIT.PREPLAN.CAS 7235
PAGE 381 EVENT MOVE 7271
PAGE 382 EVENT OFF.LINE.ATTRITION 7281
PAGE 386 EVENT PDB.ACTIVATION 7472
PAGE 387 EVENT PDB.OPERATOR 7508
PAGE 389 EVENT POSITION.REPORT 7570
PAGE 390 EVENT SCHEDULE.ARTY.MOVEMENT 7581
PAGE 391 EVENT SEND.TEAM 7601
PAGE 392 EVENT SET.DEBUG 7625
PAGE 393 EVENT START.ARTY.MOVEMENT 7660
PAGE 395 EVENT START.BATTLE 7752
PAGE 403 EVENT START.MOVE 8185
PAGE 407 EVENT STOP.ARTY.MOVEMENT 8375
PAGE 408 EVENT UPDATE.LOC 8413
PAGE 412 EVENT ACT.ATK 8606
PAGE 414 EVENT ACT.DEF 8719
PAGE 415 EVENT ACT.MOVCOR 8741

PAGE 416	EVENT ACT.MOVDIS	8771
PAGE 417	EVENT DYNAMIC.ANALYSIS.REPORT	8805
PAGE 419	PROCESS AC.ATK.TGT	8849
PAGE 428	PROCESS AIR.OBSERVER	9333
PAGE 435	PROCESS ARTY.ASSESS	9688
PAGE 438	PROCESS FORWARD.OBSERVER	9856
PAGE 443	PROCESS HC.ARRIVE.BATTLE	98
PAGE 450	PROCESS HC.RETURN.FARRP	465
PAGE 454	PROCESS HEL.TARGET.ACQUISITION	697
PAGE 463	PROCESS HOW.REPAIR	1172
PAGE 464	PROCESS MINE.ASSESS	1222
PAGE 467	PROCESS REMOTE.PILOT.VEHICLE	1357
PAGE 470	PROCESS TARGET.REPORT	1484
PAGE 478	PROCESS WITH.DRAW	1918
PAGE 480	PROCESS FIRE.MISSION	2029
PAGE 487	PROCESS ASSESSMENT	2419
PAGE 493	PROCESS SHOOT.OUT	2751
PAGE 504	PROCESS CAS.MISSION	3351
PAGE 510	PROCESS HELICOPTER.FIRE	3662
PAGE 520	ROUTINE MAIN2	4185
PAGE 523	ROUTINE SYS.INPUT	4318
PAGE 525	ROUTINE PK.INPUT	4382
PAGE 527	ROUTINE CAT.TU.INPUT	4452
PAGE 529	ROUTINE KV.INPUT	4525
PAGE 530	ROUTINE EQ.TE.INPUT	4575
PAGE 532	ROUTINE TYPE.WEAPON.INPUT	4640
PAGE 533	ROUTINE UNIT.INPUT	4682
PAGE 538	ROUTINE MFO.INPUT	4932
PAGE 539	ROUTINE READ.ORDERS	4985
PAGE 542	ROUTINE ORD.DEF	5118
PAGE 543	ROUTINE ORD.ATK	5143
PAGE 544	ROUTINE ORD.REINF	5162
PAGE 545	ROUTINE ORD.MOVDIS	5183
PAGE 546	ROUTINE ORD.MOVCOR	5208
PAGE 547	ROUTINE P.E.M.INPUT	5257
PAGE 548	ROUTINE TB.INPUT	5309
PAGE 550	ROUTINE BTRY.INPUT	5389
PAGE 553	ROUTINE FBN.FD.INPUT	5541
PAGE 555	ROUTINE MUNS.INPUT	5604
PAGE 557	ROUTINE SUBM.INPUT	5715
PAGE 558	ROUTINE HE.LA.INPUT	5752
PAGE 560	ROUTINE RUL.EN.INPUT	5822
PAGE 561	ROUTINE ST.INPUT	5859
PAGE 562	ROUTINE MCFR.INPUT	5880
PAGE 563	ROUTINE MPDB.INPUT	5917
PAGE 564	ROUTINE MAO.INPUT	5955
PAGE 565	ROUTINE SENSOR.INPUT	6013
PAGE 569	ROUTINE TBF.INPUT	6204
PAGE 570	ROUTINE DECISION.INPUT	6262
PAGE 571	ROUTINE TT.FACTORS.INPUT	6284
PAGE 572	ROUTINE FARRP.INPUT	6329
PAGE 575	ROUTINE PGM.INPUT	6451
PAGE 576	ROUTINE ILLUM.INPUT	6496
PAGE 577	ROUTINE MINE.INPUT	6535
PAGE 580	ROUTINE SMOKE.INPUT	6670
PAGE 581	ROUTINE VIS.INPUT	6713

PAGE 582 ROUTINE TACAIR.INPUT 6740
PAGE 587 ROUTINE MADS.INPUT 6980
PAGE 588 ROUTINE AC.MUNS.INPUT 7013
PAGE 590 ROUTINE TR.INPUT 7077
PAGE 592 ROUTINE AMMO.RPT 7121
PAGE 595 ROUTINE ANALYSIS.OUTPUT 7241
PAGE 599 ROUTINE BETWEEN.ROUTINE 7430
PAGE 604 ROUTINE ERROR.STOP 7667
PAGE 605 ROUTINE HEADING 7681
PAGE 606 ROUTINE KV.PRINT 7690
PAGE 609 ROUTINE KV.SCOREBOARD 7843
PAGE 611 ROUTINE OUTPUT.ATTRITION 7931
PAGE 613 ROUTINE FOR POSITION.OUT 7995
PAGE 614 ROUTINE SNAP2 8022
PAGE 618 ROUTINE SNAP.R 8203
PAGE 619 8311
PAGE 621 ROUTINE TACAIR.DATA.REPORT 8335
PAGE 624 ROUTINE OUTPUT.EXPENDITURES 8490
PAGE 628 FUNCTION ACT.RANGE 8633
PAGE 629 FUNCTION BTRY.AVAILABLE 8647
PAGE 630 FUNCTION COLLISION 8681
PAGE 631 FUNCTION COMBINATIONS 8717
PAGE 632 FUNCTION EST.RANGE 8747
PAGE 633 FUNCTION EST.TR.RANGE 8762
PAGE 634 FUNCTION FEBA.BAND 8777
PAGE 636 FUNCTION HE.WLA 8860
PAGE 638 FUNCTION ICM.WLA 8960
PAGE 640 ROUTINE EXPONENTIAL.F 9027
PAGE 641 ROUTINE NORMAL.F 9040
PAGE 642 ROUTINE WEIBULL.F 9063
PAGE 644 ROUTINE ANGLE.COMPUTE 9085
PAGE 645 ROUTINE LINE.CIRCLE 9132
PAGE 646 ROUTINE MRT.TO.FREQ 9162
PAGE 648 ROUTINE PERFORM.INSTRUMENTATION 9192
PAGE 652 ROUTINE GAMMA.F 9238
PAGE 653 PROCESS AIRBORNE.RADAR 9264
PAGE 656 ROUTINE AR.DETECTION 9386
PAGE 657 FUNCTION AR.PROB.DETECT 9445
PAGE 658 PROCESS PHOTO.IR.FLIGHT 9475
PAGE 660 FUNCTION STAY.TIME 9594
PAGE 661 ROUTINE JOHNSON.CRITERIA 9619
PAGE 662 ROUTINE PROXIMITY.REQ 9633
PAGE 663 ROUTINE TIME.REQ 9647
PAGE 714 ROUTINE PLAT.COUNT 2515
\EFFECT_OF_ZERO_SUB_CORRECTION?
PAGE 193 ROUTINE FINAL.COVERAGE 8868
\EQUIVALENCING
PAGE 13 ''SECTION FOR PERMANENT_ENTITIES 713
PAGE 18 977
PAGE 33 ''SECTION FOR TEMPORARY_ENTITIES 1878
\ERROR_CORRECTION
PAGE 258 ROUTINE FO.DETECTION 1910
PAGE 457 PROCESS HEL.TARGET.ACQUISITION 874
\EXPLICIT_DEFINE
PAGE 218 ROUTINE REQUEST.ILLUM 9993

++++++

PAGE 533 ROUTINE UNIT.INPUT 4731
PAGE 548 ROUTINE TB.INPUT 5355
PAGE 556 ROUTINE MUNS.INPUT 5705
PAGE 561 ROUTINE ST.INPUT 5873
PAGE 562 ROUTINE MCFR.INPUT 5899
PAGE 576 ROUTINE ILLUM.INPUT 6516
PAGE 577 ROUTINE MINE.INPUT 6558
PAGE 580 ROUTINE SMOKE.INPUT 6689
PAGE 582 ROUTINE TACAIR.INPUT 6772
PAGE 587 ROUTINE MADS.INPUT 7002
PAGE 628 FUNCTION ACT.RANGE 8641
PAGE 632 FUNCTION EST.RANGE 8754
PAGE 633 FUNCTION EST.TR.RANGE 8767
PAGE 638 FUNCTION ICM.WLA 8987
PAGE 644 ROUTINE ANGLE.COMPUTE 9105
PAGE 645 ROUTINE LINE.CIRCLE 9140
PAGE 653 PROCESS AIRBORNE.RADAR 9295
PAGE 656 ROUTINE AR.DETECTION 9431
PAGE 658 PROCESS PHOTO.IR.FLIGHT 9513
PAGE 662 ROUTINE PROXIMITY.REQ 9636
PAGE 709 ''PROGRAM OLDER.VERSION 2307 2308
\REDUNDANT_CODE
PAGE 634 FUNCTION FEBA.BAND 8775
\REDUNDANT_DESTROY
PAGE 486 PROCESS FIRE.MISSION 2387
\REMOVE?
PAGE 182 ROUTINE FA.BN.ASGN 8319
PAGE 494 PROCESS SHOOT.OUT 2812
\REPLACE?
PAGE 661 ROUTINE JOHNSON.CRITERIA 9622
\REQUIRED?
PAGE 652 ROUTINE GAMMA.F 9236
PAGE 662 ROUTINE PROXIMITY.REQ 9626
PAGE 663 ROUTINE TIME.REQ 9640
\ROUTINE.NAME?
PAGE 660 FUNCTION STAY.TIME 9600
\TEXT
PAGE 47 ''SECTION FOR DEFINITIONS 2658 2707
PAGE 50 2834 2839
PAGE 95 ROUTINE INIT.REINF 4647
PAGE 176 ROUTINE EST.COVERAGE 8018
PAGE 188 ROUTINE FINAL.COVERAGE 8561
PAGE 206 ROUTINE PDB.DETECTION 9437
PAGE 246 ROUTINE VOLLEY 1314
PAGE 248 ROUTINE WEIGHTED.VOLLEYS 1386
PAGE 250 ROUTINE MINE.EFFECTS 1477
PAGE 252 1587 1597
PAGE 260 ROUTINE BTRY.EFFECTS 1984 2012
PAGE 264 2243
PAGE 265 2260 2265 2309
PAGE 266 2318 2327
PAGE 271 2654 2655
PAGE 272 2665 2666 2678 2696 2697
PAGE 293 ROUTINE END.CAS.MISSION 3685
PAGE 295 3823 3840 3842
PAGE 307 ROUTINE AD.SHOOT 4313

```
    PAGE 309                                    4430 4446
    PAGE 314  ROUTINE FLIGHT.PATH               4639
    PAGE 328  ROUTINE EMPTY                     5177
    PAGE 330                                    5293
    PAGE 349  EVENT AD.ENGAGEMENT               5907
    PAGE 358  EVENT CFR.ACTIVATION              6347
    PAGE 382  EVENT OFF.LINE.ATTRITION          7283
    PAGE 386  EVENT PDB.ACTIVATION              7475
    PAGE 392  EVENT SET.DEBUG                   7641 7645
    PAGE 395  EVENT START.BATTLE                7755 7788
    PAGE 403  EVENT START.MOVE                  8194
    PAGE 408  EVENT UPDATE.LOC                  8428
    PAGE 484  PROCESS FIRE.MISSION              2291
    PAGE 488  PROCESS ASSESSMENT                2472
    PAGE 490                                    2586 2624
    PAGE 491                                    2678
    PAGE 492                                    2731
    PAGE 499  PROCESS SHOOT.OUT                 3151
    PAGE 500                                    3184 3203
    PAGE 501                                    3225 3253
    PAGE 502                                    3306
    PAGE 510  PROCESS HELICOPTER.FIRE           3675
    PAGE 515                                    3980 3981 4005
    PAGE 516                                    4006 4020
    PAGE 517                                    4084 4085 4099
    PAGE 518                                    4143 4144 4158
    PAGE 533  ROUTINE UNIT.INPUT                4684
    PAGE 536                                    4897 4900
    PAGE 537                                    4922
    PAGE 539  ROUTINE READ.ORDERS               4996
    PAGE 548  ROUTINE TB.INPUT                  5314
    PAGE 575  ROUTINE PGM.INPUT                 6453
    PAGE 577  ROUTINE MINE.INPUT                6540
    PAGE 578                                    6636
    PAGE 582  ROUTINE TACAIR.INPUT              6743
    PAGE 592  ROUTINE AMMO.RPT                  7126 7139 7141 7142 7143 7174 7175
    PAGE 593                                    7187 7190 7204 7207 7234
    PAGE 595  ROUTINE ANALYSIS.OUTPUT           7268 7277 7296
    PAGE 596                                    7311 7325
    PAGE 597                                    7362 7370 7378 7386 7394
    PAGE 598                                    7416 7420
    PAGE 607  ROUTINE KV.PRINT                  7750 7763
    PAGE 609  ROUTINE KV.SCOREBOARD             7867 7873
    PAGE 611  ROUTINE OUTPUT.ATTRITION          7932 7940
    PAGE 621  ROUTINE TACAIR.DATA.REPORT        8343 8381 8389
    PAGE 622                                    8431
    PAGE 623                                    8454 8478
    PAGE 656  ROUTINE AR.DETECTION              9390
\TO_AVOID_REDUNDANT_DESTROYS
    PAGE 481  PROCESS FIRE.MISSION              2112
    PAGE 485                                    2363
\TO_LET_CONTINUE
    PAGE 193  ROUTINE FINAL.COVERAGE            8869
\T011
    PAGE  71  ROUTINE ORIENTATION               3634
```

```
\UNNECESSARY?
    PAGE 653  PROCESS AIRBORNE.RADAR                9262
    PAGE 654                                        9354
    PAGE 658  PROCESS PHOTO.IR.FLIGHT               9474
    PAGE 714  ROUTINE PLAT.COUNT                    2505
\VAX
    PAGE  50  ''SECTION FOR DEFINITIONS             2835 2836 2837 2838
    PAGE  55  ''PROGRAM'' MAIN                      3027
    PAGE  56  ROUTINE MAIN1                         3042 3043
    PAGE  57  ROUTINE MAIN3                         3082
    PAGE 154  ROUTINE AO.DETECTION                  7095
    PAGE 276  ROUTINE AC.BOMB.EFFECTS               2803
    PAGE 280  ROUTINE AC.DF.EFFECTS                 3011
    PAGE 293  ROUTINE END.CAS.MISSION               3686
    PAGE 368  EVENT END.SIMULATION                  6666
    PAGE 392  EVENT SET.DEBUG                       7631
    PAGE 428  PROCESS AIR.OBSERVER                  9351
    PAGE 486  PROCESS FIRE.MISSION                  2405 2407
    PAGE 504  PROCESS CAS.MISSION                   3364
    PAGE 520  ROUTINE MAIN2                         4194
    PAGE 525  ROUTINE PK.INPUT                      4431
    PAGE 599  ROUTINE BETWEEN.ROUTINE               7437
    PAGE 617  ROUTINE SNAP2                         8197
    PAGE 618  ROUTINE SNAP.R                        8207
\VAX_CHG_3
    PAGE 565  ROUTINE SENSOR.INPUT                  6017
\VAX_CHG_4
    PAGE  41  ''SECTION FOR EVENTS                  2341 2347
\ZERO_SUB
    PAGE  90  ROUTINE FA.BN.MOVEMENT                4459
    PAGE  91                                        4538
    PAGE 182  ROUTINE FA.BN.ASGN                    8318
    PAGE 238  ROUTINE SWITCH.FO                      934  949
    PAGE 254  ROUTINE FO.DETECTION                  1690
    PAGE 500  PROCESS SHOOT.OUT                     3161
    PAGE 502                                        3316
\1
    PAGE   5  ''PROGRAM'' PREAMBLE                   218
    PAGE  49  ''SECTION FOR DEFINITIONS             2798
    PAGE  50  ''SECTION FOR SUBSTITUTIONS           2865 2866 2867 2868 2869 2889 2890
    PAGE  51                                        2891 2892 2893
    PAGE  58  ROUTINE CREATE.FORCE                  3143
    PAGE  59                                        3149
    PAGE  61  ROUTINE FEBA.INITIAL                  3197 3208
    PAGE  62                                        3243 3245
    PAGE  63  ROUTINE FILE.FD.SCHD                  3269
    PAGE  66  ROUTINE FORM.TF.LIST                  3424
    PAGE  68  ROUTINE GENERAL.BATTLE                3519 3551
    PAGE  73  ROUTINE ORIENTATION                   3798
    PAGE  88  ROUTINE END.MOVE                      4386
    PAGE  89  ROUTINE FA.BN.MOVEMENT                4423
    PAGE  97  ROUTINE LINE.OF.SIGHT                 4738 4757
    PAGE 100  ROUTINE LOCATE.SECTOR                 4841
    PAGE 101  ROUTINE LOS.CHECK                     4881
    PAGE 109  ROUTINE NEW.SEGMENT                   5256
    PAGE 113  ROUTINE PREPARE.LIST                  5412
```

SDDL MODS

PAGE 114 ROUTINE PREP.WITHDRAW 5491
PAGE 116 ROUTINE PROX.CHECK 5555
PAGE 117 ROUTINE PROX.POS 5593
PAGE 118 ROUTINE REIN.ARRIVE 5636
PAGE 126 ROUTINE WHAT.NEXT 5966
PAGE 130 ROUTINE CHECK.DEAD 6074 6080 6098 6123
PAGE 133 ROUTINE CHECK.FORCE 6197
PAGE 137 ROUTINE CHECK.LIST 6344 6364
PAGE 138 ROUTINE CHECK.PROX 6421 6423 6425
PAGE 139 6441 6442 6478
PAGE 140 6491
PAGE 141 ROUTINE CHECK.STREN 6515
PAGE 144 ROUTINE DEAD.UNIT 6689
PAGE 146 ROUTINE INTER.BATTLE 6771 6779
PAGE 147 6801 6805 6813 6821 6834 6844 6849 6852
PAGE 148 6858 6871 6881 6886 6889
PAGE 154 ROUTINE AO.DETECTION 7125
PAGE 156 7235
PAGE 165 ROUTINE CFR.DEGRADE 7595
PAGE 171 ROUTINE COMPARE.TRS 7834
PAGE 178 ROUTINE EST.COVERAGE 8161
PAGE 181 ROUTINE FASCAM.COMPUTATION 8254
PAGE 182 ROUTINE FA.BN.ASGN 8310 8326
PAGE 184 8402 8423
PAGE 191 ROUTINE FINAL.COVERAGE 8772
PAGE 192 8783 8789 8800
PAGE 193 8878
PAGE 194 ROUTINE FIND.START.TIME 8910 8942
PAGE 195 8963
PAGE 196 ROUTINE HE.OR.ICM.COMPUTATION 9012 9037 9042
PAGE 199 ROUTINE ILLUM.COMPUTATION 9145
PAGE 203 ROUTINE MARGINAL.EFFECTS.ADJ 9339 9353
PAGE 204 9366
PAGE 205 ROUTINE NOISE.DEGRADE 9393 9398 9405 9410
PAGE 206 ROUTINE PDB.DETECTION 9476
PAGE 208 ROUTINE PGM.MSN.ASGN 9526
PAGE 209 9556 9603
PAGE 210 9618
PAGE 212 ROUTINE PIR.DETECTION 9730 9747 9758
PAGE 213 9768
PAGE 217 ROUTINE REQUEST.FASCAM 9958
PAGE 229 ROUTINE RPV.DETECTION 543 560
PAGE 230 571 581 596
PAGE 231 ROUTINE SIZE.ESTIMATE 637 662 665 668
PAGE 233 ROUTINE SMOKE.COMPUTATION 699
PAGE 240 ROUTINE TARGET.ANALYSIS 1030
PAGE 242 ROUTINE UNIT.ENVIR 1154
PAGE 243 1213
PAGE 244 1229 1240 1263 1272
PAGE 246 ROUTINE VOLLEY 1341 1348
PAGE 251 ROUTINE MINE.EFFECTS 1536
PAGE 257 ROUTINE FO.DETECTION 1874
PAGE 258 1906 1908
PAGE 263 ROUTINE BTRY.EFFECTS 2157
PAGE 264 2246
PAGE 271 2608

PAGE 276	ROUTINE AC.BOMB.EFFECTS	2842
PAGE 280	ROUTINE AC.DF.EFFECTS	3014
PAGE 283	ROUTINE CAS.EVAL	3219
PAGE 285	ROUTINE CHECK.CAS.CONSTRAINTS	3289
PAGE 309	ROUTINE AD.SHOOT	4467
PAGE 317	ROUTINE FLIGHT.PATH	4820
PAGE 324	ROUTINE DECIDE	5063
PAGE 325	ROUTINE DEQ.FEBA.SET	5101
PAGE 326	ROUTINE DESTROY.ORD	5123
PAGE 331	ROUTINE ENQ.FEBA.SET	5315
PAGE 334	ROUTINE FINISH.COMPUTATION	5408
PAGE 335	ROUTINE FRAC.COMPUTE	5425 5428
PAGE 337	ROUTINE HC.EMPTY	5491
PAGE 348	EVENT ACT.REINF	5893
PAGE 349	EVENT AD.ENGAGEMENT	5941
PAGE 351		604' 6070
PAGE 357	EVENT BTL.ENDED	6330
PAGE 358	EVENT CFR.ACTIVATION	6367
PAGE 360	EVENT CFR.ON	6452
PAGE 362	EVENT CFR.OPERATOR	6517 6531 6533
PAGE 363		6543 6547 6589
PAGE 371	EVENT FEBA.SORTIE	6791 6830
PAGE 372		6834 6871
PAGE 373	EVENT GET.NX.ORD	6915 6930
PAGE 374		6945 6951 6967
PAGE 375	EVENT HC.DEPART.BATTLE	7017
PAGE 376	EVENT HELO.ENGAGEMENT	7041
PAGE 377		7140
PAGE 379		7213
PAGE 380	EVENT INIT.PREPLAN.CAS	7255
PAGE 383	EVENT OFF.LINE.ATTRITION	7375
PAGE 384		7426 7441
PAGE 385		7459
PAGE 386	EVENT PDB.ACTIVATION	7493
PAGE 387	EVENT PDB.OPERATOR	7535
PAGE 397	EVENT START.BATTLE	7880
PAGE 402		8168
PAGE 408	EVENT UPDATE.LOC	8416 8420
PAGE 410		8526 8529
PAGE 413	EVENT ACT.ATK	8710
PAGE 415	EVENT ACT.MOVCOR	8762
PAGE 416	EVENT ACT.MOVDIS	8799
PAGE 419	PROCESS AC.ATK.TGT	8899
PAGE 420		8957
PAGE 422		9074
PAGE 429	PROCESS AIR.OBSERVER	9390 9400 9412
PAGE 430		9454 9458 9478 9480
PAGE 431		9505 9521 9526 9539 9545
PAGE 432		9570 9576 9604 9609
PAGE 433		9622 9627 9640 9645 9659 9664 9674
PAGE 434		9679
PAGE 439	PROCESS FORWARD.OBSERVER	9921 9939
PAGE 449	PROCESS HC.ARRIVE.BATTLE	452
PAGE 451	PROCESS HC.RETURN.FARRP	560
PAGE 453		667 687
PAGE 457	PROCESS HEL.TARGET.ACQUISITION	873

```
    PAGE 460                                    1074 1075 1080
    PAGE 461                                    1099 1101 1103 1117 1126 1129
    PAGE 462                                    1158
    PAGE 463  PROCESS HOW.REPAIR                1184 1192 1207
    PAGE 464  PROCESS MINE.ASSESS               1263
    PAGE 467  PROCESS REMOTE.PILOT.VEHICLE      1401 1408 1410
    PAGE 468                                    1440 1471
    PAGE 469                                    1476
    PAGE 471  PROCESS TARGET.REPORT             1549
    PAGE 473                                    1676
    PAGE 474                                    1755 1760
    PAGE 479  PROCESS WITH.DRAW                 1983 2006
    PAGE 481  PROCESS FIRE.MISSION              2132
    PAGE 482                                    2176
    PAGE 483                                    2224
    PAGE 485                                    2356
    PAGE 487  PROCESS ASSESSMENT                2443
    PAGE 502  PROCESS SHOOT.OUT                 3279
    PAGE 505  PROCESS CAS.MISSION               3431
    PAGE 508                                    3587
    PAGE 536  ROUTINE UNIT.INPUT                4877 4886 4911
    PAGE 537                                    4917
    PAGE 540  ROUTINE READ.ORDERS               5062
    PAGE 542  ROUTINE ORD.DEF                   5124
    PAGE 546  ROUTINE ORD.MOVCOR                5214 5222
    PAGE 551  ROUTINE BTRY.INPUT                5494 5500
    PAGE 556  ROUTINE MUNS.INPUT                5670
    PAGE 596  ROUTINE ANALYSIS.OUTPUT           7338
    PAGE 597                                    7356
    PAGE 606  ROUTINE KV.PRINT                  7702 7704 7729 7734 7738 7739 7743
    PAGE 607                                    7746 7747 7751 7754 7755 7759 7761 7764 7765 7766 7770 7773 7774 7778 7781
                                                7782 7786 7787
    PAGE 608                                    7817 7819 7839
    PAGE 609  ROUTINE KV.SCOREBOARD             7861 7864 7893
    PAGE 610                                    7905 7910 7911 7921 7926
    PAGE 611  ROUTINE OUTPUT.ATTRITION          7953 7961
    PAGE 612                                    7989
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES       8513 8516 8522 8525 8533 8534 8541 8544 8545
    PAGE 625                                    8555 8556 8563 8564 8571 8574 8575 8585 8586 8598 8600
    PAGE 626                                    8612 8614
    PAGE 631  FUNCTION COMBINATIONS             8728 8733
    PAGE 637  FUNCTION HE.WLA                   8923 8939
    PAGE 638  FUNCTION ICM.WLA                  8985 9003 9013
    PAGE 640  ROUTINE EXPONENTIAL.F             9031
    PAGE 641  ROUTINE NORMAL.F                  9044 9047 9052 9054 9056
    PAGE 642  ROUTINE WEIBULL.F                 9067
    PAGE 652  ROUTINE GAMMA.F                   9241 9252 9258
    PAGE 656  ROUTINE AR.DETECTION              9405 9429
    PAGE 658  PROCESS PHOTO.IR.FLIGHT           9505 9507 9516 9518
    PAGE 659                                    9558 9577
    PAGE 660  FUNCTION STAY.TIME                9605
    PAGE 664  ''PROGRAM  OLDER.VERSION          9655
    PAGE 708                                    2219
\2
    PAGE 653  PROCESS AIRBORNE.RADAR            9266
    PAGE 658  PROCESS PHOTO.IR.FLIGHT           9477
```

```
ALPHA
    PAGE   1  ROUTINE FOR CROSS_REFERENCING          43
    PAGE 149  ROUTINE PK.COMPUTE                   6950
    PAGE 150                                       6953 6955 6957
    PAGE 151                                       7010 7031
    PAGE 581  ROUTINE VIS.INPUT                    6718
    PAGE 667  ''PROGRAM  OLDER.VERSION             9865
    PAGE 705                                       2082
    PAGE 706                                       2130
    PAGE 708                                       2255 2260
CREATE
    PAGE   1  ROUTINE FOR CROSS_REFERENCING          42
    PAGE  56  ROUTINE MAIN1                        3048 3051 3054 3057
    PAGE  58  ROUTINE CREATE.FORCE                 3105
    PAGE  60  ROUTINE CREATE.TEAMS                 3165 3174
    PAGE  61  ROUTINE FEBA.INITIAL                 3191
    PAGE  63  ROUTINE FILE.FD.SCHD                 3282 3293 3301 3311
    PAGE  64                                       3331 3332 3345
    PAGE  65  ROUTINE FILE.KAD.SENSOR              3375 3377 3382
    PAGE  68  ROUTINE GENERAL.BATTLE               3527 3533 3540 3559
    PAGE  69                                       3565 3572
    PAGE  75  ROUTINE UNIT.ASSIGNMENT              3856
    PAGE  95  ROUTINE INIT.REINF                   4649 4651
    PAGE  96  ROUTINE LINE.OF.SIGHT                4715 4719
    PAGE  97                                       4728 4747
    PAGE 101  ROUTINE LOS.CHECK                    4913 4917
    PAGE 109  ROUTINE NEW.SEGMENT                  5247 5251
    PAGE 135  ROUTINE CHECK.FOR.MINES              6320
    PAGE 144  ROUTINE DEAD.UNIT                    6660 6664 6673 6677
    PAGE 157  ROUTINE AO.DETECTION                 7283
    PAGE 167  ROUTINE CFR.DETECTION                7688
    PAGE 184  ROUTINE FA.BN.ASGN                   8414
    PAGE 206  ROUTINE PDB.DETECTION                9454
    PAGE 209  ROUTINE PGM.MSN.ASGN                 9593
    PAGE 213  ROUTINE PIR.DETECTION                9780
    PAGE 230  ROUTINE RPV.DETECTION                 611
    PAGE 246  ROUTINE VOLLEY                       1355
    PAGE 251  ROUTINE MINE.EFFECTS                 1566
    PAGE 256  ROUTINE FO.DETECTION                 1790
    PAGE 261  ROUTINE BTRY.EFFECTS                 2079
    PAGE 269                                       2533
    PAGE 284  ROUTINE CAS.EVAL                     3259
    PAGE 290  ROUTINE EMPLOY.HELICOPTERS           3541 3544 3555 3578 3581
    PAGE 291                                       3592 3627 3635 3644
    PAGE 310  ROUTINE AD.SHOOT                     4482 4484
    PAGE 314  ROUTINE FLIGHT.PATH                  4677
    PAGE 315                                       4692 4709 4718 4744
    PAGE 316                                       4748
    PAGE 317                                       4818
    PAGE 362  EVENT CFR.OPERATOR                   6503 6524
    PAGE 369  EVENT ENGAGEMENT                     6721
    PAGE 371  EVENT FEBA.SORTIE                    6820
    PAGE 372                                       6861
    PAGE 377  EVENT HELO.ENGAGEMENT                7092 7123 7128
    PAGE 378                                       7188
    PAGE 380  EVENT INIT.PREPLAN.CAS               7238
```

PAGE 387	EVENT PDB.OPERATOR	7522 7541
PAGE 395	EVENT START.BATTLE	7778
PAGE 405	EVENT START.MOVE	8309
PAGE 410	EVENT UPDATE.LOC	8564
PAGE 419	PROCESS AC.ATK.TGT	8897
PAGE 420		8929
PAGE 429	PROCESS AIR.OBSERVER	9445
PAGE 430		9496
PAGE 431		9553
PAGE 438	PROCESS FORWARD.OBSERVER	9877
PAGE 439		9919 9937
PAGE 440		1
PAGE 456	PROCESS HEL.TARGET.ACQUISITION	834 847
PAGE 467	PROCESS REMOTE.PILOT.VEHICLE	1388
PAGE 468		1431 1462
PAGE 470	PROCESS TARGET.REPORT	1499
PAGE 479	PROCESS WITH.DRAW	1980
PAGE 480	PROCESS FIRE.MISSION	2056
PAGE 493	PROCESS SHOOT.OUT	2773
PAGE 499		3106
PAGE 504	PROCESS CAS.MISSION	3386 3387
PAGE 507		3572
PAGE 525	ROUTINE PK.INPUT	4389 4414 4429
PAGE 527	ROUTINE CAT.TU.INPUT	4460 4471 4492
PAGE 528		4510
PAGE 529	ROUTINE KV.INPUT	4535 4544 4553
PAGE 530	ROUTINE EQ.TE.INPUT	4580 4597
PAGE 532	ROUTINE TYPE.WEAPON.INPUT	4647
PAGE 533	ROUTINE UNIT.INPUT	4687
PAGE 534		4738 4760
PAGE 535		4814
PAGE 538	ROUTINE MFO.INPUT	4936 4939 4942
PAGE 540	ROUTINE READ.ORDERS	5051
PAGE 542	ROUTINE ORD.DEF	5131
PAGE 543	ROUTINE ORD.ATK	5152
PAGE 544	ROUTINE ORD.REINF	5173
PAGE 545	ROUTINE ORD.MOVDIS	5195
PAGE 546	ROUTINE ORD.MOVCOR	5232
PAGE 547	ROUTINE P.E.M.INPUT	5264 5270 5276
PAGE 548	ROUTINE TB.INPUT	5317 5320 5364
PAGE 550	ROUTINE BTRY.INPUT	5398 5404
PAGE 551		5490
PAGE 553	ROUTINE FBN.FD.INPUT	5549 5563
PAGE 555	ROUTINE MUNS.INPUT	5614 5632
PAGE 557	ROUTINE SUBM.INPUT	5722
PAGE 560	ROUTINE RUL.EN.INPUT	5826
PAGE 561	ROUTINE ST.INPUT	5863
PAGE 562	ROUTINE MCFR.INPUT	5884 5887
PAGE 563	ROUTINE MPDB.INPUT	5921 5924
PAGE 564	ROUTINE MAO.INPUT	5963 5979 5989
PAGE 566	ROUTINE SENSOR.INPUT	6074 6088 6120
PAGE 567		6130 6137
PAGE 569	ROUTINE TBF.INPUT	6220
PAGE 570	ROUTINE DECISION.INPUT	6267 6268
PAGE 571	ROUTINE TT.FACTORS.INPUT	6290
PAGE 573	ROUTINE FARRP.INPUT	6419

```
    PAGE 576  ROUTINE ILLUM.INPUT                    6507
    PAGE 577  ROUTINE MINE.INPUT                     6563 6574
    PAGE 578                                         6600 6621
    PAGE 580  ROUTINE SMOKE.INPUT                    6680
    PAGE 582  ROUTINE TACAIR.INPUT                   6793
    PAGE 587  ROUTINE MADS.INPUT                     6984 6998
    PAGE 588  ROUTINE AC.MUNS.INPUT                  7017
    PAGE 590  ROUTINE TR.INPUT                       7099
    PAGE 653  PROCESS AIRBORNE.RADAR                 9306
    PAGE 654                                         9333 9370
    PAGE 656  ROUTINE AR.DETECTION                   9438
    PAGE 658  PROCESS PHOTO.IR.FLIGHT                9523
    PAGE 659                                         9556 9586
DESTROY
    PAGE   1  ROUTINE FOR CROSS_REFERENCING           42
    PAGE   2  PROGRAM REVISIONS                       94
    PAGE   3                                         149
    PAGE  58  ROUTINE CREATE.FORCE                  3145
    PAGE  59                                        3151
    PAGE  63  ROUTINE FILE.FD.SCHD                  3272
    PAGE  65  ROUTINE FILE.KAD.SENSOR               3410
    PAGE  67  ROUTINE GENERAL.BATTLE                3469 3472
    PAGE  80  ROUTINE BLOCK.LOS                     4061
    PAGE  81                                        4067 4077 4103 4117
    PAGE  84  ROUTINE CHANGE.LOC                    4211
    PAGE  85                                        4275
    PAGE  87  ROUTINE END.MOVE                      4360
    PAGE 108  ROUTINE NEW.SEGMENT                   5161 5169 5198 5205
    PAGE 109                                        5221 5238
    PAGE 112  ROUTINE PREPARE.LIST                  5390 5397
    PAGE 160  ROUTINE ATTRIT.SENSOR                 7386 7397 7421
    PAGE 161                                        7428 7433 7436 7447 7455 7463 7470 7475 7484
    PAGE 162                                        7508
    PAGE 165  ROUTINE CFR.DEGRADE                   7599
    PAGE 170  ROUTINE COMBINE.TRS                   7789
    PAGE 173  ROUTINE DUST.EFFECTS                  7916
    PAGE 174                                        7980 7995
    PAGE 184  ROUTINE FA.BN.ASGN                    8438 8441 8459
    PAGE 185                                        8491
    PAGE 194  ROUTINE FIND.START.TIME               8913
    PAGE 205  ROUTINE NOISE.DEGRADE                 9396 9408
    PAGE 206  ROUTINE PDB.DETECTION                 9479
    PAGE 210  ROUTINE PGM.MSN.ASGN                  9660
    PAGE 235  ROUTINE SMOKE.EFFECTS                  804
    PAGE 236                                         904
    PAGE 238  ROUTINE SWITCH.FO                      942  958
    PAGE 252  ROUTINE MINE.EFFECTS                  1621
    PAGE 261  ROUTINE BTRY.EFFECTS                  2074
    PAGE 271                                        2615
    PAGE 279  ROUTINE AC.BOMB.EFFECTS               2962
    PAGE 282  ROUTINE AC.DF.EFFECTS                 3142
    PAGE 287  ROUTINE CHECK.CAS.CONSTRAINTS         3398
    PAGE 293  ROUTINE END.CAS.MISSION               3694 3697 3723
    PAGE 300  ROUTINE HC.DISENGAGE                  4031 4040 4060
    PAGE 301                                        4070 4096
    PAGE 304  ROUTINE REPLACE.HC                    4215 4219 4222 4225
```

```
PAGE 312  ROUTINE INTER.HELO                    4570 4580 4603 4613
PAGE 316  ROUTINE FLIGHT.PATH                   4754
PAGE 317                                        4844
PAGE 318                                        4879 4907
PAGE 326  ROUTINE DESTROY.ORD                   5126 5128 5130 5132 5134 5136 5140
PAGE 328  ROUTINE EMPTY                         5187 5194
PAGE 329                                        5244 5254 5272 5278
PAGE 330                                        5283 5297
PAGE 334  ROUTINE FINISH.COMPUTATION            5410
PAGE 337  ROUTINE HC.EMPTY                      5498
PAGE 338                                        5558
PAGE 339                                        5595
PAGE 363  EVENT CFR.OPERATOR                    6560 6579 6580
PAGE 367  EVENT DQ.OLD.SORTIE.QUEUE             6649
PAGE 369  EVENT ENGAGEMENT                      6703
PAGE 376  EVENT HELO.ENGAGEMENT                 7063
PAGE 384  EVENT OFF.LINE.ATTRITION              7446
PAGE 387  EVENT PDB.OPERATOR                    7548 7557 7558
PAGE 393  EVENT START.ARTY.MOVEMENT             7700
PAGE 405  EVENT START.MOVE                      8313
PAGE 409  EVENT UPDATE.LOC                      8503
PAGE 410                                        8568
PAGE 427  PROCESS AC.ATK.TGT                    9314 9327
PAGE 429  PROCESS AIR.OBSERVER                  9392 9402 9404 9425
PAGE 431                                        9508 9518 9523 9528 9530 9541 9543 9547 9549 9558 9559
PAGE 432                                        9572 9574 9578 9580 9581 9592 9606 9611 9613
PAGE 433                                        9624 9629 9631 9642 9647 9649 9661 9666 9668 9676
PAGE 434                                        9681 9684
PAGE 437  PROCESS ARTY.ASSESS                   9849
PAGE 440  PROCESS FORWARD.OBSERVER              9994 9995 9996
PAGE 441                                          40   41   42   63   67   82   83
PAGE 442                                          89
PAGE 451  PROCESS HC.RETURN.FARRP                534  548
PAGE 460  PROCESS HEL.TARGET.ACQUISITION        1085
PAGE 461                                        1109 1119 1131
PAGE 464  PROCESS MINE.ASSESS                   1250
PAGE 465                                        1307
PAGE 468  PROCESS REMOTE.PILOT.VEHICLE          1452
PAGE 469                                        1473 1478
PAGE 470  PROCESS TARGET.REPORT                 1518 1535
PAGE 471                                        1582
PAGE 472                                        1605 1628
PAGE 473                                        1659
PAGE 474                                        1729
PAGE 475                                        1795
PAGE 477                                        1909
PAGE 479  PROCESS WITH.DRAW                     2017
PAGE 481  PROCESS FIRE.MISSION                  2111
PAGE 485                                        2362
PAGE 486                                        2387 2406
PAGE 489  PROCESS ASSESSMENT                    2531 2547 2572
PAGE 490                                        2591
PAGE 491                                        2651
PAGE 493  PROCESS SHOOT.OUT                     2791 2803
PAGE 495                                        2876 2879
PAGE 496                                        2958 2970
```

PAGE 498		3053 3063
PAGE 500		3156 3189
PAGE 502		3311
PAGE 503		3330 3345
PAGE 511	PROCESS HELICOPTER.FIRE	3736
PAGE 527	ROUTINE CAT.TU.INPUT	4495
PAGE 528		4513
PAGE 534	ROUTINE UNIT.INPUT	4763
PAGE 535		4817
PAGE 549	ROUTINE TB.INPUT	5369
PAGE 590	ROUTINE TR.INPUT	7104
PAGE 630	FUNCTION COLLISION	8689 8709
PAGE 654	PROCESS AIRBORNE.RADAR	9361
PAGE 659	PROCESS PHOTO.IR.FLIGHT	9543 9551 9565
EXTENDED		
PAGE 1	ROUTINE FOR CROSS_REFERENCING	43
PAGE 708	''PROGRAM OLDER.VERSION	2256 2257 2258 2259
PAGE 711		2417 2420
RELEASE		
PAGE 1	ROUTINE FOR CROSS_REFERENCING	42
PAGE 113	ROUTINE PREPARE.LIST	5458
PAGE 118	ROUTINE REIN.ARRIVE	5617
PAGE 130	ROUTINE CHECK.DEAD	6117
PAGE 131		6129
PAGE 138	ROUTINE CHECK.PROX	6415 6416
PAGE 139		6461
PAGE 140		6497 6501
PAGE 147	ROUTINE INTER.BATTLE	6853
PAGE 148		6890
PAGE 192	ROUTINE FINAL.COVERAGE	8803
PAGE 213	ROUTINE PIR.DETECTION	9789
PAGE 230	ROUTINE RPV.DETECTION	621
PAGE 269	ROUTINE BTRY.EFFECTS	2521
PAGE 273		2729
PAGE 279	ROUTINE AC.BOMB.EFFECTS	2992
PAGE 292	ROUTINE EMPLOY.HELICOPTERS	3672
PAGE 296	ROUTINE END.CAS.MISSION	3850
PAGE 347	EVENT ACT.REINF	5855
PAGE 357	EVENT BTL.ENDED	6338 6339
PAGE 373	EVENT GET.NX.ORD	6917
PAGE 374		6961 6969
PAGE 405	EVENT START.MOVE	8343
PAGE 406		8353
PAGE 409	EVENT UPDATE.LOC	8511
PAGE 410		8521
PAGE 413	EVENT ACT.ATK	8698
PAGE 429	PROCESS AIR.OBSERVER	9394 9405 9426
PAGE 431		9531 9550
PAGE 432		9582 9614
PAGE 433		9632 9650 9669 9672
PAGE 554	ROUTINE FBN.FD.INPUT	5598
PAGE 568	ROUTINE SENSOR.INPUT	6198
PAGE 593	ROUTINE AMMO.RPT	7228
PAGE 596	ROUTINE ANALYSIS.OUTPUT	7328
PAGE 597		7397
PAGE 606	ROUTINE KV.PRINT	7700

```
    PAGE 608                                    7840
    PAGE 610  ROUTINE KV.SCOREBOARD             7927
RESERVE
    PAGE   1  ROUTINE FOR CROSS_REFERENCING       42
    PAGE  56  ROUTINE MAIN1                     3063
    PAGE 112  ROUTINE PREPARE.LIST              5365
    PAGE 113                                    5451
    PAGE 130  ROUTINE CHECK.DEAD                6116 6120
    PAGE 138  ROUTINE CHECK.PROX                6390 6396 6414
    PAGE 139                                    6488
    PAGE 140                                    6500
    PAGE 147  ROUTINE INTER.BATTLE              6819
    PAGE 148                                    6856
    PAGE 191  ROUTINE FINAL.COVERAGE            8766
    PAGE 212  ROUTINE PIR.DETECTION             9711
    PAGE 229  ROUTINE RPV.DETECTION              523
    PAGE 260  ROUTINE BTRY.EFFECTS              1998
    PAGE 269                                    2503
    PAGE 276  ROUTINE AC.BOMB.EFFECTS           2805
    PAGE 289  ROUTINE EMPLOY.HELICOPTERS        3487
    PAGE 311  ROUTINE INTER.HELO                4523
    PAGE 412  EVENT ACT.ATK                     8620 8646
    PAGE 413                                    8671
    PAGE 428  PROCESS AIR.OBSERVER              9352
    PAGE 504  PROCESS CAS.MISSION               3382
    PAGE 520  ROUTINE MAIN2                     4212 4213
    PAGE 523  ROUTINE SYS.INPUT                 4327
    PAGE 525  ROUTINE PK.INPUT                  4406 4409 4431 4432
    PAGE 526                                    4440 4444
    PAGE 529  ROUTINE KV.INPUT                  4529 4538 4547 4555 4557 4559
    PAGE 553  ROUTINE FBN.FD.INPUT              5551
    PAGE 572  ROUTINE FARRP.INPUT               6340 6348
    PAGE 575  ROUTINE PGM.INPUT                 6455 6467 6468
    PAGE 577  ROUTINE MINE.INPUT                6543 6544
    PAGE 592  ROUTINE AMMO.RPT                  7129
    PAGE 595  ROUTINE ANALYSIS.OUTPUT           7246
    PAGE 606  ROUTINE KV.PRINT                  7700 7727
    PAGE 609  ROUTINE KV.SCOREBOARD             7888
    PAGE 624  ROUTINE OUTPUT.EXPENDITURES       8508
    PAGE 648  ROUTINE PERFORM.INSTRUMENTATION   9196 9197 9203 9204
TEXT
    PAGE   1  ROUTINE FOR CROSS_REFERENCING       43
    PAGE   8  ''SECTION FOR PERMANENT_ENTITIES    424
    PAGE  47  ''SECTION FOR DEFINITIONS         2658 2707
    PAGE  50                                    2834 2839 2845
    PAGE  95  ROUTINE INIT.REINF                4647
    PAGE 176  ROUTINE EST.COVERAGE              8018
    PAGE 188  ROUTINE FINAL.COVERAGE            8561
    PAGE 206  ROUTINE PDB.DETECTION             9437
    PAGE 246  ROUTINE VOLLEY                    1314
    PAGE 248  ROUTINE WEIGHTED.VOLLEYS          1386
    PAGE 250  ROUTINE MINE.EFFECTS              1477
    PAGE 260  ROUTINE BTRY.EFFECTS              1984
    PAGE 293  ROUTINE END.CAS.MISSION           3685
    PAGE 307  ROUTINE AD.SHOOT                  4313
    PAGE 314  ROUTINE FLIGHT.PATH               4639
```

```
     PAGE 328  ROUTINE EMPTY                          5177
     PAGE 349  EVENT AD.ENGAGEMENT                    5907
     PAGE 358  EVENT CFR.ACTIVATION                   6347
     PAGE 382  EVENT OFF.LINE.ATTRITION               7283
     PAGE 386  EVENT PDB.ACTIVATION                   7475
     PAGE 395  EVENT START.BATTLE                     7755
     PAGE 403  EVENT START.MOVE                       8194
     PAGE 408  EVENT UPDATE.LOC                       8428
     PAGE 510  PROCESS HELICOPTER.FIRE                3675
     PAGE 533  ROUTINE UNIT.INPUT                     4684
     PAGE 539  ROUTINE READ.ORDERS                    4996
     PAGE 548  ROUTINE TB.INPUT                       5314
     PAGE 575  ROUTINE PGM.INPUT                      6453
     PAGE 577  ROUTINE MINE.INPUT                     6540
     PAGE 582  ROUTINE TACAIR.INPUT                   6743
     PAGE 592  ROUTINE AMMO.RPT                       7126
     PAGE 611  ROUTINE OUTPUT.ATTRITION               7932
     PAGE 621  ROUTINE TACAIR.DATA.REPORT             8343
     PAGE 656  ROUTINE AR.DETECTION                   9390
USE
     PAGE   1  ROUTINE FOR CROSS_REFERENCING           43
     PAGE   2  PROGRAM REVISIONS                       78   85   99
     PAGE  18  ''SECTION FOR PERMANENT_ENTITIES       984
     PAGE  56  ROUTINE MAIN1                         3043
     PAGE  90  ROUTINE FA.BN.MOVEMENT                4469 4472 4488 4491
     PAGE  91                                        4547 4550 4561 4564
     PAGE  92                                        4578 4581
     PAGE 163  ROUTINE BTRY.FM.DEQ                   7532 7535
     PAGE 164  ROUTINE BTRY.FM.ENQ                   7557 7560
     PAGE 172  ROUTINE COPY                          7852
     PAGE 176  ROUTINE EST.COVERAGE                  8005
     PAGE 197  ROUTINE HE.OR.ICM.COMPUTATION         9099 9104
     PAGE 226  ROUTINE REQUEST.WD.FASCAM              397
     PAGE 256  ROUTINE FO.DETECTION                  1800
     PAGE 262  ROUTINE BTRY.EFFECTS                  2137
     PAGE 263                                        2159
     PAGE 265                                        2283 2290
     PAGE 285  ROUTINE CHECK.CAS.CONSTRAINTS         3317
     PAGE 286                                        3342 3388
     PAGE 295  ROUTINE END.CAS.MISSION               3814 3818
     PAGE 355  EVENT ARTY.OCCUPATION                 6238 6241
     PAGE 394  EVENT START.ARTY.MOVEMENT             7723 7726
     PAGE 407  EVENT STOP.ARTY.MOVEMENT              8388 8391
     PAGE 412  EVENT ACT.ATK                         8625
     PAGE 417  EVENT DYNAMIC.ANALYSIS.REPORT         8810 8836
     PAGE 419  PROCESS AC.ATK.TGT                    8900
     PAGE 422                                        9072
     PAGE 441  PROCESS FORWARD.OBSERVER                54   60
     PAGE 476  PROCESS TARGET.REPORT                 1873 1884
     PAGE 533  ROUTINE UNIT.INPUT                    4698
     PAGE 536                                        4909
     PAGE 577  ROUTINE MINE.INPUT                    6537
     PAGE 595  ROUTINE ANALYSIS.OUTPUT               7254
     PAGE 598                                        7424
     PAGE 606  ROUTINE KV.PRINT                      7717 7718 7735
     PAGE 607                                        7788
```

```
PAGE 609  ROUTINE KV.SCOREBOARD                  7890
PAGE 610                                         7912
PAGE 611  ROUTINE OUTPUT.ATTRITION               7938 7955
PAGE 612                                         7990
PAGE 613  ROUTINE FOR POSITION.OUT               7999 8016
PAGE 618  ROUTINE SNAP.R                         8205
PAGE 619                                         8314
PAGE 620                                         8329
PAGE 624  ROUTINE OUTPUT.EXPENDITURES            8499 8529
PAGE 625                                         8591
PAGE 647  ROUTINE OPEN.INPUT.OUTPUT.FILES        9187
PAGE 677  ''PROGRAM  OLDER.VERSION                426
```

```
SAI-SDDL PROCESSING COMPLETE

   NO ERRORS

   99 KEYWORDS
 3888 TOKENS
39008 TOKEN OCCURRENCES
  281 MODULES
  567 MODULE INVOCATIONS
  299 TABLE OF CONTENTS ENTRIES
  499 FORWARD PAGE REFERENCES
   51 STRING STORAGE BINS
    3 STRUCTURES DEFINED
    5 XREF TABLES DEFINED
```