

AD-A092 783

ARMY COMBINED ARMS STUDIES AND ANALYSIS ACTIVITY FOR--ETC F/G 15/7
CACDA JIFFY III WAR GAME. VOLUME V. PROGRAMMER'S MANUAL.(U)

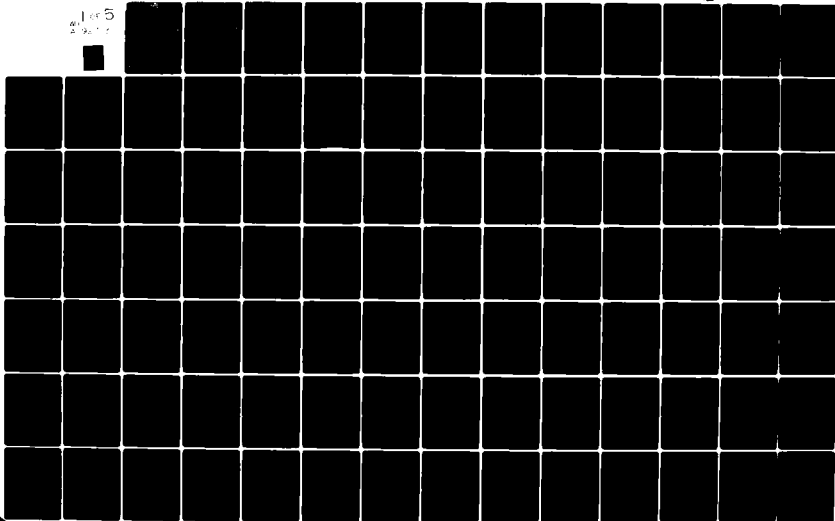
OCT 80 C L PAO

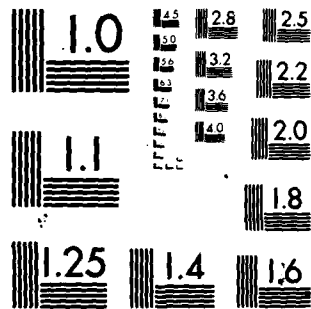
UNCLASSIFIED

CASAA-TR-9-80

NL

1 of 5
2 of 5





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

OCTOBER 1980

LEVEL

TR 9-80

12

ACN 24826

AD A092143

**CACDA JIFFY III WAR GAME
VOLUME V
PROGRAMMERS MANUAL**

Technical Report 9-80

DTIC
DEC 10 1980
C

**UNITED STATES ARMY
COMBINED ARMS CENTER**

**COMBINED ARMS
COMBAT DEVELOPMENT ACTIVITY
COMBINED ARMS STUDIES AND ANALYSIS ACTIVITY**

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

80 12 08 053

DDG FILED

12
Technical Report TR 9-80
October 1980

Combined Arms Studies and Analysis Activity
US Army Combined Arms Combat Developments Activity
Fort Leavenworth, Kansas 66027

CACDA JIFFY III WAR GAME
VOLUME V
PROGRAMMERS MANNUAL

DTIC
ELECTE
S DEC 10 1980 D

Approved by:

David J. Farmer

David J. Farmer
Chief, Technical Support Division

Richard C. Rinkel

Richard C. Rinkel
Acting Director, Research
and Support Directorate

Leland C. Pleger

Leland C. Pleger
Acting Director, CASAA

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

81-CASAA-0605

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report TR 9-80 ✓	2. GOVT ACCESSION NO. AD-A092783	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CACDA JIFFY III War Game. Volume V. Programmer's Manual.		5. TYPE OF REPORT & PERIOD COVERED 9 Final rpt.
6. AUTHOR(s) 10 Channing L. Pao Chief Compiler		7. PERFORMING ORG. REPORT NUMBER TR 9-80
8. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Combined Arms Studies and Analysis Activity Fort Leavenworth, Kansas 66027 ✓		9. CONTRACT OR GRANT NUMBER(s) 14 CASAA-TR-9-80
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 12 475
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE 11 Oct 80
		13. NUMBER OF PAGES 475
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release Distribution Unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report is one of a set of five volumes produced to document the combat assessment methodologies and automated features of the JIFFY III War Gaming process, developed to support TRADOC SCORES scenario development and force evaluation efforts. This report provides the detailed descriptions, flow diagrams, and the FORTRAN codes for all the computer routines of the JIFFY III war game model. The other four volumes in the set are the Executive Summary (Vol I), Methodology (Vol II), the classified data and their sources (Vol III), and the User's Manual (Vol IV).		

FOREWORD

The Jiffy III War Game model was used in the development of the SCORES Europe III scenario, which provides the combat developments community with a common base of assumptions, threat forces, weapons, organizations, terrain, and tactics for the 1986 timeframe. The 1977 version of the Jiffy was extensively modified and improved for the Europe III work. This report documents the Jiffy III model as used for Europe III and supersedes CACDA Jiffy War Game Documentation, Technical Manuals TR 2-77, TR 3-77, and TR 4-77. This report documents all the changes and improvements completed through April 1980.

There are five volumes of Jiffy III War Game documentation. The first volume is the Executive Summary. Volume II is the Methodology, which describes the overall Jiffy III War Game methodology including detailed descriptions of the combat assessment equations. The computer calculates the attritions based on these equations. The unclassified portions of the data are given in Volume II. Volume III contains classified data as required for the Jiffy III model. Volume IV is the User's Manual, which contains a discussion of the manual aspects and the automated features of the gaming process and exemplifies the relationship between them through some sample runs. Volume V, the Programmer's Manual, consists of descriptions, and FORTRAN code of all programs and routines associated with the Jiffy III game.

This report was compiled principally by Dr. Channing L. Pao, Dr. Robert J. Schwabauer, Mr. James H. Kennington and Mr. William D. Relph. The compilers wish to acknowledge the SCORES gaming staff of the Combined Arms Combat Development Activity who served as consultants.

ABSTRACT

This report is one of a set of five volumes produced to document the combat assessment methodologies and automated features of the Combined Arms Combat Developments Activity (CACDA) Jiffy III war gaming process. The Jiffy process was originally developed to support the TRADOC Scenario Oriented Recurring Evaluation System (SCORES) scenario development and force evaluation efforts. In 1978, the 1977 version of the Jiffy was extensively modified and improved to support Europe III scenario gaming. This report documents the Jiffy model used for that gaming. Volume II of this report contains the methodologies used in the automated routines of the Jiffy III Game. An unclassified data base, which was developed for test and demonstration purposes, is presented in Volume II. The classified data used in the Jiffy III Game during secure production runs, and their sources, are published separately as Volume III to keep the methodology volume unclassified. The other three volumes in the set are the Executive Summary (Volume I), the User's Manual (Volume IV), and the Programmer's Manual (Volume V).

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	☐
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

TABLE OF CONTENTS

	Page
FOREWORD.....	ii
ABSTRACT.....	iii
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
1. SCOPE.....	1
2. GENERAL.....	1
3. FORCE STRUCTURE GENERATION PROGRAMS.....	1
a. General.....	1
b. File Organization.....	2
c. Program Descriptions.....	2
(1) SRC Program.....	2
(2) UNIT Program.....	2
(3) PARENT Program.....	4
(4) FORCE Program.....	4
4. JIFFY GAME.....	5
a. General.....	5
b. Program Descriptions.....	5
(1) Program SUPER.....	5
(2) Pre-Assessment Subroutines.....	8
(3) Assessment Subroutines.....	8
(4) Record Keeping Subroutines.....	8
(5) Interactive Interface Subroutines.....	8
(6) Functions.....	9
(7) Postprocessor Subroutines.....	9
(8) Program Listings.....	9
REFERENCES.....	10
APPENDIXES	
A. INDEXED SEQUENTIAL FILE CREATION PROGRAMS.....	A-1
B. SRC PROGRAM.....	B-1
C. UNIT PROGRAM.....	C-1
D. PARENT PROGRAM.....	D-1
E. FORCE PROGRAM.....	E-1
F. PROGRAM SUPER.....	F-1
G. PRE-ASSESSMENT SUBROUTINES	
I EW.....	G-1
II INIT.....	G-4
III ROFA.....	G-6
IV SUPRES.....	G-16
H. ASSESSMENT SUBROUTINES	
I AHAD.....	H-1
II CANNON.....	H-16
III CLGP.....	H-27
IV FASCAM.....	H-31
V INFANT.....	H-35
VI MINE.....	H-40
VII TANK.....	H-47

TABLE OF CONTENTS (continued)

	Page
I. RECORD KEEPING SUBROUTINES	
I APPORT.....	I-1
II BUILD.....	I-8
III CKACI.....	I-9
IV CKSTOR.....	I-11
V CLEAR.....	I-13
VI COMBINE.....	I-15
VII DISPLAY.....	I-18
VIII ENDCI.....	I-24
IX FORCE.....	I-26
X LOSS.....	I-46
XI NEWHTS.....	I-48
XII RESET.....	I-50
XIII RESTART.....	I-53
XIV SHOWCI.....	I-55
J. INTERACTIVE INTERFACE SUBROUTINES	
I INTERACTIVE INTERFACE OVERVIEW.....	J-1
II SUBROUTINE DESCRIPTION AND VARIABLES.....	J-3
K. FUNCTIONS	
I INDEX 5.....	K-1
II NUMB.....	K-3
L. POSTPROCESSED SUBROUTINES	
I GENERAL DESCRIPTION.....	L-1
II SUBROUTINES DESCRIPTION.....	L-1
III COMMON ARRAYS AND VARIABLES.....	L-3
IV SAMPLE OUTPUTS FROM THE POSTPROCESSOR.....	L-7
M. JIFFY III PROGRAM LISTINGS	
I GENERAL DESCRIPTION.....	M-1
II INDEX FOR JIFFY III PROGRAMS AND SUBROUTINES (FEBRUARY 1980 VERSION).....	M-2
III PROGRAM LISTINGS.....	M-4
N. DISTRIBUTION LIST.....	N-1

LIST OF TABLES

	Page
1. Control point gamer decision	7

LIST OF FIGURES

	<u>Page</u>
1. Jiffy Game file formats -----	3
2. Jiffy Game functional flow diagram -----	6
A-1. Create program for SRC file -----	A-1
A-2. Create program for UNIT file -----	A-1
A-3. Create program for PARENT file -----	A-2
A-4. Create program for FORCE file -----	A-2
A-5. Create program for HISTORY file -----	A-3
B-1. SRC program flow diagram -----	B-2
C-1. UNIT program flow diagram -----	C-2
D-1. PARENT program flow diagram -----	D-2
E-1. FORCE program flow diagram -----	E-2
F-1. SUPER flow diagram -----	F-3
G-1. EW flow diagram -----	G-2
G-2. INIT flow diagram -----	G-5
G-3. ROFA flow diagram -----	G-7
G-4. SUPRES flow diagram -----	G-17
H-1. AHAD flow diagram -----	H-5
H-2. CANNON flow diagram -----	H-19
H-3. CLGP flow diagram -----	H-28
H-4. FASCAM flow diagram -----	H-32
H-5. INFANT flow diagram -----	H-36
H-6. MINE flow diagram -----	H-42
H-7. TANK flow diagram -----	H-50
I-1. APPORT flow diagram -----	I-3
I-3. CKACI flow diagram -----	I-10
I-4. CKSTOR flow diagram -----	I-12
I-5. CLEAR flow diagram -----	I-14
I-6. COMBINE flow diagram -----	I-16
I-7. DISPLAY flow diagram -----	I-20
I-8. ENDCI flow diagram -----	I-25
I-9. FORCE flow diagram -----	I-28
I-10. LOSS flow diagram -----	I-47
I-11. NEWHIS flow diagram -----	I-49
I-12. RESET flow diagram -----	I-51
I-13. RESTART flow diagram -----	I-54
I-14. SHOWCI flow diagram -----	I-56
J-1. Interactive interface overview flow diagram -----	J-2
K-1. INDEX 5 flow diagram -----	K-2
K-2. NUMB flow diagram -----	K-4
L-1. Postprocessor overview flow diagram -----	L-6

CACDA JIFFY III WAR GAME PROGRAMMERS MANUAL

1. SCOPE. This manual was prepared to document the computer programs associated with the CACDA "Jiffy" war gaming process. The documentation of each subroutine and program includes a discussion of the functions performed by the routine. Descriptions, logic flow diagrams and a list of variables are provided in appendixes. The entire Jiffy listing of the FORTRAN code is given in appendix M. This program was subjected to extensive review and debugging in December 1979. No significant model changes were required as a result of the sensitivity analysis and review. The program listings are from a printout dated April 1980.

2. GENERAL. The interactive programs and data files that support the CACDA "Jiffy" war gaming process reside in permanent file storage on the Control Data Corporation (CDC) 6400/6500 multiprocessor computer located at Fort Leavenworth, Kansas. The programs are written in FORTRAN and are machine dependent due to extensive use of CDC Extended FORTRAN file handling features. There are basically two groups of programs that support the CACDA "Jiffy" war gaming process:

- o a set of four programs that generate and maintain files necessary for force structure
- o create programs that initialize force structure generation files and history files
- o the Jiffy Game combat assessment program.

The four force structure generation programs are small programs that allow the gamers to build interactively a hierarchy of files based on the Army's concept of Tables of Organization and Equipment (TOE) with which they can generate task organized forces for combat assessments in the Jiffy Game. The Jiffy Game operates on these forces and determines the number of personnel casualties and weapon system losses each force suffers in combat. In addition, the Jiffy Game generates a file containing a history of the forces and the losses they incurred for the combat it has processed.

3. FORCE STRUCTURE GENERATION PROGRAMS.

a. General. A hierarchy of four interactive programs has been developed to provide nontechnical military personnel with the capability to develop systematically a set of data files from which they can generate task organized forces for assessment evaluation in the Jiffy Game. The force structure generation is based on the US Army TOE standard requirements codes (SRCs). The SRCs define the types and quantities of weapon systems found in specific subunit organizations; e.g., an infantry squad or a tank platoon. The first program of the force generation

hierarchy interactively develops a data base file of SRCs for each force. Since there is little variation in the composition of these subunit SRCs, the SRC data base, once completed, will be readily available for immediate application to any Jiffy Game-supported study. The second of the force generation programs uses the SRC data base to build interactively a file of the combat units through specification of a unique name and all SRCs that compose each unit. The file of units is then task organized into higher echelon organizations called parent units. A file of the parent units is created interactively by the third program of the hierarchy. Finally, the information on the SRC, unit, and parent unit files is consolidated into a file of the forces to be considered for combat assessments in the Jiffy Game.

b. File Organization. The type of files used in the force structure generation process and the Jiffy Game HISTORY file are CDC index sequential-random access files. These files are created and manipulated by file handling macros unique to the CDC operation systems. The files used for this application are random access files whose keys are contained in the first 20 characters (two words) of the record (the HISTORY file uses 30 character keys). The keys are arranged in sequential order in the random access index table, which allows sequential, in addition to random, accessing of the records on the file. The record formats for the four force generation files and the HISTORY file are illustrated in figure 1. Before any operations may be performed on these files, they must be created and put into permanent file storage space. This initialization process is accomplished through the execution of a small file creation program, which simply specifies the parameters essential for proper file definition. The FORTRAN programs for the creation of all five index sequential-random access files are contained in appendix A to this volume.

c. Program Descriptions.

(1) SRC program. The SRC program interactively builds the TOE SRC data base file. As noted above, this file is an indexed sequential-random access file. Each record of the SRC file contains an SRC identification word (1 to 10 alphanumeric characters) and up to 22 groups of weapon system item codes (Jiffy III Game Volume III, Appendix A) and the quantity of each type of weapon system assigned to the SRC. The format of the records of the SRC file is illustrated in figure 1(a). In addition to creating the SRC data base file, the SRC program has the capability to review any SRC that exists in the data base, add new SRCs to the file, change the quantity and/or type of personnel or weapon systems in a given SRC, delete specified SRCs, and list all SRCs with the quantity and type of weapon systems and personnel found in them. A logic flow diagram of the SRC program, a listing of the program variables, and a list of the program code are provided in appendix B to this volume.

FORCE COLOR R/B	SRC #	ITEM CODE #1	QTY OF ITEM CODE #1	ITEM CODE #2	QTY OF ITEM CODE #2	...	...	...	QTY OF ITEM CODE #22

(a) SRC File Record

FORCE COLOR R/B	UNIT ID	SRC #1	SRC #2	...	...	...	...	SRC #22

(b) UNIT File Record

FORCE COLOR R/B	PARENT ID	UNIT ID #1	UNIT ID #2	...	...	...	...	UNIT ID #18

(c) PARENT File Record

PARENT ID	UNIT ID	FORCE COLOR	SECTOR	CRITICAL INCIDENT	FPS AT 100% STRENGTH	PERMANENT		TEMPORARY	
						COMBAT INTENSITY LEVEL	PERCENT STRENGTH	COMBAT INTENSITY LEVEL	EM TYPE...

QTY OF ITEM CODE #1	QTY OF ITEM CODE #2	QTY OF ITEM CODE #3	...	...	...	...	...	...	QTY OF ITEM CODE #80

(d) FORCE File Record

CRITICAL INCIDENT	PARENT ID	UNIT ID	SECTOR	FORCE COLOR	WORDS 6 TO 90 SAME AS FORCE FILE.				

(e) HISTORY File Record

Figure 1. Jiffy Game file formats.

(2) UNIT program. Execution of the UNIT program is the second step in the force structure generation process. The UNIT program accesses the information stored on the SRC file and defines the combat units to be gamed. The program builds an indexed sequential-random access file whose records correspond to the combat units. The format of the UNIT file records is given in figure 1(b). Each record contains the unit name (1 to 10 alphanumeric characters) and up to 22 valid SRCs (the SRCs must exist on the SRC file). The SRCs specified with a unit correspond to the subunit organizations that compose the unit. For example, the SRCs specified for a tank company could possibly be a tank platoon SRC (specified three times) and a tank company headquarters SRC. In addition to building the UNIT file, the UNIT program has the capability to review the SRCs in a unit already on file, add more SRCs to a unit already on file, delete specified units, delete particular SRCs from specific units, and list all units with their SRCs. A logic flow diagram of the UNIT program, a listing of the program variables and the program code are provided in appendix C to this volume.

(3) PARENT program. The PARENT program is the third part of the force structure generation process. The PARENT program is the tool with which the military gamers can task organize interactively the combat units previously defined on the UNIT file into a file of higher echelon organizations, or parent units. The parent units are created by the program with the definition of a unique parent unit name (1 to 10 alphanumeric characters) and the specification of up to 18 valid units within its organization. The format for the records of the PARENT file is illustrated in figure 1(c). In addition to creating the PARENT file, the program may be used to review the units of parent units already on file, add new parent organizations to the file, add new units to existing parent units, delete specified parent units, delete given units of specific parent units, and list all parent organizations with their subordinate units. A logic flow diagram of the PARENT program, the PARENT program variables, and program listing are provided in appendix D to this volume.

(4) FORCE program. The FORCE program, the final step in the force structure process, interactively creates a file of the forces to be assessed in the combat routines of the Jiffy Game. The FORCE program consolidates the information defined on the files in the previous three steps of the process. The FORCE file consists of records for each unit of both forces. The format of the records of the FORCE file is presented in figure 1(d). The first 10 words of the record define the unit and its combat environment. Although some of the parameters (sector, critical incident, combat intensity) are redefined in the Jiffy Game during the actual gaming, the first 10 words are initialized in the FORCE program. The remaining 80 words (words 11 to 90 on the record) contain the quantity and indicate the type of weapon system in the unit. The position of the word denotes the type of weapon system (item code equals record word number minus 10). The value of the word is the quantity of that type of weapon system. Besides generating the FORCE file, the FORCE program provides the capabilities to add units of a specified new parent unit to the force file using information

stored in the other three files, delete all the units of a specific parent unit to the file, change the unit effectiveness, combat intensity level, and type (AD, AH, artillery, or maneuver), of any unit on the file, and list all parent units with their subordinate units, and their subordinate units and their corresponding quantities of weapon systems. It should be noted that when a unit is added to the file, the gamer is asked to input its unit effectiveness, which is the percentage of a unit's existing firepower score compared to its 100 percent firepower score. The number of each type of weapon system loaded into a unit equals the number of that weapon allocated to the unit at 100 percent strength multiplied times the unit's effectiveness. For example, if a unit had 16 tanks at 100 percent strength and it was specified to be a 50 percent unit effectiveness, only eight tanks would be loaded into the unit. A logic flow diagram of the FORCE program, a list of the program variables, and program code are provided in appendix E to this volume.

4. JIFFY GAME.

a. General. The Jiffy Game is a two-sided, interactive war game that operates on the FORCE file, the product of the force structure generation process, and determines potential rates of advance, personnel casualties and weapon system losses incurred by the units of the two forces on the FORCE file as a result of the five types of combat it plays: indirect fire, minefields, armor/ antiarmor, dismounted infantry, and attack helicopter/air defense. In addition to assessing combat, the Jiffy Game handles other administrative functions associated with the war game, such as combat loss apportionment, maintaining the FORCE file, updating the HISTORY file as required, and outputting the statistics of the battles. The Jiffy Game is written in FORTRAN and has utilized some of the features of CDC Extended FORTRAN. The program has been segmented to fit into 150k words of core on the CDC 6400 for interactive processing. The CPU processing time under the NOS/BE operating system varies with the size of the forces being gamed, but typical times vary between 10 and 60 CPU seconds per sector of combat games.

b. Program Descriptions. A functional flow diagram of program SUPER representing the overall Jiffy Game, is presented in figure 2. The following paragraphs describe each subroutine of the Jiffy Game and discuss the functions performed by the routines.

(1) Program SUPER. Program SUPER is the main program of the Jiffy game. The primary function of the main program is to serve as a control point from which a gamer can branch to the other subroutines. During execution the gamer resides at a control point known as the DECISION POINT. At this point, the gamer has a choice of the 12 decisions presented in table 1. Each gamer decision causes SUPER to branch according to the

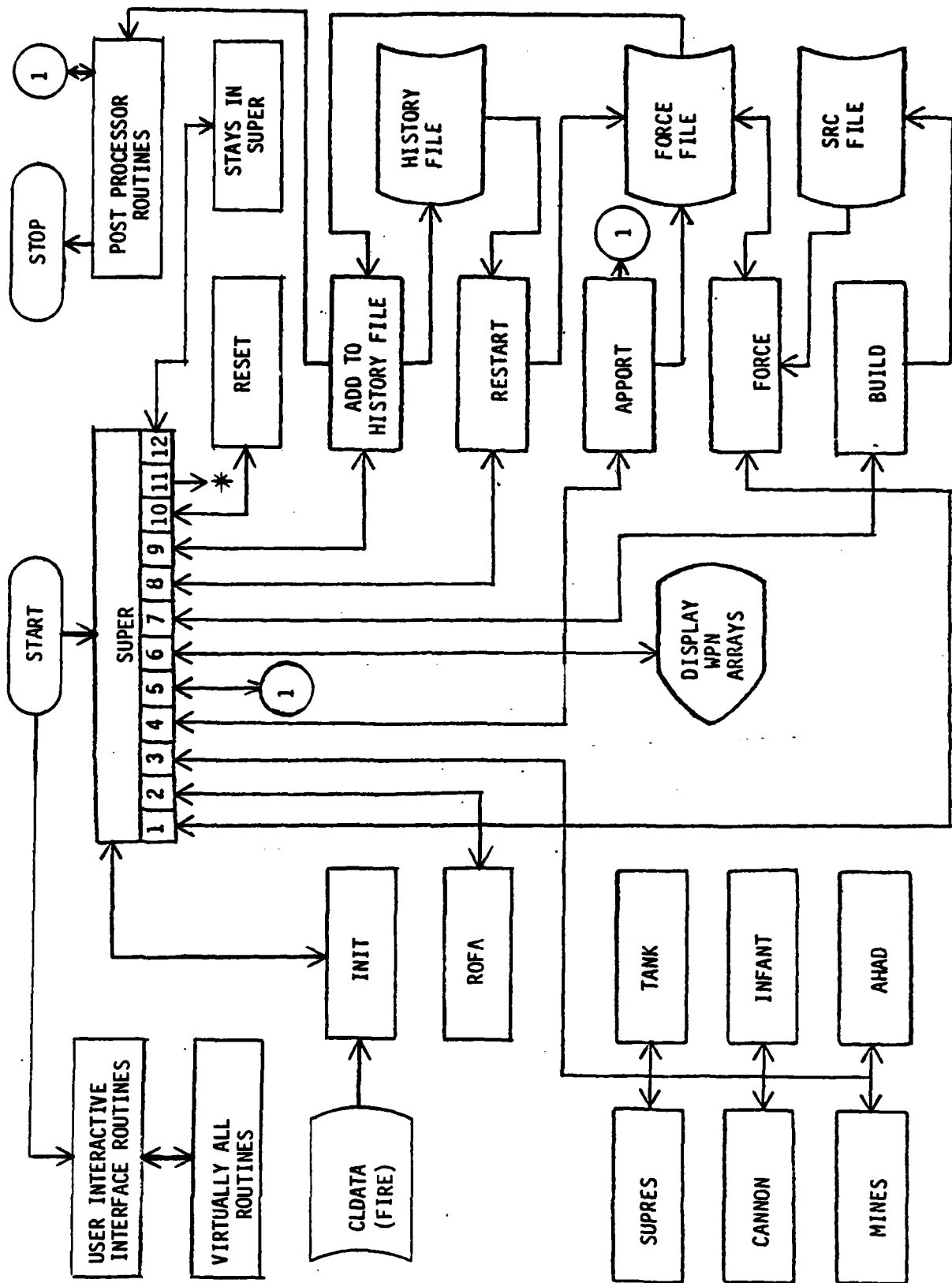


Figure 2. Jiffy game functional flow diagram. * See Super flow diagram.

Table 1. Control point gamer decisions.

Number	Description of Decision
1	Load forces into a sector
2	Calculate rate of advance
3	Assess combat
4	Apportion combat losses to units
5	Output battle statistics
6	Display weapon arrays
7	Add SRCs to TOE file
8	Restart at a previously gamed CI
9	Update history file - end game
10	Reset Element Array from input file
11	Review previous run
12	Reset terminal output file

flow diagram, as shown in appendix F, and return to the DECISION POINT. In addition, SUPER performs the following functions:

- o Calls INIT for data and array initialization.
- o Displays the game instructions, if requested.
- o Inspects the type of weapon systems being played and determines the appropriate combat assessment routine to which to branch.
- o Accepts inputs of TACAIR losses, which are determined external to the Jiffy Game.
- o Records the forces remaining at the end of a critical incident on the HISTORY file.
- o Outputs the cumulative battle statistics at the end of critical incident.
- o Provides the gamers with the capability to reinitialize the forces at some previously gamed critical incident on the HISTORY file.

The program logic flow, a list of program variables, and program codes are provided in appendix F.

(2) Pre-Assessment subroutines. Four subroutines are grouped in this category because they perform pre-assessment functions. These subroutines initialize the array in data, and determine the overall and degraded fire power, artillery missions, and suppression factors. Other major functions are to determine and display for the sector being gamed, the rate of advance, the length of the battle and fire power ratio, etc. Four subroutines are EW, INIT, ROFA, and SUPRES. The description, program logic flow, and a list of program variables, of all pre-assessment subroutines are provided in appendix G.

(3) Assessment subroutines. Seven subroutines are grouped in the assessment category because they are performed as combat assessment functions. These are AHAD, CANNON, CLGP, FASCAM, INFANT, MINE and TANK. The description, program logic flow, and a list of program variables are provided in appendix H.

(4) Record-keeping subroutines. Fourteen subroutines are grouped in this category because they apportion casualties in the Jiffy combat assessment routines to the units on the FORCE file. These subroutines are: APPORT, BUILD, CKACI, CKSTOR, CLEAR, COMBINE, DISPLAY, FNDICI, FORE, LOSS, NEWHIS, RESET, RESTART and SHOWCI. The description, program logic flow, and list of program variables of all record-keeping subroutines are provided in appendix I.

(5) Interactive interface subroutines. Ten subroutines are grouped in this category because they perform interactive functions. These functions are to write on the ANSWER file and REVIEW files. It checks for the end of ANSWER file and informs the gamer. Ten subroutines are CKANS, GFT4, HOW, JOHN, REEDA, REED4, REVIEW, SHORT, SHOWIT and TERM. The description, an overview flow diagram, and a list of program variables are provided in appendix J.

(6) Functions subroutines. Two subroutines are grouped in this category because they perform the subfunction used to convert a five subscript variable to a single subscript variable, program variables. These are INDEXS and NUMB. The description, a logic flow, and list of program variables are provided in appendix K.

(7) Postprocessor subroutines. Nineteen subroutines are grouped in this category because they produce specified game output reports as well as cumulative output reports and the killer victim matrix. These subroutines are ACCUM, BULLET, CIARM, COMMIT, GITNMS, HEADER, KILCAT, KILVIC, NEWPAG, OPTLER, POST, RATIOS, RECVRY, REPORT, SUMIOT, SUMUP, TRACE, TRANS and XOUT. The description, an overview logic flow, list of program variables and a sample output from the post processor are provided in appendix L.

(8) Program listings. Appendix M contains the complete Jiffy III program listings. This program has been debugged and reviewed several times. A sample output run is shown in appendix L. The general program structure is the same as for the original 1977 version, although many improvements and modifications have been made as discussed in volume II. Users should note that most of the unclassified data are hardwired in the program code; so each time data changes are made, changes are required in the program. The Jiffy model continues to be modified and changed as new systems are developed and incorporated in the program. The complete program listings in appendix M are the latest program as of February 1980.

REFERENCES

1. USA CACDA, CACDA Jiffy War Game Programers Manual, Technical Report TR 3-77, March 1977.

APPENDIX A
INDEXED SEQUENTIAL FILE CREATION PROGRAMS

APPENDIX A

INDEXED SEQUENTIAL FILE CREATION PROGRAMS

This appendix contains the program code listings of the five programs used to create the indexed sequential-random access files used in the CACDA "Jiffy" War Gaming process. The FORTRAN code listings are presented in figures A-1 through A-5 for the SRC, UNIT, PARENT, FORCE, and HISTORY files, respectively.

Figure A-1. Create program for SRC file.

```
PROGRAM CREATE(INPUT,OUTPUT)
DIMENSION IFIT(35),IARRAY(46)
CALL FILEIS(IFIT,3LLFN,5LTAP9,3LWSA,IARRAY,3LMNR,460,2LRL,460,3LMRL,
460,2LKA,IARRAY(1),2LKP,0,2LKL,20,3LDKI,2LNO)
CALL STOREF(IFIT,3LERL,100)
CALL OPENM(IFIT,3LNEW)
CALL PUT(IFIT)
CALL CLOSEM(IFIT)
STOP 123
END
```

Figure A-2. Create program for UNIT file.

```
PROGRAM CREATE (INPUT,OUTPUT)
DIMENSION IFIT(35),IARRAY(24)
CALL FILEIS(IFIT,3LLFN,5LTAP9,3LWSA,IARRAY,3LMNR,240,2LRL,240,3LMRL,
240,2LKA,IARRAY(1),2LKP,0,2LKL,20,3LDKI,2LNO)
CALL STOREF(IFIT,3LERL,100)
CALL OPENM(IFIT,3LNEW)
CALL PUT(IFIT)
CALL CLOSEM(IFIT)
STOP 123
END
```

Figure A-3. Create program for PARENT file.

```
PROGRAM CREATE(INPUT,OUTPUT)
DIMENSION IFIT(35),IARRAY(20)
CALL FILEIS(IFIT,3LLFN,5LTAP9,3LWSA,IARRAY,3LMNR,200,2LRL,200,3LMRL,
200,2LKA,IARRAY(1),2LKP,0,2LKL,20,3LDKI,2LNO)
CALL STOREF(IFIT,3LERL,100)
CALL OPENM(IFIT,3LNEW)
CALL PUT(IFIT)
CALL CLOSEM(IFIT)
STOP 123
END
```

Figure A-4. Create program for FORCE file.

```
PROGRAM CREATE(INPUT,OUTPUT)
DIMENSION IFIT(35),IARRAY(90)
CALL FILEIS(IFIT,3LLFN,5LTAP9,3LWSA,IARRAY,3LMNR,900,2LRL,900,3LMRL,
900,2LKA,IARRAY(1),2LKP,0,2LKL,20,3LDKI,2LNO)
CALL STOREF(IFIT,3LERL,100)
CALL OPENM(IFIT,3LNEW)
IARRAY(1)=IARRAY(2)=IARRAY(5)="INITIAL"
IARRAY(4)=0
IARRAY(3)="B"
DO 10 I=6,90
10 IARRAY(I)=0
CALL PUT(IFIT,IARRAY,900,IARRAY(1))
CALL CLOSEM(IFIT)
STOP 123
END
```

Figure A-5. Create program for HISTORY file.

```
PROGRAM CREATE(INPUT,OUTPUT)
DIMENSION IFIT(35),IARRAY(90)
CALL FILEIS(IFIT,3LLFN,5LTAPE9,3LWSA,IARRAY,3LMNR,900,2LRL,900,3LMRL,
900,2LKA,IARRAY(1),2LKP,0,2LKL,30,3LDKI,2LNO)
CALL STOREF(IFIT,3LERL,100)
CALL OPENM(IFIT,3LNEW)
IARRAY(1)="INITIAL"
IARRAY(2)="INITIAL"
IARRAY(3)="INITIAL"
DO 10 I=4,90
10 IARRAY(I)=0
CALL PUT(IFIT,IARRAY,900,IARRAY(1))
CALL CLOSEM(IFIT)
STOP 123
END
```

APPENDIX B
SRC PROGRAM

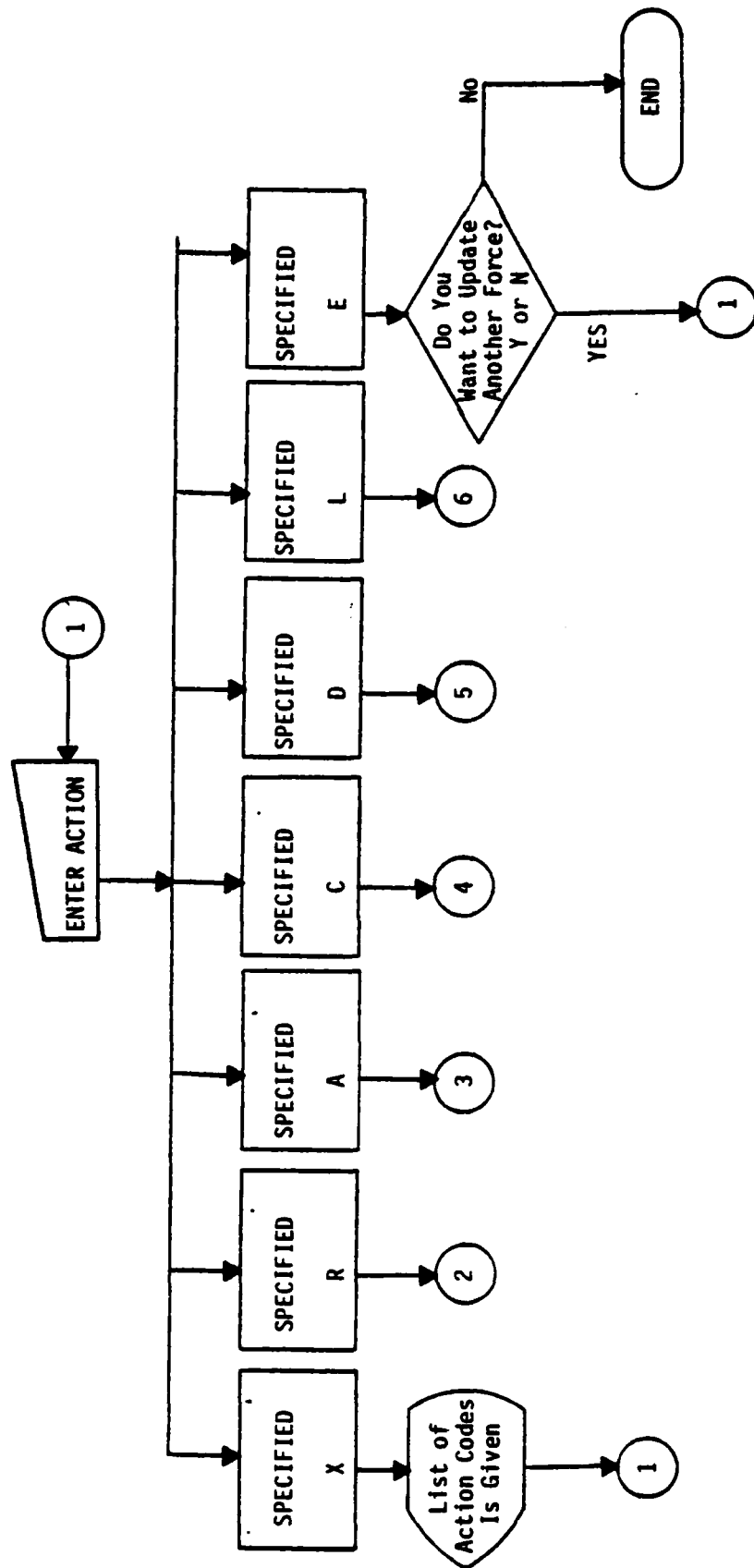
APPENDIX B

SRC PROGRAM

B-1. PROGRAM DESCRIPTION. The SRC program interactively builds the TOE SRC data base file, which contains an identification word and the quantity of each type of weapon system assigned to the SRC. The logic flow diagram, a list of variables, and a listing of FORTRAN codes of the SRC program are given in this appendix.

B-2. SUBROUTINE VARIABLE.

<u>Variable</u>	<u>Description</u>
ACHG	Weapons to change (delete)
AFRR	Weapons not found to change (delete)
AHOLD	Keeps the force type
AJ	Quantity of weapon to be added
AM	Weapon to be added
ARRAY	Work storage array (SRC File)
ARRAY (1)	Force type (key)
ARRAY (2)	SRC name (key)
ARRAY (3)	First weapon on record
ASRC	SRC name specified
ICK	Action code
IDO	Weapon listed
IFND	Number of weapons to be changed (deleted)
IFIT	FIT array (SRC File)
K	Number of weapons not found
NN	Weapon position on record
NY	Answer to question



B-2

Figure B-1 SRC program logic flow diagram. (continued next page)

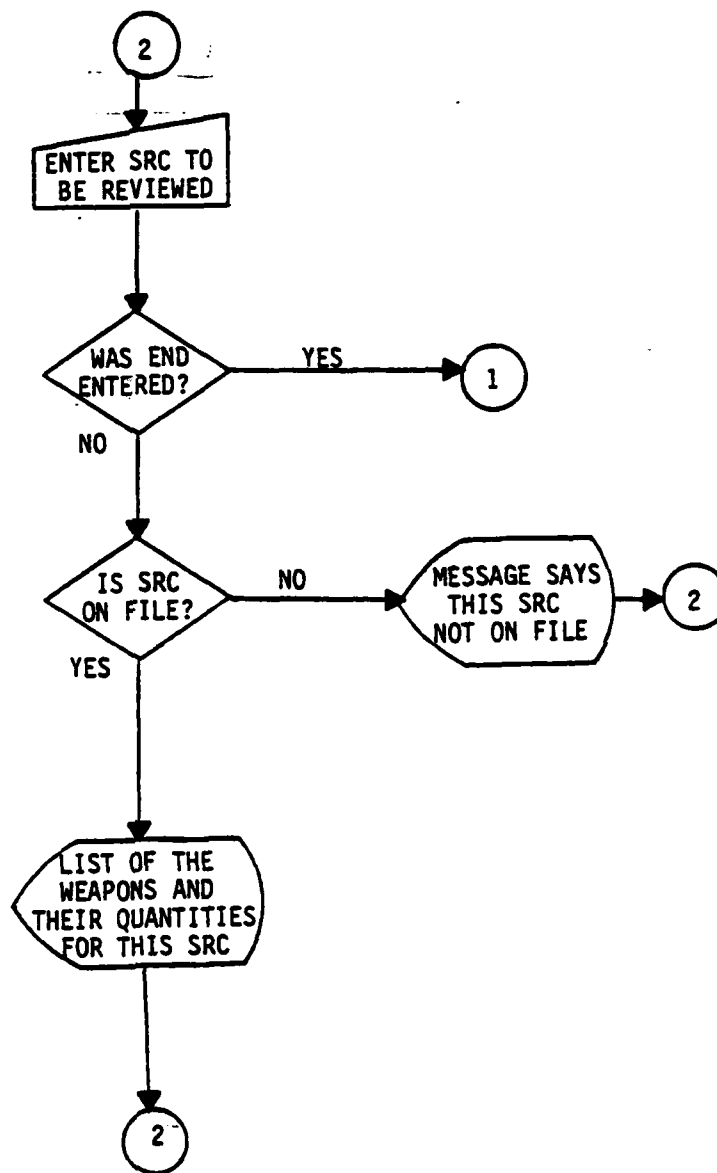


Figure 3-1 SRC program logic flow diagram (continued).

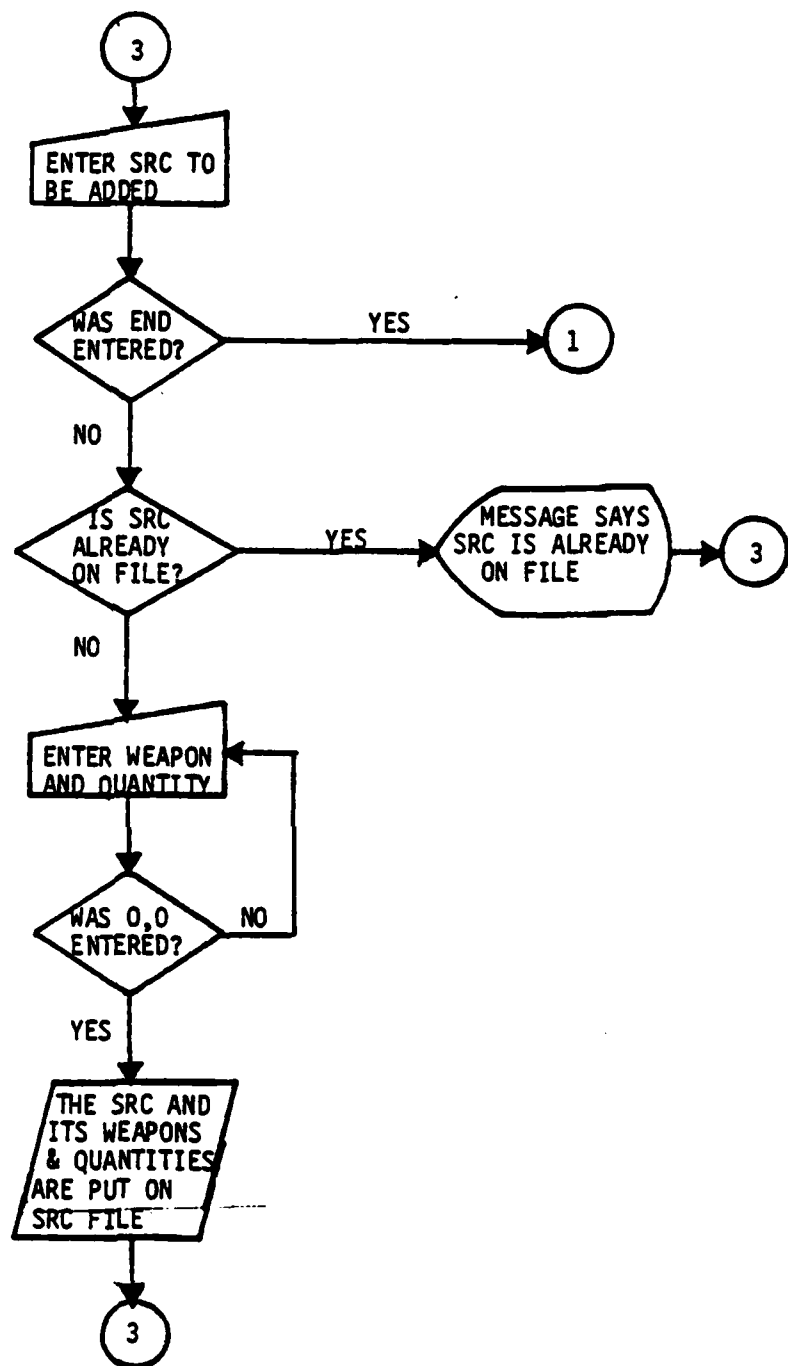


Figure B-1 SRC program logic flow diagram (continued).

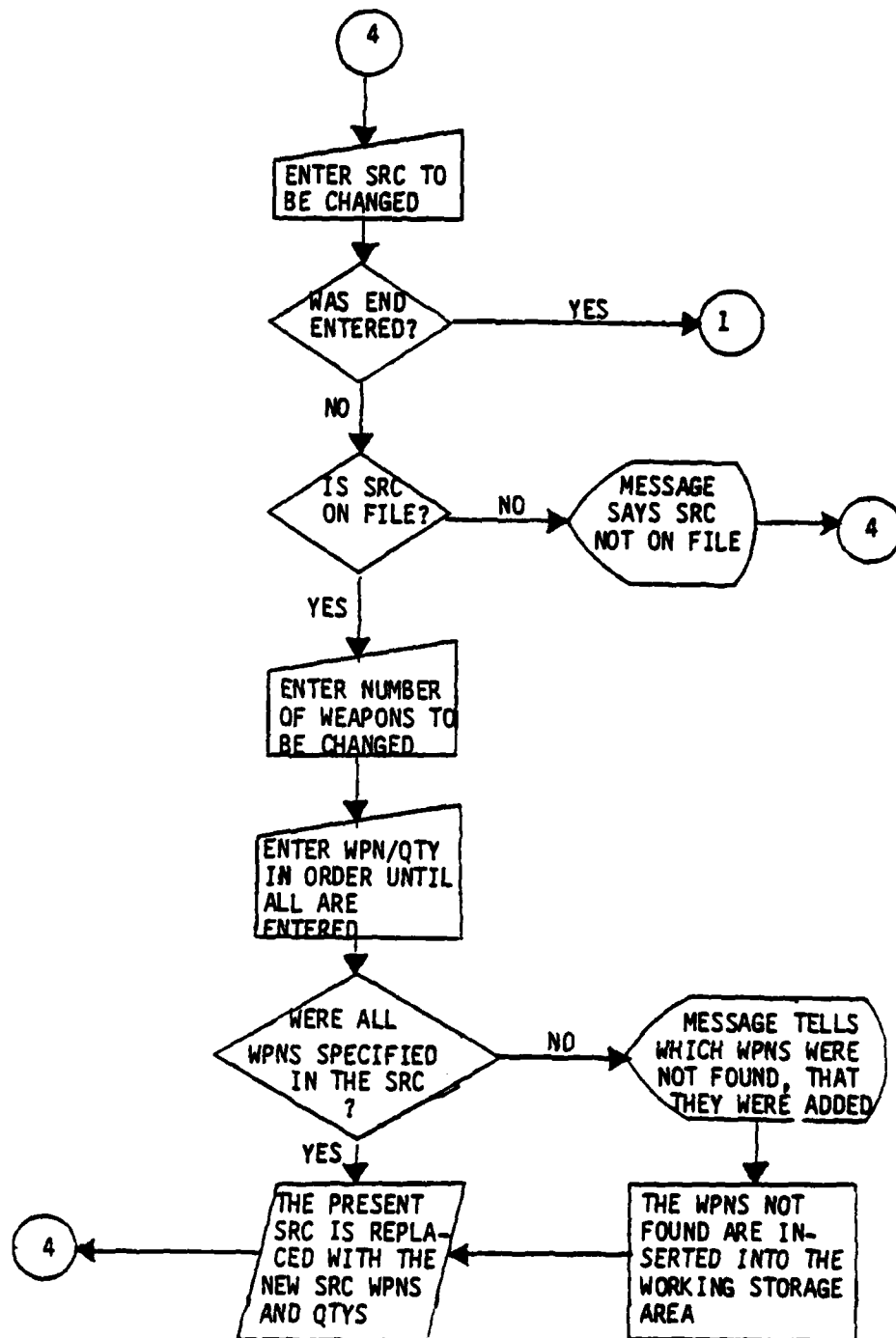


Figure B-1 SRC program logic flow diagram (continued).

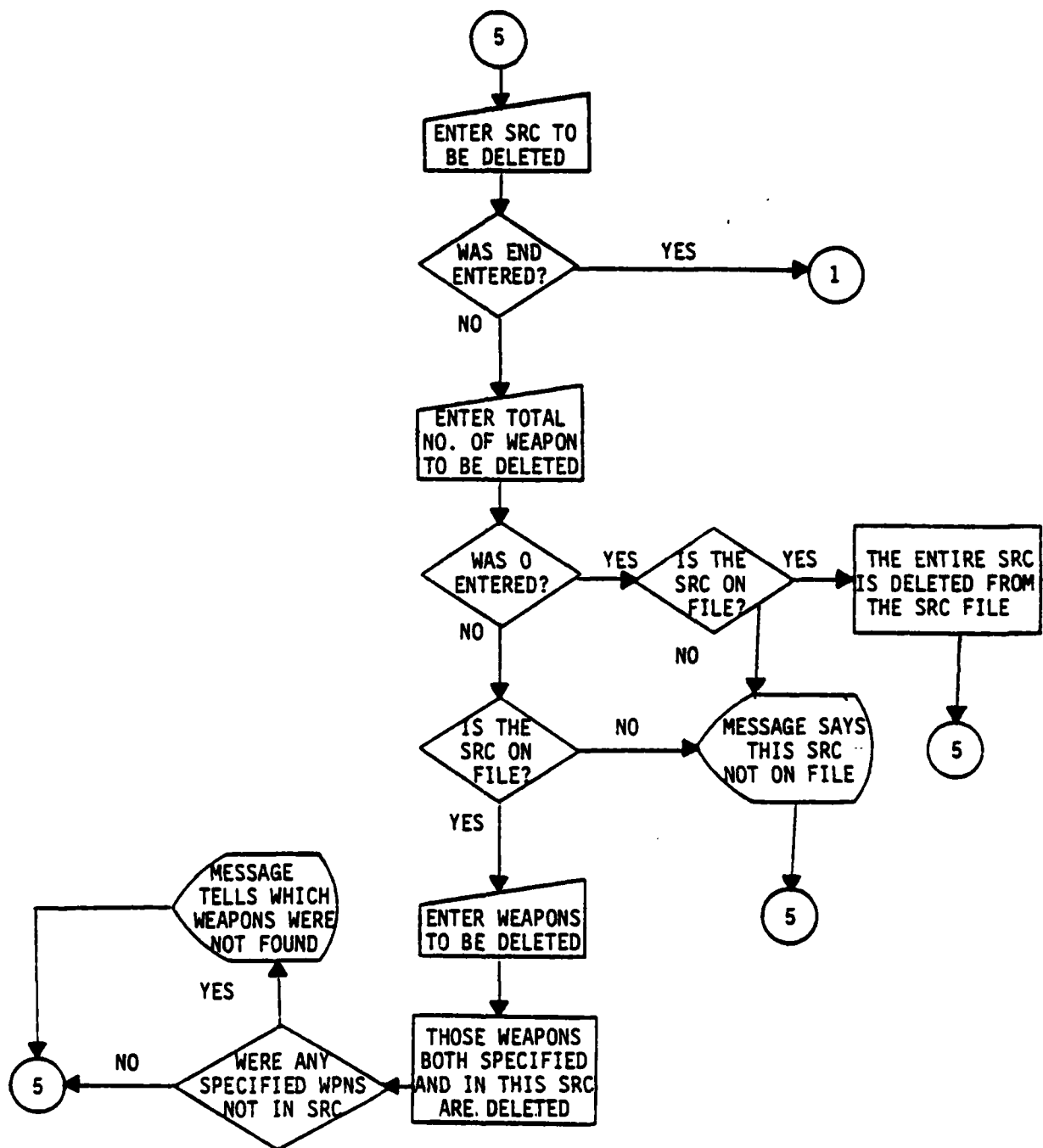


Figure B-1 SRC program logic flow diagram (continued).

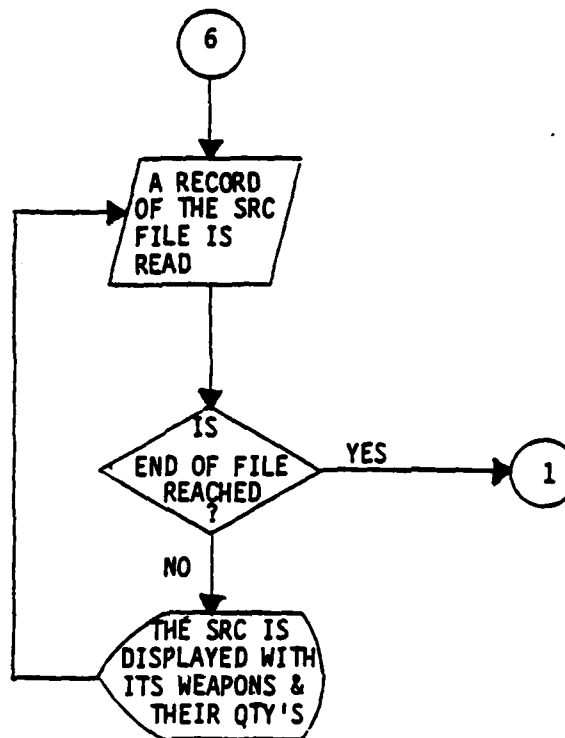


Figure B-1 SRC program logic flow diagram (concluded).

APPENDIX C
UNIT PROGRAM

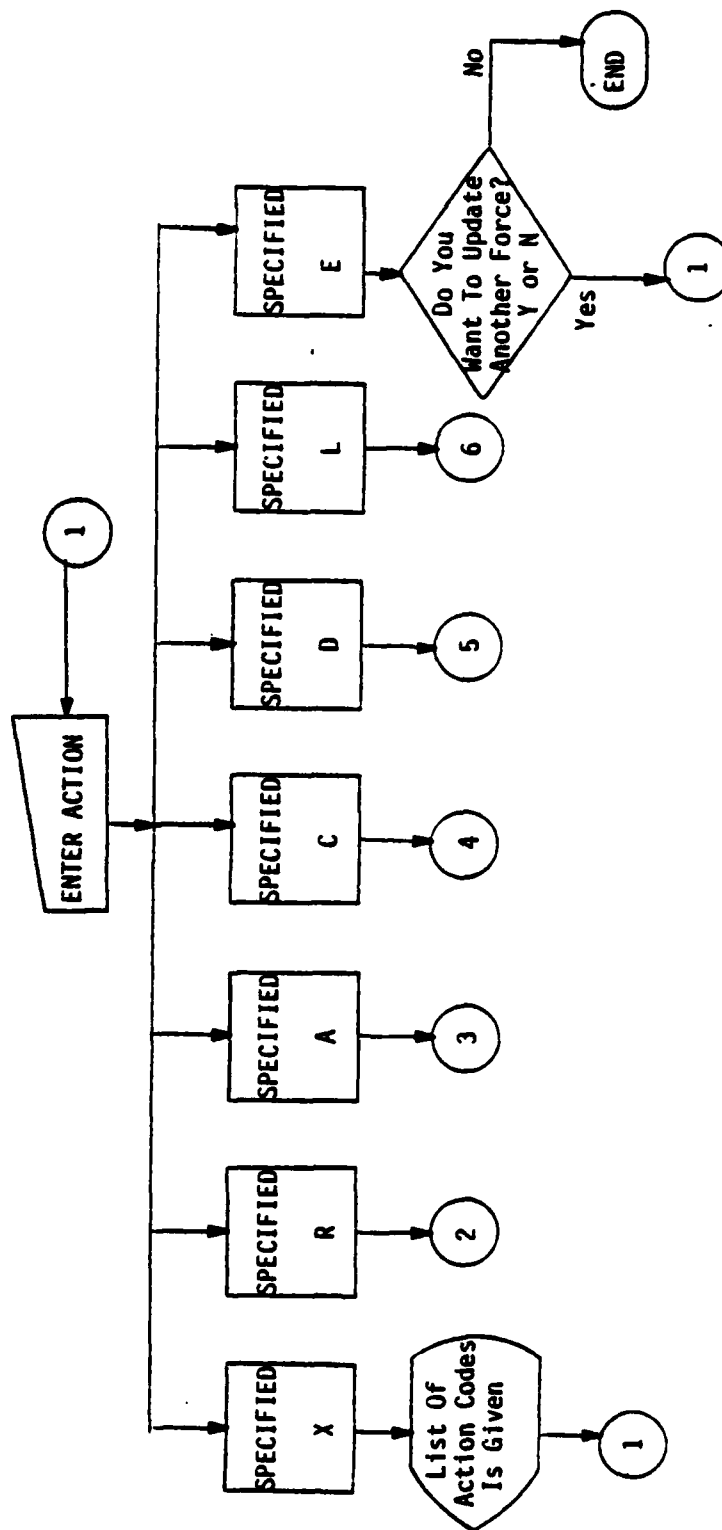
APPENDIX C

UNIT PROGRAM

C-1. SUBROUTINE DESCRIPTION. The UNIT program builds an indexed sequential-random access file whose records correspond to the combat units, and accesses the information stored on the SRC file. A list of variables and the logic flow diagram are given in this appendix.

C-2. Program Variables.

<u>Variable</u>	<u>Description</u>
AERR	SRCs already existing in unit
ARRAY	Work storage area (SRC File)
ARRAY (1)	Force type (key)
ARRAY (2)	SRC name (key)
ARRAY (3)	First weapon on record
BBERR	SRCs that do not exist
BCHG	SRCs to be added (deleted)
BERR	SRCs to be added
BFRG	SRC not on file
BHOLD	Keeps the force type
BRRAY	Work storage area (Unit File)
BRRAY (1)	Force type (key)
BRRAY (2)	Unit name (key)
BRRAY (3)	First SRC on record
BUNIT	Unit name specified
I	SRC position on record
ICK	Action Code
IFIT	FIT array (SRC File)
K	Number of SRCs not found
M	Number of SRCs to be added
N	Number of SRCs to be added
NEND	Number of SRCs to be changed (deleted)
NFIT	FIT array (Unit File)
NN	Number of SRCs that do not exist
NY	Answer to question



C-2

Figure C-1 UNIT program logic flow diagram. (Continued next page)

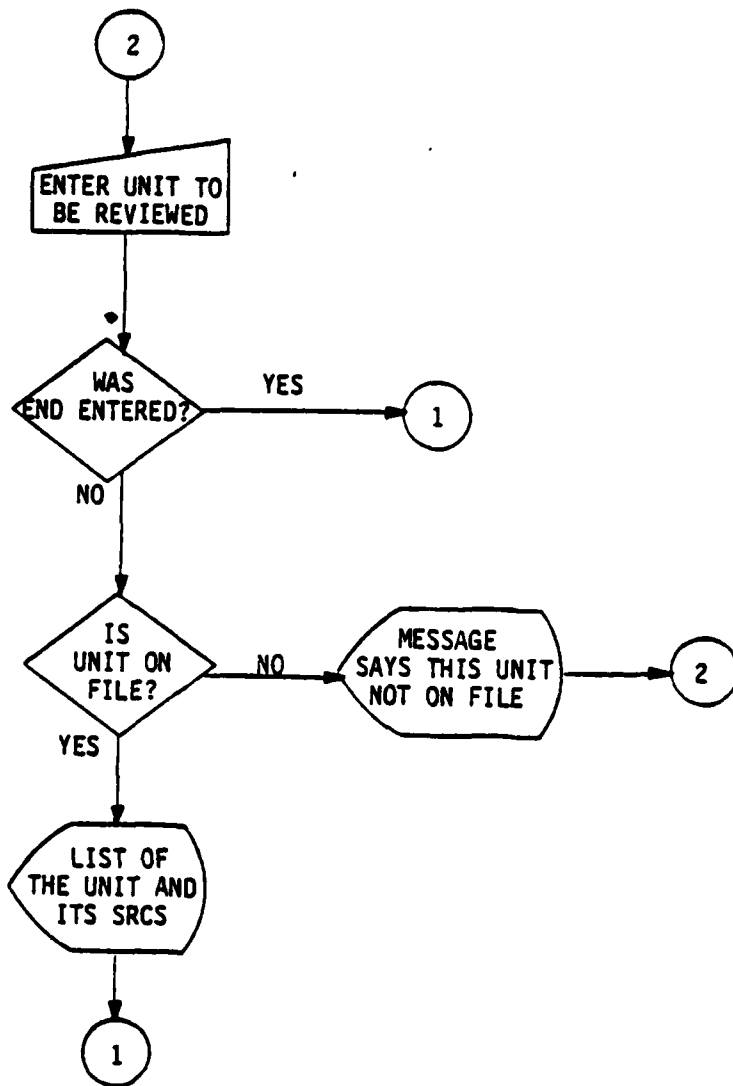


Figure C-1 UNIT program logic flow diagram
(continued).

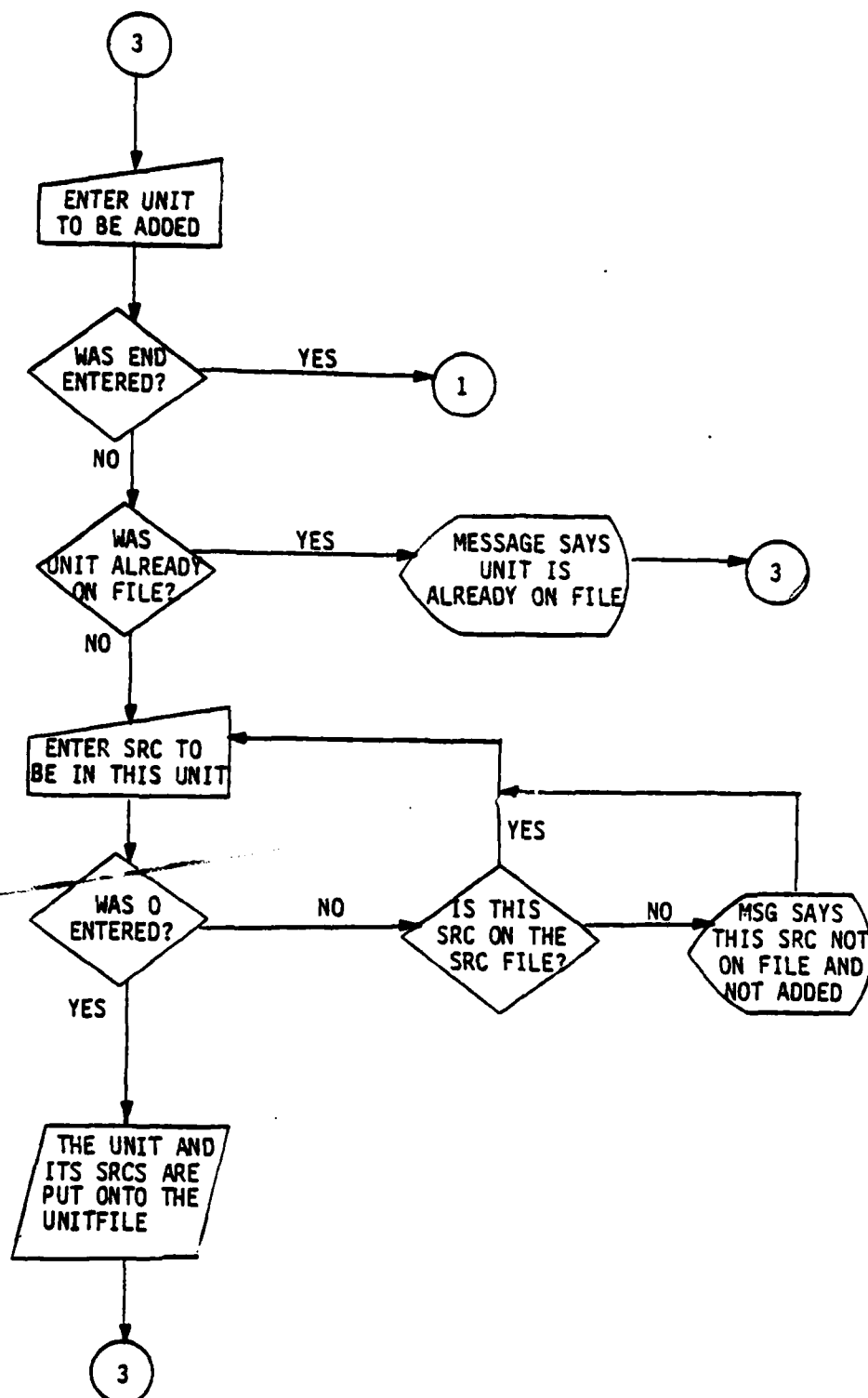


Figure C-1 UNIT program logic flow diagram (continued).

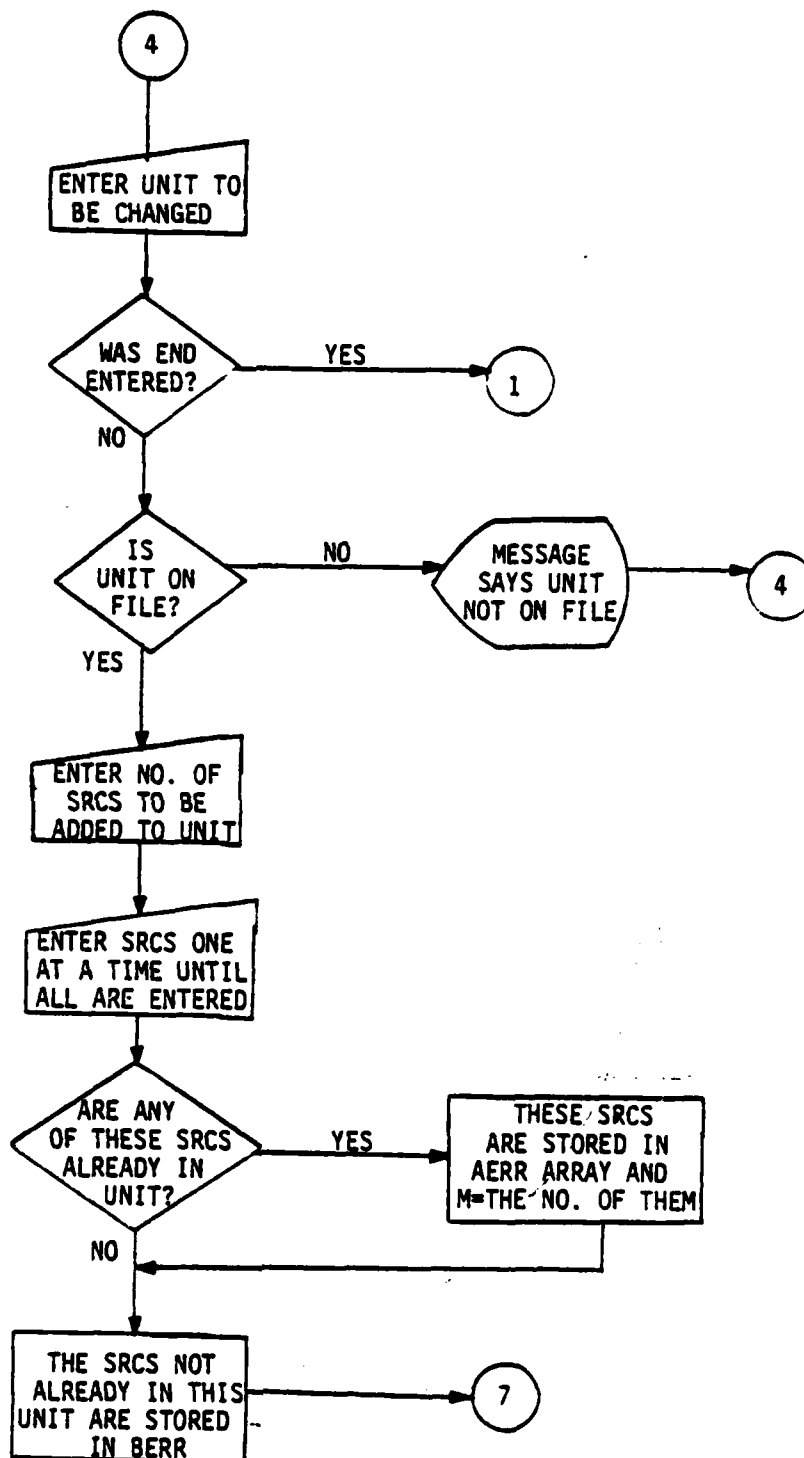


Figure C-1. UNIT program logic flow diagram (continued).

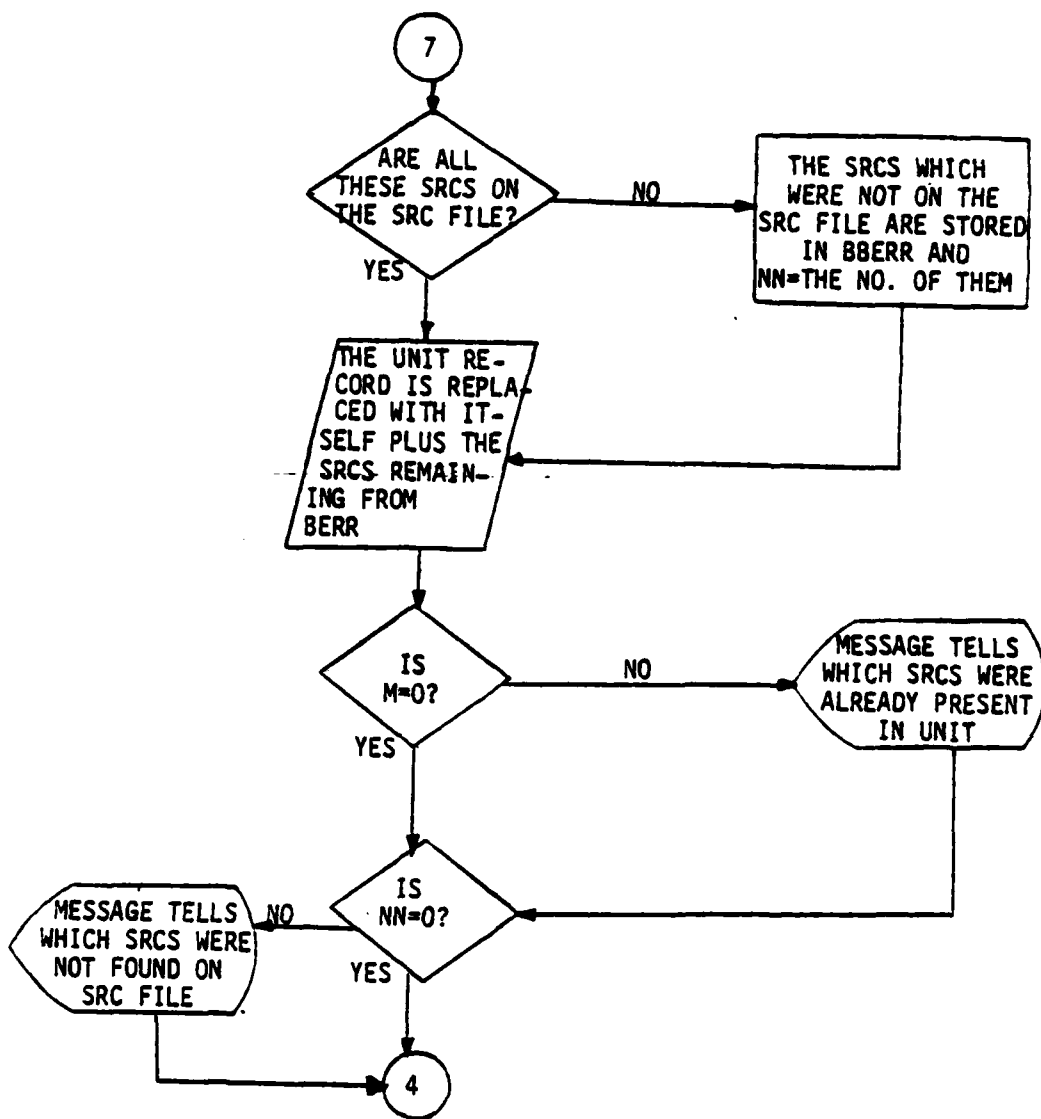


Figure C-1 UNIT program logic flow diagram
(continued).

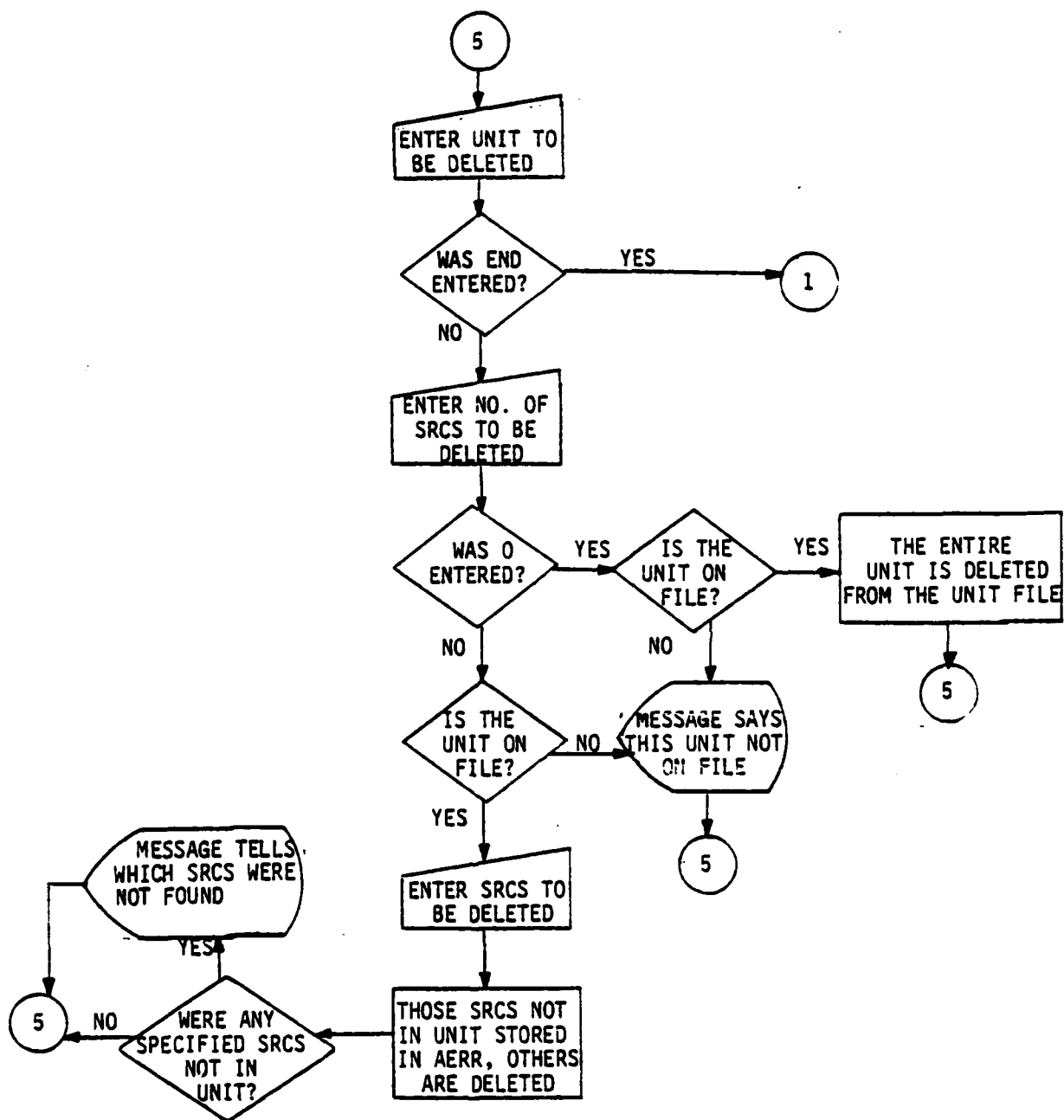


Figure C-1 UNIT program logic flow diagram (continued).

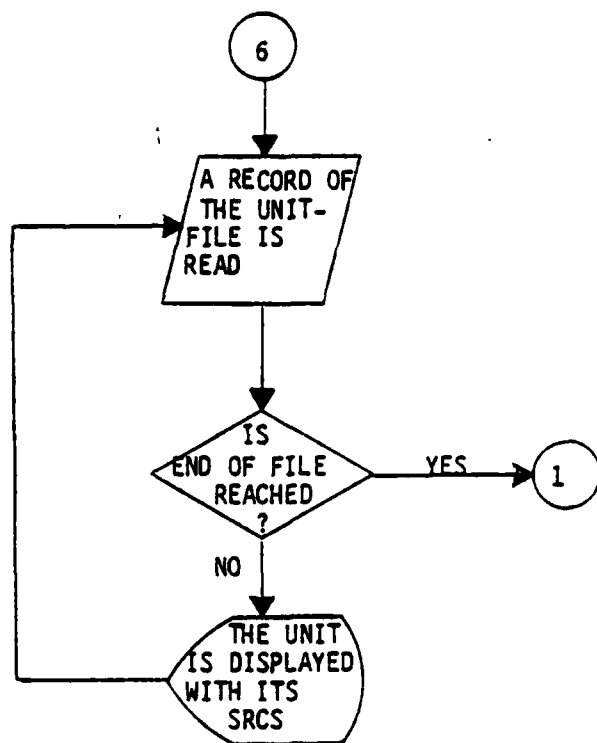


Figure C-1 UNIT program logic flow diagram (concluded).

APPENDIX D
PARENT PROGRAM

APPENDIX D

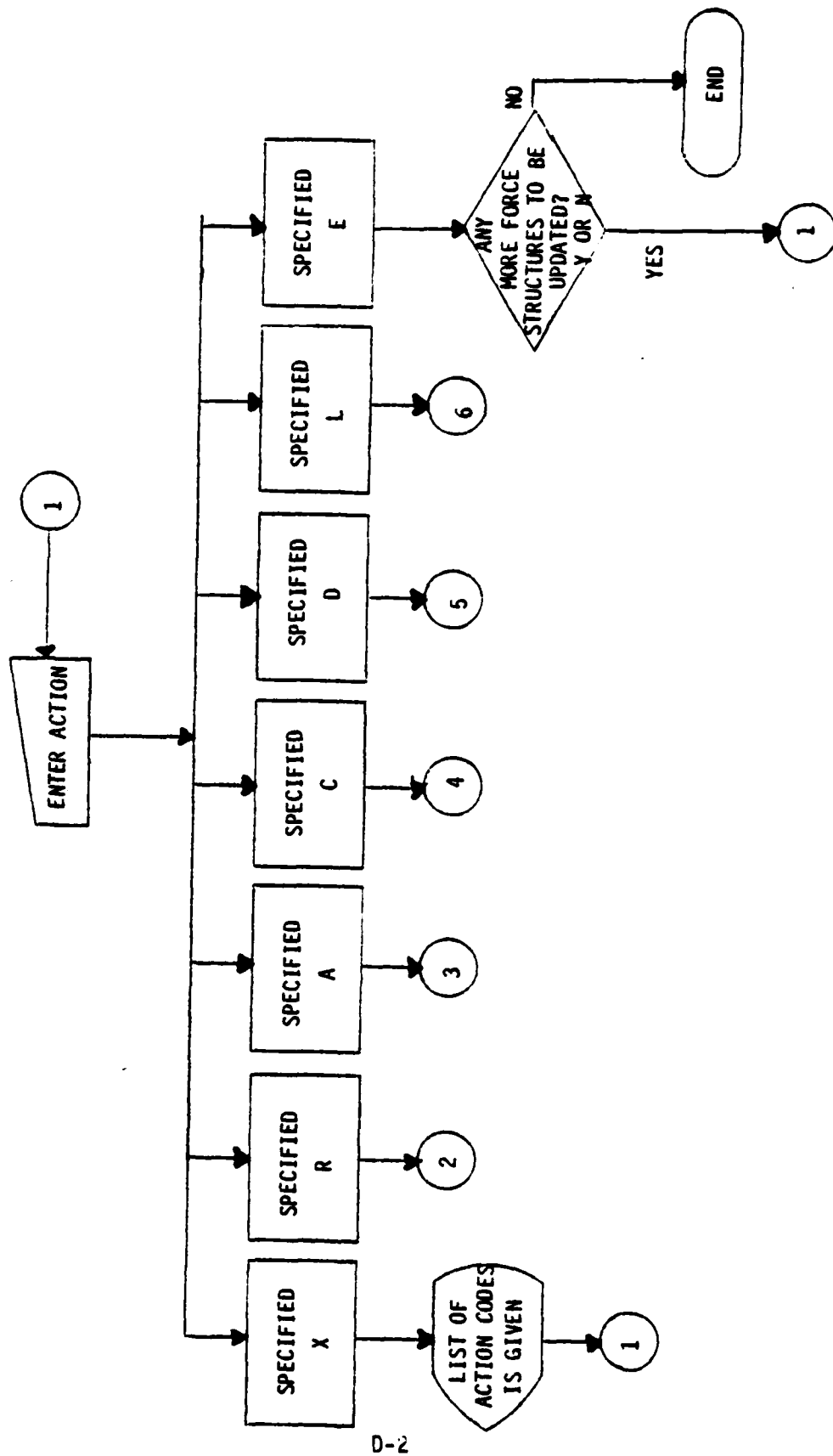
PARENT PROGRAM

D-1. SUBROUTINE DESCRIPTION.

The PARENT program is used by the military gamers, who can task organize interactively the combat units previously defined on the UNIT file into a file of higher echelon organizations, or parent units. The logic flow diagram, a list of variables, and listing of FORTRAN code are given in this appendix.

D-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
BBERR	Units that do not exist
BERR	Units that are to be added
BRRAY	Work storage area (Unit File)
BRRAY (1)	Force type (Key)
BRRAY (2)	Unit name (Key)
BRRAY (3)	First SRC on record
CCHG	Units to be added (deleted)
CERR	Units already existing
CFRC	Parent specified
CHOLD	Keeps the force type
CRRAY	Work storage area (PARENT file)
CRRAY (1)	Force type (Key)
CRRAY (2)	Parent name (Key)
CRRAY (3)	First unit on record
I	Unit position on record
ICK	Action code
K	Number of units not found
LEND	Number of units to be changed (deleted)
LFIT	FIT array (Parent File)
M	Number of units already existing
N	Number of units to be added
NFIT	FIT array (Unit File)
NN	Number of units that do not exist
NY	Answer to question



D-2

Figure D-1 PARENT program logic flow diagram. (Continued next page).

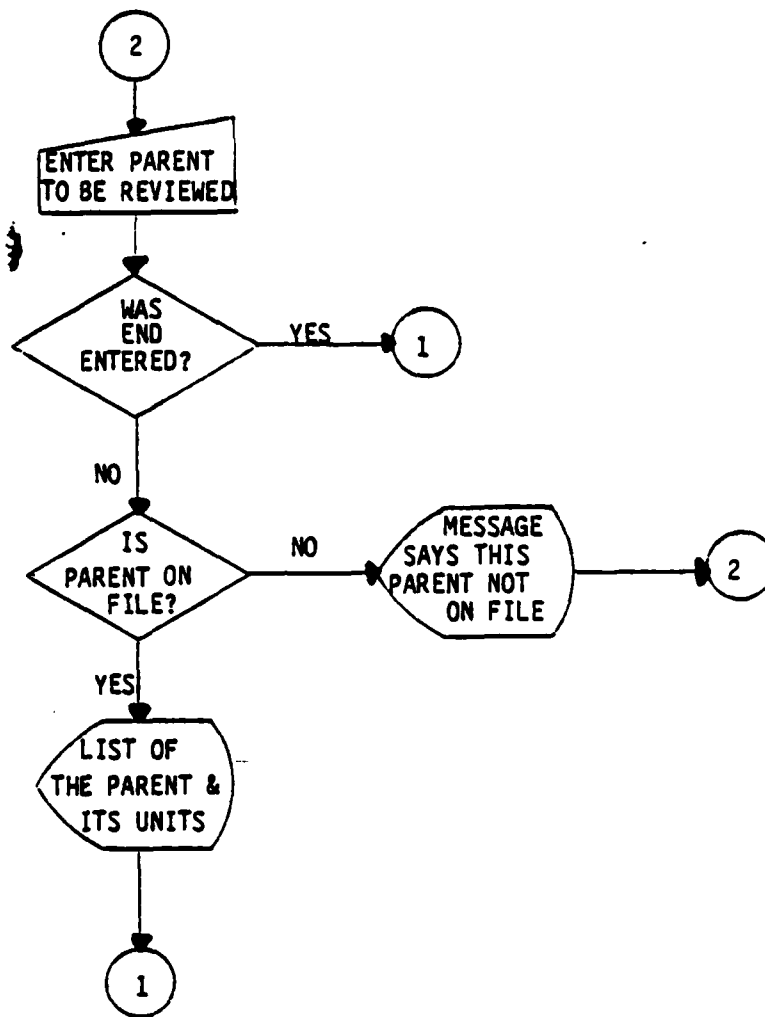


Figure D-1 PARENT program logic flow diagram (continued).

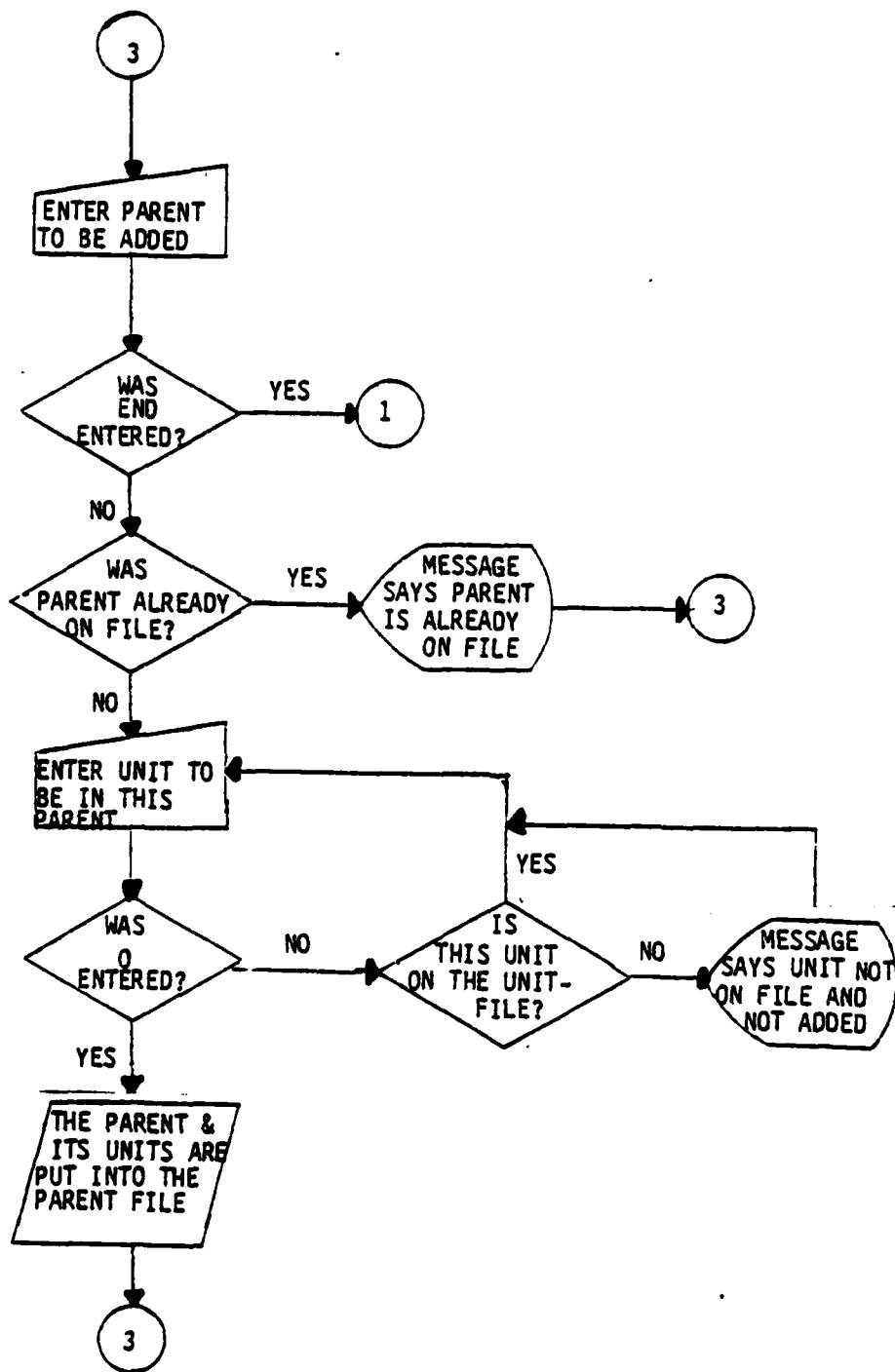


Figure D-1 PARENT program logic flow diagram (continued).

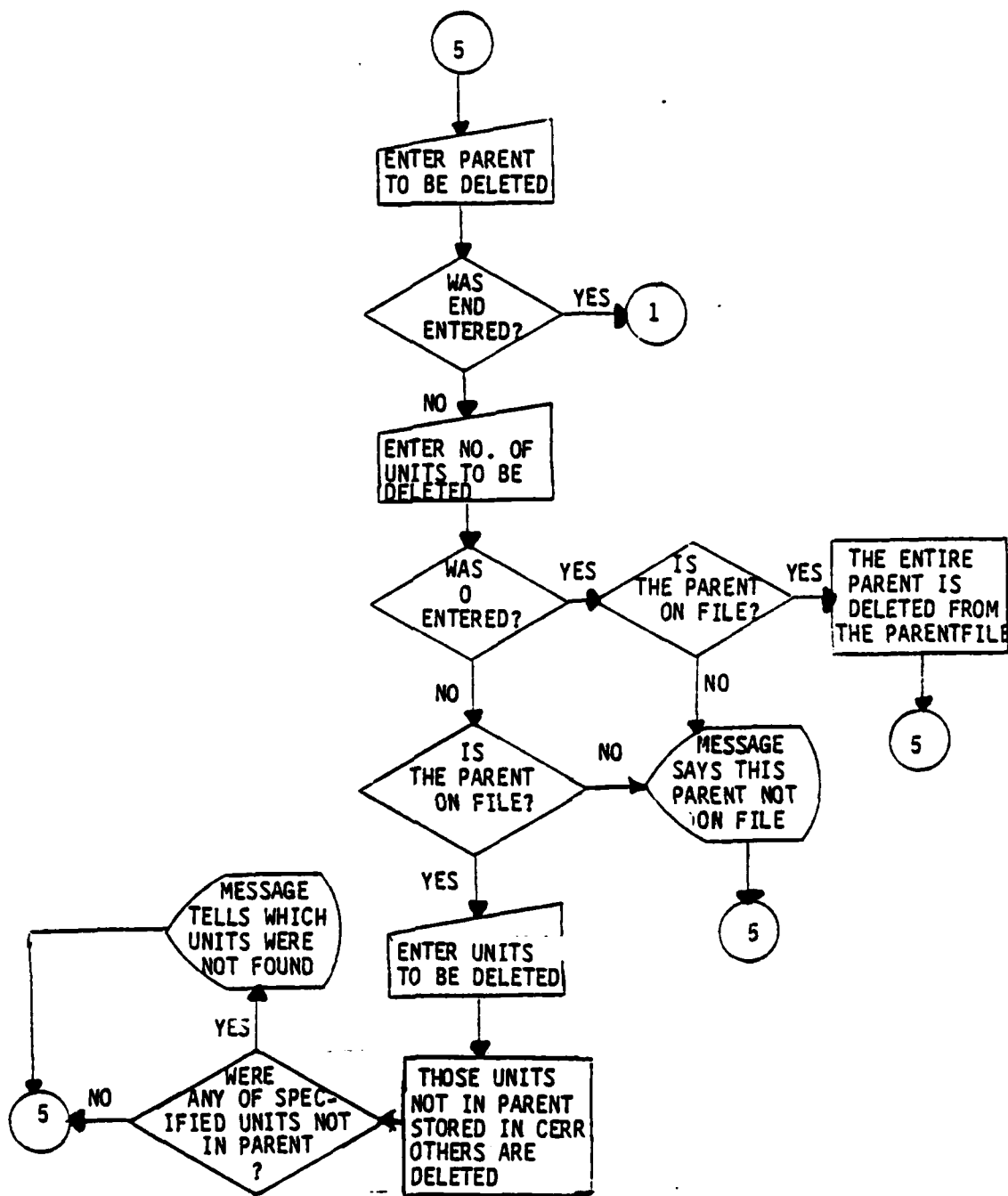


Figure D-1 PARENT program logic flow diagram (continued).

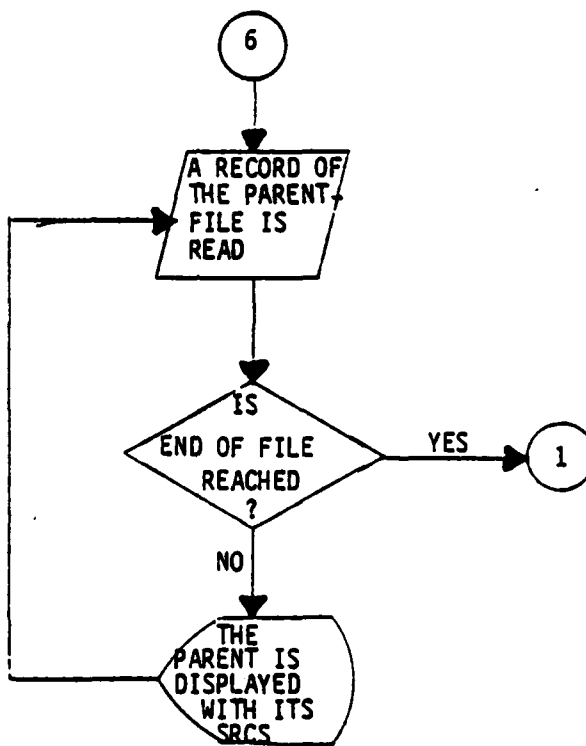


Figure D-1 PARENT program logic flow diagram (continued).

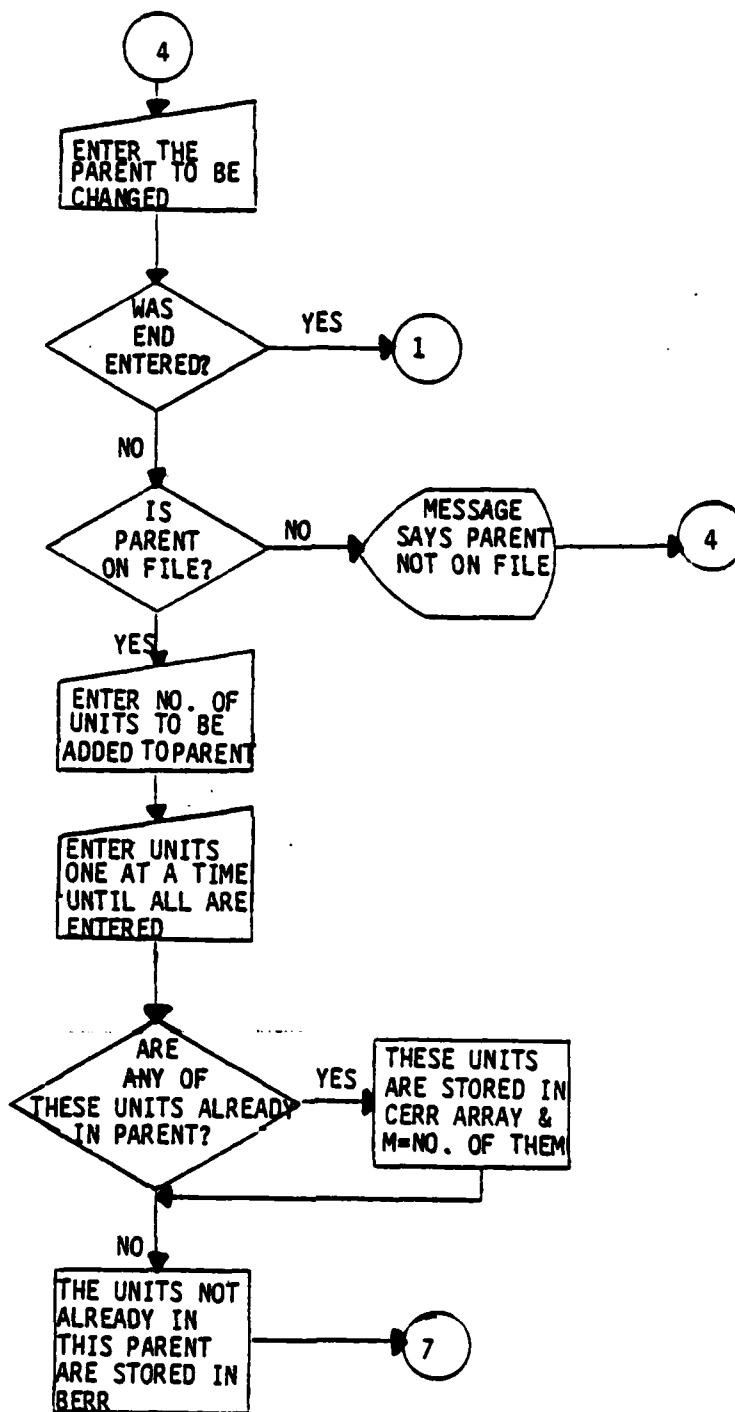


Figure D-1 PARENT program logic flow diagram (continued).

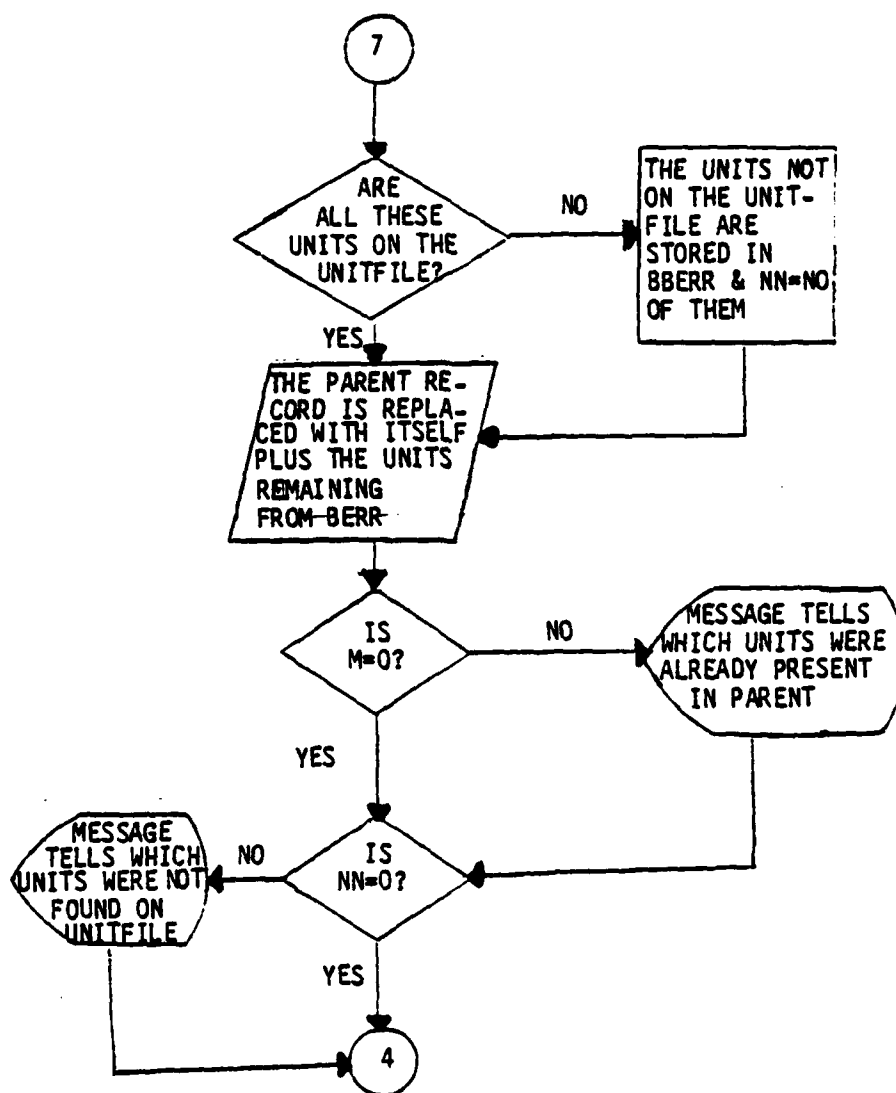


Figure D-1 PARENT program logic flow diagram (concluded).

APPENDIX E
FORCE PROGRAM

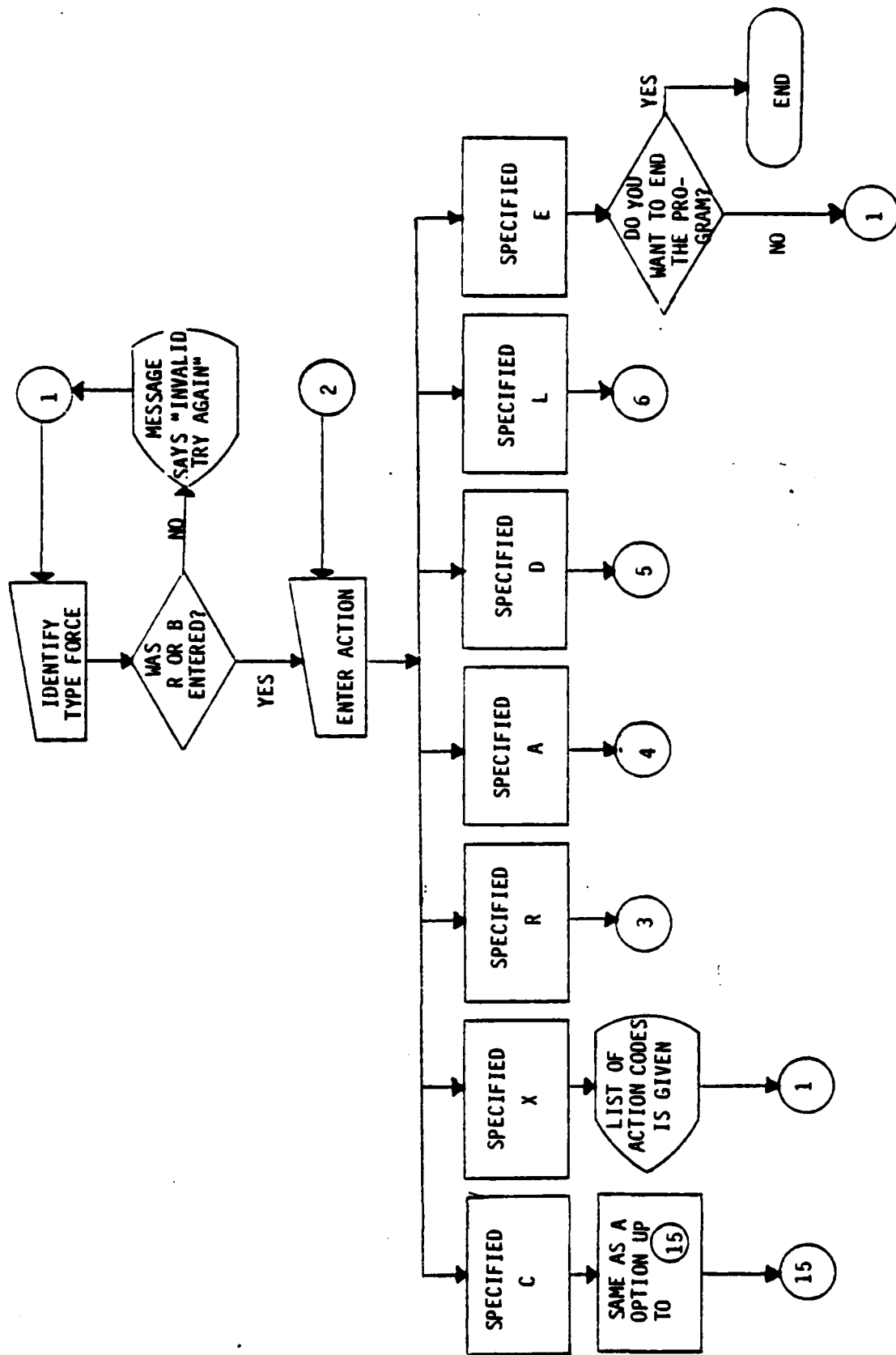
APPENDIX E

FORCE PROGRAM

E-1. SUBROUTINE DESCRIPTION. The FORCE program interactively creates a file of the forces to be assessed in the combat routines of the Jiffy game. The FORCE program consolidates the information defined on the SRC, UNIT and PARENT files. The logic flow diagram, a list of variables and listing of FORTRAN code are given in this appendix.

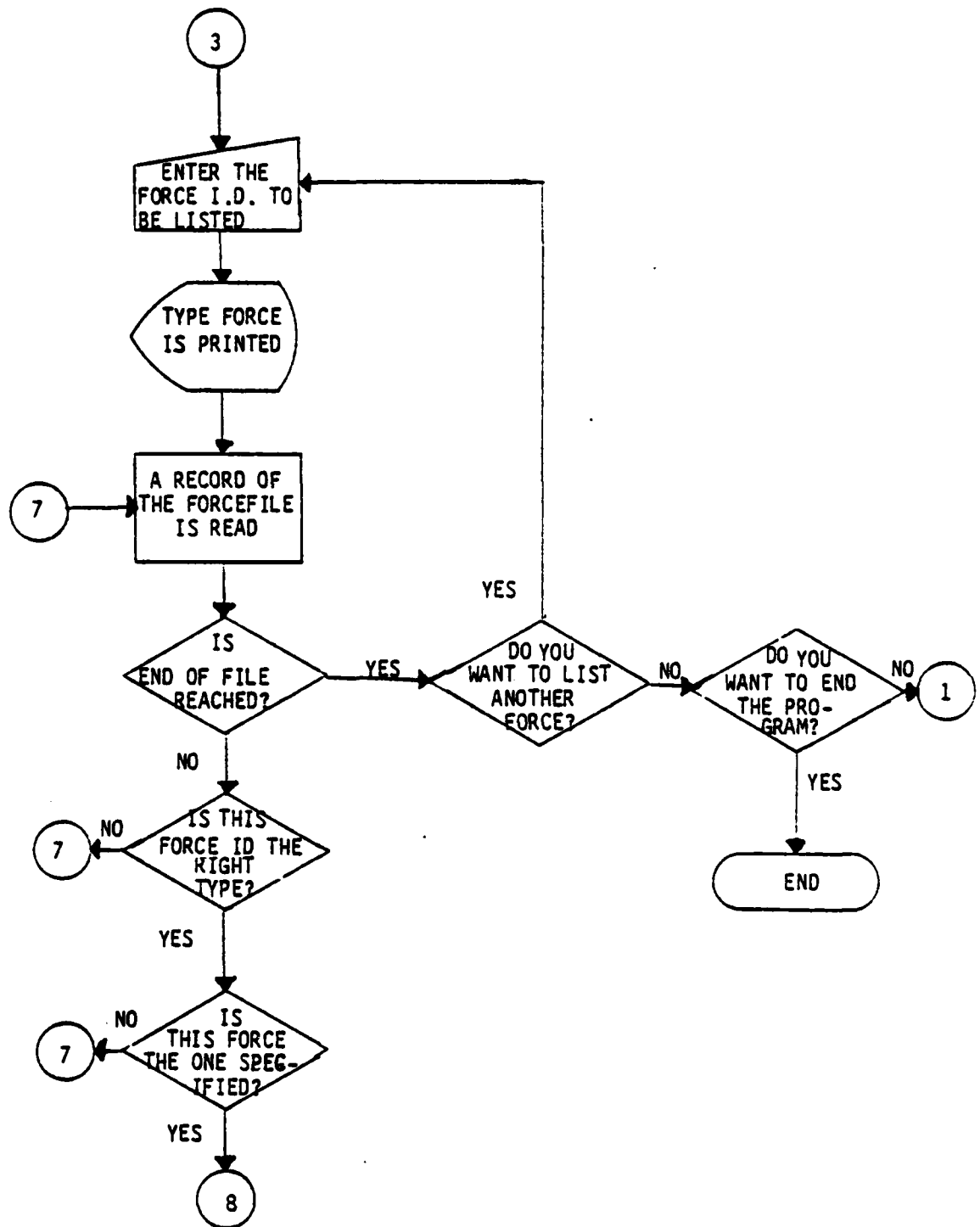
E-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AA	Keeps force type
AFOR	Work storage area (Parent File)
AH	Used to check for correct force
AHOLD	Keeps force type
ARRAY	Work storage area (Force File)
ARRAY (1)	Parent unit (key)
ARRAY (2)	Unit (key)
ARRAY (3)	Force type (number)
ARRAY (4)	Sector
ARRAY (5)	Critical Incident
ARRAY (6)	FPS @ 100%
ARRAY (7)	Permanent combat intensity level
ARRAY (8)	Percent strength
ARRAY (9)	Combat intensity level
ARRAY (10)	EW type
ARRAY (11)	Quantity of item code 1
ARRAY (75)	Quantity of item code 65
ASCENE	Force to be deleted
ASRC	Work storage area (SRC File)
ATOT	Force specified
AUID	Work storage area (Unit File)
CV	Combat value specified
FPS	Firepower score
ID	Weapon number (1-80)
IDO	Weapon listed
IFIT	FIT Array (Parent File)
JFIT	FIT Array (Unit File)
KFIT	Fit Array (SRC File)
LFIT	FIT Array (Force File)
NUMFOR	Number of forces added
TYPE	Force type specified



E-2

Figure E-1 FORCE program logic flow diagram. (Continued next page)



FigureE-1 FORCE program logic flow diagram
(Continued).

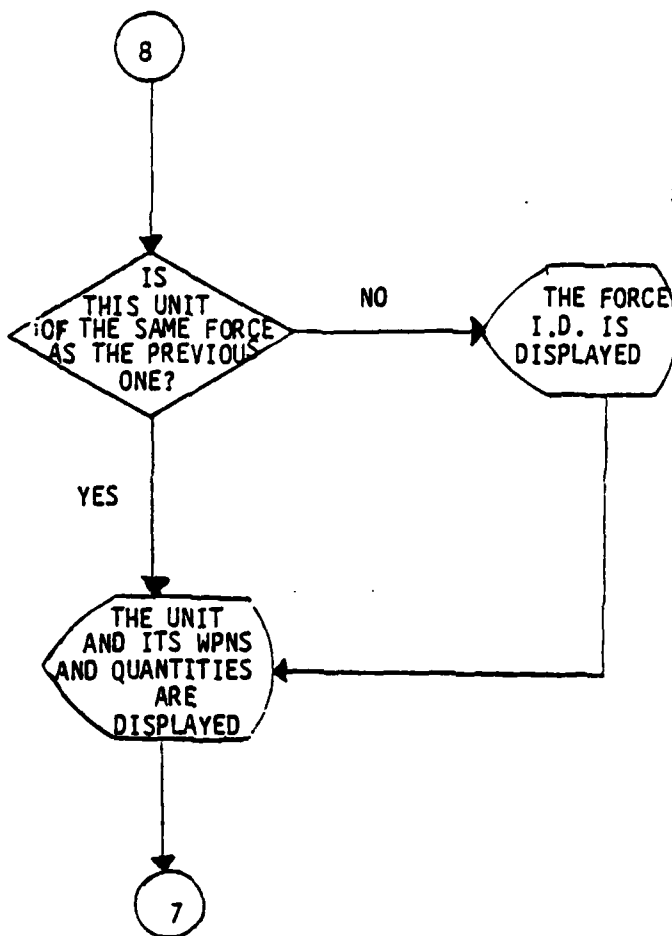


Figure E-1 FORCE program logic flow diagram (continued).

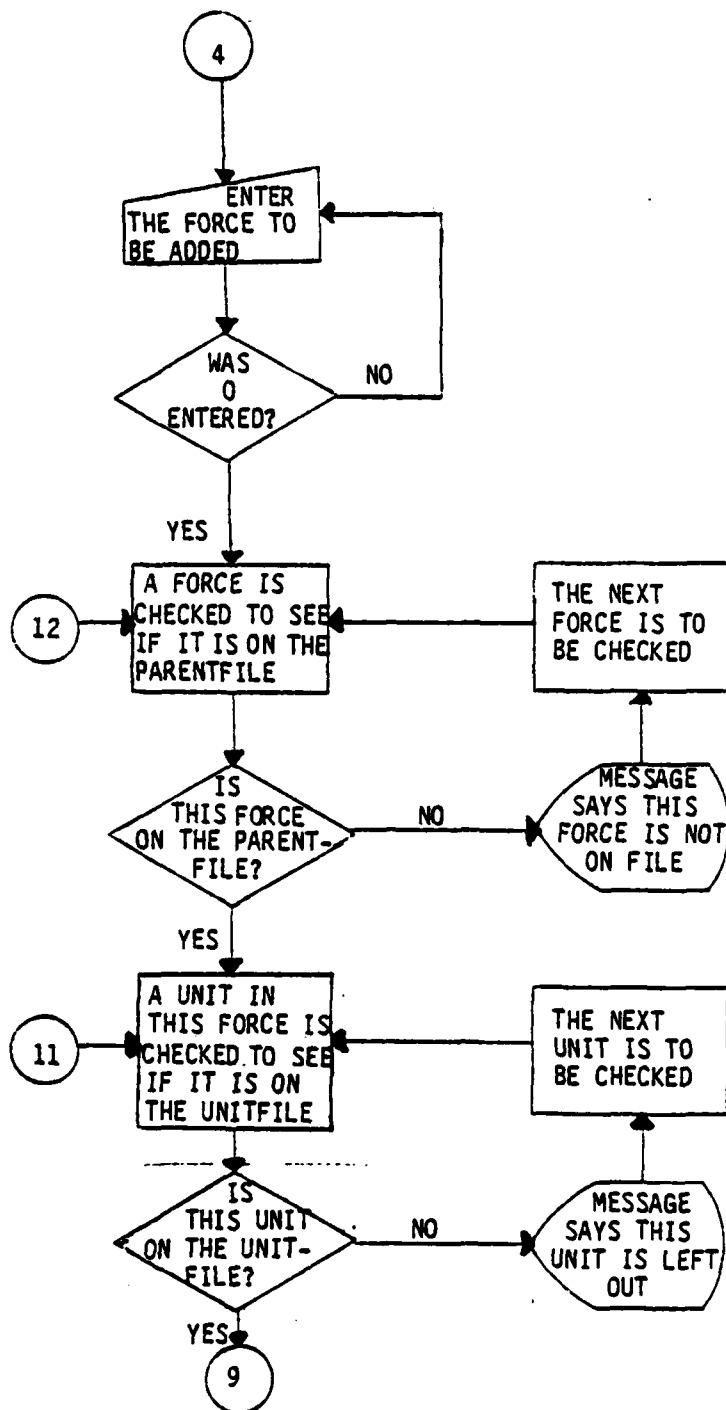


Figure E-1 FORCE program logic flow diagram (continued).

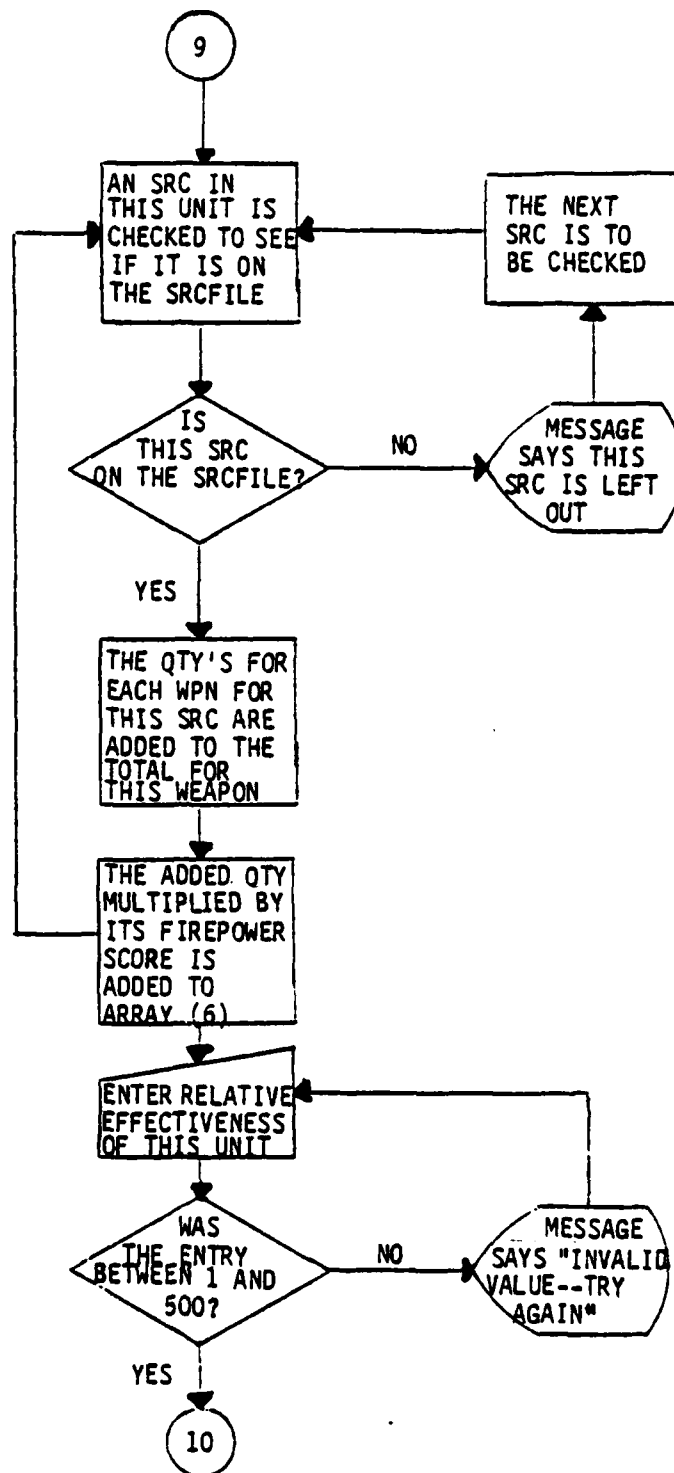


Figure E-1 FORCE program logic flow diagram (continued).

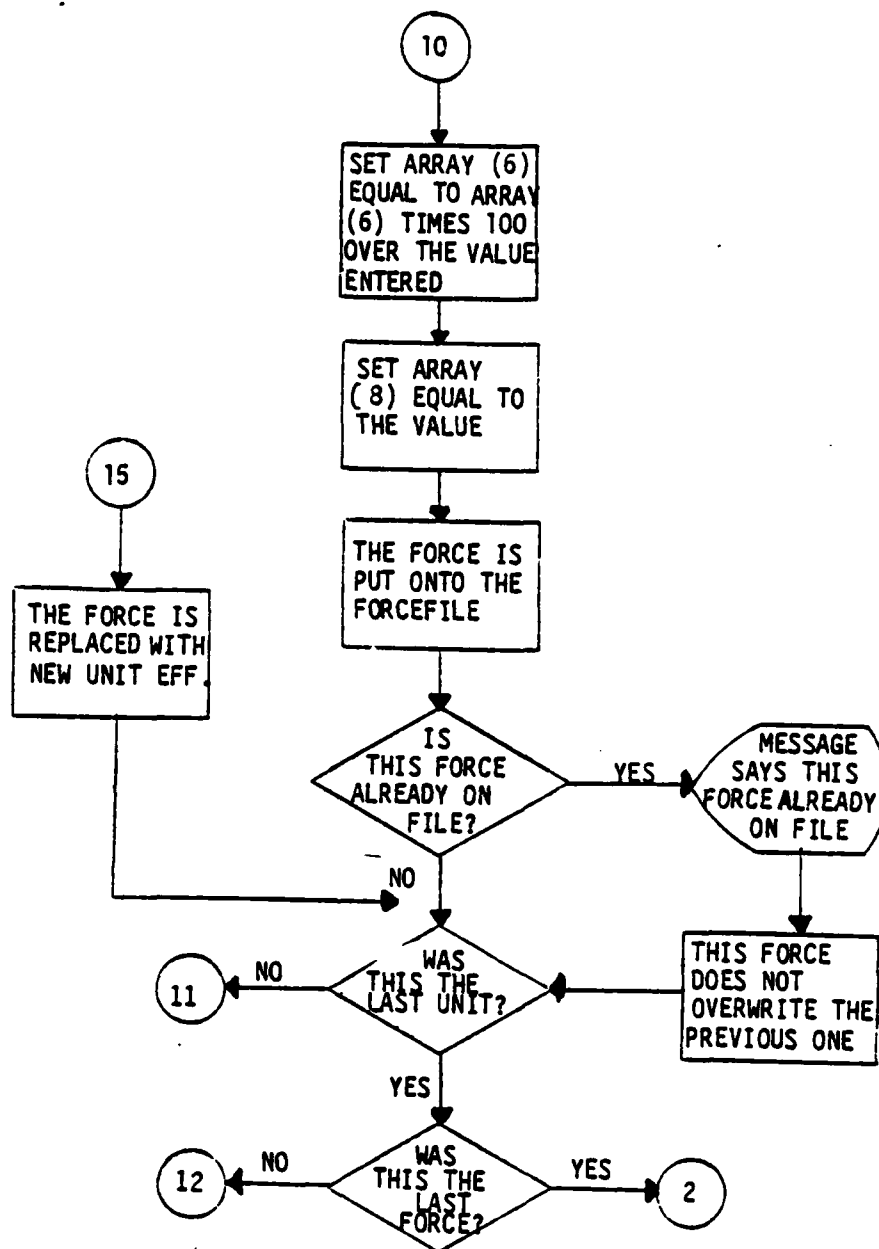


Figure E-1 FORCE program logic flow diagram
(continued).

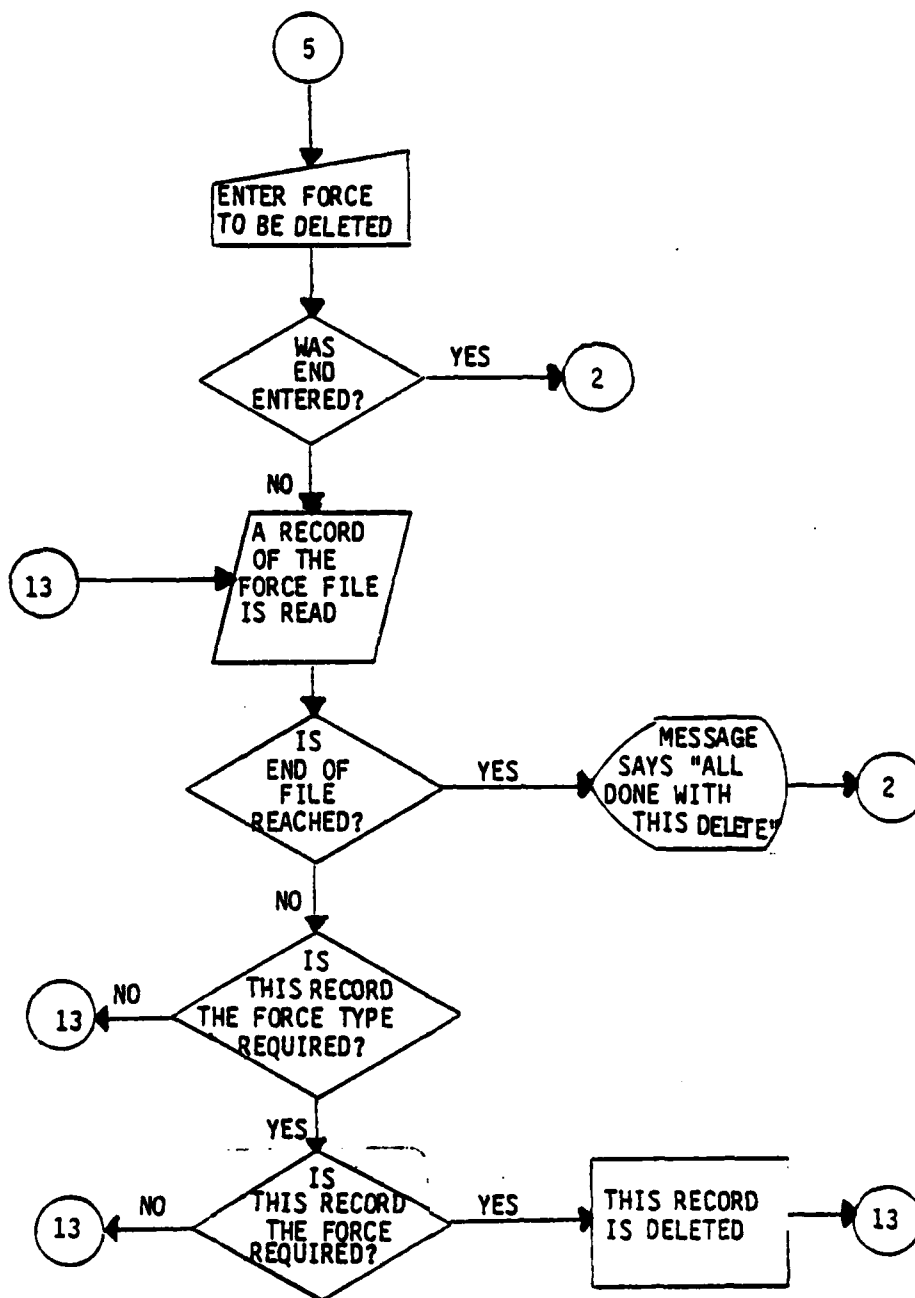


Figure E-1 FORCE program logic flow diagram (continued).

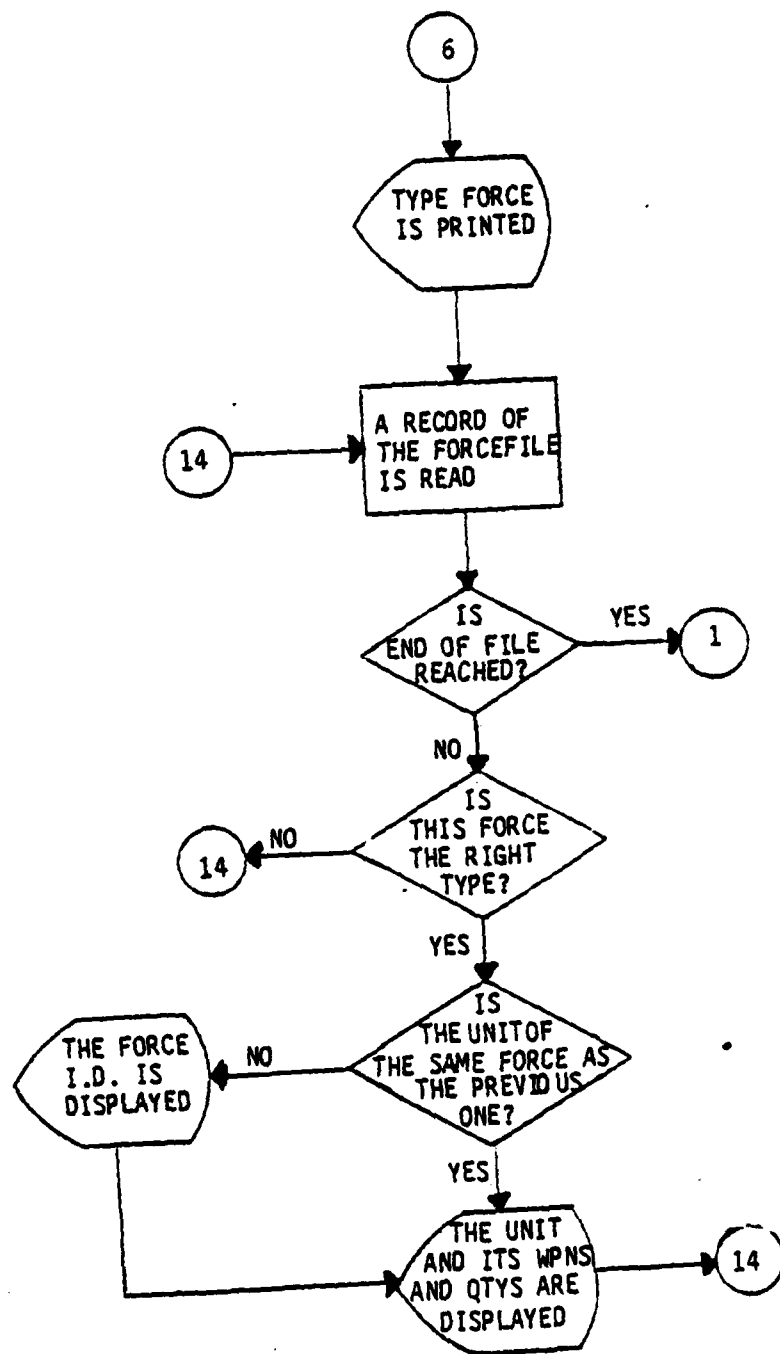


Figure E-1 FORCE program logic flow diagram (concluded).

APPENDIX F
PROGRAM SUPER

APPENDIX F

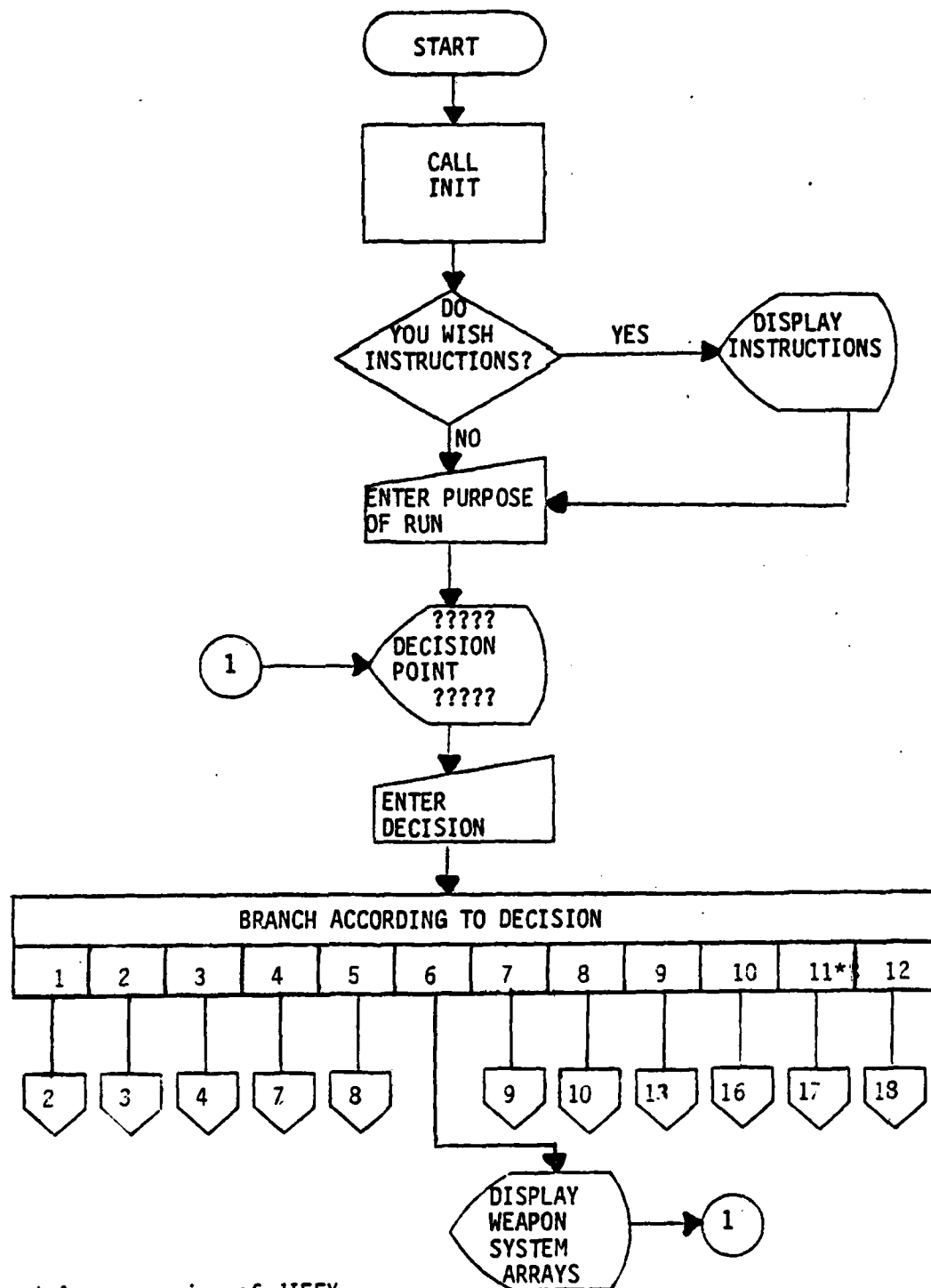
PROGRAM SUPER

F-1. SUBROUTINE DESCRIPTION. The program SUPER is the main program of the Jiffy game that serves as a control point. A functional flow diagram showing the overall Jiffy game and all the subroutine interactions, a list of variables, and the logic flow diagram are given in the appendix.

F-2. PROGRAM VARIABLES. (common variables)

<u>Variable</u>	<u>Description</u>
AA	Dummy variable for system data and time calls.
ACI	Critical incident identifier
AH	History file record array
ALOSS	Weapon loss array
APOS	Attack tactical deployment factor
ARRAY	FORCE file record array
ASCENE	Critical incident mnemonic
ASECT	Sector number
ATIME	Length of critical incident (HR)
BRRAY	SRC file record array
CFPR	Maneuver firepower ratio
CKILL	Crew kills
CLOS	Crew loss constant.
CREWS	Number of crewmen killed per weapon system
D	Number of weapons subject to loss apportionment
DPOS	Defender tactical deployment factor
ELMT	Array of weapon systems in sector
FMASS	Fraction of attacker massed
FPR	Total firepower ratio
FPS	Array of weapon system firepower scores
FSFPR	Fire support firepower ratio
FSSF	Fire support suppression factor
I	Subscript of fire weapon system.
IA	Index for attacker force
ID	Index for defender force
IENGAG	Index for tactical situation
IFIRST	Rate of advance calculation flag
IFIT	File information table for SRC file
IHIST	File information table for HISTORY file

<u>Variable</u>	<u>Description</u>
IMAXKV	Maximum size of KV table
IMOUNT	Index for attacker mobility
INX	Input response variable.
IP	Index for tactical situation table
IRUN	Index for type of run
IS	Logic flag
ITERRN	Index for type of terrain
ITHERM	Array of thermal sight flags
IVIS	Index for visibility
IWP	Index for weapon system
IYBUF	HISTORY file I/O buffer
J	Index for force color
JRUN	Batch run constant
JRUNS	New patch run constant
KEY	Data file random access key
LFIT	File information table for FORCE file
MINES	Minefield flag
MYBUF	FORCE file I/O buffer
NORD	Night or day flag
NYBUF	SRC file I/O buffer
PACK	Word packing variables
PLT	Infantryman materiel loss rates
PSN	Tactical deployment factor
SF	Suppression factor
SHOTS	Round expenditure array
SHOTSV	Temporary round expenditure array.



* A new version of JIFFY allows feature 11 to be invoked at any interactive point, not just a decision point.

SUPER flow diagram.
(continued next page)

Figure F-1

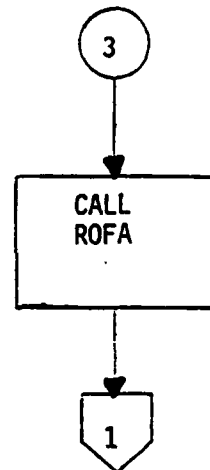
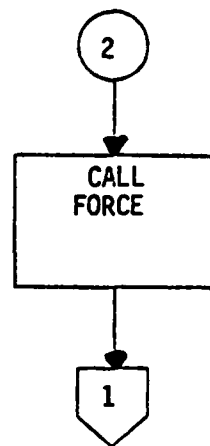


Figure F-1. SUPER flow diagram (continued).

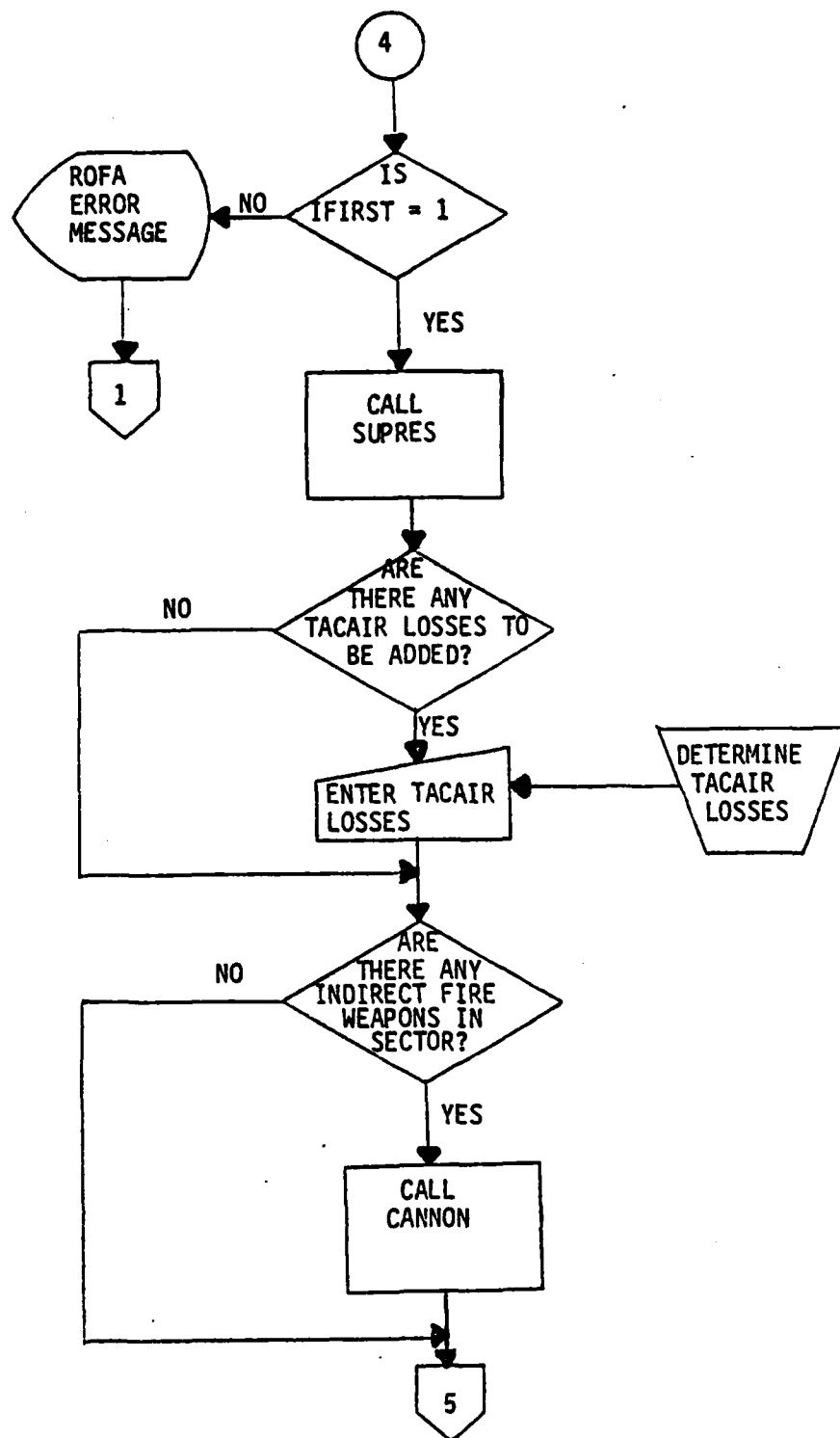


Figure F-1. SUPER flow diagram (continued).

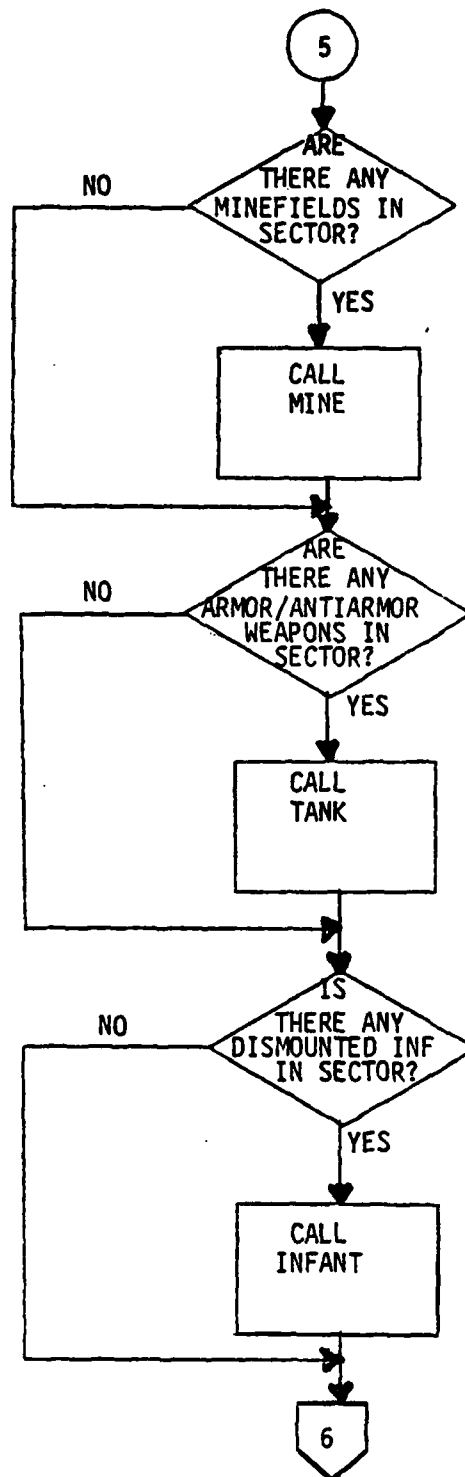


Figure F-1. SUPER flow diagram (continued).

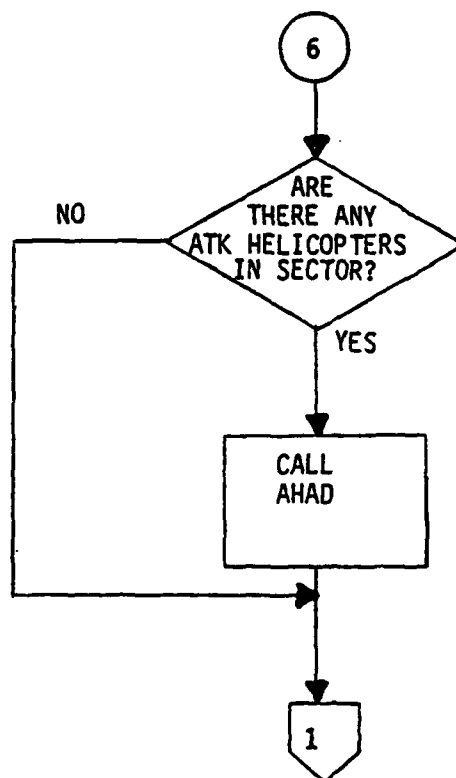


Figure F-1. SUPER flow diagram (continued).

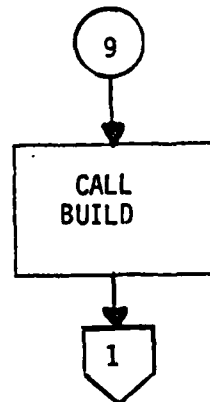
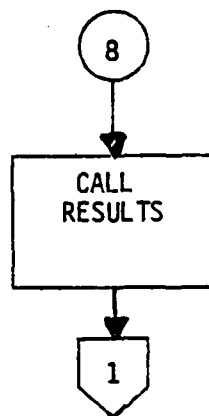
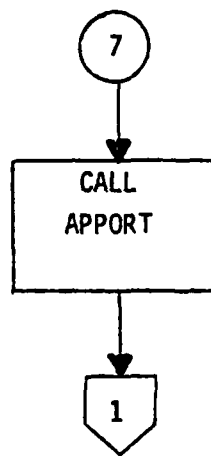


Figure F-1 SUPER flow diagram (continued).

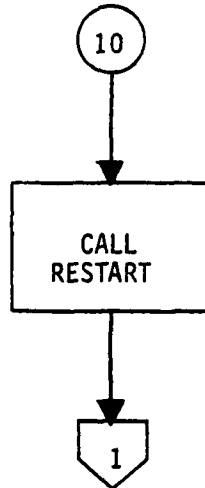


Figure F-1. SUPER flow diagram (continued)

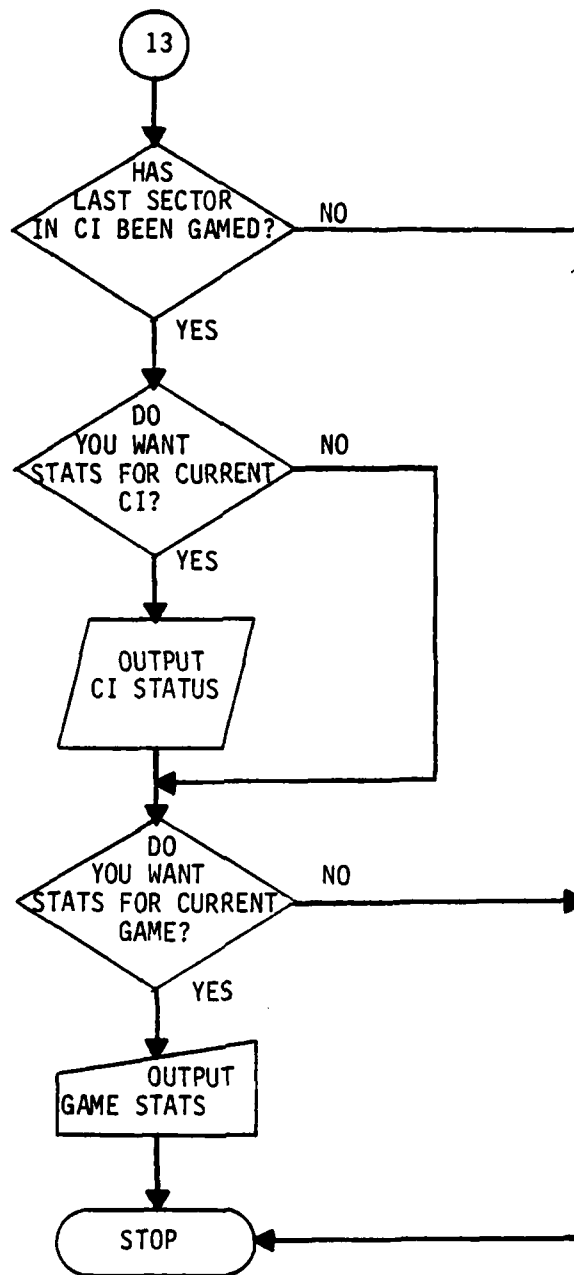


Figure F-1 SUPER flow diagram (continued)

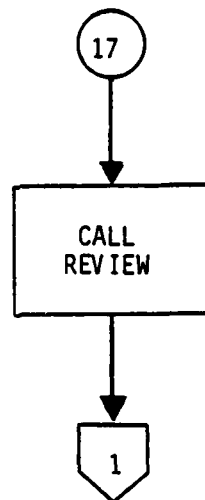
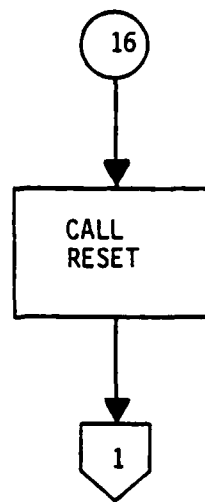


Figure F-1 SUPER flow diagram (continued)

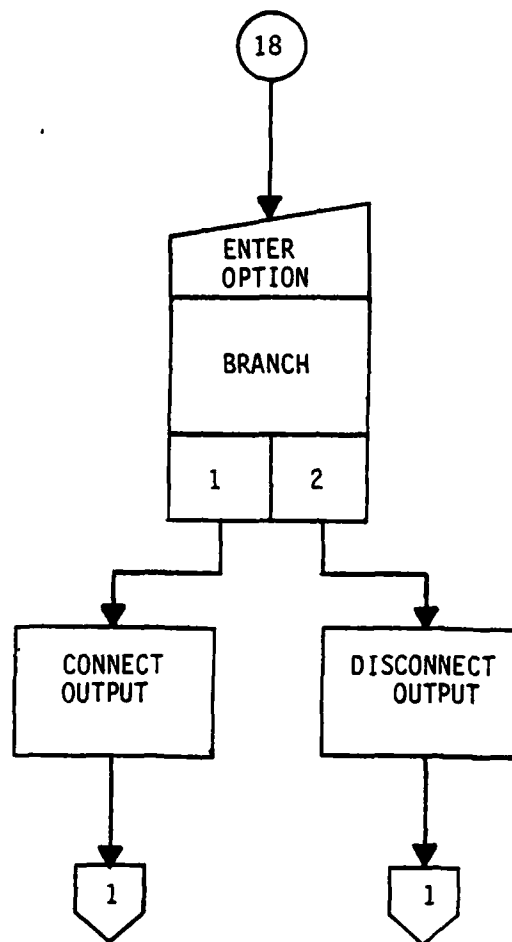


Figure F-1 SUPER flow diagram (concluded)

APPENDIX G

PRE-ASSESSMENT SUBROUTINES

Section I	EW
Section II	INIT
Section III	ROFA
Section IV	SUPRES

APPENDIX G

SECTION I. SUBROUTINE EW

G-1. SUBROUTINE DESCRIPTION. This routine calculates the factors used to degrade AD, AH, artillery and maneuver firepower score, and artillery missions.

G-2. PROGRAM VARIABLES.

<u>Variables</u>	<u>Description</u>
ACFPS	Tacair fire power score total
ADFPS	Air defense firepower score total
AHFPS	Attack helicopter firepower score total.
ARD	Dust level determiner
ICOUNT	Number of units per EW type
ISMIKE	Flag for Red smoke type
NMV	Number of maneuver units for dust effects.
PCDGR	Array for percent EW degradation
PCSMO	Percent smoked array
PCSSM	Percent self smoked arrays.

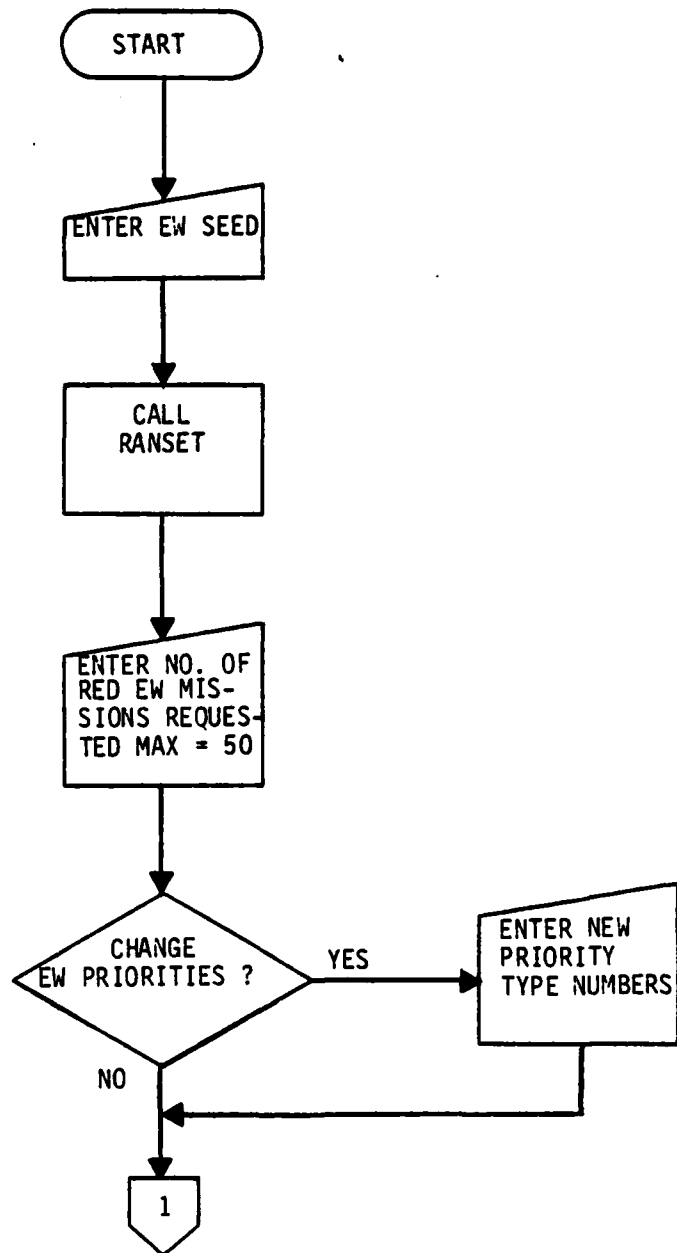


Figure G-1 EW flow diagram (continued).

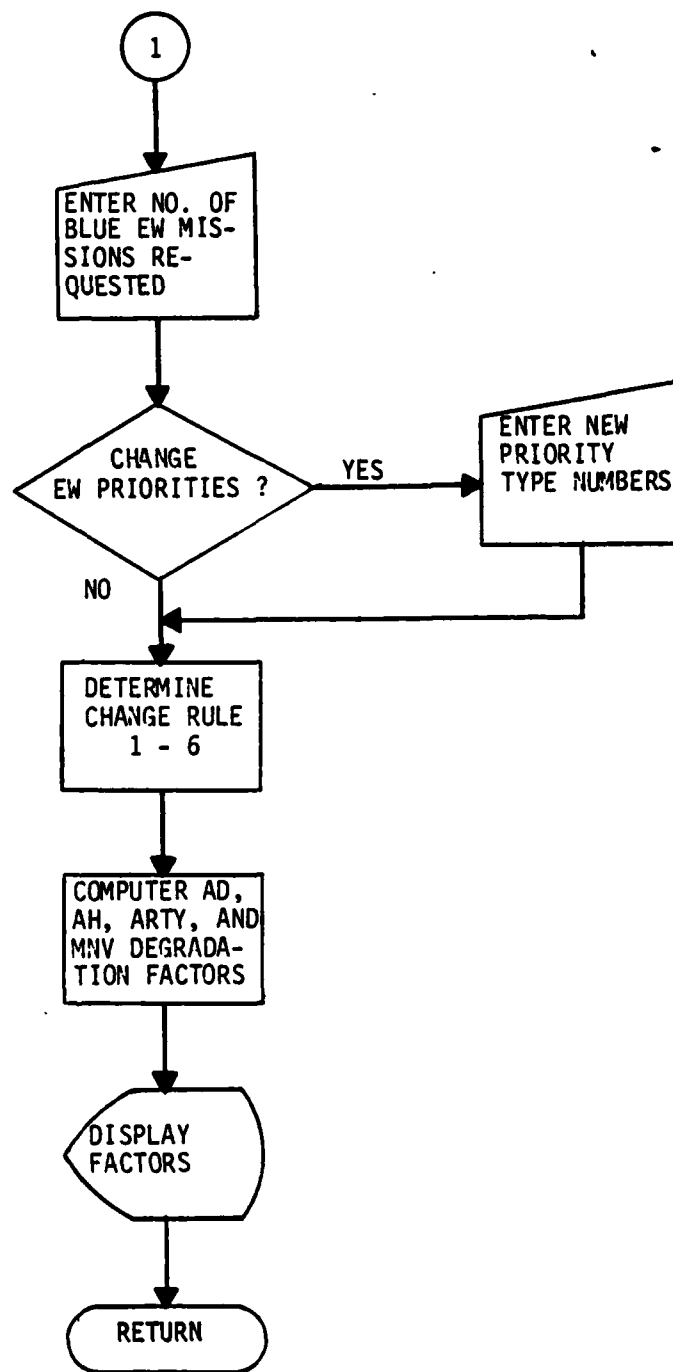


Figure G-1. EW flow diagram (concluded).

APPENDIX G

SECTION II. SUBROUTINE INIT

G-II-1. SUBROUTINE DESCRIPTION. This routine initializes the arrays in DATA/common. Note that the firepower score array (FPS) is initialized from the classified data array (CLDATA). In addition, INIT zeros the SHOTS array and initializes the word packing array variables (PACK).

G-II-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
ADIST	Attacker's covered distance
CFPR	Ground combat firepower score
CFPR	Ground combat firepower ratio
F	Fraction of sector Red force massed
FPRM	Maneuver firepower ratio
FSFPS	Fire support firepower score
FSFPR	Fire support firepower ratio
I	Weapon system index
IEL	Force index
IFPR	Rate-of-advance firepower ratio index
INX	Input response variable
ISTART	Do-loop index
ITABLE	Engagement type index
J	Force index
JVIS	Rate-of-advance visibility index
K	Weapon system index
KIND	Force color
MFPS	Maneuver firepower score
RATE	Rate-of-advance data array
RMIN	Minimum attacker firepower ratio
ROA	Rate-of-advance (KM/HR)
ROA1	Intermediate ROA calculation variable
ROA2	
STABLE	Rate of advance index determiner
TFPS	Total firepower score

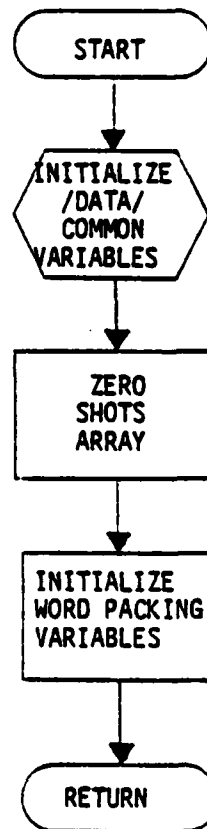


Figure G-2 INIT flow diagram.

APPENIX G

SECTION III. SUBROUTINE ROFA

G-III-1. SUBROUTINE DESCRIPTION. The ROFA routine is accessed from the main Jiffy Game program at DECISION POINT number 2 (see table 1). The primary function of this routine is to determine and display, for the sector being gamed, the rate of advance of the attacking force; the maneuver, fire support, and total firepower scores for each force; and the corresponding attacker; defender firepower ratios. To accomplish this, a number of parameters representing environmental and tactical military conditions that influence the nature of the conflict must be input interactively. Since these same factors also influence the other combat assessments, they are initialized here as variables in COMMON areas; thus, none of the combat assessment subroutine can be accessed until this routine has been executed.

2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AC	TACAIR firepower score
ADIST	Attacker's covered distance
ADIST	Attacker advance distance
AHI	History file parameter
CFPR	Maneuver firepower array
F	Fraction of sector Red force massed
FPRM	Maneuver firepower ratio
FSFPS	Fire support firepower score
FSFPR	Fire support firepower ratio
I	Weapon system index
IEL	Force index
IFPR	Rate-of-advance firepower ratio index
INX	Input response variable
INPUT	Logic flag
IS	Logic flag
ITABLE	Engagement type index
J	Force index
JVIS	Rate-of-advance visibility index
K	Weapon system index
KIND	Force index
MFPS	Maneuver firepower score
RATE	Rate-of-advance data array
RMIN	Minimum attacker firepower ratio
ROA	Rate-of-advance (KM/HR)
ROA1	Intermediate ROA calculations variable
ROA2	
STABLE	Rate of advance index determiner
TFPS	Total firepower score

NOTE: All common variables are defined in appendix F.

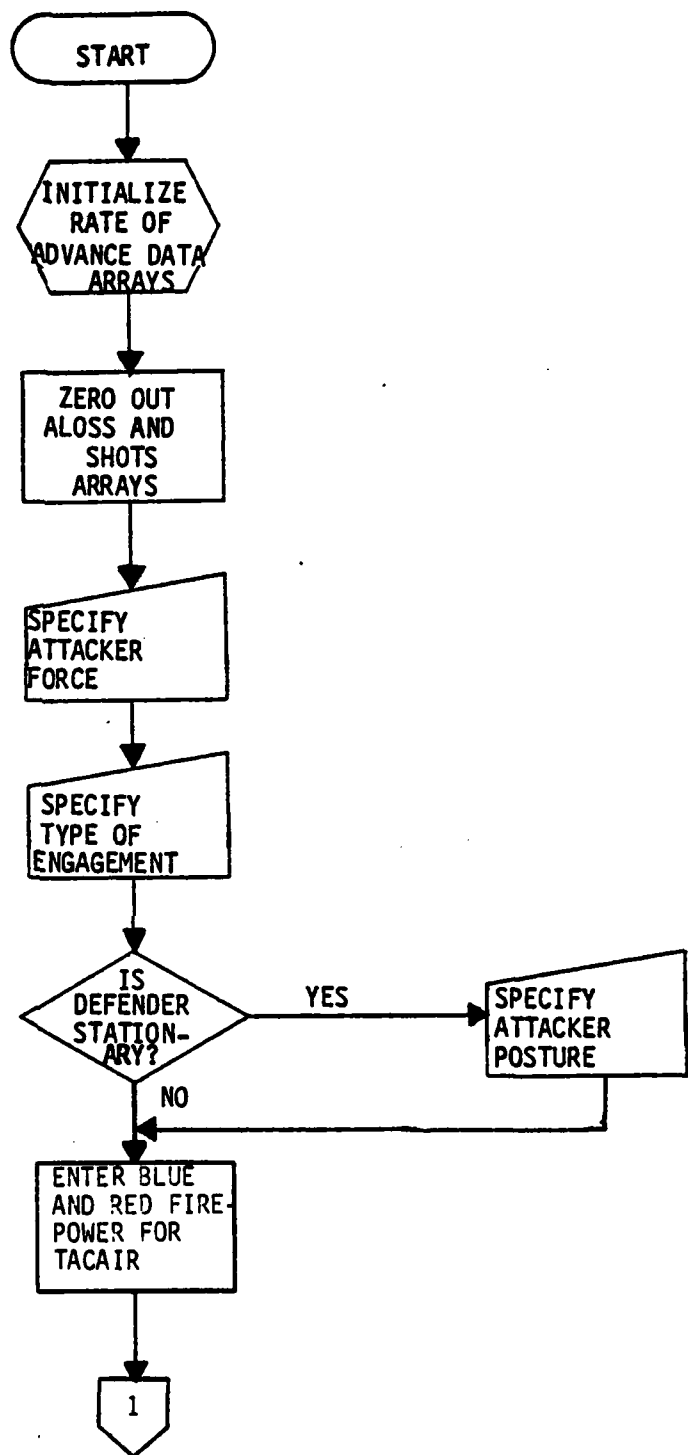


Figure G- 3 ROFA flow diagram (continued).

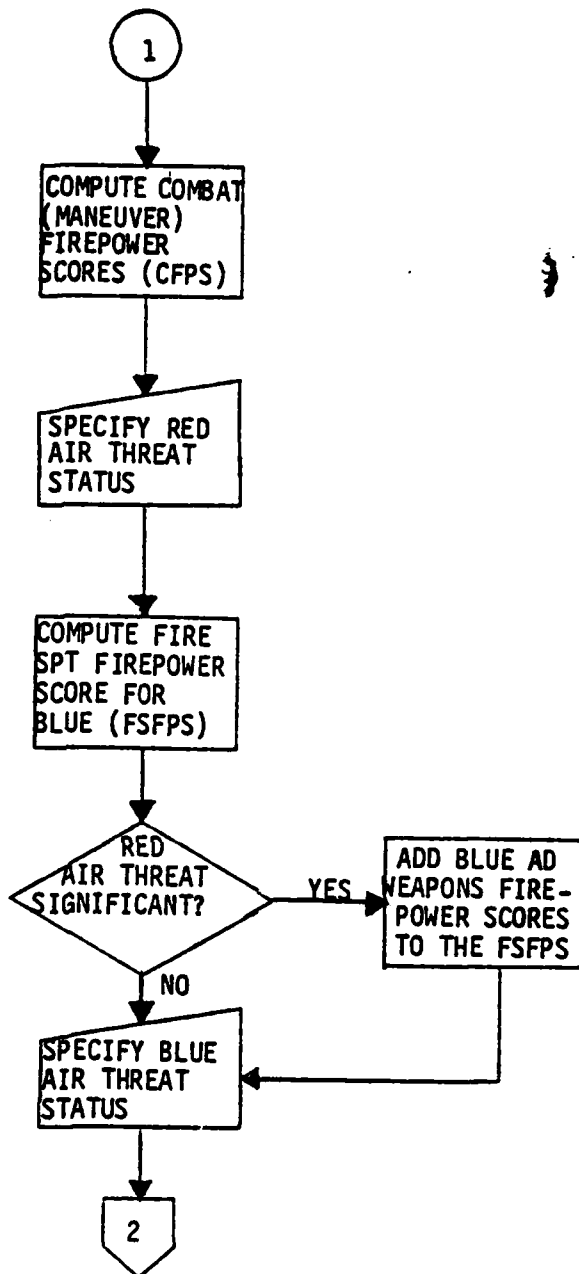


Figure G- 3 ROFA flow diagram (continued).

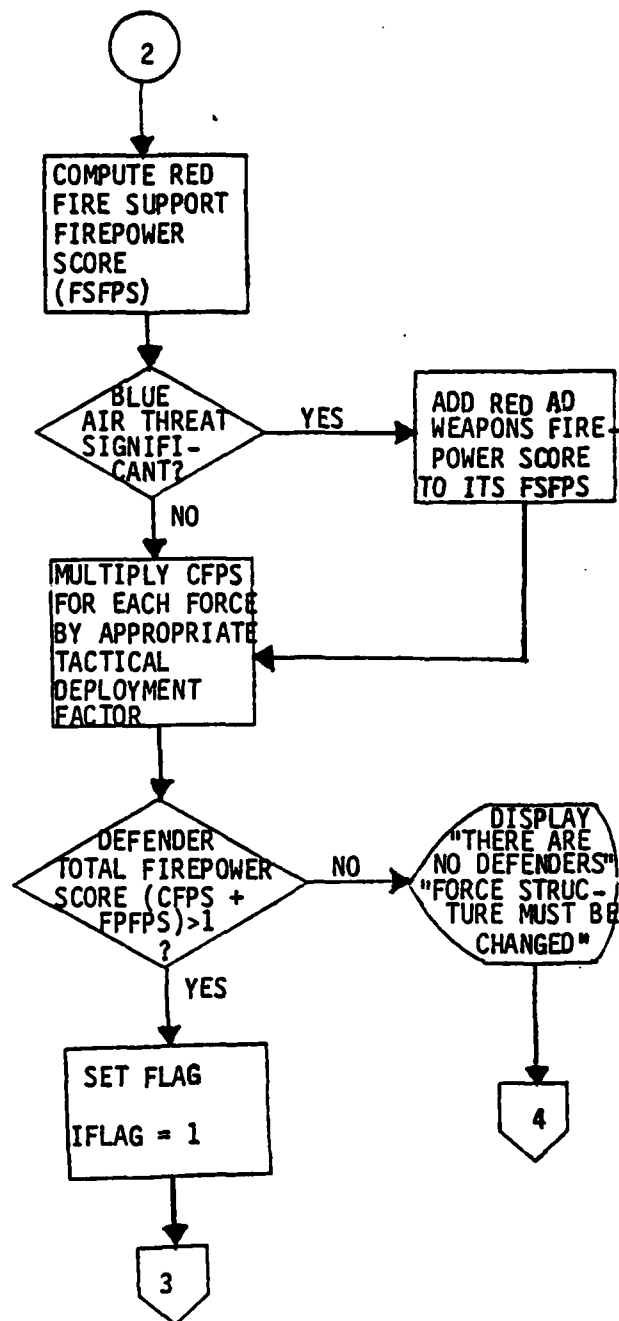


Figure G-3 ROFA flow diagram (continued).

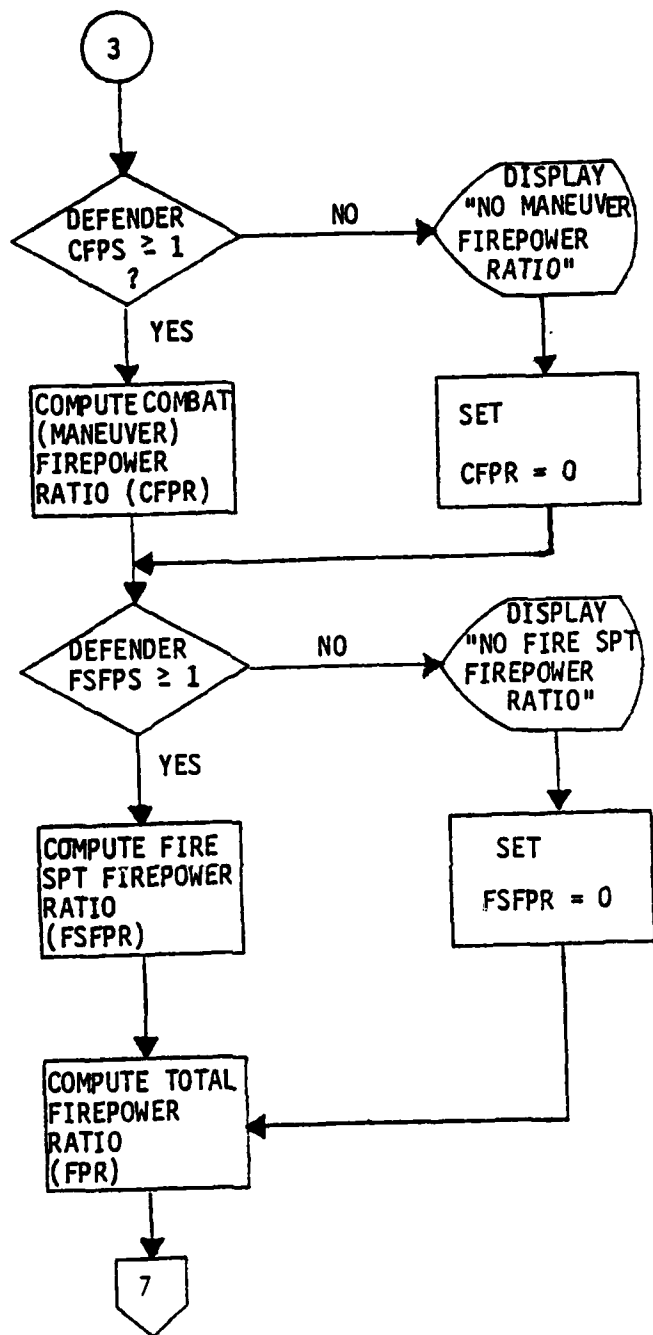


Figure G-3 ROFA flow diagram (continued).

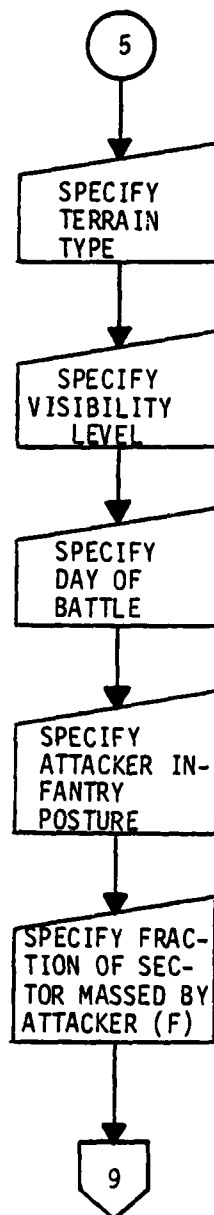


Figure G-3 ROFA flow diagram (continued).

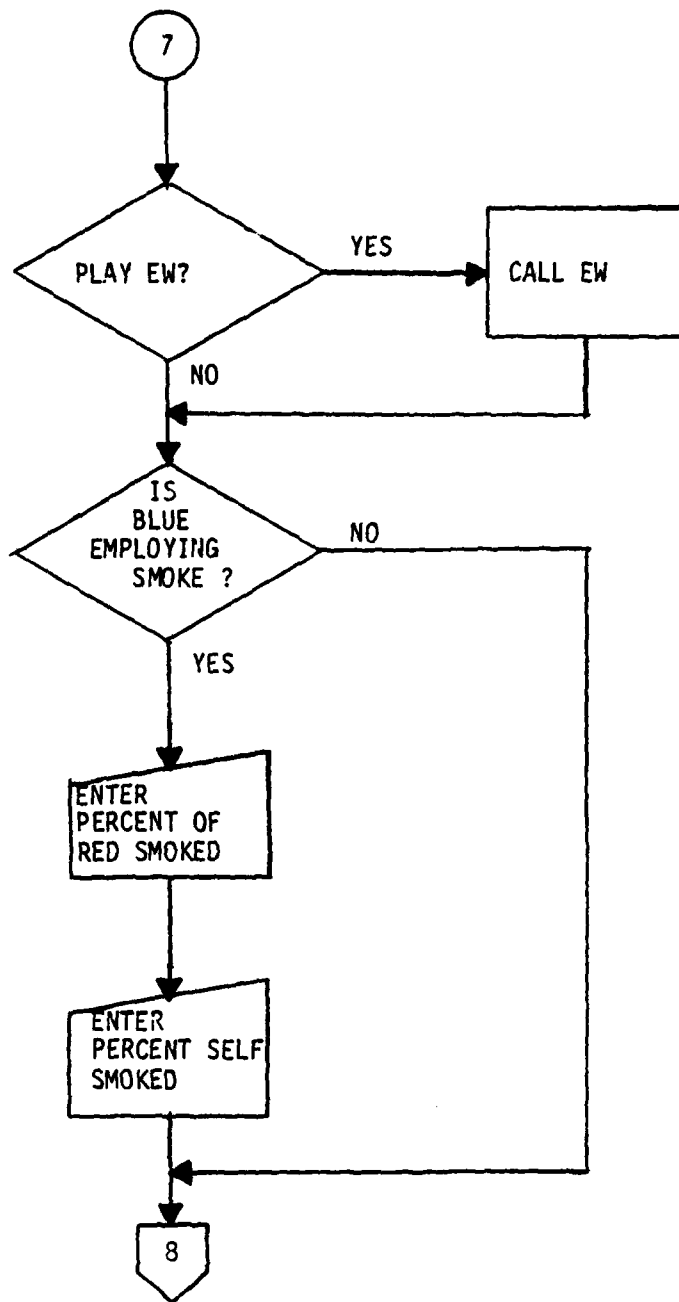


Figure G-3 ROFA flow diagram (continued).

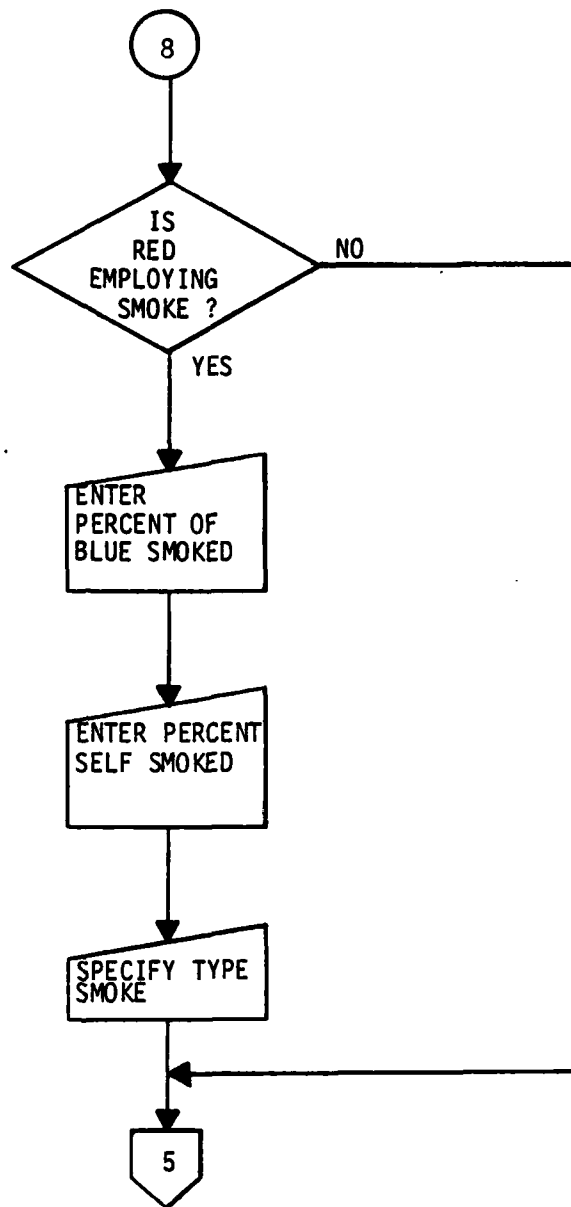


Figure G-3 ROFA flow diagram (continued).

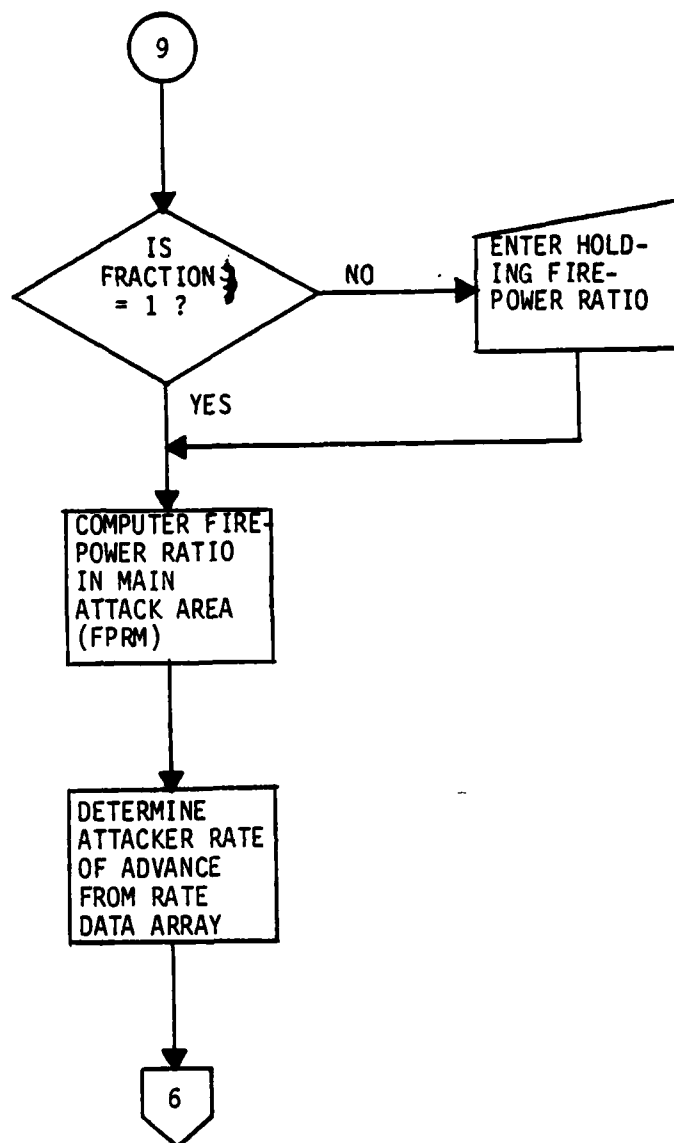


Figure G-3 ROFA flow diagram (continued).

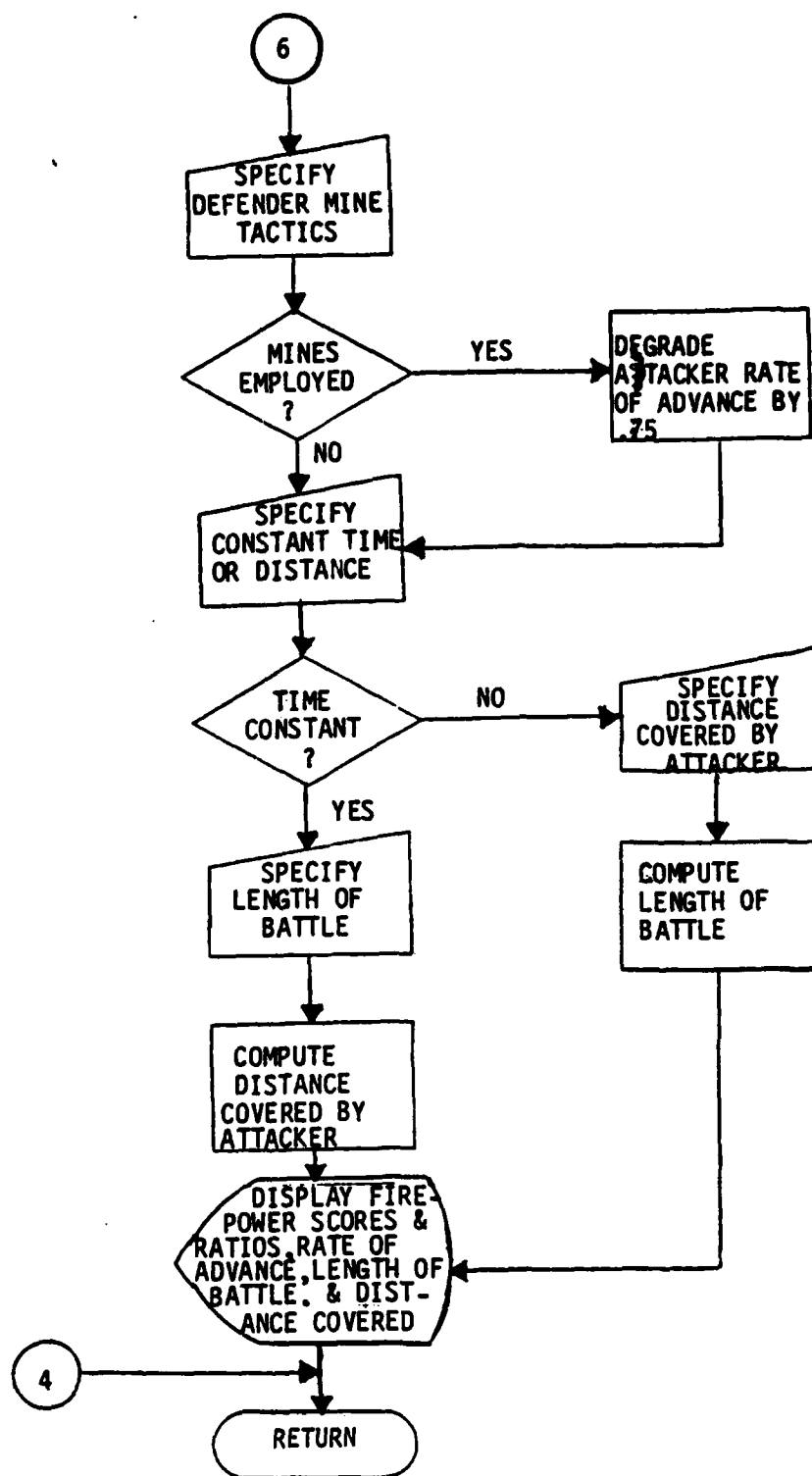


Figure G-3 ROFA flow diagram (concluded).

APPENDIX G

SECTION IV. SUBROUTINE SUPRES

G-IV-1. SUBROUTINE DESCRIPTION. The routine, which determines the suppression factors for the attacking and defending forces, is composed of an array of the suppression factors used in the Jiffy Game and a few lines of code that access the data and set the suppression factors for both forces.

G-IV-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
FACT	Suppression factor data array
I	Array index
STABLE	Limites of firepower ratios which index suppression factor data

NOTE: All common variables are defined in appendix F.



Figure G-4 SUPRES flow diagram.

APPENDIX H

ASSESSMENT SUBROUTINES

Section	I	AHAD
Section	II	CANNON
Section	III	CLGP
Section	IV	FASCAM
Section	V	INFANT
Section	VI	MINE
Section	VII	TANK

APPENDIX H

SECTION I. SUBROUTINE AHAD

H-I-1. SUBROUTINE DESCRIPTION. Program AHAD is the last of the combat assessment routines called from the program SUPER at DECISION POINT number 3 (see Table 1). The purpose of this program is to determine and display losses resulting from combat involving attack helicopters and air defense systems. Both the helicopter and AD SSKPs are stored in the classified random access file (CLDATA); several unclassified data arrays are initiated in the program itself. The program contains two sets of assessment logic, one for attack helicopter assessments against ground forces and another for air defense assessments against helicopters. The interactive definition of an attack helicopter mission initiates processing of both types of assessments, which are made for each pop-up of the helicopters in the attack cell and consequently may be cycled through several times for each mission. The number of helicopter missions to be assessed for each force is determined by the gamer; the Red helicopter/Blue air defense assessments are completed prior to beginning the Blue helicopter/Red air defense assessments. When all assessments have been completed, the cumulative losses are displayed for both forces, the LOSS subroutine is called, and the subroutine exited. This appendix contains the logic flow diagram, list of variables, and the FORTRAN source code listing for subroutine AHAD. The AHAD routine processes assessments for combat in which attack helicopters are firing at ground maneuver units while being engaged by air defense weapons. The logic flow diagram and the program variables are given in this appendix.

H-I-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
ABORT	Missile abort factor
AC	Number of helicopters entered in cell
ACAV	Helicopter operational availabilities
ACCREW	Helicopter crewmen losses
ACKILL	Mission helicopter losses
ACLOST	Popup helicopter losses
ACQ	Air Defense acquisition probabilities
ADB	Air Defense rounds per burst
ADD	Fire distribution variable
ADPK	Air Defense kill probability by range
ADUST	Dust effects factor
AHI	History file parameter
AHKILL	Cumulative probability of survival against helicopters
AIRLOSS	Helicopter losses array
AKILL	Helicopter survival probability against all AD

<u>Variable</u>	<u>Description</u>
APNSMK	Percent of helicopters unsmoked by Red
APOP	Helicopter ordnance success rates of fire
AQ	Air Defense acquisition probability
CELL	Helicopter attack cell configuration array
CLOST	Ground weapons crewmen losses
DEN	Number of scouts in cell
DENS2	Defender in high density flag
DIV	Division in number of pop-ups calculation
DUST	Dust array
ELS	Elements remaining
EXP	Total number of helicopter exposures to AD fire
FARD	Friendly barrage factor
FDF	Fire distribution factor
FE	Fraction maneuver forces engaged
FPNSMK	Percent and firers not affected by smoke
FRAC	Loss apportionment factor
GFKILL	Mission ground force losses
GIS	Mounted/dismounted infantry materiel loss factor
GNDLOS	Total ground force losses
H	Hours of flying time for helicopters
HELT	Number of helicopters remaining in a force
HVICTIM	Pass dependent helicopter victims
I	Ground weapon system index
IABORT	Mission abort flag
IAM	Status record ammo indes
IBAMO	Blue round indes
IDF	AH missile fire launch index
IECM	Electronic countermeasure index
IEL	Infantry weapon loss calculation index
IFLAG	Display header flag
IFL	Ammo expenditures calculation flag
II	Ground weapon system index
IN	AD firer index
INC	History file parameter
INX	Input response variable
IPAD	AD position indes
IPUT	Put call switch
IR	Range index
IRAMO	Red round index
IRNG	Range index
ISFL	Pass flag
IT	Terrain index
ITDASS	Smoke flag
ITGT	Ground weapon target type index
ITYP	Ordnance type index

<u>Variable</u>	<u>Description</u>
IWP	Infantry weapon index
IX	Terrain masking index
J	Ground force index
JPSN	Positioning units index for contact
JVIS	Visibility index
K	Helicopter type index
KIND	Force color
KK	Helicopter type index
KTRL	AD weapon control status factor index
MAXPOP	Maximum number of popups
MMSFE	Terrain masking index array
N	Cell popup index
NCL	History file record field index
NMAX	History file record field index
NN	Cell popup counter index
NPOP	Number of helicopter popups per sortie
NPOPUP	Number of cell popups
NTYP	Round type index
NW	History file record field index
N13	History file parameter calculation variable
N14	History file parameter calculation variable
OPAV	Weapon system operational availability
ORD	Helicopter ordnance loads
ORDEXP	Helicopter ordnance expenditure
PAP	Percent AD committed array
PCNSM	Percent of force not smoked
PDAC	Attack helicopter acquisition probabilities
PERAH	Percent AH not smoked
PF	Percent AD firers for AKM model
PHKILL	Weapon system survival probability against helicopter
PK	Helicopter probability of kill array
PKILL	Helicopter survival probability against AD weapon
PLOS	Terrain masking array
PNSMK	Percent not smoked
POPORD	Helicopter per popup ordnance expenditure
PROB	Helicopter averaged PK against target
PROB1	Helicopter PK against target in defilade
PROB2	Helicopter PK against target in open
PUN	Pass determined smoke variable
RDS	Rounds expenditure variable
RND	Rounds expenditure variable
ROUNDS	Total helicopter rounds fired
S	AD weapons suppression factor
SA	Helicopter sorties available
SFACT	Suppression factor coefficient

<u>Variable</u>	<u>Description</u>
SH	Helicopter suppression factor
SHKILI	Loss apportionment denominator for helicopter
SHTS	AD rounds expenditure variable
SKILL	Loss apportionment denominator for AD weapons
SSK	AD single engagement kill probabilities
TARD	Threat barrage factor
TMASK	Terrain masking factors
TNOW	Current total number of helicopters in cell
TSART	Initial total number of helicopters in cell
VICTIM	Total ground weapon system targets for helicopters
VKILL	Ground weapon systems killed by helicopters
WEAPC	AD weapon control status factors
WEIGHT	AD target weighting factor
XAC	Helicopters remaining in cell

NOTE: All common variables are defined in appendix F.

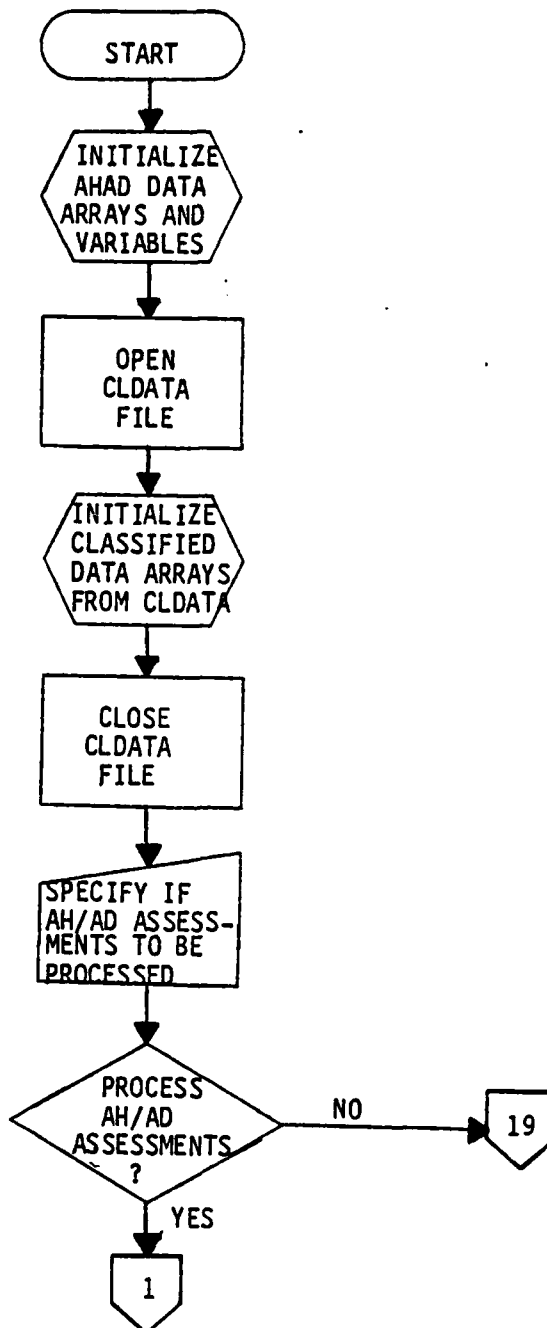


Figure H-1 AHAD flow diagram.
(Continued next page)

AD-A094 783

ARMY COMBINED ARMS STUDIES AND ANALYSIS ACTIVITY FOR--ETC F/G 15/7
CACDA JIFFY III WAR GAME. VOLUME V. PROGRAMMER'S MANUAL. (U)

OCT 80 C L PAO

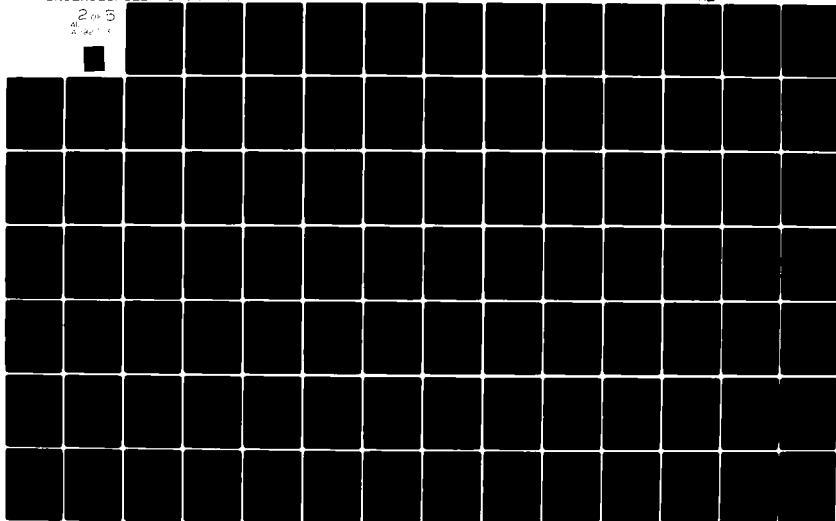
CASAA-TR-9-80

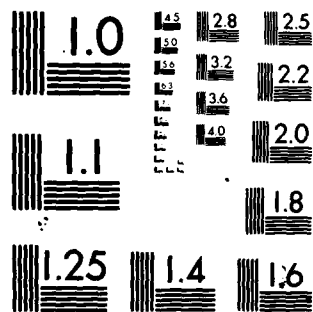
UNCLASSIFIED

NL

2 of 5

AL 100 1 1





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

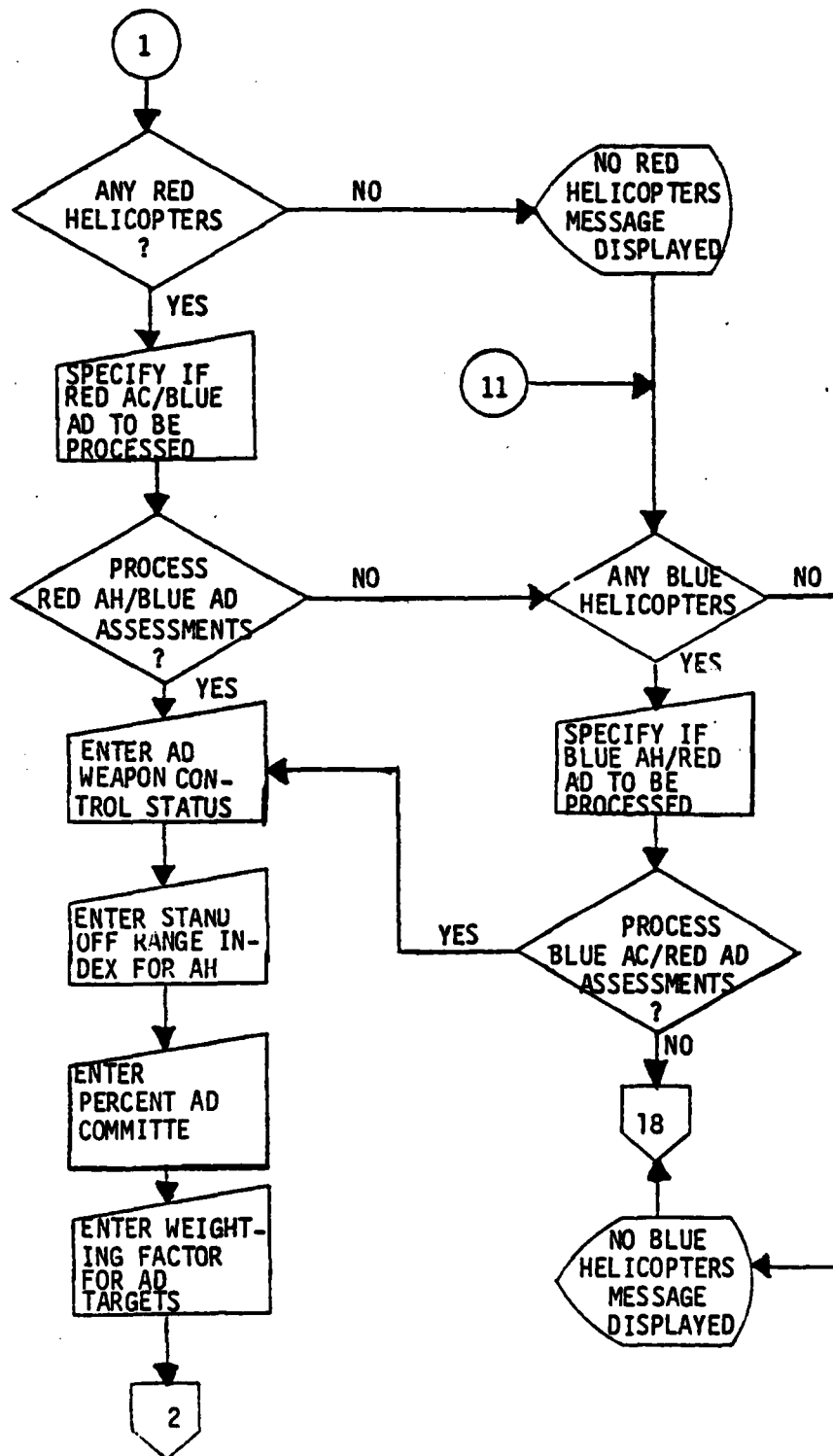


Figure H-1 AHAD flow diagram (continued).

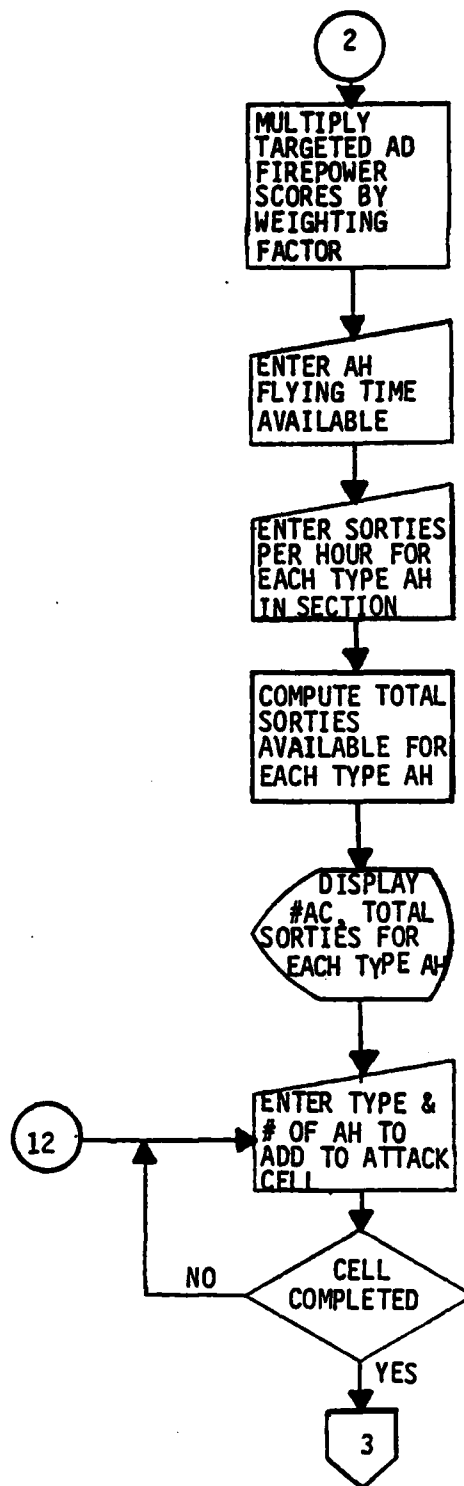


Figure H-1 AHAD flow diagram (continued).

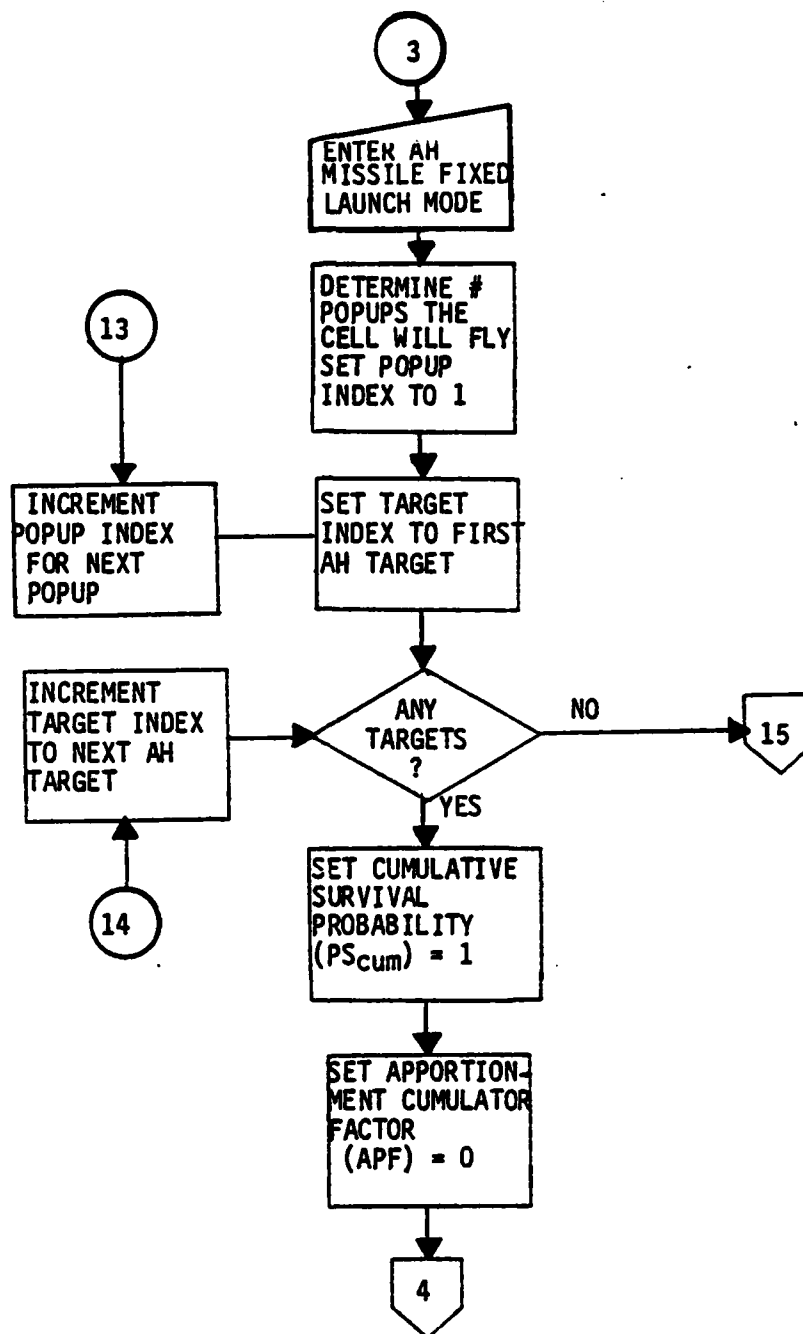


Figure H-1 AHAD flow diagram (continued).

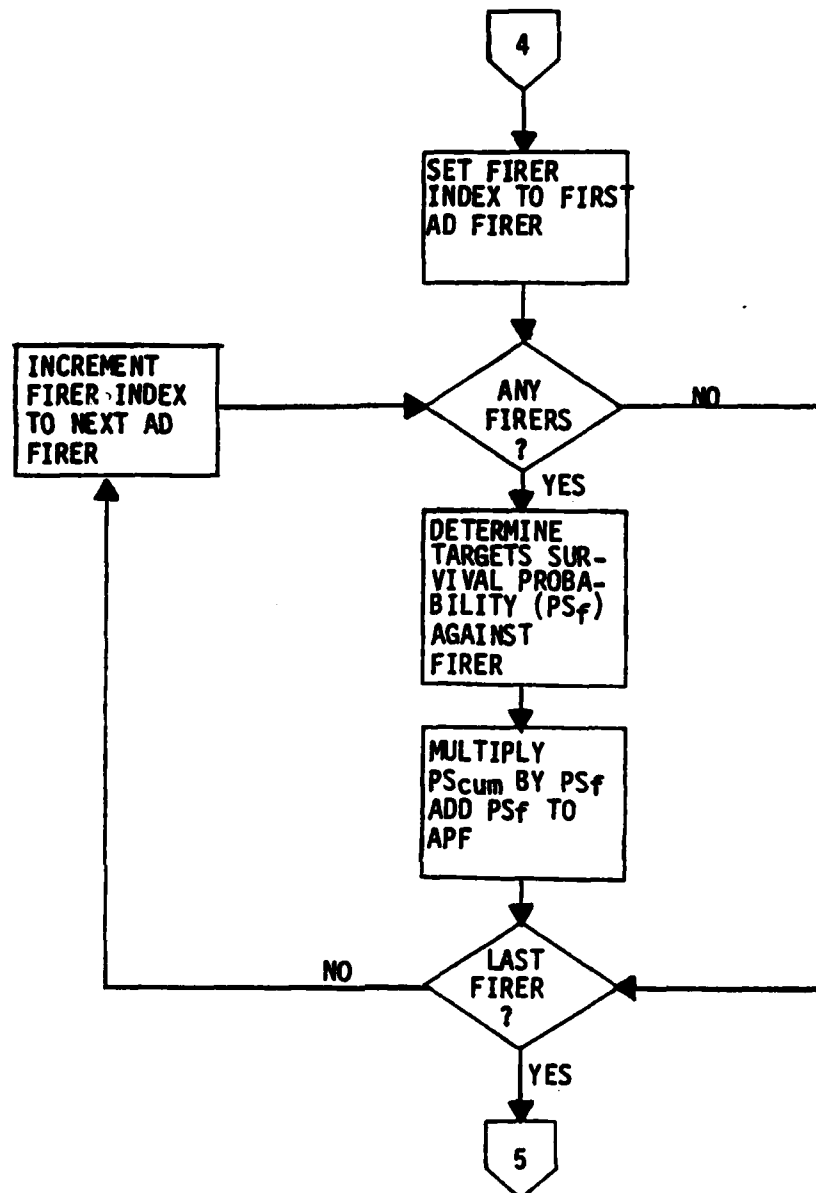


Figure H-1 AHAD flow diagram (continued).

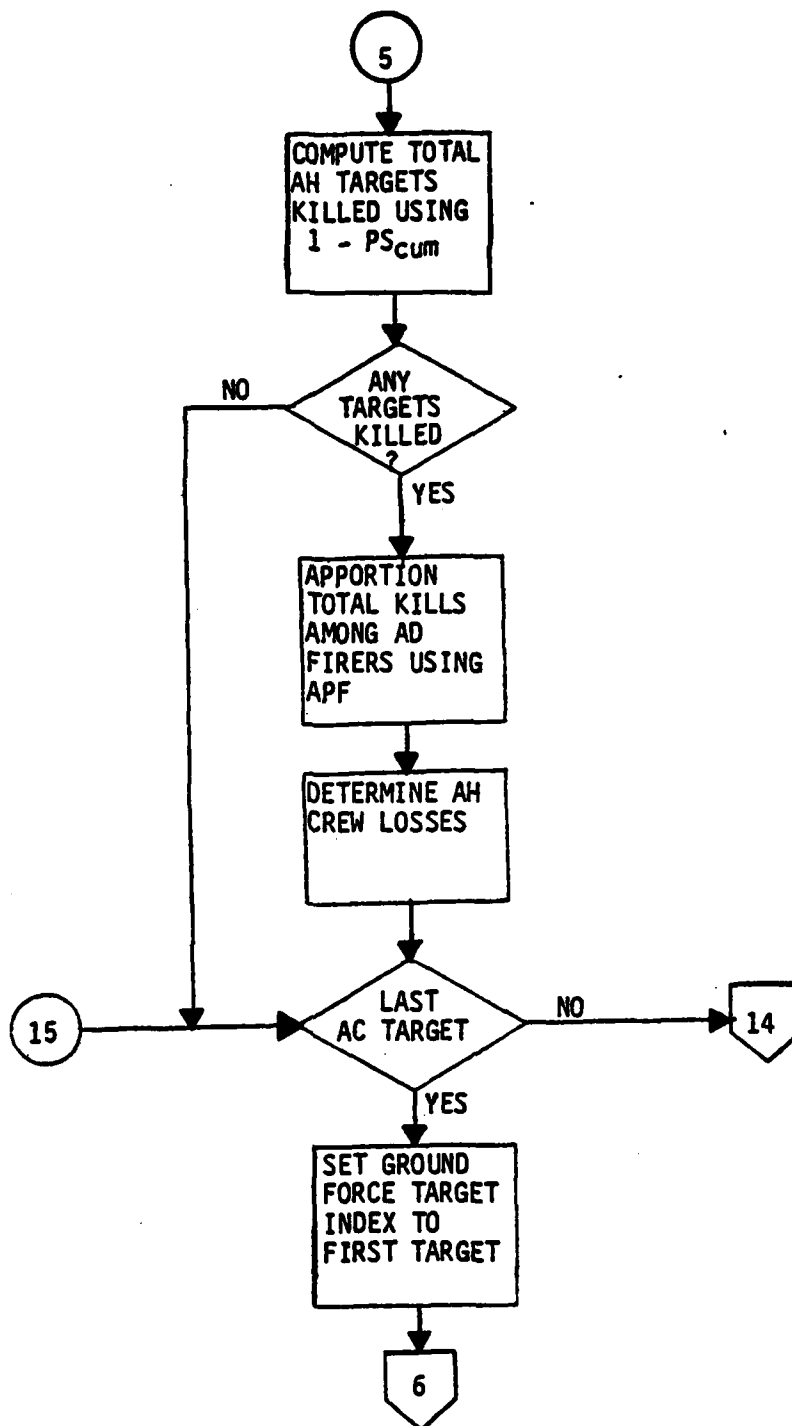


Figure H-1 AHAD flow diagram (continued).

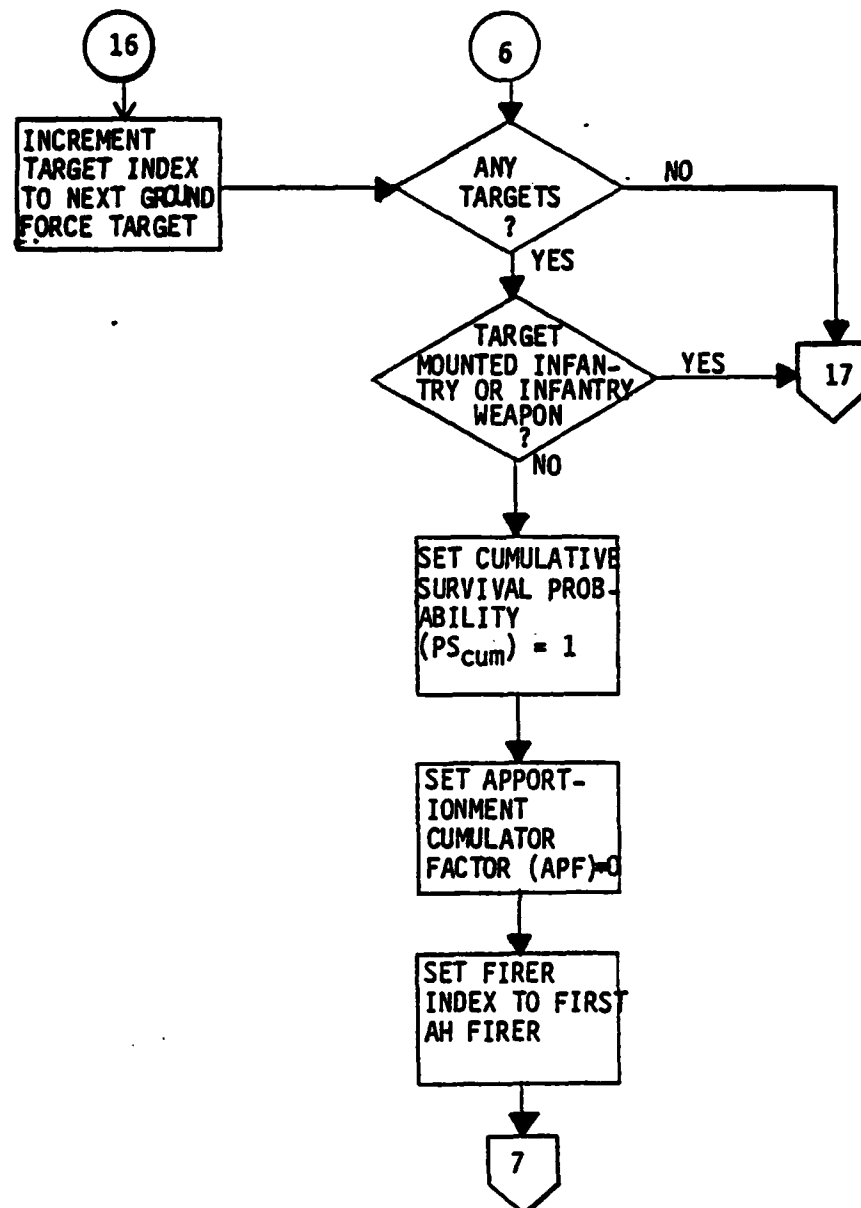


Figure H-1 AHAD flow diagram (continued).

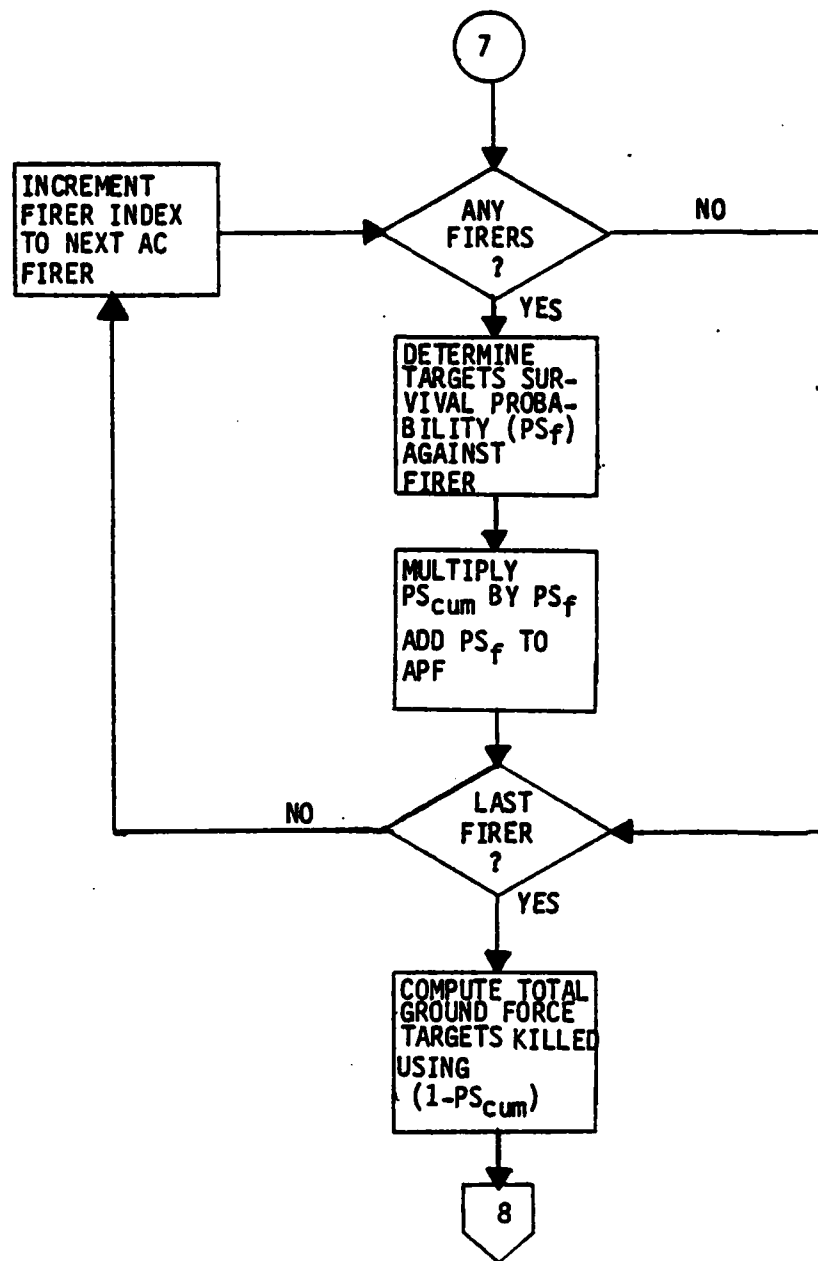


Figure H-1 AHAD flow diagram (continued).

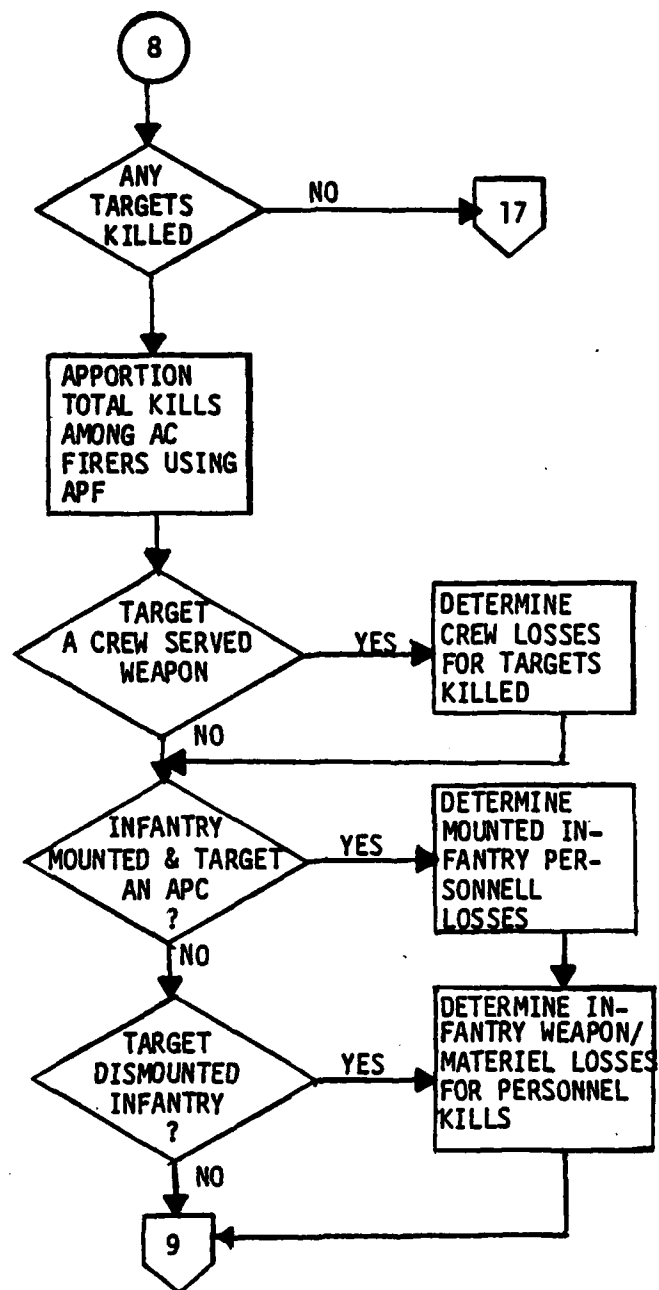


Figure H-1 AHAD flow diagram (continued).

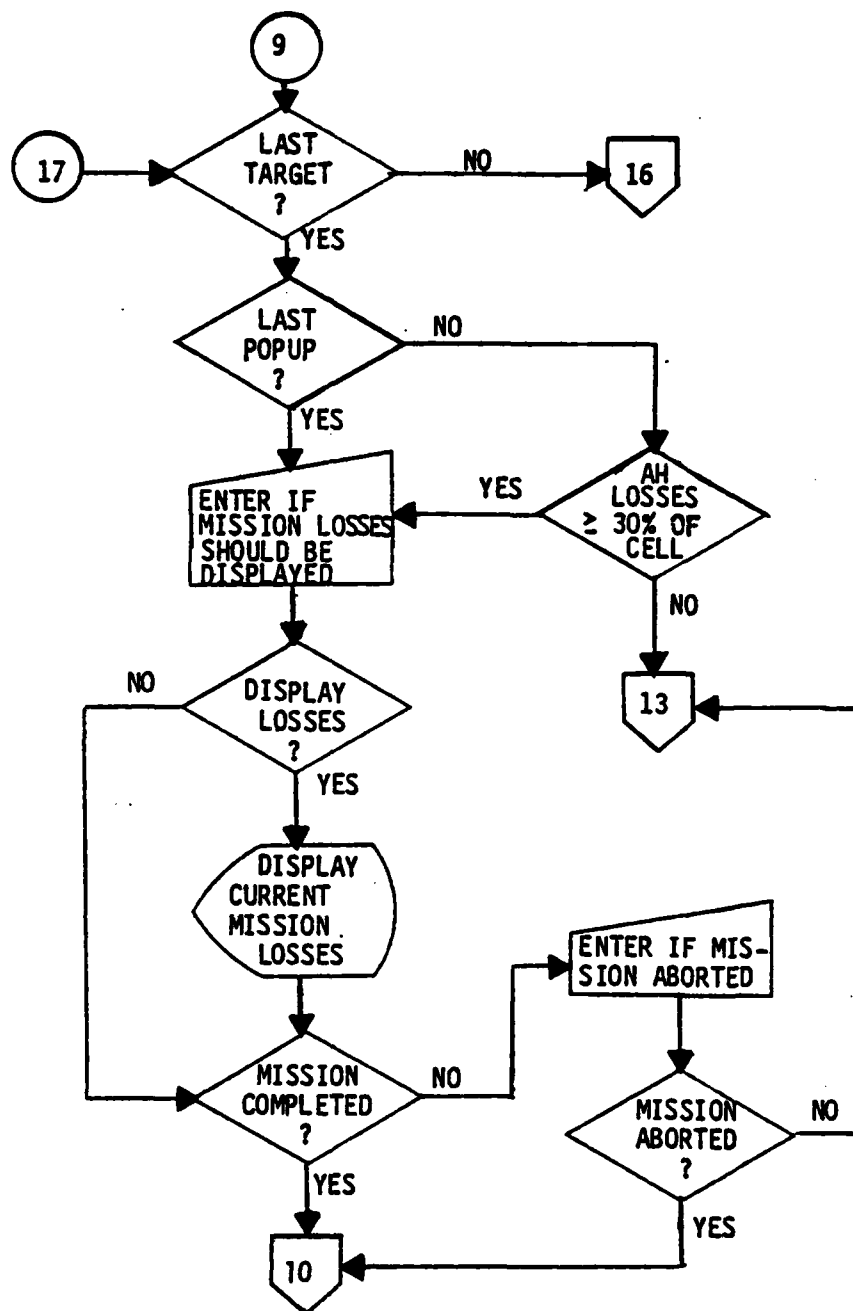


Figure H-1 AHAD flow diagram (continued).

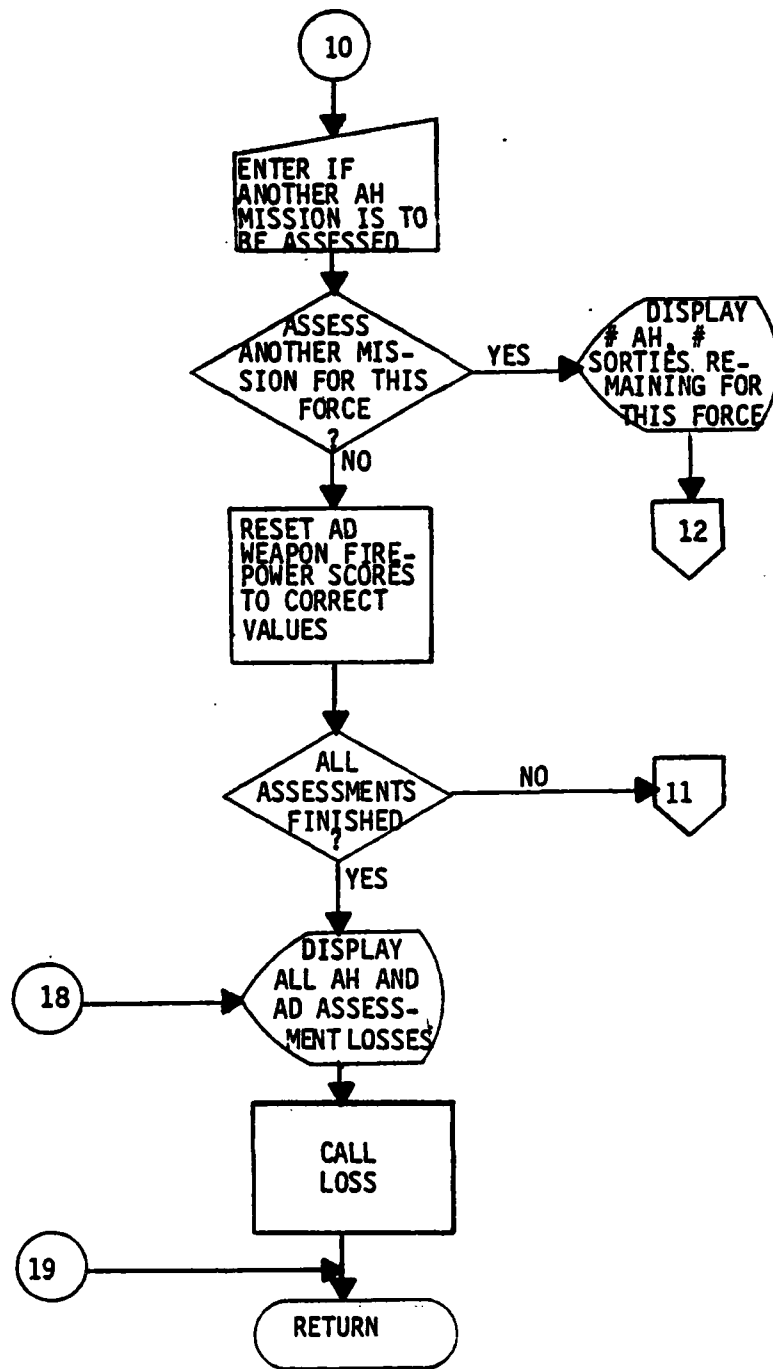


Figure H-1 AHAD flow diagram (concluded).

APPENDIX H

SECTION II. SUBROUTINE CANNON

H-II-1. SUBROUTINE DESCRIPTION. CANNON is the first combat assessment routine called by the supervisory program (SUPER) from DECISION POINT number 3 (see table 1). It requires data arrays from the classified random access file (CLDATA) in addition to the data initiated within the program itself. CANNON performs nearly all the assessments associated with mortar and field artillery fire and also displays the losses from all indirect fire missions. The routine requires a number of gamer inputs to specify the types of indirect fire missions being assessed and to set parameters that are used in the actual assessment computations. The program cycles through several nested DO loops in making the loss calculations in order to assess all possible target/firer combinations; this is done for each force firing at the opposing force and for each phase of indirect fire combat being assessed. The only indirect fire assessment not included in the CANNON routine is for cannon-launched guided projectiles (CLGP). CLGP missions are available only to the Blue force and are assessed by calling the subroutine CLGP. When all assessments have been completed, the cumulative losses are displayed, the LOSS subroutine is called, and control is returned to the supervisory program.

H-II-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
ACQ	Acquisition factor
ADSF	Air defense suppression mission flag
AHI	States record indicator
AKILL	Target survival probability against all firers
AT	Number of homogeneous area targets
BMT	Battery missions per tube
BMTH	HE missions
BMTI	ICM mission
C	Combat level variable
CBTLEV	IDF combat level required
CLEV	IDF combat level
CLOST	Crewmen lost
CM	Total CLGP missions fired
CM50	CLGP missions fired by weapon 50
CM53	CLGP missions fired by weapon 53
DE	Fraction of IDP systems deployed
F	Fraction of missions which are targeted
FAC	Fire allocation constant
FDF	Fire distribution factor
FDT	Fractional damage table
FDTATH	HE fractional damage divided by number of targets

<u>Variable</u>	<u>Description</u>
FDTATI	ICM fractional damage divided by number of targets
FLAG	Flag for type of IDF mission
FPF	Length of final protective fires (minutes)
HOURS	Length of IDF mission (hours)
HR	Length of IDF support (hours)
HRARTY	Length of artillery support (hours)
I	Firer weapon integer index
IC	ICM round type variables
ICAN	Debug flag
ICAT	IDF weapon category index
ICB	Counterbattery mission flag
ICM	ICM round type array
ICS	Close support mission flag
IFLAG	Mission flag index
II	Firer weapon mapping index
INC	History file field index
INX	Input response variable
IOP	Suppression factor index
IPOINT	Output header flag
IS	AMMO array index
ISHOT	Ammunition expenditure index
I50	Weapon 50 CLGP fire flag
I53	Weapon 52 CLGP fire flag
J	Firer's force integer index
K	Victim weapon integer index
KIND	Force color
KK	Target mapping index
L	Victim's weapon integer index
MAP	IDF target mapping array
MW	Military worth array
MWROFB	Military worth array or blue systems
MWTH	Military worth
MWTHX	Military worth variable
N	Do loop variable
NWD	History file field index
OA	Operational availability
OPERA	Operational availability
P	Gamer percent response variable
PCM	Percent not degraded by EW
PCML	Percent not degraded by smoke
PERSF	Personnel fire mission flag
PIC	Percent ICMs variable
PICM	Percent ICM array
PKILL	Target survival probability against firer
PLOSS	Total victims killed
PNTAR	Percent not targeted

<u>Variable</u>	<u>Description</u>
POK	Percent of knowledge
PREP	Lengths of prep/counter-prep fires (minutes)
RDF	Rate of fire
RPM	Rounds per mission
S	Suppression factor
SKILL	Loss apportionment factor denominator
SUPR	Weapon suppression constants
TBAT	Tubes per battery
TEMP	Blue firer documentation variable
TEMP2	Red firer documentation variable
TEMP3	Blue target documentation variable
TEMP4	Red target documentation variable
TGT	Elements per target
TKILL	Target losses to IDF systems
X	Ammo calculation variable
X2	Ammo calculation variable
X2	Ammo calculation variable
Z	SP 122's in DF role

NOTE: All common modules are defined in appendix F.

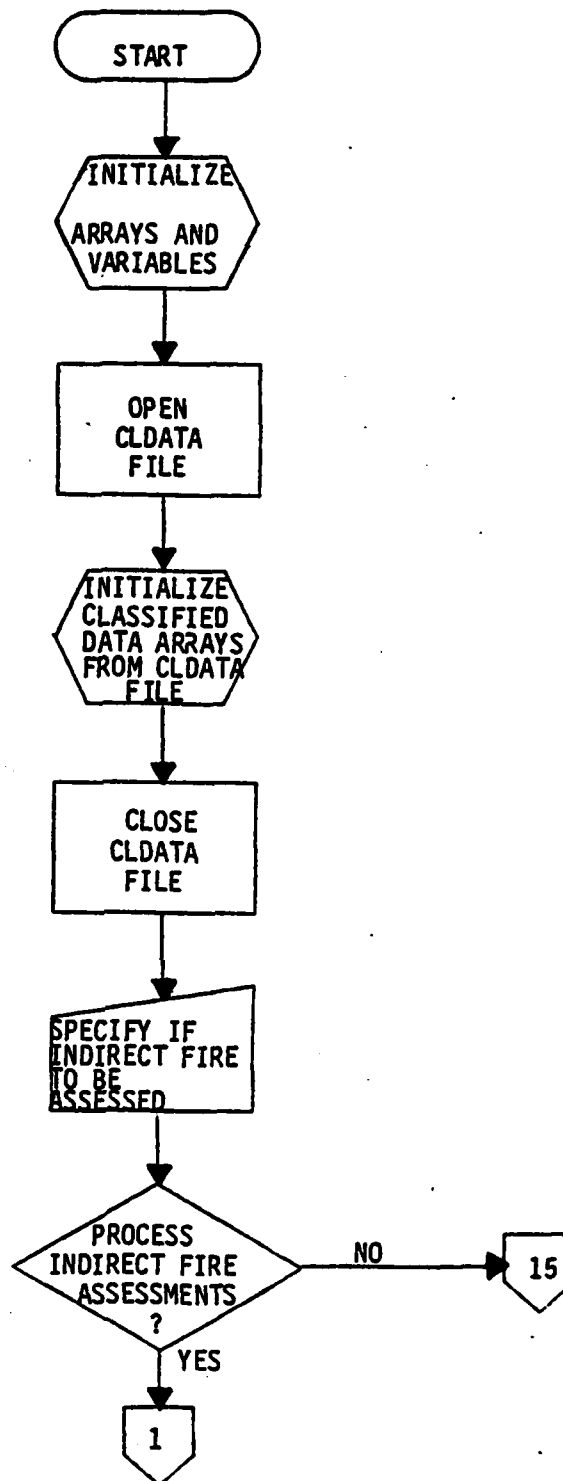


Figure H-2 CANNON flow diagram.
(Continued next page)

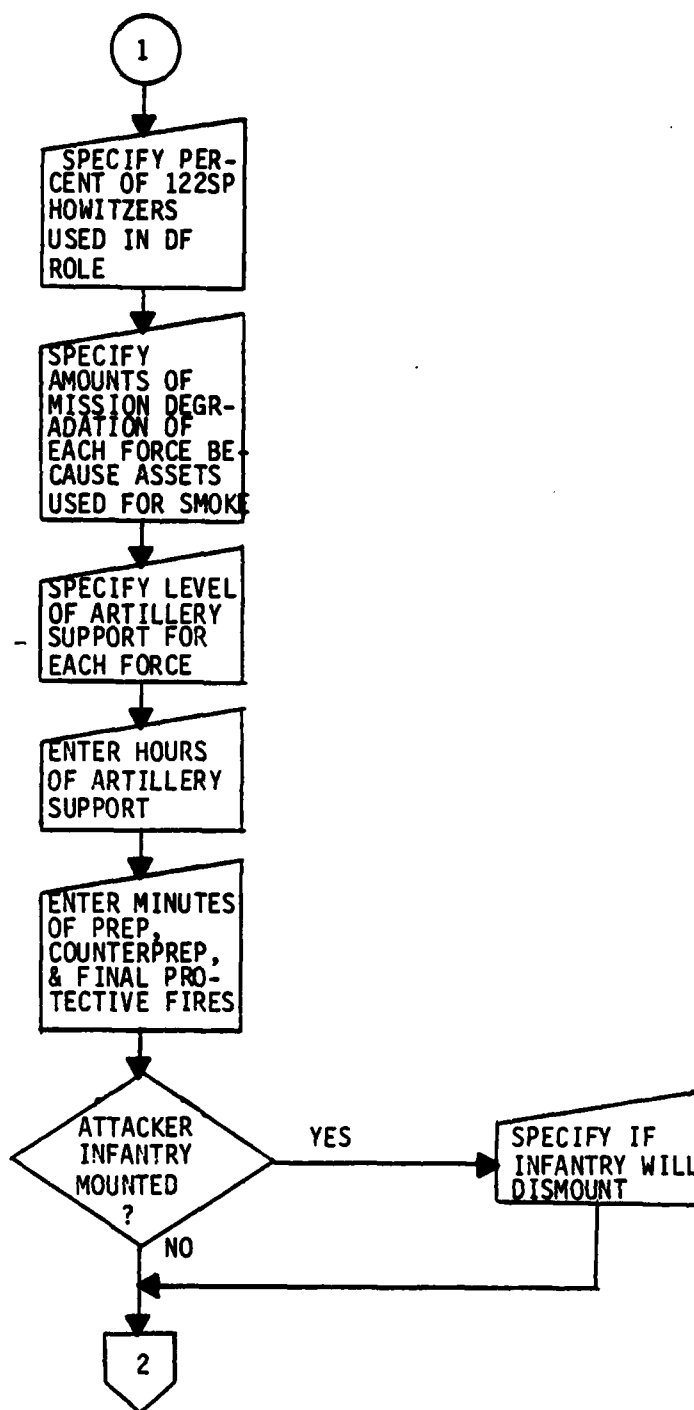


Figure H-2 CANNON flow diagram (continued).

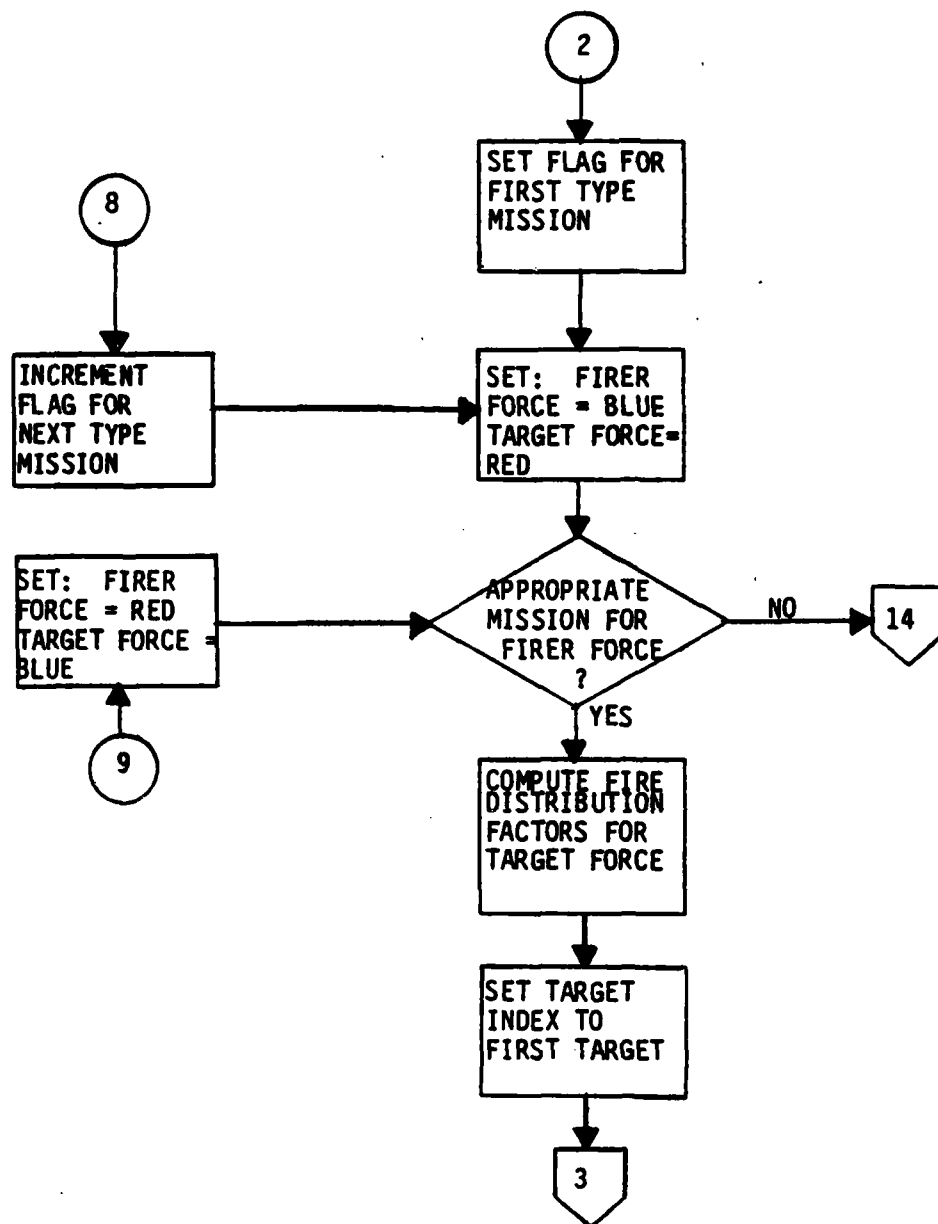


Figure H-2 CANNON flow diagram (continued).

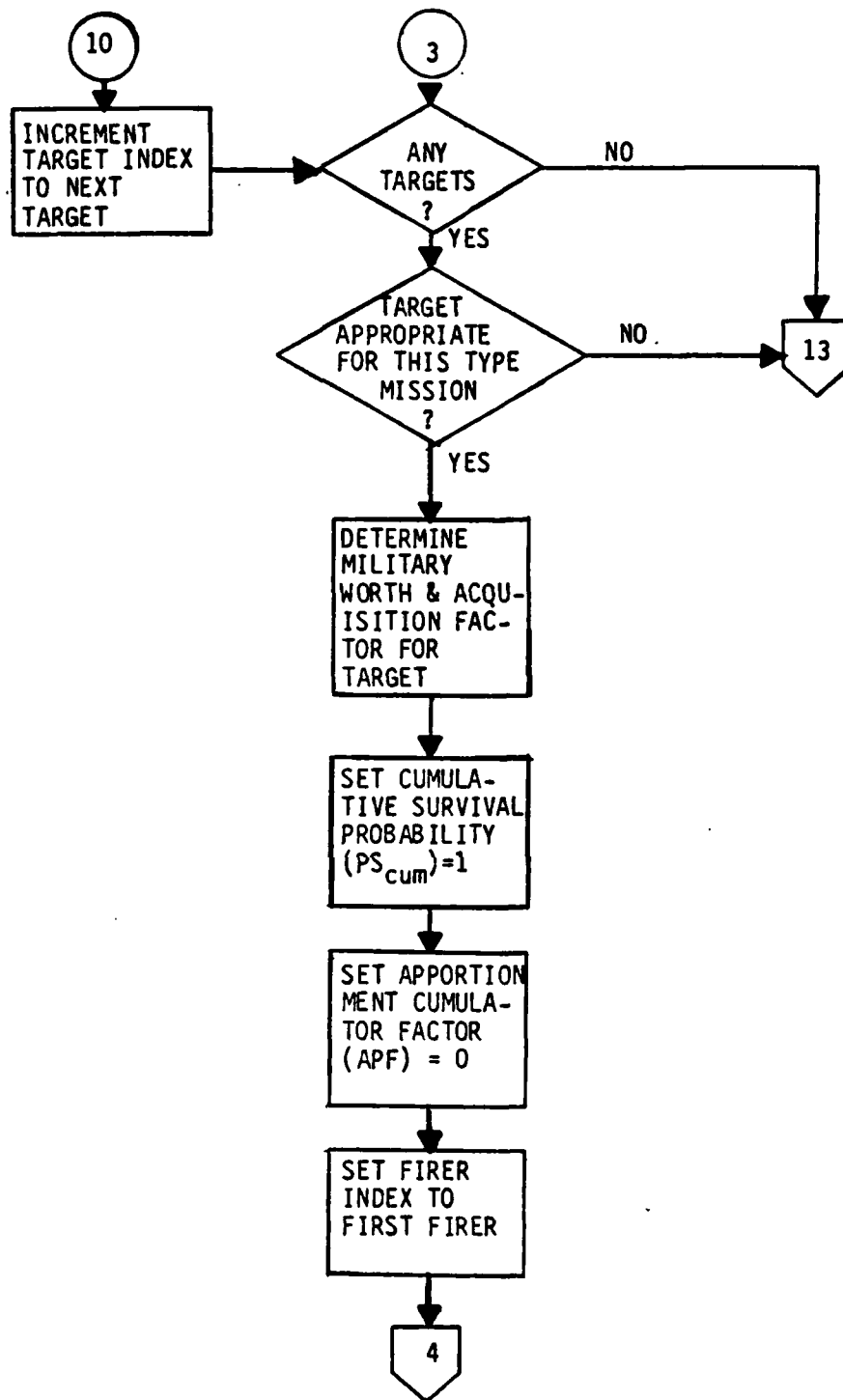


Figure H-2 CANNON flow diagram (continued).

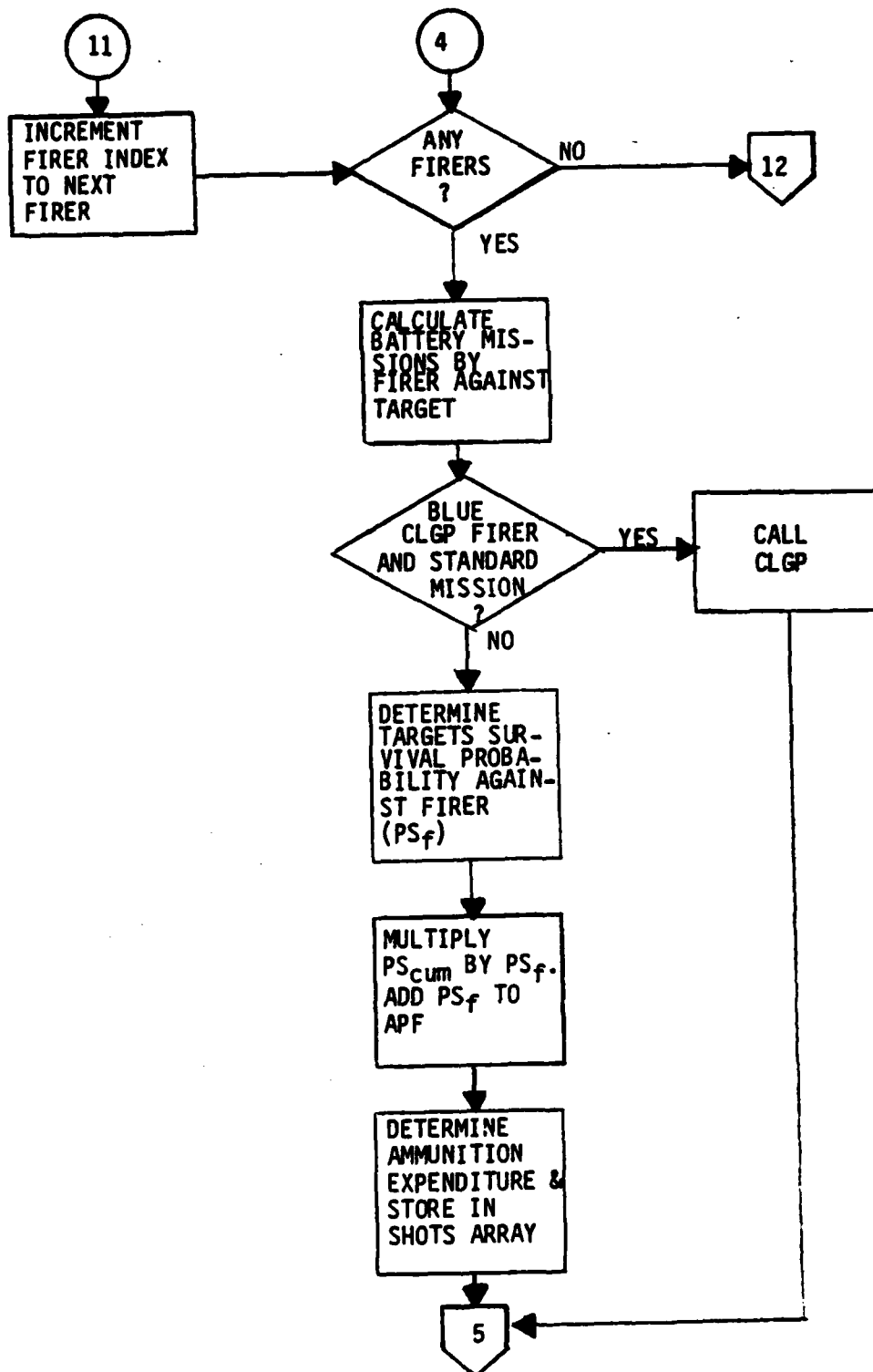


Figure H-2 CANNON flow diagram (continued).

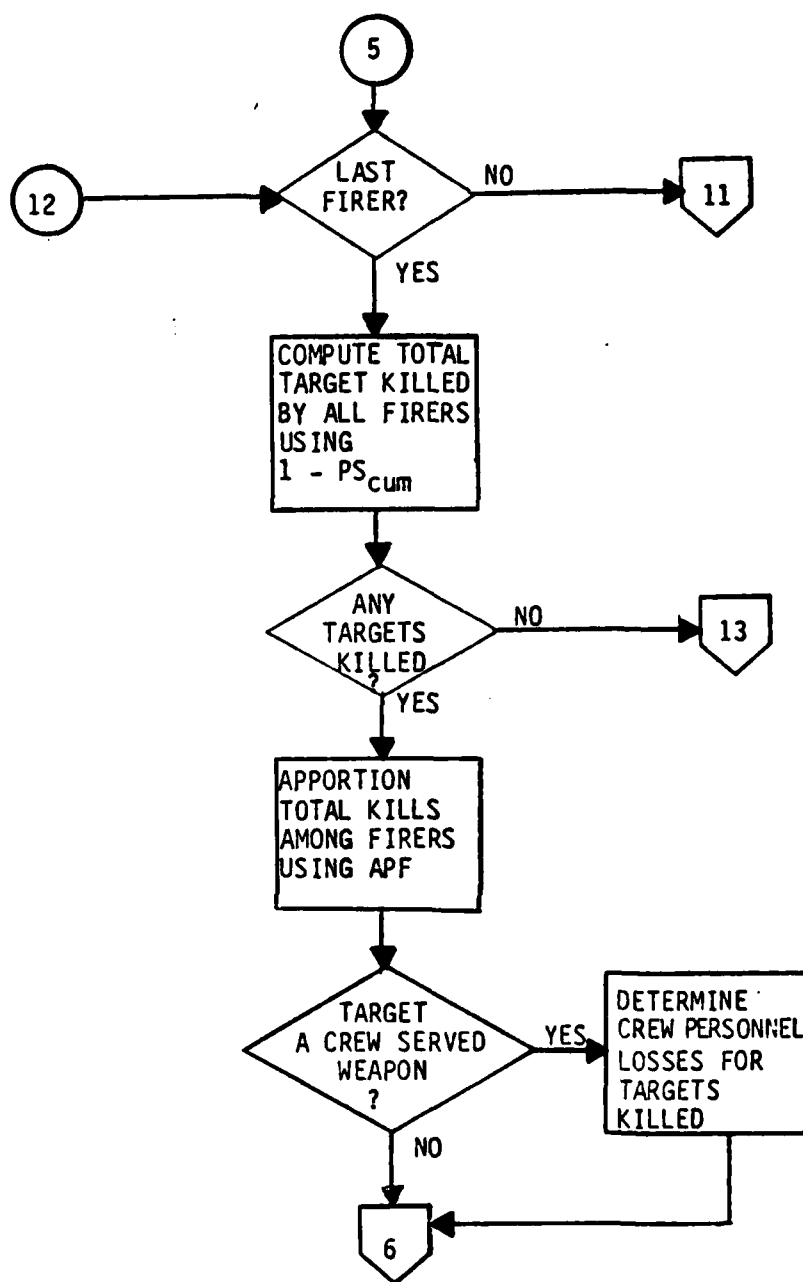


Figure H-2 CANNON

flow diagram (continued).

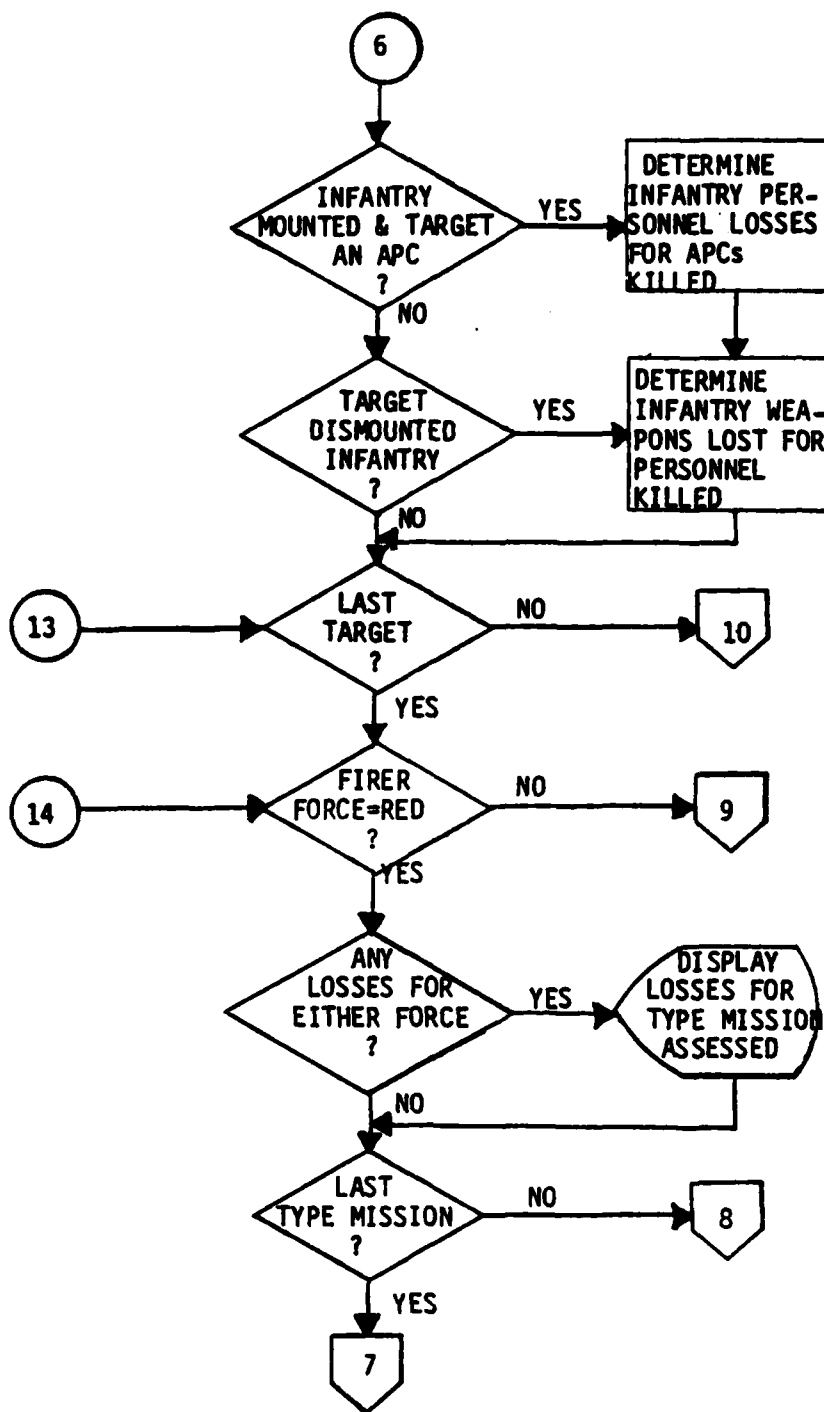


Figure H-2 CANNON

flow diagram (continued).

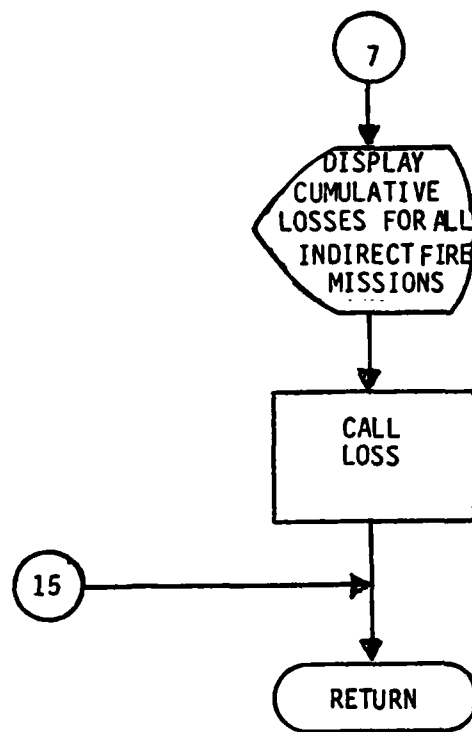


Figure H-2 CANNON flow diagram (concluded).

APPENDIX H

SECTION III. SUBROUTINE CLGP

H-III-1. SUBROUTINE DESCRIPTION. Subroutine CLGP is accessed from the indirect fire program to determine losses of Red weapons to Blue CLGP fire. The gamer inputs required are the number of CLGP missions to be fired, whether the aerial designator is in use and the cloud height. The losses are returned to the main program and displayed from there as part of the indirect fire results.

H-III-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
ADUST	Dust effect factor
AKILL	Target losses to CLGP fire
BMT	Battery missions per tube
CM	Total CLGP missions fired
CMS	Missions fired counter
DUST	Dust array
FARD	Friendly barrage factor
FD	Fractional damage
FDF	Fire distribution factor
G	Percent not aborted
I	Firer weapon index
ICH	Cloud height
IEL	Do loop variable
INX	Aerial designation flag
IPSN	Positioning units index for contact
IS	Aerial designator flag
J	Firer force index
K	Target weapon index
KK	Infantry weapon index
L	Target for index
MAX	Maximum CLGP mission to fire
OA	Operational availability
OAV	Operational availability times percent not smoked
OPERA	Operational availability
PCNSM	Percent not smoked
PEXP	Terrain adjustment factor
PRCN	GLLD suppression or RPV survivability factor
PLOSS	Total victims killed
PREC	CLGP SSKPs and GLLD suppression factor
R	Number of CLGP rounds fired
S	Response control switch
T	Number of targets available to CLGP firer
TARD	Threat barrage factor

NOTE: All common variables are defined in appendix F.

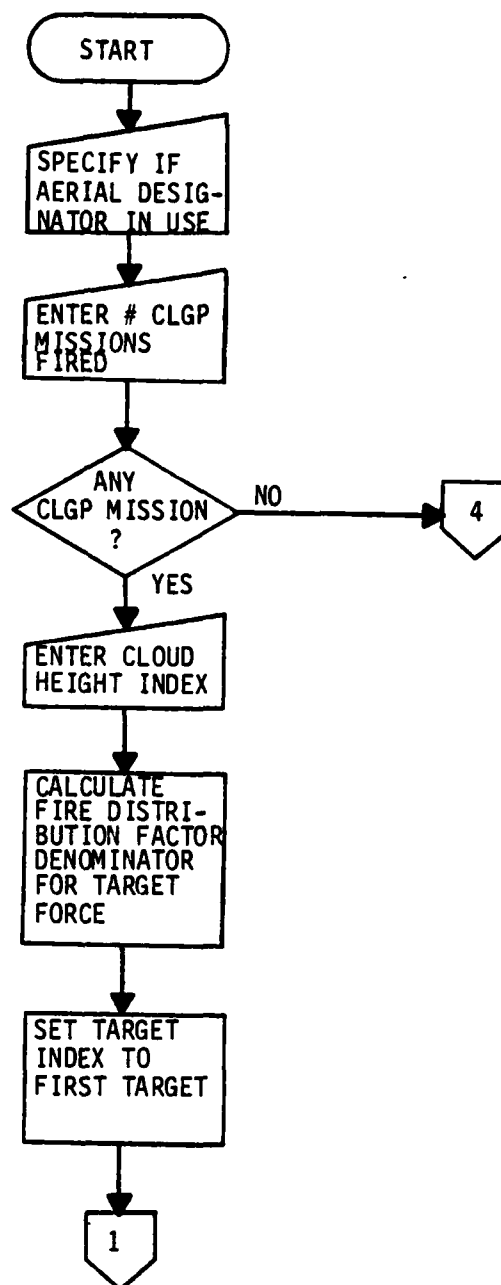


Figure H-3 CLGP flow diagram. (Continued next page)

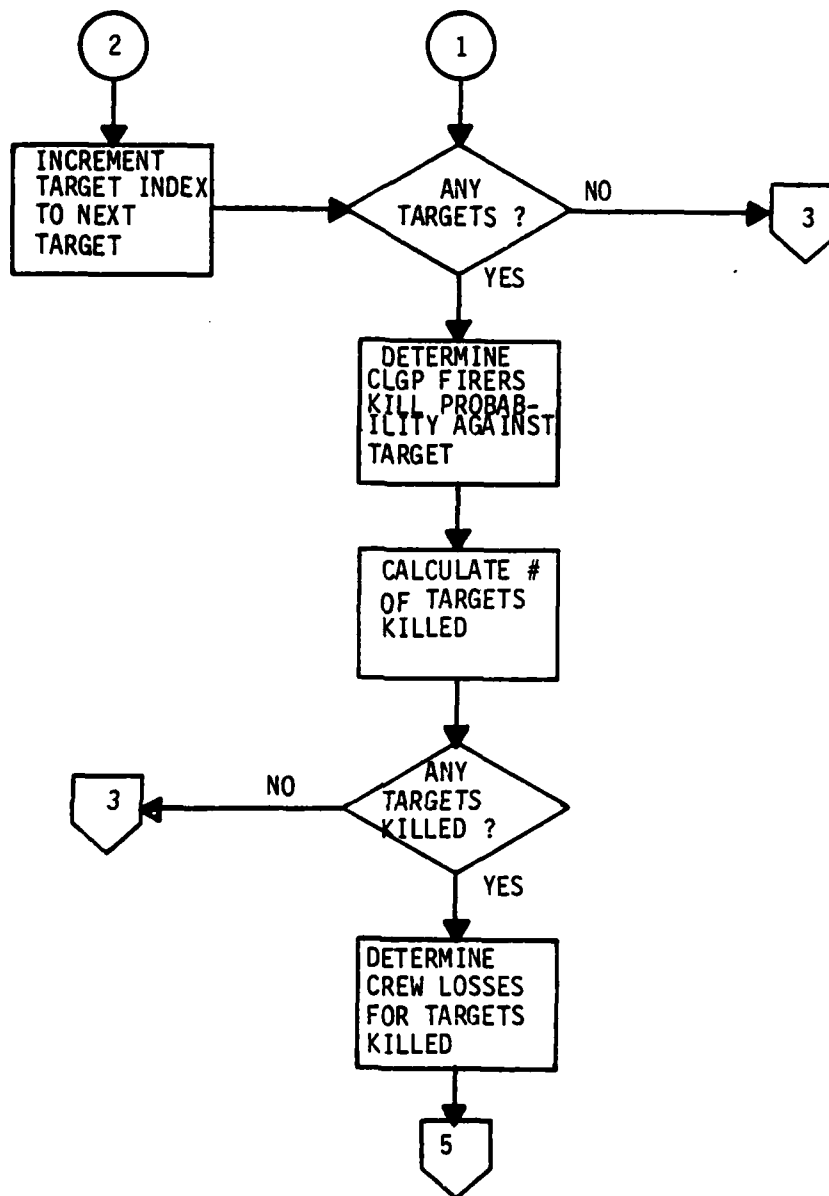


Figure H-3 CLGP flow diagram (continued).

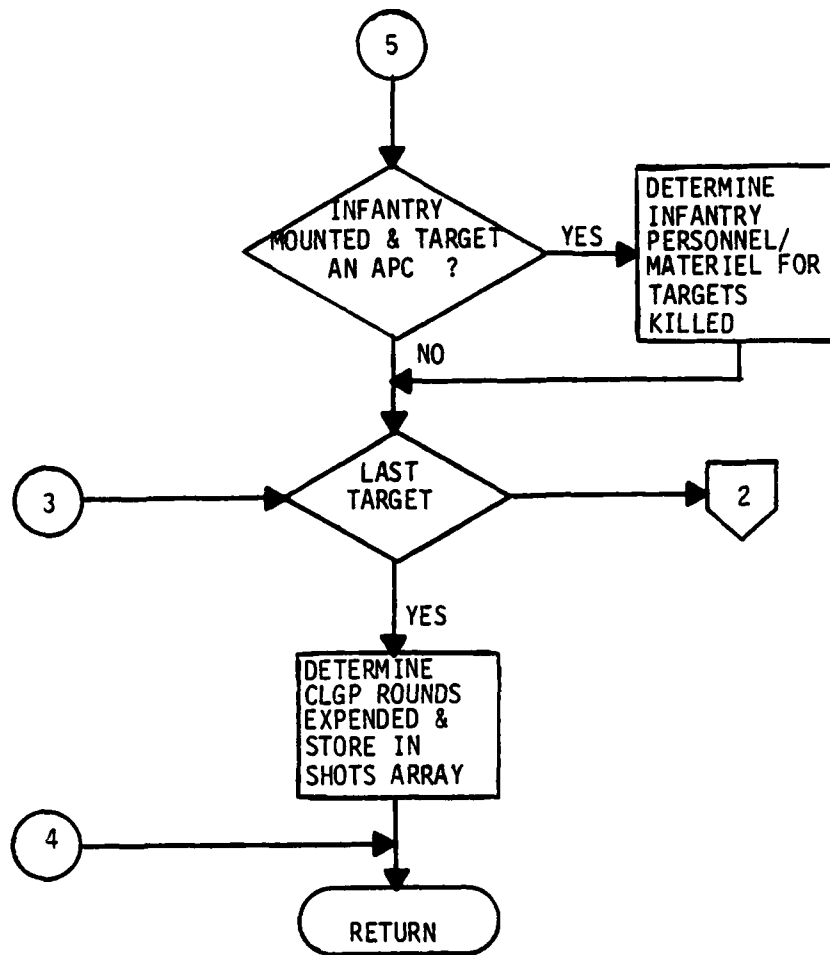


Figure H-3 CLGP flow diagram (concluded).

APPENDIX H

SECTION IV. SUBROUTINE FASCAM

H-IV-1. SUBROUTINE DESCRIPTION. This subroutine contains the logic used to assess losses to minefields composed of scatterable mines (FASCAM). The subroutine is called from the mine subroutine whenever the gamer specifies that a FASCAM minefield assessment is being processed. Although the assessment computation logic is essentially the same as for conventional minefields, the FASCAM minefields require a different set of inputs and casualty rate data.

H-IV-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AKILL	Attacker weapon system kills
CLOST	Crewmen lost
DEPTH	Minefield depth array
FATCAS	Percent tank casualties by FASCAM mines
FPCAS	Percent personnel casualties by FASCAM mines
FROBY	Minefield frontage bypassed by attacker
FRONT	Minefield frontage
II	Type of FASCAM delivery system
INX	Input response variable
J	Force index
K	Target weapon index
KK	Target weapon index
P	Percent of force entering minefield
PERCOV	Percent of units front covered by mines
PLOSS	Total victims killed
PMFNBY	Percent of MP not bypassed by attacker
TRZONE	Terrain trafficable by armor

NOTE: All common variables are defined in appendix F.

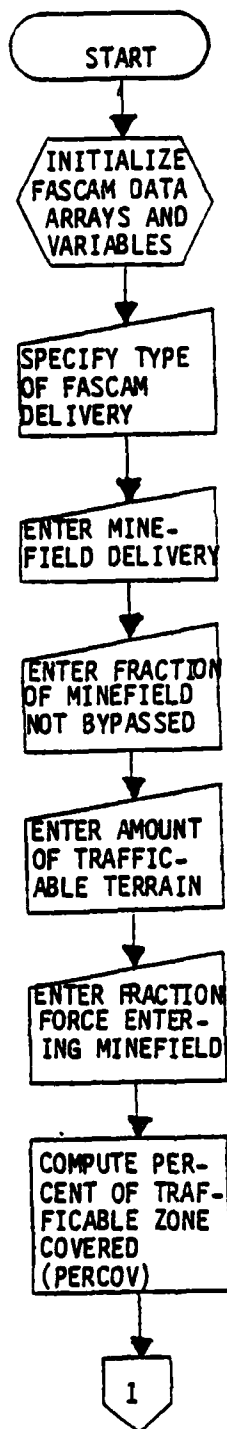


Figure H-4 FASCAM flow diagram.
(Continued next page)

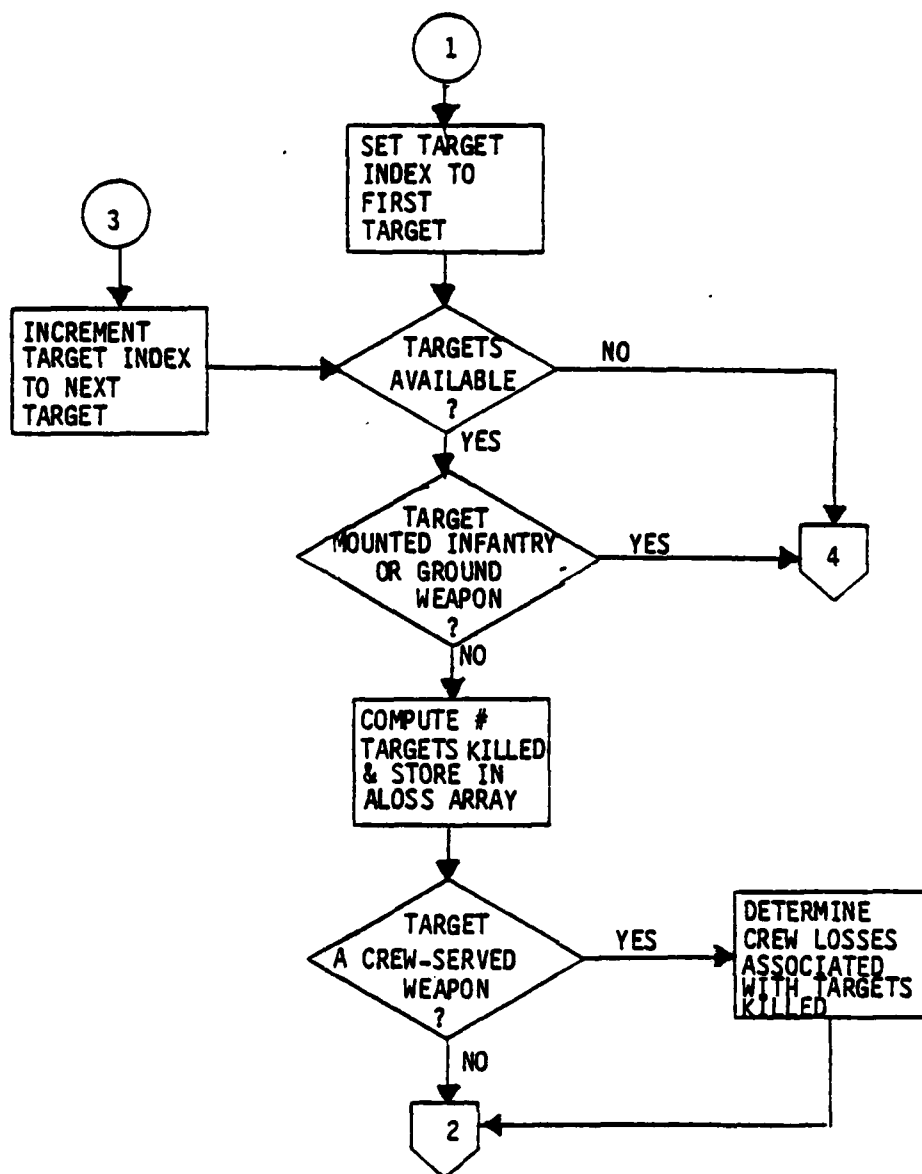


Figure H-4 FASCAM flow diagram (continued).

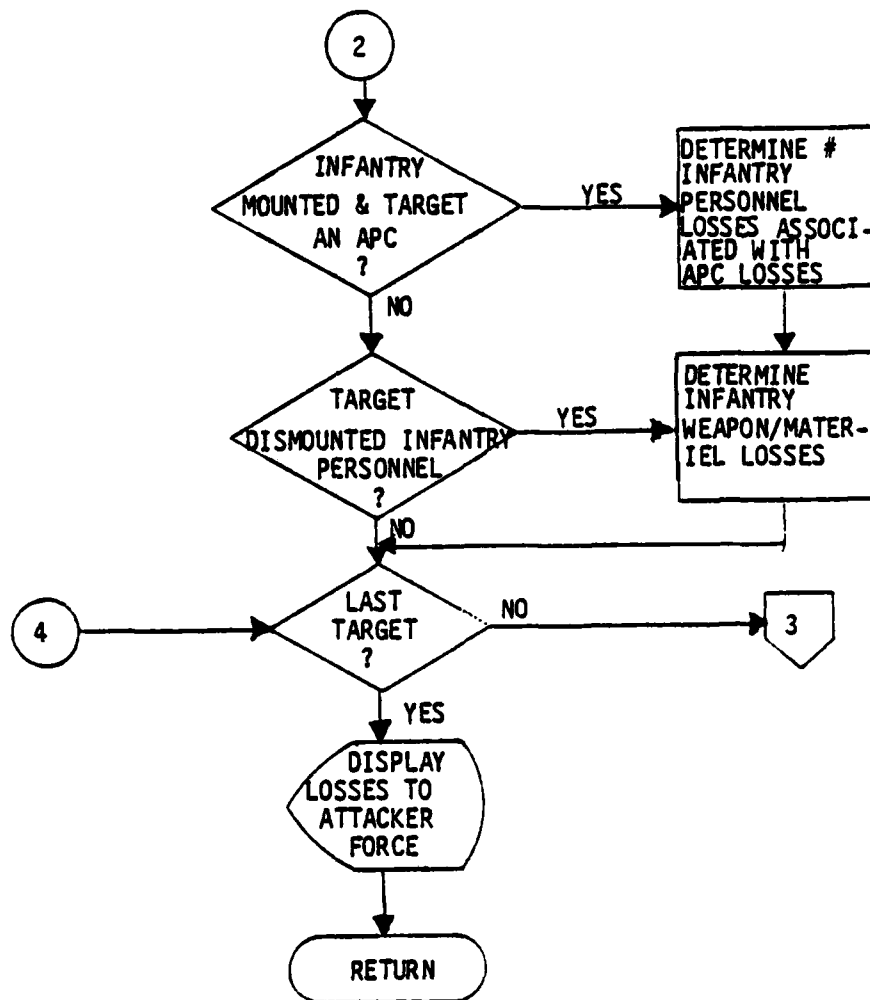


Figure H-4 FASCAM flow diagram (concluded).

APPENDIX H

SECTION V. SUBROUTINE INFANT

H-V-1. SUBROUTINE DESCRIPTION. INFANT is the fourth of the combat assessment routines called in the main Jiffy Game program from decision point 3 (see table 1). It is called whenever both forces contain infantry personnel in the weapon system (ELMT) array. The function of this subroutine is to compute and display the losses incurred as a result of dismounted infantry combat for the sector being gamed. The subroutine requires a number of interactive gamer inputs, which set the parameters necessary to carry out a one-time assessment of ambush and/or conventional dismounted infantry combat casualties suffered by each force. When assessments are finished, the results are displayed, the LOSS routine is called, and control is returned to the main program.

H-V-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
A	Ambush personnel casualty rate
AHI	History file parameter
AIL	Infantry attacker losses
AT	Personnel allocated to infantry attack
ATRIT	Personnel casualties for ambushed unit
DF	Defender's personnel casualty rate
DIL	Infantry defender losses
DT	Personnel allocated to infantry defense
DTRIT	Personnel casualties for ambushing unit
F	Fraction of maneuver forces committed
FAC	Casualty rate resolution factor
GFPR	Ground combat firepower ratio
GFPS	Ground combat firepower scores
HR	Hours of combat for assessment
HRC	Hours of conventional combat
I	Target weapon index
IAA	Attacker index in ambush
IEL	Defender index
IFLAG	Logic flag
INDEX	Target weapon flag
INX	Hours of infantry attack
J	Force index
KIND	Force color
L	Target force index
STABLE	Casualty rate index determiner array
TABLE	Ground combat personnel casualty rate
TABLE 3	Ambush personnel casualties

NOTE: All common variables are defined in appendix F.

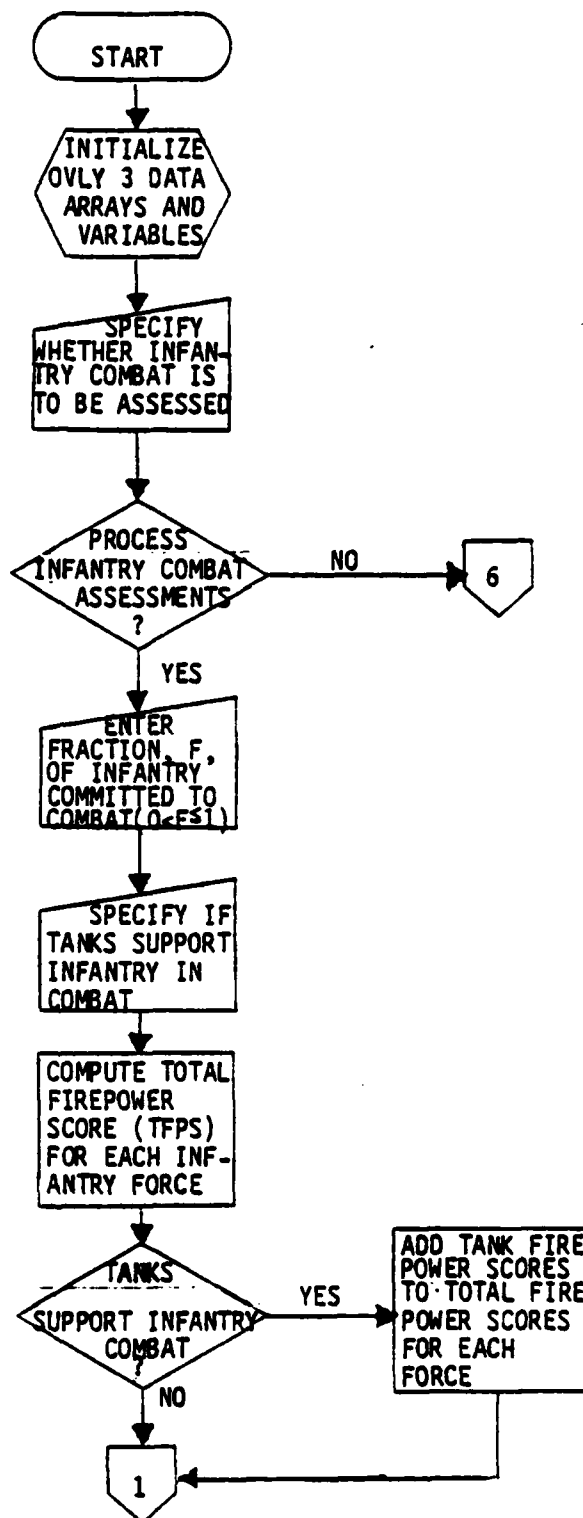


Figure H-5 INFANT flow diagram.
(Continued next page)

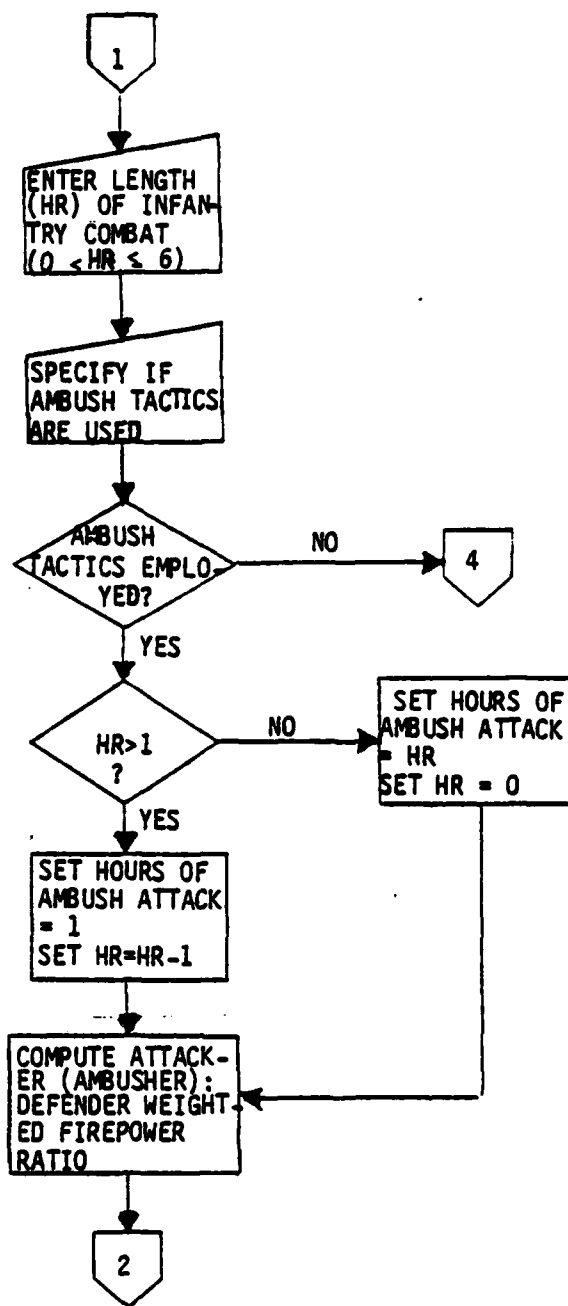


Figure H-5 INFANT

flow diagram (continued).

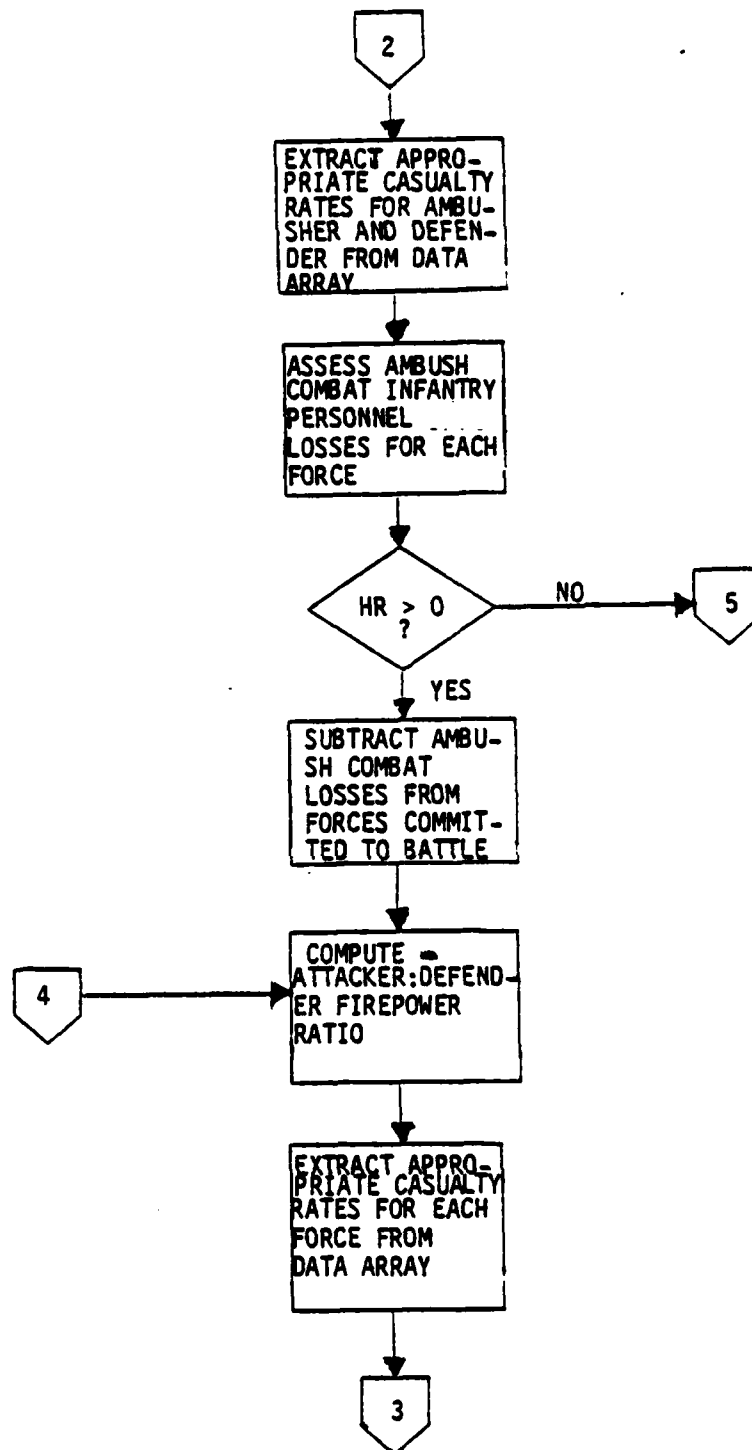


Figure H-5 INFANT

flow diagram (continued).

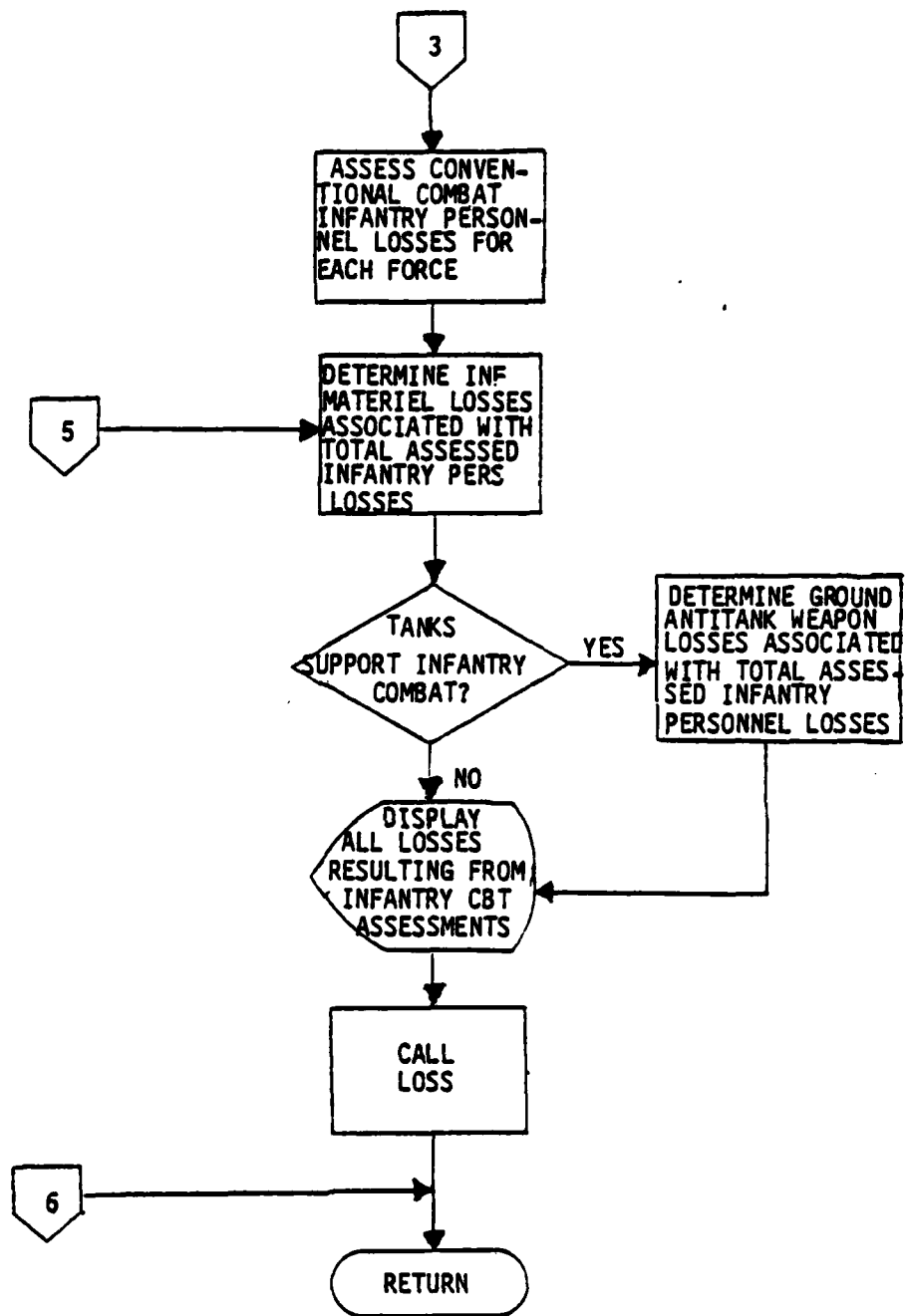


Figure H-5. INFANT flow diagram (concluded).

APPENDIX H

SECTION VI. SUBROUTINE MINE

H-VI-1. SUBROUTINE DESCRIPTION. The primary function of the MINE program is to assess and display the losses suffered by the attacking force to minefields emplaced manually or mechanically (i.e., conventional minefields). MINE also contains the control point at which the type of minefield employed is specified interactively by the gamer. At the end of any minefield assessment, the program returns to this control point; thus, several assessments can be processed employing the same or different types of minefields before control is returned to the supervisory program. Only a minimal amount of data is needed to assess minefield losses; most of the necessary parameters are set interactively by gamer inputs. The processing of assessments is terminated from the control point. After which the LOSS subroutine is called and control is returned to the main program.

H-VI-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AFRONT	Minefield frontage input variable
AHI	History file parameter
AKILL	Attacker weapon system kills
ATDEN	Antitank minefield (MF) density per square meter
ATFAC	Percent tank losses by antitank mines
BMPL	Mine planter platoons
CLOST	Crewmen losses for productive time lost due to enemy
FROBY	Minefield frontage bypassed by attacker
FRONT	Potential minefield frontage
HOURS	Hours required to lay MF strip
HRMAN	Man-hours available for emplacement of mines
HRREQ	Man-hours required to manually emplace mines
IND	Type of mine employment index
INX	Input response variable
IOK	Logic flag
J	Antitank mine density index
K	Target weapon system index
KIND	Force Color
KK	Infantry weapon system index
NUMEN	Number men to emplace mines
P	Percent of force entering minefield
PERCAS	Percent AP mines personnel casualties
PERCOV	Percent of unit's front covered by mines
PHR	Man hours available
PLOSS	Total victims killed

VariableDescription

PMFNBY	Percent of MF not bypassed by attacker
RNMPH	Mine planter hours available
STRIPW	Minefield strip width
TRZONE	Terrain trafficable by armor
WDEGF	Work degradation factor
X	Mine density input variable

NOTE: All common variables are defined in appendix F.

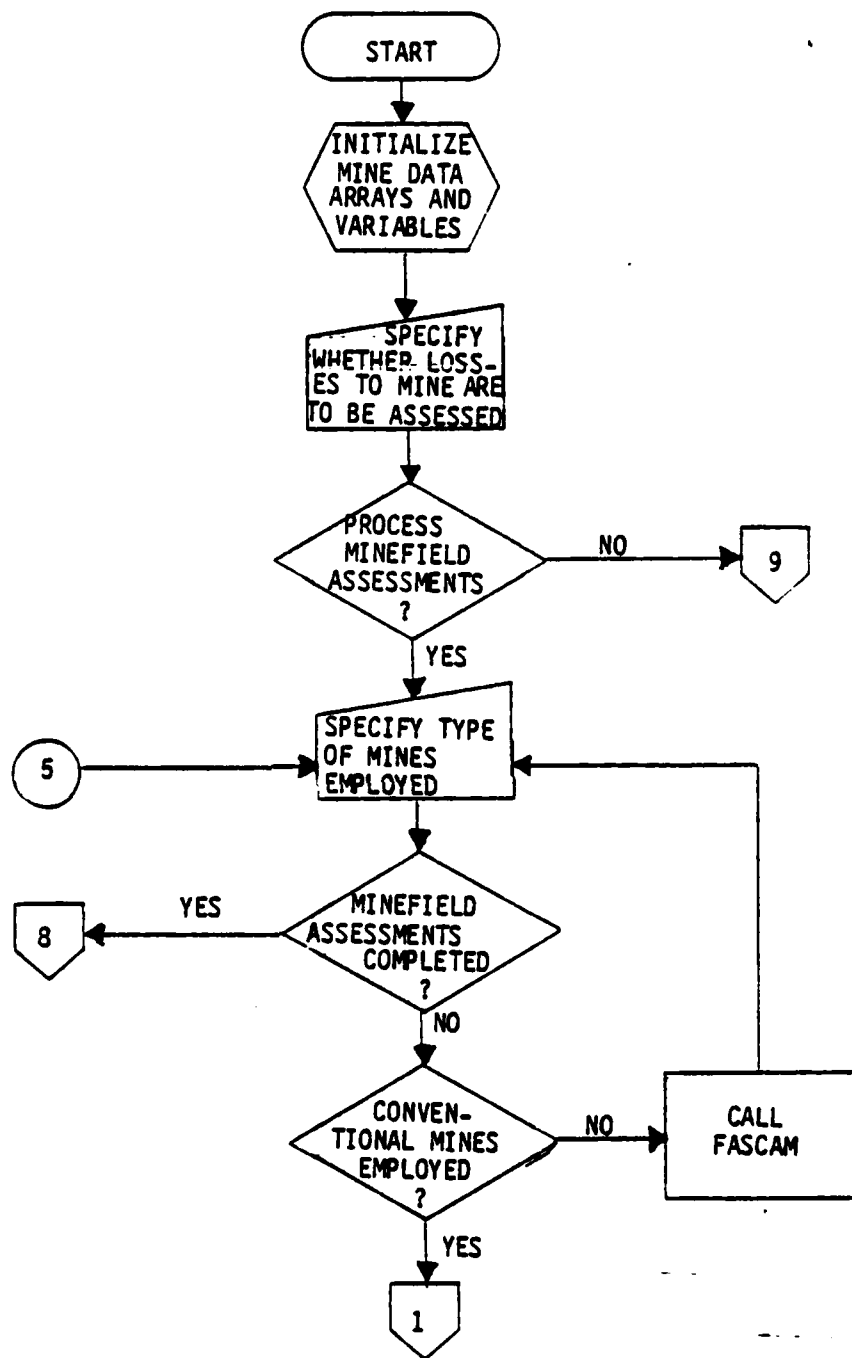


Figure H-6 MINE flow diagram.
(Continued next page)

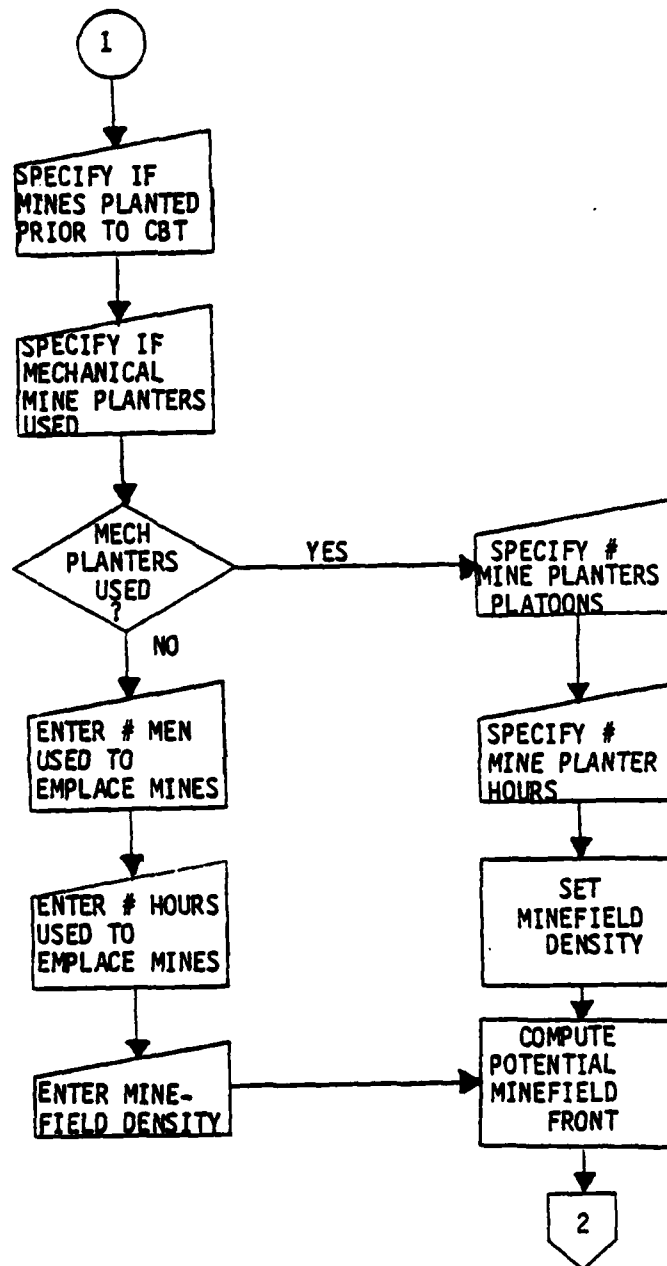


Figure H-6 MINE flow diagram (continued).

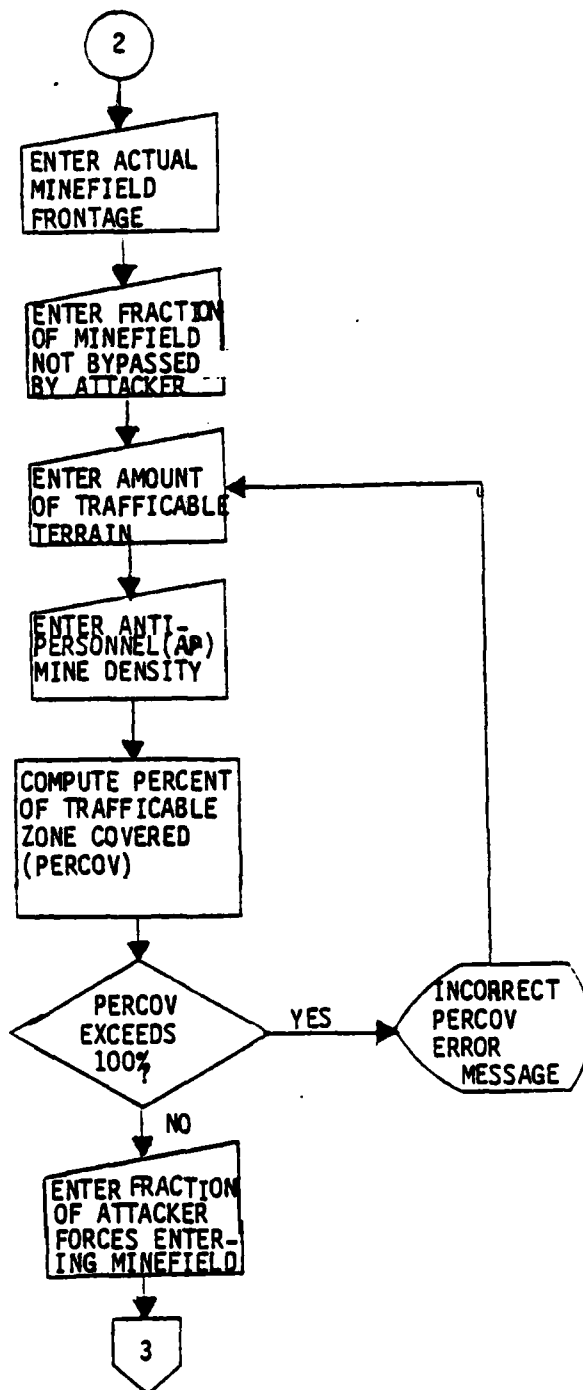


Figure H-6 MINE flow diagram (continued).

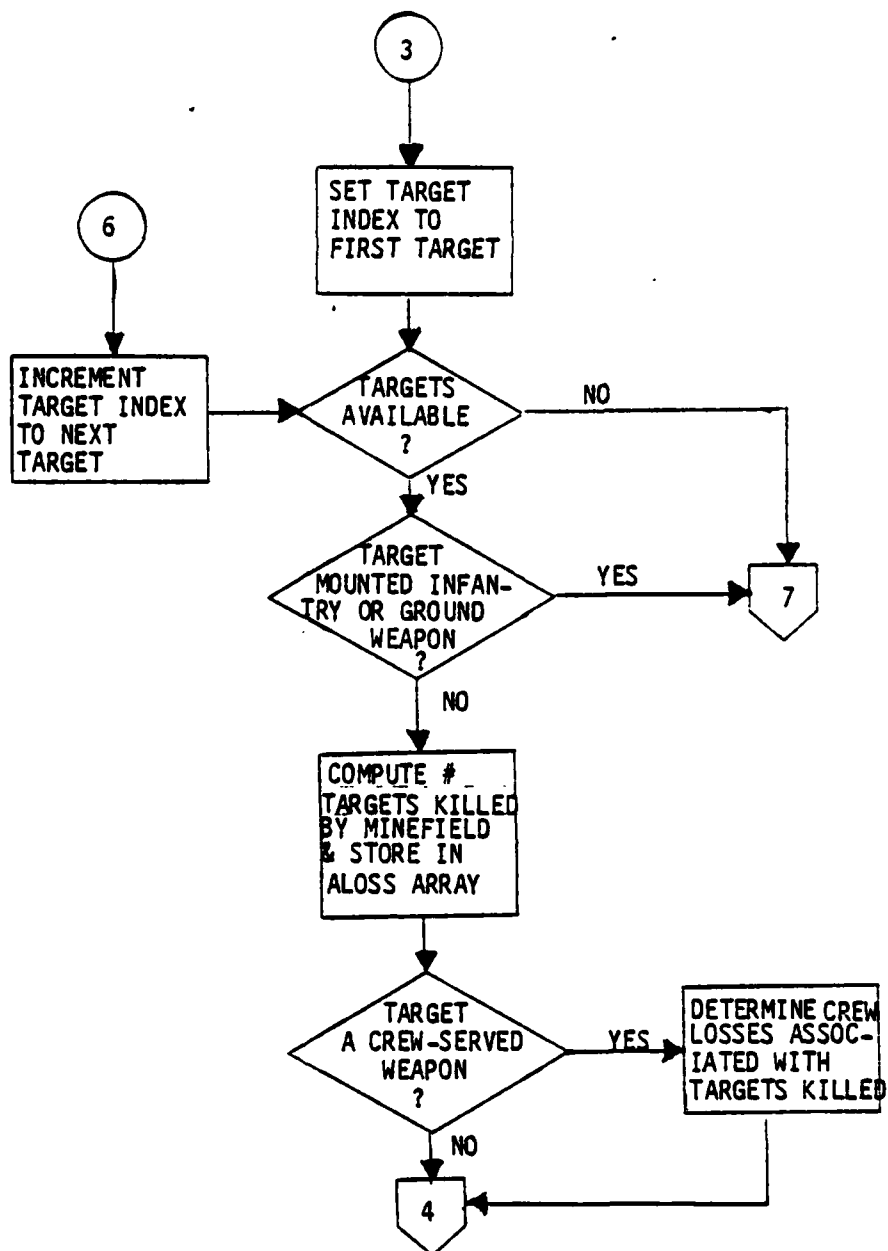


Figure H-6 MINE flow diagram (continued).

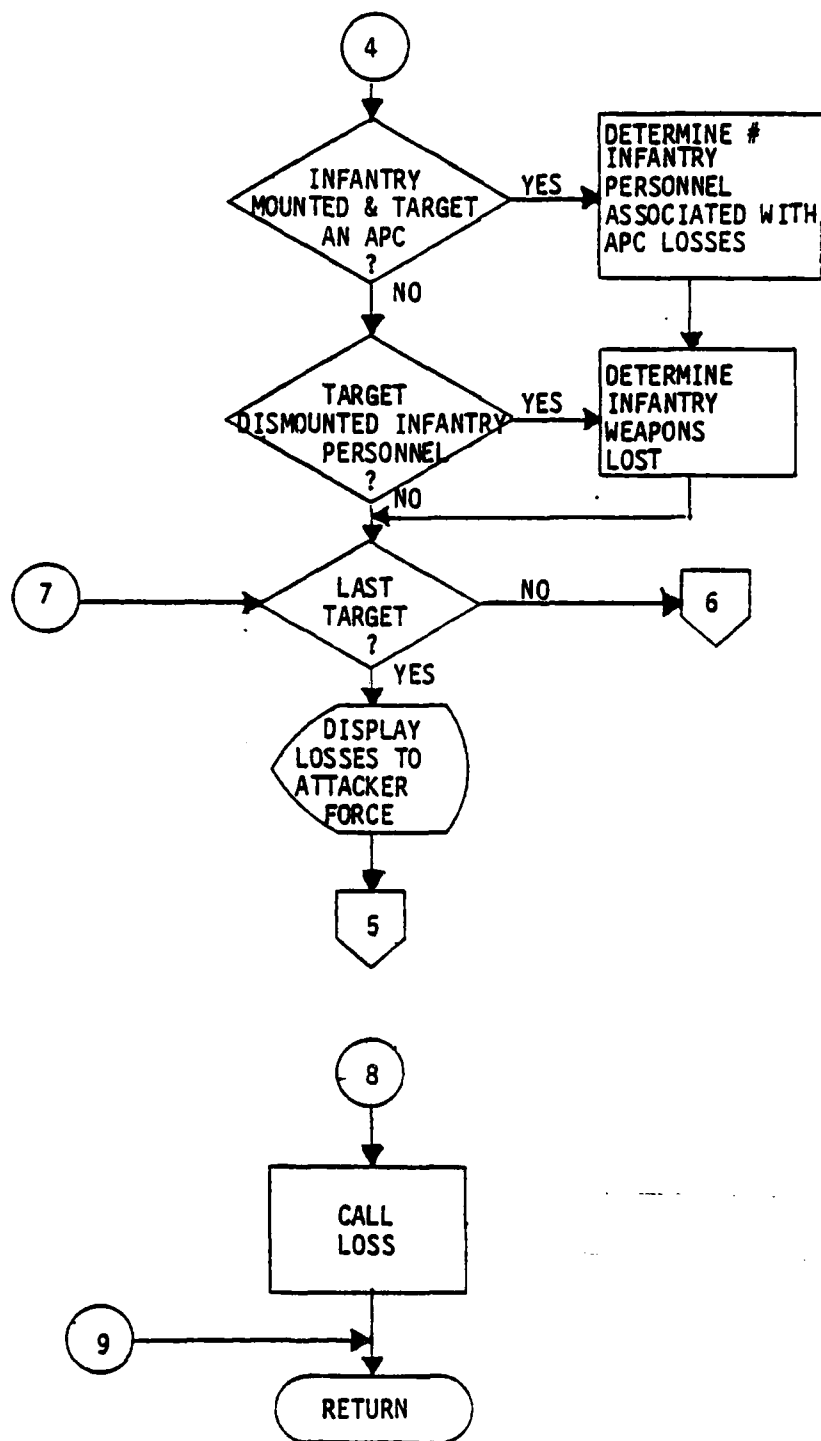


Figure H-6 MINE flow diagram (concluded).

APPENDIX H

SECTION VII. SUBROUTINE TANK

H-VII-1. SUBROUTINE DESCRIPTION. TANK is the third of the combat assessment routines called in the main Jiffy Game program from DECISION POINT number 3 (see table 1). In this routine, the losses due to combat involving tanks, other armored combat vehicles, and antitank weapons are calculated and displayed. The TANK routine cycles through a series of nested DO loops in assessing losses for each possible combination of targets and firers for both forces. The gamer inputs a range band index, which initiates the assessment logic cycle. At the end of each assessment cycle, the gamer either inputs another range band index to continue with another cycle or signals that the assessments are completed. When the assessments are finished, the results are displayed, the LOSS subroutine is called, and control is returned to the SUPER.

H-VII-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
A	Acquisition data array
ABORT	Missile abort factor
ACQ	Acquisition discriminator
ADUST	Dust effects factor
AH1	Starts record identifier
AKILL	Current losses to weapon systems
ATCREW	Number of infantrymen killed per antitank weapon
BLUE	Blue weapon system cumulative losses
CLOST	Crewmen losses
DUST	Dust array
ELMTS	Total number of targets
ELS	Total number of firers
FARD	Friendly barrage factors
FDF	Fire distribution factor
FIRE	Expected number of completed firings
FIRE1	Optimal expected number of firings
FIRE2	Optimal expected number of firings
FOX	Expected number of firing multiplication factor
I	Firer weapon index
IAM	States record ammo index
IBAMMO	Blue round index
ICAT	Target weighting category
IFLAG	Flag for displaying/suppressing table header
IFLAG2	Loss header flag
II	Range index
INC	History file parameter index

<u>Variable</u>	<u>Description</u>
INX	Input response variable
IOV2	Debug print flag
IPUT	Put call switch
IPSN	Positioning units index for attacker/defender
IRAMMO	Red round index
ISUP	Suppression degradation factor index
ITARR	Target documentation array
ITEST1	First firer in debug run
ITEST2	Last firer in debug run
ITYP	Ammunition type index
ITPASS	Visibility pass number
IVVL	Target vulnerability category
J	Firer force index
JDAY	Day/night condition variable
JFLAG	Logical variable
JJ	Attacker/defender firer force index
JOPT	Optical/thermal flag
JPSN	Positioning units index for contact
JSEAS	Season variable
JTERR	Terrain variable
JTERRN	Terrain type array
JTEST1	Side debug variable
JTEST2	Side debug variable
JVIS	Visibility variable
K	Target weapon index
KFLAG	Initial contact flag
KIND	Force color
KK	Category type index
KT	Expected number of completed firings firer index
KTEST1	First target in debug run
KTEST2	Last target in debug run
L	Target force index
LL	Attacker/defender target force index
M	Weapon system (target) index
MAXR	Range index
N	Weapon system (target) index
NWD	History file field index
N1	History file field index
N2	History file field index
OPERA	Weapon system operational availability
P	Gamer percent response variable
PCOM	Percent committed by range band
PICKET	Percent of targets visible
PKILL	Target's survival probability against firer
PLOSS	Current losses to weapon systems
PNSMK	Percent not smoked variable
POPT	Optical Pickett factor

<u>Variable</u>	<u>Description</u>
PSS	Percent self-smoked by range band
PTHM	Thermal Pickett factor
PVN	Percent not smoked array
RED	Red weapon system cumulative losses
ROUNDS	Ammunition fired per target
SET	Thermal/Optical flag
SKILL	Loss apportionment factor denominator
SS	Defilade SSKP/Final SSKP
SSKP	Weapon system single shot kill probability
SSS	Fully exposed SSKP
SUP	Suppression variable
SUPDEG	Suppression degradation factor coefficient
TARD	Threat barrage factors
TKILL	Targets killed
VICTIM	Firer's target
VISDEG	Visibility degradation factor
WTS	Weapon system category weights
XN	Weapon system engagements
XFDF	Fire distribution factor

NOTE: All common variables are defined in appendix F.

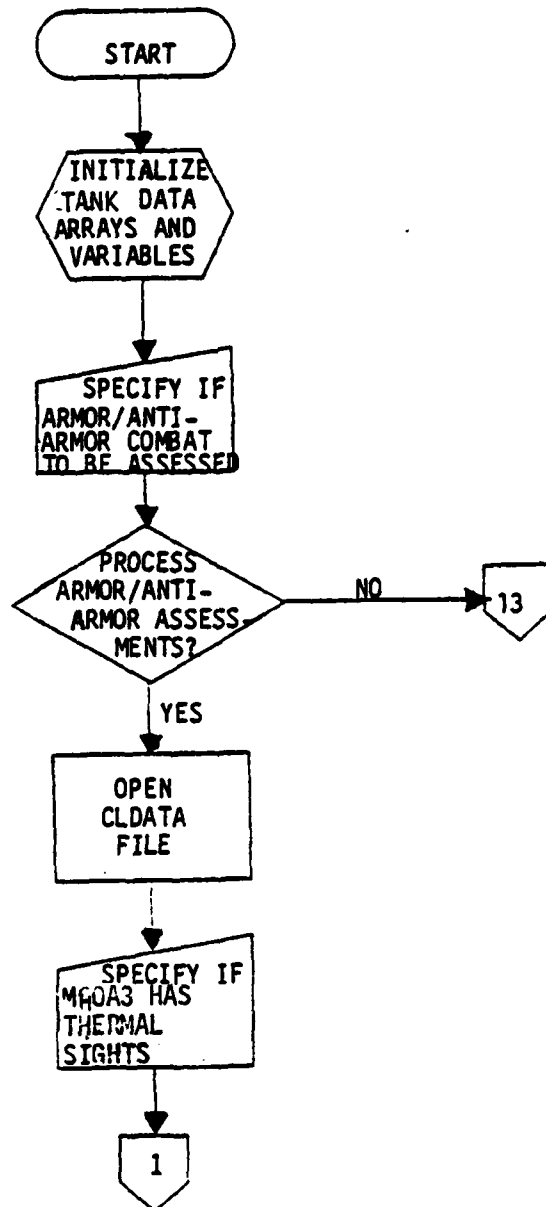


Figure H-7. Tank flow diagram.
(Continued next page)

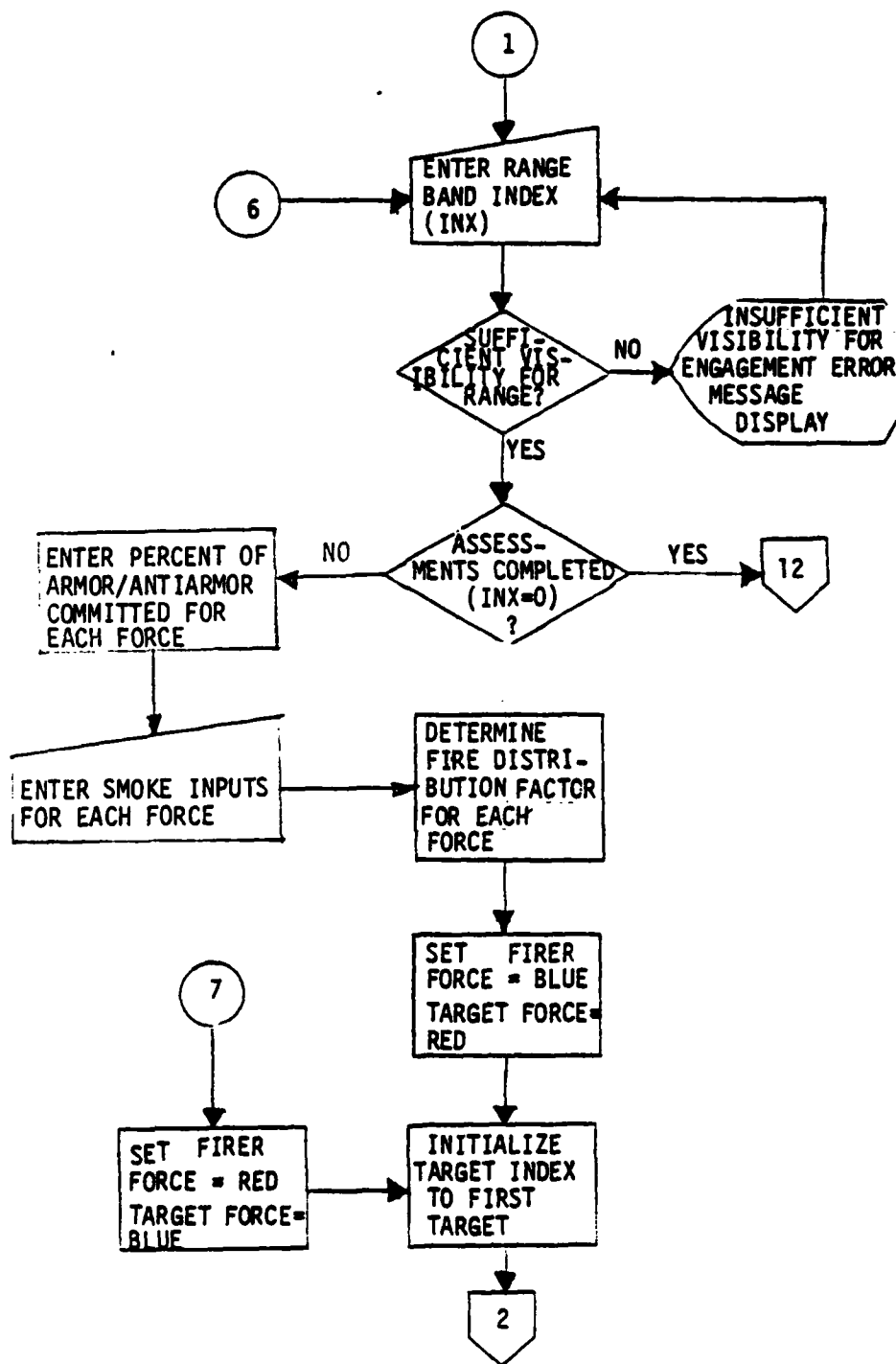


Figure H-7 Tank flow diagram (continued).

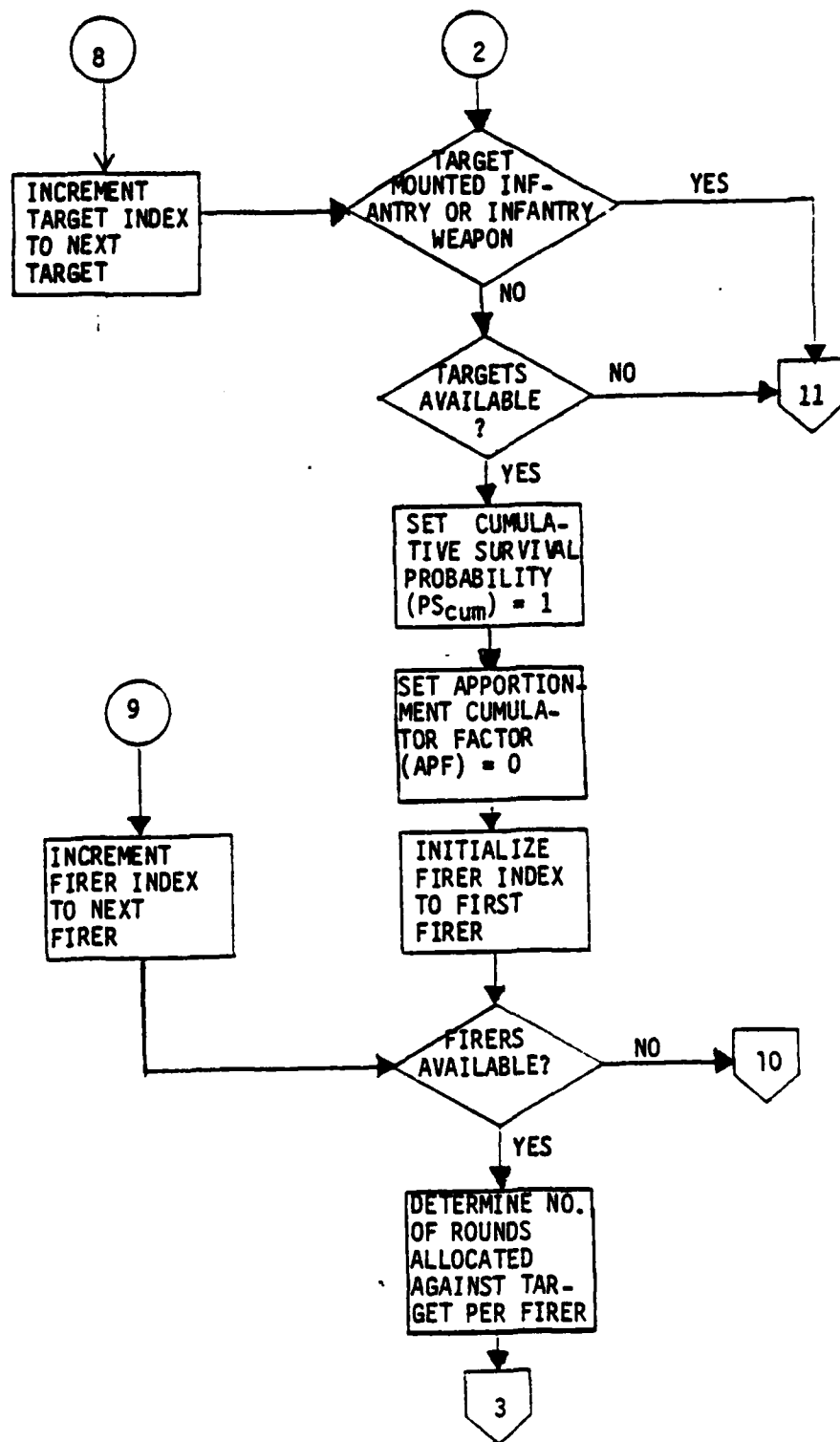


Figure H-7

Tank flow diagram (continued).

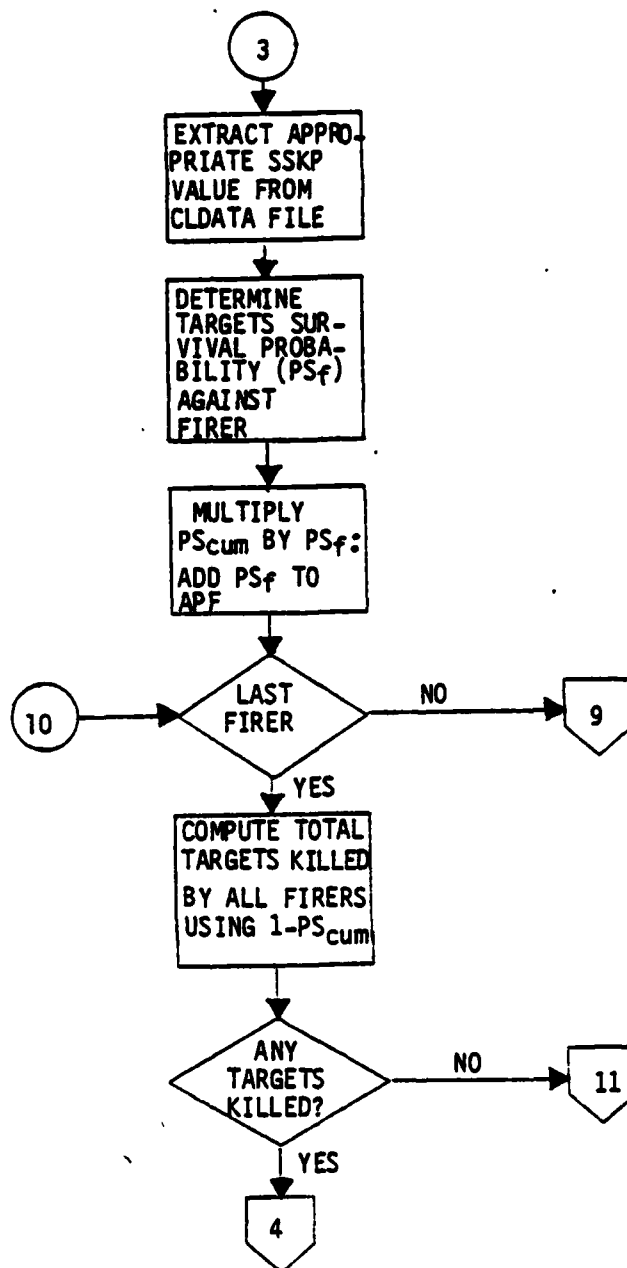


Figure H-7 Tank flow diagram (continued).

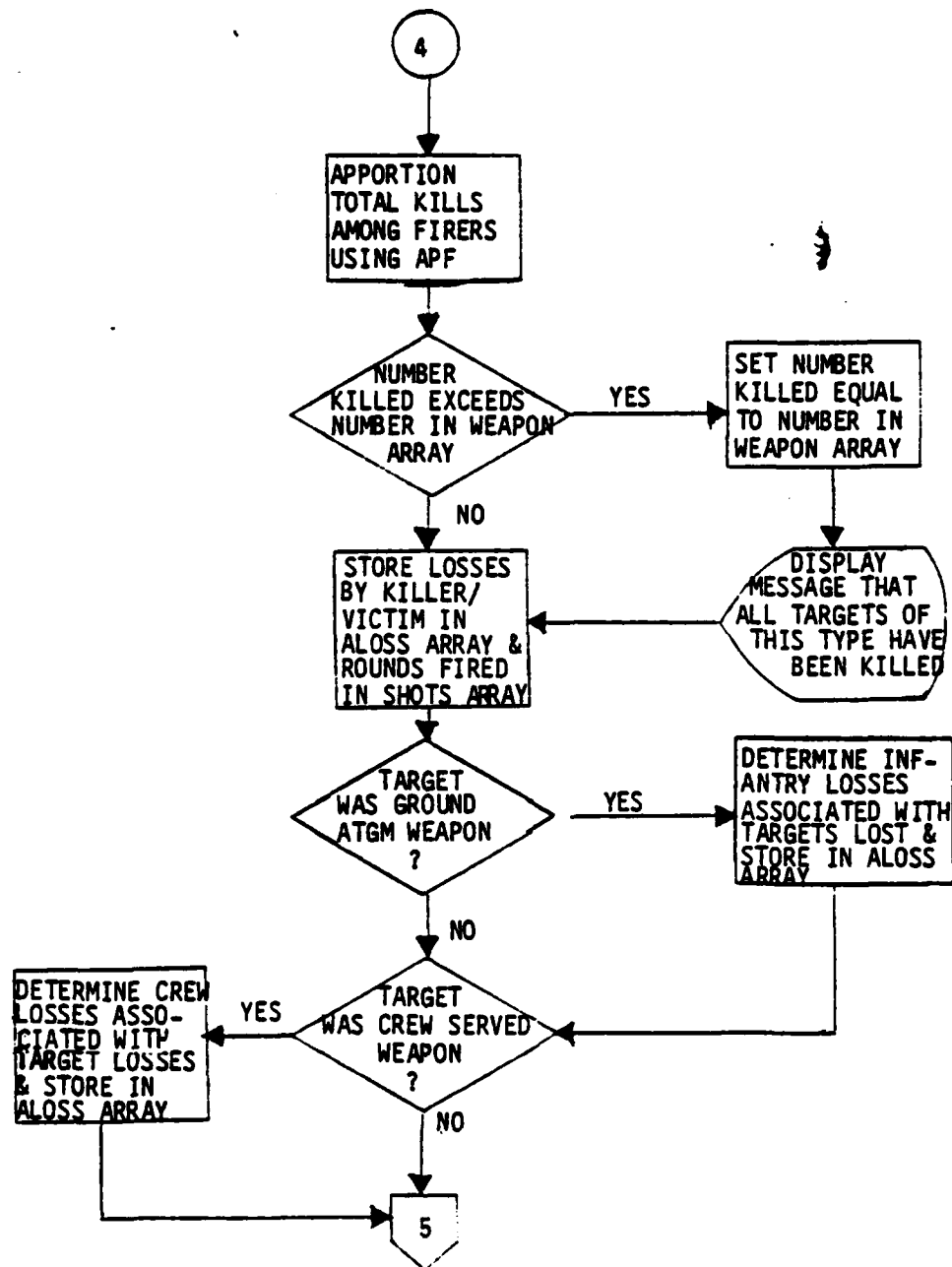


Figure H-7

Tank flow diagram (continued).

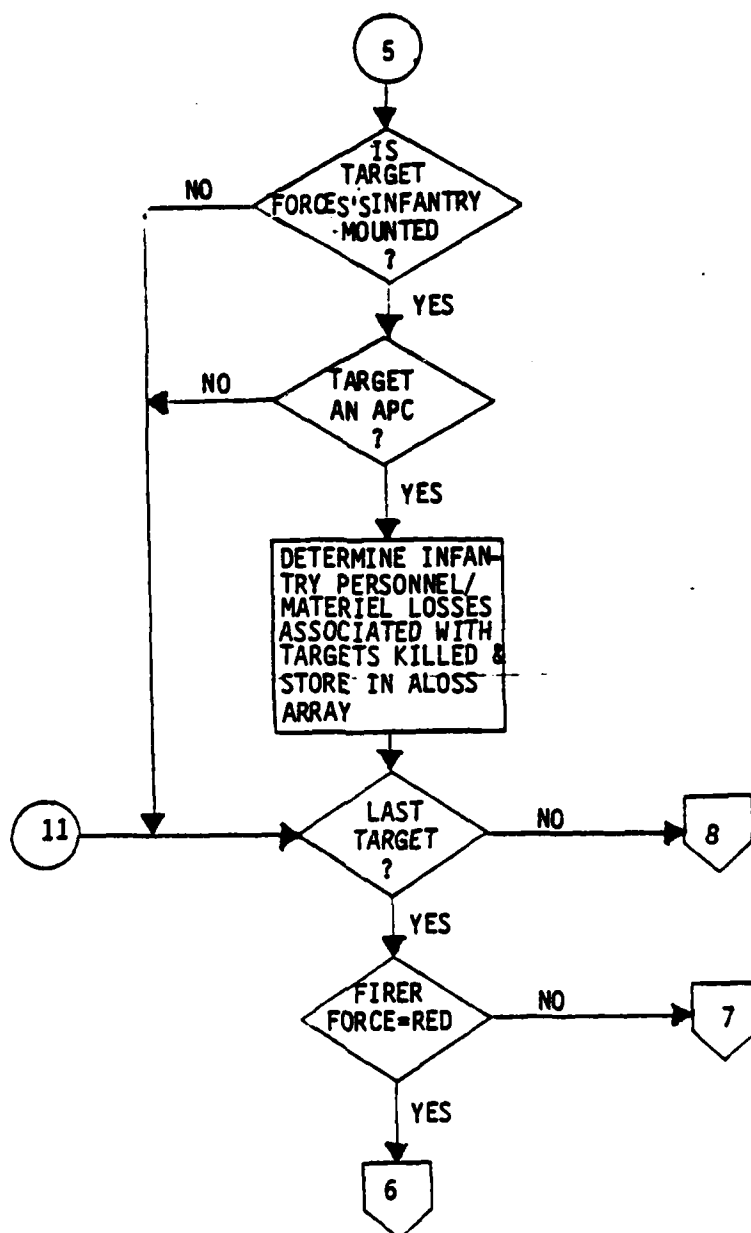


Figure H-7

Tank flow diagram (continued).

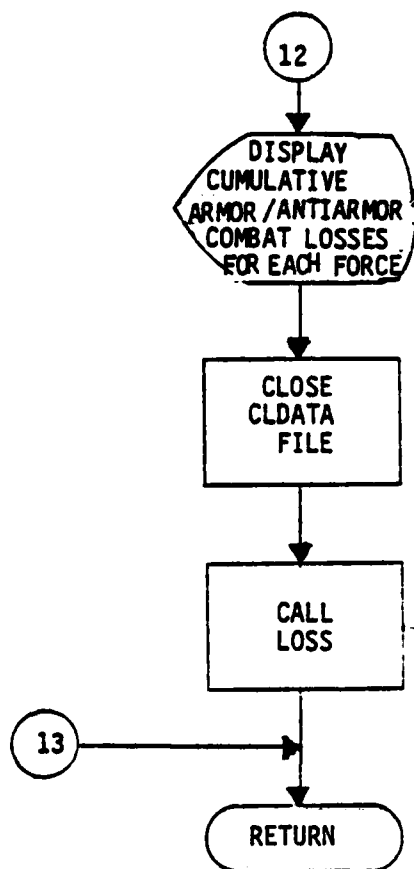


Figure H-7 Tank flow diagram (concluded).

APPENDIX I

RECORD KEEPING SUBROUTINES

Section	I	APPORT
Section	II	BUILD
Section	III	CKACI
Section	IV	CKSTOR
Section	V	CLEAR
Section	VI	COMBINE
Section	VII	DISPLAY
Section	VIII	ENDCI
Section	IX	FORCE
Section	X	LOSS
Section	XI	NEWHIS
Section	XII	RESET
Section	XIII	RESTART
Section	XIV	SHOWCI

APPENDIX I

SECTION I. SUBROUTINE APPORT

I-I-1. SUBROUTINE DESCRIPTION. This subroutine apportions the personnel casualties and weapon system losses determined in the Jiffy III Game combat assessment routines to the units on the FORCE file. The apportionment is based on an algorithm that considers quantity of losses, number of weapon systems in the unit, and the level of combat intensity of the actions in which the unit was involved during the assessment period. The combat intensity level for each unit in a given sector during a critical incident is input interactively by the gamers. The APPORT subroutine also compiles the cumulative combat statistics of all sectors for the entire critical incident. The cumulative loss and ammunition expenditures are kept on the HISTORY file. This subroutine also provides the gamers with the capability to display any specific parent unit or unit after the apportionment process by calling the DISPLAY subroutine.

I-I-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AAHI	History file parameter
AHI	History file parameter
AIRKO	Quantity of given type weapons lost to TACAIR being apportioned to unit
AKO	Quantity of given type weapons lost to ground actions being apportioned to unit
CBTINT	Combat intensity factor
CIL	Combat intensity level factor
CLOST	Number of crew personnel lost
CUMLOS	Parent unit loss array
DAIR	Quantity of weapon systems subject to apportionment for TACAIR losses
FS	Direct fire loss accumulator
HOLD	Array (I+10) for current value of I
I	File record word index; weapon system index
ICIL	Combat intensity level index
IEFF	Variable format array
IFLAG	Logic flag
IFOR	Variable format array
IHEAD	Variable format array
IHOLD	Automatic CIL allocation indicator
II	Weapon system index
INC	History file parameter index
INT	Gamer response variable
INX	Gamer response variable
IPRNT	Variable format array
IPUT	Put call switch

<u>Variable</u>	<u>Description</u>
IUN	Do loop variable
IWD	Weapon system index
IWPN	Weapon system index
J	Force identifier
JFLAG	Do loop index flag
JJ	Force identifier
K	File record word index
KIND	Force color
KNT	States counter
LHOLD	Scratch variable for combat intensity
L1	Do loop lower bond
L2	Do loop upper bond
M	Index-sequential file status variable
MFLAG	Sector and CI flag
NEW	Record existence flag
NPRNTS	Print control flag
NUNIT	Unit counter
NWD	AD losses index
NWPN	States weapon index
PAK	Large pack array
PAREFF	Parent unit effectiveness
PARENT	Parent unit identifier
PARFPS	Total parent fire power score
PARINIT	Initial firepower score of parent unit
PERS	Number of non-infantry personnel casualties
REMAIN	Weapons remaining
RLOST	Weapon remaining
SHRT	Direct fire loss accumulator
TFPS	Total firepower score
UEFF	Unit effectiveness
XL	Packed weapon system losses to all types of combat
XN	Unpacked weapon system losses to all types of combat

NOTE: All common variables are defined in appendix F.

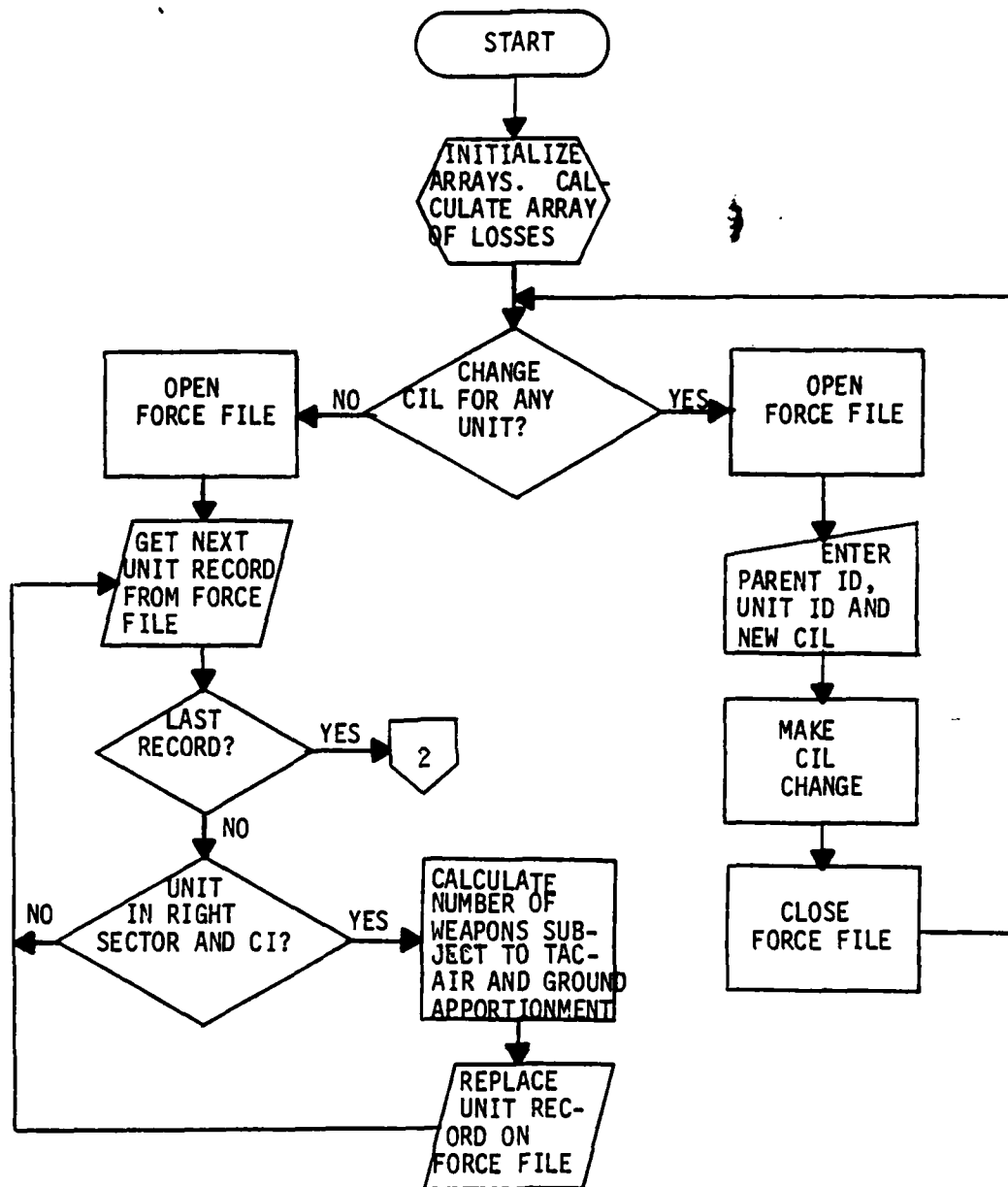


Figure I-1. APPORT flow diagram (continued next page)

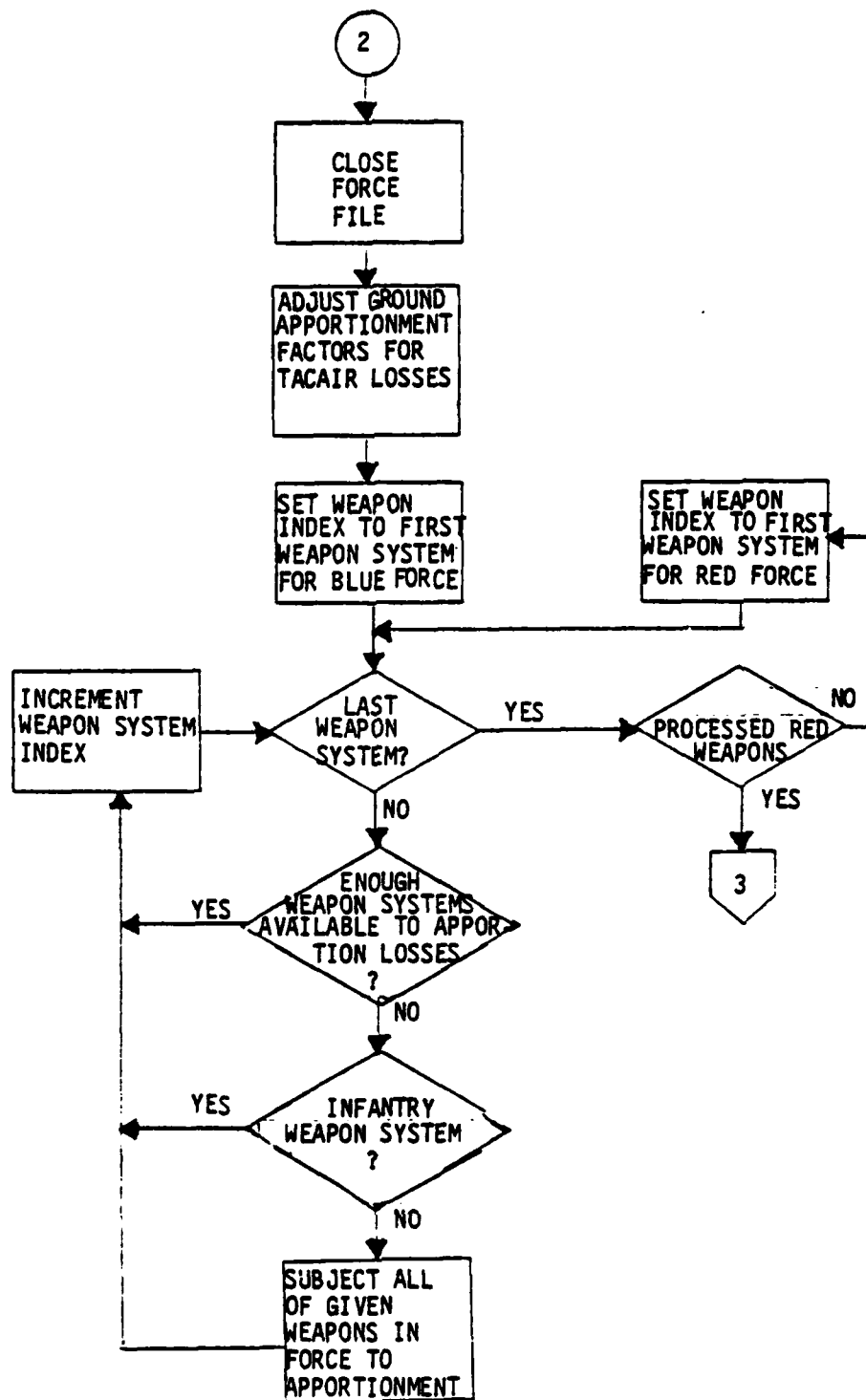


Figure 1-1 APPORT flow diagram (continued).

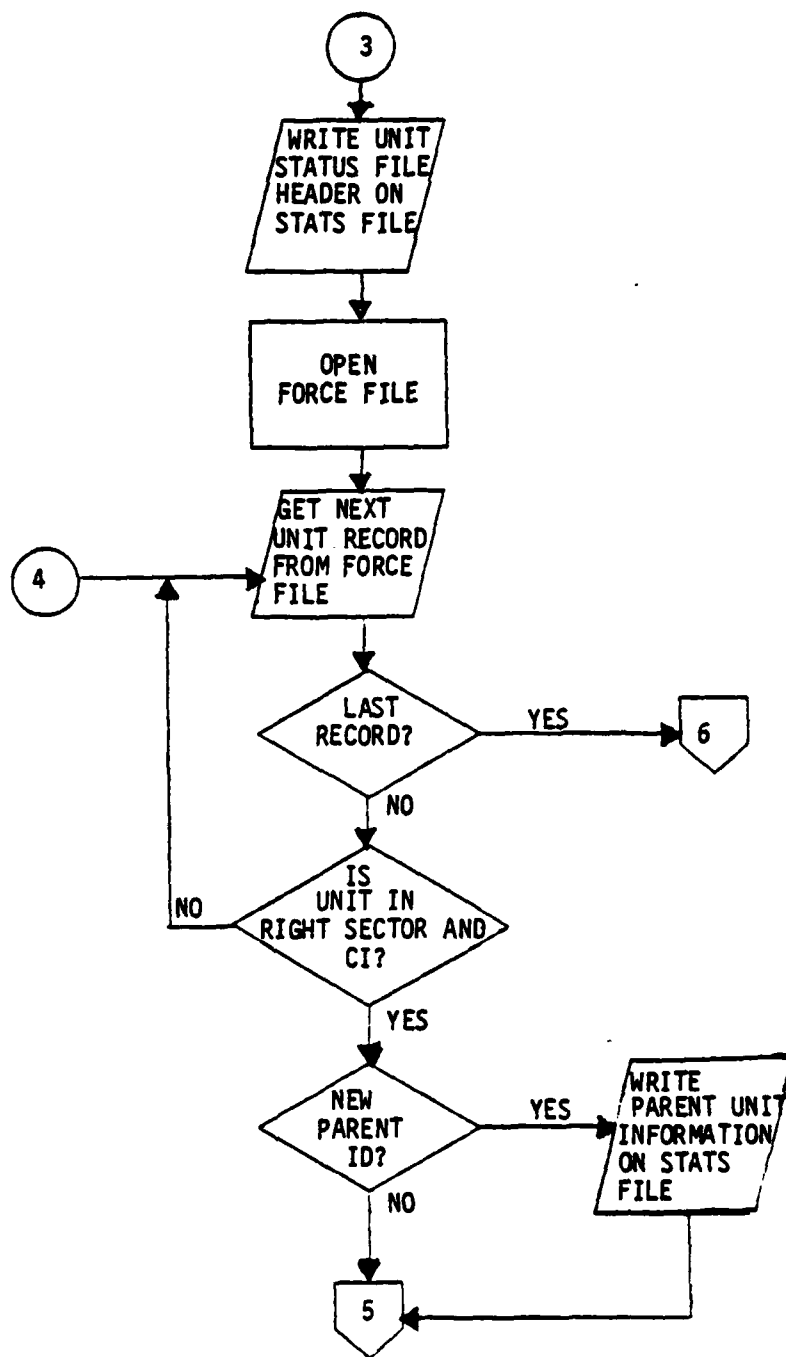


Figure I-1 APPORT flow diagram (continued).

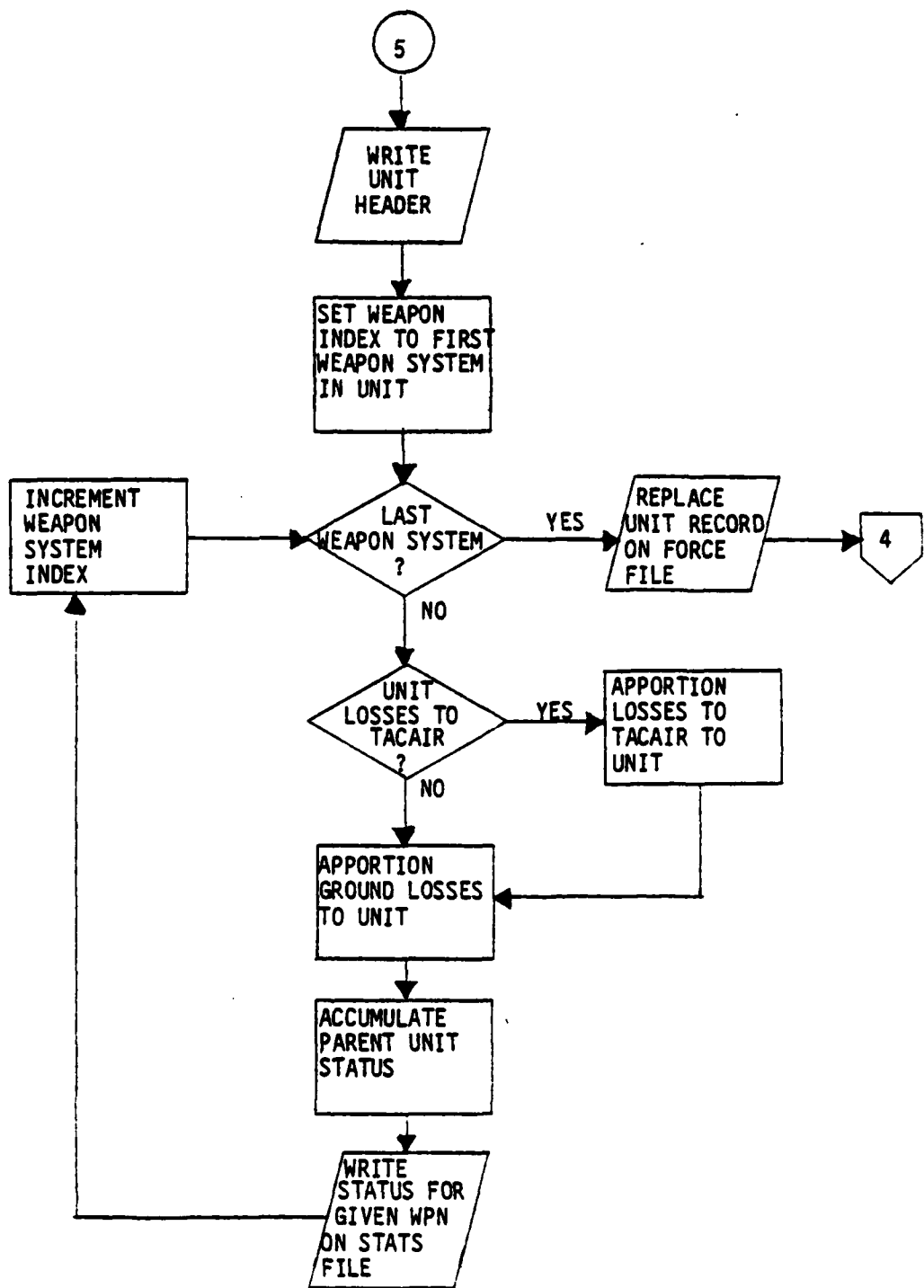


Figure I-1 APPORT flow diagram (continued).

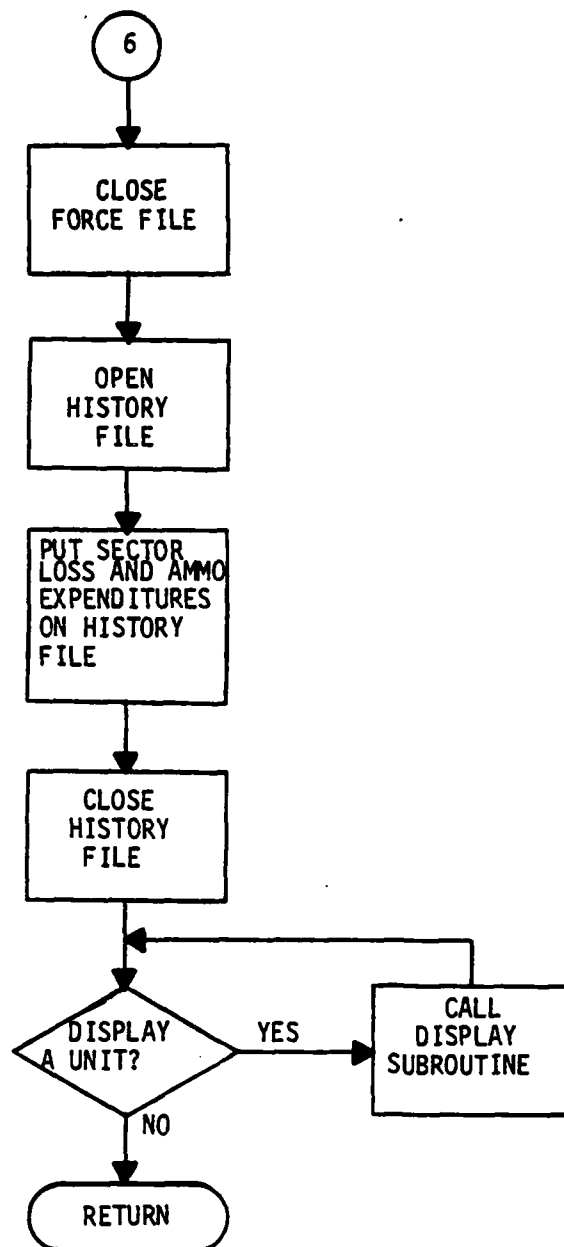


Figure I-1. APPORT flow diagram (concluded).

APPENDIX I

SECTION II. SUBROUTINE BUILD

I-II-1. SUBROUTINE DESCRIPTION. BUILD is a duplicate of the SRC program. A copy of the SRC program was included in the Jiffy Game to provide the gamers the capability to create interactively new units in a force with existing or new SRCs during actual processing of the Jiffy Game. BUILD allows the gamers to develop new SRCs. The program logic flow diagram for BUILD is identical to the flow diagram presented for the SRC program.

I-II-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AHOLD	First word of SRC record
AJ	Quantity of weapons being entered
ASRC	SRC identifier
I	SRC record word index
IID	Weapon system item code
INX	Gamer response variable
M	Index-sequential file status variable
MM	Weapon item code being entered
NN	Weapon item code word index

NOTE: All common variables are defined in appendix F.

APPENDIX I

SECTION III. SUBROUTINE CKACI

I-III-1. SUBROUTINE DESCRIPTION. This subroutine checks that the force file and history file are compatible and that post processor requests are consistent with the CI information on the history file.

I-III-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
ACIO	Old CI
CCI	CI name array
I	Do loop variable
M	Final position for force file
N	Do loop variable
NCHK	Sub-sequence CI flag
NCAST	Last CI flag
NO	Zero word index on CI record
N1	Initial CI index on CI record
N2	Final CI index on CI record
WRD1	Scratch variable for history file fields
WRD2	Scratch variable for history file fields
XN	Flag for history file field

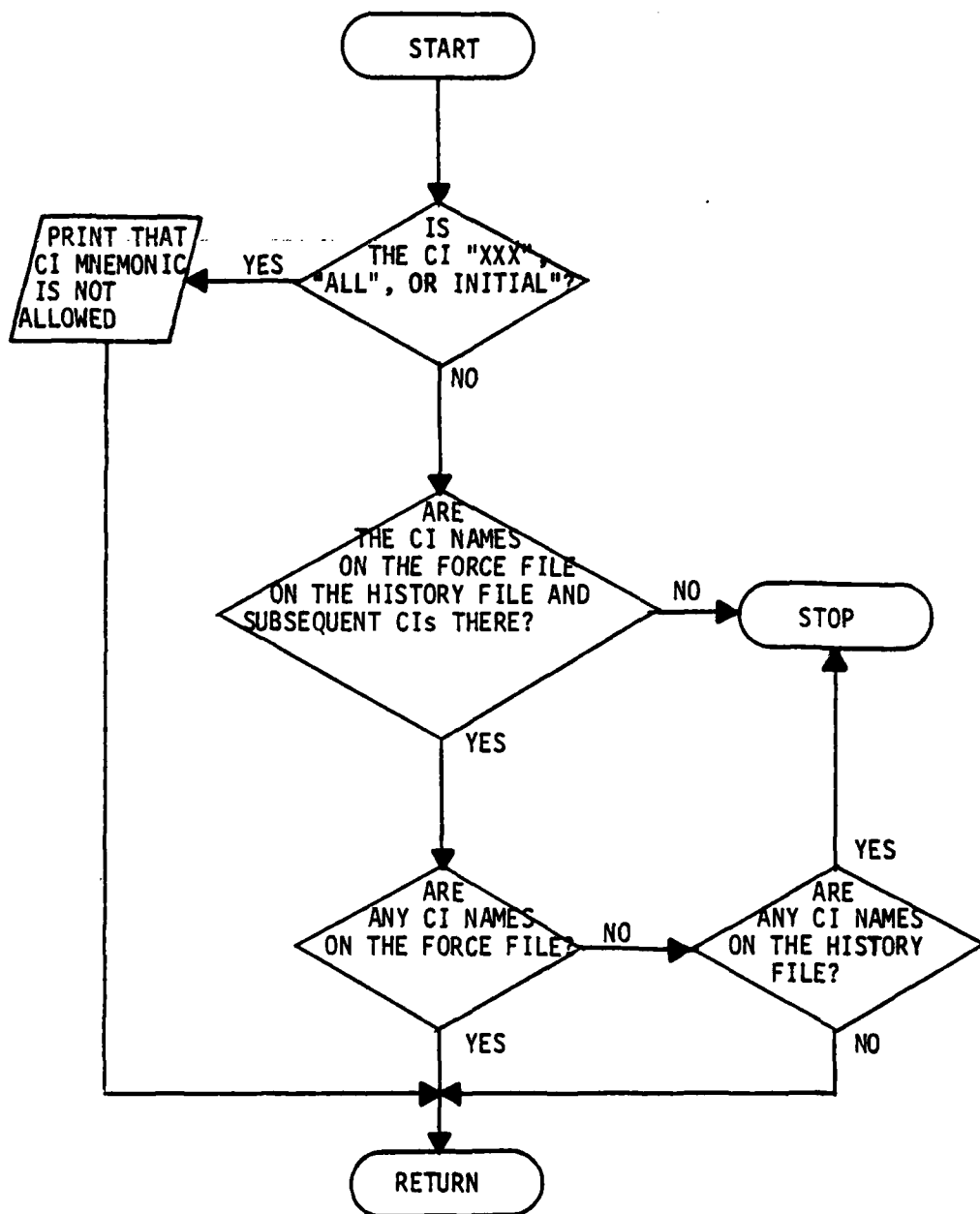


Figure I-3. CKACI flow diagram.

APPENDIX I

SECTION IV. SUBROUTINE CKSTOR

I-IV-1. SUBROUTINE DESCRIPTION. This subroutine checks to determine if a record should be a replacement for an old record on the HISTORY file or be inputted as an entirely new HISTORY file record.

I-IV-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AH1	Stats record type
I	Do loop variable
IPUT	History file record flag
N1	Beginning index for stats record initialization
N2	Final index for stats record initialization

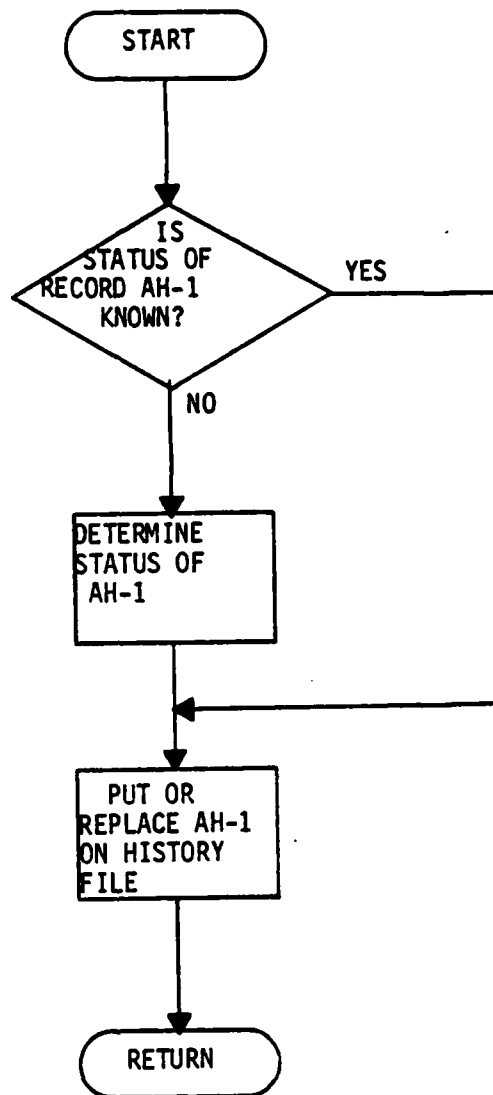


Figure I-4. CKSTOR flow diagram.

APPENDIX I

SECTION V. SUBROUTINE CLEAR

I-V-1. SUBROUTINE DESCRIPTION. This subroutine clears the HISTORY file records of a sector that is about to be regamed.

I-V-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
I	Do loop index
II	Do loop index
M	Final position of history file
NN	Stats record key length
RECS	Array of stats record names

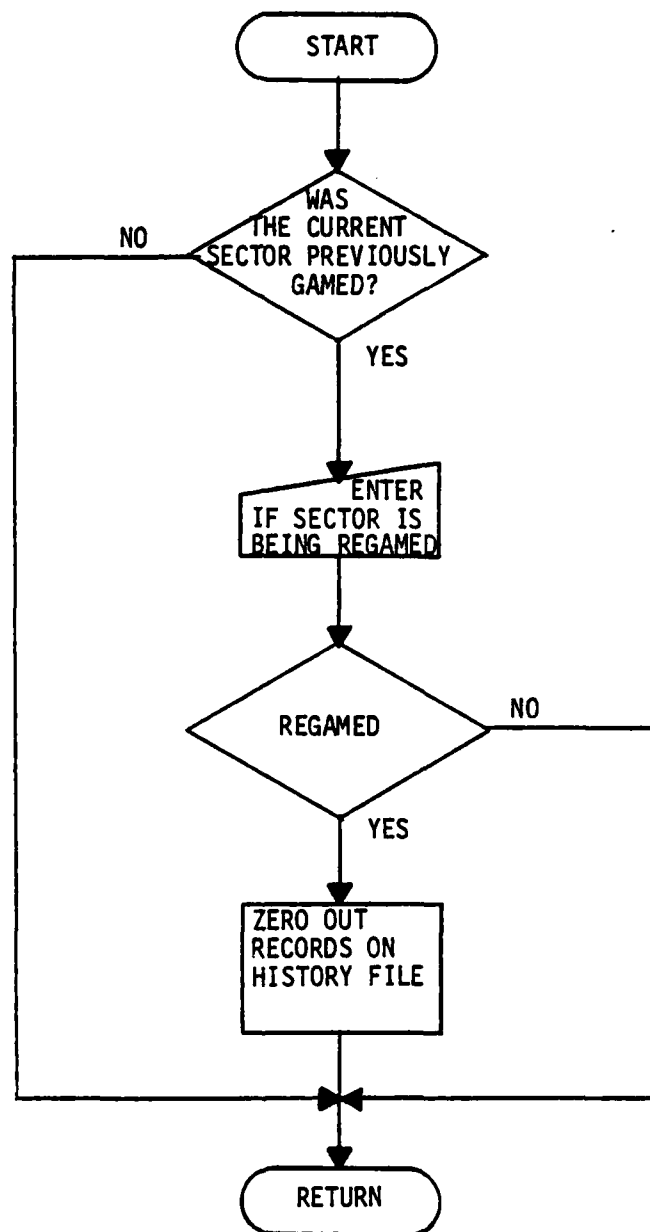


Figure I-5. CLEAR flow diagram.

APPENDIX I

SECTION VI. SUBROUTINE COMBINE

I-VI-1. SUBROUTINE DESCRIPTION. This subroutine allows the user to automatically remove a unit from the forcefile and causes its assets to be added to another unit. It is used by the gamer to disband a unit that has suffered heavy losses.

I-VI-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
DPAR	Parent of disbanding unit
DUNIT	Unit ID of disbanding unit
HPAR	Parent of host unit
HUNIT	Unit ID of host unit
I	Weapon system index, logic flag
W	Working array for Unit assets

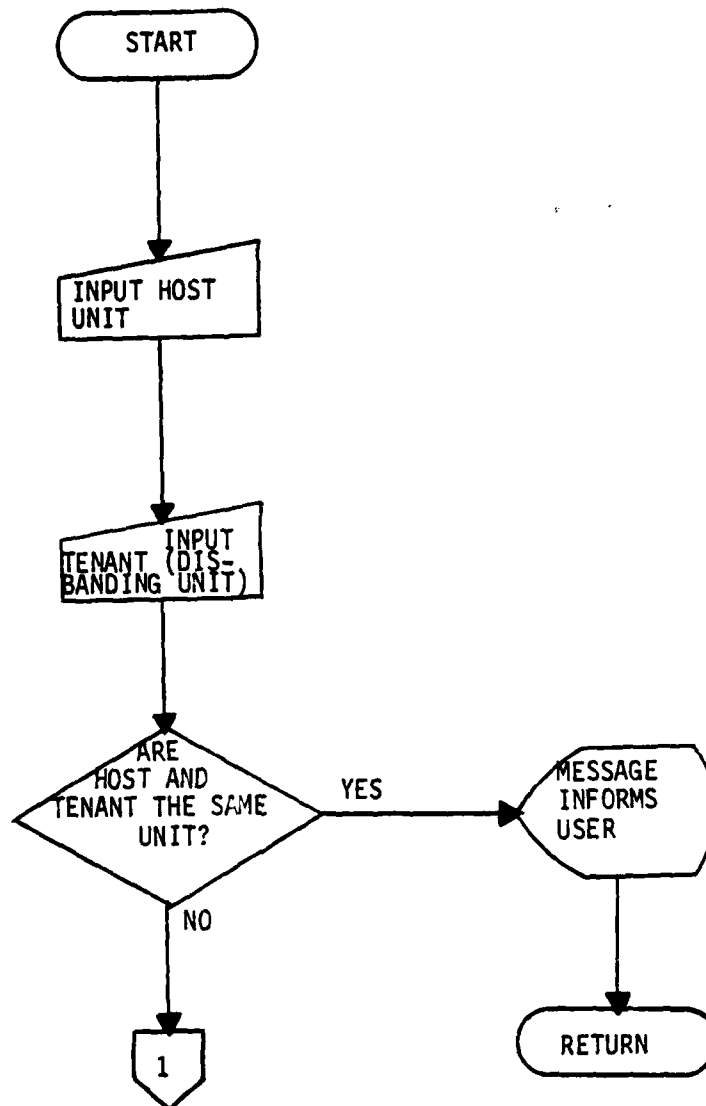


Figure I-6. COMBINE flow diagram (Continued next page).

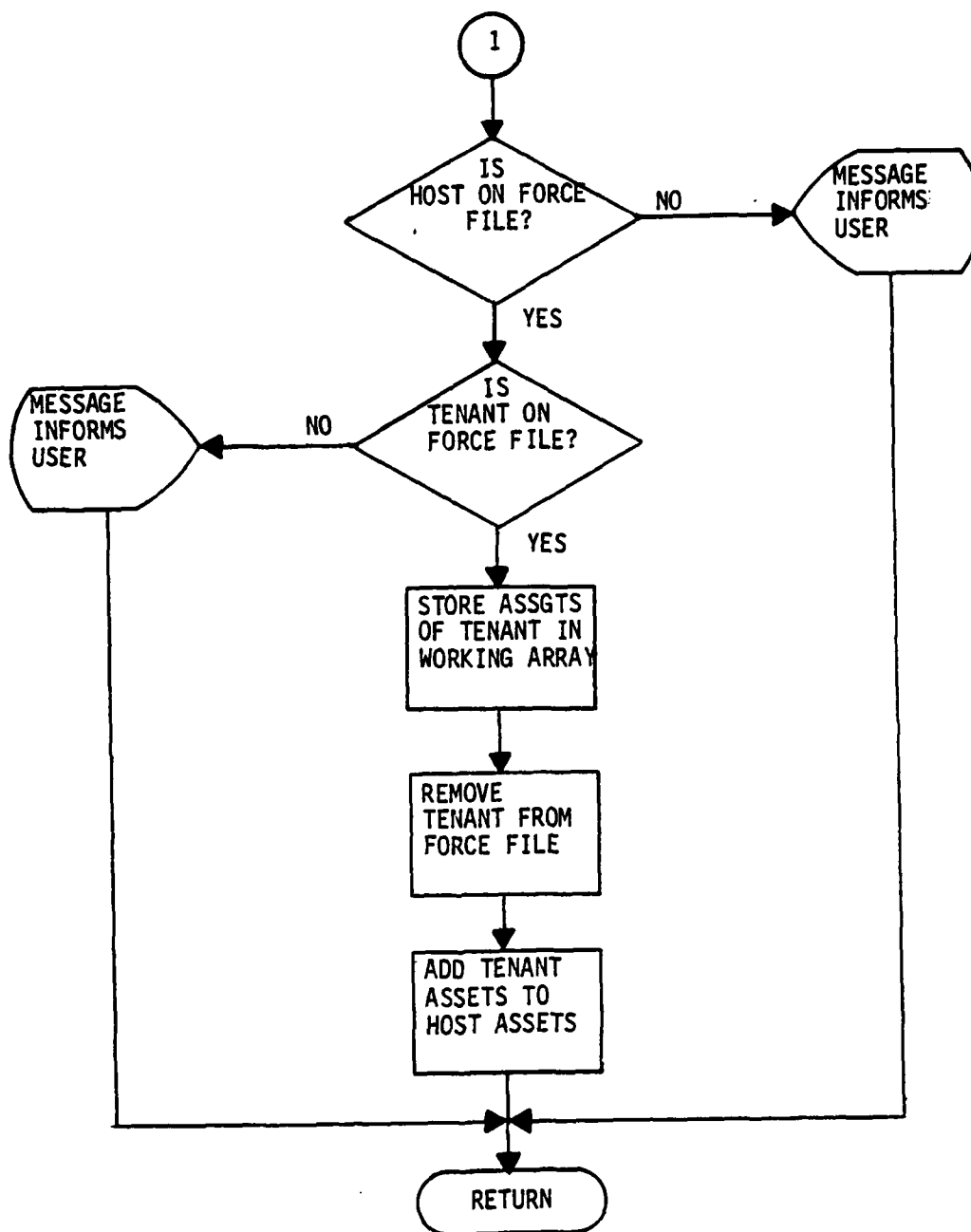


Figure I-6. COMBINE flow diagram (concluded).

APPENDIX I

SECTION VII. SUBROUTINE DISPLAY

I-VII-1. SUBROUTINE DESCRIPTION. This subroutine is called during gaming to display the status of specified units and parent units. The gamer has the option to display a particular unit or all units within a specified parent unit. The unit status parameters displayed include the unit effectiveness of the parent and subordinate unit(s) and the quantity and type of weapon systems remaining in each unit.

I-VII-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AR	Number of weapon systems
CIL	Factor for combat intensity level
I	Unit record word index
ICODE	Weapon system item code
IFLAG	Print flag
INC	Increment counter
INX	Gamer response variable
J	Force identifier
M	File status variable
PARENT	Name of parent unit
REMAIN	Number of particular weapon systems remaining in unit
TFPS	Total firepower score
UEFF	Unit effectiveness
UNIT	Name of unit
XLOST	Number of particular weapon system losses

I-VII-3. TYPES OF DISPLAYS.

<u>Display Index</u>	<u>Type Display</u>	<u>Information Displayed</u>
1	Lists all parent units on FORCE file	Parent ID, force designator, Parent unit effectiveness, sector and critical incident
2	Lists all parent units in defined sector and critical incident	Parent ID, force designator and Parent unit effectiveness

<u>Display Index</u>	<u>Type Display</u>	<u>Information Displayed</u>
3	Lists all units attached to a specific parent unit	Parent ID, Unit ID, force designator, unit effectiveness, sector and critical incident
4	Lists all weapon systems in a specific unit or unit	Parent ID, Parent unit effectiveness, Unit ID, unit effectiveness, quantity and type of weapon systems.

NOTE: All command variables are defined in appendix F.

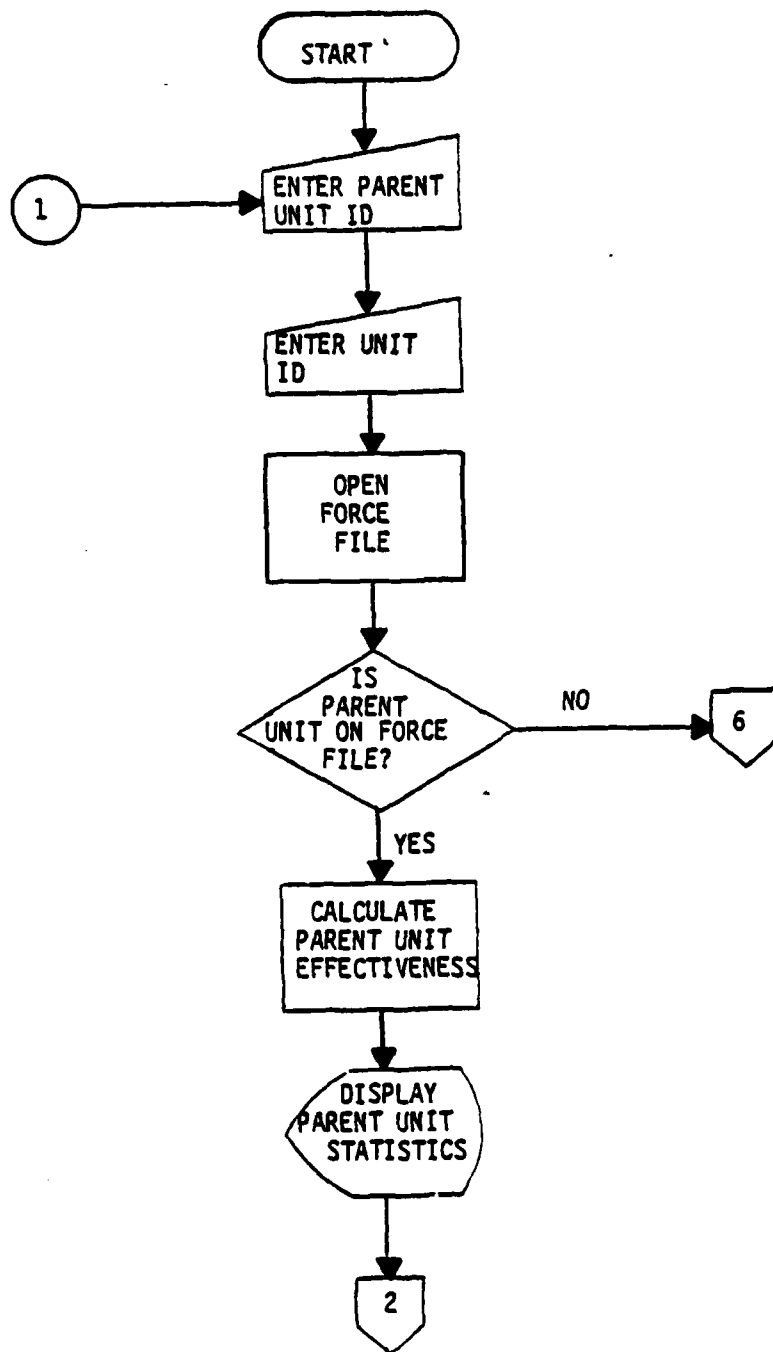


Figure I-7 DISPLAY flow diagram.
(Continued next page)

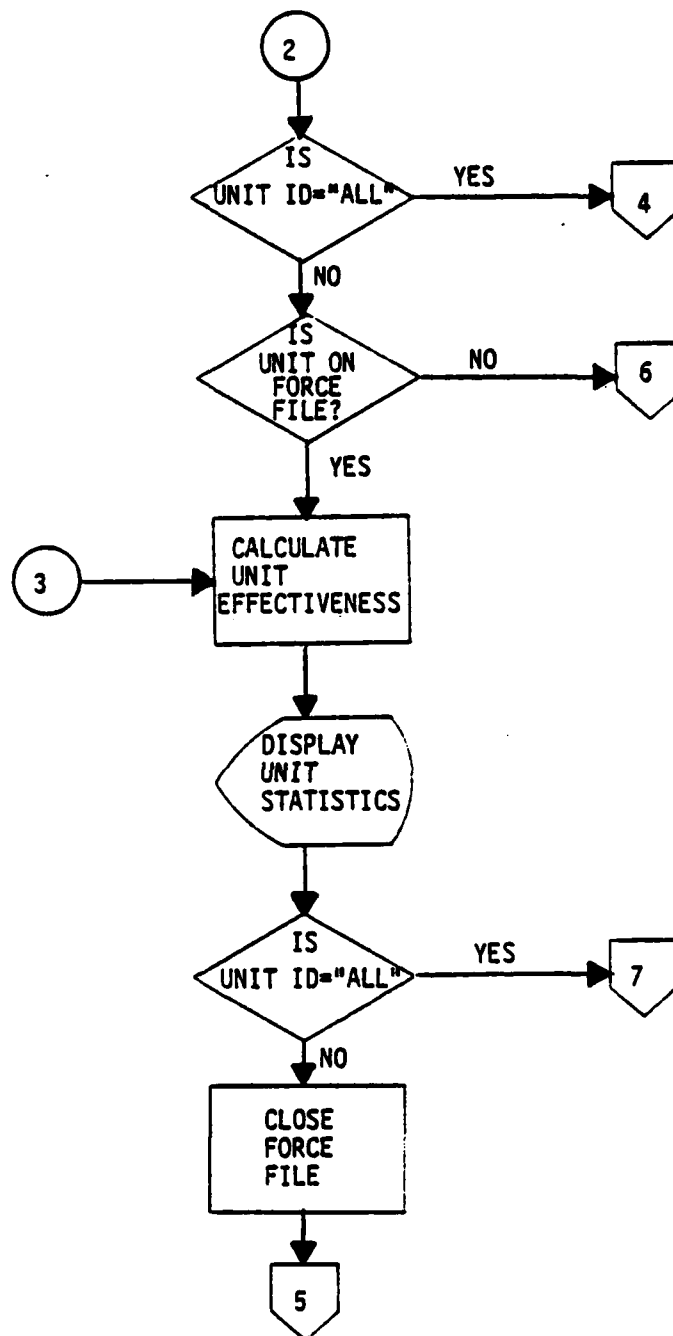


Figure I-7 DISPLAY flow diagram (continued).

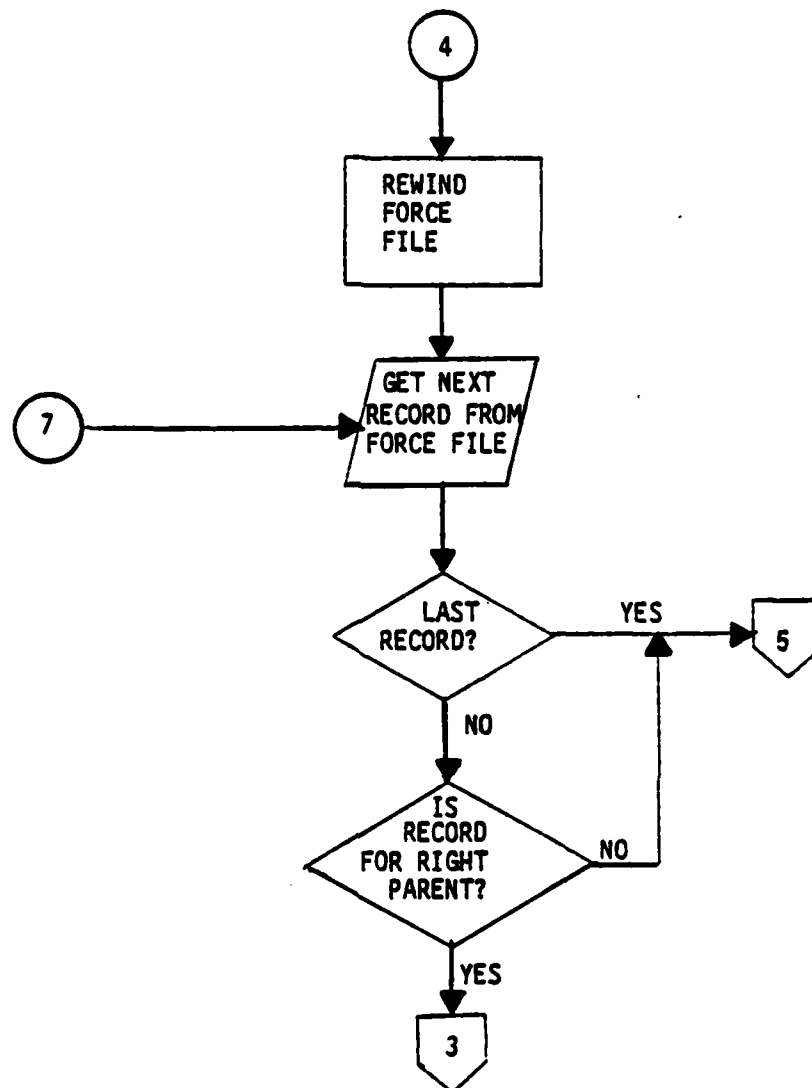


Figure I-7 DISPLAY flow diagram (continued).

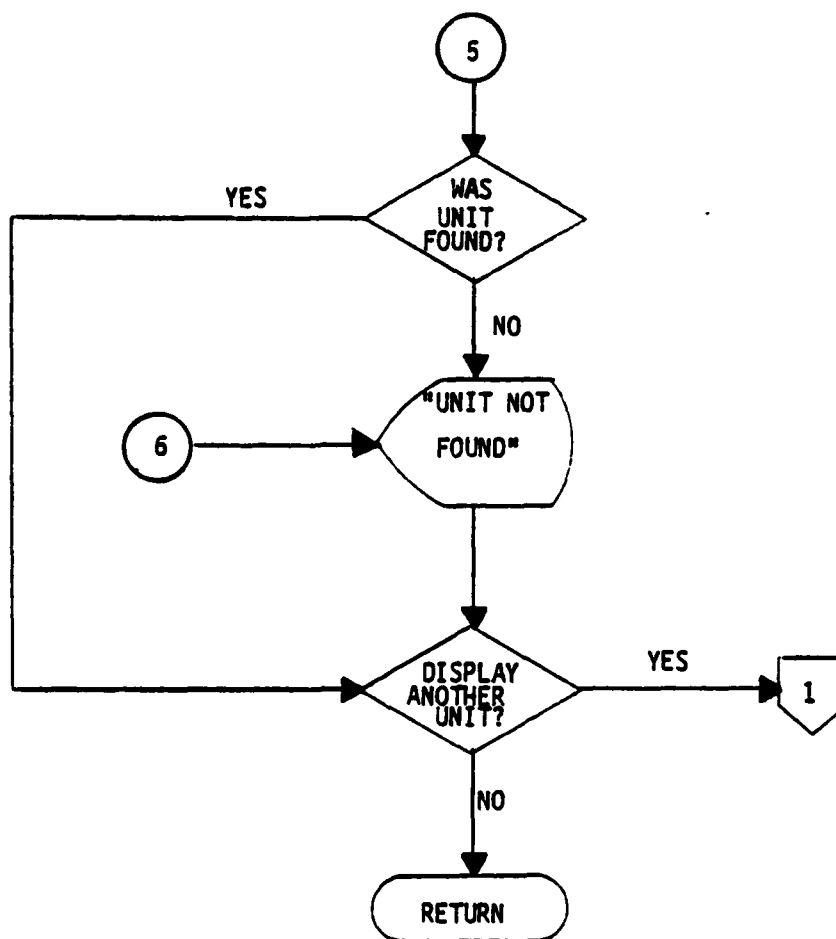


Figure I-Z DISPLAY flow: diagram (concluded).

APPENDIX I

SECTION VIII. SUBROUTINE ENDCI

I-VIII-1. SUBROUTINE DESCRIPTION. This subroutine performs end of CI booking on the history file and force file.

I-VIII-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AH1	Stats record type
I	Do loop variable
IPUT	CKSTOR input parameter
M	Final positions of force file

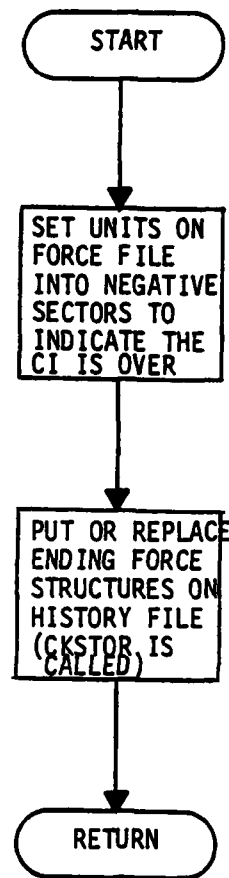


Figure I-8. ENDCI flow diagram.

APPENDIX I

SECTION IX. SUBROUTINE FORCE

I-IX-1. SUBROUTINE DESCRIPTION. FORCE is the subroutine by which the gamers manipulate their forces in the Jiffy Game. FORCE is reached by a gamer response of "1" at the DECISION POINT in SUPER (see table 1). After the gamer defines the critical incident and sector, he is presented his choice of the nine force manipulation options. Upon completion of all but OPTION 0, the game is returned to the OPTION point. A response of "0" loads the weapon systems of all units loaded into the defined sector and critical incident into the weapon system (ELMT) array for both forces and returns control to SUPER. The display option (6) provides the gamer the capability to examine the FORCE file in four ways. The four types of displays accessible at OPTION 6 are given in I-VII-3. Subroutine DISPLAY is used to display all four types.

I-IX-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AH2	History file parameter
AJ	Quantity of weapons to adjust
AVN	Parent variable
CIL	Combat intensity level factors
CV	Relative effectiveness
I	Do loop index
ICIL	Combat intensity level
IEFF	Unit effectiveness
IFLAG	Logic flag
IJ	SRC record weapon index
INC	Unit loaded counter
INX	Gamer response variable
IPRO	EW type
IPUT	Logic flag
J	Force designator
JJ	FORCE record weapon index
JREGM	Run type indicator
KIND	Force color
M	Indexed-sequential file status variable
MM	Type of weapons to adjust
NV	Storage position indicator
PAR	Parent unit ID
PARENT	Parent unit ID
TFPS	Total firepower score
UEFF	Unit effectiveness
UNIT	Unit ID
XCI	Critical incident name
XSECT	Sector number

I-IX-3. FORCE MANIPULATION OPTIONS.

<u>Response Code</u>	<u>Option Description</u>
0	Proceed with assessments
1	Load units into sector
2	Remove units from sector
3	Create a new unit
4	Adjust weapons in a unit
5	Attach a unit to a new parent
6	Display a unit
7	Delete a unit from FORCE file
8	To combine units

NOTE: All common variables are defined in appendix F.

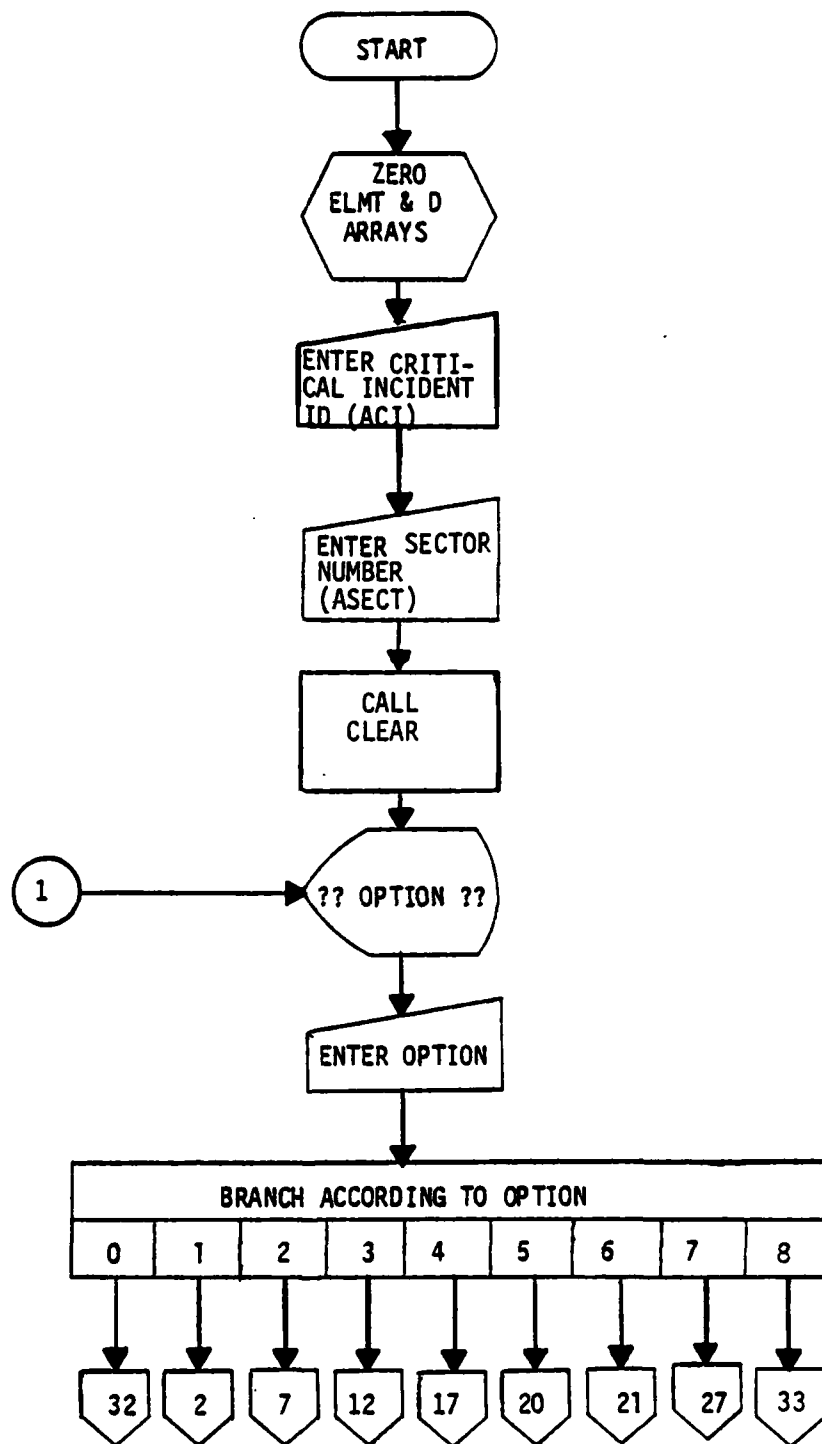


Figure I-9 FORCE flow diagram.
(Continued next page)

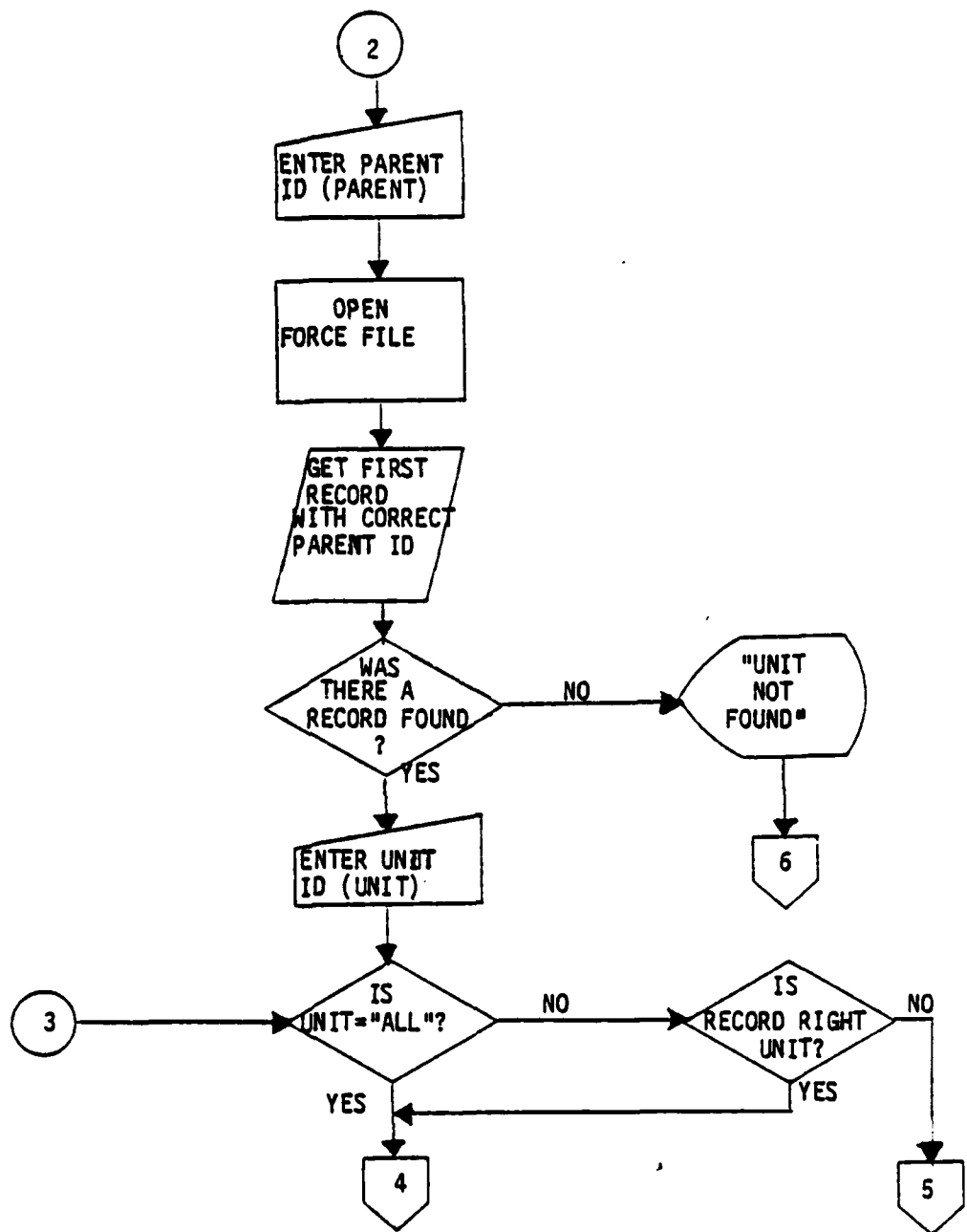


Figure I-9 FORCE flow diagram (continued).

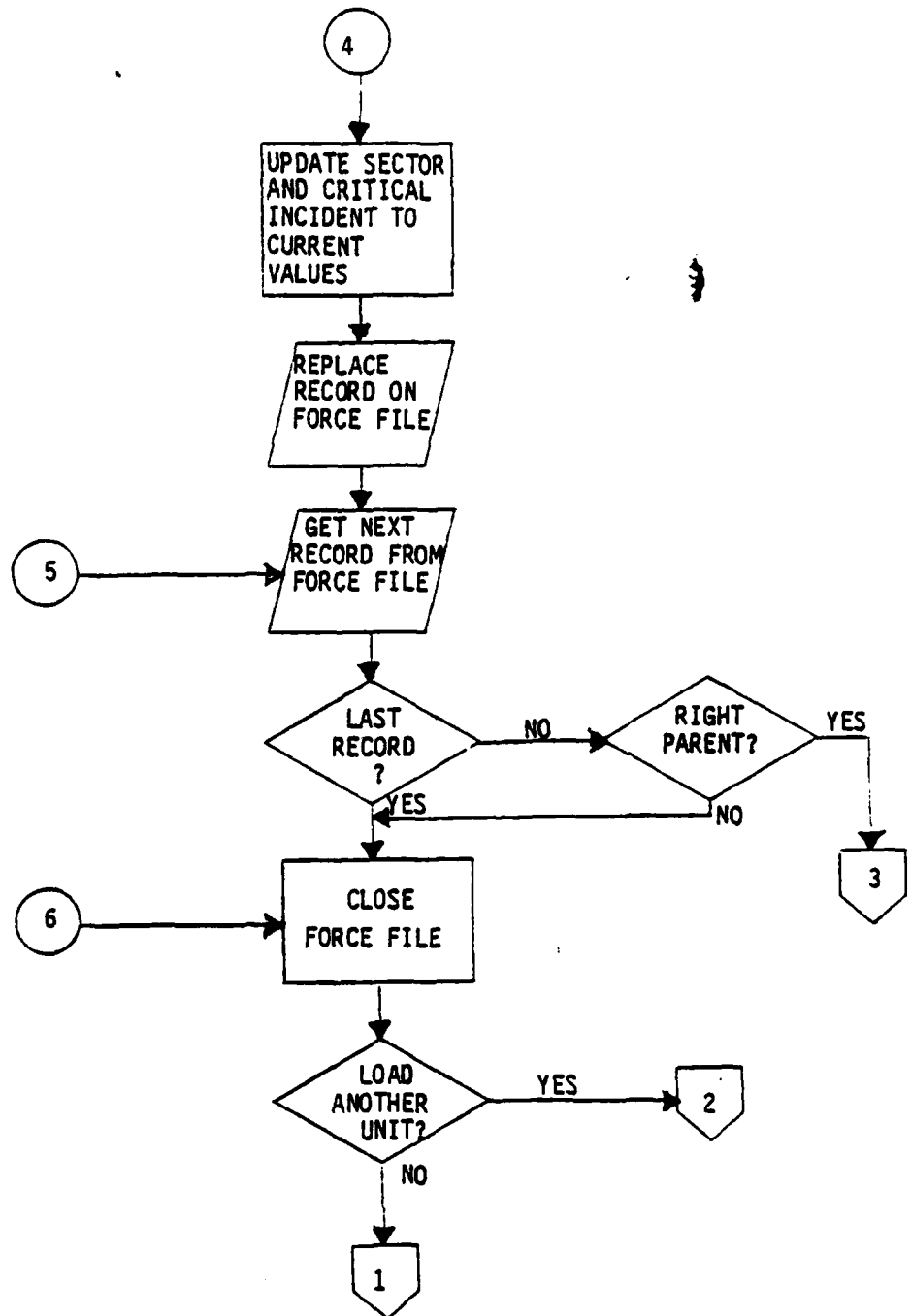


Figure I-9 FORCE flow diagram (continued).

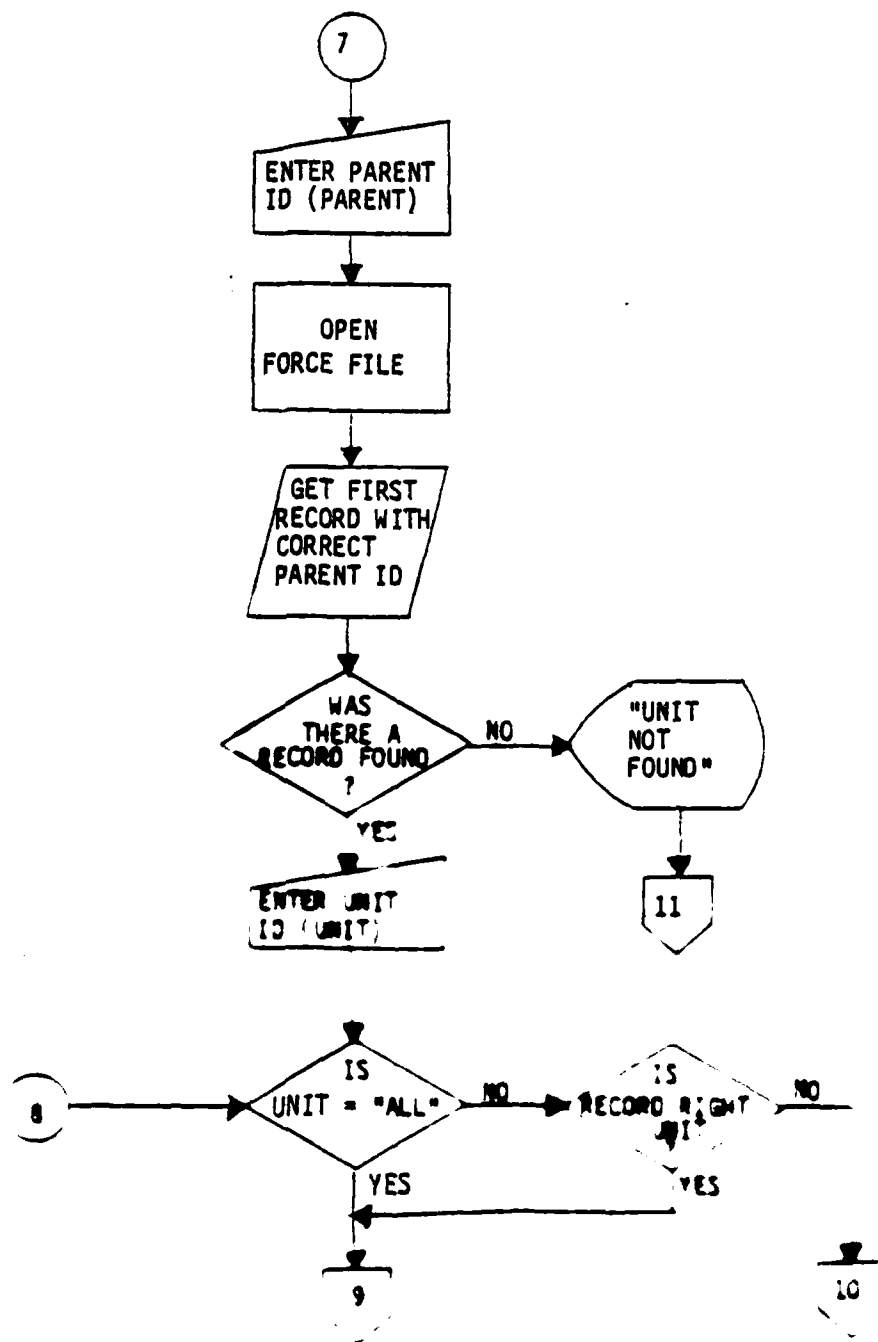


Figure I-9 FORCE flow diagram (continued).

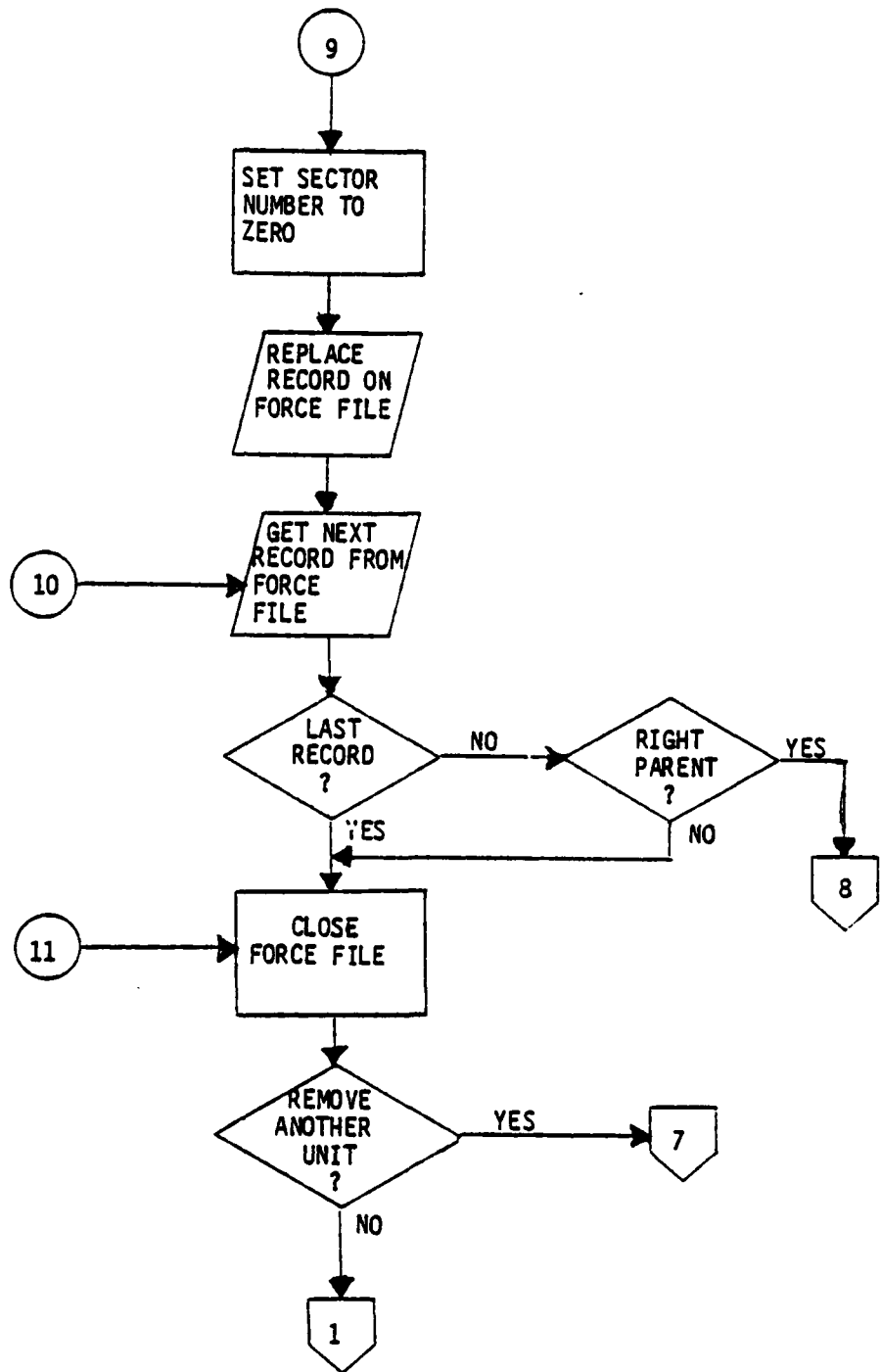


Figure I-9 FORCE flow diagram (continued).

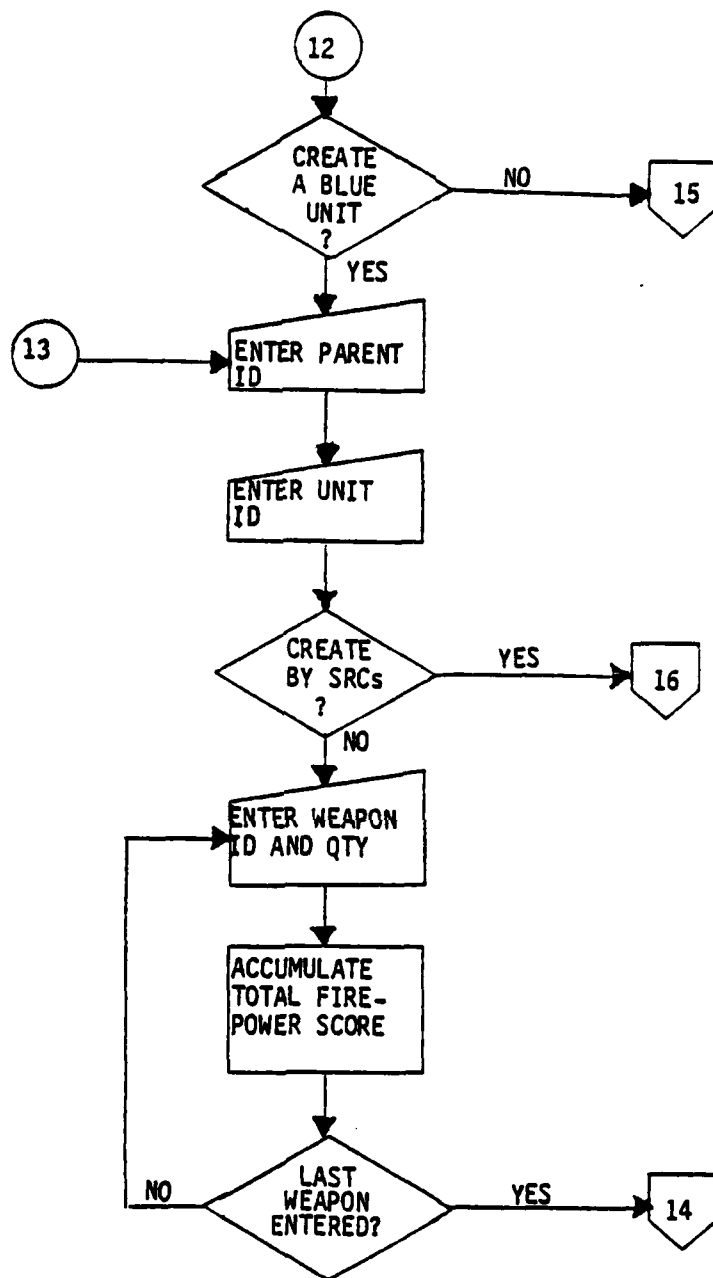


Figure I-9 FORCE flow diagram (continued).

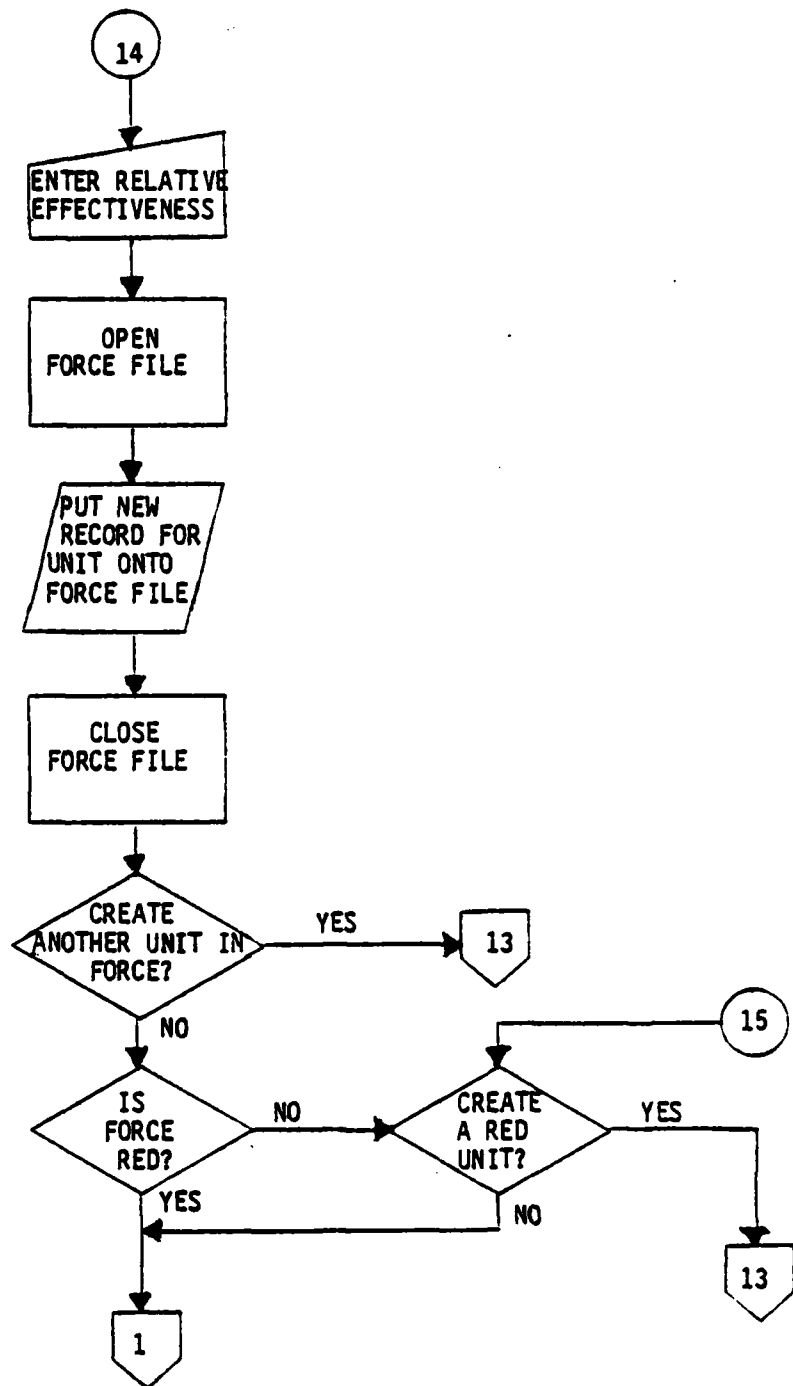


Figure I-9 FORCE flow diagram (continued).

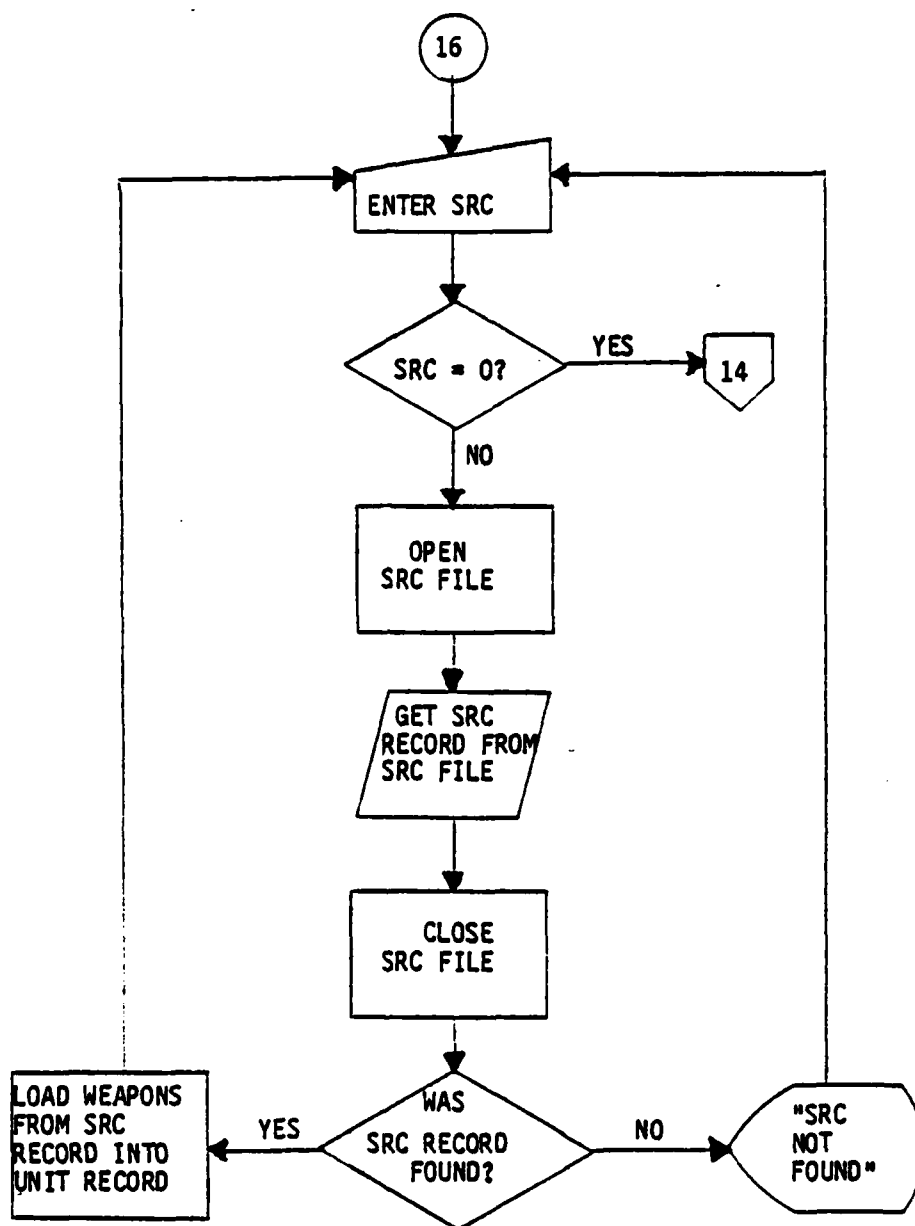


Figure I-9 FORCE flow diagram (continued).

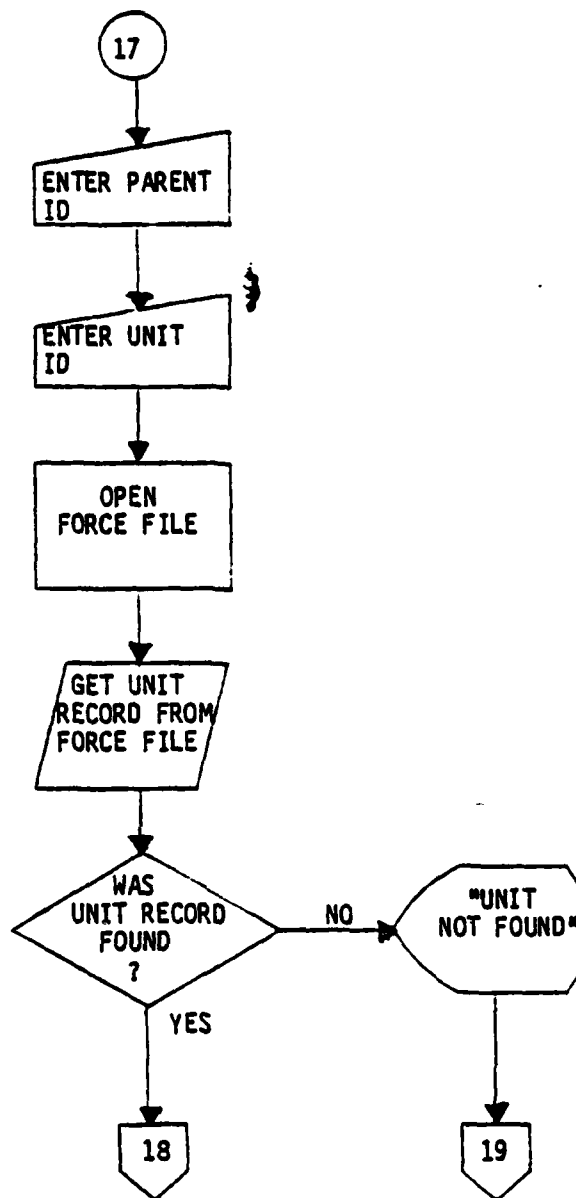


Figure I-9 FORCE flow diagram (continued).

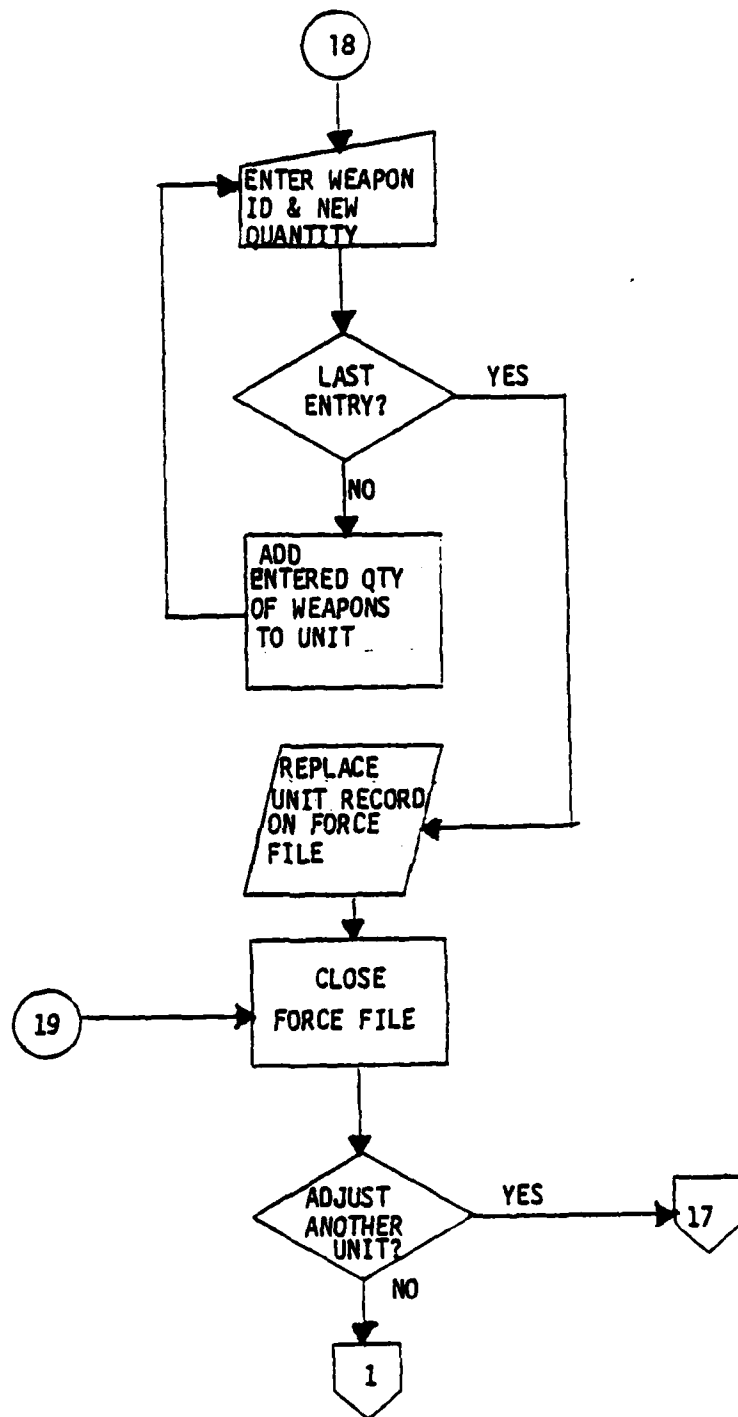


Figure I-9 FORCE flow diagram (continued).

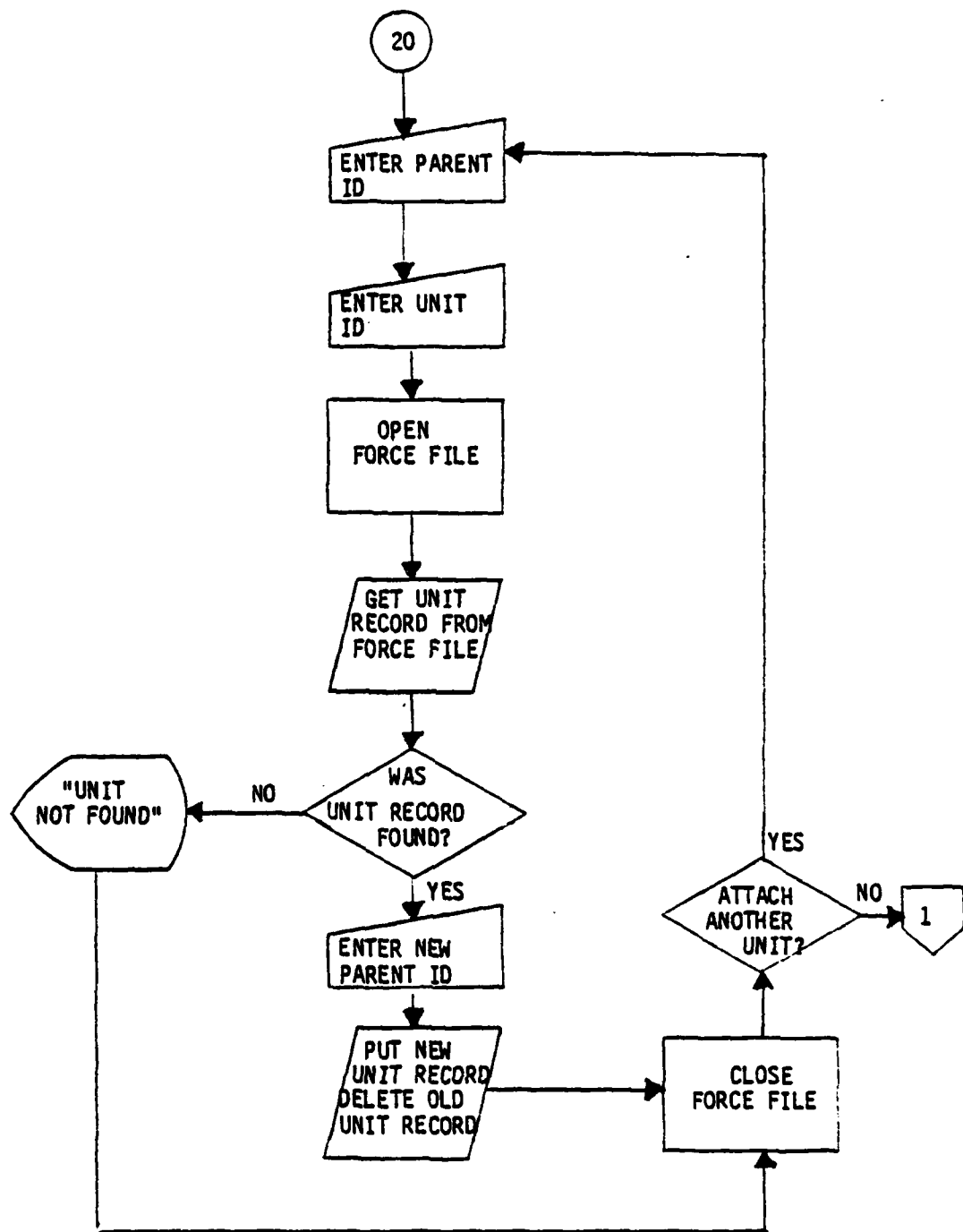


Figure I-9 FORCE flow diagram (continued).

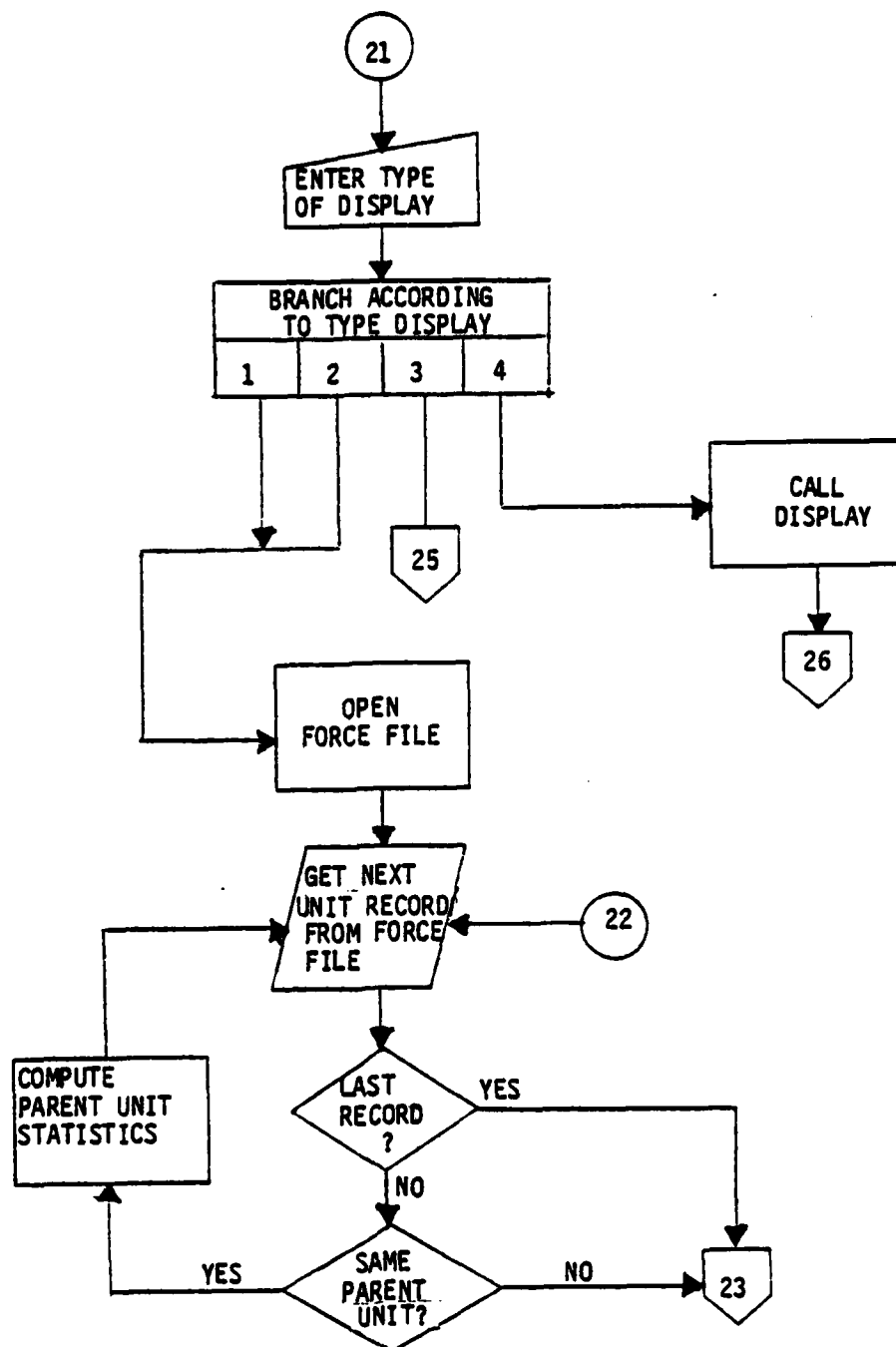


Figure I-9 FORCE flow diagram (continued).

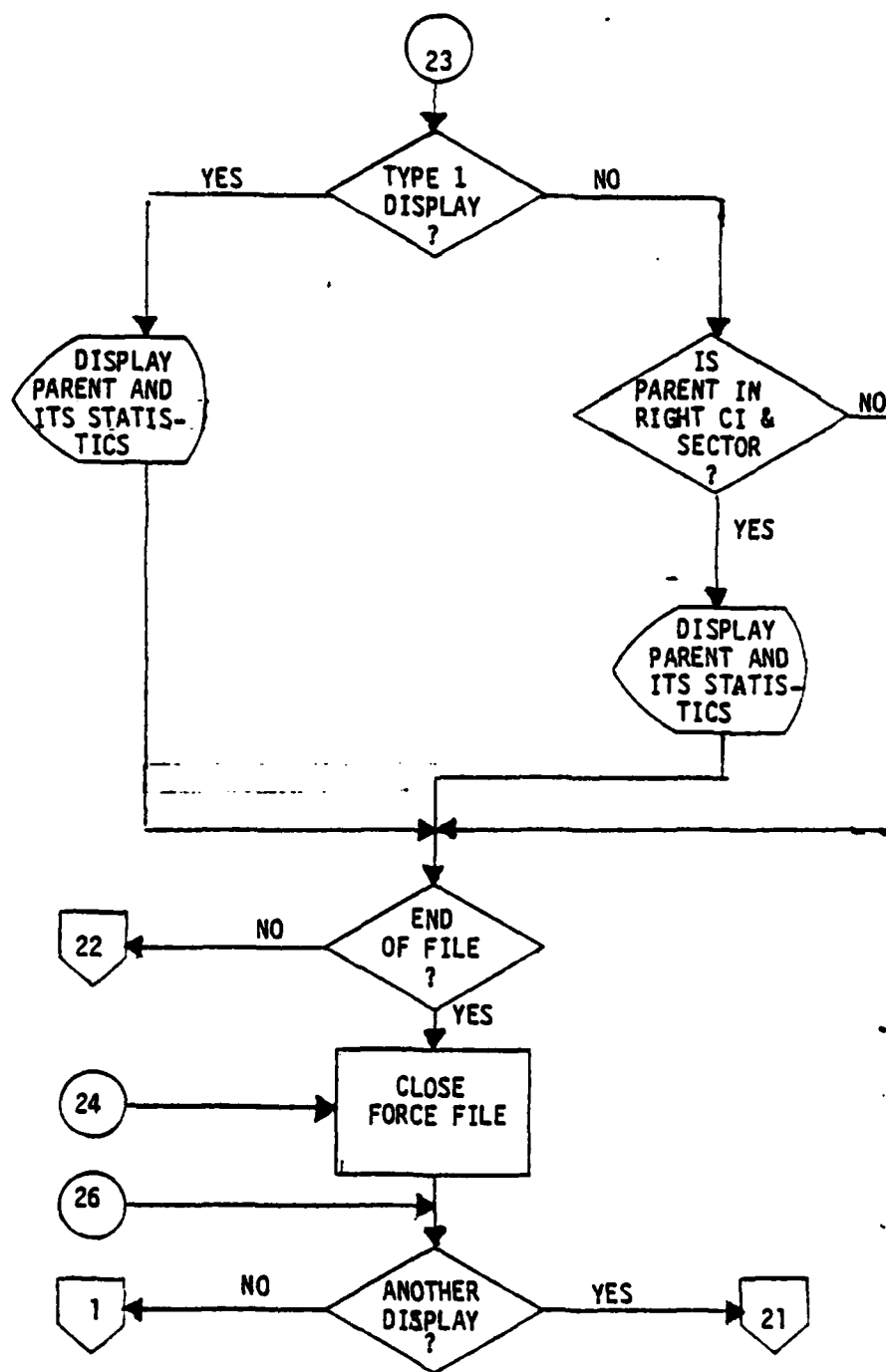


Figure I-9 FORCE flow diagram (continued).

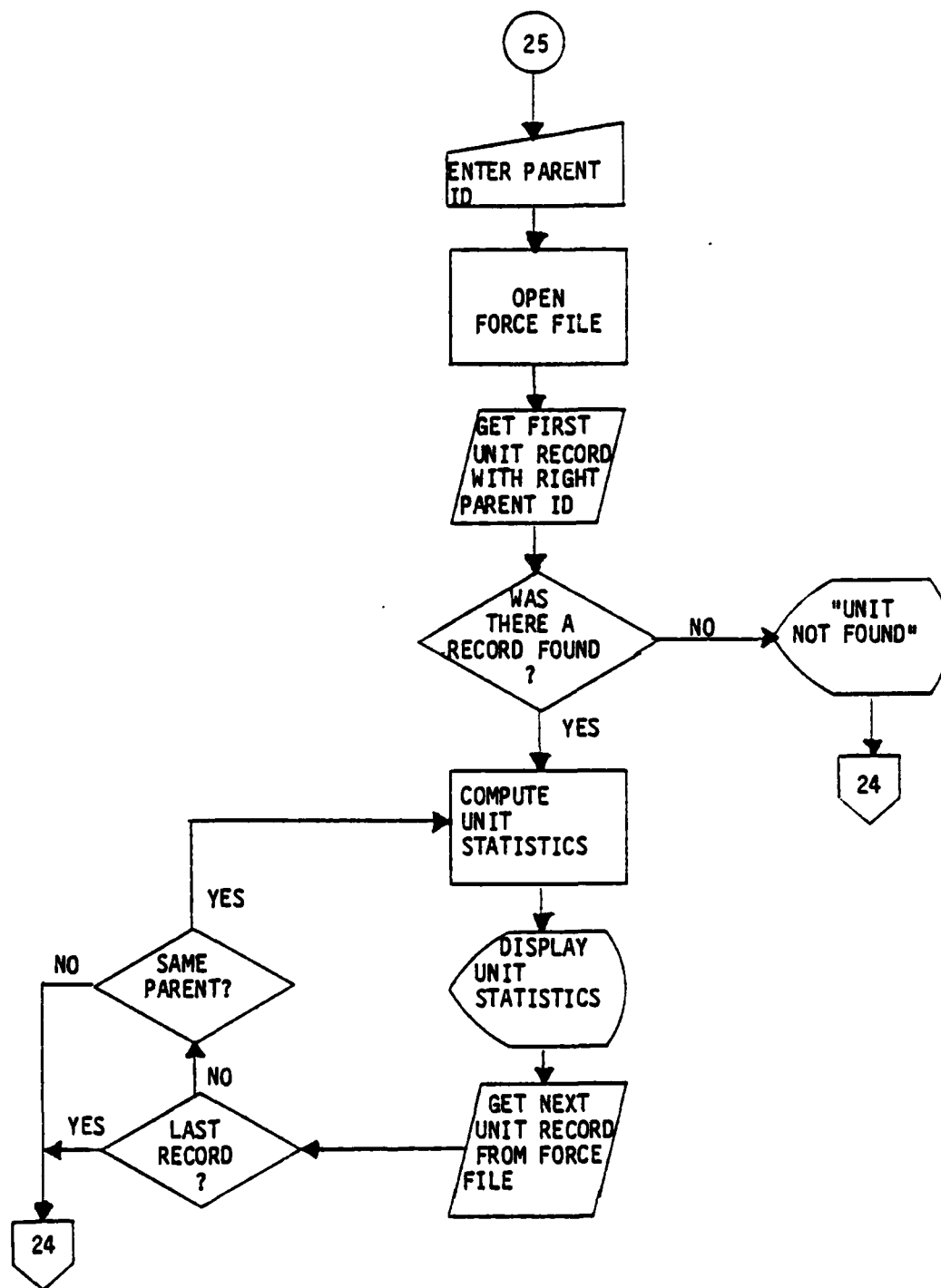


Figure I-9 FORCE flow diagram (continued).

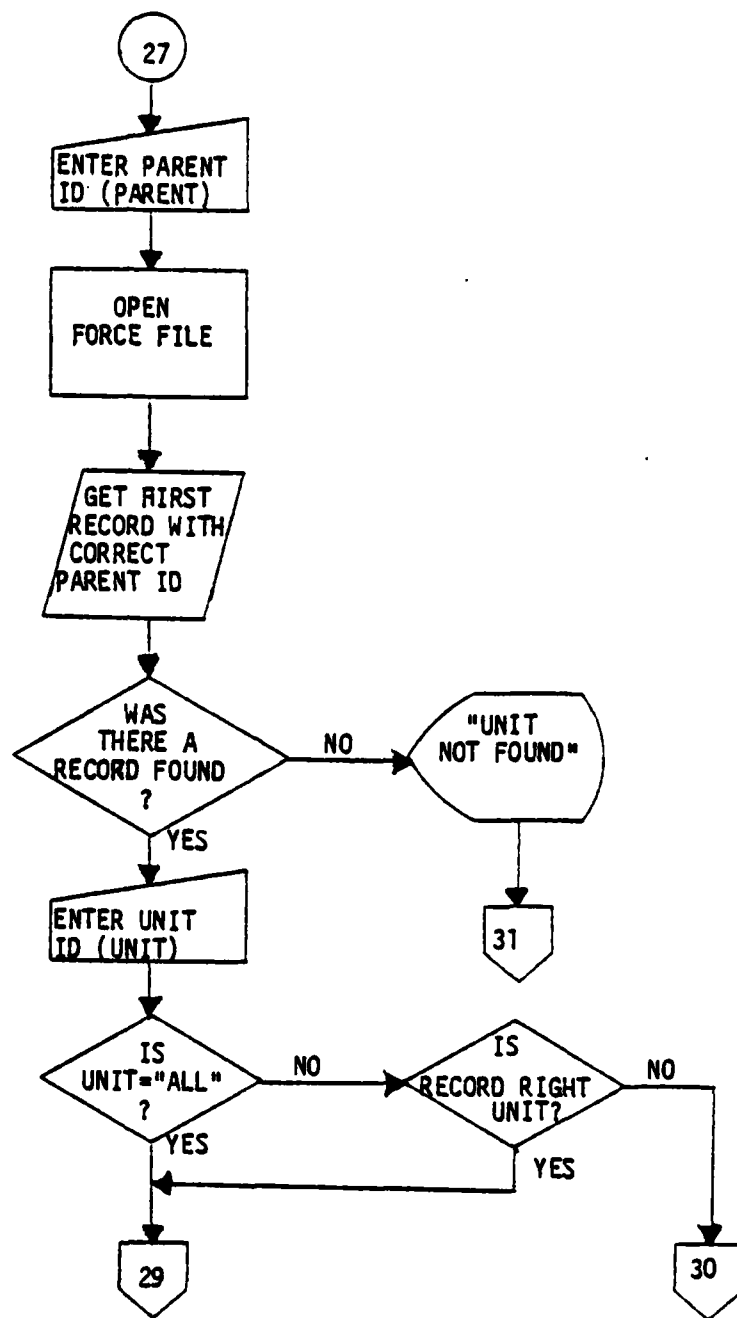


Figure I-9 FORCE flow diagram (continued).

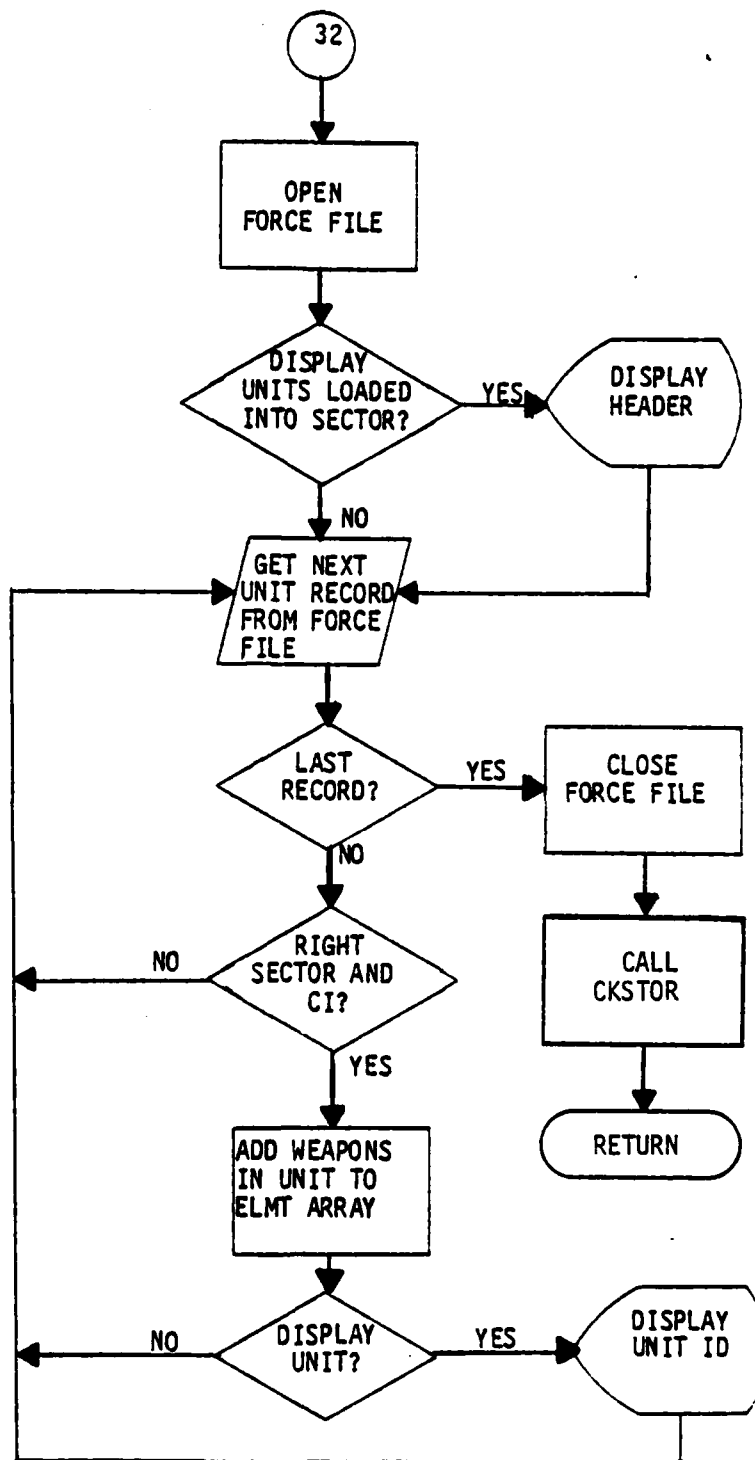


Figure I-9 FORCE flow diagram (continued)

AD-AU92 783

ARMY COMBINED ARMS STUDIES AND ANALYSIS ACTIVITY FOR--ETC F/6 15/7
CACDA JIFFY III WAR GAME. VOLUME V. PROGRAMMER'S MANUAL.(U)

OCT 80 C L PAO

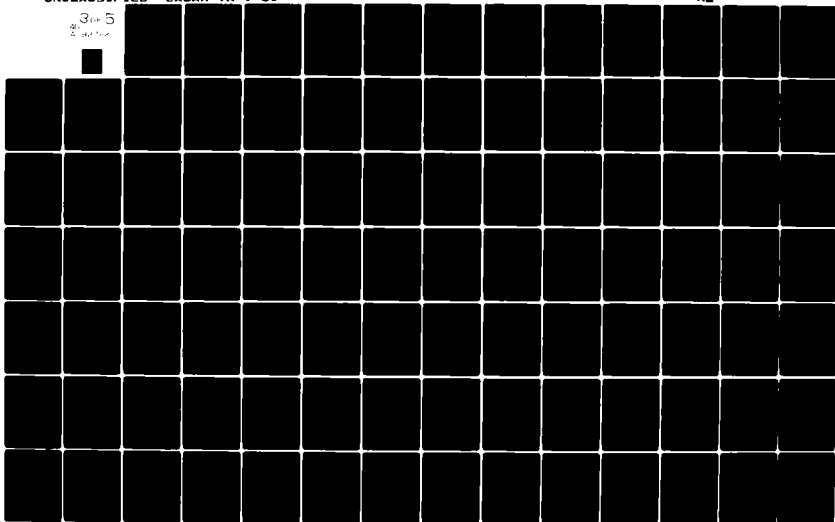
CASAA-TR-9-80

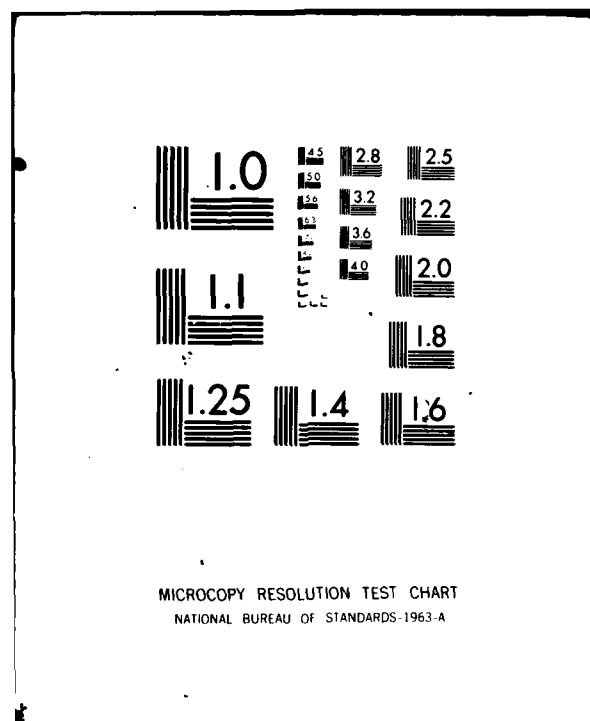
UNCLASSIFIED

NL

3 of 5

2 of 100





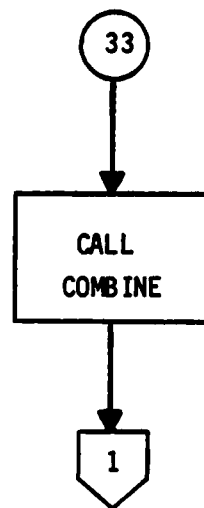


Figure I-9. FORCE flow diagram (concluded)

APPENDIX I

SECTION X. SUBROUTINE LOSS

I-X-1. SUBROUTINE DESCRIPTION. This subroutine is used to subtract weapon systems lost in the combat assessment subroutines from the weapon system arrays. This also allows the gamer to replay the combat, if the original assessment is for some reason invalid. LOSS stores the ammunition expended during a valid assessment in the ammo accumulator SHOTS.

I-X-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AKILL	The number of weapons type I kill by all firers
I	Firing weapon system index
INX	Gamer response variable
ISTART	Variable indexing beginning subscript of firers
ISTOP	Variable indexing ending subscript of firers
J	Force identifier
K	Index for weapon systems lost
KSTART	Variable indexing beginning subscript of weapon systems lost
KSTOP	Variable indexing ending subscript of weapon system lost

NOTE: All common variables are defined in appendix F.

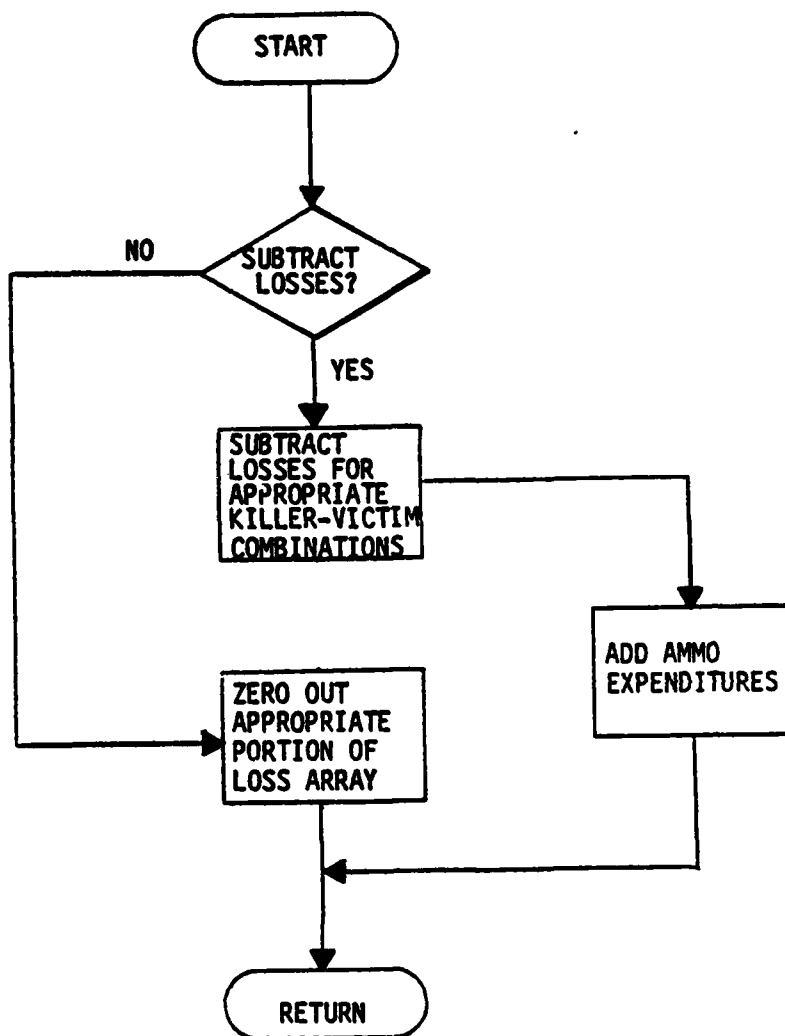


Figure I-10. LOSS flow diagram.

APPENDIX I

SECTION XI. SUBROUTINE NEWHIS

I-XI-1. SUBROUTINE DESCRIPTION. This subroutine creates a CI on the history file without losses to permit restarting at the start of the battle. The gamer is not permitted to make assessments in this CI. Also NEWHIS allows the gamer to assign units to sectors without entering the name of the unit. The names are displayed on the screen by NEWHIS.

I-XI-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AKEEP	Old parent name
I	Do loop variable
IACT	Gamer input variable
II	Do loop variable
INX	Gamer input variable
IYN	Gamer input variable
J	Do loop variable
M	Final position of force file
NEW	Existing record flag
SECT	Sector number

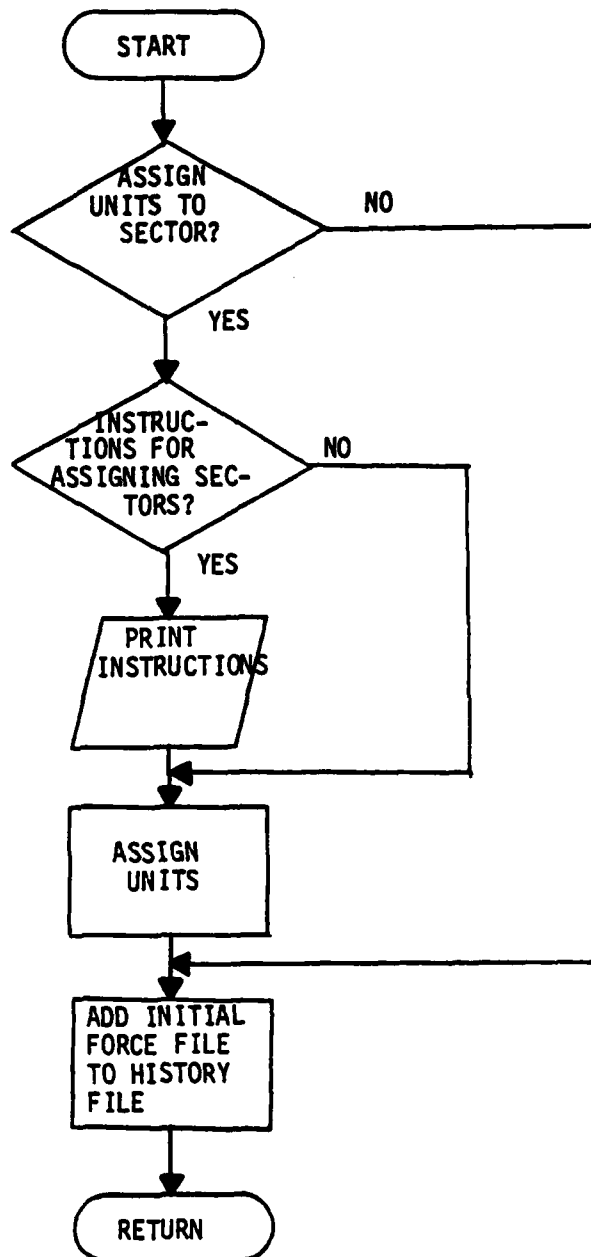


Figure I-11. NEWHIS flow diagram.

APPENDIX 'I'

SECTION XII. SUBROUTINE RESET

I-XII-1. SUBROUTINE DESCRIPTION. This subroutine permits quick investigations of the assessment routines without going through the loading processes. It is used for problems that do not requiring apportioning losses to units. It resets the ELEMNT array directly.

I-XII-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
ACIT	CI on ELEMNT file
ASECTT	Sector on ELEMNT file
I	Weapon system index
INX	Gamer response variable
MFORCE	ELEMNT file key
NCARD	ELEMNT file key
PCI	Previous CI
PSECT	Previous sector

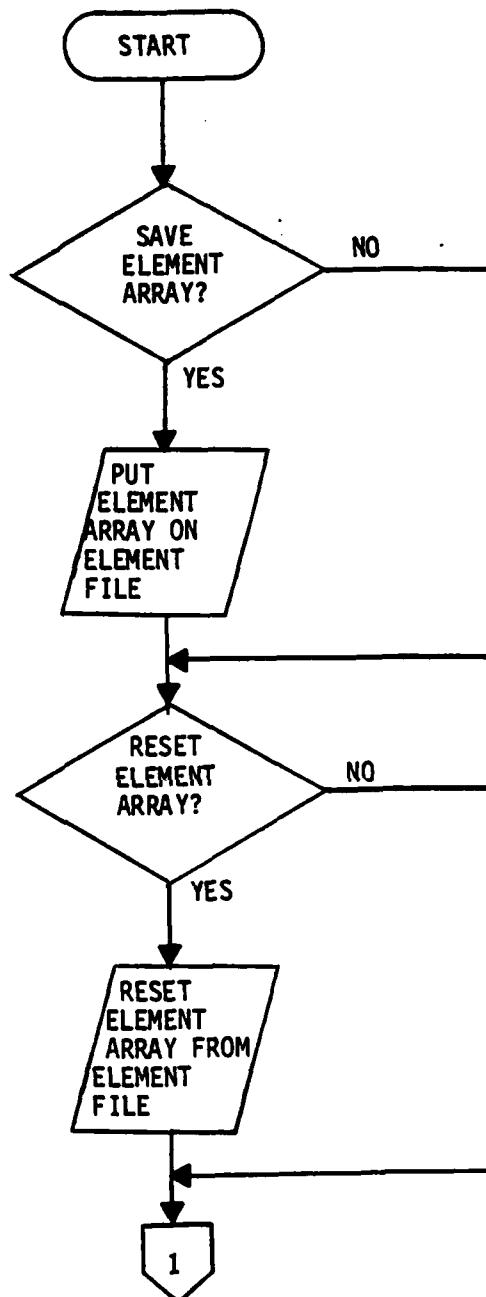


Figure I-12. RESET flow diagram (continued next page).

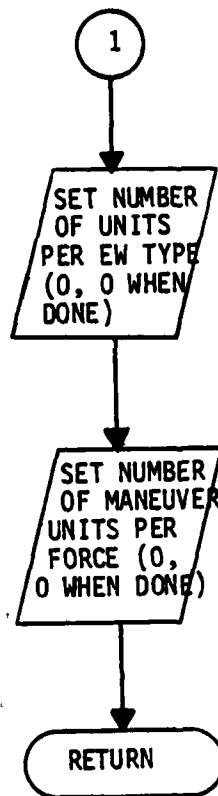


Figure I-12. RESET flow diagram (concluded).

APPENDIX I

SECTION XIII. SUBROUTINE RESTART

I-XIII-1. SUBROUTINE DESCRIPTION. This subroutine permits the gamer to load the entire force file for a given CI or at just selected sectors of the CI.

I-XIII-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
HOLD	Old CI name
IALL	Game response variable
II	Do loop index
IYN	Gamer response variable
M	Final position of history file
MM	Final position of force file record
NEW	Force file record flag

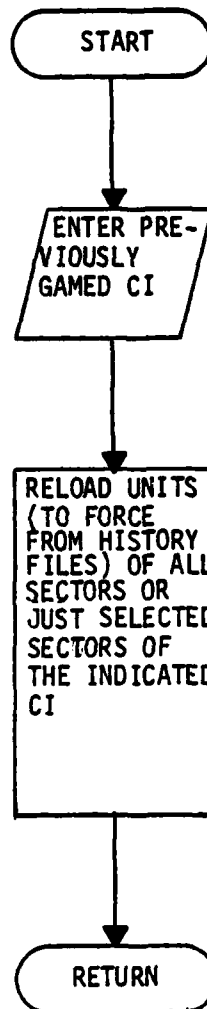


Figure I-13. RESTART flow diagram.

APPENDIX I

SECTION XIV. SUBROUTINE SHOWCI

I-XIV-1. SUBROUTINE DESCRIPTION. This subroutine permits the gamer to obtain a list of the CI's on the history file.

I-XIV-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
AKEEP	Old CI
IYN	Gamer response variable

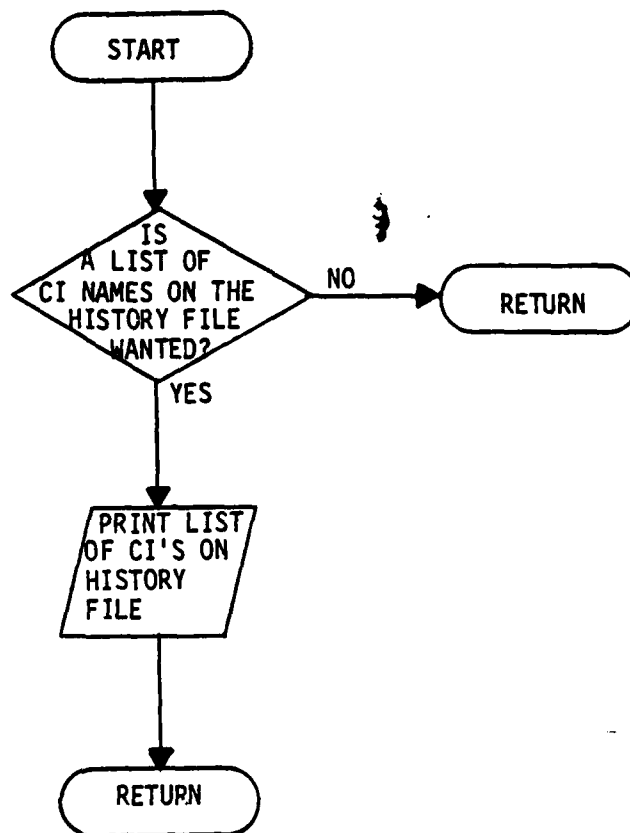


Figure I-14. SHOWCI flow diagram.

APPENDIX J

INTERACTIVE INTERFACE SUBROUTINES

Section I INTERACTIVE INTERFACE OVERVIEW

Section II SUBROUTINE DESCRIPTION AND VARIABLES

APPENDIX J

SECTION I. INTERACTIVE INTERFACE OVERVIEW

J-I. GENERAL DESCRIPTION. There are ten subroutines in this group. Each subroutine is defined in the next section. Figure J-1 shows the overview of these subroutines interactive relations. The major function is designed for more efficient use of the program. It permits quick investigation assessment routines without going through the loading processes. The other functions are to write on the ANSWER file and REVIEW files. It checks for end of ANSWER file and informs the gamer.

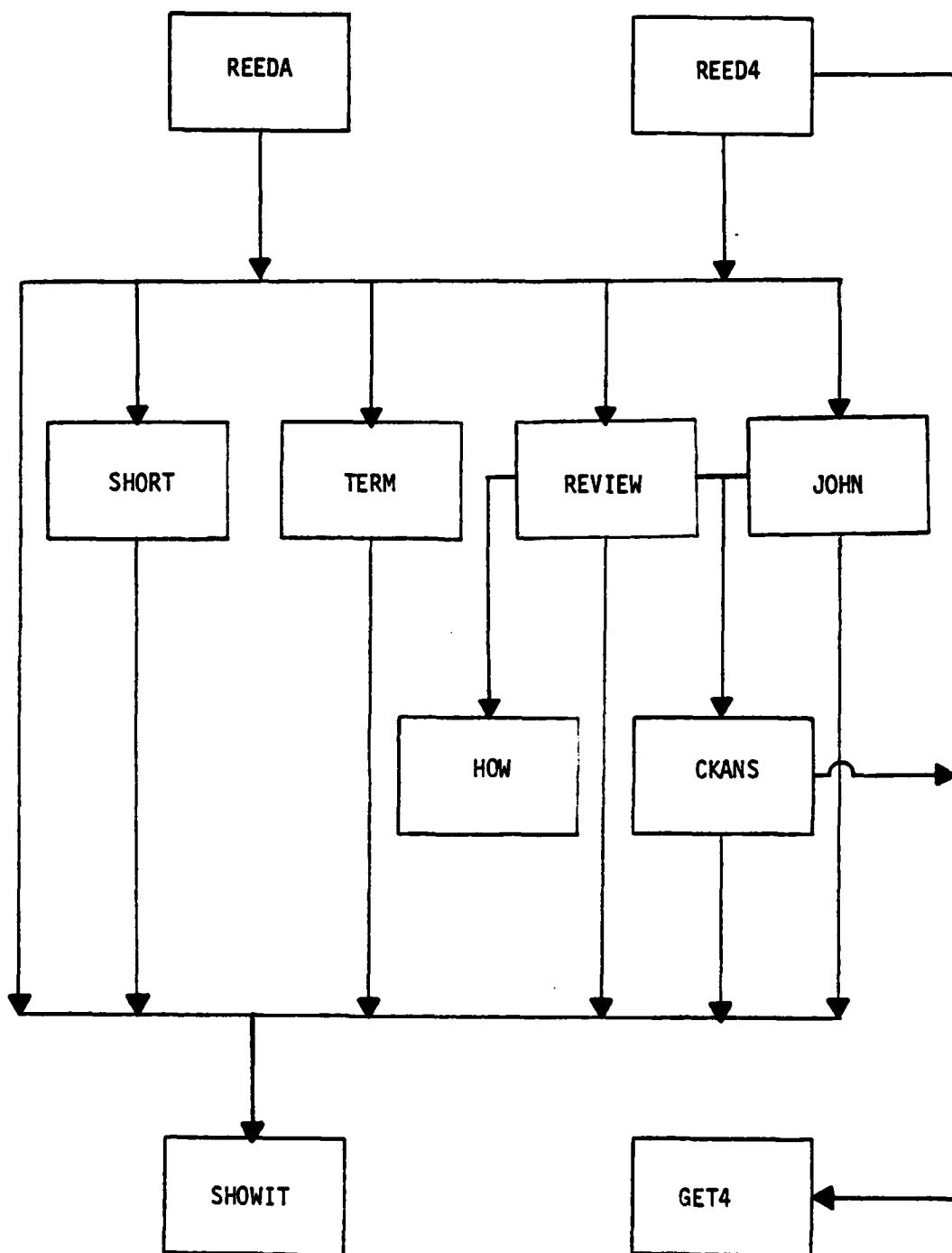


Figure J-1. Interactive interface overview flow diagram.

J-II. SUBROUTINE DESCRIPTION AND VARIABLES.

SUBROUTINE CKANS

1. CKANS.

a. DESCRIPTION. The subroutine CKANS allows the users to see and modify the answer file produced using either subroutine JOHN (batch option) or subroutine REVIEW (previous answer file option).

b. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
I	Character index
ICHANGE	Change flag
IGETS	Starting character of answer
INSRT	New answer number
INX	Gamer response variable
ISTACK	Lines of answers displayed
J	Answer number
JJ	Line number
J2	Number of last answer on line JJ

2. GET 4.

a. DESCRIPTION. This subroutine reads up to 4, positive or negative, real or integer values from gamer inputs.

b. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
IDO	Do loop index
IFACT	Index factor
IMULT	Multi function factor to produce real number
IMULT2	End of string factor
INEL	Negative flag
IOUT	Number of numbers read
LINE	Gamer input line
XNOS	Arrays of input numbers
XNUMB	Element of XNOS

3. HOW.

a. DESCRIPTION. The subroutine HOW is used to determine, through gamer input, how many answers on the REVIEW file are to be executed.

b. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
IERROR	Error return variable

4. JOHN.

a. DESCRIPTION. This subroutine processes batch mode input which allows the gamer to enter a series of answers without waiting for the computer response.

b. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
ILIM	Maximum number of answers
INX	Gamer response variable
INZ	Gamer response check variable

5. REEDA.

a. DESCRIPTION. REEDA provides a central input subroutine whenever the program needs a y or n. It checks for all the option available in the interactive package. Also REEDA has an entry point REEDA to provide for an input, up to 10 characters long, whenever the program needs it.

b. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
IARMOR	New armor flag
ICARD	Scratch array for gamer input
JDAY1	Day or night flag
XINX	Real array for gamer input.

6. REED4.

a. DESCRIPTION. This subroutine reads up to 4 numerical values and checks for all options in the interactive package. Also REED4 writes on the ANSWER file and REVIEW files. It checks for the end of the ANSWER file and informs the gamer.

b. PROGRAM VARIABLES.

Same as REEDA.

7. REVIEW.

a. DESCRIPTION. This subroutine allows the gamers to review questions and answers of the previous run and run the same game with any of the answers changed.

<u>Variable</u>	<u>Description</u>
IDEX	Answer number in review
IEDIT	Question number
IGETS	First column of new answer
IGET1	Number of answers to read from input line
IJN	Answer number in JOHN
IJOHN	Batch mode flag
IJON	Answer array
IMAX	Number of answer in IJON
IREVF	Review mode flag
I13	Storage array for current questions.

8. SHORT.

a. DESCRIPTION. This subroutine produces end of input file messages and connects the technical input file.

b. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
NORM	Normal return variable

9. SHOWIT.

a. DESCRIPTION. This subroutine rewinds the intermediate output file (Tape 51) and copies it to the terminal output file (Tape 50) and in the meantime produces the answer file and the QA (question and answer) file.

b. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
I	Word index
ISHOW	Entry point variable
IZ	Last line
J	Word index
JJ	Reverse word index
SHOWME	Terminal output entry point
SHOW3	Terminal and answer file output entry point

10. TERM.

a. DESCRIPTION. This subroutine connects or disconnects the output file depending on gamer input produces associate messages.

b. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
INX	Gamer response variable
IRUN	Fun index type
NGOT	Normal return variable

APPENDIX K

FUNCTIONS

Section I INDEX 5

Section II NUMB

APPENDIX K

SECTION I. SUBROUTINE INDEX5

K-I-1. SUBROUTINE DESCRIPTION. This subroutine is a subfunction that calculates a one-dimensional subscript from a five-dimensional variable.

K-I-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
INDEX5	Equivalent single subscript
I1	First parameter subscript
I2	Second parameter subscript
I3	Third parameter subscript
I4	Fourth parameter subscript
I5	Fifth parameter subscript
L1	Length of first parameter array
L2	Length of second parameter array
L3	Length of third parameter array
L4	Length of fourth parameter array

NOTE: All common variables are defined in appendix F.

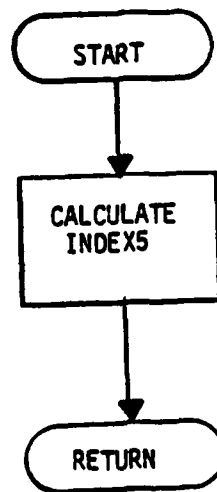


Figure K-1 INDEX5 flow diagram.

APPENDIX K

SECTION II. SUBROUTINE NUMB

K-II-1. SUBROUTINE DESCRIPTION. This subroutine is a subfunction that produces a string number left justified followed by blanks, determined by an integer input.

K-II-2. PROGRAM VARIABLES.

<u>Variable</u>	<u>Description</u>
INUMB	Scratch variable
IT	Do loop index
IX	Input of input number
I1	100th digit
I2	10th digit
I3	Unit digit
NUMB	Hollerith number output

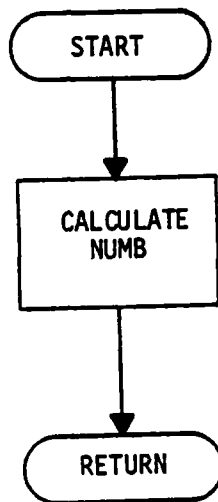


Figure K-2. NUMB flow diagram.

APPENDIX L

POSTPROCESSOR SUBROUTINES

Section	I	GENERAL DESCRIPTION
Section	II	SUBROUTINES DESCRIPTION
Section	III	COMMON ARRAYS AND VARIABLES
Section	IV	SAMPLE OUTPUTS FROM THE POSTPROCESSOR

APPENDIX L

POSTPROCESSOR SUBROUTINES

L-I. GENERAL DESCRIPTION. The Jiffy postprocessor consists of 17 subroutines and designed to assist the analysts and gamers working with the Jiffy war game in studies. The history file accessed by the postprocessor consists of records written during the subsequent to running of the Jiffy war games. The postprocessor can then summarize the results data into various levels of reports. A sample of print-outs is shown at the end of this appendix. A overview flow diagram of the postprocessor is shown in figure L-1. The description and program variables for each subroutine are given in the following section.

L-II. SUBROUTINE DESCRIPTION.

1. ACCUM. This subroutine assesses the history file to fill the Killer/Victim Array (ALOSS), initial weapon/array (BECON), ending weapon array (ELMT), and the ammunition expenditure array (SHOTS).
2. BULLET. BULLET writes out the table of ammunitions usage. It checks to see if type of report request (IOUT) is within bounds and skips to a new page. The other functions are: writes out heading information, report information array SHOTS, which generates the ammunition expenditure table for the postprocessor output file.
3. COMMIT. This subroutine sums the weapons remaining in all committed units at the end of a specified CI. A committed unit is one that is currently being gamed or has been gamed during some CI. Uncommitted units are in the sector zero and are subtracted from array BEGIN. The other functions are:
 - o checks range of CI index
 - o loads keys to access directly the index sequential history file for a CI.
 - o reads unit records and accumulates weapons committed in array ELMT.
 - o Subtracts uncommitted weapons from array BEGIN.
4. GITNMS. This subroutine retrieves the record containing the CI names and stores it in ARRAY. It indexes of first and last CI of interest in ARRAY are stored in NC11 and NC12 respectively for partial or full game output. The other functions are:
 - o sets keys for direct access to index sequential file for "CINAMES" record.

- o puts record accessed into ARRAY.
 - o searches for index of CIs of interest CCI(1) and CCI(2) and sets indexes NC11, NC12.
 - o insures CCI(1) and CCI(2) are in proper chronological order.
5. HEADER. This subroutine writes out proper heading for reports being created and checks to insure that the report parameter (IOUT) is in bounds.
 6. KILCAT. This subroutine writes losses by source of loss table to the output file, and generates losses by source of loss tables (red and blue) for postprocessor output file using array CATLOS.
 7. KILVIC. This subroutine writes out the killer/victim matrices for the six types of assessments.
 8. NEWPAG. This subroutine causes printer to skip to the top of the page and writes a brief header.
 9. OPTLER. This subroutine enables the generation of optional LER's interactively, and calculates LER. It also prints LER requested with number of weapons only and with firepower times the number of weapons.
 10. POST. This is the executive subroutine for the Jiffy game output processor. The type output to be generated is set along with the appropriate parameters and then subroutines take over to actually accumulate the desired game data and create the output file.
 11. RATIOS. RATIOS computes and writes out the headings and standard CI ratio statistics to include:
 - o Loss exchange ratio (LER).
 - o Initial force ration (IFR).
 - o Fractional exchange ratio (FER).
 - o Surviving maneuver force ratio (SMFR).
 - o Surviving maneuver force ratio differential (SMFRD).
 12. RECVRY. This subroutine computes the damage and recovery table and writes out the damage and recovery tables.
 13. REPORT. This subroutine calls all subroutines in sequence and forms reports.
 14. SUMTOT. This subroutine fills the total array by summing the BEGIN, ELMT, and D arrays. Total is then used to generate the ratio statistics.
 15. SUMUP. This subroutine fills CATLOS, CATSUM, and D from the ALOSS array.

16. TRACE. This subroutine generates sequential LERs, SMFRs, SMFRDs for partial or full game. The other functions are:

- o Sets beginning CI of interest.
- o Sets ending CI of interest.
- o Sets a new page.
- o Calls XOUT to zero arrays.
- o Calls ACCUM to fill arrays.
- o Calls RATIOS to form report.
- o If required, repeats process.

17. XOUT. This subroutine zeroes all arrays and variables.

L-III. COMMON ARRAYS AND VARIABLES.

1. Two programming methods are used to transfer data among the post processor subroutines:

- a. Arrays and variables are passed in the subroutine calling sequences.
- b. Arrays and variables are stored in COMMON blocks.

2. Definition of COMMON Arrays and Variables.

IOUT - Contains interactively entered output option

- 1. Single sector
- 2. Partial CI (multiple sector)
- 3. Full CI
- 4. Partial game
- 5. Full game

KIND (2)

- 1 = literal = "BLUE"
- 2 = literal = "RED"

TOTAL (3,2,2)(I,J,K) totals for weapons beginning/surviving/killed firepower.

I = 1	Total initial wpns
I = 2	Total surviving wpns
I = 3	Total wpns killed
J = 1	Blue weapon systems
J = 2	Red weapon systems
K = 1	Beginning numbers of weapons by type
K = 2	K = 1 times firepower score

Begin (65,2) accumulator of the number of weapons alive
(I,J) those committed at the beginning of output
period

I = 1 -65 Weapons type

J = 1 Blue

J = 2 Red

NCI1 Index of CI beginning multiple CI reports in vector ARRAY.

NCI2 Index of CI ending multiple CI reports in vector ARRAY.

CATLOS(65,11,2(I,J,K) Killer/victim loss arrays filled from ALOSS
65 weapons killed by 11 classes by Blue by Red

I = Weapon types 1-65 (Victim)

J = Weapon classes 1-11 (Killer)

K = 1 - Blue

2 - Red

CATSUM(11,2(I,J) Number of losses by category by Blue by Red

I = 1--11, category of files

J = 2, Blue

J = 2, Red

TYPE (5)

(1) literal = "ONE SECTOR"

(2) literal = "PARTIAL CI"

(3) literal = "FULL CI"

(4) literal = "PART GAME"

(5) literal = "FULL GAME"

NCI1 Index of CI in record CI NAMES of CI stored in CCI(1)

NCI2 Index of CI in record CI NAMES of CI stored in CCI(2)

LFIT(35) array used to rewind, open, close, and access the index sequential force file (not used in postprocessor).

ARRAY(90) Array used to store the names of the CI (for postprocessor only).
(I)

I = 1,2,3 unused

I = 4,.....,90 contains the name of the I-3 critical incident (CI).

MYBUE(1024) buffer used by GET and GETN utilities to access index sequential force file (not used in postprocessor).

D(80,2) Accumulator of weapon losses by weapon type
(I,J)

I = 1,.....,80, weapon type

I = 1 Blue

2 Red

ACI - Name of CI being processed

ASCENE - not used in the postprocessor

ASECT - number of sector being processed

COMMON/THREE/IHIST(35), AH(90), IYBUF(1024)

IHIST(35) array used to access index sequential history file

AH(90)	working storage for 90 word index for sequential history file
IYBUF(1024)	Buffer used by GET and BETN to access index sequential history file.
IA	Index for the attacker force
ID	Index for the defender force
IP	Index for the tactical situation table
IENGAG	Index for the tactical situation
ITERRN	Index for terrain type
IVIS	Index for visibility
IMOUNT	Index for attacker mobility
MINES	Minefield flag
CFRR	Maneuver firepower ratio
FSFPR	Fire support firepower ratio
FPR	Total firepower ratio
ATIME	Length of the CI in hours
IFIRST	Rate of advance calculation flag
IRUN	Index for run type
NORD	Night or day index (day = 1, night = 2)
FMASS	Fraction of sector (footage) massed by attacker
SF(2)	Suppression factor
FSSF(2)	Fire support suppression factor
PACK(2)	Work packing variables
ELMT(80,2)	Array of weapon system in the sector
ALOSS(80,80)	Weapon loss array (packed)
SHOTS(55,2)	Round expenditure array
CKILL(53,2)	Crew kills.
SHOT SV(55,2)	Temporary storage array for SHOTS



L-6

APPENDIX L

SECTION IV. SAMPLE OUTPUTS FROM THE POSTPROCESSOR

UNIT STATUS FILE FOR CI 24 1 SECTION 10011.

PARENT=CO EFF= 00.			UNIT=01 EFF= 00.		
EL	LOST	REMAIN	EL	LOST	REMAIN
1	33.7	884.3	1	33.7	416.3
2	241.6	4687.4	3	36.1	311.9
3	36.1	381.4	6	1.2	1210.8
6	1.2	1209.6	7	.1	112.9
7	.1	102.9	8	0.0	94.0
8	0.0	39.8	9	0.0	94.0
9	0.0	44.8	10	0.0	36.8
10	0.0	36.8	11	.3	287.7
11	.3	287.7	12	0.0	26.0
12	0.0	26.0	20	9.0	7.0
20	9.0	3.0	32	9.6	6.4
32	9.6	0.4	36	6.9	7.3
36	6.9	7.3	46	2.4	10.6
46	2.4	10.6	47	1.0	9.0
47	1.0	9.0	50	0.3	31.7
50	0.3	31.7	51	0.6	22.4
51	0.6	22.4	54	0.0	27.0
54	0.0	27.0	62	2.0	38.0
62	2.0	38.0	63	.6	22.4
63	.6	22.4	64	0.0	0.0
64	0.0	0.0	2	241.6	4687.4
PARENT=SO EFF= 00.			UNIT=01 EFF= 00.		
EL	LOST	REMAIN	EL	LOST	REMAIN
1	6.1	997.9	1	6.1	997.9
2	642.6	6162.4	3	964.2	219.0
3	964.2	289.8	6	1.0	1769.2
6	1.0	1769.2	8	.1	187.9
8	.1	187.9	9	0.0	22.0
9	0.0	22.0	11	.1	127.9
11	.1	127.9	13	0.0	36.0
13	0.0	36.0	22	10.2	.0
22	10.2	.0	26	6.3	.7
26	6.3	.7	29	99.0	38.2
29	99.0	18.2	37	10.1	29.9
37	10.1	29.9	39	.4	7.4
39	.4	7.4	43	3.6	3.6
43	3.6	3.6	44	9.1	11.9
44	9.1	11.9	50	0.6	46.6
50	0.6	46.6	51	19.7	148.3
51	19.7	148.3	53	1.6	10.4
53	1.6	10.4	54	6.7	47.3
54	6.7	47.3	63	2.0	10.0
63	2.0	10.0	2	642.6	6162.4

SECRET - TOP SECRET

ARMOR
RANGE BANDS

10011

n	δ
0	0.28
1	0.25
2	0.20
3	0.15
4	0.10
5	0.05
6	0.00
7	0.00
8	0.00
9	0.00
10	0.00

ONE SECTOR OUTPUT FOR CI- 2P

Y SECTOR- 10011.

FORCE STRUCTURE

1	636.	682.
2	6669.	6822.
3	334.	1174.
4	1222.	1767.
5	183.	8.
6	39.	188.
7	64.	22.
8	76.	8.
9	268.	128.
10	26.	8.
11	8.	36.
12	8.	67.
13	38.	8.
14	18.	8.
15	8.	11.
16	8.	138.
17	14.	3.
18	28.	18.
19	8.	8.
20	29.	138.
21	8.	7.
22	12.	8.
23	12.	8.
24	8.	66.
25	8.	8.
26	8.	8.
27	8.	7.
28	8.	21.
29	13.	8.
30	8.	8.
31	41.	93.
32	29.	168.
33	8.	12.
34	27.	24.
35	48.	8.
36	23.	28.
37	8.	8.

ONE SECTION OUTPUT FOR 21- 2F 1 SECTION- 10011.

LOSS 1 DAMAGE DISTRIBUTION									
BLUE FORCE					RED FORCE				
MPN TYPE	TOTAL DAMAGED	NUMBER LOST	EXCEED INITIATE	CUSCON DIV	MPN TYPE	TOTAL DAMAGED	NUMBER LOST	REPAIRABLE WITHIN	
1	2	3	4	5	6	7	8	9	10
1	10	10	0	0	1	10	10	0	0
2	10	10	0	0	1	10	10	0	0
3	10	10	0	0	1	10	10	0	0
4	10	10	0	0	1	10	10	0	0
5	10	10	0	0	1	10	10	0	0
6	10	10	0	0	1	10	10	0	0
7	10	10	0	0	1	10	10	0	0
8	10	10	0	0	1	10	10	0	0
9	10	10	0	0	1	10	10	0	0
10	10	10	0	0	1	10	10	0	0
11	10	10	0	0	1	10	10	0	0
12	10	10	0	0	1	10	10	0	0
13	10	10	0	0	1	10	10	0	0
14	10	10	0	0	1	10	10	0	0
15	10	10	0	0	1	10	10	0	0
16	10	10	0	0	1	10	10	0	0
17	10	10	0	0	1	10	10	0	0
18	10	10	0	0	1	10	10	0	0
19	10	10	0	0	1	10	10	0	0
20	10	10	0	0	1	10	10	0	0
21	10	10	0	0	1	10	10	0	0
22	10	10	0	0	1	10	10	0	0
23	10	10	0	0	1	10	10	0	0
24	10	10	0	0	1	10	10	0	0
25	10	10	0	0	1	10	10	0	0
26	10	10	0	0	1	10	10	0	0
27	10	10	0	0	1	10	10	0	0
28	10	10	0	0	1	10	10	0	0
29	10	10	0	0	1	10	10	0	0
30	10	10	0	0	1	10	10	0	0
31	10	10	0	0	1	10	10	0	0
32	10	10	0	0	1	10	10	0	0

1. The first step in the process of identifying a problem is to recognize that a problem exists. This is often done by comparing current performance with a desired state or goal. If there is a significant difference, a problem is identified.

ALIVE LOSSES BY SOURCE OF LOSS

L-12

7101-8135

OLIVE LOSSES BY SOURCE OF LOSS

VICTIM CAT.	UNE-COI	LI-ALCH	SMI ALCO	INF/CAV VEHICLE	MECH SIG	ISNW	DIMES	IND ELEC	AD-GUN	SHOOT DSL	LEAD DSL	ATK-MEL	LOC-IR	TOTAL
IMP CBI								15.0						15.0
LY ATCH				11.0	2.0	3.0		3.0						21.0
INF/CAV				1.0	3.0	6.0		12.0				11.0	7.0	40.0
ME ATCH				1.0	1.0	2.0		7.0				7.0	2.0	20.0
TANK				1.0	3.0	12.0		1.0				16.0	1.0	32.0
SM PIAR								18.0						18.0
AD GUN						1.0		1.0				2.0		4.0
SM HSL								9.0						9.0
AVR MEL										2.0	1.0			3.0
TRUCKS								35.0						35.0
HOV/TAP								3.0						3.0

ONE SECTION OUTPUT FOR 21-27 1 SECTION = 10011.

RED LOSSES BY SOURCE OF LOSS

VICTIM CAT.	INF. C31	LT. ATCH	CHD ATCH	INF/CAV VEHICLE	MECH ATCH	TANK	MINES	INF FIRE	AD. GUN	SHORAD USL	LRAD USL	AIR. MEL	INAGIO	TOTAL
INF CAT	45.0		200.0	170.0	250.0		100.0					57.0		1029.0
LT ATCH	5.0		10.0	0.0	11.0		0.0	0.0				1.0		27.0
20 ATCH							15.0							15.0
INF/CAV	10.0		45.0	20.0	42.0		50.0					17.0	10.0	220.0
ME ATCH	1.0		3.0	2.0	3.0		2.0					3.0	1.0	19.0
TANK	9.0		10.0	12.0	13.0		9.0					11.0	1.0	65.0
IN FIRE							35.0							35.0
AD GUN						1.0		1.0					4.0	6.0
SH MEL							10.0					3.0	3.0	22.0
AIR MEL										2.0				2.0
TRUCKS							5.0							5.0
MORTAR							12.0							12.0

ONE SECTION OUTPUT FOR CI- 24

I SECTION- 10011.

AMMUNITION EXPENDITURE			
BLUE		RED	
TYPE-NUMBER		TYPE-NUMBER	
1	46.	1	46.
2	0.	2	0.
3	126.	3	0.
4	0.	4	0.
5	0.	5	0.
6	0.	6	0.
7	0.	7	10.
8	147.	8	0.
9	0.	9	32.
10	97.	10	0.
11	0.	11	0.
12	1005.	12	0.
13	70.	13	192.
14	0.	14	0.
15	0.	15	0.
16	14400.	16	0.
17	0.	17	659.
18	0.	18	36.
19	198.	19	982.
20	0.	20	74.
21	946.	21	0.
22	0.	22	0.
23	0.	23	0.
24	0.	24	8799.
25	0.	25	0.
26	981.	26	0.
27	0.	27	0.
28	1029.	28	8692.
29	696.	29	818.
30	0.	30	2173.
31	1011.	31	628.
32	729.	32	299.
33	0.	33	4381.
34	0.	34	0.
35	320.	35	0.
36	0.	36	176.
37	185.	37	0.
38	0.	38	0.
39	0.	39	0.
40	0.	40	1409.
41	0.	41	0.
42	11985.	42	0.
43	0.	43	8408.
44	0.	44	4.
45	68.	45	0.
46	0.	46	0.
47	0.	47	0.
48	0.	48	22.
49	0.	49	0.
50	0.	50	0.
51	0.	51	2.
52	0.	52	3.
53	0.	53	0.
54	0.	54	0.
55	0.	55	0.

ONE SECTOR OUTPUT FOR CT- 2F 1 SECTOR- 10011.

KILLER-VICTIM MATRIX															
RED ARMOR															
KILLER															
VICTIM :	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
2							12.6				3.7			5.4	12.1
3							3.3				.5		13.3		1.6
11							2.3				.5		11.7		1.6
12							.2								
20							10.7				.7			.1	3.2
21							1.6				.3			.5	.6
26							1.2				.1			.2	.5
27							1.0				.3			.7	1.2
28							.9							.1	.3
29							2.1				.3			.2	.3
30															

ONE SECTOR OUTPUT FOR 21- 20 1 SECTOR - 10011.

		KILLER-VICTIM MAINTENANCE										BLUE APHOR									
VICTIM :		11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
2	-	27.4	22.9							159.2	71.0						50.2	111.0	52.0		
3	-	60.0	37.2							239.2	105.6						101.6	159.0	62.0		
6	-	60.0	37.2							239.2	105.6						101.6	159.0	62.0		
9	-	1.9	1.0							11.1	5.1						4.7	7.7	2.2		
9	-	.7	.6							9.2	2.0						1.7	2.2	1.0		
11	-	1.9	1.0							10.1	5.1						4.7	7.7	2.0		
17	-	3.0	1.0							15.5	6.0						6.5	11.3	6.1		
22	-	.1	.3							1.9	.9						.9	1.3	.2		
24	-	3.3	3.0							19.2	0.0						0.6	12.2	2.2		
26	-	.1	.1							.7	.2						.5	.6	.2		
27	-	.2	.5							2.5	1.0						1.0	1.0	.5		
29	-	3.5	3.2							19.2	0.0						0.2	12.2	2.2		
31	-																				

ONE SECTION OUTPUT FOR CI- 2P 1 SECTION- 10011.

*****KILLER-VICTIM MATRIX*****	
VICTIM	KILLER
2	2.1
17	.2
20	.3
29	.3

ONE SECTION OUTPUT FOR 21- 27 1 SECTION- 10011.

VICIN 2									
43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

ONE SECTION OUTPUT FOR CT: 37 1 SECTION- 10017.

KILLER-VICTIM MATRIX									
ONE SECTION									
VICTIM :									
	63	66	65	67	68	69	70	71	72
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0	0	0
63	0	0	0	0	0	0	0	0	0
64	0	0	0	0	0	0	0	0	0
65	0	0	0	0	0	0	0	0	0
66	0	0	0	0	0	0	0	0	0
67	0	0	0	0	0	0	0	0	0
68	0	0	0	0	0	0	0	0	0
69	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0
71	0	0	0	0	0	0	0	0	0
72	0	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0	0	0
74	0	0	0	0	0	0	0	0	0
75	0	0	0	0	0	0	0	0	0
76	0	0	0	0	0	0	0	0	0
77	0	0	0	0	0	0	0	0	0
78	0	0	0	0	0	0	0	0	0
79	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0
81	0	0	0	0	0	0	0	0	0
82	0	0	0	0	0	0	0	0	0
83	0	0	0	0	0	0	0	0	0
84	0	0	0	0	0	0	0	0	0
85	0	0	0	0	0	0	0	0	0
86	0	0	0	0	0	0	0	0	0
87	0	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	0	0	0
89	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0
91	0	0	0	0	0	0	0	0	0
92	0	0	0	0	0	0	0	0	0
93	0	0	0	0	0	0	0	0	0
94	0	0	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0	0
97	0	0	0	0	0	0	0	0	0
98	0	0	0	0	0	0	0	0	0
99	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0

SECRET-TOPI.

ONE SECTIVE OUTPUT FOR CI- 2P

T SECTOR- 10011.

KILLER-VICTIM MATRIX							
BLUE HELICOPTER							
VICTIM	59	60	61	62	63	64	65
2				100.0			
3				92.0			
4				92.0			
8				2.0			
9				1.1			
11				2.0			
17				10.9			
22				2.1			
24				6.7			
26				.6			
27				2.9			
29				0.1			
30				3.1			

ONE SECTION REPORT FOR CT- 24

SECTION- 0011.

.....KILLER-VICTIM MATRIX.....
RED 200
VICTIM : 31 32 33 34 35 36 37 38 39 40 41 42
.....KILLER
62 " 1.9 .9
63 " 1.9 .9

KILLER-VICTIM MATRIX													
VICTIM	KILLER												
	31	32	33	34	35	36	37	38	39	40	41	42	
03							2						

ONE 35301

NOTES

VICTIM -

1

739

• •

Abstract

•

Abstract

1

•

100

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

9.3.3

28 •

Tip

2.

3

37

1

28

THE

•

36

;

1

;

1

15.

Abstract

1

•••••

09988999

ONE SECTOR OUTPUT FOR C1-24 1 SECTOR-18011.

***** RATIO STATISTICS *****									
***** STANDARD OUTPUT *****									
ALL MAJOR WEAPON SYSTEMS									
WEIGHTED EQUALLY					WEIGHTED BY FYI				
LEA	388.3			3.07	58442.2			2.07	
	123.0				24392.7				
SHFR(BLUE)	216.2			.66	94977.3			.69	
	348.0				78978.0				
SHFR(RED)	375.7			.90	182942.7			.67	
	755.0				152489.0				
SHFRD	.66	.90	.16		.69	.67	.82		
IFR	755.0			2.22	152489.0			1.43	
	348.0				78978.0				
FFR	3.1			1.38	2.1			1.07	
	2.2				1.9				

ONE SECTOR OUTPUT FOR CI- 2P 1 SECTOR- 100IT.

***** RATIO STATISTICS *****									
(STANDARD OUTPUT)									
APHOR WEAPON SYSTEMS ONLY									
WEIGHTED EQUALLY					WEIGHTED BY FPS				
LER	301.9		3.27		33519.4		2.13		
	91.0				19798.3				
SHPR (BLUE)	34.2		.30		6198.7		.21		
	131.0				19999.0				
SHPR (PEO)	97.2		.16		2974.0		.00		
	399.0				36910.0				
SHPR (O)	.30	.16	.16		.21	.00	.13		
IPR	399.0		2.74		36910.0		1.03		
	131.0				19999.0				
FEP	3.3		1.20		2.1		1.16		
	2.7				1.0				

 13CH924 //// END OF LIST ////
 13CH924 //// END OF LIST ////

APPENDIX M
JIFFY III PROGRAM LISTINGS

APPENDIX M

JIFFY III PROGRAM LISTINGS

M-I. GENERAL DESCRIPTION.

M-I-1. This appendix contains the FORTRAN Program Source Code for the entire Jiffy III program. This method of presentation should be more efficient and useful than fragmenting the code into subroutines for inclusion in the various appendixes.

M-I-2. Several sensitivity tests have been run, resulting in modifications to model logic. In addition, the Jiffy III program was subjected to extensive review and debugging in December 1979. No significant model changes were required as a result of the sensitivity analysis and review; however, a few input data changes were made, which are incorporated in volume III.

M-I-3. The program listings in this appendix are from a printout dated 8 February 1980. The Jiffy model, however, is the subject of continued modification and improvement for future studies.

M-II. Index for Jiffy III Programs and Subroutines (Feb 80 version).

Name	Line Sequence Numbers		Page
	From	To	
PROGRAM BUILD (SRC Generation Program)	000001	000224	M-4
PROGRAM UNIT	000001	000294	M-9
PROGRAM PARENT	000001	000296	M-16
PROGRAM MATRIX (Force Generation Program)	000100	004690	M-23
PROGRAM SUPER	110	3140	M-33
SUBROUTINE SHOWIT	3390	3920	M-40
" REEDA, REEDB	3940	4520	M-42
" REED4	4540	4950	M-44
" GET4	4970	5450	M-45
" JOHN	5470	5960	M-46
" CKANS	5980	6560	M-47
" REVIEW	6580	7490	M-49
" HOW	7510	7640	M-51
" SHORT	7650	7800	M-52
" TERM	7820	8090	M-53
FUNCTION NUMB	8110	8240	M-54
" CKSTOR	8260	8580	M-55
" ENDCI	8600	9250	M-56
" RESTART	9270	10530	M-58
" SHOWCI	10550	10890	M-61
" INIT	10910	11350	M-62
FUNCTION INDEX5	11370	11460	M-63
SUBROUTINE LOSS	11480	11880	M-64
" DISPLAY	11900	13040	M-65
" RESET	13060	13800	M-68
" SUPRES	13820	14250	M-70
" ROFA	14270	18840	M-71
" EW	18860	20180	M-81
" TANK	20200	27680	M-84
" INFANT	27700	29480	M-100
" MINE	29500	31920	M-104
" FASCAM	31940	33160	M-110
" CANNON	33180	42070	M-113
" CLGP	42090	43670	M-132
" FORCE	43690	49760	M-136
" COMBINE	49780	50330	M-149
" CKACI	50350	51820	M-151
" NEWHIS	51840	52890	M-155
" CLEAR	52910	53390	M-158
" APPORT	53410	59110	M-159
" BUILD	59130	60430	M-171
" AHAD	60450	67530	M-174
" POST	71960	73140	M-189

<u>Name</u>	<u>Line Sequence Numbers</u>		
	<u>From</u>	<u>To</u>	<u>Page</u>
SUBROUTINE XOUT	73160	73680	M-192
" GITNMS	73700	74530	M-194
" ACCUM	74550	77240	M-196
" COMMIT	77260	77850	M-202
" SUMTOT	77870	78380	M-204
" SUMUP	78400	78800	M-206
" HEADER	78820	79280	M-207
" REPORT	79300	79690	M-208
" RECVRY	80570	81810	M-209
" KILCAT	81830	82270	M-212
" BULLET	82740	83080	M-214
" KILVIC	83100	84420	M-215
" OPTLER	84440	85270	M-218
" RATIOS	85290	86990	M-220
" NEWPAG	87010	87310	M-224
" TRACE	87330	87820	M-225

PROGRAM BUILD(INPUT,OUTPUT,TAPE9,TAPE6=INPUT)	000031
COMMON/ONE/IFIT(35),IFLG(1)	000032
DIMENSION ARRAY(46),ACMG(44),AERR(22),MYBUF(1024)	000033
19 ICK=0	000034
20 CALL FILEIS(IFIT,3LLFN,5LTAPE9,2LKA,ARRAY,2LPH,1LR,	000035
.3LFMI,3LYES,3LBS,1024,3LFMB,MYBUF)	000036
CALL CONNOC(5LTAPE6)	000037
IF(ICK.EQ.1H0) GO TO 9	000038
IF(ICK.EQ.1H1) GO TO 9	000039
ARRAY(1)=-SRC	000040
101 FORMAT(1A10)	000041
AHOLD=ARRAY(1)	000042
9 ARRAY(1)=AHOLD	000043
IF(ICK.EQ.1H0) GO TO 800	000044
IF(ICK.EQ.1H1) GO TO 16	000045
10 DO 11 I=2,46	000046
ARRAY(I) = 0	000047
11 CONTINUE	000048
DO 12 I = 1,44	000049
ACMG(I) = 0	000050
12 CONTINUE	000051
DO 13 I = 1,22	000052
AERR(I) = 0	000053
13 CONTINUE	000054
C	000055
C ABOVE DO LOOPS ZERO OUT WORK ARRAYS	000056
C	000057
14 PRINT 102	000058
102 FORMAT(1X,"ENTER ACTION TYPE(X FOR LIST) - ")	000059
111 FORMAT(1X,"FOLLOWING ACTIONS CAN BE EXECUTED",/,	000060
11X,"R= READ (REVIEW) A RECORD",/,	000061
21X,"A= ADD A NEW SRC",/,	000062
31X,"C= CHANGE/ADD WPN ID'S/QTYS WITHIN AN EXISTING SRC",/,	000063
41X,"D= DELETE AN SRC AND/OR WPN SYS IC WITHIN THE SRC",/,	000064
51X,"L= LIST ALL SRC'S ON FILE",/,	000065
61X,"E= END THE PROGRAM")	000066
READ(6,103) ICK	000067
103 FORMAT(1A1)	000068
IF(ICK.EQ.1HX) PRINT 111	000069
IF(ICK.EQ.1HR) GO TO 16	000070
IF(ICK.EQ.1HR) GO TO 500	000071
IF(ICK.EQ.1HA) GO TO 600	000072
IF(ICK.EQ.1HC) GO TO 700	000073
IF(ICK.EQ.1HD) GO TO 800	000074
IF(ICK.EQ.1HL) GO TO 1000	000075
IF(ICK.EQ.1HE) GO TO 900	000076
PRINT 104	000077
104 FORMAT(1X,"ACTION CODE ERROR-TRY AGAIN")	000078
GO TO 16	000079

500 PRINT 501	000050
501 FORMAT(1X,"READ-ENTER SRC(ENO TO EXIT)- ")	000051
READ(6,502) ASRC	000052
CALL OPENM(IFIT,3LI-0)	000053
502 FORMAT(1A10)	000054
IF(ASRC.EQ.3MEN0) GO TO 16	000055
ARRAY(2) = ASRC	000056
ARRAY(3) = 99999	000057
CALL GET(IFIT,ARRAY,ARRAY(1))	000058
IF(ARRAY(3).EQ.99999) GO TO 550	000059
PRINT 503 ,ARRAY(2)	000060
503 FORMAT(1X,"SRC=",A10,5X," IO QTY=")	000061
DO 505 I=3,45,2	000062
IF (ARRAY(I).EQ. 0) GO TO 505	000063
IDO=ARRAY(I)	000064
PRINT 504, (IDO,ARRAY(I+1))	000065
504 FORMAT(20X,I3,1X,F5.0)	000066
505 CONTINUE	000067
CALL CLOSEM(IFIT)	000068
GO TO 500	000069
550 PRINT 551 , ASRC	000070
551 FORMAT(1X,"SRC =",A10," NOT ON FILE")	000071
CALL CLOSEM(IFIT)	000072
GO TO 500	000073
16 CALL CLOSEM(IFIT)	000074
GO TO 10	000075
600 PRINT 601	000076
601 FORMAT(1X,"ADD-ENTER NEW SRC(ENO TO EXIT)-- ")	000077
CALL OPENM(IFIT,3LI-0)	000078
READ(6,502) ASRC	000079
IF(ASRC.EQ.3MEN0) GO TO 16	000080
ARRAY(2)=ASRC	000081
ARRAY(3) = 99999	000082
CALL GET(IFIT,ARRAY,ARRAY(1))	000083
IF(ARRAY(3).NE.99999) GO TO 610	000084
7000 FORMAT(1X,"ENTER TOTAL NO. OF WPN SYS IDS ",/)	000085
NN=1	000086
PRINT 7001	000087
7001 FORMAT(1X,"ENTER WPN ID,QTY--0,0 IF DONE ")	000088
807 READ(6,*) AM,AJ	000089
IF(AM.EQ.0) GO TO 808	000090
NN=NN+2	000091
ARRAY(NN)=AM	000092
ARRAY(NN+1)=AJ	000093
PRINT*,"NEXT--"	000094
GO TO 807	000095
808 CONTINUE	000096
CALL PUT (IFIT,ARRAY,460,ARRAY(1))	000097

N=IFETCH(IFIT,3,1RS)	000098
IF (N.EQ.4468) GO TO 618	000099
603 FORMAT(1X,"SRC",A18," ALREADY ON FILE")	000100
GO TO 612	000101
618 PRINT 603, ASRC	000102
612 DO 511 I=2,46	000103
ARRAY(I)=0	000104
611 CONTINUE	000105
CALL CLOSEN(IFIT)	000106
GO TO 600	000107
700 PRINT 701	000108
701 FORMAT(1X,"CHANGE-ENTER SRC(ENO TO EXIT)-- ")	000109
READ(6,702) ASRC	000110
CALL OPENN(IFIT,3,1-0)	000111
702 FORMAT(A18)	000112
IF (ASRC.EQ.3WEND) GO TO 14	000113
ARRAY(3) = 99999	000114
ARRAY(2) = ASRC	000115
CALL GET(IFIT,ARRAY,ARRAY(1))	000116
IF (ARRAY(3).NE. 99999) GO TO 750	000117
799 PRINT 703 , ARRAY(2)	000118
703 FORMAT(1X,"SRC ",A18," NOT ON FILE")	000119
CALL CLOSEN(IFIT)	000120
GO TO 700	000121
750 PRINT 7000	000122
READ(6,*) IEND	000123
IEND=IEND*2	000124
PRINT 7002	000125
7002 FORMAT(1X,"ENTER MPN ID, QTY, ID, QTY, ---",/)	000126
READ(6,*) (ACHG(I),I=1,IEND)	000127
K=0	000128
DO 700 J=1,IEND,2	000129
IF (ACHG(J).EQ. 0) GO TO 700	000130
DO 770 I=3,45,2	000131
IF (ARRAY(I).EQ.ACHG(J)) GO TO 776	000132
770 CONTINUE	000133
K=K+1	000134
AERR(K) = ACHG(J)	000135
DO 775 L=3,45,2	000136
IF (ARRAY(L).NE.0) GO TO 775	000137
ARRAY(L)=ACHG(J)	000138
ARRAY(L+1)=ACHG(J+1)	000139
GO TO 700	000140
775 CONTINUE	000141
GO TO 700	000142
776 ARRAY(I+1) = ACHG(J+1)	000143
780 CONTINUE	000144
790 CALL REPLC(IFIT,ARRAY,460,ARRAY(1))	000145

IF (K.EQ.0) GO TO 791	000146
PRINT 706, ASRC, (AERR(I) , I=1,K)	000147
706 FORMAT(1X,"FOLLOWING WPN IDS NOT FOUND FOR SRC",A10,/,	000148
11X,"THEREFORE WERE ADDED TO THE SRC RECORD",/,22(F5.0))	000149
791 CALL CLOSEN(IFIT)	000150
GO TO 700	000151
800 PRINT 801	000152
801 FORMAT(1X,"DELETE-ENTER SRC(ENO TO EXIT)-- ")	000153
READ(6,1265) ASRC	000154
CALL OPENH(IFIT,3LI-0)	000155
1265 FORMAT(A10)	000156
IF(ASRC.EQ.3HEND) GO TO 16	000157
PRINT 7003	000158
7003 FORMAT(" ENTER TOTAL NO. OF WPN SYSTEMS TO BE DELETED--",	000159
7"ENTER 0 IF ALL -")	000160
READ(6,*) IEND	000161
ARRAY(2)=ASRC	000162
ARRAY(3)=99999	000163
CALL GET(IFIT,ARRAY, ARRAY(1))	000164
IF(ARRAY(3).EQ.99999) GO TO 840	000165
IF(IEND.EQ.0) GO TO 890	000166
PRINT*," ENTER WPN SYS ID'S TO BE DELETED "	000167
READ(6,*) (ACHG(I),I=1,IEND)	000168
IF(ARRAY(3).NE.99999) GO TO 858	000169
840 PRINT*," SRC ",ARRAY(2)," NOT ON FILE "	000170
CALL CLOSEN(IFIT)	000171
GO TO 800	000172
858 IF(ACHG(1).EQ.0) GO TO 890	000173
K=0	000174
DO 860 J=1,IEND	000175
IF(ACHG(J).EQ.0) GO TO 860	000176
DO 860 I=3,4,5,2	000177
IF (ARRAY(I).EQ.0) GO TO 860	000178
IF(ARRAY(I).EQ.ACHG(J)) GO TO 866	000179
860 CONTINUE	000180
K=K+1	000181
AERR(K) = ACHG(J)	000182
GO TO 860	000183
866 ARRAY(I) = 0	000184
ARRAY(I+1) = 0	000185
880 CONTINUE	000186
CALL REPLC(IFIT,ARRAY,460,ARRAY(1))	000187
IF(K.EQ.0) GO TO 891	000188
PRINT 8000,ASRC,(AERR(I),I=1,K)	000189
8000 FORMAT(1X,"THE FOLLOWING WPN I.D.'S WERE NOT FOUND FOR SRC"	000190
6,1X,A10,/,1X,22(F5.0))	000191
891 CALL CLOSEN(IFIT)	000192
GO TO 800	000193

898	CALL OLTE(IPIT,ARRAY(1))	000194
	CALL CLOSEN(IPIT)	000195
	GO TO 888	000196
1088	CALL OPENN(IPIT,3LI-0)	000197
1188	CONTINUE	000198
	CALL GETN(IPIT,ARRAY,ARRAY(1))	000199
	N=I*ETCH(IPIT,2LFP)	000200
	IF(4.EQ.1088) GO TO 15	000201
	IF(AMOLD.EQ.ARRAY(1)) GO TO 1208	000202
	GO TO 1188	000203
15	CALL CLOSEN(IPIT)	000204
	GO TO 20	000205
1208	PRINT 503,ARRAY(2)	000206
	DO 1205 I=3,4,5,2	000207
	IF(ARRAY(I).EQ.0) GO TO 1205	000208
	IDQ=ARRAY(I)	000209
	PRINT 504,(IDQ,ARRAY(I+1))	000210
1205	CONTINUE	000211
	DO 1206 I=2,4,6	000212
	ARRAY(I)=0	000213
1206	CONTINUE	000214
	GO TO 1188	000215
988	PRINT 9888	000216
9888	FORMAT(1X,"ANY MORE FORCE STRUCTURES TO BE UPDATED? ",/,)	000217
	READ(6,9881) NY	000218
9881	FORMAT(A1)	000219
	IF(NY.EQ.1NY) GO TO 19	000220
	PRINT 981	000221
981	FORMAT(1X,"ALL DONE JOB HAS ENDED")	000222
	STOP	000223
	END	000224

PROGRAM UNIT(INPUT,OUTPUT,TAPE10,TAPE9,TAPE5=INPUT)	000001
COMMON/ONE/IPIT(35),NFI(35)	000002
DIMENSION BARRAY(24),ARRAY(46),BCHG(22),	000003
1AERR(22),BERR(22),BBERR(22),NYBUF(1024),NYBUF(1024)	000004
19 ICK=0	000005
20 CALL FILEIS(NFI,3LLFN,6LTAPE10,2LKA,BARRAY(1),2LPH,1LR,	000006
.3LPW1,3LYES,3LBFS,1024,3LPW8,NYBUF)	000007
CALL FILEIS(IPIT,3LLFN,5LTAPE9,2LKA,ARRAY(1),2LPH,1LR,	000008
.3LP4I,3LYES,3LBFS,1024,3LPW8,NYBUF)	000009
IF(ICK.EQ.1M0) GO TO 9	000010
IF(ICK.EQ.1M1) GO TO 9	000011
BARRAY(1)="SRC"	000012
101 FORMAT(A10)	000013
ARRAY(1)=BARRAY(1)	000014
BHOLD=BARRAY(1)	000015
9 BARRAY(1)=BARRAY(1)=BHOLD	000016
IF(ICK.EQ.1M0) GO TO 880	000017
IF(ICK.EQ.1M1) GO TO 14	000018
111 FORMAT(1X,"FOLLOWING ACTIONS CAN BE EXECUTED",/,	000019
11X,"R=READ(REVIEW) A RECORD",/,	000020
21X,"A=ADD A NEW UNIT",/,	000021
31X,"C=ADD SRC'S WITHIN AN EXISTING UNIT",/,	000022
41X,"O=DELETE A UNIT AND/OR SRC'S WITHIN THE UNIT",/,	000023
51X,"L=LIST ALL UNITS ON FILE",/,	000024
61X,"E=END THE PROGRAM")	000025
10 DO 11 I=2,24	000026
BARRAY(I)=0	000027
IF(I.GT.22) GO TO 11	000028
BCHG(I)=0	000029
AERR(I)=0	000030
11 CONTINUE	000031
C	000032
C ABOVE DO LOOP ZERO OUT WORK ARRAYS	000033
C	000034
14 PRINT 102	000035
102 FORMAT(1X,"ENTER ACTION TYPE(X FOR LIST) ---")	000036
READ(6,103) ICK	000037
103 FORMAT(A1)	000038
IF(ICK.EQ.1M1) PRINT 111	000039
IF(ICK.EQ.1M1) GO TO 14	000040
IF(ICK.EQ.1M2) GO TO 580	000041
IF(ICK.EQ.1M3) GO TO 680	000042
IF(ICK.EQ.1M4) GO TO 780	000043
IF(ICK.EQ.1M0) GO TO 880	000044
IF(ICK.EQ.1M1) GO TO 1080	000045
IF(ICK.EQ.1M2) GO TO 980	000046
PRINT 104	000047
104 FORMAT(1X,"ACTION CODE ERROR-TRY AGAIN ")	000048
GO TO 14	000049

C		000050
C	**THIS PORTION OF PROGRAM IS TO READ(REVIEW) UNITS**	000051
C		000052
	500 PRINT 501	000053
	501 FORMAT(1X,"READ-ENTER UNIT ID(END TO EXIT)-- ")	000054
	READ(6,502) BUNIT	000055
	CALL OPENH(NFIT,3LI-0)	000056
	502 FORMAT(A10)	000057
	IF(BUNIT.EQ.3HEND) GO TO 16	000058
	BARRAY(2)=BUNIT	000059
	BARRAY(3)=99999	000060
	CALL GET(NFIT,BARRAY,BARRAY(1))	000061
	IF(BARRAY(3).EQ.99999) GO TO 550	000062
	PRINT 503,BARRAY(2)	000063
	503 FORMAT(1X,"UNIT=",A10,5X,"SRC")	000064
	DO 505 I=3,24	000065
	IF(BARRAY(I).EQ.0) GO TO 505	000066
	PRINT 504,(BARRAY(I))	000067
	504 FORMAT(17X,A10)	000068
	505 CONTINUE	000069
	CALL CLOSEH(NFIT)	000070
	GO TO 500	000071
	550 PRINT 551,BUNIT	000072
	551 FORMAT(1X,"UNIT=",A10,"NOT ON FILE")	000073
	CALL CLOSEH(NFIT)	000074
	PRINT 1111,BARRAY(2)	000075
C		000076
C	**THIS PORTION OF PROGRAM IS TO ADD A NEW UNIT**	000077
C		000078
	600 PRINT 601	000079
	601 FORMAT(1X,"ADD-ENTER NEW UNIT ID(END TO EXIT)-- ")	000080
	READ(6,502) BUNIT	000081
	CALL OPENH(NFIT,3LI-0)	000082
	IF(BUNIT.EQ.3HEND) GO TO 16	000083
	BARRAY(2)=BUNIT	000084
	BARRAY(3)=99999	000085
	CALL GET(NFIT,BARRAY,BARRAY(1))	000086
	IF(BARRAY(3).NE.99999) GO TO 610	000087
	6000 FORMAT(1X,"ENTER TOTAL NO. OF SRC.S",/)	000088
	I=0	000089
	CALL OPENH(NFIT,3LI-0)	000090
	PRINT 506	000091
	506 FORMAT(1X,"ENTER SRC(0 IF DONE)--")	000092
	6501 I=I+1	000093
	READ(6,502) BARRAY(I+2)	000094
	IF(BARRAY(I+2).NE."0") GO TO 6500	000095
	BARRAY(I+2)=0	000096
	CALL CLOSEH(NFIT)	000097

	GO TO 616	000098
6508	CONTINUE	000099
	ARRAY(2)=BRRAY(I*2)	000100
	ARRAY(3)=99999	000101
	CAL. GET(IFIT,ARRAY,ARRAY(1))	000102
	IF(ARRAY(3).NE.99999) GO TO 602	000103
	BFRG=ARRAY(2)	000104
	GO TO 604	000105
602	CONTINUE	000106
	PRINT*,"NEXT="	000107
	GO TO 6501	000108
616	CAL. PUT(NFIT,BRRAY,2*8,BRRAY(1))	000109
	NO=IFETCH(NFIT,3LIRS)	000110
	IF(40.EQ.4468) GO TO 610	000111
603	FORMAT(1X,"UNIT ",A18," ALREADY ON FILE")	000112
	GO TO 612	000113
610	PRINT 603,BUNIT	000114
	GO TO 612	000115
604	PRINT 6002, BFRG	000116
6002	FORMAT(1X,"SRC ",A18," NOT ON FILE AND NOT ADDED ")	000117
	I=I-1	000118
	GO TO 602	000119
612	DO 611 I=3,24	000120
	BRRAY(I)=0	000121
611	CONTINUE	000122
	CALL CLOSE(NFIT)	000123
	GO TO 600	000124
16	CALL CLOSE(NFIT)	000125
	GO TO 10	000126
C		000127
C	**THIS PORTION OF PROGRAM ADDS SRC'S TO AN EXISTING UNIT**	000128
C		000129
700	DO 702 I=1,22	000130
	BERR(I)=0	000131
	BBERR(I)=0	000132
	AERR(I)=0	000133
	BCHS(I)=0	000134
702	CONTINUE	000135
	PRINT 701	000136
701	FORMAT(1X,"CHANGE-ENTER UNIT ID(ENO TO EXIT)-- ")	000137
	READ(6,502) BUNIT	000138
	CALL OPENH(NFIT,3LI-0)	000139
	IF(3UNIT.EQ.3HEND) GO TO 16	000140
	BRRAY(3)=99999	000141
	BRRAY(2)=BUNIT	000142
	CAL. GET(NFIT,BRRAY,BRRAY(1))	000143
	IF(BRRAY(3).NE.99999) GO TO 750	000144
799	PRINT 703,BRRAY(2)	000145

703	FORMAT(1X,"UNIT ",A10," NOT ON FILE")	000146
	CALL CLOSE(INFIT)	000147
	GO TO 700	000148
750	PRINT 6000	000149
	READ(6,*) NEND	000150
	DO 760 I=1,NEND	000151
	PRINT*,"ENTER SRC NO. ".I," ---"	000152
	READ(6,502) BCHG(I)	000153
760	CONTINUE	000154
	N=N+1	000155
	M=M+1	000156
C		000157
C	**ARRAY BERR KEEPS THOSE SRC'S TO BE ADDED**	000158
C	**ARRAY AERR KEEPS THOSE SRC'S ALREADY EXISTING IN THIS UNIT**	000159
C	**ARRAY BBERR KEEPS THE SRC'S IN BERR WHICH DO NOT EXIST**	000160
C		000161
	DO 780 J=1,NEND	000162
	IF(BCHG(J).EQ.0) GO TO 780	000163
	DO 770 I=3,24	000164
	IF(BARRAY(I).EQ.BCHG(J)) GO TO 776	000165
770	CONTINUE	000166
	M=M+1	000167
	BERR(M)=BCHG(J)	000168
	GO TO 780	000169
776	M=M+1	000170
	AERR(M)=BCHG(J)	000171
780	CONTINUE	000172
	NN=0	000173
	CALL OPENH(IFIT,3LI-0)	000174
	DO 785 I=1,M	000175
	ARRAY(2)=BERR(I)	000176
	ARRAY(3)=99999	000177
	CALL GET(IFIT,ARRAY,ARRAY(1))	000178
	IF(ARRAY(3).NE.99999) GO TO 715	000179
	NN=NN+1	000180
	BBERR(NN)=BERR(I)	000181
	GO TO 785	000182
715	DO 785 JK=3,24	000183
	IF(BARRAY(JK).NE.0) GO TO 785	000184
	BARRAY(JK)=BERR(I)	000185
	GO TO 785	000186
785	CONTINUE	000187
789	CONTINUE	000188
	CALL CLOSE(INFIT)	000189
	CALL REPLC(INFIT,BARRAY,240,BARRAY(1))	000190
707	IF(4.EQ.0) GO TO 709	000191
	PRINT 708,BUNIT	000192
708	FORMAT(1X,"FOLLOWING SRC.S WERE ALREADY PRESENT IN",	000193

1= UNIT -,A10)	000194
DO 7100 I=1,M	000195
PRINT 605,AERR(I)	000196
605 FORMAT(12X,A10)	000197
7100 CONTINUE	000198
709 IF(MN.EQ.0) GO TO 7201	000199
706 PRINT 7000	000200
7000 FORMAT(1X,"FOLLOWING SRC.S NOT FOUND ON TOE-FILE",/,	000201
21X,"THEREFORE WERE NOT ADDED ")	000202
DO 7200 I=1,MN	000203
PRINT 605,BGERR(I)	000204
7200 CONTINUE	000205
7201 CALL CLOSEH(NFIT)	000206
GO TO 700	000207
C	000208
C **THIS PORTION OF THE PROGRAM DELETES AN ENTIRE UNIT OR**	000209
C **DELETES SRC'S WITHIN A UNIT**	000210
C	000211
800 PRINT 801	000212
801 FORMAT(1X,"DELETE-ENTER UNIT ID(END TO EXIT)-- ")	000213
READ(6,8000) BUNIT	000214
CALL OPENH(NFIT,3LI-0)	000215
8000 FORMAT(A10)	000216
IF(BUNIT.EQ.3HEND) GO TO 10	000217
PRINT 8801	000218
8801 FORMAT(1X,"ENTER TOTAL NO. OF SRC.S TO BE DELETED--",	000219
3"ENTER 0 TO DELETE THE ENTIRE UNIT ")	000220
READ(6,*) NEND	000221
BARRAY(2)=BUNIT	000222
BARRAY(3)=99999	000223
CALL GET(NFIT,BARRAY,BARRAY(1))	000224
IF(BARRAY(3).EQ.99999) GO TO 840	000225
IF(NEND.EQ.0) GO TO 890	000226
DO 8500 I=1,NEND	000227
PRINT*, "ENTER SRC NO. -.I.- --"	000228
READ(6,502) BCHG(I)	000229
8500 CONTINUE	000230
GO TO 850	000231
840 PRINT1111,BUNIT	000232
1111 FORMAT(1X,A10," NOT ON FILE")	000233
CALL CLOSEH(NFIT)	000234
GO TO 800	000235
890 IF(BCHG(1).EQ.0) GO TO 890	000236
K=0	000237
DO 900 J=1,NEND	000238
IF(BCHG(J).EQ.0) GO TO 880	000239
DO 960 I=3,24	000240
IF(BARRAY(I).EQ.0) GO TO 860	000241

IF(BRRAY(I).EQ.BCHG(J)) GO TO 866	000242
868 CONTINUE	000243
K=K+1	000244
C	000245
C **AERR HERE KEEPS TRACK OF THE SRC'S NOT FOUND FOR THE UNIT**	000246
C	000247
AERR(K)=BCHG(J)	000248
GO TO 880	000249
866 BRRAY(I)=0	000250
880 CONTINUE	000251
CALL REPLC(NFIT,BRRAY,248,BRRAY(1))	000252
IF(K.EQ.0) GO TO 8101	000253
PRINT 8002,BUNIT	000254
8002 FORMAT(1X,"THE FOLLOWING SRC'S WERE NOT FOUND FOR UNIT ",	000255
4A10,1X)	000256
DO 8100 I=1,K	000257
PRINT 605,AERR(I)	000258
8100 CONTINUE	000259
8101 CALL CLOSEN(NFIT)	000260
GO TO 800	000261
890 CALL DLTE(NFIT,BRRAY(1))	000262
CALL CLOSEN(NFIT)	000263
GO TO 800	000264
C	000265
C **THIS PORTION OF THE PROGRAM LISTS ALL UNITS PRESENTLY ON FILE**	000266
C	000267
1000 CALL OPENH(NFIT,3LI-0)	000268
1100 CONTINUE	000269
CALL GETN(NFIT,BRRAY,BRRAY(1))	000270
M=IFETCH(NFIT,2LFP)	000271
IF(1.EQ.1000)GO TO 15	000272
IF(3HOLD.EQ.BRRAY(1)) GO TO 1200	000273
GO TO 1100	000274
15 CALL CLOSEN(NFIT)	000275
GO TO 20	000276
1200 PRINT 503,BRRAY(2)	000277
DO 1205 I=3,24	000278
IF(BRRAY(I).EQ.0) GO TO 1205	000279
PRINT 504, BRRAY(I)	000280
1205 CONTINUE	000281
DO 1206 I=2,24	000282
BRRAY(I)=0	000283
1206 CONTINUE	000284
GO TO 1100	000285
900 PRINT 9000	000286
9000 FORMAT(1X,"ANY MORE FORCE STRUCTURES TO BE UPDATED? ",/)	000287
READ(6,9001) NY	000288
9001 FORMAT(A1)	000289

```
IF(IY.EQ.1MY) GO TO 19  
PRINT 901  
901 FORMAT(1X," ALL DONE JOB HAS ENDED ")  
STOP  
END
```

```
000290  
000291  
000292  
000293  
000294
```


	PROGRAM PARENT(INPUT,OUTPUT,TAPE11,TAPE10,TAPE6=INPUT)	000001
	COMMON/ONE/NFIT(35),LFIT(35)	000002
	DIMENSION CRRAY(20),BRRAY(24),CGMG(10),	000003
	ICERR(10),BERR(10),SBERR(10),NYBUF(1024),MYBUF(1024)	000004
19	ICK=0	000005
20	CALL FILEIS(LFIT,3LLFN,6LTAPE11,2LKA,CRRAY(1),2LPN,1LR,	000006
	.3LFWI,3LYES,3LBFS,1024,3LFWB,NYBUF)	000007
	CALL FILEIS(NFIT,3LLFN,6LTAPE10,2LKA,BRRAY(1),2LPN,1LR,	000008
	.3LFWI,3LYES,3LBFS,1024,3LFWB,NYBUF)	000009
	IF(ICK.EQ.1H0) GO TO 9	000010
	IF(ICK.EQ.1HL) GO TO 9	000011
	CRRAY(1)="SRC"	000012
101	FORMAT(A10)	000013
	BRRAY(1)=CRRAY(1)	000014
	CHOLD=CRRAY(1)	000015
	9 CRRAY(1)=BRRAY(1)=CHOLD	000016
	IF(ICK.EQ.1H0) GO TO 300	000017
	IF(ICK.EQ.1HL) GO TO 14	000018
111	FORMAT(1X,"FOLLOWING ACTIONS CAN BE EXECUTED",/)	000019
	11X,"R=READ(REVIEW) A RECORD",/)	000020
	21X,"A=ADD A NEW PARENT",/)	000021
	31X,"C=ADD UNIT,S WITHIN AN EXISTING PARENT",/)	000022
	41X,"D=DELETE A PARENT AND/OR UNIT,S WITHIN THE PARENT",/)	000023
	51X,"L=LIST ALL PARENTS ON FILE",/)	000024
	61X,"E=END THE PROGRAM")	000025
10	DO 11 I=2,20	000026
	CRRAY(I)=0	000027
	IF(I.GT.10) GO TO 11	000028
	CGMG(I)=0	000029
	CERR(I)=0	000030
11	CONTINUE	000031
C		000032
C	ABOVE DO LOOP ZERO OUT WORK ARRAYS	000033
C		000034
14	PRINT 102	000035
102	FORMAT(1X,"ENTER ACTION TYPE(X FOR LIST) -- ")	000036
	READ(6,103) ICK	000037
103	FORMAT(A1)	000038
	IF(ICK.EQ.1HX) PRINT 111	000039
	IF(ICK.EQ.1HX) GO TO 14	000040
	IF(ICK.EQ.1HR) GO TO 300	000041
	IF(ICK.EQ.1HA) GO TO 600	000042
	IF(ICK.EQ.1HC) GO TO 700	000043
	IF(ICK.EQ.1HD) GO TO 800	000044
	IF(ICK.EQ.1HL) GO TO 1000	000045
	IF(ICK.EQ.1HE) GO TO 900	000046
	PRINT 104	000047
104	FORMAT(1X,"ACTION CODE ERROR-TRY AGAIN ")	000048
	GO TO 14	000049

C		000050
C	**THIS PORTION OF PROGRAM IS TO READ (REVIEW) PARENTS**	000051
C		000052
	500 PRINT 501	000053
	501 FORMAT(1X,"READ-ENTER PARENT ID(ENO TO EXIT)-- ")	000054
	READ(6,502) CFRC	000055
	502 FORMAT(A10)	000056
	CALL OPENM(LFIT,3LI-0)	000057
	IF(CFRC.EQ.3HEND) GO TO 16	000058
	CRRAY(2)=CFRC	000059
	CRRAY(3)=99999	000060
	CALL GET(LFIT,CRRAY,CRRAY(1))	000061
	IF(CRRAY(3).EQ.99999) GO TO 550	000062
	PRINT 503,CRRAY(2)	000063
	503 FORMAT(1X,"PARENT=",A10,5X,"UNIT")	000064
	DO 505 I=3,20	000065
	IF(CRRAY(I).EQ.0) GO TO 505	000066
	PRINT 504,(CRRAY(I))	000067
	504 FORMAT(17X,A10)	000068
	505 CONTINUE	000069
	CALL CLOSEM(LFIT)	000070
	GO TO 500	000071
	550 PRINT 551,CFRC	000072
	551 FORMAT(1X,"PARENT",A10,"NOT ON FILE")	000073
	CALL CLOSEM(LFIT)	000074
	GO TO 500	000075
16	CALL CLOSEM(LFIT)	000076
	GO TO 18	000077
C		000078
C	**THIS PORTION OF PROGRAM IS TO ADD A NEW PARENT**	000079
C		000080
	600 PRINT 601	000081
	601 FORMAT(1X,"ADD-ENTER NEW PARENT ID(ENO TO EXIT)-- ")	000082
	READ(6,502)CFRC	000083
	CALL OPENM(LFIT,3LI-0)	000084
	IF(CFRC.EQ.3HEND) GO TO 16	000085
	CRRAY(2)=CFRC	000086
	CRRAY(3)=99999	000087
	CALL GET(LFIT,CRRAY,CRRAY(1))	000088
	IF(CRRAY(3).NE.99999) GO TO 610	000089
	6000 FORMAT(1X,"ENTER TOTAL NO. OF UNIT,S",/)	000090
	I=0	000091
	CALL OPENM(NFIT,3LI-0)	000092
	PRINT 506	000093
	506 FORMAT(1X,"ENTER UNIT(0 IF DONE)--")	000094
	6501 I=I+1	000095
	READ(6,502) CRRAY(I+2)	000096
	IF(CRRAY(I+2).NE."0") GO TO 6500	000097

	CRRAY(I+2)=0	000098
	CALL CLOSE(INFIT)	000099
	GO TO 616	000100
C		000101
C		000102
6508	NE=8	000103
	BRRAY(2)=CRRAY(I+2)	000104
	BRRAY(3)=99999	000105
	CALL GET(INFIT,BRRAY,BRRAY(1))	000106
	IF(BRRAY(3).NE.99999) GO TO 602	000107
	CFRC=BRRAY(2)	000108
	GO TO 604	000109
602	CONTINUE	000110
	PRINT*, "NEXT--"	000111
	GO TO 6501	000112
616	CALL PUT(LFIT,CRRAY,200,CRRAY(1))	000113
	NO=IFETCH(LFIT,3,IRS)	000114
	IF(NO.EQ.4468) GO TO 610	000115
603	FORMAT(1X,"PARENT ",A10," ALREADY ON FILE")	000116
	GO TO 612	000117
610	PRINT 603,CRRAY(2)	000118
	GO TO 612	000119
604	PRINT 602, CFRC	000120
6002	FORMAT(1X,"UNIT ",A10,"NOT ON FILE AND NOT ADDED ")	000121
	I=I-1	000122
	GO TO 602	000123
612	DO 311 ILK=3,29	000124
	CRRAY(ILK)=8	000125
611	CONTINUE	000126
	CALL CLOSE(LFIT)	000127
	GO TO 608	000128
C		000129
C	**THIS PORTION OF PROGRAM ADDS UNIT'S TO AN EXISTING PARENT**	000130
C		000131
708	DO 702 I=1,10	000132
	BERR(I)=0	000133
	BBERR(I)=0	000134
	CBRR(I)=0	000135
	CCMG(I)=0	000136
702	CONTINUE	000137
	PRINT 701	000138
701	FORMAT(1X,"CHANGE-ENTER PARENT ID(ENO TO EXIT)-- ")	000139
	READ(6,502) CFRC	000140
	CALL OPENH(LFIT,3LI-0)	000141
	IF(CFRC.EQ.3HEND) GO TO 16	000142
	CRRAY(3)=99999	000143
	CRRAY(2)=CFRC	000144
	CALL GET(LFIT,CRRAY,CRRAY(1))	000145

IF(BRRAY(3).NE.99999) GO TO 750	000146
799 PRINT 703,CRRAY(2)	000147
703 FORMAT(1X,"PARENT ",A10," NOT ON FILE")	000148
CALL CLOSEN(LFIT)	000149
GO TO 700	000150
750 PRINT 6000	000151
READ(6,*) LEND	000152
DO 760 I=1,LEND	000153
PRINT*, "ENTER UNIT NO. ",I," --"	000154
READ(6,502) CCHG(I)	000155
760 CONTINUE	000156
N=0	000157
M=0	000158
C	000159
C **ARRAY BERR KEEPS THOSE UNIT'S TO BE ADDED**	000160
C **ARRAY CERR KEEPS THOSE UNIT'S ALREADY EXISTING IN THIS PARENT**	000161
C **ARRAY BBERR KEEPS THE UNIT'S IN BERR WHICH DO NOT EXIST**	000162
C	000163
DO 780 J=1,LEND	000164
IF(CCHG(J).EQ.0) GO TO 780	000165
DO 770 I=3,20	000166
IF(BRRAY(I).EQ.CCHG(J)) GO TO 776	000167
770 CONTINUE	000168
N=N+1	000169
BERR(N)=CCHG(J)	000170
GO TO 780	000171
776 N=N+1	000172
CERR(N)=CCHG(J)	000173
780 CONTINUE	000174
NN=3	000175
CALL OPENN(INFIT,3LI-0)	000176
DO 705 I=1,M	000177
BRRAY(2)=BERR(I)	000178
BRRAY(3)=99999	000179
CAL GET(INFIT,BRRAY,BRRAY(1))	000180
IF(BRRAY(3).NE.99999) GO TO 715	000181
NN=NN+1	000182
BBERR(NN)=BERR(I)	000183
GO TO 705	000184
715 DO 785 JK=3,20	000185
IF(BRRAY(JK).NE.0) GO TO 705	000186
CRRAY(JK)=BERR(I)	000187
GO TO 705	000188
785 CONTINUE	000189
705 CONTINUE	000190
CALL CLOSEN(INFIT)	000191
CALL REPLC(LFIT,CRRAY,200,CRRAY(1))	000192
707 IF(4.EQ.0) GO TO 709	000193

PRINT 700,CPRC	000194
700 FORMAT(1X,"FOLLOWING UNIT,S WERE ALREADY PRESENT IN",	000195
1- PARENT ",A10)	000196
DO 7100 I=1,M	000197
PRINT 605,CERR(I)	000198
605 FORMAT(12X,A10)	000199
7100 CONTINUE	000200
709 IF(4N.EQ.0) GO TO 7201	000201
706 PRINT 7000	000202
7000 FORMAT(1X,"FOLLOWING UNIT,S NOT FOUND ON UNIT-FILE",/,	000203
21X,"THEREFORE WERE NOT ADDED ")	000204
DO 7200 I=1,MN	000205
PRINT 605,BERR(I)	000206
7200 CONTINUE	000207
7201 CALL CLOSEN(LFIT)	000208
GO TO 700	000209
C	000210
C **THIS PORTION OF THE PROGRAM DELETES AN ENTIRE PARENT OR**	000211
C **DELETES UNIT'S WITHIN A PARENT**	000212
C	000213
800 PRINT 801	000214
801 FORMAT(1X,"DELETE-ENTER PARENT ID (END TO EXIT) -- ")	000215
READ(6,0000) CPRC	000216
CALL OPENH(LFIT,3LI-0)	000217
8000 FORMAT(A10)	000218
IF(CPRC.EQ.3HEND) GO TO 16	000219
PRINT 8001	000220
8001 FORMAT(1X,"ENTER TOTAL NO. OF UNIT,S TO BE DELETED-",	000221
3-ENTER 0 TO DELETE THE ENTIRE PARENT ")	000222
READ(6,*) LEND	000223
CRRAY(2)=CPRC	000224
CRRAY(3)=99999	000225
CALL GET(LFIT,CRRAY,CRRAY(1))	000226
IF(CRRAY(3).EQ.99999) GO TO 840	000227
IF(LEND.EQ.0) GO TO 890	000228
DO 8500 I=1,LEND	000229
PRINT*, "ENTER UNIT NO. ",I," --"	000230
READ(6,502) CCHG(I)	000231
8500 CONTINUE	000232
GO TO 850	000233
840 PRINT1111,CRRAY(2)	000234
1111 FORMAT(1X,A10," NOT ON FILE")	000235
CALL CLOSEN(LFIT)	000236
GO TO 800	000237
850 IF(CCHG(1).EQ.0) GO TO 890	000238
K=0	000239
DO 880 J=1,LEND	000240
IF(CCHG(J).EQ.0) GO TO 880	000241

DO 860 I=3,20	000242
IF(CRRAY(I).EQ.0) GO TO 860	000243
IF(CRRAY(I).EQ.CCHG(J)) GO TO 866	000244
860 CONTINUE	000245
K=K+1	000246
C	000247
C **CERR HERE KEEPS TRACK OF THE UNIT'S NOT FOUND FOR THE PARENT**	000248
C	000249
CERR(K)=CCHG(J)	000250
GO TO 888	000251
866 CRRAY(I)=0	000252
880 CONTINUE	000253
CALL REPLC(LFIT,CRRAY,200,CRRAY(1))	000254
IF(K.EQ.0) GO TO 880	000255
PRINT 8802,CERC	000256
8802 FORMAT(1X,"THE FOLLOWING UNIT'S WERE NOT FOUND FOR PARENT ",	000257
4A10,1X)	000258
DO 8100 I=1,K	000259
PRINT 805,CERR(I)	000260
8100 CONTINUE	000261
CALL CLOSEN(LFIT)	000262
GO TO 880	000263
890 CALL CLTE(LFIT,CRRAY(1))	000264
CALL CLOSEN(LFIT)	000265
GO TO 880	000266
C	000267
C **THIS PORTION OF THE PROGRAM LISTS ALL PARENTS PRESENTLY ON FILE**	000268
C	000269
1000 CALL OPENM(LFIT,3LI-0)	000270
1100 CONTINUE	000271
CALL GETN(LFIT,CRRAY,CRRAY(1))	000272
N=I*ETCH(LFIT,2LFP)	000273
IF(N.EQ.1000) GO TO 15	000274
IF(CHOLD.EQ.CRRAY(1)) GO TO 1200	000275
GO TO 1100	000276
15 CALL CLOSEN(LFIT)	000277
GO TO 20	000278
1200 PRINT 903,CRRAY(2)	000279
DO 1205 I=3,20	000280
IF(CRRAY(I).EQ.0) GO TO 1205	000281
PRINT 904, CRRAY(I)	000282
1205 CONTINUE	000283
DO 1206 I=2,20	000284
CRRAY(I)=0	000285
1206 CONTINUE	000286
GO TO 1100	000287
900 PRINT 9000	000288
9000 FORMAT(1X,"ANY MORE PARENT STRUCTURES TO BE UPDATED? ",/)	000289

READ (6,9881) NY	000290
9801 FORMAT (A1)	000291
IF (NY.EQ.1NY) GO TO 19	000292
PRINT 981	000293
981 FORMAT (1X, " ALL DONE JOB HAS ENDED ")	000294
STOP	000295
END	000296

M-23

.IX,"P=READ(REVIEW) P.PARENTS",/,	000590
.IX,"L=LIST ALL PARENTS",/,	000600
.IX,"I=INITIALIZE FORCE FILE",/,	000610
.IX,"E=END THE PROGRAM"	000620
25 PRINT 111	000630
110 FORMAT(IX,"ENTER ACTION TYPE(X FOR LIST)-- ")	000640
TCMS=0	000650
READ(10,120) IKK	000660
120 FORMAT(A1)	000670
IF(IKK.EQ.1XX) PRINT 111	000680
IF(IKK.EQ.1XX) GO TO 25	000690
IF(IKK.EQ.1XX) GOTO 235	000700
IF(IKK.EQ.1XX) GO TO 232	000710
IF(IKK.EQ.1XX) GO TO 230	000720
IF(IKK.EQ.1XX) GO TO 231	000730
IF(IKK.EQ.1XX) GO TO 235	000740
IF(IKK.EQ.1XX) GO TO 232	000750
IF(IKK.EQ.1XX) GOTO 231	000760
PRINT,"ACTION CODE ERROR "	000770
GO TO 25	000780
201 FORMAT(A1)	000790
205 TCMS=1	000800
PRINT,"ENTER PARENT I.O.() IF DONE"	000810
READ(201,APRAY(1))	000820
IF(APRAY(1).EQ."") GOTO 2	000830
PRINT,"ENTER UNIT I.O."	000840
READ(201,APRAY(2))	000850
APRAY(3)=99999.	000860
CALL OPENN(LFIT,3LI-0)	000870
CALL GET(LFIT,APRAY,APRAY(1))	000880
IF(APRAY(3).EQ.99999.) PRINT 211,APRAY(2),APRAY(1)	000890
IF(APRAY(3).EQ.99999.) GOTO 235	000900
GOTO 209	000910
211 FORMAT("UNIT I.O. ",A1," HAS NOT BEEN IDENTIFIED",/,	000920
" AS BELONGING TO PARENT ",A10," TRY AGAIN-")	000930
202 PRINT,"ENTER MAJOR FORCE() IF DONE"	000940
300 READ(10,201) ATOT(1)	000950
IF(ATOT(1).EQ."") GO TO 9	000960
AFOR(1)=AUID(1)=ASRC(1)="SRC"	000970
CALL OPENN(LFIT,3LI-0)	000980
AFOR(5)="FORCE"	000990
APRAY(4)=0.	001000
CALL OPENN(LFIT,3LI-0)	001010
AFOR(2) = ATOT(1)	001020
AFOR(3) = 99999.	001030
CALL GET(LFIT,AFOR,AFOR(1))	001040
IF(AFOR(3).EQ.99999.) GO TO 600	001050
CALL OPENN(LFIT,3LI-0)	001060

00 1000 J=3,2)	001070
IF(AFOR(J),EQ.3) GO TO 3999	001080
AUTO(2) = JFOR(J)	001090
IF(AFOR(J),EQ.3) GO TO 3999	001100
AUTO(2) = JFOR(J)	001110
IF(AFOR(J),EQ.3) GO TO 3999	001120
AUTO(2) = JFOR(J)	001130
IF(AFOR(J),EQ.3) GO TO 3999	001140
AUTO(2) = JFOR(J)	001150
IF(AFOR(J),EQ.3) GO TO 3999	001160
AUTO(2) = JFOR(J)	001170
IF(AFOR(J),EQ.3) GO TO 3999	001180
AUTO(2) = JFOR(J)	001190
IF(AFOR(J),EQ.3) GO TO 3999	001200
AUTO(2) = JFOR(J)	001210
IF(AFOR(J),EQ.3) GO TO 3999	001220
AUTO(2) = JFOR(J)	001230
AUTO(3) = 99999.	001240
CALL GET(JFIT,AUTC,AUTO(1))	001250
IF(AUTO(3),EQ.99999.) GO TO 7000	001260
ARRAY(6)=1	001270
CALL OPENM(KFIT,ALI=0)	001280
DO 1000 K = 1, 24	001290
IF(AUTO(K),EQ.0) GO TO 3000	001300
ASRC(2) = AUTO(K)	001310
ASPC(3) = 99999.	001320
CALL GET(KFIT,ASRC,ASRC(1))	001330
IF(ASPC(3),EQ.99999.) GO TO 2000	001340
00 2000 L = 3,45,2	001350
IF(ASPC(L),EQ.3) GO TO 2100	001360
ID = ASRC(L) + 1	001370
ARRAY(ID) = ARRAY(ID) + ASPC(L+1)	001380
ARRAY(5) = ARRAY(6) + ASRC(L+1) * FPS(ID-10,AA)	001390
2000 CONTINUE	001400
3000 CONTINUE	001410
CALL CLOSEM(KFIT)	001420
ARRAY(1)=ATOT(1)	001430
ARRAY(2)=AUTO(2)	001440
GOTO 3099	001450
200 YCHG=1	001460
CALL OPENM(LFIT,ALI=0,ILR)	001470
200 CALL GETN(LFIT,ARRAY,ARRAY(1))	001480
WRITE(6) (LFIT,2LFP)	001490
IF(4.EQ.1100) GOTO 210	001500
1000 POINT 9999,ARRAY(2)	001510
3665 FORMAT(' ENTER GEL EFF, GEL INTENSITY LVL, AND UNIT TYPE OF "1210")	001520
READ*,CV,CIL,UT	001530
IF(CIL.LT.1..OR.CIL.GT.3) GOTO 3677	001540

IF(UT.LT.0.02.UT.GT.0.1)GOTO6A88	001550
IF(CV.5E-01.AND.CV.LE.100.)GOTO3100	001560
PRINT*, "INVALID DEL EFF--TRY AGAIN "	001570
GOTO3099	001580
4877 POINT*, "INVALID CAT INTENSITY LEVEL--ENTRY MUST BE 3 TO 5"	001590
GOTO3099	001600
3384 PRINT*, "INVALID UNIT TYPE--TRY AGAIN"	001610
GOTO3099	001620
3150 DO 3155 II=1,33	001630
IF(ARRAY(II+1).LE.0.)GOTO3150	001640
ARRAY(II+1)=ARRAY(II+1)*CV/100.	001650
3150 CONTINUE	001660
ARRAY(7)=CIL	001670
ARRAY(3)=CV	001680
ARRAY(9)=CIL	001690
ARRAY(10)=UT	001700
IF(ICHG.EQ.1)GOTO204	001710
CALL PUT(LEFF,ARRAY,900,ARRAY(1))	001720
IF(LEFF.EQ.0)GOTO204	001730
IF(ICHG.EQ.1)GO TO 501	001740
GOTO207	001750
206 CALL PEOG(LEFF,ARRAY,900,ARRAY(1))	001760
IF(CHK.EQ.0)GOTO209	001770
210 CALL CLOSE*(JFIT)	001780
207 DO 2031 IJ=11,30	001790
2031 ARRAY(IJ)=0.	001800
IF(ICHG.EQ.1.AND.IKK.EQ.0)GOTO215	001810
IF(ICHG.EQ.1.AND.IKK.EQ.1)GOTO814	001820
3999 CONTINUE	001830
4503 CONTINUE	001840
CALL CLOSE*(JFIT)	001850
5000 CALL CLOSE*(IFIT)	001860
CALL CLOSE*(ILFIT)	001870
IF(ATOT(1).NE.0) PRINT*, "NEXT--"	001880
IF(ATOT(1).NE.0) GO TO 350	001890
3082 FORMAT(17F1)	001900
GO TO 9	001910
911 PRINT*, "DO YOU WANT TO CHANGE THE UNIT EFF, CIL, OR TYPE FOR	001920
ALL UNITS?"	001930
READ(17,201)INX	001940
IF(INX.EQ."Y")GOTO204	001950
IF(INX.EQ."N")GOTO814	001960
PRINT*, "RESPONSE MUST BE Y OR N "	001970
GOTO814	001980
814 CALL OPEN*(LEFF,3LI-0.1LP)	001990
CALL OPEN*(JFIT,3LI-0)	002000
CALL OPEN*(KFIT,3LI-0)	002010
512 CALL GETN(LEFF,ARRAY,ARRAY(1))	002020

```

      *IF ETC4(LFIT,2LEB)                                022030
      IF (M.F2.1000) GOTOC400                            022040
      AUID(1)=ASPC(1)*"SPC"                               022050
      AUID(2)=ARRAY(2)                                    022060
      AUID(3)=99999.                                       022070
      CALL GET(JFIT,AUID,AUID(1))                         022080
      IF (AUID(3).EQ.99999.) PRINT 7002,ARRAY(2)         022090
      IF (AUID(3).EQ.99999.) GOTOC705                    022100
7002 FORMAT(1X,"UNIT ID",A11," NOT ON UNIT FILE. THEREFORE NO WPA ID/QT") 022110
      1YS=7.1X,"WILL BE ASSOCIATED WITH THIS UNIT.")   022120
      NC750 I=11.00                                       022130
      1001V(I)=1.                                         022140
750 CONTINUE                                             022150
      111 ARRAY(6)=0.                                     022160
      012 NC710 K=3.00                                   022170
      IF (AUID(K).EQ.11GOTC711)                          022180
      ASRC(2)=AUID(K)                                     022190
      ASRC(3)=99999.                                       022200
      CALL GET(MFIT,ASRC,ASPC(1))                         022210
      IF (ASRC(3).EQ.99999.) PRINT 3001,AUID(K),ARRAY(2),ARRAY(1) 022220
      IF (ASPC(3).EQ.99999.) GOTOC713                     022230
      NC710 L=3.00+5.00                                   022240
      IF (ASPC(L).EQ.11GOTC731)                           022250
      ID=ASRC(L)+1                                         022260
      1001V(ID)=1001V(ID)+ASRC(L)+1                     022270
      1001V(6)=ARRAY(6)+ASRC(L)+1*EPS(ID-10,ARRAY(3))    022280
730 CONTINUE                                             022290
710 CONTINUE                                             022300
      NC750 I=1.00                                       022310
      IF (1001V(I+1).LE.11GOTC760)                       022320
      1001V(I+1)=1001V(I+1)*ARRAY(8)/100.              022330
760 CONTINUE                                             022340
765 1001V(6)="RESTART"                                  022350
      CALL REPLIC(LFIT,1001V,300,ARRAY(1))              022360
      GOTOC12                                              022370
400 CALL CLOSE4(JFIT)                                    022380
      CALL CLOSE4(LFIT)                                    022390
      CALL CLOSE4(KFIT)                                    022400
      GOTC9                                               022410
500 PRINT 501,ARRAY(3),1001V(1),1001V(2)              022420
501 FORMAT(1Y,3(2Y,11),1X,"ALREADY ON FILE")          022430
      GO TO 400                                           022440
600 PRINT 611                                           022450
611 FORMAT(1X,"DELETE-ENTER FORCE TO END TO EXIT)--")   022460
      READ(10,211) ASCENE                                 022470
      CALL OPEN4(LFIT,3LY=0)                              022480
      IF (ASCENE.EQ.34END) GO TO 613                     022490
613 CALL GET4(LFIT,ARRAY,1001V(1))                     022500

```

4=IFETCH(LFIT,2LEP)	002510
IF(4.EQ.1000) GO TO 612	002520
IF(ARRAY(3).NE.AM) GO TO 610	002530
IF(ARRAY(1).NE.ASCENE) GO TO 610	002540
CALL CLTE(LFIT,ARRAY(1))	002550
GO TO 610	002560
612 PRINT*, "ALL DONE WITH THIS DELETE"	002570
612 CALL CLOSE(LFIT)	002580
GO TO 9	002590
6002 PRINT 6001, ATOT(1)	002600
6001 FORMAT(1X,"ENC I.O.",A10," NOT PREVIOUSLY IDENTIFIED",1X,/,1X,	002610
1,"YOU WILL HAVE TO UPDATE THE FORCE FILE IF YOU WANT HIM INCLUDED",	002620
2/, " IN THE BATTLE")	002630
GO TO 6100	002640
7000 PRINT 7001, AFOR(J), ATOT(1)	002650
7001 FORMAT(1X,"UNIT ID",A10," HAS NOT BEEN PREVIOUSLY IDENTIFIED",/,	002660
1X,"AS BEING INCLUDED IN FORCE ID ",A10,/,	002670
2X,"THE FORCE IS BEING CREATED BUT THIS UNIT WAS LEFT OUT")	002680
GO TO 4100	002690
8000 PRINT 8001, AUTO(K), AFOR(J), ATOT(1)	002700
8001 FORMAT(1X," SPC ",A10,1X,"NOT ON THE FILE",/,	002710
1X,"THEFORE THE WPN I.R./QTY,S ASSOCIATED WITH THIS SRC",/,	002720
2X,"WILL NOT BE INCLUDED IN UNITID ",A10,1X,"FRC ID ",A10)	002730
GO TO 3100	002740
9000 PRINT*, "ENTER FORCE ID TO BE LISTED - "	002750
READ(10,201) ASCENE	002760
CALL OPENM(LFIT,3LI-0)	002770
ARRAY(1)=ASCENE	002780
PRINT 9025,TYPE	002790
9025 FORMAT(1X,"TYPE FORCE= ",A10//)	002800
AM=0	002810
CALL OPENM(LFIT)	002820
9020 CONTINUE	002830
CALL GETM(LFIT,ARRAY,ARRAY(1))	002840
4=IFETCH(LFIT,2LEP)	002850
IF(4.EQ.1000) GO TO 9040	002860
IF(AA.NE.ARRAY(3)) GO TO 9020	002870
IF(ASCENE.NE.ARRAY(1)) GO TO 9020	002880
IF(AM.EQ.ARRAY(1)) GO TO 9010	002890
PRINT 9025,ARRAY(1)	002900
9025 FORMAT(1X,"FORCE ID= ",A10)	002910
9001 PRINT 9027,ARRAY(2)	002920
9027 FORMAT(5X,"UNIT ID: ",A10,"; LAST CI: ",A10,"; SECTOR: ",	002930
1F3.1/23X,"WPN"/24X,"ID QTY")	002940
GO 9030 IF=11,90	002950
IF(ARRAY(1).EQ.1) GO TO 9030	002960
100= 7-10	002970
PRINT 9029,100,ARRAY(1)	002980

3225	FORMAT(24X,"Z,5X,F4.0)	002990
4031	CONTINUE	003000
	AVE=PPAY(1)	003010
	PRINT 9335,AVEAY(2),ARRAY(3)	003020
	GO TO 3121	003030
3040	CONTINUE	003040
	CALL CLOSEM(LFIT)	003050
	PRINT*, "DO YOU WANT TO LIST ANOTHER FORCE? "	003060
	READ(10,3002) IJK	003070
	IF(IJK.EQ.1) GO TO 9335	003080
3050	PRINT 9351	003090
3051	FORMAT(1Y," ACTION COMPLETE ",/,	003100
	1Y " DO YOU WANT TO END THE PROGRAM")	003110
	READ(10,3003) IJJ	003120
	IF(IJJ.EQ.1) GO TO 9905	003130
	GO TO 1	003140
250	CALL OPENM(LFIT,3LI-0)	003150
	PRINT 303	003155
	PRINT 3025,TYPE	003160
303	FORMAT("4")	003165
	PRINT 303	003170
	AVE=PCOFF/PTCOFF=0.	003180
	CALL REWIND(LFIT)	003190
252	CALL GETM(LFIT,AVEAY,PPAY(1))	003200
	M=PTCOFF(LFIT,2LE0)	003210
	IF(M.EQ.1000) GO TO 505	003220
	IF(11.NE.ARRAY(3)) GO TO 252	003230
	IF(14.EQ.0) GO TO 253	003240
	IF(14.EQ.0) GO TO 0702	003250
505	IF(MUNIT.EQ.0) GO TO 0132	003260
	PCOFF=309.	003270
	IF(PCOFF.NE.1) PARCOFF=PCOFF/PTCOFF*100.	003280
	PRINT 303	003290
	PRINT 331	003300
	PRINT 301	003310
	PRINT 301	003320
301	FORMAT(21X,24**,"F(21Y,14I)")	003330
	PRINT 152,14	003340
	PRINT 334	003350
304	FORMAT("4",20Y,14*)	003360
102	FORMAT("4",X,"D=CENT=",410,"")	003370
	L1=1	003380
120	IF(JFLAG.EQ.1) L1=6	003390
	L2=L1+4	003400
	NPRINTS=0	003410
	DO 310 TUN=L1,L2	003420
	IF(STATS(1,TUN).EQ.0) GO TO 310	003430
	T=STAT(2)=NUNE(2UN*10)	003440

IF(IUN.GT.5)IHEAD(2)=NUMB((IUN-5)*22)	003450
IF(JFLAG.EQ.1.AND.NPRINTS.EQ.0)PRINT 300	003460
IF(JFLAG.EQ.1.AND.NPRINTS.EQ.0)PRINT 301	003470
IF(JFLAG.EQ.1.AND.NPRINTS.EQ.0)PRINT 301	003480
PRINT IHEAD,STATS(1,IUN)	003490
NPRINTS=NPRINTS+1	003500
112 CONTINUE	003510
IF(NPRINTS.EQ.0)GOTO123	003520
130 FORMAT(7F4.1,2E(***))	003530
PRINT 301	003540
IEFF(2)=NUMB(0)	003550
IF(JFLAG.NE.1)PRINT IEFF,PAIEFF	003560
DO 125 IUN=L1,L2	003570
IF(STATS(1,IUN).EQ.0)GOTO125	003580
IEFF(2)=NUMB(IUN*22)	003590
IF(IUN.GT.5)IEFF(2)=NUMB((IUN-5)*22)	003600
PRINT IEFF,STATS(2,IUN)	003610
125 CONTINUE	003620
PRINT 301	003630
DO 126 IUN=L1,L2	003640
IF(STATS(1,IUN).EQ.0)GOTO125	003650
ICT(2)=NUMB(IUN*22)	003660
IF(IUN.GT.5)ICT(2)=NUMB((IUN-5)*22)	003670
PRINT ICT,STATS(3,IUN)	003680
126 CONTINUE	003690
PRINT 301	003700
DO 127 IUN=L1,L2	003710
IF(STATS(1,IUN).EQ.0)GOTO127	003720
ISECT(2)=NUMB(IUN*22)	003730
IF(IUN.GT.5)ISECT(2)=NUMB((IUN-5)*22)	003740
PRINT ISECT,STATS(4,IUN)	003750
127 CONTINUE	003760
PRINT 301	003770
PRINT 301	003780
IFQ(2)=NUMB(0)	003790
IF(JFLAG.NE.1)PRINT IFQ	003800
DO 190 IUN=L1,L2	003810
IF(STATS(1,IUN).EQ.0)GOTO143	003820
IFQ(2)=NUMB(IUN*22)	003830
IF(IUN.GT.5)IFQ(2)=NUMB((IUN-5)*22)	003840
PRINT IFQ	003850
190 CONTINUE	003860
PRINT 301	003870
PRINT 301	003880
KNT=1	003890
DO 521 I=1,10	003900
IF(JFLAG.EQ.1.AND.IBLK.EQ.0)GOTO622	003910
IF(JFLAG.EQ.1)GOTO140	003920

	IF (CUMPAY(I).EQ.0) GOTO 620	113930
	IPRNT(2)=NUMB(I)	113940
	PRINT IPRNT,I,CUMPAY(I)	113950
141	ISLNK=0	113960
	IF (KNT.GT.25) GOTO 620	113970
	DO 150 IUN=L1,L2	113980
	IF (STATS(4+KNT,IUN).EQ.0) GOTO 611	113990
	IPRNT(2)=NUMB(IUN+22)	114000
	IF (IUN.GT.5) IPRNT(2)=NUMB((IUN-5)*22)	114010
	WPN=STATS(4+KNT,IUN)-IFIX(STATS(4+KNT,IUN)/100.)*100.	114020
	REMAIN=IFIX(STATS(4+KNT,IUN)/100.)/10.	114030
	PRINT IPRNT,IWPN,REMAIN	114040
141	IF (STATS(4+KNT,IUN).EQ.0) ISLNK=ISLNK+1	114050
150	CONTINUE	114060
	PRINT *01	114070
	IF (ISLNK.NE.5) KNT=KNT+1	114080
	IF (JFLAG.EQ.1.AND.ISLNK.NE.5) GOTO 610	114090
620	CONTINUE	114100
	ISLNK=0	114110
123	JFLAG=JFLAG+1	114120
	IF (JFLAG.EQ.1) GOTO 622	114130
	PRINT *03	114140
132	DO 131 IUN=1,10	114150
	DO 131 IWPN=1,30	114160
	STATS(IWPN,IUN)=0.	114170
131	CONTINUE	114180
	JFLAG=0	114190
303	FORMAT(' ',132(' '))	114200
	IF (N.EQ.100) GOTO 620	114210
702	IF (ARRAY(I).NE.AMINUNIT) 0	114220
	PLRCEFF=0CEFF+RTEFF=1.	114230
	DO 614 I=1,30	114240
614	CUMPAY(I)=0.	114250
240	NUNIT=NUNIT+1	114260
	STATS(1,NUNIT)=ARRAY(2)	114270
	STATS(3,NUNIT)=ARRAY(5)	114280
	STATS(4,NUNIT)=ARRAY(4)	114290
	CEFF=0.	114300
	WPN=0	114310
	DO 267 I=1,30	114320
	IF (ARRAY(I).EQ.0) GO TO 253	114330
	ID0=I-1	114340
	WPN=WPN+1	114350
	CEFF=(CEFF+ARRAY(I)*PPS(ID0,JJ)	114360
	0CEFF=0CEFF+ARRAY(I)*PPS(ID0,JJ)	114370
	STATS(WPN,NUNIT)=ID0+IFIX(ARRAY(I)*100.001)*100.	114380
	IF (WPN.GT.30) PRINT*, "THERE ARE MORE WEAPONS THAN STATS ARRAY ALLOC"	114390
	145"	114400

	SUMDAY(T-1)=SUMDAY(T-1)+ADPAY(I)	004410
243	CONTINUE	004420
	A=ADPAY(1)	004430
	UEFF=999.	004440
	IF(ADPAY(5).GE.1)UEFF=OFFF/ADPAY(5)*100.	004450
	STATS(2,NUNIT)=UEFF	004460
	DTUEFF=DTUEFF+ADPAY(5)	004470
	GO TO 252	004480
254	CALL CLOSE4(LFTT)	004490
	GO TO 1	004500
9900	PRINT 9900	004510
9900	FORMAT(1Y,"ALL DONE***JOB HAS ENDED")	004520
9935	FORMAT(" EFFECTIVENESS OF ".410," = ",F4.0)	004530
9999	STOP	004540
	END	004550
	FUNCTION NUMF(TX)	004560
	I1=(TX-100*I1)/10	004570
	I2=(TX-100*I1)/10	004580
	I3=(TX-100*I1-10*I2)	004590
	NUMF=I1+I2+I3	004600
	DO 120 TT=1,9	004610
	INUMF=SHIFT(INUMF,6)	004620
	IF(TT.EQ.1)INUMF=09(INUMF,I2+333)	004630
	IF(TT.EQ.2)INUMF=09(INUMF,I3+336)	004640
	IF(TT.GT.2)INUMF=09(INUMF,558)	004650
150	CONTINUE	004660
	NUMF=INUMF	004670
	RETURN	004680
	END	004690

```

PROGRAM SUPER (INPUT=64, TAPE51=64, ANSWER=64, STATS=64, TAPE6=STATS, 000110
1 FIRE=64, TAPE10=FIRE, OUTPUT=TAPE51, DEBUG=OUTPUT, TAPE50=64, QA=64, 000120
2 TAPE52=QA, TAPE61=INPUT, REVIEW=64, TAPE62=REVIEW, 000130
1 TAPE99=64, LOGDAT, TAPE7=LOGDAT, ELEMENT=64, TAPE98=ELEMNT, 000140
1 TAPE5=ANSWER, CLCATA=64, TAPE3=CLCATA, TAPE70=64, 000150
5 TAPE71=512, TAPE72=512) 000152
C 000154
C JIFFY SYSTEM FILE: 000154
C 000158
C 000160
C LOCAL PROGRAM 000162
C NAME NAME DESCRIPTION 000164
C 1. ANSWER TAPE5 ANSWER FILE OUTPUT (RUN TYPE = 4) 000166
C 2. CLCATA TAPE3 CLASSIFIED DATA BASE 000170
C 3. ELEMENT TAPE38 ELEMENT ARRAY FILE USED BY DECISION POINT 1000172
C 4. FIRE TAPE10 CLASSIFIED EXPECTED NO. OF FIRINGS 000174
C 5. INPUT TAPE61 GAMER INPUT 000175
C 6. LOGDAT TAPE7 NOT USED AT THIS TIME 000179
C 7. QA TAPE52 QUESTION/ANSWER OUTPUT FILE COMBINED WITH 000180
C STATS FILE 000182
C 8. REVIEW TAPE62 Q/A INPUT FILE FROM PRECEDING GAME FOR 000184
C REVIEW OPTION 000186
C 9. STATS TAPE6 STATISTICAL SUMMARY 000194
C 10. TAPE50 TERMINAL OUTPUT FILE 000190
C 11. TAPE51 OUTPUT INTERMEDIATE OUTPUT FILE: PRODUCES ANSWER 000192
C FILE, QA FILE, AND TAPE50 000194
C 12. TAPE70 POSTPROCESSOR SCRATCH FILE 000196
C 13. TAPE71 SORT/MERGE SCRATCH FILE 000198
C 14. TAPE72 SORT/MERGE SCRATCH FILE 000199
C 15. TAPE99 SCRATCH FILE 000200
C 000201
COMMON IA, IO, IP, IENGAG, ITERN, IVIS, IMOUNT, MINES, CFPS, FSFR, FR, 000202
1 ATIME, IFIRST, IRUN, NORO, FMAS, IYAKV, 000203
2 SF(2), FSSF(2), PACK(2), 000204
3 ELMT(80,2), ALOSS(66,66), SHOTS(55,2), CKILL(53,2), CHOTEV(55,2) 000205
COMMON/REED/JDAY1, XINX(4), ICARD(2), IAPCR 000206
COMMON/DATA/FPS(80,2), CREWS(53,2), APOS(12), JPOST(6), ITHETM(21,2), 000210
1 PSN(6,2,2), PLT(15), KEY(171) 000220
COMMON/ONE/LFIT(35), ARRAY(90), IYBUF(1024), C(81,2), AC, 000223
1 ASCENE, ASECT, YSECT 000224
COMMON/TWO/IFIT(35), BFRAY(46), YVBUF(1024) 000225
COMMON/THREE/IMST(35), AM(90), YBUF(1024) 000226
COMMON/EM/POCGR(2,4), ICUNT(8), ISMOKE(2), PCMO(2), PCSCH(2), AND(2) 000270
1, NMU(2) 000280
COMMON/FPS/40FPS(2), AMFPS(2), 40FPS(2) 000290
COMMON/OUTPT/CCI(2), ICUT, KINC(2), TOTAL(3,2,2), BEGIN(55,2), NC21, 000300
1 NC22, CATLOS(65,13,2), CATSUM(11,2), TYPE(5), BRIG(10,1), BRIG 000310
COMMON/REVIEW/IGET1, GETS, IREV(1,10), IDEX, IREV, IYAX, ISPEED 000320

```

1	1JON(100),1JN,1JON,113(13),1EDIT	000330
C		000340
	CALL FILEIS(1FIT,3LLFN,6LTAPES,2LKA,ARRAY,2LPM,1LR,	000350
	13LPS,1024,3LFW9,MYBUF)	000360
	CALL FILEIS(1FIT,3LLFN,5LTAPES,2LKA,3CPAY,2LPM,1LR,	000370
	13LPS,1024,3LFW9,MYBUF)	000380
	CALL FILEIS(1HIST,3LLFN,5LTAPES,2LKA,AM,2LPM,1LR,	000390
	13LPS,1024,3LFW9,IYBUF)	000400
C.....	TERMINAL OUTPUT IS NOW TAPE5C	000410
	CALL DISCON(6LOUTPUT)	000420
	REWINO 51	000430
	REWINO 52	000440
	IMAKV=66	000450
C		000460
	KIND(1)="BLUE"	000470
	KIND(2)="RED"	000480
	PRINT 100	000490
100	FORMAT("1")	000500
110	FORMAT(1A10)	000510
	CALL INL	000520
	IFIRST=0	000530
	IXNAV="N"	000540
	JRUN=3	000550
	JRUN=5	000560
	PRINT *, " C A C C A J I F F Y W A R G A M E "	000570
	WRITE(52,115)DATE(AA),TIME(AA)	000580
115	FORMAT(1X,"DATE = ",A10,5X,"TIME = ",A10)	000590
	GO TO 150	000600
120	PRINT *, " SPECIFY PURPOSE OF THIS RUN "	000610
	CALL REED4	000620
	IRUN=IX(1)	000630
	IF(IRUN.EQ.1.OR.IRUN.EQ.4)WRITE(5,110)IXNAV	000640
	IF(IRUN.EQ.1)WRITE(5,*)IRUN	000650
	IF(IRUN.EQ.4)WRITE(5,*)JRLNS	000660
	BACKSPACE 52	000670
	WRITE(52,125)	000680
125	FORMAT(3X,"4")	000690
	IF(IRUN.GE.1.AND.IRUN.LE.5)GO TO 150	000700
	IF(IRUN.EQ.999)GO TO 133	000710
	PRINT 190	000720
130	PRINT *, " "	000730
	PRINT *, " ENTER 1 = MAKE MODIFIED ANSWER FILE (OBSOLETE)"	000740
	PRINT *, " ENTER 2 = TO GET OUTPUT OF RESULTS INTERACTIVELY"	000750
	PRINT *, " ENTER 3 = GET MODIFIED BATCH OUTPUT (OBSOLETE)"	000760
	PRINT *, " ENTER 4 = TO GET TERMINAL OUTPUT AND MAKE ANSWER FILE"	000770
	PRINT *, " ENTER 5 = GET BATCH OUTPUT"	000780
	GO TO 120	000790
140	CALL FORCE	000800

GO TO 190	000410
150 PRINT *, "DO YOU WISH TO SEE INSTRUCTIONS? (YES/NO)"	000420
CALL REECA(INX)	000430
IF(INX.EQ."N")GO TO 120	000440
GO TO 120	000450
160 IF(IFIRST.GE.0)GO TO 165	000460
PRINT *, "DECISION POINT 5 ONLY"	000470
GO TO 170	000480
165 CALL RQFA	000490
GO TO 130	000500
170 PRINT *, "ENTER 1 TO LOAD FORCES INTO A SECTION"	000510
PRINT *, " 2 TO CALCULATE RATE-OF-ADVANCE"	000520
PRINT *, " 3 TO ASSESS COMBAT"	000530
PRINT *, " 4 TO APPORTION CBT LOSSES TO UNITS"	000540
PRINT *, " 5 TO DISPLAY BATTLE STATISTICS"	000550
PRINT *, " 6 TO DISPLAY WEAPON ARRAYS"	000560
PRINT *, " 7 TO ADD SRC'S TO TCE FILE"	000570
PRINT *, " 8 TO RESTART AT A PREVIOUSLY GAMED CG"	000580
PRINT *, " 9 TO END GAME AND/OR UPDATE HISTORY FILE"	000590
PRINT *, "10 TO RESET ELEMENT ARRAY FROM INPUT FILE"	000600
PRINT *, "12 TO RESET TERMINAL OUTPUT FILE"	000610
180 PRINT *, "??? D E C I S I O N P O I N T ???"	000620
CALL REECA	000630
INX=XINX(1)	000640
IF(INX.EQ.999)GO TO 170	000650
IF(INX.GE.1.AND.INX.LE.12)GO TO 230	000660
PRINT 190	000670
190 FORMAT(" INCORRECT RESPONSE - TRY AGAIN")	000680
GO TO 170	000690
200 GO TO (140,160,210,530,470,480,510,520,570,720,730,740).INX	000700
210 IF(IFIRST.EQ.1)GO TO 220	000710
PRINT *, "RATE-OF-ADVANCE MUST BE CALCULATED BEFORE ASSESSMENTS ARE	000720
K ENTERED"	000730
GO TO 170	000740
220 CALL SUP-ES	000750
GO TO 340	000760
230 DO 240 J=1,2	000770
DO 240 I=43,54	000780
IF(ELMT(I,J).GT.0.)GO TO 250	000790
240 CONTINUE	000800
GO TO 260	000810
250 CALL CANNON	000820
260 IF(MINES.NE.1)GO TO 270	000830
CALL MINE	000840
270 DO 280 J=1,2	000850
DO 280 I=11,30	000860
IF(ELMT(I,J).GT.0.)GO TO 290	000870
280 CONTINUE	000880

GO TO 330	001290
290 CALL TANK	001300
300 (FIELDT(3,1),LE,0,OR,ELMT(3,2),LE,1)GO TO 310	001310
CALL INFANT	001320
310 CONTINUE	001330
DO 320 J=1,2	001340
DO 320 I=59,65	001350
IF(ELMT(I,J).GT.0)GO TO 330	001360
320 CONTINUE	001370
GO TO 130	001380
330 CALL A1AC	001390
GO TO 180	001400
340 PRINT *, "DO YOU WISH TO INCLUDE ANY LOSSES DUE TO TACAIR FOR APPRO-	001410
IMATIONMENT?"	001420
CALL REECA(INX)	001430
IF(INX.EQ."N")GO TO 230	001440
350 CONTINUE	001450
J=1	001460
360 PRINT 370,KIND(J)	001470
370 FORMAT(" ANY ",A4," LOSSES? ")	001480
CALL REECA(INX)	001490
IF(INX.EQ."N")GO TO 460	001500
380 PRINT *, "ENTER WPN ID, # LOST (1,3 WHEN CODE)-- "	001510
390 CALL REECA	001520
IWP=XINX(1)	001530
XLDS=XINX(2)	001540
IF(IWP.EQ.0)GO TO 460	001550
IF(XLOS.GT.ELMT(IWP,J))GO TO 420	001560
IF(XLOS.LT.1)GO TO 450	001570
ELMT(IWP,J)=ELMT(IWP,J)-XLOS	001580
ALOSS(66,IWP)=ALOSS(66,IWP)+FIX(XLOS*10.)+PACK(J)	001590
C AUTOMATIC ASSESSMENT OF CREW LOSSES TO TACAIR	001600
IF(IWP.LE.12,OR,IWP.GE.66)GO TO 410	001610
CLOS=XLDS*CREWS(IWP-12,J)	001620
IS=0	001630
IF(CLOS.GT.ELMT(2,J))IS =1	001640
IF(IS.EQ.1)PRINT 400,ELMT(2,J)	001650
400 FORMAT(" ONLY ",F4.0," OF CREW PERSONAL REMAIN IN SEC'D. - ")	001660
* " CREW LOSSES NOT SUBTRACTED"	001670
* " CREW PERSONAL UPTO NUMBER IN SEC'D CAN BE REMOVED SEPARATELY"	001680
IF(IS.EQ.1)ELMT(2,J)=ELMT(2,J)-CLOS	001690
IF(IS.EQ.0)ALOSS(66,2)=	001700
* ALOSS(66,2)+FIX(CLOS*10.)+PACK(J)	001710
C END AUTO OF CREW	001720
410 PRINT *, "NEXT-- "	001730
GO TO 390	001740
420 PRINT 430,ELMT(IWP,J),IWP,KIND(J)	001750
430 FORMAT(" OVERKILL!!! ONLY ",F4.0," OF WEAPON ",I3," REMAIN IN "	001760

1HE "144" FORCE FOR THIS SECTOR."	111770
440 PRINT *, "LAST ENTRY IGNORED"	111780
GO TO 410	111790
450 PRINT *, "A LOST MUST BE ENTERED AS A POSITIVE NUMBER."	111800
GO TO 440	111810
460 IF(J.EQ.2)GO TO 230	111820
J=2	111830
GO TO 360	111840
C	111850
470 IF(IRUN.EQ.1)PRINT *, " SECTOR STATS PRINTED HERE."	111860
IF(IPUN.NE.1)CALL POST	111870
IF(IPUN.EQ.1.OR.IRUN.EQ.3)GO TO 180	111880
PRINT *, "RETRIEVE ANY LOGDAT FILES?"	111890
475 CALL PEECA(IYN)	111900
IF(IYN.EQ."N")GO TO 180	111910
XCI=ACI	111920
PRINT *, "ENTER CI NAME- "	111930
CALL REECB(ACI)	111940
C	111950
CALL ARMOUT	111960
ACI=XCI	111970
PRINT *, "ANOTHER LOGDAT FILE?"	111980
GO TO 475	111990
C	112000
480 PRINT *, " FORCE STRUCTURES"	112010
PRINT *, "ELEMENT BLUE RED"	112020
GO 500 I=1,30	112030
IF(ELMT(I,1).EQ.0..AND.ELMT(I,2).EQ.0.)GO TO 570	112040
PRINT 490,I,ELMT(I,1),ELMT(I,2)	112050
490 FORMAT(" ",I2,"12.0,F4.0)	112060
500 CONTINUE	112070
GO TO 190	112080
C	112090
510 CALL BUILD	112100
GO TO 180	112110
C	112120
CALL SUBROUTINE RESTART TO RELOAD PREVIOUS FORCE FILE	112130
C	112140
520 CALL RESTART	112150
GO TO 180	112160
C	112170
C	112180
C	112190
530 CALL ARMOUT	112200
530 CALL APPORT	112210
GO TO 180	112220
C	112230
540 FORMAT(1A1)	112240
550 FORMAT(" ",1A1)	112250
560 FORMAT(" INCORRECT--MUST BE Y OR N--TRY AGAIN ")	112260

570 PRINT 590,ACI	002250
580 FORMAT(" HAS THE LAST SECTOR BEEN GAMED FOR CI ",I410,"?")	002260
CALL REECA(INX)	002270
IF(INX.EQ."N")GO TO 590	002280
I3PI=0	002290
582 I3RI=I3RI+1	002300
IF(I3PI.GT.10)GO TO 590	002310
PRINT *, "ENTER BRIGADE NAME (END IF DONE) --"	002320
CALL FEETB(I3RI)	002330
IF(I3RI.EQ."END")GO TO 590	002340
BRIG(I3PI,1)=I3RI	002350
NBRIG=I3PI	002360
I3ECT=2	002370
PRINT *, "ENTER SECTOR IN BRIGADE (0 IF DONE) --"	002380
584 CALL REECA	002390
IS=XINX(1)	002400
IF(IS.EQ.0)GO TO 582	002410
BRIG(I3RI,I3ECT)=IS	002420
SECT=I3ECT+1	002430
IF(I3ECT.GT.10)GO TO 582	002440
PRINT *, "NEXT --"	002450
GO TO 584	002460
590 PRINT 600,ACI	002470
600 FORMAT(" DO YOU WANT CI STATS OUTPUT FOR CI ",I410,"? ")	002480
CALL REECA(IYN)	002490
IF(IYN.EQ."N")GO TO 630	002500
610 IF(IRUN.NE.1)GO TO 620	002510
PRINT *, " CUMULATIVE STATS PRINTED HERE."	002520
GO TO 630	002530
620 IRUN=IRUN+5	002540
A3ECT=0.	002550
CALL POST	002560
IRUN=IRUN-5	002570
630 PRINT 640,ACI	002580
640 FORMAT(" DO YOU WANT GAME STATS OUTPUT THROUGH CI ",I410,"? ")	002590
CALL FEEDA(IYN)	002600
IF(IYN.EQ."N")GO TO 670	002610
650 IF(IRUN.NE.1)GO TO 660	002620
PRINT *, "GAME OUTPUT GENERATED HERE."	002630
GO TO 670	002640
660 IRUN=IRUN+5	002650
XCI=ACI	002660
ACI=0.	002670
CALL POST	002680
ACI=XCI	002690
IRUN=IRUN-5	002700
670 IF(IPIUN.EQ.1)GO TO 590	002710
	002720

	IF(ACI.EQ.0.)PRINT 580	002730
680	FORMAT(" CI NAME NOT SET--HISTORY AND FORCE FILES NOT UPDATED.")	002740
	IF(ACI.EQ.3.)GO TO 690	002750
	CALL ENCCI	002760
C	CALL ARMCUT	002770
C		002780
690	IF(IRUN.EQ.1.OR.IRUN.EQ.4)PRINT *, " "ANSWER" FILE HAS BEEN CREATED."	002790
	IF(IRUN.EQ.4)WRITE(5,700)ACI,ASEC,DATE(AA),TIME(AA)	002800
700	FORMAT(1X,"CI ",410,5X,"SEC"OR ",F6.0,2(5X,410))	002810
C		002820
710	STOP 1	002830
C		002840
C	DECISION POINT 10	002850
C		002860
720	CONTINUE	002870
	CALL RESET	002880
	GO TO 180	002890
C		002900
C	DECISION POINT 11	002910
C		002920
730	CONTINUE	002930
	IF(IRUN.EQ.3.OR.IRUN.EQ.5)GO TO 180	002940
	IF(IREVF.NE.0)GO TO 180	002950
C	CALL REVIEW(INX)	002960
	GO TO 180	002970
C		002980
C	DECISION POINT 12	002990
C		003000
740	CONTINUE	003010
	PRINT *, "ENTER OPTION - 1 TO CONNECT; 2 TO DISCONT "	003020
	CALL REED4	003030
	INX=XINX(1)	003040
	IF(INX.EQ.1.OR.INX.EQ.2) GO TO 750	003050
	PRINT 190	003060
	GO TO 740	003070
750	GO TO (760,770),INX	003080
760	CALL CONNCT(50)	003090
	GO TO 180	003100
770	CALL DISCON(50)	003110
	GO TO 180	003120
	END	003130
		003140

SUBROUTINE SHOWIT	003390
COMMON IA, IC, IF, IENGAG, ITERKN, IVIS, IMOUNT, MINES, CPFR, PSFR, FR,	003391
1 ATIME, IFIRST, JGUN, NORD, FMASS, MAXKV,	003401
2 SF(2), FSSF(2), PACK(2),	003402
3 ELAT(80,2), ALOSS(66,66), SHOTS(55,2), CKILL(53,2), SHOTCV(55,2)	003410
COMMON/REED/JOAY1, XINX(4), ICARD(20), IAFOR	003440
COMMON/REVIEW/IGET1, IGETS, IPEV(1,10), IDEX, IREVF, IMAX, ISPEED	003450
1 IJON(100), IJN, IJOHN, I13(13), IEDIT	003460
DIMENSION IZ(8)	003470
C	003480
ISHOW=1	003490
GO TO 50	003500
ENTRY SHOWME	003510
ISHOW=2	003520
GO TO 50	003530
ENTRY SHOW3	003540
ISHOW=3	003550
50 CONTINUE	003560
REWIND 51	003570
100 READ(51,140) (I13(I), I=1,13)	003580
IF (EOF(51).EQ.0) GO TO 130	003590
DO 110 J=1,13	003600
JJ=14-J	003610
IF (I13(JJ).EQ.0) GO TO 110	003620
DO 105 I=1,9	003630
IZ(I)=I13(I)	003640
105 CONTINUE	003650
GO TO 120	003660
110 CONTINUE	003670
120 CONTINUE	003680
IF (ISHOW.EQ.1) WRITE(50,140) (I13(I), I=1, JJ)	003690
GO TO 130	003700
130 IF (ISHOW.EQ.2) GO TO 135	003710
132 IEDIT=IEDIT+1	003720
IF (JJ.EQ.1.AND.I13(1).EQ.0) GO TO 134	003730
WRITE(52,150) (I13(I), I=1,9), IEDIT	003740
IF (ISHOW.EQ.3) WRITE(50,140) (I13(I), I=1, JJ)	003750
GO TO 136	003760
134 WRITE(52,150) IZ, IEDIT	003770
IF (ISHOW.EQ.3) WRITE(50,140) IZ	003780
GO TO 146	003790
135 CONTINUE	003800
IF (JJ.EQ.1.AND.I13(1).EQ.0) WRITE(50,140) IZ	003810
WRITE(50,140) (I13(I), I=1, JJ)	003820
146 I13(1)=IZ(1)	003830
I13(2)=IZ(2)	003840
I13(3)=IZ(3)	003850
136 CONTINUE	003860

BEHIND 51
RETURN

003270
003840
003890
003900
003910
003920

C
140 FORMAT(13A10)
150 FORMAT(9A10,I6)
END

AD-AU94 783

ARMY COMBINED ARMS STUDIES AND ANALYSIS ACTIVITY FOR--ETC F/G 15/7
CACDA JIFFY III WAR GAME. VOLUME V. PROGRAMMER'S MANUAL.(U)

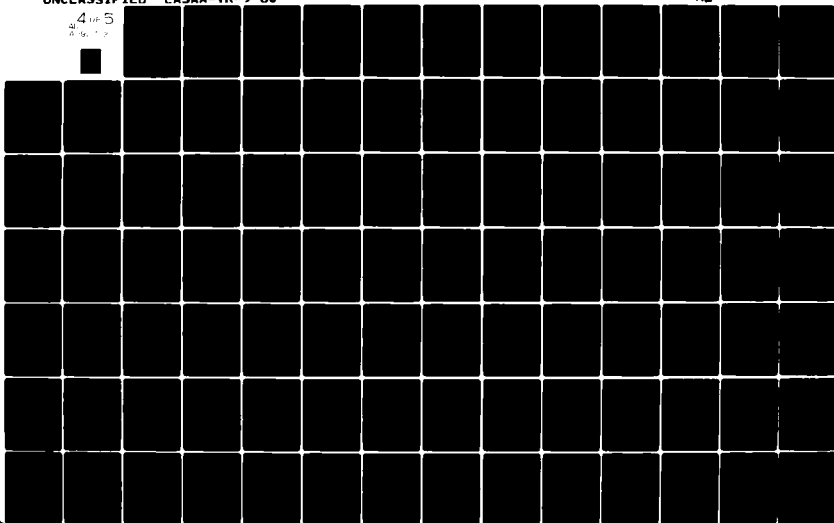
OCT 80 C L PAO

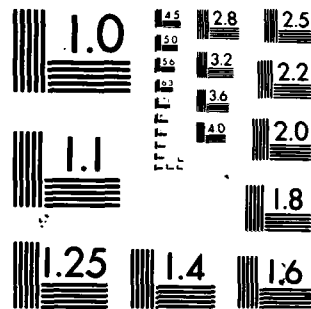
CASAA-TR-9-80

UNCLASSIFIED

NL

4 of 5
00000000





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SUBROUTINE REEPA(INX)	003940
COMMON /A,17,18,19,ENGAB,ITERM,IVIS,INCUN,INFS,OFF,PSPP,PD,	003950
1 ATIME,IFIRST,IRUN,NORD,PMAS,IMAXV,	003960
2 SF(2),FSSF(2),PACK(2),	003970
3 ELMT(40,2),ALOSS(66,66),SHOTS(55,2),CKILL(53,2),SHOTEV(55,2)	003980
COMMON/REED/JCAY1,XINX(4),ICARD(20),IARMOR	003990
COMMON/REVIEW/IGET1,IGETS,IREFV(1,10),ICEX,IREF,IMAX,ISPEF	004000
1,IJON(100),IJN,IJOHN,I13(13),IECIT	004010
C	004020
100 CALL SHOWIT	004030
110 IF(IREFV.EQ.0)GO TO 120	004040
112 CALL REVIEW(INX)	004050
IF(IREFV.EQ.0)GO TO 120	004060
GO TO 130	004070
120 IF(IJOHN.EQ.0)GO TO 125	004080
122 CALL JOHN(INX)	004090
IF(IJOHN.EQ.0)GO TO 125	004100
GO TO 130	004110
125 READ 240,INX	004120
IF(EOF(61).NE.0)CALL SHORT,RETURNS(125)	004130
IF(INX.EQ."9999")GO TO 122	004140
IF(INX.EQ."9998")GO TO 112	004150
130 CALL TERM(IRUN,INX),RETURNS(113)	004160
ENCODE(10,200,INX)INX	004170
IF(IRUN.EQ.4)WRITE(5,200)INX	004180
140 IF(IRUN.EQ.5)WRITE(50,220)INX	004190
WRITE(52,230)INX	004200
IF(INX.EQ."Y".OR.INX.EQ."N")RETURN	004210
PRINT 210	004220
GO TO 100	004230
C	004240
ENTRY REE09	004250
150 CALL SHOWIT	004260
160 IF(IREFV.EQ.0)GO TO 170	004270
162 CALL REVIEW(INX)	004280
IF(IREFV.EQ.0)GO TO 170	004290
GO TO 180	004300
170 IF(IJOHN.EQ.0)GO TO 175	004310
172 CALL JOHN(INX)	004320
IF(IJOHN.EQ.0)GO TO 175	004330
GO TO 180	004340
175 READ 240,INX	004350
IF(EOF(61).NE.0)CALL SHORT,RETURNS(175)	004360
IF(INX.EQ."9999")GO TO 172	004370
IF(INX.EQ."9998")GO TO 162	004380
180 CALL TERM(IRUN,INX),RETURNS(160)	004390
IF(IRUN.EQ.4)WRITE(5,240)INX	004400
190 IF(IRUN.EQ.5)WRITE(50,250)INX	004410

WRITE(52,260)INX
RETURN

210 FORMAT(1A1)
210 FORMAT(1Y,"INCORRECT--MUST BE Y OR N--TRY AGAIN")
220 FORMAT(1Y,1A1)
230 FORMAT(3Y,1A1)
240 FORMAT(1A10)
250 FORMAT(1X,1A10)
260 FORMAT(3Y,1A10)
END

104420
004430
104440
004450
104460
004470
104480
004490
104500
004510
104520

SUBROUTINE REED4	004540
COMMON IA,IO,IP,ENGAG,ITERRN,IVIS,IMCUNT,MINES,CEFF,FSFAC,F02,	004550
1 ATIME,IFIRST,IRUN,N070,FNASS,IMAYKV,	004560
2 SF(2),FSSF(2),PACK(2),	004570
3 ELMT(30,2),ALOSS(66,66),SHOTS(55,2),CKZLL(53,2),SHOTSV(55,2)	004580
COMMON/REED/JDAY1,XINX(4),ICARD(20),IAFPCR	004590
COMMON/REVIEW/IGET1,IGETS,IDEV(1,1),ICEX,IREVF,IMAX,ISPEED	004600
1,IJON(100),IJN,IJOHN,I13(13),IEDIT	004610
C	004620
100 CALL SHOOT	004630
110 IF(IPEVF.EQ.1)GO TO 130	004640
112 CALL RE/DEM(INV)	004650
IF(IPEVF.EQ.1)GO TO 130	004660
DECODE(10,150,INV)(ICARD(I1),I1=1,10)	004670
DO 120 I=11,20	004680
120 ICARD(I1)=14	004690
GO TO 140	004700
130 IF(IJOHN.EQ.0)GO TO 135	004710
132 CALL IJRN(INV)	004720
IF(IJOHN.EQ.0)GO TO 135	004730
DECODE(10,150,INV)(ICARD(I1),I1=1,10)	004740
DO 137 I1=11,20	004750
137 ICARD(I1)=14	004760
GO TO 140	004770
135 READ 150,ICARD	004780
IF(EOF(61).NE.0)CALL SHOOT,RETURNS(135)	004790
IF(ICARD(1).NE."9".OR.ICARD(2).NE."9".OR.ICARD(3).NE."9")GO TO 134	004800
IF(ICARD(4).EQ."9")GO TO 132	004810
IF(ICARD(4).EQ."9")GO TO 112	004820
136 ENCODE(10,150,INV)(ICARD(I1),I1=1,10)	004830
140 CALL IFS(IRUN,INV),RETURNS(141)	004840
IF(IRUN.EQ.4)WRITE(5,150)ICARD	004850
IF(IRUN.EQ.5)WRITE(50,150)ICARD	004860
IGET1=4	004870
CALL GET4(ICARD,XINX)	004880
WRITE(52,170)ICARD	004890
RETURN	004900
C	004910
150 FORMAT(20A1)	004920
160 FORMAT(1X,23A1)	004930
170 FORMAT(3X,23A1)	004940
END	004950

SUBROUTINE GET4(LINE,XNOS)	115075
COMMON /A,IC,IP,ENGAG,STEPPN,IVCS,INCUN,NTNES,CFP,PSFF,PR,	115080
1 ATIME,PIST,MIN,AD,FM,ASS,IM4KV,	115090
2 SP(2),PSSF(2),PACK(2),	115100
3 EL17(A0,2),A,SS(65,66),SMUTS(55,2),CKZ(53,2),SMUTV(55,2)	115110
COMMON/REED/JDAY1,XINX(4),ICAP(25),TAPPC-	115120
COMMON/FEVTEW/IGET1,IGET2,FEV(1,10),TDEX,TFF/5,TAX,IGEEC	115130
1 IJON(100),IJ4,IJOM,I13(13),IEP	115140
DI=NGION LINE(20),XNOS(4),NCS(10)	115150
DAT NOS/"0","1","2","3","4","5","6","7","8","9"/	115160
XNOS(1)=-1	115170
IOUT=0	115180
IGETS=1	115190
100 IGT=0	115200
XNUMB=1	115210
IMULT=1	115220
IMULT2=10	115230
IFACT=1	115240
INEG=0	115250
110 DO 120 ICO=1,13	115260
IF(LINE(IGETS).EQ.NOS(ICO))GO TO 140	115270
120 CONTINUE	115280
IF(LINE(IGETS).EQ."")GO TO 150	115290
IF(LINE(IGETS).EQ."")GO TO 130	115300
GO TO 180	115310
130 INEG=1	115320
GO TO 170	115330
140 XNUMB=(XNUMB*IMULT*IMULT2+ICO-1)/IMULT	115340
GO TO 160	115350
150 IFACT=10	115360
IMULT2=1	115370
160 IMULT=IMULT*IFACT	115380
IGT=1	115390
170 IGETS=IGETS+1	115400
IF(IGETS.LE.72)GO TO 110	115410
180 IF(IGT.NE.1)GO TO 190	115420
IOUT=IOUT+1	115430
IF(INEG.EQ.1)XNUMB=-XNUMB	115440
XNOS(IOUT)=XNUMB	115450
190 DO 200 ICG=1,4	115460
IGETS=IGETS+1	115470
IF(IGETS.GT.72)RETURN	115480
IF(LINE(IGETS).EQ."")GO TO 200	115490
IF(LINE(IGETS).NE."")GO TO 210	115500
200 CONTINUE	115510
RETURN	115520
210 IF(IGT.EQ.IGET1)RETURN	115530
GO TO 130	115540
END	115550

	SUBROUTINE JOHN(INY)	105470
	COMMON IA,IS,IP,IEGAG,ITERRN,IVIS,INCUN,MINES,CPPR,FSPP,FPF,	105480
	1 ATIME,IFIRST,IRUN,NQOC,FMASS,IMAXV,	105490
	2 SF(2),FSSF(2),PACK(2),	105500
	3 ELVT(30,2),ALOSS(66,66),SHOTS(55,2),CKILL(53,2),SHOTSV(35,2)	105510
	COMMON/REED/JDAY1,XINX(4),ICARD(20),IARMCA	105520
	COMMON/REVIEW/IGET1,IGETS,IREV(1,10),IREX,IREVF,MAX,ISPEED	105530
	1 IJON(100),IJN,IJOHN,I13(13),IEPIT	105540
C		105550
	IF(IJOHN.EQ.0)GO TO 140	105560
	50 IF(IJN.NE.1)ILINIGO TO 150	105570
	INY=IJON(IJN)	105580
	IJN=INY+1	105590
	IF(INY.EQ."")GO TO 50	105600
	IF(INY.EQ."9999")GO TO 150	105610
	IJOHN=1	105620
	IF(ITERN.NE.3.AND.IPLAN.NE.5)WRITE(50,130)INY	105630
	130 FORMAT(1X,A10)	105640
	RETURN	105650
C		105660
	150 IJOHN=0	105670
	PRINT(50,*)"END OF BATCH MODE"	105680
	CALL SHOWME	105690
	RETURN	105700
C		105710
	180 IJN=1	105720
	ILIN=C	105730
	PRINT(50,*)"----- ENTER BATCH MODE -----"	105740
	PACT(50,*)	105750
	182 ILIN=ILIN+1	105760
	IF(ILIN.GT.99)WRITE(50,140)ILIN	105770
	183 FOR=ATIME,13,"")	105780
	READ 135,INX,INZ	105790
	185 FOR=ATIME(10,11)	105800
	IF(INZ.EQ."")GO TO 196	105810
	PRINT(50,*)"ANSWER TOO LONG---ENTER PLEASE "	105820
	PRINT(50,*)	105830
	ILIN=ILIN+1	105840
	GO TO 182	105850
	186 IJON(ILIN)=INX	105860
	IF(ILIN.EQ.1)GO TO 410	105870
	IF(INX.NE."9999")GO TO 142	105880
	IF(ILIN.EQ.1)GO TO 150	105890
C		105900
C	DOUBLECHECK IJON LIST BEFORE RETURNING	105910
C		105920
	410 CALL CKANS(ILIN)	105930
	GO TO 40	105940
C		105950
	END	105960

SUBROUTINE CKANS(ILIN)	115940
COMMON IA,IB,IP,IEGAG,ITE99N,IVIS,IMOUNT,MINES,CPRR,FSPR,PR,	115940
1 ATIME,IFIPST,IRUN,NOPD,FMASS,IMAXV,	116030
2 SF(2),FSSF(2),PACK(2),	116110
3 ELMT(80,2),ALOSS(66,66),SHOTS(55,2),CKILL(53,2),SMUTSV(55,2)	116020
COMMON/REED/JOAY1,XINX(4),ICARD(20),IARPOH	116130
COMMON/REYAEH/IGET1,IGETS,IREV(1,10),IDEX,IREVF,IMAX,ISPEED	116140
1 IJCN(100),IJN,IJOHN,I13(13),IJCIT	116150
C	116160
100 PRINT(51,*)"REVIEW ANSWER?"	116170
110 READ 250,INX	116180
IF(INX.EQ."Y".OR.INX.EQ."N")GO TO 120	116190
PRINT(51,240)	116200
GO TO 110	116210
120 IF(INX.EQ."N")GO TO 210	116220
IF(IJCN.GT.0)GO TO 125	116230
PRINT(51,*)"NONE"	116240
GO TO 135	116250
125 ISTACK=(ILIN-1)/4+1	116260
IF(ISTACK.LT.0)ISTACK=0	116270
DO 133 JJ=1,ISTACK	116280
IF(IJ.JT.ILIN)GO TO 135	116290
J2=JJ+3-ISTACK	116300
IF(IJ2.GT.ILIN)J2=ILIN	116310
WRITE(52,130)(J,IJOH(J),J=JJ,J2,ISTACK)	116320
133 CONTINUE	116330
130 FORMAT(1X,4(I3," ",A10,3X))	116340
135 CONTINUE	116350
WRITE(50,140)	116360
140 FORMAT(1X,"GO?"	116370
150 READ 250,INX	116380
IF(INX.EQ."Y".OR.INX.EQ."N")GO TO 160	116390
PRINT(50,240)	116400
GO TO 150	116410
160 IF(INX.EQ."Y")GO TO 230	116420
ICHANGE=0	116430
180 PRINT(50,*)"INPUT NO., ANSWER: 0 TO END "	116440
PRINT(50,*)	116450
190 READ 260,ICARD	116460
IGET1=1	116470
CALL GET4(ICARD,XINX)	116480
IF(XINX(1).EQ.0.)GO TO 220	116490
INSET=XINX(1)	116500
IF(INSET.LE.100)GO TO 210	116510
PRINT(51,*)"ANSWER NO. GT 100"	116520
GO TO 180	116530
210 ICHARGE=1	116540
IGET2=15-ICHARGE	116550

5-0335(1,250,2JON(INSRT)) (2040(2),I=10000,2012)	15-15
IF(INSRT.GT.0) ILIP=INSRT	15-17
GO TO 190	15-18
220 IF(1CHANGE,VE.3)GO TO 170	15-19
230 CALL SHOWNE	15-20
RETURN	15-21
C	15-22
240 FORMAT(1X,"INCOPRECT--MUST BE Y OR N--TRY AGAIN")	15-23
250 FORMAT(1A1)	15-24
260 FORMAT(20A1)	15-25
END	15-26

SUBROUTINE REVIEW(INY)	006590
COMMON IA,IO,IP,IENGAG,IERRN,I/IS,IMCUNT,MINES,CPPR,FSPP,FPR,	006590
1 ATIME,IFIRST,IBUN,NQRO,PHASS,IMAXV,	006590
2 SP(2),FSSF(2),PACK(2),	006590
3 ELMT(80,2),ALOSS(66,66),SMCTS(55,2),CKILL(53,2),SHOTS(55,2)	006590
COMMON/REED/JDAY1,XINX(4),ICARD(20),IARMC	006590
COMMON/REVIEW/IGET1,GETS,IPEV(1,10),ICEX,IREF,IPEV,IPAX,ISPEE	006590
1 ,IJON(100),IJN,IJON,I13(13),IEDIT	006590
C	006590
IF(IREF.NE.0)GO TO 100	006590
GO TO 130	006590
100 IF(IDEX.GT.IMAX)GO TO 140	006590
110 INY=IJON(IDEX)	006590
IDEX=IDEX+1	006590
IF(INY.EQ."")GO TO 110	006590
IF(INY.EQ."9998")GO TO 140	006590
IREF=1	006590
IF(IRUN.NE.3.AND.IRUN.NE.5)WRITE(50,130)INY	006590
130 FORMAT(1X,A10)	006590
RETURN	006590
C	006590
140 IREF=0	006590
CALL CONNEC(50)	006590
PRINT(50,*)"END OF REVIEW MODE"	006590
CALL SHOWS	006590
RETURN	006590
C	006590
180 IREF=1	006590
230 IPEV=C	006590
PRINT(50,*)"----- ENTER REVIEW MODE -----"	006590
PRINT(50,*)	006590
IFINO=0	006590
ISHMAX=1E6	006590
ISHMIN=1E6	006590
235 READ(62,240)(ICARD(J),J=1,9),IEDIT	006590
IF(EOF(62).NE.3)GO TO 355	006590
240 FORMAT(1A10,I6)	006590
250 JJ=9	006590
GO 260 J=1,7	006590
JJ=JJ-1	006590
IF(ICARD(JJ).EQ."")GO TO 260	006590
GO TO 270	006590
260 CONTINUE	006590
JJ=1	006590
270 CONTINUE	006590
IF(JJ.GT.2)GO TO 310	006590
RECDE(20,280,ICARD)ICOL1,ICOL4,ICOL14	006590
280 FORMAT(A3,A10,A7)	006590

IF(ICOL1.NE." ".OR.ICOL14.NE." ")GO TO 310	007360
IF(IFIND.EQ.0)GO TO 350	007370
IMAX=IMAX+1	007380
IJON(IMAX)=ICOL4	007390
WRITE(50,300)IMAX,(ICARD(I),I=1,JJ)	007400
300 FORMAT(1X,I3," ",2A10)	007410
GO TO 340	007420
310 CONTINUE	007430
IF(LEDT.LT.ISHMAX)GO TO 315	007440
PRINT(50,PRINTNO MATCH FOUND)	007450
IFIND=1	007460
ISMAX=1E6	007470
CALL HON(IMAX),RETURN(400)	007480
GO TO 317	007490
315 WRITE(50,20)JJGO TO 317	007500
IF(LEDT.LT.ISHMIN)ISHMIN=LEDT	007510
317 IF(IFIND.EQ.1)GO TO 319	007520
IF(ICARD(1).NE.I13(1).OR.ICARD(2).NE.I13(2).OR.	007530
ICARD(3).NE.I13(3))GO TO 319	007540
IFIND=1	007550
CALL HON(IMAX),RETURN(400)	007560
319 IF(IFIND.EQ.1)WRITE(50,240)(ICARD(I),I=1,JJ)	007570
GO TO 350	007580
340 IF(IMAX.GE.IABS(IMAX))GO TO 360	007590
IF(IMAX.GE.10)GO TO 360	007600
350 CONTINUE	007610
GO TO 235	007620
355 IF(IFIND.NE.0)GO TO 360	007630
ISHMAX=ISHMIN	007640
REWIND 62	007650
GO TO 235	007660
C	007670
C DOUBLECHECK IJON LIST BEFORE RETURNING	007680
C	007690
360 CONTINUE	007700
IF(IMAX.LT.0)CALL CONNEX(50)	007710
CALL CXANS(IMAX)	007720
GO TO 110	007730
C	007740
400 CONTINUE	007750
BACKSPACE 62	007760
GO TO 140	007770
C	007780
END	007790

[illegible]

SUBROUTINE SHORT, RETURNS (NORM)	007560
COMMON IA, IO, IP, IENGAG, ITEPPN, IVIS, INCUNT, NINES, CFBS, FSFBS, FPR,	007570
1 ATIME, IFIRST, IRUN, NOPD, FMASS, IMAXKV,	007580
2 SF(2), FSSF(2), PACK(2),	007590
3 ELVT(80,2), ALOSS(66,66), SHOTS(55,2), CKILL(55,2), SHOTS(55,2)	007600
C	007710
CALL CONNED(53)	007720
WRITE(50,103)	007730
100 FCPMAT(1X,"END OF ANSWER FILE - INPUT FROM TERMINAL ")	007740
CALL SHOWME	007750
REWINO 51	007760
CALL CONNED(61)	007770
IF (IRUN.EQ.3.GR.IRUN.EQ.5) IRUN=2	007780
RETURN NORM	007790
END	007800

	SUBROUTINE TEEM(IRUN,INX),RETURN(NGUT)	007420
C		007430
	IF(INX.NE."OFF")GO TO 100	007440
	PRINT(90,*)"DISCONNECT ANSWER FILE OUTPUT"	007450
	GO TO 500	007460
100	IF(INX.NE."OFF2")GO TO 200	007470
	IF(IRUN.EQ.5)WRITE(50,610)INX	007480
	PRINT(91,*)"OUTPUT DISCONNECTED"	007490
	CALL DISCON(5)	007500
	GO TO 510	007510
200	IF(INX.NE."ON")RETURN	007520
	IF(IRUN.EQ.5)WRITE(50,610)INX	007530
	CALL CONNCT(50)	007540
	PRINT(91,*)"CONNECT ANSWER FILE OUTPUT"	007550
	GO TO 510	007560
500	CONTINUE	007570
	IF(IRUN.EQ.5)WRITE(50,610)INX	007580
510	CONTINUE	007590
	IF(INX.EQ."OFF")INX="OFF2"	007600
	IF(IRUN.EQ.4.OR.IRUN.EQ.3)WRITE(5,600)INX	007610
	WRITE(52,620)INX	007620
	CALL SHOW3	007630
	RETURN NGUT	007640
C		007650
600	FORMAT(1A10)	007660
610	FORMAT(1X,1A10)	007670
620	FORMAT(3X,1A10)	007680
	END	007690

FUNCTION NUMB (IX)	008110
I1=X/100	008120
I2=(IX-100*I1)/10	008130
I3=(IX-100*I1-10*I2)	008140
NUMB=I1+333	008150
DO 100 IT=1,9	008160
NUMB=SHIFT(NUMB,6)	008170
IF(IT.EQ.1) NUMB=OR(NUMB,2+333)	008180
IF(IT.EQ.2) NUMB=OR(NUMB,3+333)	008190
IF(IT.EQ.3) NUMB=OR(NUMB,555)	008200
100 CONTINUE	008210
NUMB=NUMB	008220
RETURN	008230
END	008240

S. ARROUTINE: KSTOR(AH,IPUT)	108351
COMMON/ONE/IFC(17),ARRAY(9),MYBUF(1024),O(80,2),ICI.	108352
1 ASCE NE,SSCT	108353
COMMON/THREE/IMIST(35),AH(90),CYBUF(1024)	108354
CALL OPEN(IMIST,311-0,11)	108355
IF(IPUT.EQ.0) GO TO 120	108356
IPUT=0	108357
N1=4	108358
N2=90	108359
AH(1)=AH1	108360
AH(2)=ACT	108361
AH(3)=ASECT	108362
AH(4)=99999	108363
CALL GET(IMIST,AH,AH(1))	108364
IF(AH(4).NE.99999) IPUT=1	108365
IF(IPUT.EQ.0) GO TO 100	108366
IF(AH1.EQ."C1 AH") N1=1	108367
IF(AH1.NE."INPUT 36") GO TO 130	108368
IF(ARRAY(4).GT.0) N1=ARRAY(4)	108369
IF(ARRAY(5).GT.ARRAY(4)) N2=ARRAY(5)	108370
140 DO 110 I=N1,N2	108371
AH(I)=I.	108372
110 CONTINUE	108373
C	108374
C	108375
C *** PUT OR REPLC RECORD ON FILE ***	108376
120 CONTINUE	108377
IF(IPUT.EQ.0) CALL PUT(IMIST,AH,90,AH(1))	108378
IF(IPUT.EQ.1) CALL REPLC(IMIST,AH,90,AH(1))	108379
CALL CLOSE(IMIST)	108380
IPUT=1	108381
130 RETURN	108382
END	108383

SUBROUTINE ENCCI	10460
COMMON IA, ID, IP, IENGAG, ITERRN, IVIS, INCUNT, MINES, CPPE, FSPFF, FOF,	10461
1 AT, ME, IFIRST, IRUN, HORG, FMASS, IMAXV,	10462
2 SF(2), FSSF(2), PACK(2),	10463
3 ELMT(81,2), ALQSS(66,56), SHOTS(55,2), CKILL(57,2), SHOTSV(55,2)	10464
COMMON/ONE/LFIT(35), ARRAY(93), IVBUF(1024), C(30,2), ACT,	10465
1 ASCENE, ASEC	10466
COMMON/THREE/HIST(35), AM(90), IVBUF(1024)	10467
DO 100 J=1,2	10468
DO 100 I=1,90	10469
ELMT(I,J)=0.	10470
100 CONTINUE	10471
CALL OPEN(LFIT, JLT=0, IL=0)	10472
CALL OPEN(HIST, JLT=0, IL=0)	10473
110 CALL GETNILEIT, ARRAY, ARRAY(1)	10474
ME=FETCH(LFIT, 2, LF=0)	10475
IF(ME.EQ.9999)GO TO 100	10476
AM(1)=ACT	10477
AM(2)=ARRAY(1)	10478
AM(3)=ARRAY(2)	10479
AM(4)=99999.	10480
CALL SET(HIST, AM, AM(1))	10481
IF(AM(4).NE.99999..AND.1.FAY(5).EQ.AC)GO TO 150	10482
IF(AM(4).NE.99999.)GO TO 140	10483
IF(ARRAY(5).EQ.AC)GO TO 120	10484
ARRAY(5)=AC	10485
IF(ARRAY(4).GT.0.)ARRAY(4)=1.*ARRAY(4)	10486
CALL REPLC(LFIT, ARRAY, 900, ARRAY(1))	10487
120 AM(6)=ARRAY(4)	10488
AM(5)=ARRAY(3)	10489
DO 130 I=6,90	10490
AM(I)=ARRAY(I)	10491
130 CONTINUE	10492
CALL PUT(HIST, AM, 900, AM(1))	10493
GO TO 160	10494
140 ARRAY(5)=ACT	10495
ARRAY(4)=AM(4)	10496
DO 150 I=6,90	10497
ARRAY(I)=AM(I)	10498
IF(I.GT.75)ARRAY(I)=0.	10499
150 CONTINUE	10500
CALL REPLC(LFIT, ARRAY, 900, ARRAY(1))	10501
160 IF(ARRAY(4).GT.0.)GO TO 110	10502
J=ARRAY(3)	10503
IF(J.LT.1.02, J.GT.2)GO TO 110	10504
DO 170 I=1,90	10505
ELMT(I,J)=ELMT(I,J)+ARRAY(I+1)	10506
170 CONTINUE	10507

GO TO 110	009080
100 CALL CLOSEN (IHIST)	009100
CALL CLOSEN (LEFT)	009101
ASECT=0.	009110
AM1="BLUE START"	009120
J=1	009130
190 IPUT=-1	009140
CALL CKSTOR (AM1, IPUT)	009150
DO 200 I=1, 99	009160
AM(I+10)=EL*(I, J)	009170
200 CONTINUE	009180
CALL CKSTOR (AM1, IOUT)	009190
IF (I.EQ. 210) GO TO 210	009200
J=2	009210
A-1="RED START"	009220
GO TO 100	009230
210 RETURN	009240
END	009250

```

S. H. G. E. RESTART
COMMON I3, I7, IP, JENGAS, ITERAN, IVIS, IMOUNT, MINES, CFRK, PCFF, .75
1 ATTEN, GET, .75, IPUN, NCCO, F, AGS, .75, AAK, .75
2 SP(2), FSSF(2), PACK(2),
3 ELIT(10, 2), ALOSS(66, 66), SHOTS(55, 2), CKILL(53, 2), SHOTEV(55, 2)
COMMON/FEED/JDAY1, XIN(4), ICARD(20), IAR, PCF
COMMON/CATA/FDS(80, 2), CDEMS(53, 2), AFOS(12), CPOS(6), ITH, M(20, 2),
1 PSN(6, 2, 2), PLT(15), KEY(171)
COMMON/ONE/LFIT(35), APRAY(90), MYBUF(1024), C(99, 2), ACI,
1ASCENE, ASECT, YSECT
COMMON/TWO/LFIT(35), SPRAY(46), NYBUF(1024)
COMMON/THREE/THIST(35), AH(90), IYBUF(1024)
COMMON/EN/PCOGR(2, 4), ICOUNT(8), ISNUKE(2), PCSMO(2), PCSM(2), ARD(2)
1, MMU(2)
HOLD=ACI
130 PRINT *, " ENTER CI PREVIOUSLY GAMED -- "
CALL REEC3(ACI)
IF(ACI.EQ."999")GO TO 340
C SEARCH FOR CI
CALL OPENM(IMIST, 3LI=0.1LR)
AH(1)=ACI
AH(4)=90909.
CALL GET(IMIST, AH, AH(1), 0, 10)
IF(AH(4).EQ.90909.)GO TO 320
C C4 FOUND--SELECT RESTART OPTION
130 PRINT *, " RELOAD ENTIRE FORCE FILE? "
CALL REECA(IALL)
IF(IALL.EQ."N")GO TO 220
IF(IALL.EQ."Y")GO TO 160
PRINT 300
GO TO 130
160 CALL OPENM(LFIT, 3LI=0.1LR)
170 APRAY(1)=0.
CALL GETN(LFIT, APRAY, APRAY(1))
MM=IFETCH(LFIT, 2LFP)
IF(MM.EQ.1008)GO TO 180
CALL 3LFE(LFIT, APRAY(1))
GO TO 170
180 APRAY(1)=AH(2)
APRAY(2)=AH(3)
APRAY(3)=AH(5)
APRAY(4)=AH(4)
APRAY(5)=AH(1)
DO 190 I=6, 90
IF(I.GT.79)GO TO 185
APRAY(I)=AH(I)
GO TO 190
195 APRAY(I)=

```

190	CONTINUE	00975
	CALL PUT(LFIT,ARRAY,900,ARRAY(1))	00976
	CALL GET(LFIT,AM,AM(1))	00977
	IF(AM(1).EQ.0)GO TO 200	00978
	IF(AM(1).NE.ACI)GO TO 200	00979
	GO TO 190	00980
C	FULL RESTART COMPLETED--CLOSE FILES & EXIT ROUTINE	00981
200	CALL CLOSE(LFIT)	00982
	CALL CLOSE(IHIST)	00983
	PRINT 210,ACI	00984
210	FORMAT(" FORCE FILE RESTARTED AT CI ",A10)	00985
	ACI=0.	00986
	IFIRST=0	00987
	GO TO 360	00988
C	BEGIN SECTOR RELOAD	00989
220	PRINT *, " ENTER SECTOR # TO RELOAD-- "	00990
	CALL REED4	00991
	ASECT=XINX(1)	00992
	IF(ASECT.GE.0..AND.ASECT.LE.99999.1GO TO 230	00993
	PRINT *, " INCORRECT--SECTOR #'S ARE FROM 0 THRU 99999 ONLY!"	00994
	GO TO 220	00995
230	CALL OPENMILFIT,3LI=0,1LR)	00996
	IFLAG=0	00997
240	IF(AM(4).NE.ASECT)GO TO 260	00998
	NEW=0	00999
	ARRAY(1)=AM(2)	01000
	ARRAY(2)=AM(3)	01001
	ARRAY(4)=90909.	01002
C	GET UNIT'S RECORD FROM EXISTING FORCEFILE IF THERE	01003
C	CALL GET(LFIT,ARRAY,ARRAY(1))	01004
	IF(ARRAY(4).EQ.90909.)NEW=1	01005
	ARRAY(4)=AM(4)	01006
	ARRAY(5)=AM(1)	01007
	ARRAY(3)=AM(5)	01008
	DO 250 I=6,90	01009
	ARRAY(I)=AM(I)	01010
250	CONTINUE	01011
C	IF(NEW.EQ.0)CALL REPLC(LFIT,ARRAY,900,ARRAY(1))	01012
	IF(NEW.EQ.1)CALL PUT(LFIT,ARRAY,900,ARRAY(1))	01013
	IFLAG=IFLAG+1	01014
260	CALL GET(IHIST,AM,AM(1))	01015
	IF(AM(1).EQ.0)GO TO 270	01016
	IF(AM(1).NE.0)GO TO 270	01017

270	CALL 3,OFFN(1,FC)	11027
	PRINT 3,ACT,ASECT,FLAG	11028
280	FORMAT(" CI ",A10," SECTOR ",F6.0," RECORD ",F6.0," " UNLS COMPLETE,"//)	11029
C		11030
290	PRINT *, " RELOC ANOTHER SECTOR? "	11031
	CALL RELOC(IYN)	11032
	IF(IYN.EQ."Y")GO TO 310	11033
	IF(IYN.NE."N")PRINT 300	11034
300	FORMAT(" INCORRECT ENTRY--TRY AGAIN")	11035
	IF(IYN.NE."N")GO TO 290	11036
	CALL CLOSE(IHIST)	11037
	ACT=0.	11038
	GO TO 360	11039
310	AM(1)=ACT	11040
	CALL GET(IHIST,AM,AM(1),0,10)	11041
	GO TO 220	11042
320	CALL CLOSE(IHIST)	11043
	PRINT 330,ACT	11044
330	FORMAT(" CI ",A10," NOT FOUND IN HISTORY FILE,"//)	11045
340	CALL SHOWCI	11046
	ACT=HOLD	11047
350	PRINT *, " ABORT RESTART OPTION? "	11048
	CALL RELOC(IYN)	11049
	IF(IYN.EQ."Y")GO TO 350	11050
	IF(IYN.EQ."N")GO TO 120	11051
	PRINT 300	11052
	GO TO 350	11053
360	RETURN	
	END	

SUBROUTINE SHMOCI	11559
COMMON LA,TP,IP,LENGA,G,STEPN,IVIS,IMCUNT,MINES,CPP,FSSP,FPS,	11560
1 ATIME,FIRST,IRUN,NO-2,F-ASS,MAXV,	11561
2 SF(2),FSSF(2),PACK(2),	11562
3 ELY(10,2),ALOS(66,66),SMOTE(55,2),CKILL(53,2),SMOTEV(55,2)	11563
COMMON/DATA/FPS(93,2),CREWS(93,2),ARDS(12),PPS(6),ITHE M(20,2),	11564
1 PSY(6,2,2),P-T(15),KEY(17)	11565
COMMON/ONE/LEFT(36),A-FAY(90),Y-UF(1124),C(60,2),AC,	11566
1ACGENE,A-ECT,Y-ECT	11567
COMMON/TH-EE/FFIT(35),B-FAY(66),MY-UF(1624)	11568
COMMON/TH-EE/CHST(35),AH(90),Y-UF(1124)	11569
COMMON/TH-EE/CDGR(2,4),ICOUNT(9),CSHME(2),PCSMO(2),PCSSM(2),ARD(2)	11570
1,AMU(2)	11571
IF(AC.EQ."999")GO TO 117	11572
100 PRINT *, " DO YOU WANT A LIST OF CIES ON FILE? "	11573
CALL REEDR(IYN)	11574
IF(IYN.EQ."N")GO TO 140	11575
110 PRINT *, " CIES ON HISTORY FILE-- "	11576
AKEEP=99999	11577
CALL OPENH(IHIST,3LI=0,1L?)	11578
120 CALL GETN(IHIST,AM,AM(1))	11579
M=IFETCH(IHIST,2LFF)	11580
IF(M.EQ.1003)GO TO 140	11581
IF(AM(1).EQ.AKEEP.OR.AM(1).EQ."CI LOSSES".OR.AM(1).EQ."CI AMMO")	11582
GO TO 120	11583
AKEEP=AM(1)	11584
PRINT 130,AKEEP	11585
130 FORMAT(" ",22X,A10)	11586
GO TO 120	11587
140 CALL CLOSEH(IHIST)	11588
150 RETURN	11589
170 FORMAT(" INCORRECT ENTRY--TRY AGAIN!")	11590
180 FORMAT(1A1)	11591
190 FORMAT(1X,1A1)	11592
END	11593


```

      FUNCTION INDEX5(I1,I2,I3,I4,I5,L1,L2,L3,L4)      111370
C      THIS FUNCTION RETURNS THE 1 DIMENSIONAL ELEMENT NUMBER OF AN ARRAY 111380
C      SIMULATING ONE OF 5 DIMENSIONS      111390
C      IT IS ASSUMED THE DATA IS STORED BY COLUMNS AND YOU ARE SEEKING 111400
C      ELEMENT (I1,I2,I3,I4,I5) OF ARRAY (L1,L2,L3,L4,N) 111410
C      INDEX5=(1+(I1-1)*L1+(I2-1)*L2+(I3-1)*L3+(I4-1)*L4+(I5-1))) 111420
      RETURN      111430
      END      111440

```

SUBROUTINE LOSS(ISTART,ISTOP,KSTART,KSTOP,INX)	011480
COMMON JA,IO,IP,IENGAG,ITERRN,IVIS,INCUNT,HINES,CFPS,FSFP,PPR,	011490
1 ATIME,IFIRST,IRUN,NOPD,FMASS,IMAXKV,	011500
2 SF(2),FSSF(2),PACK(2),	011510
3 ELMT(30,2),ALOSS(66,66),SHOTS(55,2),CKILL(53,2),SHOTSV(55,2)	011520
COMMON/DATA/FPS(30,2),CREWS(53,2),APOS(12),JPOS(6),ITHE M(2,2),	011530
1 PSY(5,2,2),PLT(15),KEY(17)	011540
IF(INX.EQ."YES")GO TO 140	011550
IF(INX.EQ."NO")GO TO 130	011560
100 PRINT -,"DO YOU WISH TO SUBTRACT LOSSES FROM FORCE STRUCTURE?"	011570
CALL GETA(INX)	011580
IF(INX.EQ."N")GO TO 130	011590
140 DO 170 J=1,2	011600
DO 170 I=ISTART,ISTOP	011610
DO 170 K=KSTART,KSTOP	011620
IF(J.EQ.2)GO TO 150	011630
AKILL=ALOSS(I,K)/PACK(1)/1.	011640
GO TO 150	011650
150 AKILL=(ALOSS(I,K)-IFX(ALOSS(I,K)/PACK(1)*PACK(1)/1.	011660
160 IF(AKILL.LE.0.)GO TO 170	011670
ELMT(K,J)=ELMT(K,J)-AKILL	011680
IF(ELMT(K,J).LT.0.)ELMT(K,J)=0.	011690
170 CONTINUE	011700
C ZEPH OUT AMMO ACCUMULATOR	011710
DO 180 I = 1,2	011720
DO 180 J = 1,55	011730
SHOTS(J,I)=SHOTS(J,I)+SHOTSV(J,I)	011740
SHOTSV(J,I)=0.0	011750
180 CONTINUE	011760
INX="YES"	011770
GO TO 220	011780
190 DO 200 I=ISTART,ISTOP	011790
DO 200 K=KSTART,KSTOP	011800
200 ALOSS(I,K)=0.	011810
DO 210 I=1,2	011820
DO 210 J=1,55	011830
SHOTSV(J,I)=0.0	011840
210 CONTINUE	011850
INX="NO"	011860
220 RETURN	011870
END	011880

SUBROUTINE CDISPLAY	111900
DIMENSION CIL(5),ICODE(10),REMAIN(10),XLOST(10)	111910
COMMON IA,IO,IP,IENGAG,ITERRN,IVIS,IMQUNT,M,NES,CFFP,FSFPP,FPS,	111920
1 ATIME,IFIRST,IRUN,NOPO,FPASS,IMAXKV,	111930
2 SF(2),FSSF(2),PACK(2),	111940
3 ELVT(10,2),ALOSS(66,66),SHOTS(55,2),CKILL(53,2),SMOTSV(55,2)	111950
COMMON/DATA/FPS(10,2),CREWS(53,2),APOS(12),DPOS(6),TIMEIN(20,2),	111960
1 PSN(4,2,2),PLT(15),KEY(193)	111970
COMMON/QNE/LEFT(35),ARRAY(10),MYBUF(1024),C(80,2),ACT,	111980
1ASCENE,ASECT	111990
DATA(CIL(I),I=1,5)/1000.,5.,4.,2.,5.,1./	112000
10 FORMAT(1A10)	112010
1A FORMAT(" ",1A10)	112020
C	112030
PRINT*,"ENTER PARENT OF UNIT(S) TO BE DELAYED -"	112040
CALL READ3(PARENT)	112050
C	112060
TFPS=0.	112070
UEFF=0.	112080
CALL OPENH(LEFT,3L2=0,1,2)	112090
A=PAY(1)=PARENT	112100
ARRAY(7)=99999.	112110
100 CALL GETN(LEFT,A=PAY,ARRAY(1),2,10)	112120
IF(ARRAY(7).NE.99999.)GOTO111	112130
PRINT325,PARENT	112140
GOTO503	112150
C	112160
110 PRINT*,"ENTER UNIT ID (OF ALL) -"	112170
CALL READ3(UNIT)	112180
C	112190
IFLAG=0	112200
113 IF(UNIT.EQ."ALL")GOTO120	112210
IF(UNIT.EQ.ARRAY(2))GOTO120	112220
112 CALL GETN(LEFT,ARRAY,ARRAY(1))	112230
M=IFETCH(LEFT,2LEFT)	112240
IF(M.EQ.1009)GOTO200	112250
IF(ARRAY(1).NE.PARENT)GOTO200	112260
GOTO113	112270
C	112280
120 J=ARRAY(3)	112290
TFPS=TFPS+ARRAY(6)	112300
C	112310
DO 150 I=1,80	112320
AR=ARRAY(I+10)-IFIX(ARRAY(I+10)/130000.)*100000.	112330
IF(AR.LE.0.)GOTO150	112340
UEFF=UEFF+AR*FPS(I,J)	112350
150 CONTINUE	112360
GOTO112	112370

C		112387
200	CALL CLOSEM(LFIT)	112390
	IF (TFPS.LE.0.) GOTO 300	112400
	UEFF=UEFF/TFPS*100.	112410
	P=INT205,PARENT,UEFF	112420
205	FORMAT(' ',//,1X,A10,2X,"EFF=",F4.0,/,1X,67(" "))	112430
	CALL OPENM(LFIT,3LI-0,1LR)	112440
	ARRAY(1)=PARENT	112450
	ARRAY(7)=99999.	112460
	CALL GET(LFIT,ARRAY,ARRAY(1),0,10)	112470
	IF (ARRAY(7).NE.99999.) GOTO 155	112480
	PRINT 305,PARENT	112490
	GOTO 500	112500
C		112510
155	IF L=0	112520
160	IF (UNIT.EQ."ALL") GOTO 260	112530
	IF (UNIT.EQ.ARRAY(2)) GOTO 260	112540
170	CALL GETM(LFIT,ARRAY,ARRAY(1))	112550
	M=IFETCH(LFIT,2LFB)	112560
	IF (M.EQ.1003) GOTO 500	112570
	IF (ARRAY(1).NE.PARENT) GOTO 500	112580
	GOTO 100	112590
C		112600
260	J=ARRAY(3)	112610
	UEFF=1.	112620
	DO 210 I=1,10	112630
	A=ARRAY(I+10)-1/10*(ARRAY(I+10)/100000.)-100000.	112640
	IF (A.LE.0.) GOTO 210	112650
	UEFF=UEFF+A*FOS(I,J)	112660
210	CONTINUE	112670
	IF (ARRAY(6).GT.0.) GOTO 211	112680
	UEFF=999.	112690
	GOTO 212	112700
211	UEFF=UEFF/ARRAY(6)*100.	112710
212	P=INT215,ARRAY(2),UEFF	112720
215	FORMAT(' ',A10,2X,"EFF=",F4.0,/,1X,67(" "))	112730
C		112740
	INC=0	112750
	DO 220 I=1,80	112760
	IF (ARRAY(I+10).EQ.0.) GOTO 225	112770
	INC=INC+1	112780
	ICODE(INC)=I	112790
	XLOST(INC)=IFIX(ARRAY(I+10)/100000.)/10.	112800
	PEMA IN(INC)=ARRAY(I+10)-XLOST(INC)*100000.	112810
	IF (INC.LT.9) GOTO 220	112820
C		112830
230	P=INT235,(ICODE(INX),INX=1,INC)	112840
235	FORMAT(' ITEM CODE',13(3X,I2,1X))	112850

PRINT240.(REMAIN(INX),INX=1,INC)	112960
240 FORMAT(" * REMAIN ",10F6.1)	112970
PRINT245.(XLCST(INX),INX=1,INC)	112980
245 FORMAT(" * LOST ",10(1X,F5.1))	112990
PRINT255	112995
255 FORMAT(" *.67(".")	112910
INC=0	112920
IFLAG=1	112930
225 IF(INC.NE.0.AND.I.EQ.80)GOTO230	112940
220 CONTINUE	112950
GOTO170	112960
500 CALL CLOSEM(LFIT)	112970
GOTO600	112980
300 PRINT305,UNIT	113110
305 FORMAT(" UNIT *.A10," WAS NOT FOUND")	113115
600 RETURN	113120
END	113125

SUBROUTINE RESET	113160
COMMON IA,IO,IF,IENGAG,ITERM,IVIS,IZHUNT,IMINFS,CPFR,FSPP,FE,	113170
1 AT,ME,PF,ST,ISUN,NOC,FMASS,IBAYKV,	113180
2 SF(2),FSSF(2),PACK(2),	113190
3 EMT(30,2),AJSS(66,66),THCT(66,2),CKALL(53,2),SHUTTV(35,2)	113200
COMMON/REO/ICAY1,XINX(4),ICALL(27),IARPCF	113210
COMMON/UNE/LEIT(35),ARRAY(90),ISDF(1124),C(8),2),ACT,	113220
1ASCENE,ASECT,YSECT	113230
COMMON/EN/PCOGR(2,4),PCOIT(8),ISCKE(2),PCMC(2),PCSCM(2),APC(2)	113240
1,IMU(2)	113250
DIMENSION NGA,C(3)	113260
C	113270
PRINT *, "DO YOU WANT TO SAVE THE ELEMENT ARRAY?"	113280
CALL REEC4(INX)	113290
IF(INX.EQ."N")GO TO 130	113300
IFORCE="FORCE"	113310
WRITE(98,100)IFORCE,ACT,ASECT	113320
100 FORMAT(1X,A10,2X,A10,2X,F8.0)	113330
DO 120 I=1,40	113340
IF(ELMT(I,1).EQ.0..AND.ELMT(I,2).EQ.0.)GO TO 120	113350
WRITE(98,110)I,ELMT(I,1),ELMT(I,2)	113360
110 FORMAT(3X,I2,F12.3,F8.0)	113370
120 CONTINUE	113380
IFORCE="END"	113390
WRITE(98,100)IFORCE	113400
C	113410
130 PRINT *, "DO YOU WANT TO RESET ELEMENT ARRAY?"	113420
CALL REEC4(INX)	113430
IF(INX.EQ."N")GO TO 200	113440
DO 140 I=1,65	113450
ELMT(I,1)=ELMT(I,2)=0	113460
140 CONTINUE	113470
PRINT *, "ENTER PREVIOUS CI - "	113480
CALL REEC3(PC1)	113490
PRINT *, "ENTER PREVIOUS SECTOR - "	113500
CALL REEC4	113510
PSECT=XINX(1)	113520
REWINO 98	113530
150 READ(98,100)IFORCE,ACT,ASECT	113540
IF(EOF(98).NE.0)GO TO 190	113550
IF(IFORCE.NE."FORCE")GO TO 150	113560
IF(ACT.NE.PC1)GO TO 150	113570
IF(ASECT.NE.PSECT)GO TO 150	113580
160 READ(98,170)NCARD	113590
170 FORMAT(1X,I4)	113600
IF(EOF(98).NE.0)GO TO 200	113610
IF(NCARD(1).EQ."END")GO TO 200	113620
DECODE(30,130,NCARD)I,ELMT(I,1),ELMT(I,2)	113630

150	FORMAT(2X,12,F12.1,F9.1)	11354
	GO TO 160	11355
190	PRINT *,"DI AND SEC'D ARE FOUND IN E-ENT FILE"	11356
	GO TO 130	11357
C		11358
200	CONTINUE	11359
	DO 210 I=1,3	11360
	ICOUNT(I)=1	11361
210	CONTINUE	11362
	PRINT *,"EW UNITS PER TYPES: 1 EACH TYPE"	11363
	NMU(1)=NMU(2)=1	11364
	PRINT *,"ENTER EW TYPE (1-3), AND UNITS/TYPE; OR 0,0"	11365
250	CALL REEC4	11366
	I=XINX(1)	11367
	IF(I.LT.1.OR.I.GT.3)GO TO 260	11368
	ICOUNT(I)=XINX(2)	11369
	PRINT *,"NEXT - "	11370
	GO TO 250	11371
260	PRINT *,"ENTER FORCE AND NMU/FORCE; OR 0,0"	11372
270	CALL REEC4	11373
	I=XINX(1)	11374
	IF(I.LT.1.OR.I.GT.2)RETURN	11375
	NMU(I)=XINX(2)	11376
	PRINT *,"NEXT - "	11377
	GO TO 270	11378
220	CONTINUE	11379
	END	11380

	SUBROUTINE SUPPRES	11342
	COMMON IA, ID, IP, IENGAG, ITERN, IVIS, I'CUPT, IINES, CFPP, FSPF, PPF,	11343
	1 ATIME, IFIRST, IBUA, NQ30, FMASS, IMAXKV,	11344
	2 SF(2), FSSF(2), PACK(2),	11345
	3 ELVT(90,2), ALOSS(66,66), SHOTS(55,2), CKILL(53,2), SHOTSV(55,2)	11346
	COMMON/DATA/FPS(80,2), CREWS(93,2), AFOS(12), CPUS(6), ITNE-H(23,2),	11347
	1 PSN(6,2,2), PLT(15), KEY(171)	11348
	DIMENSION FACT(12,6,2), STALE(11)	11349
C	FACT(12,6,2) AND STALE(11) ARE USED AS	11350
C	FACT(11,6,2) AND STALE(10)	11351
C	*8.* BELOW IS NOT USED	11352
	DATA (STALE(I), I=1,11)/.6,1.,.1,5,2.,.2,5,3.,.3,5,4.,.5,6.,.6,./	11353
	DATA ((FACT(I,J,K), K=1,2), J=1,6), I=1,12)/	11354
	12,1,8,3,1,2,7,4.,.4,3,7,1,4,2,2,5,2,7,15,0,3,0,11,1,3,3,3,4,1,4,4,4.,	11355
	21,2,2,2,1,4,14,3,3,6,9,3,3,9,6,6,4,5,3,2,2,7,3,1,1,8,1,5,2,4,9,3,4,	11356
	3,4,5,6,4,4,4,5,4,4,3,3,3,6,2,4,2,4,1,2,3,0,1,6,1,5,4,6,3,7,6,5,6,	11357
	4,2,3,4,2,2.,.2,8,1,0,3,5,7,2,6,9,4,4,6,3,7,7,6,2,3,5,1,1,3,3,4.,	11358
	59,3,6,6,3,7,3,4,5,7,3,2,7,7,3,2,1,4,3,1,7,3,3,3,4,5,4,5,3,9,4,3,3,	11359
	6,7,2,6,4,1,2,3,6,6,1,6,2,4,8,5,0,6,3,9,5,4,2,3,6,2,4,3,1,7,3,1,	11360
	71,4,5,4.,.7,6,0,5,4,12,3,8,11,4,2,1,11,3,1,6,9,9,1,3,5,2,7,6,4,5,	11361
	8,3,1,3,5,3,5,12,6,2,0,14,4,1,5,12,3,1,2,3,3,6,9,1,5,0,13,1,7,3,16,	11362
	98,1,9/	11363
C	MULTIPLIES SUPPRESSION FACTOR ARRAY FOR SENSITIVITY FUNCS	11364
	DO 90 N=1,144	11365
	FACT(N)=FACT(N)*3	11366
90	CONTINUE	11367
C		11368
	DO 100 I=1,10	11369
	IF(FPP.LE.STALE(I))GO TO 110	11370
100	CONTINUE	11371
	I=11	11372
110	SF(IA)=FACT(I,IENGAG,2)/100.	11373
	SF(ID)=FACT(I,IENGAG,1)/100.	11374
	DO 120 I=1,10	11375
	IF(FSPF.LE.STALE(I))GO TO 130	11376
120	CONTINUE	11377
	I=11	11378
C	MAKES PS SUP FACTOR INDEP OF TAC SITUATION	11379
C	(USES MEETING ENGAGEMENT FACTORS)	11380
130	FSSF(IA)=FACT(I,1,2)/100.	11381
	FSSF(ID)=FACT(I,1,1)/100.	11382
	PRINT *, "SF(IA), SF(ID)", SF(IA), SF(ID)	11383
	PRINT *, "FSSF(IA), FSSF(ID)", FSSF(IA), FSSF(ID)	11384
	END	11385

```

SUBROUTINE 3GFA                                114270
COMMON IA,IO,IP,IENGAG,ITERRN,IVIS,IMOUNT,MINES,CFFP,FSFP,FPD, 114280
1 AT,ME,IFIRST,IBUN,ABR,F,ASS,IMAKV, 114290
2 SF(2),FSSF(2),PACK(2), 114300
3 FLAT(40,2),ALOSS(66,66),SHOTS(55,2),CKILL(53,2),SHOTCV(55,2) 114310
COMMON/REED/JDAY1,XINX(4),ICARD(23),IARMOR 114320
COMMON/JATA/FPS(40,2),CREWS(53,2),ABUS(12),CPDS(6),ITHEE(123,2), 114330
1 PSN(5,2,2),PLT(15),KEY(17), 114340
COMMON/ONE/LEIT(39),ARRAY(90),MYQUE(1,24),Q(92,2),ACI, 114350
1 ASCENE,ASECT,YSECT 114360
COMMON/THREE/HIST(35),AM(90),ITRUF(1024) 114370
COMMON/EW/PCOGR(2,4),ICOUNT(8),ISMOKE(2),FCSMO(2),PCSSM(2),ARC(2) 114380
1 ,NU(2) 114390
COMMON/FFSC/ACFPS(2),BWFPS(2),ACFPS(2) 114400
DIMENSION STALE(11),FATE(1440),CFPS(2),FFCPS(2),TFPS(2) 114410
DIMENSION PCN(2),PCFS(2),P(2),ITHEE(20,2) 114420
DATA (PCN(I),I=1,2)/1.,1./ 114430
DATA (PCFS(I),I=1,2)/1.,1./ 114440
DATA (STALE(I),I=1,11)/.3,1.,1.5,2.,2.5,3.,3.5,4.,5.,6.,7./ 114450
C 114460
DATA (FATE(I),I=1,2441)/ 114470
10.,3.,.5.,.9,1.3,1.4,1.6,1.7,1.4,2.1,2.2,2.3,3.,.9,1.7,2.1,2.2,2.3,14480
2.,6,2.,9,1.1,3.,4,3,9,4,6,1.,.3.,.5.,.8.,.0,1.2,1.3,1.4,1.4,1.6,114490
3.,.5.,.4.,.8,1.3,1.4,1.4,1.7,1.8,2.1,2.1,2.6,3.,.3.,.2.,.3.,.5.,.7.,.9,1.114500
42,1.3,1.4,1.6,1.9,0.,.0.,.4.,.5.,.9,1.0,1.2,1.4,1.5,1.7,1.4,2.,.3.,.3,14510
5.,.4.,.9,1.0,1.2,1.3,1.4,1.4,1.7,1.4,2.,.6,0.,.5.,.5,1.3,1.6,1.9,2.1,2.314520
63,2.,6,2.,7,3.3,3.9,0.,.0.,.2.,.3.,.5.,.7.,.8.,.8.,.9,1.1,1.2,1.2,1.,.1.,.7,14530
7.5,1.0,1.2,1.4,1.4,1.6,1.7,1.9,2.3,0.,.0.,.1.,.1.,.3.,.4.,.5.,.7.,.3.,.9,14540
99,1.0,0.,.0.,.1.,.3.,.7.,.3.,.9,1.2,1.2,1.3,1.4,1.7,2.,.3.,.3.,.5.,.8.,.9,1214550
9.,.3,1.2,1.3,1.4,1.6,1.7,1.,.0.,.4.,.8,1.2,1.2,1.3,1.6,2.1,2.1,2.6,3.314560
A,0.,.0.,.1.,.2.,.4.,.4.,.5.,.7.,.8.,.8.,.3.,.9,0.,.0.,.2.,.4.,.4.,.9,1.2,1.314570
9,1.4,1.6,2.1,0.,.0.,.1.,.3.,.3.,.4.,.5.,.7.,.7.,.7.,.8,1.,.3.,.5.,.7,14580
C.,.8.,.9,1.0,1.2,1.3,1.6,0.,.0.,.1.,.2.,.4.,.5.,.7.,.9,1.0,1.2,1.2,1.3,1.,.3,14590
D0.,.2.,.4.,.4.,.9,1.1,1.0,1.2,1.3,1.3,1.,.4,0.,.0.,.1.,.2.,.3.,.4.,.5.,.6,14600
E,7.,.7.,.8,1.,.2.,.2.,.4.,.5.,.7.,.7.,.8.,.9.,.9,2.,.3.,.1.,.2.,.3.,.3.,.4,14610
F,4.,.5.,.6.,.7.,.9,0.,.0.,.1.,.2.,.4.,.5.,.5.,.6.,.7.,.7.,.8.,.9,14620
DATA (FATE(I),I=249,576)/ 14630
10.,.3.,.6,1.0,1.5,1.6,1.8,1.9,2.1,2.4,2.5,2.5,2.,.5.,.9,1.7,1.9,2.2,14640
22,9,3,2,3,9,3,8,4,4,5,1,1,2,2,3,6,9,1,0,1,2,1,3,1,5,1,6,1,5,1,9,14650
3,3.,.3.,.5.,.9,1.5,1.6,1.9,1.9,2.1,2.2,2.4,2.4,0.,.1.,.2.,.7.,.6.,.7.,.9,14660
41,0,1,2,1,3,1,5,1,6,0.,.1.,.2.,.4.,.9,1,0,1,2,1,3,1,5,1,6,1,3,2,1,0.,.3,14670
5,3.,.5.,.9,1.2,1.3,1.5,1.6,1.8,1.9,2.1,2.4,0.,.3.,.6,1.2,1.4,2.1,2.4,14680
62,6,2,9,3,1,3,7,4,4,7.,.1.,.2.,.3.,.6.,.7.,.9.,.9,1,3,1,2,1,2,1,3,0.,.2,14690
7,3.,.6,1,2,1,3,1,5,1,6,1,9,2,1,2,6,0.,.1.,.1.,.1.,.3.,.4.,.6.,.9.,.7,14700
8,3,1,0,1,2,2.,.1.,.2.,.3.,.7.,.9,1,0,1,2,1,3,1,5,1,6,1,9,0.,.2.,.3.,.5.,14710
99,1,0,1,2,1,3,1,5,1,6,1,9,1,8,0.,.3.,.5.,.9,1,3,1,5,1,6,1,4,2,2,2,4,14720
A2,9,3,7,0.,.1.,.2.,.3.,.4.,.4.,.6.,.7.,.7.,.9,3,1,5,1,2.,.4.,.9,1,3,1,1,14730
d2,1,3,1,5,1,6,1,8,2,4,0.,.0.,.1.,.1.,.3.,.3.,.4.,.6.,.7.,.7.,.7.,.9,0.,.1,14740

```

M-72

M-73

	PRINT 210	015710
240	PRINT *, " FOR FRONTAL ATTACK.....ENTER 1"	015720
	PRINT *, " SINGLE ENVELOPMENT, ENTER 2"	015730
	PRINT *, " DOUBLE ENVELOPMENT, ENTER 3"	015740
	GO TO 237	015750
250	IP=3*ENGAG-9*INV	015760
C	EW AND SMOKE INPUTS	015770
260	FORMAT(" ENTER PERCENT (DECIMAL) OF ",4," SMOKE")	015780
270	CONTINUE	015790
	AM(11)=IA	015800
	AM(12)=ENGAG	015810
	AM(13)=IAF	015820
	DO 290 IEL=1,2	015830
	KIND="BLUE"	015840
	IF(IEL.EQ.2)KIND="SE"	015850
	PRINT 290,KIND	015860
280	FORMAT(" ENTER THE ",4," FIREPOWER SCORE (0 OF "LRE) FOR "AGAT:")	015870
	CALL REECA	015880
	AC=XINX(1)	015890
	ACFPS(IEL)=AC	015900
	AM(13+IEL)=AC	015910
290	CONTINUE	015920
	DO 300 IEL=1,2	015930
	FSPS(IEL)=0.	015940
	DO 300 I=43,58	015950
	FSPS(IEL)=FSPS(IEL)+ELMT(I,IEL)*FPS(I,IEL)	015960
300	CONTINUE	015970
	DO 310 IEL=1,2	015980
	AMFPS(IEL)=0.	015990
	DO 310 I=59,64	016000
	AMFPS(IEL)=AMFPS(IEL)+ELMT(I,IEL)*FPS(I,IEL)	016010
310	CONTINUE	016020
	DO 330 IEL=1,2	016030
	ADFPS(IEL)=0.	016040
	IS=2	016050
	IF(IEL.EQ.2)IS=1	016060
	IF(AMFPS(IE).EQ.0.01GO TO 330	016070
	DO 320 I=31,42	016080
	ADFPS(IEL)=ADFPS(IEL)+ELMT(I,IEL)*FPS(I,IEL)	016090
320	CONTINUE	016100
330	CONTINUE	016110
	DO 350 IEL=1,2	016120
	DO 340 JCAT=1,4	016130
	PCDJR(IEL,JCA)=0.	016140
340	CONTINUE	016150
350	CONTINUE	016160
360	PRINT *, "PLAY EW?"	016170
	CALL REECA(INX)	016180

IF(INX.EQ."Y")GO TO 370	116190
IF(INX.EQ."N")GO TO 390	116200
GO TO 360	116210
C	116220
370 CALL EW	116230
C	116240
380 GO 390 TEL=1,2	116250
PCSMO(TEL)=0.	116260
PCSSM(TEL)=0.	116270
390 CONTINUE	116280
GO 490 J=1,2	116290
K=1	116300
IF(J.EQ.1)K=2	116310
KIND="BLUE"	116320
IF(J.EQ.2)KIND="RED"	116330
400 PRINT 410,KIND	116340
410 FORMAT(" IS ",A4," EMPLOYING SMOKE?")	116350
CALL RECD(INX)	116360
IF(INX.EQ."Y")GO TO 420	116370
IF(INX.EQ."N")GO TO 480	116380
GO TO 480	116390
420 K=K+1	116400
IF(J.EQ.2)KIND="BLUE"	116410
430 PRINT 260,KIND	116420
CALL RECD	116430
PCSMO(3-J)=XINX(1)	116440
IF(PCSMO(3-J).GE.0..AND.PCSMO(3-J).LE.99)GO TO 440	116450
PRINT 165	116460
GO TO 430	116470
440 PRINT *, "ENTER PERCENT(DECIMAL) OF SELF SMOKE"	116480
CALL RECD	116490
PCSSM(J)=XINX(1)	116500
IF(PCSSM(J).GE.0..AND.PCSSM(J).LE.1)GO TO 450	116510
PRINT 160	116520
GO TO 440	116530
450 IF(J.EQ.1)GO TO 480	116540
460 PRINT *, "ENTER TYPE OF REC SMOKE - "	116550
CALL RECD	116560
INX=XINX(1)	116570
IF(INX.EQ.999)PRINT 470	116580
IF(INX.EQ.1.02,INX.EQ.2)GO TO 440	116590
GO TO 460	116600
470 FORMAT(" ENTER 1 FOR 40 OR IF ONLY SELF CURSURING SMOKE IS EMPLOYED. ENTER 2 FOR OTHER TYPE SMOKE.")	116610
480 ISMOKE(J)=INX	116620
490 CONTINUE	116630
AM(18)=PCSMO(2)	116640
AM(19)=PCSSM(1)	116650
	116660

AM(20)=PCSMQ(1)	016670
AM(21)=PCSSM(2)	016680
IF(ISHOKE(2).NE."N")AM(22)=ISHOKE(2)	016690
900 DO 520 IEL=1,2	016700
CFPS(IEI)=0.	016710
DO 520 I=6,30	016720
PNSMK=(1.-PCSMQ(IEI))*(1.-PCSMQ(3-IEI)/2.)*(1.-PCSSM(IEI))	016730
IF(I.EQ.6)AMFPS(IEI)=AMFPS(IEI)*(1.-(1.-PNSMK)/2.)	016740
IF(AMFPS(IEI).EQ.0.)ACFPS(3-IEI)=0.	016750
IF(I.LT.11)GO TO 510	016760
IF(ISHOKE(2).NE.2.AND.ITHERM(I-11,IEI).EQ.2)PNSMK=1.	016770
IF(IEI.EQ.2.AND.ISHOKE(2).EQ.2.AND.ITHERM(I-11,IEI).EQ.2)PNSMK=1.-	016780
PCSMQ(1)/2.	016790
IF(IEI.EQ.1.AND.ISHOKE(2).EQ.2.AND.ITHERM(I-11,IEI).EQ.2)PNSMK=1.-	016800
PCSMQ(1)	016810
510 CFPS(IEI)=CFPS(IEI)+ELMT(I,IEI)+FPS(I,IEI)*PNSMK	016820
920 CONTINUE	016830
DO 530 I=1,2	016840
CFPS(I)=CFPS(I)*(1.-PCOGR(I,4))	016850
FSFPS(I)=FSFPS(I)*(1.-PCCGP(I,3))	016860
AMFPS(I)=AMFPS(I)*(1.-PCOGR(I,2))	016870
ACFPS(I)=ACFPS(I)*(1.-PCCGP(I,1))	016880
CFPS(I)=CFPS(I)+AMFPS(I)	016890
FSFPS(I)=FSFPS(I)+ACFPS(I)	016900
530 CONTINUE	016910
CFPS(IA)=CFPS(IA)+APOS(IP)	016920
CFPS(ID)=CFPS(ID)+CPOS(IEGAG)	016930
IF(CFPS(ID)+FSFPS(ID).GE.1)GO TO 540	016940
PRINT *,"***THERE ARE NO DEFENDERS***"	016950
PRINT *," FORCE STRUCTURE MUST BE CHANGED"	016960
IF(IRUN.EQ.1)PRINT *," PROGRAM STOPPED---ATTACH FORCE FILE BEFORE	016970
1 RESTARTING"	016980
IF(IRUN.EQ.1)STOP	016990
GO TO 1000	017000
540 IFIRST=1	017010
IF(CFPS(ID).GE.1)GO TO 550	017020
PRINT *,"THERE IS NO MANEUVER FF FATIO"	017030
CFPR=0.	017040
GO TO 560	017050
550 CFPR=CFPS(IA)/CFPS(ID)	017060
560 IF(FSFPS(ID).GE.1)GO TO 570	017070
PRINT *,"THERE IS NO FIRE SUPPORT FF FATIO"	017080
FSFPR=0.	017090
GO TO 580	017100
570 FSFPR=FSFPS(IA)/FSFPS(ID)	017110
580 FPR=(CFPS(IA)+FSFPS(IA))/(CFPS(ID)+FSFPS(ID))	017120
C NOTE-ALL AUTOMATIC WEAPONS WERE CONSIDERED IN MANEUVE FIREPOWER.	017130
590 PRINT *,"ENTER TERRAIN TYPE"	017140

CALL REEDA	117150
I*ERRN=XINX(1)	117160
AM(23)=I*ERRN	117170
IF(I*ERRN.EQ.999)GO TO 600	117180
IF(I*ERRN.GE.1.AND.I*ERRN.LE.4)GO TO 610	117190
PRINT 210	117200
600 PRINT *, "FOR OPEN TERRAIN.....ENTER 1"	117210
PRINT *, "ROLLING TERRAIN.....ENTER 2"	117220
PRINT *, "HILLY TERRAIN.....ENTER 3"	117230
PRINT *, "MOUNTAINOUS TERRAIN. ENTER 4"	117240
GO TO 590	117250
610 PRINT *, "ENTER VISIBILITY FACTOR"	117260
CALL REEDA	117270
IVIS=XINX(1)	117280
IF(IVIS.EQ.999)GO TO 620	117290
AM(25)=IVIS	117300
IF(IVIS.GE.1.AND.IVIS.LE.5)GO TO 630	117310
PRINT 210	117320
620 PRINT *, "FOR VISTABILITY OF 100% ENTER 1"	117330
PRINT *, "85% ENTER 2"	117340
PRINT *, "65% ENTER 3"	117350
PRINT *, "45% ENTER 4"	117360
PRINT *, "30% ENTER 5"	117370
GO TO 610	117380
630 JVIS=(IVIS+1)/2	117390
640 PRINT *, "IS THIS A DAY BAT*LE?"	117400
CALL REEDA(XINX)	117410
IF(INX.EQ."Y".OR.INX.EQ."N")GO TO 650	117420
GO TO 640	117430
650 JOAY1="NIGHT"	117440
IF(INX.EQ."Y")JOAY1="DAY"	117450
NORD=2	117460
IF(INX.EQ."Y")NORD=1	117470
AM(24)=NORD	117480
PRINT *, "WILL NEW ARMOR BE FULLY USED?"	117490
CALL REEDA(IARMGR)	117500
660 PRINT *, "IS ATTACKER MOUNTED"	117510
CALL REEDA(INX)	117520
IF(INX.EQ."N")IMOUNT=1	117530
IF(INX.EQ."Y")IMOUNT=2	117540
AM(26)=IMOUNT	117550
IF(INX.EQ."N".OR.INX.EQ."Y")GO TO 670	117560
GO TO 660	117570
670 PRINT *, "ENTER FFACTION OF SECTOR ATTACKER MASSEC (MAX=1)"	117580
CALL REEDA	117590
FMASS=XINX(1)	117600
AM(27)=FMASS	117610
IF(FMASS.GT.0..AND.FMASS.LE.1)GO TO 670	117620

PRINT 210	017630
GO TO 670	017640
675 IF (FMASS.EQ.1.0) GO TO 710	017650
680 PRINT *, " ENTER HOLDING FIREPOWER RATIO- C<MIN<=2 "	017660
CALL REEC4	017670
RMIN=XINX(1)	017680
AM(20)=RMIN	017690
IF (RMIN.GT.0.AND.RMIN.LE.2.0) GO TO 690	017700
PRINT 210	017710
GO TO 680	017720
690 FPR=(FPR-RMIN*(1.-FMASS))/FMASS	017730
700 IF (FPR.LT.RMIN.OR.FMASS.EQ.1.0) FPR=RMIN	017740
IF (IENGAG.LE.3) GO TO 710	017750
STALE(2)=1.4	017760
STALE(3)=1.7	017770
710 DO 720 I=1,11	017780
IF (FPR.LT.STALE(I)) GO TO 730	017790
720 CONTINUE	017800
I=12	017810
730 IFPR=I	017820
ITABLE=IENGAG	017830
IF (ITABLE.EQ.3) ITABLE=2	017840
IF (ITABLE.GT.3) ITABLE=ITABLE-1	017850
C ENTER RATE ARRAY AND READ ROA.	017860
ROA=RATE (INDEXS(IFPR,IMOUNT,JVIS,ITERFA,ITABLE,12,2,3,4))	017870
IF (IFPR.LT.12) GO TO 740	017880
ROA2=ROA1	017890
GO TO 750	017900
740 ROA2=RATE (INDEXS(IFPR+1,IMOUNT,JVIS,ITERFA,ITABLE,12,2,3,4))	017910
750 ROA=ROA1+(ROA2-ROA1)*(FPR-STALE(IFPR-1))/(STALE(IFPR)-STALE(IFPR-1))	017920
1.1)	017930
IF (ROA1.LE.0.1) ROA=0.	017940
IF (IENGAG.EQ.3.AND.IMOUNT.EQ.2) ROA=ROA*1.5	017950
760 PRINT *, " ARE MINES EMPLOYED IN THIS SECTION "	017960
CALL REEC4 (INX)	017970
IF (INX.EQ."Y") GO TO 770	017980
IF (INX.EQ."N") GO TO 790	017990
GO TO 760	018000
770 IF (IENGAG.NE.4.AND.IENGAG.NE.5) ROA=.75*ROA	018010
MINES=1	018020
GO TO 790	018030
780 MINES=2	018040
790 PRINT *, " SHOULD TIME OR DISTANCE BE HELD CONSTANT "	018050
CALL REEC4	018060
INX=XINX(1)	018070
IF (INX.EQ.999) GO TO 800	018080
IF (INX.EQ.1) GO TO 810	018090
IF (INX.EQ.2) GO TO 830	018100

PRINT 210	119110
800 PRINT " FOR CONSTANT TIME.....ENTER 1"	119120
PRINT " CONSTANT DISTANCE,ENTER 2"	119130
GO TO 790	119140
810 PRINT "ENTER ATTACK TIME IN HOURS (MAX 24)"	119150
CALL REED4	119160
ATIME=XINX(1)	119170
IF(ATIME.LE.24.AND.ATIME.GE.1)GO TO 820	119180
PRINT 210	119190
GO TO 910	119200
820 ACIST=304*ATIME	119210
GO TO 860	119220
930 PRINT "ENTER ATTACK DISTANCE IN METERS (MAX 75100.)"	119230
CALL REED4	119240
ADIST=XINX(1)	119250
IF(ADIST.GT.0..AND.ADIST.LE.75100.)GO TO 840	119260
PRINT 210	119270
GO TO 930	119280
840 IF(ROA.EQ.0.)GO TO 850	119290
ATIME=(ACIST/1000.)/ROA	119300
ADIST=ADIST/1000.	119310
GO TO 960	119320
850 ATIME=0.	119330
ADIST=0.	119340
860 AM(29)=MINES	119350
AM(30)=ATIME	119360
AM(31)=ACIST	119370
AM(32)=ROA	119380
PRINT 870	119390
870 FORMAT("I")	119400
TFPS(IA)=CFPS(IA)+FSFPS(IA)	119410
TFPS(ID)=CFPS(ID)+FSFPS(ID)	119420
POINT "-----RATE OF ADVANCE-----"	119430
1-----	119440
PRINT 880	119450
880 FORMAT("I",63X,"I")	119460
890 FORMAT("I FP RATIO IN SECTOR'S MAIN ATTACK AREA",5("I"),	119470
C F4.1,13X,"I")	119480
900 FORMAT("I TOTAL FP RATIO",28("I"),F4.1,13X,"I")	119490
910 FORMAT("I MANEUVER FP RATIO",25("I"),F4.1,13X,"I")	119500
920 FORMAT("I FIRE SUPPORT FP RATIO",21("I"),F4.1,13X,"I")	119510
930 FORMAT("I RATE-OF-ADVANCE (KPH)",20("I"),F5.2,13X,"I")	119520
940 FORMAT("I DURATION OF ATTACK (HR)",18("I"),F5.1,13X,"I")	119530
950 FORMAT("I DISTANCE ADVANCED (KM)",19("I"),F5.1,13X,"I")	119540
960 FORMAT("I MANEUVER FP SCORE",25("I"),F3.0,"/",F3.0,"I")	119550
970 FORMAT("I FIRE SUPPORT FP SCORE",21("I"),F3.0,"/",F3.0,"I")	119560
980 FORMAT("I TOTAL FP SCORE",28("I"),F3.0,"/",F3.0,"I")	119570
PRINT 960,CFPS(IA),CFPS(ID)	119580

PRINT 970,FSFPS(IA),FSFPS(ID)	018590
PRINT 980,TFPS(IA),TFPS(ID)	018600
PRINT 890,FPM	018610
PRINT 900,FPR	018620
PRINT 910,CFPR	018630
PRINT 920,FSFPR	018640
PRINT 890	018650
PRINT 930,ROA	018660
PRINT 940,ATIME	018670
PRINT 950,ADIST	018680
PRINT 960	018690
PRINT *, "*****"	018700
1*****	018710
PRINT 970	018720
990 IF (IRUN.EQ.1) PRINT *, " BATTLE CHARACTERISTICS PRINTED HERE"	018730
DO 995 J=1,2	018740
AM(32+J)=CFPS(J)	018750
AM(34+J)=FSFPS(J)	018760
AM(36+J)=TFPS(J)	018770
995 CONTINUE	018780
AM(39)=FPM	018790
AM(40)=FPR	018800
AM(41)=CFPR	018810
AM(42)=FSFPR	018820
CALL CKSTOR(AM1,IPU)	018830
1000 END	018840

```

SUBROUTINE EW
COMMON LA, ID, IP, IENGAG, ITEXRN, IVIS, INCUNT, MINES, CFPR, FSFP, FP,
1 ATIME, IFIRST, IRUN, NO-C, FMASS, MAYKV,
2 SF(2), FSSF(2), PACK(2),
3 ELMT(30,2), ALOSS(66,66), SHOTS(55,2), CKILL(53,2), SHOTSV(55,2)
COMMON/REED/JDAY1, XINX(4), ICARD(23), IARPOF
COMMON/DATA/FPS(30,2), CREWS(53,2), AFOS(12), CPDS(6), ITHE-M(23,2),
1 PSN(6,2,2), PLT(15), KEY(171)
COMMON/ONE/LFIT(35), ARRAY(90), YTRUF(162), C(83,2), ACT,
1 ASCENE, ASECT, YSECT
COMMON/EH/PCDGR(2,4), ICOUNT(8), ZSMOKE(2), PCSHC(2), PCSTM(2), A33(2)
1 ,NMU(2)
COMMON/FPSG/ACFPS(2), AHFPS(2), ACFPS(2)
DIMENSION IJUL(2,4), ART(2,6), RMNV(6), NJMRQ(2,4), IP(4),
1 PCEFF(4), PCFPR(4), JCQNT(6)
DATA ((IJUL(I,J), J=1,4), I=1,2)/2,1,2,1,2,1,1,1/
DATA ((ART(I,J), J=1,6), I=1,2)/0,2,2,4,4,6,4,4,5,7,9,
1.99/
DATA (RMNV(I), I=1,6)/0,5,35,1,1,2/
DATA ((NJMRQ(I,J), J=1,4), I=1,2)/1,3,1,3,6,3,7,15/
C
100 FORMAT(" INCORRECT RESPONSE-TRY AGAIN")
C
PRINT *, "ENTER EW SEED="
CALL REEDB(INX)
ISEED=INX+ASECT
CALL RANSE(ISEED)
DO 360 J=1,2
KIND=" SEED"
IF(J.EQ.2) KIND="BLUE"
140 PRINT 150, KIND
150 FORMAT(" ENTER THE # OF ".A4." EW MISSIONS REQUESTED (MAX=53)")
CALL REED4
NJAMS=XINX(1)
IF(NJAMS.LT.0.OR.NJAMS.GT.50) GO TO 160
IF(NJAMS.EQ.0) GO TO 360
GO TO 170
160 PRINT 100
GO TO 140
C
170 DO 180 I=1,4
IF(J.EQ.1) JCOUNT(I)=ICOUNT(I)
IF(J.EQ.2) JCOUNT(I)=ICOUNT(I+4)
IPK(I)=0
180 CONTINUE
C
190 PRINT *, "CHANGE EW PRIORITY?"
CALL REED4(INX)

```

	IF(INX.EQ.999)GO TO 200	019340
	IF(INX.EQ."Y")GO TO 200	019350
	IF(INX.EQ."N")GO TO 230	019360
	GO TO 190	019370
C		019380
	200 PRINT *, " THE PRESENT ORDER OF PRIORITY IS:"	019390
	PRINT *, " "	019400
	PRINT *, " 1 AIR DEFENSE ARTILLERY"	019410
	PRINT *, " 2 ATTACK HELICOPTERS"	019420
	PRINT *, " 3 ARTILLERY UNITS"	019430
	PRINT *, " 4 MANEUVER UNITS"	019440
C		019450
	IF(INX.EQ.999)GO TO 190	019460
	210 PRINT *, " ENTER THE 4 ITEM NUMBERS FROM HIGH TO LOW PRIORITY"	019470
	CALL REEC4	019480
	IX=XINX(1)	019490
	IX=XINX(2)	019500
	IY=XINX(3)	019510
	IZ=XINX(4)	019520
	IPR(IY)=1	019530
	IPR(IX)=2	019540
	IPR(IY)=3	019550
	IPR(IZ)=4	019560
	IC=1	019570
	DO 220 I=1,4	019580
	IC=IC+IPR(I)	019590
	IF(IPR(I).LT.0.OR.IPR(I).GT.4)IC=1	019600
	220 CONTINUE	019610
	IF(IC.NE.10)PRINT 133	019620
	IF(IC.NE.10)GO TO 213	019630
	GO TO 250	019640
C		019650
	230 DO 240 I=1,4	019660
	IPR(I)=I	019670
	240 CONTINUE	019680
C		019690
	250 DO 260 I=1,4	019700
	PCEF(I)=0	019710
	260 CONTINUE	019720
C		019730
	IF(AHFPS(J).EQ.0)JCOUNT(1)=0	019740
	IF(AHFPS(J).EQ.0)JCOUNT(2)=0	019750
	270 DO 280 JS=1,4	019760
	DO 280 I=1,4	019770
	IF(IPR(I).NE.IS)GO TO 280	019780
	IF(JCOUNT(I).EQ.0)GO TO 280	019790
	IX=MAX(J,NJMRQ(J,I))	019800
	IF(IX.GT.JCOUNT(I))IX=JCOUNT(I)	019810

IF (IX.EQ.1)GO TO 289	119820
PCEFF(I)=FLOAT(IX)/JCOUNT(I)	119830
NJAMS=NJAMS-IX*NJMRG(J,I)	119840
288 CONTINUE	119850
290 CONTINUE	119860
C	019870
DO 300 I=1,4	019880
IF (PCEFF(I).EQ.0)GO TO 303	019890
NRN=RANF(X)*132.	019900
IF (NRN.GE.1.AND.NRN.LE.16)M=1	019910
IF (NRN.GE.17.AND.NRN.LE.33)M=2	019920
IF (NRN.GE.34.AND.NRN.LE.50)M=3	019930
IF (NRN.GE.51.AND.NRN.LE.67)M=4	019940
IF (NRN.GE.68.AND.NRN.LE.84)M=5	019950
IF (NRN.GE.85.AND.NRN.LE.100)M=6	019960
IF (NRN.EQ.0)M=6	019970
PRINT *, " CHANCE ROLL WAS ",M	019980
IVC=IVUL(J,I)	019990
IF (I.EQ.1.OR.I.EQ.3)PCFPR(I)=AR*(IVC,M)	020000
IF (I.EQ.2.OR.I.EQ.4)PCFPR(I)=MNV(M)	020010
300 CONTINUE	020020
C	020030
DO 350 I=1,4	020040
PCDGR(J,I)=PCFPR(I)*PCEFF(I)	020050
KIND="BLUE"	020060
IF (J.EQ.2)KIND="RED"	020070
IF (I.EQ.1)PRINT 310,KIND,PCDGR(J,I)	020080
IF (I.EQ.2)PRINT 320,KIND,PCDGR(J,I)	020090
IF (I.EQ.3)PRINT 330,KIND,PCDGR(J,I)	020100
IF (I.EQ.4)PRINT 340,KIND,PCDGR(J,I)	020110
310 FORMAT(" PERCENT ",A4,"AOA FP SCORE DEGRADED- ",F4.2)	020120
320 FORMAT(" PERCENT ",A4,"AH FP SCORE DEGRADED- ",F4.2)	020130
330 FORMAT(" PERCENT ",A4,"ARY FP 1 MISSIONS DEGRADED- ",F4.2)	020140
340 FORMAT(" PERCENT ",A4,"MNV FP SCORE DEGRADED- ",F4.2)	020150
350 CONTINUE	020160
360 CONTINUE	020170
END	020180

```

SUBROUTINE TANK
COMMON IA, ID, IP, IENGAG, ITERN, IVIS, IVOUNT, MINES, OFPE, FSPF, FP,
1 ATIME, IFIRST, IRUN, NUPD, FMAS, IMAXV,
2 SFI(2), FSSF(2), PACK(2),
3 ELVT(10,2), ALOSS(66,66), SHOTS(55,2), CKILL(53,2), SHOTSJ(55,2)
COMMON/REED/JDAY1, XINX(4), ICARD(20), IARMOF
COMMON/DATE/FP(10,2), CREWS(53,2), AFOS(12), JFOS(6), ITIME(21,2),
1 PSM(6,2,2), PLT(15), KEY(171)
COMMON/QNE/LFIT(35), APRAY(90), NYBUF(1024), Q(10,2), AC,
1 ASCENE, ASECT
COMMON/THREE/THST(35), AM(90), TVBUF(1024)
COMMON/EW/PCOGR(2,4), ICUNT(8), ISMOKE(2), PCSM(2), PCSSM(2), AR(2)
1, NMU(2)
C
LOGICAL IFLAG, JFLAG, KFLAG
DIMENSION PDF(2), SUPDEG(3), A(6,2), FIRE(6,29,2), ATCHEW(5)
1, JFERN(4)
DIMENSION FOX(6,4), PICKET(6,5,2)
DIMENSION SSKP(32), ACQ(2), OFEPA(22,2), PLOSS(32,2,2), WTS(4,2),
1 PKILL(20)
DIMENSION ISAMMO(20), IPAMMO(20)
DIMENSION PUN(4), STORE(22,2)
DIMENSION PCQ1(2), PSS(2)
DIMENSION ICA(22,2)
DIMENSION OUST(4,6,2)
C
DATA ((ICUST(I,J,K),J=1,6),I=1,4),K=1,2)/
1 2*1..96..42..73..54,2*1..92..79..68..6,
2 2*1..79..6..34..0..1..71..40,
3 1..94..74..58..46..3..1..93..7..54..41..09,
4 1..92..52..34,2*0..1..44,4*2./
C..... TARGET CATEGORY
DATA ICA(1,5*4,3,2,2,3,3,2,3,3,2,2,2,
1 4*1,2,5*4,3,3,4*2,3,3,4*2/
DATA (ATCHEW(I),I=1,5)/1..2..3..2..3./
DATA ((WTS(I,J),J=1,2),I=1,4)/.6..6,4,3,5,7,5,9,7,4,10..1./
C DATA ((WTS(I,J),J=1,2),I=1,4)/.6..6,5,3,6,8,5,3,6,9,10..1./
C DATA ((I(I),I=1,5)/1..95..65..45..3/
DATA ((ISAMMO(I),I=1,20)/12,10,4,11,9,3,3,6,5,3,4,
1 3,14,0,0,8,4,8,3,0/
DATA ((IPAMMO(I),I=1,20)/10,11,4,0,6,2,2,4,6,7,8,8,3,13
1 4,6,7,9,4,0,14/
DATA ((A(I,J),I=1,6),J=1,2)/.9..9..9..9..9..9..75..75,2*33..53/
DATA (SUPDEG(I),I=1,3)/1..2..96,3,52/
DATA (JTERN/"OPEN","ROLLING","HILLY","MOUNTAIN"
1 /
DATA ((OPERA(K,J),K=1,22),J=1,2)/
1 .95..81..93..95..95,

```

d.79..79..67..79..78.	122690
e.81..81..81..81..81..81..81..81..81..81.	122690
f.60..75.	122710
g.95..95..93..08..96.	122710
h.79..79..78..62..00.	122720
i.81..81..81..81..81..81..81..81..81..81.	122730
j.85..85/	122740
C	122750
C	122760
C	122770
C	122780
C.....	122790
C	122800
C	122810
C	122820
C	122830
C	122840
C	122850
C	122860
C	122870
C	122880
C	122890
C	122900
C	122910
C	122920
C	122930
C	122940
C	122950
C	122960
C	122970
C	122980
C	122990
C	123000
C	123010
C	123020
C	123030
C	123040
C	123050
C	123060
C	123070
C	123080
C	123090
C	123100
C	123110
C	123120
C	123130
C	123140
C	123150

	5	1.0,	1.0,	.90,	0.6,	0.3,	0.3/	121165
C								121170
C								121180
C								121190
C								121200
C								121210
C								121220
C								121230
C								121240
C								121250
C								121260
C								121270
C								121280
C								121290
C								121300
C								121310
C								121320
C								121330
								121340
	100							121350
								121360
								121370
								121380
								121390
	110							121400
C								121410
								121420
								121430
								121440
								121450
	115							121460
								121470
								121480
								121490
	117							121500
C								121510
								121520
								121530
								121540
								121550
C								121560
C								121570
C								121580
C								121590
C								121600
								121610
	120							121620
								121630

IF(JDAY.NE.JDAY1)GO TO 130	021640
GO TO 150	021650
130 READ(13,140)JSEAS,JDAY,JTERR,JOPT,JVIS,((FIRE(KK,K,1),KK=1,6),K=1,6)	021660
1 290	021670
140 FORMAT(1X,5A10,(/1X,20F6.3))	021680
IF(EOF(10).NE.0)GO TO 150	021690
GO TO 120	021700
150 CONTINUE	021710
C	021720
C.....READ THERMAL DATA	021730
REWIND 10	021740
160 IF(JOPT.NE."THERM")GO TO 170	021750
IF(JTERR.NE.JTERRN(ITERRN))GO TO 170	021760
IF(JDAY.NE.JDAY1)GO TO 170	021770
GO TO 180	021780
170 READ(10,140)JSEAS,JDAY,JTERR,JOPT,JVIS,((FIRE(KK,K,2),KK=1,6),K=1,6)	021790
1 1,K=1,29)	021800
IF(EOF(10).NE.0)GO TO 190	021810
GO TO 160	021820
180 CONTINUE	021830
C REDEFINE FIRE IN SOME CASES	021840
DO 185 I=3,6	021850
185 FIRE(I,1,ITHEM(1,1))=0	021860
C	021870
CALL OPENMS(3,KEY,171,0)	021880
C.....REDEFINE JVIS FOR DUST ARPA	021890
JVIS=JVIS-1	021900
IF(JVIS.EQ.0)JVIS=1	021910
C	021920
C	021930
190 FDF(1)=0.	021940
FDF(2)=0.	021950
DO 200 I=1,32	021960
DO 200 J=1,2	021970
DO 200 K=1,2	021980
200 PLOSS(I,J,K)=0.	021990
C	022000
C INTERVISIBILITY ENTRY	022010
C	022020
ACQ(1A)=A(ENGAG,2)	022030
ACQ(1D)=A(ENGAG,1)	022040
210 PRINT *, "ENTER RANGE INDEX BETWEEN ATTACKER & DEFENDER"	022050
CALL REEO4	022060
MAXR=XINX(1)	022070
II=6	022080
IF(MAXR.LT.(C.OR.MAXR.GT.6)GO TO 230	022090
IF(NMD.LE.84)ARRAY(NMD)=MAXR	022100
DO 220 I=1,4	022110

220	PUNT1)=1.	122120
	IF(MAXR.EQ.999)GO TO 230	122130
	IF(MAXR.EQ.9)GO TO 1010	122140
C		122150
C	CHECK RANGE LIMITATIONS	122160
C		122170
	POPT=PICKET(MAXR,IVIS,1)	122180
	PTMH=PICKET(MAXR,IVIS,2)	122190
	IF(POPT.NE.0.)GO TO 270	122200
	PRINT *,"VISIBILITY INSUFFICIENT FOR ENGAGEMENT AT SPECIFIED RANGE	122210
	1 WITH OPTICS "	122220
	IF(PTMH.NE.0.)GO TO 270	122230
	CALL CONNCT(50)	122240
	PRINT *,"VISIBILITY INSUFFICIENT FOR ENGAGEMENT AT SPECIFIED RANGE	122250
	2"	122260
	DO 225 I=1,MAXR	122270
	II=1+MAXP-I	122280
	IF(PICKET(II,IVIS,2).NE.0.)GO TO 230	122290
	225 CONTINUE	122300
	GO TO 1010	122310
230	PRINT *," IF RANGE IS BETWEEN:"	122320
	GO TO (260,257,255,250,245,240),II	122330
240	PRINT *," 3000 & 2501 ENTER 6"	122340
245	PRINT *," 2500 & 2001 ENTER 5"	122350
250	PRINT *," 2000 & 1501 ENTER 4"	122360
255	PRINT *," 1500 & 1001 ENTER 3"	122370
257	PRINT *," 1000 & 501 ENTER 2"	122380
260	PRINT *," 500 & 0 ENTER 1"	122390
	PRINT *," **TO STOP** ENTER 0"	122400
C		122410
	GO TO 210	122420
270	DO 310 J=1,2	122430
	KIND="BLUE"	122440
	IF(J.EQ.2)KIND="RED"	122450
280	PRINT 290,KIND	122460
290	FORMAT(" ENTER PERCENT OF ",A4,"APMCR/ANTIARMOR COMMITTED ")	122470
	CALL REEC4	122480
	P=XINX(1)	122490
	IF(P.GE.0..AND.P.LE.1)GO TO 300	122500
	PRINT 1160	122510
	GO TO 280	122520
300	PCOM(J)=P	122530
	IF(NWC.LE.84)ARRAY(NWC+J)=PCOM(J)	122540
310	CONTINUE	122550
C		122560
	DO 390 J=1,2	122570
	JFLAG=.FALSE.	122580
	IF(J.GE.JTEST1.AND.J.LE.JTEST2).FLAG=.TRUE.	122590

PSS(J)=0.	022600
IF(ISHOKE(J).EQ."N")GO TO 390	022610
K=1	022620
IF(J.EQ.1)K=2	022630
KIND="BLUE"	022640
IF(J.EQ.2)KIND="RED"	022650
320 PRINT 330,KIND	022660
330 FORMAT(" IS ",A4," EMPLOYING SMOKE? ")	022670
CALL REEDA(INX)	022680
IF(INX.EQ."Y") GO TO 340	022690
IF(INX.EQ."N") GO TO 390	022700
GO TO 320	022710
340 KIND="RED"	022720
IF(J.EQ.2)KIND="BLUE"	022730
350 PRINT 360,KIND	022740
360 FORMAT(" ENTER PERCENT(DECIMAL) ",A4," WEAPONS OBSERVED")	022750
CALL REEC4	022760
P=XINX(1)	022770
IF(P.GE.0..AND.P.LE.1.)GO TO 370	022780
PRINT 1160	022790
GO TO 390	022800
370 PRINT *, "ENTER PERCENT (DECIMAL) OF SELF SMOKE "	022810
CALL REEC4	022820
PSS(J)=XINX(1)	022830
IF(PSS(J).GE.0..AND.PSS(J).LE.1.)GO TO 380	022840
PRINT 1160	022850
GO TO 370	022860
380 CONTINUE	022870
IF(JFLAG)PRINT(50,*) "PUN,P,PSS = ",PUN,P,PSS	022880
PUN(3-K)=PUN(3-K)*(1.-P/2.)	022890
PUN(K)=PUN(K)*(1.-P)	022900
IF(J.EQ.2)PUN(3)=1.-P	022910
IF(J.EQ.2)PUN(4)=1.-P/2.	022920
N1=NWO+2	022930
N2=NWO+4	022940
IF(NWC.LE.84)ARRAY(N1+J)=P	022950
IF(NWO.LE.34)ARRAY(N2+J)=PSS(J)	022960
390 CONTINUE	022970
PUN(1)=PUN(1)*(1.-PSS(1))	022980
PUN(2)=PUN(2)*(1.-PSS(2))	022990
C	023000
400 FOF(1)=0.	023010
FOF(2)=0.	023020
DO 410 J=1,2	023030
FOF(J)=0.	023040
DO 410 K=1,32	023050
IF(ELMT(K,J)-PLOSS(K,J,1).LT.0.)GO TO 410	023060
IF(J.EQ.1A.AND.IMPUNT.EQ.2.AND.K.LT.16)GO TO 410	023070

[illegible]

FARD=OUST(JVIS,MAXR,2)	023560
IF(IARD(L).LT.84.)FARD=OUST(JVIS,MAXR,1)	023570
IF(IARD(L).LT.21.)FARD=1.	023580
C CHECK THREAT BARRAGE	023590
TARD=OUST(JVIS,MAXR,2)	023600
IF(IARD(J).LT.84.)TARD=OUST(JVIS,MAXR,1)	023610
IF(IARD(J).LT.21.)TARD=1.	023620
C USE DOMINANT EFFECT	023630
AJUST=FARD	023640
IF(TARD.LT.FARD)AJUST=TARD	023650
C	023660
C..TARGET LOOP..TARGET LOOP..TARGET LOOP..TARGET LOOP..TARGET LOOP..TARGET	023670
C	023680
C TARGET LOOP	023690
C	023700
C..TARGET LOOP..TARGET LOOP..TARGET LOOP..TARGET LOOP..TARGET LOOP..TARGET	023710
C	023720
DO 900 K=1,32	023730
KFLAG=.FALSE.	023740
IF(JFLAG.AND.K.GE.KTEST1.AND.K.LE.KTEST2)KFLAG=.TRUE.	023750
IF(KFLAG)PRINT(50,*)"***** TARGET NO. = ",K	023760
IF(KFLAG.AND.FDF(L).LE.C)PRINT(50,*)"NO TARGETS"	023770
IF(FDF(L).LE.0.)GO TO 890	023780
IF(L.EQ.1A.AND.INCOUNT.EQ.2.AND.K.LT.16)GO TO 890	023790
IF(KFLAG)PRINT(50,*)"ELMT,PLOSS,PCOM,PUN = ",	023800
1 ELMT(K,L),PLOSS(K,L,1),PCOM(L),PLN	023810
ELMTS=(ELMT(K,L)-PLOSS(K,L,1))*PCOM(L)	023820
IF(ITPASS.EQ.1)ELMTS=ELMTS*PUN(L)	023830
IF(ITPASS.EQ.2.AND.ISMOKE(2).NE.2)ELMTS=ELMTS*(1.-PUN(L))	023840
IF(ITPASS.EQ.2.AND.ISMOKE(2).EQ.2)ELMTS=ELMTS*(PUN(L)+2)-PUN(L)	023850
IF(KFLAG)PRINT(50,*)"VICTIM ELMTS = ",ELMTS	023860
IF(ELMTS.LT.1)GO TO 880	023870
C.....SET NO. OF TARGETS	023880
VISDEG=PICKET(MAXR,IVIS,2)	023890
IF(PICKET(MAXR,IVIS,1).NE.0..AND.ITPASS.EQ.1)VISDEG=	023900
1 PICKET(MAXR,IVIS,1)	023910
VICTIM=ELMTS*OPERA(K-10,L)-VISDEG*ACO(J)	023920
IF(KFLAG)PRINT(50,*)"VICTIMS = ",VICTIM	023930
IF(VICTIM.EQ.0.)GO TO 900	023940
IPSN=1	023950
IF(L.EQ.10)IPSN=2	023960
IF(IPSN.EQ.1)JPSN=2	023970
IF(IPSN.EQ.2)JPSN=1	023980
IF(L.EQ.2)GO TO 470	023990
GO TO (510,510,510,510,510,480,480,480,480,480,490,500,500,500,500,	024000
1 880,490,500,490,490,500,500,500),K-10	024010
470 GO TO (510,510,510,510,510,480,480,480,480,480,490,490,500,500,	024020
1 500,500,490,490,500,500,500,500),K-10	024030

480 N=1	124060
GO TO 520	124061
490 N=2	124062
GO TO 520	124063
500 N=3	124064
GO TO 520	124065
510 N=4	124100
520 KK=1	124110
IF(K.GT.15)KK=4	124120
IF(K.EQ.13.OR.K.GT.20)KK=3	124130
IF(K.GT.24.AND.K.NE.27)KK=2	124140
IF(KFLAG)PRINT(50,*)"VUL CAT (N), KK = ",N, KK	124150
LL=1	124160
IF(LL.EQ.1)LL=2	124170
AKILL=1.	124180
SKILL=0.	124190
C	124200
C..FIRER LOOP..FIRER LOOP..FIRER LOOP..FIRER LOOP..FIRER LOOP..FIRER LOOP	124210
C	124220
C FIRER LOOP	124230
C	124240
C..FIRER LOOP..FIRER LOOP..FIRER LOOP..FIRER LOOP..FIRER LOOP..FIRER LOOP	124250
C	124260
DO 430 I=1,30	124270
IFLAG=.FALSE.	124280
IF(IJFLAG.AND.KFLAG.AND.I.GE.ITEST1.AND.I.LE.ITEST2)IFLAG=.TRUE.	124290
IF(IFLAG)PRINT(50,*)"---FIRER NO. = ",I	124300
PKILL(I-10)=1.	124310
IF(ITHERM(I-10,J).NE.2.AND.ITPASS.EQ.2)GO TO 430	124320
C.....CHECK NO. OF TARGETS	124330
VISDEG=PICKET(MAXR,IVIS,ITHERM(I-10,J))	124340
IF(IFLAG.AND.VISDEG.EQ.0.)PPINT(50,*)"FIRER CANT SEE"	124350
IF(VISDEG.EQ.0.)GO TO 430	124360
INC=1+(I-1)*4	124370
C IF NO AMMO, GET OUT	124380
IF(J.EQ.1.AND.IBANNO(I-10).EQ.0) GO TO 420	124390
IF(J.EQ.2.AND.IBANNO(I-10).EQ.0) GO TO 420	124400
IF(J.EQ.2.AND.I.EQ.21.AND.MAXP.GE.5)GO TO 420	124410
IF(J.EQ.1A.AND.IMOUNT.EQ.2.AND.(I.LT.15.OR.(I.EQ.15.AND.J.EQ.1))	124420
1)GO TO 420	124430
ELS=(ELMT(I,J)-PLOSS(I,J,1))*PCOM(J)	124440
IF(ELS.LT.1)GO TO 420	124450
IF(J.EQ.1)KT=I-10	124460
IF(J.EQ.2)KT=I	124470
IF(I.GE.21.AND.J.EQ.1)KT=3	124480
IF(J.EQ.1.AND.KT.GT.10)GO TO 420	124490
IF(J.EQ.2.AND.KT.EQ.30)GO TO 420	124500
IF(J.EQ.2.AND.KT.EQ.25)KT=15	124510

	INX=1	124520
	IF(I.GE.16.AND.I.LE.29.AND.K.GE.16.ANC.(K.LE.29.OR.K.EQ.33))	124530
1	INX=2	124540
	IF(J.EQ.1.AND.I.GE.21.AND.I.LE.30.ANC.PAYR.GT.3) INX=2	124550
	IF(J.EQ.1.AND.I.GE.21.AND.I.LE.30.ANC.K.GE.16.ANC.K.LE.29.AND.	124560
1	MAXR.LE.3) INX=2	124570
	IF(J.EQ.2.AND.I.GE.21.AND.I.LE.30.ANC.MAXR.GT.1) INX=2	124580
	IF(J.EQ.2) GO TO 540	124590
	IF(I.LT.16.OR.I.GT.29) GO TO 530	124600
C	---BLUE---	124610
C	TANK FIRER	124620
	ISUP=1	124630
	ITYP=IRAMMO(I-10)	124640
	IF(N.EQ.1.AND.I.EQ.16) ITYP=1	124650
	IF(N.EQ.1.AND.I.EQ.17) ITYP=1	124660
	IF(N.EQ.1.AND.I.EQ.18) ITYP=7	124670
	IF(N.EQ.1.AND.I.EQ.19) ITYP=4	124680
	IF(N.EQ.1.AND.I.EQ.20) ITYP=1	124690
	GO TO 560	124700
C	OTHER THAN TANK FIRER	124710
530	CONTINUE	124720
	ISUP=2	124730
	ITYP=IRAMMO(I-10)	124740
	IF(MAXR.LE.3.AND.N.NE.1.ANC.(I.EQ.21.OP.I.EQ.26.OR.I.EQ.29))	124750
1	ITYP=13	124760
	GO TO 560	124770
C	---RED---	124780
C	TANK FIRER	124790
540	CONTINUE	124800
	IF(I.LT.16.OR.I.GT.19) GO TO 550	124810
	ISUP=1	124820
	ITYP=IRAMMO(I-10)	124830
	IF(N.EQ.1.AND.I.EQ.16) ITYP=1	124840
	IF(N.EQ.1.AND.I.EQ.17) ITYP=1	124850
	IF(N.EQ.1.AND.I.EQ.18) ITYP=3	124860
	IF(N.EQ.1.AND.I.EQ.19) ITYP=5	124870
	GO TO 560	124880
C	OTHER THAN TANK FIRER	124890
550	CONTINUE	124900
	ISUP=2	124910
	ITYP=IRAMMO(I-10)	124920
	IF(MAXR.LE.2.AND.N.NE.1.AND.(I.EQ.21.OP.I.EQ.29)) ITYP=7	124930
	IF(MAXR.LE.1.AND.N.NE.1.AND.I.EQ.22) ITYP=7	124940
560	IF(I.LE.15) ISUP=3	124950
	PNSMK=PUN(J)	124960
	IF(ITHERM(I-10,J).EQ.2.AND.ISMOKE(1).NE.2) PNSMK=PUN(L)	124970
	IF(ITHERM(I-10,J).EQ.2.AND.ISMOKE(2).EQ.2) PNSMK=PUN(J+2)	124980
1	PUN(L)/PUN(L+2)	124990

	IF (ITPASS.EQ.2.AND.ISMOKE(2).NE.2)PNSMK=1.-PUN(L)	025005
	IF (ITPASS.EQ.2.AND.ISMOKE(2).EQ.2)PNSMK=PUN(J+2)*	025010
1	(1.-PUN(L)/PUN(L+2))	025020
	IN=EL*OPEPA(I-10,J)*PNSMK	025030
	IAM=INC+INX+1	025040
	IF (K.EQ.22.AND.L.EQ.1)KK=2	025050
	IF (ITPASS.EQ.1)XFCF=FCF(L)*PUN(L)	025060
	IF (ITPASS.EQ.2.AND.ISMOKE(2).NE.2)XFCF=FCF(L)*(1.-PUN(L))	025070
	IF (ITPASS.EQ.2.AND.ISMOKE(2).EQ.2)XFCF=FCF(L)*(PUN(L+2)-PUN(L))	025080
	FIRE2=FIRE(MAXR,KT,ITHER*(I-10,J))	025090
	IF (ITHER*(I-10,J).EQ.1)GO TO 565	025100
	IF (PUN(1).NE.1..CR.PUN(2).NE.1.)GO TO 565	025110
C		025120
C	CHANGE THERMALS TO OPTICS IF NO SMOKE PRESENT	025130
C		025140
	FIRE1=FIRE(MAXR,KT,1)	025150
	IF (FIRE1.LE.FIRE2)GO TO 565	025160
	IF (PICKET(MAXR,IVIS,1).EQ.0.)GO TO 565	025170
	PIPE2=PIPE1	025180
565	CONTINUE	025190
	IF (IENGAG.NE.1.AND.J.EQ.ID)PIPE2=PIPE2*FCX(MAXR,ITER*N)	025200
	SUP=1.-SF(J)*SUPCE(IISUP)	025210
	IF (SUP.LE..2)SUP=.2	025220
	ROUNDS=PIPE2*SUP*ELMTS*	025230
1	IPERA(K-10,L)*MTS(KK,LL)/XFCF	025240
	IF (ROUNDS.LE.0.)GO TO A20	025250
	IF (J.EQ.2)GO TO 570	025260
	GO TO (640,630,620,790,620,600,610,590,540,670,710,820,640,820,	025270
3	820,710,620,710,820,920).I-10	025280
C	ALLOWS PLAY OF SU100 INSTEAD OF JSU152	025290
570	GO TO (690,670,660,820,770,650,650, 820,720,820,700,700,750,	025300
2	790,730,740,760,730,820,740).I-10	025310
580	N=1	025320
	GO TO 800	025330
590	M=2	025340
	GO TO 800	025350
600	N=3	025360
	GO TO 800	025370
610	N=4	025380
	GO TO 800	025390
620	N=5	025400
	GO TO 800	025410
630	N=6	025420
	GO TO 800	025430
640	N=7	025440
	GO TO 800	025450
650	N=8	025460
	GO TO 800	025470

660	M=9	025480
	GO TO 800	025490
670	M=10	025500
	GO TO 800	025510
680	M=11	025520
	GO TO 800	025530
690	M=12	025540
	GO TO 800	025550
700	M=13	025560
	GO TO 800	025570
710	M=14	025580
	GO TO 800	025590
720	M=15	025600
	GO TO 800	025610
730	M=16	025620
	GO TO 800	025630
740	M=17	025640
	GO TO 800	025650
750	M=18	025660
	GO TO 800	025670
760	M=19	025680
	GO TO 800	025690
770	M=20	025700
	GO TO 800	025710
780	M=21	025720
	GO TO 800	025730
790	M=22	025740
800	IF(J.EQ.2.AND.I.EQ.21.AND.MXR.LE.2) M=10	025750
	NBR=INDEX5(2,MXR,N,M,0,2,6,4,0)	025760
	IFIND=(NBR-1)/32+1	025770
	CALL READMS(3,SSKP,32,IFIND)	025780
	INX=NBR-(NBR/32*32)	025790
	IF(INCX.EQ.0) INOX=32	025800
	SS=SSKP(INOX)	025810
	IF(L.EQ.10) SS=SS*2.	025820
	IF(K.EQ.22.AND.L.EQ.1) N=3	025830
	NBR=INDEX5(1,MXR,N,M,0,2,6,4,0)	025840
	IFIND=(NBR-1)/32+1	025850
	CALL READMS(3,SSKP,32,IFIND)	025860
	INOX=NBR-(NBR/32*32)	025870
	IF(INCX.EQ.0) INOX=32	025880
	SSS=SSKP(INOX)	025890
	IF(L.EQ.10) SSS=SSS*2.	025900
	SS=(SS+SSS)/3.	025910
C	INCREASE VUL OF OLD TANKS	025920
	IF(K.LT.17.OR.K.GT.20) GO TO 810	025930
C.....	NEW ARMOR FULLY USED	025940
	IF(J.EQ.2.AND.K.EQ.19.AND.IAPCR.EQ."Y") GO TO 810	025950

```

IF(I.LT.16.OR.I.GT.20) SS=2*SS 125960
IF(J.EQ.1.AND.I.EQ.19) SS=2*SS 125970
IF(J.EQ.1.AND.I.EQ.19) SS=SS+.1 125980
IF(J.EQ.2.AND.I.GE.16.AND.I.LE.20) SS=SS+.1 125990
810 CONTINUE 126000
C 126010
C SS=SS*2 126120
C IF(SS.GT.1)SS=1 126030
C IF(SS/VICTIM.GT.1)GO TO 420 126040
IEVE=1 126150
IF(J.EQ.10)IEVE=2 126160
ABORT=1. 126070
IF(J.EQ.1.AND.ITYP.GE.8.AND.ITYP.LE.11)ABORT=AQUST 125980
IF(J.EQ.2.AND.ITYP.GE.9.AND.ITYP.LE.9)ABORT=AQUST 126090
C DEGRADE FOP FALSE TARGETS 126100
C IF(J.EQ.1)ABORT=ABORT*.7 126110
C IF(J.EQ.2)ABORT=ABORT*.3 126120
C 126130
PKILL(I-10)=(1.-SS/VICTIM)**(XN*ABORT*ROUNDS-1.) 126140
AKILL=AKILL*PKILL(I-10) 126150
SKILL=SKILL*(1.-PKILL(I-10)) 126160
SHOTSV(ITYP,J)=SHOTSV(ITYP,J)+XN*ROUNDS 126170
IF(IFLAG)PRINT(50,*)"PKILL,SHOTS SV = ",PKILL(I-10), 126180
1 SHOTS SV(ITYP,J) 126190
IF(J.EQ.1)AM(IAM)=AM(IAM)+XN*ROUNDS 126200
IF(J.EQ.1)AM(INC)=AM(INC)+(1-CPEGA(I-10,J)*PNSHK*PCOM(J)) 126210
820 CONTINUE 126220
830 CONTINUE 126230
C..END OF FIRE LOOP..END OF FIRE LOOP..END OF FIRE LOOP..END OF FIRE 126240
C 126250
TKILL=(1.-AKILL)*VICTIM 126260
IF(IFLAG)PRINT(50,*)"TKILL,AKILL = ",TKILL,AKILL 126270
IF(TKILL.LE.0)GO TO 840 126280
IF(SKILL.LE.0)GO TO 840 126290
C 126300
C.....KILLS ALLOCATION LOOP 126310
C 126320
DO 870 I=11,30 126330
IFLAG=.FALSE. 126340
IF(JFLAG.AND.(FLA(ANC.I.GE.ITEST1.AND.I.LE.ITEST2))IFLAG=.TRUE. 126350
IF(IFLAG)PRINT(50,*)"---FIRE NO. = ",I 126360
AKILL=TKILL*(1.-PKILL(I-10))/SKILL 126370
AKILL=IFIX(AKILL*10.+5)/10. 126380
ALOSS(I,K)=ALOSS(I,K)+IFIX(AKILL*10.+0.01)*PACK(L) 126390
PLOSS(K,L,2)=PLOSS(K,L,2)+IFIX(AKILL*10.+0.01)/10. 126400
IF(K.GT.15)GO TO 840 126410
ALOSS(I,3)=ALOSS(I,3)+IFIX(AKILL*ATCPH(K-10)*10.+0.01)*PACK(L) 126420
PLOSS(3,L,2)=PLOSS(3,L,2)+IFIX(AKILL*ATCPH(K-10)*10.+0.01)/10. 126430

```

IF(K,LT,13)GO TO 460	026440
440 ALOSS(I,2)=ALOSS(I,2)+IFIX(AKILL*CREWS(K-12,L)*10+.031)*PACK(L)	026450
PLOSS(I,L,2)=PLOSS(I,L,2)+IFIX(AKILL*CREWS(K-12,L)*10+.031)/10.	026460
IF(K.NE.23.AND.K.NE.24.AND.K.NE.25.AND.K.NE.29)GO TO 460	026470
IF(IMOUNT.EQ.1.OR.L.EQ.10)GO TO 460	026480
AKILL=AKILL*6.	026490
GO 450 KK=3,15	026500
IF(ELMT(KK,L).LE.0)GO TO 850	026510
ALOSS(I,KK)=ALOSS(I,KK)+IFIX(AKILL*ELT(KK)*10+.031)*PACK(L)	026520
PLOSS(KK,L,2)=PLOSS(KK,L,2)+IFIX(AKILL*ELT(KK)*10+.031)/10.	026530
850 CONTINUE	026540
860 IF(IQV2.EQ.0)PRINT *, "OVLY2-2 I,AKILL,ALCSS,PLOSS "	026550
1 ,I,AKILL,ALCSS(I,K),PLOSS(K,L,2)	026560
870 CONTINUE	026570
C..END OF KILLS ALLOCATION LOOP..END OF KILLS ALLOCATION LOOP..END OF C	026580
880 IF(IQV2.EQ.0)PRINT *, "OVLY2-3 L,IA,IMOUNT,K,ELMT,PLOSS,PCOM,FUN	026590
1 ,L,IA,IMOUNT,K,ELMT(K,L),PLOSS(K,L,1),PCOM(L),FUN(L)	026600
2 ,TKILL,SKILL	026610
890 CONTINUE	026620
900 CONTINUE	026630
C..END OF TARGET LOOP..END OF TARGET LOOP..END OF TARGET LOOP..END OF	026640
C	026650
910 CONTINUE	026660
C..END OF SIDE LOOP..END OF SIDE LOOP..END OF SIDE LOOP..END OF SIDE LOOP	026670
C	026680
ITPASS=ITPASS+1	026690
IF(ITPASS.EQ.2)GO TO 460	026700
C..END OF OBSCURATION LOOP..END OF OBSCURATION LOOP..END OF OBSCURATION	026710
C	026720
920 DO 1000 J=1,2	026730
IFLAG2=0	026740
DO 1000 I=1,32	026750
INX=I	026760
PLOSS(I,J,1)=PLOSS(I,J,2)+PLOSS(I,J,1)	026770
IF(PLOSS(I,J,1).LE.ELMT(INX,J))GO TO 960	026780
PRINT *, "ALL OF ELMT ",INX, " IN FORCE ",J, " HAVE BEEN KILLED"	026790
GO 950 K=11,30	026800
BLUE=IFIX(ALOSS(K,INX)/PACK(1))/10.	026810
RED=(ALOSS(K,INX)-IFIX(ALCSS(K,INX)/PACK(1))*PACK(1))/10.	026820
IF(J.EQ.2)GO TO 930	026830
BLUE=IFIX(BLUE*ELMT(INX,J)/PLOSS(I,J,1)*10+.5)/10.	026840
GO TO 940	026850
930 RED=IFIX(RED*ELMT(INX,J)/PLOSS(I,J,1)*10+.5)/10.	026860
940 ALOSS(K,INX)=BLUE*PACK(1)*10+RED*10.	026870
950 CONTINUE	026880
PLOSS(I,J,1)=ELMT(INX,J)	026890

960	PLOSS(I,J,2)=1.	126922
	IF(PLOSS(I,J,1).LT..1)GO TO 1003	126932
	IF(IFLAG2.EQ.1)GO TO 980	126942
	PRINT *, " "	126952
	KIND="RED"	126962
	IF(J.EQ.1)KIND="BLUE"	126972
	IF(IRUN.NE.1)PRINT 970,KIND	126982
970	FORMAT(" ",1X,A4," LOSSES TO THIS POINT",/,7X,"ITE4 * LOST")	126992
980	TKILL=PLOSS(I,J,1)	127002
	IF(IRUN.NE.1)PRINT 990,INX,TKILL	127012
990	FORMAT(" ",7X,I2,5X,F6.1)	127120
	IFLAG2=1	127130
1000	CONTINUE	127140
	NWO=NWO+7	127150
	GO TO 210	127160
C	OUTPUT RESULTS.	127172
1010	IF(IRUN.EQ.1)GO TO 1130	127182
	PRINT 1020	127192
1020	FORMAT("I")	127100
	PRINT *, "-----ARMOR ASSESSMENTS-----"	127110
	PRINT 1030	127120
1030	FORMAT(" I",24X,"I")	127130
	DO 1120 J=1,2	127140
	L=1	127150
	IF(J.EQ.L)L=2	127160
	IFLAG2=0	127170
	DO 1110 K=1,32	127180
	AKILL=PLOSS(K,L,1)	127190
	CLOST=0.	127200
	IF(K.LT.16.AND.K.NE.13)GO TO 1040	127210
	CLOST=AKILL*CPWS(K-12,L)	127220
1040	IF(AKILL.LT..1.AND.CLOST.LT..1)GO TO 1110	127230
	IF(IFLAG2.EQ.1)GO TO 1070	127240
	IF(J.EQ.2)GO TO 1050	127250
	PRINT *, "I TOTAL RED LOSSES I"	127260
	GO TO 1060	127270
1050	PRINT *, "I TOTAL BLUE LOSSES I"	127280
1060	IFLAG2=1	127290
	PRINT *, "I ITEM * LOST CREW I"	127300
	PRINT 1030	127310
1070	IF(K.LT.16.AND.K.NE.13)GO TO 1080	127320
	PRINT 1100,K,AKILL,CLOST	127330
	GO TO 1110	127340
1080	PRINT 1090,K,AKILL	127350
1090	FORMAT(" I",4X,I2,3X,F6.1,10X,"I")	127360
1100	FORMAT(" I",4X,I2,F9.1,F6.1,4X,"I")	127370
1110	CONTINUE	127380
	PRINT 1030	127390

IF(J.EQ.2)GO TO 1129	027400
PRINT *, "-----"	027410
PRINT 1030	027420
1120 CONTINUE	027430
PRINT *, "-----"	027440
PRINT 1029	027450
1130 IF(IRUN.EQ.1)PRINT *, " ARMOR ASSESSMENT PRINTED HERE"	027460
CALL CLOSMS(3)	027470
INX=0	027480
CALL LOSS(11,30,1,32,INX)	027490
IF(INX.EQ."YES")CALL CKSTOR(AH1,IPUT)	027500
C	027510
IPUT=-1	027520
AH1="INPUT 2"	027530
CALL CKSTOR(AH1,IPUT)	027540
DO 1135 I=7,90	027550
AH(I)=ARRAY(I)	027560
1135 CONTINUE	027570
CALL CKSTOP(AH1,IPUT)	027580
C.....TRANSFER OF SP 122 WEAPONS BACK TO ORIGINAL ARRAY	027590
1140 ELMT(50,2)=ELMT(50,2)+ELMT(25,2)	027600
ELMT(25,2)=0	027610
DO 1150 I=1,INAKV	027620
ALOSS(I,50)=ALOSS(I,50)+ALOSS(I,25)	027630
ALOSS(I,25)=0	027640
1150 CONTINUE	027650
C	027660
1160 FORMAT(" INCORRECT - RESPONSE MUST BE GE (.0 AND LE 1.0")	027670
END	027680

SUBROUTINE INFAN*	127702
COMMON IA,IO,IP,IEGAG,ITERM,IVIS,INCUNT,MINES,CPPR,FSFPR,FP,	127710
1 ATIME,TFIRST,TPUN,NORC,FMAS,IMAXKV,	127720
2 SF(2),FSSF(2),PACK(2),	027730
3 ELMT(AQ,2),ALOSS(66,66),SHOTS(55,2),CKILL(53,2),SHOTSV(55,2)	027740
COMMON/REED/JDAY1,XIN(4),ICARD(20),IARPOP	027750
COMMON/DATA/FPS(80,2),CREWS(53,2),APOS(12),CPOS(6),ITERM(20,2),	027760
1 PSN(6,2,2),PLT(15),KEY(171)	027770
COMMON/ONE/LFIT(35),ARRAY(90),MYBUF(1024),C(80,2),ACI,	027780
1ASCENE,ASECT	027790
COMMON/THREE/THYST(35),AH(90),TBUF(1024)	027800
DIMENSION GFPS(2),TABLE(7,2,4),STALE(5),FAC(2)	027810
1,TABLE3(5,2)	027820
DIMENSION F(2)	027830
DATA ((TABLE(I,J,K),J=1,2),I=1,7),K=1,4)/	027840
1,12,.05,.08,.06,.06,.06,.05,.08,.05,.08,.05,.09,.04,.10,.09,.16,.00	027850
25,.35,.04,.07,.04,.07,.03,.09,.03,.13,.12,.11,.04,.05,.35,.34	027860
3,.36,.03,.06,.03,.08,.02,.09,.02,.19,.08,.02,.35,.03,.13,.13	027870
44,.03,.05,.02,.06,.02,.36/	027880
DATA (STALE(I),I=1,4)/1,.2,.2,6,3,1/	027890
DATA ((TABLE3(I,J),J=1,2),I=1,5)/10,.20,.20,.15,.35,.10,.51,.5,	027900
1 70,.2/	027910
ARRAY(4)=75	027920
ARRAY(5)=90	027930
DO 90 I=6,90	027940
90 ARRAY(I)=0	027950
100 PRINT *, "DO YOU WISH TO PROCESS INFANTRY ASSESSMENTS?"	027960
CALL REEDA(INX)	027970
IF(INX.EQ."Y")GO TO 110	027980
IF(INX.EQ."N")GO TO 510	027990
GO TO 100	028000
110 PRINT *, " INFANTRY ASSESSMENTS"	028010
PRINT *, " "	028020
ARRAY(75)=1	028030
120 DO 160 J=1,2	028040
KIND="BLUE"	028050
IF(J.EQ.2)KIND="RED"	028060
130 PRINT 140,KIND	028070
140 FORMAT(" ENTER FRACT OF ",A4," MANEUVER FORCES COMMITTED(MAX 1.)")	028080
CALL REED4	028090
F(J)=XINX(1)	028100
ARRAY(75+J)=F(J)	028110
IF(F(J).GT.0..AND.F(J).LE.1,GO TO 160	028120
PRINT 150	028130
150 FORMAT("INCORRECT RESPONSE - TRY AGAIN")	028140
GO TO 130	028150
160 CONTINUE	028160
170 PRINT *, "DO TANKS SUPPORT THE DISMOUNTED INFANTRY IN THIS SECTOR"	028170

CALL REEDA(INX)	028180
IF(INX.EQ."Y")ARRAY(78)=1	028190
IF(INX.EQ."Y")GO TO 210	028200
IF(INX.EQ."N")GO TO 210	028210
GO TO 170	028220
210 INDEX=9	028230
IF(INX.EQ."Y")INDEX=15	028240
AIL=0.	028250
AIL=0.	028260
DO 220 IEL=1,2	028270
GFPS(IEL)=0.	028280
FAC(IEL)=1.	028290
IF(ELMT(3,IEL)*F(IEL).GT.72.)FAC(IEL)=2.	028300
DO 220 I=6,30	028310
IF(INX.EQ."N".AND.I.GE.16.AND.I.LE.20)GO TO 220	028320
GFPS(IEL)=GFPS(IEL)+ELMT(I,IEL)*FAC(I,IEL)	028330
220 CONTINUE	028340
IF(GFPS(ID).GE.1.)GO TO 230	028350
PRINT *, "THERE ARE NO DEFENCES--ASSESSMENTS CANNOT BE MADE."	028360
GO TO 510	028370
230 PRINT *, "ENTER # HOURS OF INFANTRY ATTACK (MAX = 6)."	028380
CALL REEDA	028390
HR=INX(1)	028400
ARRAY(79)=HR	028410
HRC=HR	028420
IF(HR.GT.0..AND.HR.LE.6.)GO TO 240	028430
PRINT 150	028440
GO TO 230	028450
240 PRINT *, "ARE AMBUS+ TACTICS BEING EMPLOYED"	028460
CALL REEDA(INX)	028470
IF(INX.EQ."Y")ARRAY(80)=1	028480
IF(INX.EQ."Y")GO TO 360	028490
IF(INX.EQ."N")GO TO 250	028500
GO TO 240	028510
250 GFPR=GFPS(IA)*APOS(IP)/(GFPS(ID)*CPOS(IENGAG))	028520
DO 260 I=1,6	028530
IF(GFPR.LE.I*.5)GO TO 270	028540
260 CONTINUE	028550
I=7	028560
270 INX=I	028570
GO TO (300,310,310,280,290,290),IENGAG	028580
290 I=1	028590
GO TO 320	028600
290 I=2	028610
GO TO 320	028620
300 I=3	028630
GO TO 320	028640
310 I=4	028650

320	HR=HRC	029560
	IF(HR.LE.0)GO TO 430	029570
	IFLAG=0	029580
	ATRT=TABLE(INX,1,I)/FAC(IA)	029590
	OTRT=TABLE(INX,2,I)/FAC(ID)	029600
330	IF(IA.EQ.1)GO TO 340	029610
	AT=ELMT(3,2)*F(2)-AIL	029620
	OT=ELMT(3,1)*F(1)-OIL	029630
	GO TO 350	029640
340	AT=ELMT(3,1)*F(1)-AIL	029650
	OT=ELMT(3,2)*F(2)-OIL	029660
350	AIL=AIL+AT*(1.-(1.-ATRT)**HR)	029670
	OIL=OIL+OT*(1.-(1.-OTRT)**HR)	029680
	IF(IFLAG.NE.0)GO TO 250	029690
	GO TO 430	029700
360	HRC=HR-1.	029710
	IF(HR.GT.1)HF=1.	029720
	PRINT *, "IS BLUE AMBUSHING RED"	029730
	CALL REEDA(IFLAG)	029740
	IF(IFLAG.EQ."Y")GO TO 370	029750
	IF(IFLAG.EQ."N")GO TO 390	029760
	GO TO 360	029770
370	GFPR=4.5*GFPS(1)/GFPS(2)	029780
	IAA=1	029790
	GO TO 390	029800
380	GFPR=4.5*GFPS(2)/GFPS(1)	029810
	IAA=2	029820
390	DO 400 I=1,4	029830
	IF(GFPR.LT.STALE(I))GO TO 410	029840
400	CONTINUE	029850
	I=5	029860
410	A=TABLE3(I,2)/100.	029870
	OF=TABLE3(I,1)/100.	029880
	IF(IAA.NE.1A)GO TO 420	029890
	ATRT=A	029900
	OTRT=OF	029910
	GO TO 330	029920
420	ATRT=OF	029930
	OTRT=A	029940
	GO TO 330	029950
430	IF(IRUN.EQ.1)GO TO 500	029960
	PRINT *, "----INFANTRY ASSESSMENTS----"	029970
440	FORMAT(" I",25X," I")	029980
	DO 490 J=1,2	029990
	PRINT 440	030000
	IFLAG=0	030010
	KIND="BLUE"	030020
	L=1	030030

IF(L.NE.J)GO TO 452	129140
KIND="REC"	129150
L=2	129160
458 DO 490 I=1,INDEX	129170
IF(ELMT(I,L).LE.0.)GO TO 490	129180
A=AIL	129190
IF(L.EQ.10)A=0IL	129200
IF(I.EQ.2.OR.I.EQ.4.OR.I.EQ.5)GO TO 490	129210
IF(I.EQ.7.AND.L.EC.2)GO TO 490	129220
A=A*PL*(I)	129230
A=IFIX(A*10.+.5)/10.	129240
ALOSS(3,I)=ALOSS(3,I)+IFIX(A*13.+.301)*PACK(L)	129250
IF(R.LT..1)GO TO 490	129260
IF(IFLAG.EQ.1)GO TO 470	129270
PRINT 460,KIND	129280
460 FORMAT(" I",3X,A4,1X,"INFANTRY LOSSES",3X,"I")	129290
PRINT *, "I ITEM ALOST "	129300
IFLAG=1	129310
470 PRINT 480,I,A	129320
480 FORMAT(" I",5X,I2,7X,F6.1,6X,"I")	129330
490 CONTINUE	129340
PRINT 440	129350
PRINT *, "*****"	129360
500 IF(IRUN.EQ.1)PRINT *, " TOTAL INFANTRY LOSSES PRINTED HERE"	129370
INX=0	129380
CALL LOSS(3,3,1,15,INX)	129390
C	129400
AM1="INPUT 36"	129410
IPUT=-1	129420
CALL CKSTOR(AM1,IPUT)	129430
IF(INX.EQ."NO")GO TO 510	129440
DO 505 I=75,90	129450
505 AM(I)=ARPAY(I)	129460
CALL CKSTOR(AM1,IPUT)	129470
510 END	129480

SUBROUTINE MINE	029500
COMMON I4, I0, IP, IENGAG, IERRN, IVIS, IMCUNT, MINES, CFPP, FSPP, FPP,	029510
1 ATIME, IFRST, IRUN, IORD, FMASS, IMAXV.	029520
2 SF(2), FSSF(2), PACK(2),	029530
3 FLMT(80,2), ALOSS(66,66), SHOTS(55,2), CKILL(53,2), SHOTSV(55,2)	029540
COMMON/REED/JDAY1, XINY(4), ICAFD(20), IARMOR	029550
COMMON/DATA/FPS(10,2), CREMS(51,2), AFOS(12), CPDS(6), ITHEF=(20,2),	029560
1 PSN(6,2,2), PLT(15), KEY(171)	029570
COMMON/THREE/THIST(35), AM(90), IY9UF(1024)	029580
DIMENSION ATOEN(5), ATFAC(5), HREQ(5), FLOSS(32,2,2)	029590
DATA IATDEN(J), J=1,5)/.2, .5, 1, 2, 3, /	029600
DATA (ATFAC(J), J=1,5)/.39, .29, .52, .63, .75/	029610
DATA (HREQ(J), J=1,5)/.23, .279, .323, /	029620
110 FORMAT(NUMBER NOT WITHIN DOCTRINES BOUNCARY - TRY AGAIN")	029630
AM1="INPUT 4"	029640
IPUT=-1	029650
CALL CKSTOR(AM1, IPUT)	029660
M1=11	029670
AM(10)=10	029680
140 PRINT *, "DO YOU WISH TO PROCESS MINE ASSESSMENTS?"	029690
CALL REEDA(INX)	029700
IF(INX.EQ."Y") GO TO 150	029710
IF(INX.EQ."N") GO TO 630	029720
GO TO 140	029730
150 PRINT *, "SELECT TYPE OF MINE EMPLOYMENT"	029740
CALL REED4	029750
IND=XINX(1)	029760
IF(M1.LE.90)AM(M1)=INC	029770
IF(IND.EQ.1)GO TO 160	029780
IF(IND.EQ.2)GO TO 170	029790
IF(IND.EQ.0)GO TO 620	029800
IF(IND.EQ.999)GO TO 160	029810
PRINT 110	029820
160 PRINT *, "FOR CONVENTIONAL MINES.....ENTER 1"	029830
PRINT *, " FASCAM MINES.....ENTER 2"	029840
PRINT *, " **TO END**.....ENTER 3"	029850
GO TO 150	029860
C	029870
C ASSUMES DEFENDER EMPLACED MINFIELD(S	029880
C	029890
170 CALL FASCAM(FLOSS)	029900
GO TO 150	029910
180 KIND="BLUE"	029920
IF(I0.EQ.2)KIND="RED"	029930
IOK=0	029940
IF(M1.GT.90)IOK=1	029950
PRINT *, "ARE MINES LAID PRIOR TO COMMENCEMENT OF HOSTILITIES?"	029960
CALL REEDA(INX)	029970

IF(INX.EQ."Y")GO TO 190	129940
IF(INX.EQ."N")GO TO 290	029990
GO TO 180	030000
190 WDEGF=.9	030010
GO TO 205	030020
200 WDEGF=.7	030030
205 IF(IOK.LE.0)AH(N1+1)=WDEGF	030040
210 PRINT 220,KIND	030050
220 FORMAT(" WILL ".A4," HAVE THE CAPABILITY TO EMPLOY MECHANICAL",	030060
1 " MINE PLANTERS?"	030070
CALL REED4(INX)	030080
IF(INX.EQ."Y") GO TO 230	030090
IF(INX.EQ."N") GO TO 280	030100
GO TO 210	030110
230 PRINT *, "ENTER NUMBER OF MECHANICAL MINE PLANTER PLATCONS (MAX 30)	030120
Z"	030130
C MECHANICAL EMPLACEMENT OF MINEFIELD	030140
CALL REED4	030150
BMPL=XINX(1)	030160
IF(IOK.LE.0)AH(N1+2)=BMPL	030170
IF(BMPL.GE.1.AND.BMPL.LE.30) GO TO 240	030180
PRINT 110	030190
GO TO 230	030200
240 PRINT *, "ENTER NUMBER OF AVAILABLE MINE PLANTER HOURS (MAX 300)"	030210
CALL REED4	030220
RNMPH=XINX(1)	030230
IF(IOK.LE.0)AH(N1+3)=RNMPH	030240
IF(RNMPH.GE.1.AND.RNMPH.LE.300)GO TO 250	030250
PRINT 110	030260
GO TO 240	030270
250 IF(KIND.EQ."RED")GO TO 260	030280
HOURS=6.	030290
STRIPW=2000.	030300
GO TO 270	030310
260 HOURS=2.	030320
STRIPW=1000.	030330
270 FRONT=(BMPL*RNMPH*WDEGF)/HOURS*STRIPW	030340
J=2	030350
GO TO 340	030360
280 PRINT *, "ENTER NUMBER OF MEN USED TO EMPLACE MINES (MAX 1000)"	030370
CALL REED4	030380
NUMEN=XINX(1)	030390
IF(IOK.LE.0)AH(N1+4)=NUMEN	030400
IF(NUMEN.GE.1.AND.NUMEN.LE.1000)GO TO 290	030410
PRINT 110	030420
GO TO 240	030430
290 PRINT *, "ENTER HOURS AVAILABLE FOR EMPLACEMENT OF MINES (MAX 300)"	030440
CALL REED4	030450

HRMAN=XINX(1)	030460
IF(IOK.LE.0)AH(N1+5)=HRMAN	030470
IF(HRMAN.GE.1.AND.HRMAN.LE.300)GO TO 300	030480
PRINT 110	030490
GO TO 290	030500
300 PHR=NUMEN*HRMAN*WDEGF	030510
310 PRINT *,"SELECT MINEFIELD DENSITY"	030520
CALL REED4	030530
J=XINX(1)	030540
IF(IOK.LE.0)AH(N1+6)=J	030550
IF(J.EQ.999)GO TO 320	030560
IF(J.GE.1.AND.J.LE.5)GO TO 330	030570
PRINT 110	030580
320 PRINT *,"FOR DENSITY .0013 MINE/SQ METER.....ENTER 1"	030590
PRINT *," .0033 MINE/SQ METER.....ENTER 2"	030600
PRINT *," .0066 MINE/SQ METER.....ENTER 3"	030610
PRINT *," .0130 MINE/SQ METER.....ENTER 4"	030620
PRINT *," .0290 MINE/SQ METER.....ENTER 5"	030630
GO TO 310	030640
330 FRONT=PHR/HRREQ(J)*100.	030650
FRONT=IFIX(FRONT+.5)	030660
IF(IOK.LE.0)AH(N1+7)=FRONT	030670
340 PRINT 150,FRONT	030680
350 FORMAT(" POTENTIAL MINEFIELD FRONTAGE IS ",F8.0)	030690
PRINT *," "	030700
PRINT *,"ENTER ACTUAL MF FRONTAGE (MAX=POTENTIAL)"	030710
CALL REED4	030720
AFRONT=XINX(1)	030730
IF(IOK.LE.0)AH(N1+8)=AFRONT	030740
IF(AFRONT.LE.FRONT)GO TO 360	030750
PRINT 110	030760
GO TO 340	030770
360 FRONT=AFRONT	030780
370 PRINT *,"ENTER FRACTION OF MINE FIELD NOT BYPASSED BY ATTACKER (MAX=1.0)"	030790
1X=1.0)	030800
CALL REED4	030810
PHFNBY=XINX(1)	030820
IF(IOK.LE.0)AH(N1+9)=PHFNBY	030830
IF(PHFNBY.EQ.999)GO TO 380	030840
IF(PHFNBY.GE.0..AND.PHFNBY.LE.1.)GO TO 390	030850
PRINT 110	030860
380 PRINT *,"FOR EXAMPLE: 0. MEANS ALL OF THE MF CAN BE BYPASSED"	030870
PRINT *,"OR, 1. MEANS NONE OF THE MF CAN BE BYPASSED"	030880
GO TO 370	030890
390 FROBY=FRONT*PHFNBY	030900
PRINT 400,FROBY	030910
400 FORMAT(" ENTER AMOUNT OF "AFFICABLE TERRAIN ("F8.0,"-10000. 1)"	030920
1)	030930

CALL REED4	130940
TRZONE=XINX(1)	130950
IF(IOK.LE.0)AH(N1+10)=TRZONE	130960
IF(1RZONE.GE.FROBY.AND.TRZONE.LE.100000) GO TO 417	130970
PRINT 110	130980
GO TO 390	130990
410 PRINT *, "ENTER AP MINE DENSITY (SQ METER) - (MIN=.013-MAX=.150)"	131000
CALL REED4	131010
X=XINX(1)	131020
IF(IOK.LE.0)AH(N1+11)=X	131030
IF(X.GE..13.AND.X.LE..160) GO TO 420	131040
PRINT 110	131050
GO TO 417	131060
420 INX=X*150.	131070
SHOTSX(16,10)=SHOTSX(16,10)+X*150*FRONT	131080
INX=INX/4+1	131090
PERCAS=(INX+1)/10.	131100
PERCOV=(FRONT*PMFNBY)/TRZONE	131110
IF(PERCOV.LE.1) GO TO 430	131120
PRINT *, "PERCENT OF ZONE COVERED MUST BE BETWEEN ZERO(0) AND"	131130
PRINT *, "ONE(1) - CHECK THE TRAFFICABLE ZONE DEFENDED TO SEE"	131140
PRINT *, "IF IT IS LARGER THAN THE RESULT OF MULTIPLYING MF"	131150
PRINT *, "FRONTAGE TIMES PERCENT OF MF NOT BYPASSED"	131160
GO TO 390	131170
430 PRINT *, "ENTER PERCENT (DECIMAL) OF FORCES ENTERING MF (MAX=.5)"	131180
CALL REED4	131190
P=XINX(1)	131200
IF(IOK.LE.0)AH(N1+12)=P	131210
IF(P.GE..0.AND.P.LE..5) GO TO 440	131220
PRINT 110	131230
GO TO 430	131240
440 DO 500 K=1,32	131250
IF(ELMT(K,1A)-PLOSS(K,1A,2).LE.0) GO TO 500	131260
IF(K.LT.16.AND.K.NE.3) GO TO 500	131270
IF(K.EQ.3.AND.IMOUNT.EQ.2) GO TO 500	131280
IF(K.NE.3) GO TO 450	131290
AKILL=PERCOV*(ELMT(3,1A)-PLOSS(3,1A,2))*P*PERCAS	131300
AKILL=IFIX(AKILL*10.+5)/10.	131310
GO TO 440	131320
450 AKILL=PERCOV*(ELMT(K,1A)-PLOSS(K,1A,2))*P*ATFAC(J)	131330
AKILL=IFIX(AKILL*10.+5)/10.	131340
ALOSS(5,K)=ALOSS(5,K)+IFIX(AKILL*10.+001)*PACK(1A)	131350
PLOSS(K,1A,1)=PLOSS(K,1A,1)+IFIX(AKILL*10.+001)/13.	131360
IF(K.EQ.1) GO TO 460	131370
CLOST=AKILL*CPWS(K-12,1A)	131380
GO TO 470	131390
460 CLOST=AKILL*2.	131400
470 ALOSS(5,2)=ALOSS(5,2)+IFIX(CLOST*10.+001)*PACK(1A)	131410

PLOSS(2,IA,1)=PLOSS(2,IA,1)+FIX(CLOST*11,+.331)/10.	031422
IF(IMOUNT.EQ.1.OR.(K.NE.21.AND.K.NE.25))GO TO 500	031430
AKILL=FIX(AKILL*60,+.5)/10.	031440
480 DO 490 KK=3,15	031450
IF(ELMT(KK,IA)-PLOSS(KK,IA,2).LE.0)GO TO 490	031460
ALOSS(5,KK)=ALOSS(5,KK)+FIX(AKILL*PLT(KK)*10,+.001)*PACK(IA)	031470
PLOSS(KK,IA,1)=PLOSS(KK,IA,1)+FIX(AKILL*PLT(KK)*11,+.331)/10.	031480
490 CONTINUE	031490
500 CONTINUE	031500
IF(IRUN.EQ.1)GO TO 600	031510
INX=0	031520
DO 590 K=1,32	031530
AKILL=PLOSS(K,IA,1)	031540
IF(AKILL.LT..1)GO TO 590	031550
IF(K.GE.2.AND.K.LE.15)GO TO 520	031560
IF(K.EQ.1)GO TO 510	031570
CLOST=AKILL*CREWS(K-12,IA)	031580
GO TO 520	031590
510 CLOST=AKILL*2.	031600
520 IF(INX.EQ.1)GO TO 550	031610
PRINT 530	031620
530 FORMAT('1')	031630
PRINT *, '-----MINEFIELD ASSESSMENTS-----'	031640
PRINT 540	031650
540 FORMAT('1',27X,'1')	031660
PRINT *, '1' ATTACKER LOSSES '1'	031670
PRINT *, '1' ITEM * LOST CREW '1'	031680
PRINT 540	031690
INX=1	031700
550 IF(K.GE.2.AND.K.LE.15)GO TO 570	031710
PRINT 560,K,AKILL,CLOST	031720
560 FORMAT('1',5X,I2,F9.1,F7.1,4X,'1')	031730
GO TO 590	031740
570 PRINT 580,K,AKILL	031750
580 FORMAT('1',5X,I2,F9.1,11X,'1')	031760
590 CONTINUE	031770
IF(INX.EQ.0)GO TO 600	031780
PRINT 540	031790
PRINT *, '-----'	031800
PRINT 530	031810
600 IF(IRUN.EQ.1)PRINT *, ' LOSSES TO MINEFIELD PRINTED HERE'	031820
DO 610 K=1,32	031830
DO 610 J=1,2	031840
PLOSS(K,J,2)=PLOSS(K,J,2)+PLOSS(K,J,1)	031850
PLOSS(K,J,1)=0.	031860
610 CONTINUE	031870
GO TO 150	031880
620 INX=0	031890

CALL LOSS(5,5,16,30,INX)
IF(INX.EQ."YES")CALL CKSTOR(AM1,IPUT)
630 END

031900
031910
031920

SUBROUTINE FASCAM(FLOSS)	032184
COMMON IA, ID, IP, IENGAG, ITERN, IVIS, IWCUNT, MINES, CFBP, FSPBP, FPP,	032185
1 ATIME, IFIRST, IRUN, IORD, FMASS, IMA)KV,	032186
2 SF(2), FSSP(2), PACK(2),	032187
3 ELMT(40,2), ALOSS(66,66), SHOTS(55,2), CKILL(53,2), SHOTSV(55,2)	032188
COMMON/REED/JOAY1, XINX(4), ICARD(20), IARMCR	032189
COMMON/DATE/FPS(40,2), CREWS(53,2), APOS(12), DPOS(6), ITHERR(20,2),	032190
1 PSN(6,2,2), PLT(15), KEY(171)	032191
DIMENSION FATCAS(3), FPCAS(3), FLOSS(32,2,2)	032192
DIMENSION FATDEN(3), FPDEN(3), DEPTH(3)	032193
DATA (FATCAS(I), I=1,3)/.56,.29,.2/	032194
DATA (FPCAS(I), I=1,3)/.40,.09,.0/	032195
DATA FATDEN /.001,.01,.001/	032196
DATA FPDEN /.001,.003,.0/	032197
DATA DEPTH/175,.250,.0/	032198
100 FORMAT(" NUMBER NOT WITHIN DOCTRINE BOUNDARY - TRY AGAIN")	032199
110 PRINT *, "SELECT TYPE OF FASCAM DELIVERY"	032200
CALL REED4	032201
II=XINX(1)	032202
IF(II.EQ.999)GO TO 120	032203
IF(II.GE.1.AND.II.LE.3)GO TO 130	032204
PRINT 100	032205
120 PRINT *, "FOR ARTILLERY.....ENTER 1"	032206
PRINT *, " GEMMS.....ENTER 2"	032207
PRINT *, " SLUMINES.....ENTER 3"	032208
GO TO 110	032209
130 PRINT *, "ENTER MINEFIELD FRONTAGE (MAX= 100000)"	032210
CALL REED4	032211
FRONT=XINX(1)	032212
IF(FRONT.GE.100..AND.FRONT.LE.100000)GO TO 140	032213
PRINT 100	032214
GO TO 130	032215
140 CONTINUE	032216
SHOTSV(16,1)=SHOTSV(16,1)+(FATDEN(II)*FPDEN(II))*DEPTH(II)*FRONT	032217
PRINT *, "ENTER FRACTION OF MF NOT BYPASSED BY ATTACKER (MAX=1.)"	032218
CALL REED4	032219
PMFNBY=XINX(1)	032220
IF(PMFNBY.EQ.999)GO TO 150	032221
IF(PMFNBY.GE..0.AND.PMFNBY.LE.1)GO TO 160	032222
PRINT 100	032223
150 PRINT *, "FOR EXAMPLE: 0. MEANS ALL OF MF CAN BE BYPASSED"	032224
PRINT *, "OR 1. MEANS NONE OF MF CAN BE BYPASSED"	032225
GO TO 140	032226
160 FROBY=FRONT*PMFNBY	032227
PRINT 170, FROBY	032228
170 FORMAT(" ENTER AMOUNT OF TRAFFICABLE EFFRAIN ("F.A.C."-10000). 4)	032229
1)	032230
CALL REED4	032231

TRZONE=XINX(1)	032410
IF (TRZONE.GE.FROBY.AND.TRZONE.LE.10000.)GO TO 180	032420
PRINT 110	032430
GO TO 160	032440
180 PRINT "ENTER PERCENT (DECIMAL) OF FORCE ENTERING WF (MAX=.5)"	032450
CALL REED4	032460
P=XINX(1)	032470
IF (P.GT.0..AND.P.LE..5)GO TO 190	032480
PRINT 100	032490
GO TO 180	032500
190 PERCOV=FFROBY/TRZONE	032510
DO 250 K=1,32	032520
IF (ELMT(K,IA)-PLOSS(K,IA,2).LE.0.)GO TO 250	032530
IF (K.LT.16.AND.K.NE.3)GO TO 250	032540
IF (K.EQ.3.AND.IMOUNT.EQ.2)GO TO 250	032550
IF (K.NE.3)GO TO 200	032560
AKILL=PERCOV*(ELMT(3,IA)-PLOSS(3,IA,2))*F*FPCAS(II)	032570
AKILL=IFIX(AKILL*10.+5)/10.	032580
GO TO 230	032590
200 AKILL=PERCOV*(ELMT(K,IA)-PLOSS(K,IA,2))*F*FPCAS(II)	032600
AKILL=IFIX(AKILL*10.+5)/10.	032610
ALOSS(5,K)=ALOSS(5,K)+IFIX(AKILL*10.+0(1))*PACK(IA)	032620
PLOSS(K,IA,1)=PLOSS(K,IA,1)+IFIX(AKILL*10.+0(1))/10.	032630
IF (K.EQ.1)GO TO 210	032640
CLOST=AKILL*CREWS(K-12,IA)	032650
GO TO 220	032660
210 CLOST=AKILL*2.	032670
220 ALOSS(5,2)=ALOSS(5,2)+IFIX(CLOST*10.+0(1))*PACK(IA)	032680
PLOSS(2,IA,1)=PLOSS(2,IA,1)+IFIX(CLOST*10.+0(1))/10.	032690
IF (IMOUNT.EQ.1.OR.(K.NE.21.AND.K.NE.25))GO TO 250	032700
AKILL=IFIX(AKILL*60.+5)/10.	032710
230 DO 240 KK=3,15	032720
IF (ELMT(KK,IA)-PLOSS(KK,IA,2).EQ.0.)GO TO 240	032730
ALOSS(5,KK)=ALOSS(5,KK)+IFIX(AKILL*FLT(KK)*10.+0(1))*PACK(IA)	032740
PLOSS(KK,IA,1)=PLOSS(KK,IA,1)+IFIX(AKILL*FLT(KK)*10.+0(1))/11.	032750
240 CONTINUE	032760
250 CONTINUE	032770
IF (IRUN.EQ.1)GO TO 350	032780
INX=0	032790
DO 340 K=1,32	032800
AKILL=PLOSS(K,IA,1)	032810
IF (AKILL.LT..1)GO TO 340	032820
IF (K.GE.2.AND.K.LE.15)GO TO 270	032830
IF (K.EQ.1)GO TO 260	032840
CLOST=AKILL*CREWS(K-12,IA)	032850
GO TO 270	032860
260 CLOST=AKILL*2.	032870
270 IF (INX.EQ.1)GO TO 300	032880

PRINT 280	032990
280 FORMAT("1")	032990
PRINT *, "----FASCOM ASSESSMENTS----	032990
PRINT 290	032990
290 FORMAT("I", 24X, "I")	032990
PRINT *, "I ATTACKER LOSSES I"	032990
PRINT *, "I ITEM LOST CREW I"	032990
PRINT 290	032990
INX=1	032990
300 IF(K.GE.2.AND.K.LE.15)GO TO 320	032990
PRINT 310,K,AKILL,CLOST	032990
310 FORMAT("I", 3X, I2, F9.1, F7.1, 3X, "I")	032990
GO TO 340	032990
320 PRINT 330,K,AKILL	032990
330 FORMAT("I", 3X, I2, F9.1, 10X, "I")	032990
340 CONTINUE	032990
IF(INX.EQ.0)GO TO 350	032990
PRINT 290	032990
PRINT *, "*****"	032990
PRINT 280	032990
350 IF(IRUN.EQ.1)PRINT *, "LOSSES TO MINEFIELD PRINTED HERE"	032990
DO 360 K=1,32	032990
DO 360 J=1,2	032990
PLOSS(K,J,2)=PLOSS(K,J,2)+PLOSS(K,J,1)	032990
PLOSS(K,J,1)=0.	032990
360 CONTINUE	032990
RETURN	032990
END	032990

SUBROUTINE CANNON	033180
COMMON IA, ID, IP, IENGAG, ITERN, IVIS, INCUNT, MINES, CFPR, FSPR, FPR,	033190
1 ATIME, IFIRST, IRUN, MORO, FPASS, ZMAXV,	033200
2 SF(2), FSSF(2), PACK(2),	033210
3 ELMT(80,2), ALOSS(66,66), SHOTS(55,2), CKILL(53,2), SHOTSV(55,2)	033220
COMMON/REED/JOAY1, XINX(4), ICAFO(20), IAFPCR	033230
COMMON/DATA/FPS(80,2), CREWS(53,2), APOS(12), OPOS(6), ITHE=H(20,2),	033240
1 PSN(6,2,2), PLT(15), KEY(171)	033250
COMMON/ONE/LFIT(35), ARRAY(90), MYBUF(1024), D(80,2), AC1,	033260
1ASCENE, ASECT	033270
COMMON/THREE/THIST(35), JH(90), IYBUF(1024)	033280
COMMON/EN/PCDGP(2,4), ICUNT(8), ISHOKE(2), FCSHO(2), PCSSN(2), ARC(2)	033290
1, NMU(2)	033300
DIMENSION TBAT(13,2), SUPR(4), ISHOT(13,2), FDF(5,2), PREP(2), MAP(55),	033310
1 PLOSS(55,2,2), HP(2), PERSE(2), RCF(13,3,2), RPM(13,2), FKILL(13),	033320
2 CBTLEV(6), TGT(17,2), CPRA(17,2), FOK(55,2), FCT(15,17,2), MW(17),	033330
3 TCS(2), TCS(2), CLEV(3,2), ANSF(2)	033340
DIMENSION IARTP(2)	033350
DIMENSION PCM(2), PICH(13,2), ICH(13,2), PCM1(2), OA(13,2)	033360
REAL MW, MNTM	033370
REAL MWFOP(17)	033380
DIMENSION TEMP(6,13), TEMP3(3,17)	033390
DIMENSION TEMP2(6,13), TEMP4(3,17)	033400
C	033410
C DATA MAPPING INDEX.	033420
C DATA (MAP(I), I=1,55)/	033430
1 6,3,1,9*0,2,2,1,1,3,4,4,4,3,10*9,	033440
3 9,8,8,8,10,10,10,8,8,7,7,7,10,11,10,9,9,11,11,12,	033450
5 12,11,11,12,12/	033460
C	033470
C INDIRECT FIRE WEAPON SUPPRESSION COEFFICIENTS.	033480
C DATA (SUPR(I), I=1,4)/3.52, 3*2.86/	033490
C	033500
C CALL OPENHS(3,KEY,171,0)	033510
C POK TABLE	033520
C CALL REACHS(3,POK,110,36)	033530
C DEGRADE POK TABLE	033540
C (IST 99 -- RED POK OF BLUE)	033550
C DO 9999 I=1,110	033560
C POK(I)=POK(I)*.2	033570
C 9999 CONTINUE	033580
C	033590
C ICF FRACTIONAL DAMAGE TABLE.	033600
C CALL READMS(3,FDT,510,37)	033610
C	033620
C FRACTION OF ARTY PER SUPPORT LEVEL.	033630
C DATA (CBTLEV(I), I=1,6)/.35,.67,1.,1.96,2.96,4./	033640
C DATA (C9TLEV(I), I=1,6)/.19,.34,.51,1.0,1.51,2.04/	033650

C 113462
 C IDF RATES OF FIRE. 113470
 CALL READMS(3,POF,78,78) 113482
 CALL CLCSMS(3) 113490
 C 113700
 C 113710
 C 113720
 C 113730
 C 113740
 C 113745
 C 113750
 C..UNCLASSIFIED DATA BASE..UNCLASSIFIED DATA BASE..UNCLASSIFIED DATA BASE 113755
 C 113755
 C 113760
 C..FIRER DATA..FIRER DATA..FIRER DATA..FIRER DATA..FIRER DATA..FIRER DATA 113770
 C 113790
 C TUBES PER BATTERY (TBAT) 113790
 C ROUND INDEX (ISHT) 113800
 C OPERATIONAL AVAILABILITY (OA) 113810
 C ROUNDS PER MISSION PER TUBE (RPM) 113820
 C PERCENT ICM (PICM) 113830
 C ICM ROUND TYPE (ICM) 113840
 C 113850
 C DATA TEMP/ 113860
 C 113870
 C.....BLUE FIRERS 113880
 C HEAP 113890
 C NO. FIRER VARIABLES 113900
 C TBAT ISHT OA RPM PICM ICM 113910
 43 3. 17. .94. 6. .0. 0. 113920
 44 3. 19. .94. 6. .0. 0. 113930
 45 4. 21. .94. 6. .0. 0. 113940
 46 4. 21. .94. 6. .0. 0. 113950
 47 3. 19. .94. 6. .0. 0. 113960
 48 6. 23. .76. 6. .0. 0. 113970
 49 0. 0. .00. 6. .0. 0. 113980
 50 4. 26. .76. 6. .63. 14. 113990
 51 4. 29. .61. 6. .75. 15. 114000
 52 0. 0. .00. 6. .0. 0. 114010
 53 6. 26. .61. 6. .67. 14. 114020
 54 12. 32. .76. 1. .0. 0. 114030
 55 9. 33. .76. 16. .0. 0. 114040
 C 114050
 C DATA TEMP/ 114060
 C 114070
 C.....RED FIRERS 114080
 C TBAT ISHT OA RPM PICM ICM 114090
 43 5. 17. .76. 6. .0. 0. 114100
 44 6. 19. .76. 6. .0. 0. 114110

45	6..	21..	.75.	6..	.0.	0..	034120
46	6..	21..	.61.	6..	.3.	0..	034130
47	0..	1..	.00.	6..	.1.	1..	034140
48	6..	24..	.75.	6..	.3.	0..	034150
49	6..	29..	.75.	6..	.2.	14..	034160
50	6..	24..	.61.	6..	.0.	0..	034170
51	6..	28..	.61.	6..	.2.	14..	034180
52	6..	26..	.75.	6..	.0.	0..	034190
53	6..	31..	.75.	6..	.25.	15..	034200
54	6..	33..	.61.	40..	.0.	0..	034210
55	6..	34..	.61.	6..	.5.	15..	034220
C							034230
C							034240
C	..FIRER DATA..FIRER DATA..FIRER DATA..FIRER DATA..FIRER DATA..FIRER DATA						034250
C							034260
C	..TARGET DATA..TARGET DATA..TARGET DATA..TARGET DATA..TARGET DATA..TARGET DATA						034270
C							034280
C	ELEMENTS PER TARGET (TGT)						034290
C	TARGET OP AV (OPERA)						034300
C	MILITARY WORTH (MWROF3,MW)						034310
C							034320
C	DATA TEMP3/						034330
C							034340
C	BLUE TARGETS						034350
C	TARGET						034360
C	CAT.		TARGET VARIABLES				034370
C		TGT	OPERA		MWROF3		034380
	1	49..	1.0.		31.0.		034390
	2	10..	.81.		130.0.		034400
	3	10..	.75.		127.2.		034410
	4	10..	.75.		127.2.		034420
	5	10..	.95.		159.0.		034430
	6	3..	.93.		11.0.		034440
	7	1..	.83.		180.2.		034450
	8	2..	.75.		169.4.		034460
	9	2..	.60.		153.6.		034470
	10	4..	.94.		141.4.		034480
	11	6..	.76.		201.6.		034490
	12	4..	.75.		212.2.		034500
	13	49..	1.0.		42.4.		034510
	14	2..	.81.		105.0.		034520
	15	10..	.75.		84.4.		034530
	16	10..	.75.		84.8.		034540
	17	10..	.95.		144.47		034550
C							034560
C	DATA TEMP4/						034570
C							034580
C	RED TARGETS						034590

C	PRINT(52,*)"*****ARTILLERY"	035140
	DO 100 J=1,2	035190
	PCM(J)=1.	035110
	PCM(J)=1.	035120
	IARTP(J)=0	035130
	100 CONTINUE	035140
C	110 PRINT *, "DO YOU WISH TO PROCESS INDIRECT FIRE ASSESSMENTS? "	035150
	CALL REEDA(INX)	035160
	IF(INX.EQ."Y")GO TO 150	035170
	IF(INX.EQ."N")GO TO 1070	035180
	PRINT 140	035190
	140 FORMAT(" INCORRECT ENTRY - TRY AGAIN")	035200
	GO TO 110	035210
C		035220
C	150 PRINT *, "INDIRECT FIRE ASSESSMENTS"	035230
C		035240
C	*** INITIALIZATION ***	035250
C		035260
	FLAG=0.	035270
	CH50=0.	035280
	CH53=0.	035290
C		035300
	DO 160 J=1,2	035310
	ARD(J)=0.	035320
	160 PERSF(J)=0.	035330
C		035340
	DO 170 K=1,55	035350
	DO 170 J=1,2	035360
	DO 170 L=1,2	035370
	170 PLOSS(K,J,L)=0.	035380
	INPUT=-1	035390
	AH1="CI CANNON"	035400
	CALL CKSTOR(AH1,INPUT)	035410
C		035420
C	- INTERACTIVE INPUTS -	035430
C		035440
	ARRAY(4)=10	035450
	ARRAY(5)=74	035460
	DO 171 I=6,90	035470
	171 ARRAY(I)=0	035480
	ARRAY(10)=1	035490
	172 PRINT *, "ENTER PERCENT (DECIMAL) 122 SP HOWITZERS USED IN OF POLE	035500
	1 "	035510
	CALL REED4	035520
	P=XINX(1)	035530

IF(P.GE.0.AND.P.LE.1)GO TO 174	035560
PRINT 140	035570
GO TO 172	035580
174 Z=PELMT(50,2)	035590
ELMT(50,2)=ELMT(50,2)-Z	035600
ELMT(29,2)=Z	035610
C	035620
C	035630
006940 MISSIONS FOR EW	035640
DO 180 J=1,2	035650
PCMT(J)=PCM(J)*(1.-PCOCR(J,3))	035660
180 CONTINUE	035670
C	035680
C	035690
190 DO 250 J=1,2	035700
IF(I.SMOKE(J).EQ."N")GO TO 250	035710
KIND="BLUE"	035720
IF(J.EQ.2)KIND="RED"	035730
200 PRINT 210,KIND	035740
210 FORMAT(" IS ",A4," EMPLOYING SMOKE?")	035750
CALL REEDA(INX)	035760
IF(INX.EQ."Y")GO TO 220	035770
IF(INX.EQ."N")GO TO 250	035780
GO TO 200	035790
220 PRINT 230,KIND	035800
230 FORMAT(" ENTER PERCENT MISSIONS DEFPAC OF ",A4," SMOKE EMPLOYERS")	035810
CALL REEDA	035820
P=XINX(1)	035830
IF(P.GE.0.AND.P.LE.1)GO TO 240	035840
PRINT 140	035850
GO TO 220	035860
240 PCMT(J)=PCMT(J)*(1.-P)	035870
ARRAY(10+J)=P	035880
250 CONTINUE	035890
C	035900
C	035910
DO 330 J=1,2	035920
NWD=13	035930
IF(J.EQ.2)NWD=16	035940
KIND="BLUE"	035950
IF(J.EQ.2)KIND="RED"	035960
260 PRINT 270,KIND	035970
270 FORMAT(" ENTER 3 ",A4," ARTY FIRING LEVELS (CST SPT,AC SUP,CTR-3T-	035980
"Y) -")	035990
CALL REEDA	036000
ICS(J)=XINX(1)	036010
ACSF(J)=XINX(2)	036020
ICST(J)=XINX(3)	036030
ARRAY(NWD+J)=ICS(J)	036040

ARRAY(NMO+1)=AOSF(J)	136140
ARRAY(NMO+2)=ICB(J)	136150
IF(ICS(J).LT.0.OR.ICS(J).GT.6)GO TO 290	136160
IF(AOSF(J).LT.0.OR.AOSF(J).GT.6)GO TO 290	136170
IF(ICB(J).GE.0.AND.ICB(J).LE.6)GO TO 300	136180
290 PRINT 140	136190
PRINT *,"ENTER 0 FOR NO FIRES"	136200
PRINT *," 1 FOR LIGHT INTERMITTENT FIRES"	136210
PRINT *," 2 FOR FIRES BASED ON 2/3 BASIC LOAD"	136220
PRINT *," 3 FOR FIRES BASED ON TOTAL BASIC LOAD"	136230
PRINT *," 4 FOR FIRES BASED ON 2/3 DAILY RESUPPLY RATE"	136240
PRINT *," 5 FOR FIRES BASED ON TOTAL DAILY RESUPPLY RATE"	136250
PRINT *," 6 FOR APPROX. SUSTAINED RATE OF FIRE"	136260
GO TO 260	136270
C	136280
SET ARTY COMBAT LEVEL.	136290
C	136300
330 IF(ICS(J).EQ.0)GO TO 310	136310
INX=ICS(J)	136320
CLEV(1,J)=CSTLEV(INX)	136330
ICS(J)=1	136340
310 IF(AOSF(J).EQ.0)GO TO 320	136350
INX=AOSF(J)	136360
CLEV(2,J)=CSTLEV(INX)	136370
AOSF(J)=1	136380
320 IF(ICB(J).EQ.0)GO TO 330	136390
INX=ICB(J)	136400
CLEV(3,J)=CSTLEV(INX)	136410
ICB(J)=1	136420
330 CONTINUE	136430
C	136440
340 PRINT *,"ENTER # HOURS OF ARTY SUPPORT (0-24, TIME,")	136450
CALL REED4	136460
HRARTY=XINX(1)	136470
ARRAY(19)=HRARTY	136480
IF(HRARTY.GE.0.AND.HRARTY.LE.24)GO TO 350	136490
PRINT 140	136500
GO TO 340	136510
C	136520
350 PRINT *,"ENTER # MINUTES OF PREP FIRE (0-60)"	136530
CALL REED4	136540
PREP(1A)=XINX(1)	136550
ARRAY(20)=PREP(1A)	136560
IF(PREP(1A).EQ.0)GO TO 370	136570
IF(PREP(1A).GT.0.AND.PREP(1A).LE.60)GO TO 340	136580
PRINT 140	136590
GO TO 350	136600
C	136610

360 PRINT *, "ENTER # MINUTES OF COUNTER-PREP FIRES (0-50)"	136520
CALL REED4	136530
PPEP(ID)=XINX(1)	136540
ARRAY(21)=PPEP(ID)	136550
IF(PPEP(ID).GE.0..AND.PPEP(ID).LE.60.)GO TO 370	136560
PRINT 140	136570
GO TO 360	136580
C	136590
370 PRINT *, "ENTER # MINUTES OF FINAL PROTECTIVE FIRE (0-50)"	136600
CALL REED4	136610
FPF=XINX(1)	136620
ARRAY(22)=FPF	136630
IF(FPF.GE.0..AND.FPF.LE.60.)GO TO 380	136640
PRINT 140	136650
GO TO 370	136660
C	136670
C CALC. ACTUAL # HOURS OF ICF SUPPORT	136680
C ATTACKING FORCE	136690
380 HOURS=HRARTY	136700
HR(IA)=HOURS-(PPEP(IA)/50.)	136710
C DEFENDING FORCE	136720
HR(ID)=HOURS-(PREP(ID)/50.)-(FPF/50.)	136730
C	136740
C	136750
C SPECIAL MISSION LOGIC FLAGS	136760
C	136770
PERSF(IA)=1.	136780
IF(IMOUNT.EQ.2)GO TO 390	136790
PERSF(ID)=1.	136800
GO TO 400	136810
C	136820
390 PRINT *, "WILL ATTACKER DISMOUNT INFANTRY DURING THIS CI?"	136830
CALL REED4(INX)	136840
IF(INX.EQ."Y")PERSF(ID)=1./HR(ID)	136850
IF(INX.EQ."Y")ARRAY(23)=1	136860
IF(INX.EQ."Y".OR.INX.EQ."N")GO TO 400	136870
PRINT 140	136880
GO TO 390	136890
400 CONTINUE	136900
C DO 95 J=1,2	136910
C IF(ELMT(50,J).EQ.0..AND.ELMT(51,J).EQ.0.)GO TO 95	136920
C DO 94 I=1,15	136930
C FDT(I,11,J)=FDT(I,12,J)	136940
C 94 CONTINUE	136950
C 95 CONTINUE	136960
C	136970
C *** FIRE DISTRIBUTION FACTOR (FDF) ***	136980
C	136990

410	DO 420 INX=1.5	037000
	DO 420 J=1.2	037010
420	DO FLINX, J=0.	037020
C	IFLAG=FLAG+1	037030
C	CYCLE ROUTINE FOR 30TH FORCES.	037040
C	DO 720 J=1.2	037050
	L=1	037060
	IF (L.EQ.J) L=2	037070
	IF (FLAG.EQ.2.AND.J.EQ.1) GO TO 720	037080
C		037090
C	BRANCH FOR FB PIPES	037100
C	IF (FLAG.EQ.2) GO TO 490	037110
	IF (FLAG.EQ.3.AND.PREP(J).EQ.0) GO TO 520	037120
C		037130
C	PDF DENOMINATOR FOR STANDARD IDF MISSIONS.	037140
C		037150
C	ITERATE FOR ALL POSSIBLE IDF TARGETS.	037160
	DO 480 K=1.55	037170
	IF (MAP(K).EQ.3) GO TO 480	037180
	IF (L.EQ.1.AND.K.GE.54) GO TO 480	037190
	IF (ELHT(K,L)-PLOSS(K,L,2).LE.0) GO TO 480	037200
C	TEST FOR DISMOUNTED INFANTRY DURING PREP/C-PREP PIPES.	037210
	IF (K.EQ.3.AND.FLAG.EQ.0.AND.PERSF(J).NE.1) GO TO 480	037220
C		037230
C	SET ACQ FACTOR.	037240
	KK=2	037250
	INX=2	037260
	IF (L.EQ.1) INX=1	037270
	ACQ=POK(K,L)	037280
	IF ((K.GE.3.AND.K.LE.32).OR.(K.GE.43.AND.K.LE.47)) ACQ=POK(K,L)*	037290
	1 PSN(IENGAG,INX,KK)+.5*(1.-PSN(IENGAG,INX,KK))	037300
C		037310
C	CALCULATE PRESENTED TARGET AREAS.	037320
	FAC=1.	037330
	IF (K.EQ.3) FAC=PERSF(J)	037340
	IF (K.GE.31.AND.K.LE.42) FAC=ACSF(J)	037350
	IF (K.LT.31.AND.ICS(J).EQ.0) GO TO 480	037360
	IF (K.GE.43.AND.K.LE.55.AND.ICB(J).EQ.0) GO TO 480	037370
	KK=MAP(K)	037380
	IF (KK.LE.5.AND.L.EQ.10) KK=KK+12	037390
	AT=ACQ*(ELHT(K,L)-PLOSS(K,L,2))*OPEFA(KK,L)/GT(KK,L)	037400
	MNTH=MN(KK)	037410
	IF (J.EQ.2) MNTH=MNROFB(KK)	037420
	IF (FLAG.EQ.3) MNTH=1	037430
	FAC=AT*MNTH*FAC	037440
C		037450
		037460
		037470

C	FILTER OUT INAPPROPRIATE SPECIAL MISSIONS	037440
	IF (FAC.LE.0) GO TO 480	037490
C		037500
C	BRANCH AS PER TARGETING SCHEME	037510
C		037520
	IF (K.EQ.1.OR.K.EQ.4A.OR.K.EQ.55) GO TO 450	037530
	IF (K.EQ.3.OR.K.EQ.12.OR.K.EQ.13) GO TO 430	037540
	IF (K.LT.16.OR.K.EQ.42) GO TO 480	037550
	IF (K.EQ.37.OR.(J.EQ.1.AND.(K.EQ.35.CR.K.EQ.36))) GO TO 430	037560
	IF (K.GE.21.AND.K.LE.36) GO TO 440	037570
	IF (K.GE.40.AND.K.LE.41) GO TO 470	037580
	IF (K.GE.43.AND.K.LE.47) GO TO 430	037590
	GO TO 460	037600
C		037610
C	LT. MORTARS	037620
	430 FDF(1,J)=FDF(1,J)+FAC	037630
C		037640
C	HVV. MORTARS	037650
	440 FDF(2,J)=FDF(2,J)+FAC	037660
C		037670
C	LT. ARTY	037680
	450 FDF(3,J)=FDF(3,J)+FAC	037690
C		037700
C	MED. ARTY	037710
	460 FDF(4,J)=FDF(4,J)+FAC	037720
C		037730
C	HVV. ARTY	037740
	470 IF (K.GE.49.AND.K.LE.53) FAC=FAC*2.	037750
	FDF(5,J)=FDF(5,J)+FAC	037760
	480 CONTINUE	037770
	GO TO 520	037780
C		037790
C		037800
C	FDF DENOMINATOR CALC. FDF FF FIRES.	037810
C		037820
C	TEST FOR SPECIAL IDF MISSIONS	037830
	490 IF (FLAG.EQ.2..AND.(J.EQ.1A.CR.FFF.EQ.0.)) GO TO 520	037840
C		037850
C	ONLY FORWARD WEAPON SYSTEMS ARE SPECIAL MISSION TARGETS.	037860
	DO 510 K=1,32	037870
	IF (L.EQ.1.AND.K.GE.54) GO TO 510	037880
	IF (ELMT(K,L)-PLOSS(K,L,2).LE.0) GO TO 510	037890
C		037900
C	SET ACQ FACTOR.	037910
	KK=2	037920
	IF (FLAG.EQ.0.) KK=1	037930
	INX=2	037940
	IF (L.EQ.1A) INX=1	037950

	ACQ=POK(K,L)	037960
	IF((K.GE.3.AND.K.LE.32).OR.(K.GE.43.AND.K.LE.47))ACQ=POK(K,L)	037970
	1 PSN(IEGAG,INX,KK)*.5*(1.-PSN(IEGAG,INX,KK))	037980
C	CALCULATE PRESENTED TARGET AREAS.	037990
C		038000
C	FAC=1.	038010
	IF(K.EQ.3)FAC=PERSF(J)	038020
	KK=MAP(K)	038030
	IF(KK.LE.5.AND.L.EQ.1)KK=KK+12	038040
	AT=ACQ*(ELMT(K,L)-PLOSS(K,L,2))*OPERA(KK,L)/TGT(KK,L)	038050
	FAC=AT*FAC	038060
C	FILTER OUT INAPPROPRIATE SPECIAL MISSIONS.	038070
	IF(FAC.LE.1)GO TO 510	038080
C		038090
C	FILTER OUT INAPPROPRIATE SPECIAL MISSION TARGETS.	038100
C	IF(K.LT.15.OR.K.GT.27)GO TO 510	038110
	DO 500 INX=1,5	038120
	FQF(INX,J)=FQF(INX,J)+FAC	038130
	500 CONTINUE	038140
	510 CONTINUE	038150
C		038160
C		038170
C	*** IDF ASSESSMENTS ***	038180
C		038190
C		038200
C	ITERATE FOR ALL IDF TARGETS.	038210
C		038220
	520 I50=0	038230
	I53=0	038240
	DO 710 K=1,55	038250
	IF(MAP(K).EQ.3)GO TO 710	038260
	SKILL=0.	038270
	AKILL=1.	038280
	IF(L.EQ.1.AND.K.GE.54)GO TO 710	038290
	IF(ELMT(K,L)-PLOSS(K,L,2).LE.0.)GO TO 710	038300
C		038310
C	SET ACQ PARMS.	038320
	KK=2	038330
	IF(FLAG.EQ.0.)KK=1	038340
C		038350
C	DETERMINE ACQ FACTOR.	038360
	INX=2	038370
	IF(L.EQ.1)INX=1	038380
	ACQ=POK(K,L)	038390
	IF((K.GE.3.AND.K.LE.32).OR.(K.GE.43.AND.K.LE.47))ACQ=POK(K,L)	038400
	1 PSN(IEGAG,INX,KK)*.5*(1.-PSN(IEGAG,INX,KK))	038410
C	CALCULATE PRESENTED TARGET AREAS.	038420
	FAC=1.	038430

	IF(K.LT.3)FAC=ICS(J)	138440
	IF(K.EQ.3.AND.FAC.EQ.1)FAC=PERSF(J)	138450
	IF(K.GE.31.AND.K.LE.42)FAC=ADSF(J)	138460
	IF(K.GE.43.AND.K.LE.55)FAC=ICB(J)	138470
	KK=MAP(K)	138480
	IF(KK.LE.5.AND.L.EQ.10)KK=KK+12	138490
	AT=ACQ*(ELHT(K,L)-PLOSS(K,L,2))*CPERA(KK,L)/TGT(KK,L)	138500
C	SET MILITARY WORTH OF TARGETS	138510
	MMTH=MM(KK)	138520
	IF(J.EQ.2)MMTH=MMROFB(KK)	138530
	IF(FLAG.NE.1)MMTH=1	138540
C	ITERATE FOR ALL IDF WEAPON SYSTEMS.	138550
	DO 570 I=41,55	138560
	INC=11*(I-41)*4	138570
	PKILL(I-42)=1.	138580
	IF(ELHT(I,J)-PLOSS(I,J,2).LE.0)GO TO 670	138590
C		138600
C	DETERMINE CATEGORY OF IDF WEAPON	138610
	ICAT=4	138620
	IF(I.EQ.43)ICAT=1	138630
	IF(I.GE.44.AND.I.LE.47)ICAT=2	138640
	IF(I.EQ.48.OR.I.EQ.55)ICAT=3	138650
	IF(I.EQ.52.OR.(I.EQ.53.AND(J.EQ.2)))ICAT=5	138660
	IF(FDF(ICAT,J).EQ.0)GO TO 670	138670
	MMTHX=MMTH	138680
	IF(K.GE.49.AND.K.LE.55.AND.ICAT.EQ.5)MMTHX=MMTH*2.	138690
C		138700
C	SET # HOURS OF IDF SUPPORT.	138710
	HOURS=HR(J)	138720
	IF(FLAG.EQ.0)HOURS=PREP(J)/60.	138730
	IF(FLAG.EQ.2)HOURS=FPF/60.	138740
	IF(FLAG.EQ.2.AND.J.EQ.14)HOURS=0.	138750
	IF(HOURS.LT.0)HOURS=0.	138760
C	SET FRACTION OF MISSIONS FIRED AT TARGETED OBJECTIVES.	138770
	F=.47	138780
	IF(ICAT.LT.3)F=.97	138790
	IF(J.EQ.2.AND.ICAT.GT.2)F=.93	138800
	IF(J.EQ.2.AND.(I.EQ.44.OR.I.EQ.50))F=1.0	138810
	IF(J.EQ.1.AND.(I.GE.46.AND.I.LE.47))F=1.0	138820
C		138830
C	CALC. SUPPRESSION FACTOR	138840
	IOP=1	138850
	IF(J.EQ.1.AND.I.GE.46)IOP=2	138860
	S=1.-FSSF(J)*SUPR(IOP)	138870
	IF(S.LE.2)S=.2	138880
	IF(FLAG.EQ.2)S=1.	138890
C		138900
C	SET THE FRACTION OF MORTARS ACTIVE.	138910

	DE=PSN(IENGAG,J,2)	038920
	IF(FLAG.EQ.1)DE=PSN(IENGAG,J,1)	038930
C		038940
C	ALL ARTY IS ACTIVE.	038950
	IF(ICAT.GT.2)DE=1.	038960
	IF(K.LE.30) C=CLEV(1,J)	038970
	IF(K.GE.31.AND.K.LE.42) C=CLEV(2,J)	038980
	IF(K.GE.43.AND.K.LE.55) C=CLEV(3,J)	038990
	IF(ICAT.GT.2.AND.FLAG.EQ.1)CE=C	039000
C		039010
C	SET MAPPING INDEX	039020
	II=MAP(I)	039030
C	BRANCH FOR FP FIRES.	039040
	IF(FLAG.EQ.2)GO TO 540	039050
C		039060
C	FILTER OUT INAPPROPRIATE TARGETS FOR STANDARD IDF MISSIONS.	039070
	IF(K.EQ.1.OR.K.EQ.42.OR.K.EQ.55)GO TO 520	039080
	IF(K.EQ.3.OR.K.EQ.12.OR.K.EQ.13)GO TO 530	039090
	IF(K.LT.16.OR.K.EQ.42)GO TO 710	039100
	IF(K.EQ.37.OR.(J.EQ.1.AND.(K.EQ.35.OR.K.EQ.36)))GO TO 530	039110
	IF(K.GE.21.AND.K.LE.36)GO TO 540	039120
	IF(K.GE.40.AND.K.LE.41)GO TO 570	039130
	IF(K.GE.43.AND.K.LE.47)GO TO 530	039140
	GO TO 560	039150
C		039160
	530 IF(ICAT.EQ.1)GO TO 590	039170
C		039180
	540 IF(ICAT.EQ.2)GO TO 590	039190
C		039200
	550 IF(ICAT.EQ.3)GO TO 590	039210
C		039220
	560 IF(ICAT.EQ.4)GO TO 590	039230
C		039240
	570 IF(ICAT.EQ.5)GO TO 590	039250
	GO TO 570	039260
C		039270
C	FILTER OUT INAPPROPRIATE TARGETS FOR SPECIAL IDF MISSIONS	039280
	580 IF(K.GT.32)GO TO 710	039290
	IF(K.EQ.3.AND.FLAG.EQ.0.AND.PEPSF(J).NE.1)GO TO 710	039300
	IF(K.LT.16.AND.K.NE.3.AND.K.NE.12.AND.K.NE.13)GO TO 710	039310
	IF(FLAG.EQ.2.AND.(K.LT.16.OR.K.GT.27))GO TO 710	039320
C		039330
C	CALCULATE BATTERY MISSIONS PER TUBE FOR THIS TYPE MISSION.	039340
	590 CONTINUE	039350
	IF(TBAT(I-42,J).LE.0)GO TO 670	039360
	BMT=(ELMT(I,J)-PLOSS(I,J,2))/TBAT(I-42,J)*CA(I-42,J)*CE*F*HOURS	039370
	L=ROF(I-42,FLAG,J)/PPH(I-42,J)*FAC*AT*MTMX/DOF(ICAT,J)	039380
	BMT=BMT*PCN(J)	039390

C	DEGRAD OF MISSIONS DUE TO NEW PLAY CF SMCNE	139400
	IF(I.EQ.1.AND.I.GE.44.AND.I.LE.47)BMT=BMT*PCM1(J)	139410
	IF(I.EQ.2.AND.(I.EQ.48.OR.I.EQ.50))BMT=BMT*PCM1(J)	139420
C		139430
	IF(FLAG.NE.1..OR.J.EQ.2)GO TO 600	139440
	IF(I.NE.50.AND.I.NE.53)GO TO 610	139450
	IF(I.EQ.50.AND.ISQ.EQ.1)GO TO 600	139460
	IF(I.EQ.53.AND.IS3.EQ.1)GO TO 600	139470
	IF(FLAG*AT*PDF(ICAT,J).LE.0)GO TO 610	139480
	BMT=BMT*PDF(ICAT,J)/(FAC*AT*MWTHX)	139490
	CALL CLGP(BMT,I,LOSS,CP,1)	139500
	BMT=BMT*FAC*AT*MWTHX/PDF(ICAT,J)	139510
	IF(I.EQ.50)ISQ=1	139520
	IF(I.EQ.50)CM50=CM	139530
	IF(I.EQ.53)IS3=1	139540
	IF(I.EQ.53)CM53=CM	139550
C		139560
C	CALC. AMMO EXPENDITURES	139570
600	IF(I.EQ.2..OR.FLAG.NE.1..OR.(I.NE.50.AND.I.NE.53))GO TO 610	139580
	IF(I.EQ.50)CM=CM50	139590
	IF(I.EQ.53)CM=CM53	139600
C	REDUCE CONVENTIONAL FIRE BECAUSE OF CLGP MISSIONS	139610
	BMT=BMT-(2.*CM/IBAT(I-42,J)*FAC*AT*MWTHX/PDF(ICAT,J))	139620
	IF(BMT.GE.0.)GO TO 604	139630
	BMT=0	139640
	PRINT *, "CLGP MISSIONS CAN NOT BE PROPERLY APPORTIONED TO ",K	139650
	PRINT *, "LOWER MISSIONS OR EQUALIZE LEVELS"	139660
608	CONTINUE	139670
C		139680
610	IS=ISHT(I-42,J)	139690
	ICAN=0	139700
C	AMMO EXP CALC	139710
	X=IBAT(I-42,J)*BMT/(I-42,J)*BMT/F	139720
	IF(J.EQ.2)GO TO 620	139730
C	BLUE	139740
C	NON-TARGETED AMMO EXP CALC	139750
	IF(IS.EQ.17..OR.IS.EQ.23..OR.IS.EQ.26)	139760
1	SHOTSX(IS+1,J)=SHOTSX(IS+1,J)+.03*X	139770
	IF(IS.EQ.17..OR.IS.EQ.23..OR.IS.EQ.26)	139780
1	AM(INC+1)=AM(INC+1)+.03*X	139790
	IF(ICAN.NE.0)PRINT *, "IS+1", IS, SHOTSX(IS+1,J)	139800
C	IMPROVED CONVENTIONAL MUNITIONS-ANTI PERSONNEL (ICM-4)	139810
	IF(K.EQ.3.AND.IS.EQ.23) IS=IS+2	139820
	IF(K.EQ.3.AND.IS.EQ.23) INC=INC+2	139830
	IF(I.GE.44.AND.I.LE.47).AND.PCM1(J).GT.0.)X=X/PCM1(J)	139840
	IF(IS.EQ.32) INC=INC+2	139850
C	TARGETED AMMO EXP	139860
	X1=X	139870

IF(I.LT.44.OR.I.GT.47)X1=.97*X	139940
X2=X1*PICH(I-42,J)	139941
X1=X1-X2	139942
AM(ING)=AM(ING)+X1	139943
AM(ING+2)=AM(ING+2)+X2	139944
SHOTSV(IS,J)=SHOTSV(IS,J)+X1	139945
SHOTSV(IS+2,J)=SHOTSV(IS+2,J)+X2	139946
C AMMO EXP FOR NEW PLAY OF SMOKE	139950
IF(IS.NE.21.AND.IS.NE.19)GO TO 630	139960
SHOTSV(IS+1,J)=SHOTSV(IS+1,J)+X*(1.-PCM1(J))	139970
SHOTSV(IS,J)=SHOTSV(IS,J)-X*(1.-PCM1(J))	139980
AM(ING+1)=AM(ING+1)+X*(1.-PCM1(J))	139990
AM(ING)=AM(ING)-X*(1.-PCM1(J))	140000
C END BLUE	140010
GO TO 630	140020
C	140030
C RED	140040
C NEXT LINE FOR NON-TARGETED AMMO EXP CALC	140050
620 IF(I.LT.52.AND.I.NE.46.AND.I.NE.44.AND.I.NE.50)	140060
1 SHOTSV(IS+1,J)=SHOTSV(IS+1,J)+.07*X	140070
C	140080
C TARGETED AMMO EXP	140090
IF((I.EQ.48.OR.I.EQ.57).AND.PCM1(J).GT.1)X=X/PCM1(J)	140100
X1=X	140110
IF(I.LT.52.AND.I.NE.46.AND.I.NE.44.AND.I.NE.57)X1=.93*X	140120
X2=PICH(I-42,J)*X1	140130
X1=X1-X2	140140
SHOTSV(IS,J)=SHOTSV(IS,J)+X1	140150
IF(I.EQ.31.OR.I.EQ.34)GO TO 623	140160
SHOTSV(IS+2,J)=SHOTSV(IS+2,J)+X2	140170
GO TO 627	140180
623 SHOTSV(IS+1,J)=SHOTSV(IS+1,J)+X2	140190
627 CONTINUE	140200
C AMMO EXP FOR NEW PLAY OF SMOKE	140210
IF(IS.NE.24)GO TO 630	140220
SHOTSV(IS+1,J)=SHOTSV(IS+1,J)+X*(1.-PCM1(J))	140230
SHOTSV(IS,J)=SHOTSV(IS,J)-X*(1.-PCM1(J))	140240
C END RED	140250
C SET ARO AFRAY	140260
630 CONTINUE	140270
IF(IFLAG.NE.2)GO TO 635	140280
X=(ELMT(I,J)-PLOSS(I,J,2))*OA(I-42,J)*S*F*DE*ROF(I-42,1,J)*FAC*AT	140290
1 *MTHX/FDF(ICAT,J)	140300
IF(CLEV(1,J).LE..67)GO TO 635	140310
IF(K.LT.3.OR.K.GT.38)GO TO 635	140320
IF(J.EQ.1.AND.K.EQ.38)GO TO 635	140330
IF(J.EQ.2.AND.(K.GE.33.AND.K.LE.34))GO TO 635	140340
ARO(L)=ARO(L)+X/60.	140350

635	CONTINUE	140360
C	CALCULATE LOSSES	140370
C	CODE FOR BOTH HE AND ICM	140380
	IC=ICM(I-42,J)	140390
	PIC=PICM(I-42,J)	140400
	BMT=(1-PIC)*BMT	140410
	BMT=BMT	140420
	FOTATH=FCT(I-42,KK,J)/A	140430
	FOTATI=0	140440
	IF(IC.EQ.0)GO TO 637	140450
	FOTATI=FCT(IC,KK,J)/AT	140460
	IF(J.EQ.1.AND.I.EQ.53) FOTATI=.5*FOTATI	140470
	IF(FOTATH.GE.1.AND.FOTATI.GE.1)GO TO 713	140480
637	PKILL(I-42)=1.	140490
	IF(FOTATH.GE.1)GO TO 640	140500
	PKILL(I-42)=(1.-FOTATH)**BMT	140510
640	CONTINUE	140520
	IF(FOTATI.GE.1)GO TO 650	140530
	PKILL(I-42)=PKILL(I-42)*(1.-FOTATI)**BMT	140540
650	CONTINUE	140550
C	END HE AND ICM MIX CODE	140560
	GO TO 660	140570
660	AKILL=SKILL*PKILL(I-42)	140580
	SKILL=SKILL*(1.-PKILL(I-42))	140590
670	CONTINUE	140600
	TKILL=(1.-AKILL)*AT*TGT(KK,L)	140610
C		140620
C	DISTRIBUTE LOSSES	140630
	IF(SKILL.LE.0)GO TO 713	140640
	DO 700 I=43,55	140650
	IF(ELMT(I,J)-FLOSS(I,J,2).LE.0)GO TO 700	140660
	AKILL=TKILL*(1.-PKILL(I-42))/SKILL	140670
	AKILL=IFIX(AKILL*10.+5)/10.	140680
C		140690
C	BRANCH FOR PERSONNEL TARGETS	140700
	IF(K.EQ.3)GO TO 640	140710
	ALOSS(I,K)=ALOSS(I,K)+IFIX(AKILL*10.+501)*PACK(L)	140720
	PLOSS(K,L,1)=PLOSS(K,L,1)+IFIX(AKILL*10.+501)/10.	140730
C		140740
C	ACCESS CREW KILLS.	140750
	IF(K.GT.12)ALOSS(I,2)=ALOSS(I,2)+IFIX(AKILL*CREWS(K-12,1)*10.+501)	140760
	1)*PACK(L)	140770
	IF(K.GT.12)PLOSS(2,L,1)=PLOSS(2,L,1)+IFIX(AKILL*CREWS(K-12,L)*10.	140780
	+501)/10.	140790
	IF(K.LT.21.OR.K.GT.25.OR.IMOUNT.EQ.1)GO TO 713	140800
	IF(L.EQ.10)GO TO 700	140810
	AKILL=IFIX(AKILL*60.+5)/10.	140820
C		140830

C	ASSESSMENT OF DISMOUNTED INFANTRY WEAPONS.	340940
680	DO 890 KK=3,15	340950
	IF(ELMT(KK,L)-PLOSS(KK,L,2).LE.0.)GO TO 690	340960
	ALOSS(I,KK)=ALOSS(I,KK)+IFIX(AKILL*PLT(KK)*10.+0(1)*PACK(L)	340970
	PLOSS(KK,L,1)=PLOSS(KK,L,1)+IFI(AKILL*PLT(KK)*10.+0(1))/10.	340980
690	CONTINUE	340990
700	CONTINUE	340900
710	CONTINUE	340910
720	CONTINUE	340920
	IF(IFLAG.NE.2)GO TO 930	340930
	IF(NMU(1).EQ.0)NMU(1)=1	340940
	IF(NMU(2).EQ.0)NMU(2)=1	340950
	APD(1)=APD(1)/NMU(1)	340960
	ARD(2)=ARD(2)/NMU(2)	340970
	IF(IA.EQ.1)ARD(2)=ARD(2)/FMASS	340980
	IF(IA.EQ.2)ARD(1)=ARD(1)/FMASS	340990
	IF(CH50.GT.0.)CALL CLGP(3MT,50,PLOSS,CH50,2)	341000
	IF(CH53.GT.0.)CALL CLGP(3MT,53,PLOSS,CH53,2)	341010
	X=APD(1)	341020
	IF(ARD(2).GT.0.)X=ARD(2)	341030
	IF(X.LT.21)PRINT *, "NO JUST EFFECTS"	341040
	IF(X.GE.21.AND.X.LT.84)PRINT *, "LIGHT DUST EFFECTS"	341050
	IF(X.GE.84)PRINT *, "HEAVY DUST EFFECTS"	341060
	GO TO 930	341070
C		341080
725	IF(IPUN.EQ.1)GO TO 830	341090
	IF(IRUN.EQ.3)GO TO 930	341100
730	IF(IFLAG-2) 740,750,760	341110
740	PRINT *, "DO YOU WISH TO SEE PREP PRINTS?"	341120
	GO TO 770	341130
750	PRINT *, "DO YOU WISH TO SEE STANDAPD PRINTS?"	341140
	GO TO 770	341150
760	PRINT *, "DO YOU WISH TO SEE PPF PRINTS?"	341160
770	CONTINUE	341170
	CALL REEDA(INX)	341180
	IF(INX.EQ."N")GO TO 830	341190
	IF(INX.EQ."Y")GO TO 790	341200
	GO TO 730	341210
790	PRINT 990	341220
	IF(IFLAG-2) 800,810,820	341230
800	PRINT *, "-----PREP/C-PREP ASSESSMENTS-----"	341240
	GO TO 900	341250
810	PRINT *, "STANDAPD IDF MISSION ASSESSMENT"	341260
	GO TO 900	341270
820	PRINT *, "-----PPF ASSESSMENTS-----"	341280

	GO TO 900	041321
C		041330
C	SUM LOSSES DUE TO LAST TYPE OF MISSION.	041340
830	DO 840 K=1.55	041350
	DO 940 L=1.2	041360
	PLOSS(K,L,2)=PLOSS(K,L,2)+PLCSS(K,L,1)	041370
840	PLOSS(K,L,1)=0.	041380
	FLAG=FLAG+1	041390
C		041400
C	BRANCH FOR NEXT TYPE OF MISSION	041410
	IF(FLAG.LT.3)GO TO 410	041420
C		041430
C	OUTPUT RESULTS	041440
	IF(IRUN.EQ.1)GO TO 1060	041450
	PRINT 990	041460
890	FORMAT("I")	041470
	PRINT "*****INDETECT FIRE ASSESSMENTS*****"	041480
900	DO 1050 J=1,2	041490
	IPOINT=0	041500
	L=1	041510
	IF(L.EQ.J)L=2	041520
	DO 1050 K=1,55	041530
	TKILL=0.	041540
	CLOST=0.	041550
	IF(FLAG.EQ.3)GO TO 920	041560
	AKILL=PLOSS(K,L,1)	041570
	IF(K.LT.16.AND.K.NE.13)GO TO 910	041580
	CLOST=AKILL*CREWS(K-12,L)	041590
910	TKILL=AKILL	041600
	GO TO 970	041610
920	DO 960 I=43,55	041620
	IF(J.EQ.2)GO TO 930	041630
	AKILL=(ALOSS(I,K)-IFIX(ALCSS(I,K)/PACK(1))*PACK(1))/10.	041640
	GO TO 940	041650
930	AKILL=IFIX(ALOSS(I,K)/PACK(1))/10.	041660
940	IF(K.LT.16.AND.K.NE.13)GO TO 950	041670
	CLOST=CLOST+AKILL*CREWS(K-12,L)	041680
950	TKILL=TKILL+AKILL	041690
960	CONTINUE	041700
970	IF(TKILL.LT..1.AND.CLOST.LT..1)GO TO 1050	041710
	IF(IPOINT.EQ.1)GO TO 1010	041720
	IF(J.EQ.2)GO TO 990	041730
	PRINT 980	041740
980	FORMAT("I",31X,"I")	041750
	PRINT "I RED LOSSES TO BLUE I"	041760
	GO TO 1030	041770
990	PRINT 990	041780
	PRINT "*****"	041790

	PRINT 980	041920
	PRINT *, "I BLUE LOSSES TO RED I"	041917
1000	IPONT=1	041920
	PRINT *, "I ITEM #LOST CREW I"	041930
	PRINT 980	041940
1010	IF(K.LT.16.AND.K.NE.1)GO TO 1030	041950
	PRINT 1020,K,TKILL,CLCST	041960
	GO TO 1050	041970
1020	FORMAT(" I",7X,I2,F9.1,F7.1,6X,"I")	041980
1030	PRINT 1040,K,TKILL	041990
1040	FORMAT(" I",7X,I2,F9.1,11X,"I")	041900
1050	CONTINUE	041910
	PRINT 980	041920
	PRINT *, "*****"	041930
	IF(FLAG.NE.3)GO TO 430	041940
	PRINT 890	041950
1060	IF(IRUN.EQ.1)PRINT *, " INCTECT FIRE ASSESSMENTS PRINTED HERE"	041960
	INX=0	041970
	CALL LOSS(43,50,1,50,INX)	041980
	IF(INX.EQ."NO")GO TO 1070	041990
	CALL CKSTOR(AH1,IPUT)	042000
	IPUT=-1	042010
	AH1="INPUT 36"	042020
	CALL CKSTOR(AH1,IPUT)	042030
	DO 1065 I=10,76	042040
1065	AH(I)=ARRAY(I)	042050
	CALL CKSTOR(AH1,IPUT)	042060
1070	END	042070

SUBROUTINE CLGP(BMT,I,PLOSS,CM,IFLG)	342090
COMMON IA, ID, IP, IENGAG, ITERRN, IVIS, IMOUNT, MINES, CFP6, FSFP, FPR,	342110
1 ATIME, IFIRST, IPUN, NUDD, FPASS, IMAXKV,	342115
2 SF(2), FSSF(2), PACK(2),	342120
3 FLIT(80,2), ALOSS(66,66), SHOTS(55,2), CKILL(53,2), SHOTSV(55,2)	342130
COMMON/REED/JOAY1, XINX(4), ICARD(20), IAPHOR	342140
COMMON/DATA/FPS(80,2), CREWS(53,2), AFOS(12), CPQS(4), TIME3M(20,2),	342150
1 PSN(6,2,2), PLT(15), KEY(171)	342160
COMMON/EH/PCQGR(2,4), ICOUNT(8), ISMOKE(2), PCSMO(2), PCSSM(2), ASD(2)	342170
1, NMU(2)	342180
DIMENSION PLOSS(55,2,2), OA(17)	342190
COMMON/FCX/PREC(9), DEGRAD(5,5,2)	342200
DIMENSION OPERA(8), PCNSM(2)	342210
DIMENSION DUST(5,2)	342220
DATA ((DUST(I,J), I=1,5), J=1,2)/3*.6,2*.0,...,33,4*.7./	342230
DATA OA/3*.78,.62,.0,10*.41,2*.85/	342240
DATA (OPERA(I), I=1,8)/2*.75,2*.61,2*.75,2*.61/	342250
C DATA (IPAQ(I,J), I=1,5), J=1,4)/.33,.73,.93,.97,.99,	342260
C 1 .33,.73,.93,.96,.96,.14,.12,.37,.03,.01,.5*2./	342270
IF (IVIS.EQ.1) JVIS=1	342280
IF (IVIS.EQ.2.OR.IVIS.EQ.3) JVIS=2	342290
IF (IVIS.EQ.4) JVIS=3	342300
IF (IVIS.EQ.5) JVIS=4	342310
IF (IFLG.EQ.1) GO TO 105	342320
C IF (ITERRN.EQ.1.OR.ITERRN.EQ.2) PEXP=.86	342330
IF (ITERRN.EQ.3.OR.ITERRN.EQ.4) PEXP=.9	342340
L=2	342350
J=1	342370
CMS=0.	342380
C CHECK FRIENDLY SARRAGE	342390
FARD=DUST(IVIS,2)	342400
IF (ARD(L).LT.84.) FARD=DUST(IVIS,1)	342410
IF (ARD(L).LT.21.) FARD=1.	342420
C CHECK THREAT SARRAGE	342430
TARD=DUST(IVIS,2)	342440
IF (ARD(J).LT.84.) TARD=DUST(IVIS,1)	342450
IF (ARD(J).LT.21.) TARD=1.	342460
C USE DOMINANT THREAT	342470
ACUST=FARD	342480
IF (TARD.LT.FARD) ACUST=TARD	342490
C DO 100 IEL=1,2	342500
PCNSM(IEI)=(1.-PCSMO(IEI))*(1.-PCSMO(13-IEI)/2.)*(1.-PCSSM(IEI))	342510
100 CONTINUE	342520
GO TO 130	342530
105 MAX=2*GMT	342540
110 PRINT *, " IS AN AERIAL DESIGNATOR IN USE?"	342550

CALL REECA(INX)	042570
S=1	042580
IF(INX.NE."Y".AND.INX.NE."N")S=2	042590
IF(S.EQ.2)PRINT *,"INCORRECT ENTRY"	042600
IF(S.EQ.2)GO TO 110	042610
120 PRINT *,"ENTER # CLGP MISSIONS TO FIRE (MAX=,MAX,)"	042620
CALL REED4	042630
CM=XINX(1)	042640
IF(CM.GT.MAX)PRINT *,"TOO BIG. CM RESET TO MAX"	042650
IF(CM.GT.MAX)CM=MAX	042660
IF(CM.GE.0.AND.CM.LE.MAX)RETURN	042670
PRINT *,"INCORRECT ENTRY - TRY AGAIN"	042680
GO TO 120	042690
130 PRINT *,"ENTER CLOUD HEIGHT - (1-5) "	042700
CALL REED4	042710
ICH=XINX(1)	042720
IF(ICH.LT.0.OR.ICH.GT.5)GO TO 140	042730
IF(ICH.EQ.999)GO TO 150	042740
GO TO 160	042750
C	042760
140 PRINT *,"INCORRECT ENTRY"	042770
150 PRINT *,"ENTER 0 IF RANGE IS LESS THAN 1500 M"	042780
PRINT *,"ENTER 1 IF RANGE IS 1500-1999 M"	042790
PRINT *,"ENTER 2 IF RANGE IS 2000-2499 M"	042800
PRINT *,"ENTER 3 IF RANGE IS 2500-2999 M"	042810
PRINT *,"ENTER 4 IF RANGE IS 3000-4499 M"	042820
PRINT *,"ENTER 5 IF RANGE IS 4500 M OR MORE"	042830
GO TO 170	042840
160 CONTINUE	042850
C	042860
FDF=0.	042870
DO 170 K=16,32	042880
170 FDF=FDF+ELHT(K,L)*OA(K-15)*PCNSH(L)	042890
IF(INX.EQ."N")GO TO 180	042900
DO 180 K=48,55	042910
IF(K.EQ.52)GO TO 180	042920
FDF=FDF+ELHT(K,L)*OPERA(K-47)/2	042930
180 CONTINUE	042940
190 CONTINUE	042950
C	042960
C READ CLGP PK AND DEGRADATION FACTORS (PREC,DEGRAD)	042970
C	042980
CALL GPCNMS(3,KEY,171,3)	042990
CALL READMS(3,PREC,59,35)	043000
CALL CLOSMS(3)	043010
IS=1	043020
IF(INX.EQ."Y")IS=2	043030
IF(ICH.GE.1.AND.ICH.LE.5)DEGRAD(6-ICH,IVIS,IS)	043040

IF(ICH.EQ.0)G=0	343052
DO 230 K=16,55	343060
IF(ELMT(K,L).LE.3)GO TO 230	343072
IF(K.GT.32.AND.K.LT.48)GO TO 230	343080
IF(INX.EQ."N".AND.K.GT.32)GO TO 230	343090
IF(K.EQ.52)GO TO 230	343100
IF(K.GE.48)GO TO 280	343110
QAV=QAI(K-15)*PCNSM(L)	343120
200 IF(K.GE.48)QAV=QPERA(K-47)/2	343130
FD=PREC(4)	343140
IF(K.GT.17)FD=PREC(5)	343150
IF(K.GT.20)FD=PREC(7)	343160
IF(K.EQ.21.OR.K.EQ.22.OR.K.EQ.26.OR.K.EQ.28)FD=PREC(6)	343170
IF(K.EQ.30)FD=PREC(5)	343180
IF(K.EQ.31.OR.K.EQ.32)FD=PREC(6)	343190
IF(K.GE.48)FD=PREC(5)	343200
PRCN=PREC(8)	343210
IF(INX.EQ."Y".AND.K.GE.44)PRCN=PREC(9)	343220
FD=FD*PRCN	343230
IPSN=1	343240
IF(L.EQ.ID)IPSN=2	343250
C	343260
IF(ELMT(K,L)*QAV.LE.3)GO TO 230	343270
R=CM*ELMT(K,L)*QAV/PCF	343280
IF(K.GE.16.AND.K.LE.32)R=2*PCNSM(L)	343290
CMS=CM*F	343300
R=R*G*PEXP*QAV	343310
C	343320
DEGRADE FOR FALSE TARGETS	343330
R=R*.7	343340
C	343350
T=ELMT(K,L)*QAV*PSN(ENGAG,IPSN,2)	343360
IF(INX.EQ."Y".AND.K.GE.50)T=ELMT(K,L)*QAV	343370
IF(FD/T.GT.1)GO TO 230	343380
AKILL=(1-(1-FD/T)**(2*Q))**T	343390
AKILL=IFIX(AKILL*10+.5)/10.	343400
C	343410
BRANCH FOR PERSONNEL TARGETS	343420
IF(K.EQ.3)GO TO 210	343430
ALOSS(49,K)=ALOSS(49,K)+2*FIX(AKILL*10+.5)G(1)*PACK(L)	343440
PLOSS(K,L,1)=PLOSS(K,L,1)+FIX(AKILL*10+.5)G(1)/13.	343450
C	343460
ACCESS CREW KILLS.	343470
IF(K.GT.12)ALOSS(49,2)=ALOSS(49,2)+FIX(AKILL*CFEWS(K-12,L)*13+.5)G(1)*PACK(L)	343480
IF(K.GT.12)PLOSS(2,L,1)=PLOSS(2,L,1)+FIX(AKILL*CFEWS(K-12,L)*13+.5)G(1)/13.	343490
C+.001/10.	343500
IF(K.LT.21.OR.K.GT.25.OR.IMOUNT.EQ.1)GO TO 230	343510
IF(L.EQ.ID)GO TO 230	343520

	AKILL=IF1)(AKILL*50.+5)/10.	143530
C		143540
C	ASSESSMENT OF DISMOUNTED INFANTRY WEAPONS.	143550
	210 DO 220 KK=3,15	143560
	IF (ELMT(KK,L)-PLOSS(KK,L,2).LE.C.)GC TO 220	143570
	ALOSS(49,KK)=ALOSS(49,KK)+IFIX(AKILL*PLT(KK)*10.+001)*PACK(L)	143580
	PLOSS(KK,L,1)=PLOSS(KK,L,1)+IFIX(AKILL*PLT(KK)*10.+001)/10.	143590
	220 CONTINUE	143600
	230 CONTINUE	143610
C		143620
	CN=CMS	143630
	SHOTSV(35,J)=SHOTSV(35,J)+2.*CN	143640
	RETURN	143650
C		143660
	END	143670

SUBROUTINE FOPCE	143690
COMMON IA, ID, IP, IENGAG, IEPON, IVIS, IMOUNT, MINES, CPOB, FSFPA, FPR,	143710
1 ATIME, IFIPST, IRUN, NCRC, FMASS, IAXKV,	143710
2 SF(2), FSSF(2), PACK(2),	143720
3 ELMT(10,2), ALOSS(66,66), SHOTS(5,2), CKILL(5,2), SHOTSV(55,2)	143730
COMMON/REEJ/JOAY1, XINX(4), ICARD(20), IARPCF	143740
COMMON/DATA/FPS(80,2), CREWS(53,2), AFOS(12), OPOS(6), ITHER(2,2),	143750
1 PSW(6,2,2), PLT(15), KEY(171)	143760
COMMON/EM/PCDGR(2,4), ICOUNT(4), ISMOKE(2), PCSMO(2), ECSSM(2), ARD(2),	143770
1 NMU(2)	143780
COMMON/ONE/LEIT(35), ARRAY(90), NYBUF(1024), C(80,2), ACT,	143790
ASCENE, ASECT	143800
COMMON/TWO/IFIT(35), BARRAY(46), NYBUF(1024)	143810
COMMON/THREE/THIST(35), AH(90), IVBUF(1024)	143820
1 DIMENSION CIL(5)	143830
DIMENSION IRRAY(2)	143840
EQUIVALENCE (ARRAY, IRRAY)	143850
DATA (CIL(I), I=1,5)/1000., 5., 4., 2.5, 1./	143860
DO 100 J=1,2	143870
NMU(J)=0	143880
DO 100 I=1,90	143890
0(I, J)=0.	143900
100 ELMT(I, J)=0.0	143910
IF (ACT.NE.0.) GO TO 180	143920
C	143930
110 PRINT *, "ENTER CI MNEMONIC -"	143940
CALL REECB(ACI)	143950
120 FORMAT(1A10)	143960
CALL CKACI(ACI0)	143970
IF (ACI.NE.0..AND.ACT.NE.ACIO) GO TO 180	143980
IF (ACI.EQ.0.) GO TO 110	143990
ASECT=0.	144000
PRINT 130, ACI	144010
130 FORMAT(" YOU CANNOT DO ASSESSMENTS FOR CI ", A10)	144020
C	144030
ACI=0.	144040
C 140 PRINT *, "DO YOU WISH TO RESET CI MNEMONIC? "	144050
C CALL REECA(IYN)	144060
C IF (IYN.EQ."Y") GO TO 110	144070
C IF (IYN.EQ."N") GO TO 130	144080
C PRINT 500	144090
C GO TO 140	144100
150 FORMAT(" ", 1A10)	144110
IF (IST=-1)	144120
GO TO 190	144130
160 FORMAT(1A1)	144140
170 FORMAT(" ", 1A1)	144150
C	144160
190 PRINT *, "ENTER SECTOR NUMBER -"	144170

CALL REED4	344170
ASECT=XINX(1)	344180
IF(ASECT.GE.0.AND.ASECT.LE.99999)GO TO 190	344190
PRINT *, "INVALID SECTOR ENTERED!"	344200
GO TO 180	344210
C	344220
190 ISEED=ACI+ASECT	344230
CALL RANSET(ISEED)	344240
IREGM=0	344250
IF(IRUN.EQ.1.OR.IRUN.EQ.3)IREG4="Y"	344260
CALL CLFAR(IREGM)	344270
200 PRINT *, "???"	344280
CALL REED4	344290
INX=XINX(1)	344300
IF(INX.EQ.999)GO TO 210	344310
IF(INX.GE.0.AND.INX.LE.9)GO TO 220	344320
PRINT *, "INVALID OPTION ENTERED!"	344330
210 PRINT *, "ENTER 0 TO PROCEED WITH ASSESSMENTS"	344340
PRINT *, "1 TO LOAD UNITS INTO SECTOR"	344350
PRINT *, "2 TO REMOVE UNITS FROM SECTOR"	344360
PRINT *, "3 TO CREATE A NEW UNIT"	344370
PRINT *, "4 TO ADJUST WPNS IN A UNIT"	344380
PRINT *, "5 TO ATTACH A UNIT TO A NEW PARENT"	344390
PRINT *, "6 TO DISPLAY A UNIT"	344400
PRINT *, "7 TO DELETE A UNIT FROM FORCE FILE"	344410
PRINT *, "8 TO COMBINE UNITS"	344420
PRINT *, "9 TO AUTOLOAD SECTOR"	344430
GO TO 230	344440
220 IF(INX.EQ.0)GO TO 1190	344450
GO TO (290,760,470,380,850,880,680,280,230), INX	344460
C	344470
C	344480
230 PRINT *, "LOAD UNITS INTO THIS SECTOR FROM PREVIOUS CI?"	344490
CALL REED4(INX)	344500
IF(INX.EQ."Y")GO TO 240	344510
IF(INX.EQ."N")GO TO 230	344520
PRINT 500	344530
GO TO 230	344540
C	344550
240 INC=0	344560
CALL OPENM(LFIT,3LI-0,1LR)	344570
250 CALL GETN(LFIT,ARRAY,ARRAY(1))	344580
M=IFETCH(LFIT,2LFR)	344590
IF(M.EQ.1008)GO TO 270	344600
IF(ARRAY(4).EQ.ASECT.OR.ARRAY(4).EQ.-1.*ASECT)GO TO 260	344610
GO TO 250	344620
C	344630
260 ARRAY(5)=ACI	344640

AD-AU92 783

ARMY COMBINED ARMS STUDIES AND ANALYSIS ACTIVITY FOR--ETC F/6 15/7
CACDA JIFFY III WAR GAME, VOLUME V. PROGRAMMER'S MANUAL.(U)

OCT 80 C L PAO

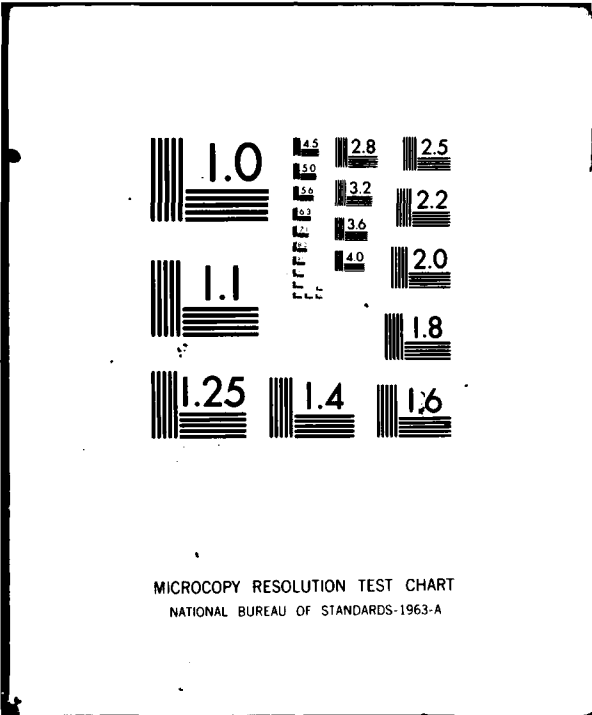
UNCLASSIFIED

CASAA-TR-9-80

NL

5
10

END
DATE
FILMED
1
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

CALL REPLC(LFIT,ARRAY,900,ARRAY(1))	044650
IF(ARRAY(4).EQ.ASECT)INC=INC+1	044660
GO TO 250	044670
C	044680
270 CALL CLOSE(LFIT)	044690
PRINT *,INC," UNITS WERE LOADED INTO THIS SECTOR"	044700
GO TO 280	044710
C	044720
C	044730
280 CALL COMBINE	044740
GO TO 290	044750
C	044760
C	044770
290 PRINT *,"ENTER PARENT OF UNIT(S) TO BE LOADED INTO SECTOR --"	044780
CALL REEDR(ARRAY(1))	044790
PARENT=ARRAY(1)	044800
ARRAY(7)=99999	044810
CALL OPEN(LFIT,3LI=0,1LR)	044820
CALL GE(LFIT,ARRAY,ARRAY(1),0,10)	044830
IF(ARRAY(7).NE.99999)GO TO 310	044840
PRINT 100,PARENT	044850
300 FORMAT(" UNIT ",A10," IS NOT ON FCPCE FILE!")	044860
GO TO 360	044870
C	044880
C	044890
310 PRINT *,"ENTER UNIT ID (OR ALL) --"	044900
CALL REEDR(UNIT)	044910
C	044920
IFLAG=1	044930
320 IF(UNIT.EQ."ALL")GO TO 340	044940
IF(ARRAY(2).EQ.UNIT)GO TO 340	044950
330 CALL GETN(LFIT,ARRAY,ARRAY(1))	044960
M=IFETCH(LFIT,2LPP)	044970
IF(M.EQ.1008)GO TO 350	044980
IF(ARRAY(1).NE.PARENT)GO TO 350	044990
GO TO 320	045000
C	045010
340 ARRAY(4)=ASEC	045020
ARRAY(5)=ACI	045030
IFLAG=1	045040
CALL REPLC(LFIT,ARRAY,900,ARRAY(1))	045050
GO TO 330	045060
C	045070
350 IF(IFLAG.EQ.0)PRINT 300,UNIT	045080
360 CALL CLOSE(LFIT)	045090
370 PRINT *,"LOAD ANOTHER UNIT?"	045100
CALL REEDA(INX)	045110
IF(INX.EQ."Y")GO TO 290	045120

IF(INX.EQ."N")GO TO 200	045130
PRINT 500	045140
GO TO 370	045150
C	045160
C	045170
380 PRINT *, "ENTER PARENT OF UNIT TO BE ADJUSTED -"	045180
CALL REED08(ARRAY(1))	045190
C	045200
PRINT *, "ENTER UNIT I:-"	045210
CALL REED08(ARRAY(2))	045220
C	045230
ARRAY(7)=90909	045240
CALL OPENH(LFIT,3LI-0.1LR)	045250
CALL GET(LFIT,ARRAY,ARRAY(1))	045260
IF(ARRAY(7).EQ.90909)GO TO 460	045270
PRINT *, "ENTER WPN ID,NEW QTY--0.0 WHEN DONE "	045280
390 CALL REED06	045290
MM=XINX(1)	045300
AJ=XINX(2)	045310
IF(MM.EQ.0) GO TO 420	045320
IF(MM.LT.0.OR.MM.GT.80)GO TO 410	045330
IF(ARRAY(MM+10)+AJ.LT.0)GO TO 400	045340
ARRAY(MM+10)=ARRAY(MM+10)+AJ	045350
PRINT *, "NEXT--"	045360
GO TO 390	045370
400 PRINT *, "ENTRY REDUCES * WEAPONS IN UNIT BELOWZERO - ENTRY IGNORED"	045380
GO TO 390	045390
410 PRINT *, "INVALID ITEM CODE - ENTRY IGNORED!"	045400
GO TO 390	045410
420 IF(IRUN.EQ.1)GO TO 440	045420
J=ARRAY(3)	045430
UEFF=0.	045440
DO 430 I=1,80	045450
IF(ARRAY(I+10).LE.0)GO TO 430	045460
UEFF=UEFF+ARRAY(I+10)*POS(I,J)	045470
430 CONTINUE	045480
IF(ARRAY(6).EQ.0)GO TO 435	045490
UEFF=UEFF/ARRAY(6)*100.	045500
435 ARRAY(9)=UEFF	045510
CALL REFLC(LFIT,ARRAY,900,ARRAY(1))	045520
440 CALL CLOSEH(LFIT)	045530
450 PRINT *, "ANYMORE UNITS TO CHANGE? "	045540
CALL REED04(INX)	045550
IF(INX.EQ."Y")GO TO 390	045560
IF(INX.EQ."N")GO TO 200	045570
PRINT 500	045580
GO TO 450	045590

460 PRINT 300, ARRAY(2)	345610
APRAY(7)=0.	345620
GO TO 440	345630
C	345640
470 KING="BLUE"	345650
ARRAY(1)="SRC"	345660
J=1	345670
480 PRINT 490, KING	345680
490 FORMAT(" ARE THERE ANY ".A4." UNITS TO CREATE?")	345690
CALL REEDA(INX)	345700
IF(INX.EQ."Y")GO TO 510	345710
IF(INX.EQ."N")GO TO 670	345720
PRINT 500	345730
GO TO 480	345740
500 FORMAT(" INCORRECT! RESPONSE MUST BE YES OR NO - TRY AGAIN")	345750
C	345760
510 ARRAY(3)=J	345770
DO 520 INX=11,90	345780
520 ARRAY(INX)=0	345790
PRINT *, "ENTER PARENT UNIT ID -"	345800
CALL REEDB(ARRAY(1))	345810
C	345820
PRINT *, "ENTER UNIT ID -"	345830
CALL REEDB(ARRAY(2))	345840
530 PRINT *, "CREATE BY SPC'S?"	345850
CALL REEDA(INX)	345860
IF(INX.EQ."Y")GO TO 610	345870
IF(INX.EQ."N")GO TO 540	345880
PRINT 500	345890
GO TO 530	345900
540 APRA(5)=0	345910
PRINT *, "ENTER WPN ID, QTY--C,0 WHEN DONE "	345920
550 CALL REEDC	345930
MM=XINX(1)	345940
AJ=XINX(2)	345950
IF(MM.EQ.0) GO TO 560	345960
ARRAY(MM+10)=ARRAY(MM+10)+AJ	345970
ARRAY(6)=ARRAY(6)+AJ*FPS(MM, J)	345980
PRINT *, "NEXT -"	345990
GO TO 550	346000
560 PRINT *, "ENTER RELATIVE EFFECTIVENESS -"	346010
CALL REEDC	346020
CV=XINX(1)	346030
IF(CV.GE.0..AND.CV.LE.10000)GO TO 570	346040
PRINT *, "INVALID REL. EFF. - TRY AGAIN!"	346050
GO TO 540	346060
570 ARRAY(6)=ARRAY(6)+100./CV	346070
ARRAY(5)=CV	346080

C	IF (IRUN.EQ.1) GO TO 580	046090
	CALL OPENM(LFIT,3LI-0,1LR)	046100
	CALL PUT(LFIT,ARRAY,900,ARRAY(1))	046110
	CALL CLOSEM(LFIT)	046120
	580 DO 590 I=1,90	046130
	590 ARRAY(I)=0	046140
	600 PRINT *, "CREATE ANOTHER UNIT FOR THIS FORCE?"	046150
	CALL RECA(INX)	046160
	IF (INX.EQ."V") GO TO 510	046170
	IF (INX.EQ."N") GO TO 670	046180
	PRINT 500	046190
	GO TO 600	046200
	610 PRINT *, "ENTER SRC--0 WHEN DONE"	046210
	ARRAY(6)=0	046220
	620 CALL RECB(ARRAY(2))	046230
	IF (BARRAY(2).EQ."0") GO TO 560	046240
	BARRAY(3)=90909	046250
	CALL OPENM(IFIT,3LI-0,1LR)	046260
	CALL GET(IFIT,BARRAY,BARRAY(1))	046270
	CALL CLOSEM(IFIT)	046280
	IF (BARRAY(3).EQ.90909) GO TO 650	046290
	DO 630 JJ=1,22	046300
	JJ=JJ*2+1	046310
	INX=BARRAY(JJ)	046320
	IF (INX.EQ.0) GO TO 630	046330
	ARRAY(INX+10)=ARRAY(INX+10)+BARRAY(JJ+1)	046340
	ARRAY(6)=ARRAY(6)+BARRAY(JJ+1)*FPS(INX,J)	046350
	630 CONTINUE	046360
	DO 640 I=3,24	046370
	640 BARRAY(I)=0	046380
	PRINT *, "NEXT--"	046390
	GO TO 620	046400
	650 PRINT 660,BARRAY(2)	046410
	BARRAY(3)=0	046420
	660 FORMAT(" SRC ",1A10," NOT ON FILE")	046430
	PRINT *, "NEXT--"	046440
	GO TO 620	046450
C	670 IF (J.EQ.2) GO TO 230	046460
	J=2	046470
	KIND="READ"	046480
	GO TO 440	046490
C	680 PRINT *, "ENTER PARENT OF UNIT(S) TO BE DELETED --"	046500
	CALL RECB(ARRAY(1))	046510
	CALL OPENM(LFIT,3LI-0,1LR)	046520
	PARENT=ARRAY(1)	046530
		046540
		046550
		046560

ARRAY(7)=90909	34657C
CALL GET(LFIT,ARRAY,ARRAY(1),0,10)	34659C
IF(ARRAY(7).NE.90909)GO TO 690	34659C
PRINT 300,PARENT	34660C
GO TO 740	34661C
C	34662C
690 PRINT *,"ENTER UNIT ID (OR ALL) -"	34663C
CALL REECB(UNIT)	34664C
C	34665C
IFLAG=0	34666C
700 IF(UNIT.EQ."ALL")GO TO 720	34667C
IF(UNIT.EQ.ARRAY(2))GO TO 720	34668C
710 CALL GETN(LFIT,ARRAY,ARRAY(1))	34669C
M=IFETCH(LFIT,2LFF)	34670C
IF(M.EQ.1000)GO TO 730	34671C
IF(ARRAY(1).NE.PARENT)GO TO 730	34672C
GO TO 700	34673C
C	34674C
720 CALL CLCE(LFIT,ARRAY(1))	34675C
IFLAG=1	34676C
GO TO 710	34677C
C	34678C
730 IF(IFLAG.EQ.1)PRINT 300,UNIT	34679C
740 CALL CLOSE(LFIT)	34680C
750 PRINT *,"ANOTHER UNIT TO DELETE?"	34681C
CALL REEOA(INX)	34682C
IF(INX.EQ."Y")GO TO 680	34683C
IF(INX.EQ."N")GO TO 200	34684C
PRINT 500	34685C
GO TO 750	34686C
C	34687C
C	34688C
760 PRINT *,"ENTER PARENT OF UNIT(S) TO BE REMOVED -"	34689C
CALL REECB(ARRAY(1))	34690C
CALL OPEN(LFIT,3LI-0,1LR)	34691C
PARENT=ARRAY(1)	34692C
ARRAY(7)=90909	34693C
CALL GET(LFIT,ARRAY,ARRAY(1),1,10)	34694C
IF(ARRAY(7).NE.90909)GO TO 770	34695C
PRINT 300,PARENT	34696C
GO TO 930	34697C
C	34698C
770 PRINT *,"ENTER UNIT ID (OR ALL) -"	34699C
CALL REEOB(UNIT)	34700C
C	34701C
IFLAG=0	34702C
780 IF(UNIT.EQ."ALL")GO TO 930	34703C
IF(UNIT.EQ.ARRAY(2))GO TO 300	34704C

790	CALL GETN(LFIT,ARRAY,ARRAY(1))	047090
	M=J-ETCH(LFIT,2LFP)	047100
	IF(M.EQ.1)GO TO 820	047110
	IF(ARRAY(1).NE.PARENT)GO TO 820	047120
	GO TO 790	047130
C		047140
800	IF(ARRAY(4).GT.0,ARRAY(4)=-1,ARRAY(4))	047150
	ARRAY(5)=ACI	047160
	IFLAG=1	047170
	GO 810 IT=76.90	047180
	ARRAY(II)=0.	047190
810	CONTINUE	047200
	CALL REPLC(LFIT,ARRAY,900,ARRAY(1))	047210
	GO TO 790	047220
C		047230
820	IF(IFLAG.EQ.0)PRINT 300,UNIT	047240
C		047250
830	CALL CLOSE(LFIT)	047260
840	PRINT *,"REMOVE ANOTHER UNIT?"	047270
	CALL REECA(INX)	047280
	IF(INX.EQ."Y")GO TO 760	047290
	IF(INX.EQ."N")GO TO 230	047300
	PRINT 900	047310
	GO TO 840	047320
C		047330
C		047340
850	PRINT *,"ENTER PARENT IC OF UNIT BEING ATTACHED -"	047350
	CALL REECB(PARENT)	047360
C		047370
	PRINT *,"ENTER UNIT ID -"	047380
	CALL REECB(ARRAY(2))	047390
C		047400
	CALL OPEN(LFIT,3LI-0,1LP)	047410
	ARRAY(1)=PARENT	047420
	ARRAY(7)=99999.	047430
	CALL GET(LFIT,ARRAY,ARRAY(1))	047440
	IF(ARRAY(7).EQ.99999.)GO TO 870	047450
C		047460
	PRINT *,"ENTER NEW PARENT IC -"	047470
	CALL REECB(PAP)	047480
	ARRAY(1)=PAR	047490
	CALL PUT(LFIT,ARRAY,900,ARRAY(1))	047500
	ARRAY(1)=PARENT	047510
	CALL OLTE(LFIT,ARRAY(1))	047520
	CALL CLOSE(LFIT)	047530
C		047540
860	PRINT *,"ATTACH ANOTHER UNIT?"	047550
	CALL REECA(INX)	047560

	IF(INX.EQ."Y")GO TO 150	047530
	IF(INX.EQ."N")GO TO 200	047540
C		047550
	170 PRINT 307,UNIT	047560
	ARRAY(7)=0.	047570
	GO TO 160	047580
C		047590
C	DISPLAY SECTION	047600
	180 PRINT *, "ENTER TYPE OF DISPLAY -"	047610
	CALL REED4	047620
	INX=XINX(1)	047630
	IF(INX.EQ.999)GO TO 890	047640
	IF(INX.GE.1.AND.INX.LE.4)GO TO 910	047650
	PRINT *, "INCORRECT ENTRY!!!"	047660
	190 PRINT *, "ENTER 1 TO DISPLAY ALL PARENT UNITS IN FORCEFILE"	047670
	PRINT 900,ASECT,ACT	047680
	200 FORMAT(" 2 TO DISPLAY ALL PARENT UNITS IN SECTOR ",F6.0,	047690
	1" IN CI ",A10)	047700
	PRINT *, " 3 TO DISPLAY UNITS IN A SPECIFIC PARENT"	047710
	PRINT *, " 4 TO DISPLAY WEAPONS IN A UNIT"	047720
	GO TO 890	047730
C		047740
	210 GO TO (920,923,1078,1179),INX	047750
C		047760
C	DISPLAY ALL PARENT UNITS	047770
	920 PARENT="ALL"	047780
	IFLAG=0	047790
	UEFF=0.	047800
	TFPS=0.	047810
	CALL OPENM(LFIT,3LI-0,1LR)	047820
	930 CALL GETN(LFIT,ARRAY,ARRAY(1))	047830
	M=IFETCH(LFIT,2LF0)	047840
	IF(M.EQ.1008)GO TO 1030	047850
	IF(ARRAY(1).NE.PARENT)GO TO 970	047860
	940 XSECT=ARRAY(4)	047870
	XCI=ARRAY(5)	047880
	IF(INX.EQ.1)GO TO 950	047890
	IF(ASECT.NE.XSECT.OR.ACI.NE.XCI)GO TO 930	047900
	950 J=ARRAY(3)	047910
	TFPS=TFPS+ARRAY(6)	047920
	DO 960 I=1,30	047930
	IF(ARRAY(I+10).LE.0)GO TO 960	047940
	UEFF=UEFF+ARRAY(I+10)*FPS(I,J)	047950
	960 CONTINUE	047960
	GO TO 930	047970
C		047980
	970 IF(IFLAG.EQ.1)GO TO 1000	047990
	PRINT *, " "	048000

IF(INX.EQ.1)GO TO 940	049010
PRINT *,"FORCE ID EFF"	049020
GO TO 990	049030
980 PRINT *,"FORCE ID EFF SECT C1"	049040
990 IFLAG=1	049050
GO TO 1060	049060
1000 IF(M.EQ.1008.AND.IFLAG.EQ.0)GO TO 1160	049070
IF(INX.EQ.2.AND.(ASECT.NE.XSECT.OR.ACI.NE.XCI))GO TO 1060	049080
IF(TFPS.GT.0)GO TO 1010	049090
UEFF=999.	049100
GO TO 1020	049110
1010 UEFF=UEFF/TFPS*100.	049120
1020 IF(INX.EQ.1)GO TO 1040	049130
PRINT 1030,J,PARENT,UEFF	049140
1030 FORMAT(" ",I3,4X,A10,2X,F4,C)	049150
GO TO 1060	049160
1040 PRINT 1050,J,PARENT,UEFF,XSECT,XCI	049170
1050 FORMAT(" ",I3,4X,A10,2X,2(F4.0,2X),A10)	049180
1060 PARENT=ARRAY(1)	049190
UEFF=0.	049200
TFPS=0.	049210
IF(M.EQ.1008)GO TO 1160	049220
GO TO 940	049230
C	049240
DISPLAY UNITS IN SPECIFIC UNIT	049250
1070 PRINT *,"ENTER PARENT ID ="	049260
CALL REEUB(PARENT)	049270
ARRAY(1)=PARENT	049280
C	049290
ARRAY(7)=99999.	049300
IFLAG=0	049310
CALL OPENH(LFIT,3LI-0,1LR)	049320
CALL GET(LFIT,ARRAY,ARRAY(1),0,10)	049330
IF(ARRAY(7).NE.99999.)GO TO 1090	049340
PRINT 300,PARENT	049350
GO TO 1160	049360
C	049370
1080 CALL GETN(LFIT,ARRAY,ARRAY(1))	049380
M=IFETCH(LFIT,2LFP)	049390
IF(M.EQ.1009)GO TO 1160	049400
IF(PARENT.NE.ARRAY(1))GO TO 1160	049410
1090 UEFF=0.	049420
J=ARRAY(3)	049430
IF(ARRAY(4).LT.0.)ARRAY(4)=0.	049440
DO 1100 I=1,80	049450
TF(ARRAY(I+10).LE.0.)GO TO 1100	049460
UEFF=UEFF+ARRAY(I+10)*FPS(I,J)	049470
1100 CONTINUE	049480

IF (ARRAY(6).GT.0.)GO TO 1113	048490
UEFF=999.	048500
GO TO 1120	048510
1110 UEFF=UEFF/ARRAY(6)*100.	048520
1120 IF (IFLAG.EQ.1)GO TO 1147	048530
IFLAG=1	048540
PRINT *, " "	048550
PRINT *, "FORCE PARENT UNIT EFF SECT CI"	048560
PRINT 1130,J,PARENT,ARRAY(2),UEFF,ARRAY(4),ARRAY(5)	048570
1130 FORMAT(" ",I3,4X,2(A10,2X),2(F4.0,2X),A10)	048580
GO TO 1040	048590
1140 PRINT 1150,ARRAY(2),UEFF,ARRAY(4),ARRAY(5)	048600
1150 FORMAT(" ",19X,A10,2X,2(F4.0,2X),A10)	048610
GO TO 1040	048620
1160 PRINT *, " "	048630
CALL CLOSE(LEFT)	048640
GO TO 1140	048650
1170 CALL DISPLAY	048660
1180 PRINT *, "ANOTHER DISPLAY?"	048670
CALL REEDA(INX)	048680
IF (INX.EQ."Y")GO TO 0 000	048690
IF (INX.EQ."N")GO TO 0 200	048700
PRINT 000	048710
GO TO 1190	048720
C	048730
1190 PRINT *, "DO YOU WANT TO SEE CODES?"	048740
CALL REEDA(INX)	048750
IF (INX.EQ."Y")GO TO 1195	048760
GO TO 2220	048770
1195 CONTINUE	048780
00 2210 JKIND=1,2	048790
CALL OPENH(LEFT,3LI-0,1LR)	048800
PRINT 2180,ASECT,ACI	048810
2180 FORMAT(" UNITS LOADED INTO SECTOR ",F6.0," FOR CI ",A10, "/",	048820
1 " FORCE PARENT UNIT NEW PARENT NEW UNIT")	048830
2190 CALL GETN(LEFT,ARRAY,ARRAY(1))	048840
N=IFETCH(LEFT,2LFB)	048850
IF (N.EQ.1000)GO TO 2205	048860
J=ARRAY(3)	048870
IF (ARRAY(4).NE.ASECT.OP.ACI.NE.ARRAY(5).CR...VE.JKIND)GO TO 2190	048880
IR1=IRRAY(1)	048890
IR2=IRRAY(2)	048900
CALL TRANS(IR1,IR1)	048910
CALL TRANS(IR2,IR2)	048920
PRINT 2200,J,ARRAY(1),ARRAY(2),IR1,IR2	048930
2200 FORMAT(1X,I3,8X,A10,3X,A10,1X,A10,5X,A10)	048940
GO TO 2190	048950
2205 CALL CLOSE(LEFT)	048960

2210 CONTINUE	344970
C 2220 IF (ZPEGM.EQ."N") GO TO 1220	344980
CALL OPENH(LFIT,3LI-0.1LR)	344990
IPUT=-1	049000
AM1="CI AM"	349010
CALL GKSTOR(AM1,IPUT)	349020
AVN=0.	349030
NU=0	349040
DO 1200 I=1,4	349050
AM(I+3)=0.	349060
ICOUNT(I)=0	349070
1200 CONTINUE	349080
1210 PRINT *, "DO YOU WISH TO SEE UNITS LOADED INTO SEC-OR?"	349090
CALL REEDA(INX)	349100
IF (INX.EQ."Y".OR.INX.EQ."N") GO TO 1220	349110
PRINT 500	349120
GO TO 1210	349130
1220 IF (INX.EQ."N") GO TO 1240	349140
C POINT 1230,ASECT,ACT	349150
1230 FORMAT(" UNITS LOADED INTO SECTOR ",F6.0," FOR CI ",A10,"//",	349160
1 " POPCE PARENT UNIT TYPE CIL EFF")	349170
1240 CALL GETN(ILFIT,ARRAY,ARRAY(1))	349180
N=IFETCH(LFIT,2LFF)	349190
IF (N.EQ.1000) GO TO 1290	349200
C IF (ARRAY(4).NE.ASECT.AND.ARRAY(4).NE.-1.ASECT) GO TO 1240	349210
IF (ARRAY(5).NE.AC1) GO TO 1240	349220
IF (ARRAY(4).LT.0.) GO TO 1240	349230
J=ARRAY(3)	349240
IEFF=ARRAY(9)	349250
ICIL=ARRAY(9)	349260
IPRO=ARRAY(10)	349270
IF (INX.EQ."N") GO TO 1260	349280
POINT 1250,J,ARRAY(1),ARRAY(2),IPRO,ICIL,IEFF	349290
1250 FORMAT(" ",I3,8X,A10,3X,A10,3X,I2,3X,I2,4X,I3,4X)	349300
1260 DO 1270 I=1,80	349310
IF (ARRAY(I+10).EQ.0) GO TO 1270	349320
ELMT(I,J)=ELMT(I,J)+ARRAY(I+10)	349330
1270 CONTINUE	349340
C IF (IPRO.EQ.0) GO TO 1240	349350
II=IPRO+4*(J-1)	349360
ICOUNT(II)=ICOUNT(II)+1	349370
IF (J.EQ.1.AND.IPRO.EQ.4.AND.ICIL.EQ.4) NHU(J)=NHU(J)+1	349380
IF (J.EQ.2.AND.IPRO.EQ.4.AND.ICIL.EQ.4) NHU(J)=NHU(J)+3	349390
IF (IPRO.NE.2.OR.ARRAY(1).EQ.AVN.OR.J.NE.1) GO TO 1240	349400

IF (APRAY(71).EQ.0..AND..APRAY(72).EQ.0.)GO TO 1240	049450
AM(NU)=APRAY(1)	049460
AVN=APRAY(1)	049470
NU=NU+1	049480
GO TO 1240	049490
C	049500
1290 CALL CLOSE=(LEFT)	049510
IFIRST=0	049520
IF (NU.GT.9)AM(9)="PLUS"	049530
IF (AM(4).NE.0.)CALL CKSTOR(AM1,IPUT)	049540
AM1="BLUE START"	049550
J=1	049560
1300 IPUT=-1	049570
CALL CKSTOR(AM1,IPUT)	049580
DO 1310 I=1,80	049590
AM(I+17)=ELMT(I,J)	049600
1310 CONTINUE	049610
CALL CKSTOR(AM1,IPUT)	049620
IF (J.EQ.2)GO TO 1320	049630
J=2	049640
AM1="RED START"	049650
GO TO 1300	049660
1320 CONTINUE	049670
C DEGRADE * OF ARTY *UBES ACTIVE IN SECTOR	049680
C DO 1330 J=1,2	049690
C DO 1340 I=43,99	049700
C ELMT(I,J)=ELMT(I,J)*.85	049710
C1340 CONTINUE	049720
C1330 CONTINUE	049730
C PRINT *, "ARTY DEGRADED"	049740
C	049750
END	049760

	SUBROUTINE COMBINE	049790
	COMMON/ONE/LFIT(35),ARRAY(90),M19UF(1024),D(80,2),ACT,	049790
	ASCENE,ASECT,YSECT	049800
	DIMENSION W(90)	049810
C	INPUT HOST & NEW "TENANT"	049820
100	PRINT *, " PARENT ID OF HOST = "	049830
	CALL REECB(MPAR)	049840
	PRINT *, " UNIT ID = "	049850
	CALL REECB(MUNIT)	049860
110	PRINT *, " PARENT ID OF DISBANDING UNIT = "	049870
	CALL REECB(OPAR)	049880
	PRINT *, " UNIT IC = "	049890
	CALL REECB(OUNIT)	049900
C	CHECK IF HOST AND TENANT ARE DIFFERENT	049910
	I=0	049920
	IF(MPAR.EQ.OPAR.AND.MUNIT.EQ.OUNIT) I=1	049930
	IF(I.EQ.1)PRINT *, " DISBANDING UNIT IS SAME AS HOST"	049940
	IF(I.EQ.1)GO TO 140	049950
C	CHECK IF HOST IS ON FORCE FILE	049960
	I=0	049970
	ARRAY(7)=90909	049980
	ARRAY(1)=MPAR	049990
	ARRAY(2)=MUNIT	050000
	CALL OPENH(LFIT,3LI=0,1LR)	050010
	CALL GET(LFIT,ARRAY,ARRAY(1))	050020
	IF(ARRAY(7).EQ.90909)I=1	050030
	IF(I.EQ.1)PRINT *, " HOST NOT ON FORCE FILE"	050040
	IF(I.EQ.1)GO TO 140	050050
C	CHECK IF TENANT IS ON FORCE FILE	050060
	I=0	050070
	ARRAY(7)=90909	050080
	ARRAY(1)=OPAR	050090
	ARRAY(2)=OUNIT	050100
	CALL REWNO(LFIT)	050110
	CALL GET(LFIT,ARRAY,ARRAY(1))	050120
	IF(ARRAY(7).EQ.90909)I=1	050130
	IF(I.EQ.1)PRINT *, " DISBANDING UNIT IS NOT ON FORCE FILE"	050140
	IF(I.EQ.1)GO TO 140	050150
C	STORE ASSETS OF TENANT IN WORKING ARRAY	050160
	DO 120 I=1,90	050170
	W(I)=ARRAY(I+10)	050180
120	CONTINUE	050190
C	REMOVE TENANT FROM FORCE FILE	050200
	CALL OLTE(LFIT,ARRAY(1))	050210
C	ADD TENANT ASSETS TO HOST ASSETS	050220
	ARRAY(1)=MPAR	050230
	ARRAY(2)=MUNIT	050240
	CALL REWNO(LFIT)	050250

CALL GET(LFIT,ARRAY,ARRAY(1))	150250
DO 130 I=1,90	050270
ARRAY(I+10) = ARRAY(I+10) + W(I)	050290
130 CONTINUE	050290
CALL REPLC(LFIT,ARRAY,900,ARRAY(1))	050300
CALL CLOSE(LFIT)	050310
140 RETURN	050320
END	050330

SUBROUTINE CKACT(ACIO)	350350
COMMON/ONE/LFIT(35),ARRAY(90),MYBUF(1024),D(63,2),ACI,	350360
ASCENE,ASECT	350370
COMMON/THREE/THIST(35),AH(90),IYBUF(1024)	350380
DIMENSION CCI(2)	350390
IF(ACI.NE."XXX".AND.ACI.NE."ALL".AND.ACI.NE."INITIAL")	350400
*GO TO 100	350410
PRINT *, "CI MNEMONICS ---XXX--- --INITIAL--- --ALL--- ARE NOT ALLOWED.	350420
2-	350430
ACI=0.	350440
GO TO 400	350450
100 N1=0	350460
N2=0	350470
ACIO=0.	350480
CALL CPENM(LFIT,3LI-0,1LR)	350490
DO 120 N=1,2	350500
CCI(N)=0.	350510
110 CALL GETN(LFIT,ARRAY,ARRAY(1))	350520
M=IFETCH(LFIT,2LF#)	350530
IF(M.EQ.1009)GO TO 130	350540
IF(ARRAY(5).EQ.CCI(1).OR.ARRAY(5).EQ.C.C.OR.ARRAY(5).EQ."INITIAL")	350550
*GO TO 110	350560
IF(ARRAY(5).EQ."FORCE")GO TO 110	350570
CCI(N)=ARRAY(5)	350580
120 CONTINUE	350590
130 CALL CLOSEM(LFIT)	350600
CALL CPENM(THIST,3LI-0,1LR)	350610
AH(1)="CI NAMES"	350620
AH(2)=1.	350630
140 AH(6)=99999.	350640
CALL GET(THIST,AH,AH(1),0,20)	350650
IF(AH(4).NE.99999.)GO TO 160	350660
C THIS CI NAMES RECORD NOT FOUND--ADD IT TO HISTORY FILE	350670
DO 150 I=3,90	350680
AH(I)=0.	350690
150 CONTINUE	350700
CALL PUT(THIST,AH,900,AH(1))	350710
N0=4	350720
GO TO 180	350730
C FIND FIRST ZERO WORD ON CI NAMES RECORD	350740
160 N0=0	350750
DO 170 I=4,90	350760
IF(AH(I).NE.0.)GO TO 170	350770
N0=I	350780
GO TO 190	350790
170 CONTINUE	350800
180 IF(N0.EQ.4.AND.AH(2).EQ.1)GO TO 460	350810
IF(AH(2).EQ.1.)ACIO=AH(4)	350820

IF(CCI(1).EQ.0.)GO TO 290	050430
IF(N6.EQ.4)GO TO 210	050440
DO 200 I=4,90	050450
IF(AH(I).NE.CCI(1))GO TO 190	050460
N1=I	050470
NRD1=AH(2)	050480
GO TO 200	050490
190 IF(AH(I).NE.CCI(2).CP.CCI(2).EQ.0.)GO TO 230	050500
N2=I	050510
NRD2=AH(2)	050520
200 CONTINUE	050530
210 IF(N1.EQ.0)GO TO 220	050540
IF(CCI(2).EQ.0.)GO TO 290	050550
IF(N2.EQ.0)GO TO 220	050560
IF(NRD1.EQ.NRD2)GO TO 410	050570
GO TO 390	050580
220 IF(N6.NE.0)GO TO 230	050590
AH(2)=AH(2)+1.	051000
GO TO 140	051010
230 IF(CCI(2).EQ.0.)GO TO 250	051020
PRINT 240,CCI	051030
240 FORMAT(" CI NAMES FOUND ON FORCEFILE ARE ",A10," AND ",A10/ " * " AT LEAST ONE OF THESE C.I.'S IS NOT RECORDED ON HISTORYFILE.")	051040
GO TO 270	051050
250 PRINT 260,CCI(1)	051060
260 FORMAT(" FORCEFILE IS AT END OF CI ",A10/ " * " THIS CI NAME IS NOT RECORDED ON HISTORYFILE.")	051070
270 PRINT *, "PLEASE CHECK THAT CORRECT FILES ARE ATTACHED."	051080
CALL CLOSEHIST	051090
CALL SHOWCI	051100
PRINT *, "PROGRAM STOPPED"	051110
CALL SHCWIT	051120
STOP	051130
290 CONTINUE	051140
PRINT *, "FORCEFILE HAS NO CI NAMES ON UNIT RECORDS"	051150
PRINT *, "HISTORYFILE HAS AT LEAST ONE CI RECORDED ON IT--"	051160
PRINT *, "IT SHOULD BE A BLANK FILE."	051170
GO TO 270	051180
200 XN=0.	051190
IF(N1.GE.90)GO TO 330	051200
N2=N1+1	051210
IF(AH(2).NE.NRD1)GO TO 340	051220
300 IF(ACI.EQ.AH(N2).OR.ACI.EQ.CCI(1))GO TO 470	051230
IF(N2.EQ.N0)GO TO 460	051240
PRINT 310,CCI(1),AH(N2)	051250
310 FORMAT(" FORCEFILE IS AT END OF CI ",A10/ " * " NEXT CI TO GAME MUST BE ",A10)	051260
PRINT 320,AH(N2)	051270
	051280
	051290
	051300

320	FORMAT(" CI MNEMONIC HAS BEEN SET TO ",A10)	051310
	ACI=AM(N2)	051320
	GO TO 470	051330
330	N2=6	051340
	IF(AM(2).EQ.WPD1+1.)GO TO 300	051350
	XN=1.	051360
340	AM(2)=WRC1+YN	051370
350	AM(6)=99999.	051380
	CALL GET(IHIST,AM,AM(1),0,20)	051390
	IF(AM(4).EQ.0.)NO=4	051400
	IF(AM(4).NE.99999.)GO TO 330	051410
	IF(AM(2).EQ.WRD1)GO TO 370	051420
	CO 360 I=3,90	051430
	AM(5)=0.	051440
360	CONTINUE	051450
	CALL PUT(IHIST,AM,900,AM(1))	051460
	N3=6	051470
	GO TO 300	051480
370	IF(AM(1).EQ."CI NAMES")GO TO 380	051490
	AM(1)="CI NAMES"	051500
	GO TO 350	051510
380	PRINT *, "HISTORYFILE RECORD OF CI NAMES CANNOT BE FOUND"	051520
	GO TO 270	051530
390	IF(WRD1.GT.WRD2)GO TO 400	051540
	NLAST=2	051550
	IF(N1.EQ.90.AND.N2.EQ.4)GO TO 420	051560
	GO TO 440	051570
400	NLAST=1	051580
	IF(N2.EQ.90.AND.N1.EQ.4)GO TO 420	051590
	GO TO 440	051600
410	NCHK=IABS(N2-N1)	051610
	IF(NCHK.NE.1)GO TO 440	051620
	NLAST=2	051630
	IF(N2.LT.N1)NLAST=1	051640
420	IF(CCI(NLAST).EQ.ACI)GO TO 470	051650
	PRINT 430,CCI(NLAST)	051660
430	FORMAT(" FORCEFILE SHOWS YOU ARE CURRENTLY GAMING CI ",A10/	051670
	* " CI MNEMONIC HAS BEEN SET TO THIS.")	051680
	ACI=CCI(NLAST)	051690
	GO TO 470	051700
440	PRINT 450,CCI	051710
450	FORMAT(" CI NAMES FOUND ON FORCEFILE ARE ",A10," AND ",A10/	051720
	* " THESE ARE NOT SUBSEQUENT CI'S ACCORDING TO HISTORYFILE.")	051730
	GO TO 270	051740
460	AM(N0)=ACI	051750
	CALL DEPLC(IHIST,AM,900,AM(1))	051760
	IF(N0.NE.4.OR.AM(2).NE.1.)GO TO 470	051770
	CALL NEWHIS	051780

ACTD=ACT
470 CALL CLOSEN(ZHIST)
488 RETURN
END

151730
151830
151810
151920

SUBROUTINE NEWHIS	051460
COMMON IA, ID, IP, IENGAG, ITERN, IVIS, INCUNT, MINES, CFPR, FSPFR, FPR,	051460
1 ATIME, IFIRST, IBUN, AGRD, FMASS, IMAXV,	051460
2 SF(2), FSSF(2), PACK(2),	051470
3 ELMT(80,2), ALOSS(65,66), SHOTS(55,2), CKILL(53,2), SHOTSV(55,2)	051480
COMMON/REED/JOAY1, XINX(4), ICARD(20), IARMOR	051490
COMMON/ONE/LFIT(35), ARRAY(90), YMU(1024), C(80,2), ACZ, ASCENE, ISECT	051900
COMMON/HOPE/INIST(35), AM(90), IYBUF(1024)	051910
DO 100 J=1,2	051920
DO 100 I=1,80	051930
ELMT(I,J)=0.	051940
100 CONTINUE	051950
PRINT 260,ACI	051960
110 PRINT *, "DO YOU WANT TO ASSIGN UNITS TO SECTORS? "	051970
CALL REECA(IYN)	051980
IF(IYN.EQ."Y")GO TO 120	051990
IF(IYN.EQ."N")GO TO 130	052000
PRINT 280	052010
GO TO 110	052020
120 PRINT *, "INSTRUCTIONS FOR ASSIGNING SECTORS? "	052030
CALL REECA(INX)	052040
IF(INX.EQ."N")GO TO 170	052050
PRINT *, "EACH PARENT UNIT NAME WILL BE DISPLAYED ON THE SCREEN AND	052060
" YOU MUST MAKE AN INPUT."	052070
PRINT *, "ENTER THE LETTER A IF ALL UNITS IN THE PARENT ARE IN THE	052080
" SAME SECTOR."	052090
PRINT *, "YOU WILL THEN ENTER THE SECTOR # FOR THAT PARENT."	052100
PRINT *, "ANY ENTRY OTHER THAN THE LETTER A INDICATES THAT EACH UN	052110
" T IN THAT PARENT WILL BE ASSIGNED A SECTOR #."	052120
PRINT *, "EACH UNIT NAME WILL THEN BE DISPLAYED AND YOU MUST ENTER	052130
" ITS SECTOR #."	052140
PRINT *, "UNITS INITIALLY UNCOMMITTED SHOULD BE ASSIGNED TO SECTOR	052150
" 0 (ZERO)."	052160
130 AKEEP=0.	052170
IACT=0.	052180
CALL OPENH(LFIT,3LI=0,1LR)	052190
140 CALL GETH(LFIT,ARRAY,ARRAY(1))	052200
M=I*F*CH(LFIT,2LFF)	052210
IF(M.EQ.1000)GO TO 210	052220
IF(IYN.EQ."N")GO TO 160	052230
IF(ARRAY(1).EQ.AKEEP.AND.IACT.EQ."A")GO TO 180	052240
IF(ARRAY(1).EQ.AKEEP)GO TO 170	052250
AKEEP=ARRAY(1)	052260
PRINT 290,ARRAY(1)	052270
CALL REECA(IACT)	052280
IF(IACT.NE."A")GO TO 170	052290
150 PRINT *, "SECTOR # - "	052300
160 CALL REECA	052310

SECT=XTMX(1)	152320
IF(SECT.GE.0.)GO TO 190	152330
PRINT 300	152340
IF(FACT.EQ."ALL")GO TO 157	152350
IF(FACT.EQ."A")GO TO 150	152360
170 PRINT 310,ARRAY(2)	152370
GO TO 180	152380
180 ARRAY(4)=SECT	152390
190 ARRAY(5)=ACT	152400
CALL REPLC(LFIT,ARRAY,900,ARRAY(1))	152410
NEW=0	152420
AM(1)=ACI	152430
AM(2)=ARRAY(1)	152440
AM(3)=ARRAY(2)	152450
AM(4)=99999.	152460
CALL GET(IMIST,AM,AM(1))	152470
IF(AM(4).NE.99999.)NEW=1	152480
AM(4)=ARRAY(4)	152490
AM(5)=ARRAY(5)	152500
J=ARRAY(3)	152510
DO 200 I=6,90	152520
AM(II)=ARRAY(II)	152530
IF((J.NE.1.AND.J.NE.2).OR.II.LT.11)GO TO 200	152540
ELMT(II-10,J)=ELMT(II-10,J)+ARRAY(II)	152550
200 CONTINUE	152560
IF(NEW.EQ.0)CALL PUT(IMIST,AM,900,AM(1))	152570
IF(NEW.EQ.1)CALL REPLC(IMIST,AM,900,AM(1))	152580
GO TO 140	152590
210 CALL CLOSEM(LFIT)	152600
J=1	152610
AM(1)="BLUE START"	152620
220 AM(2)=ACI	152630
AM(3)=0.	152640
AM(4)=99999.	152650
NEW=0	152660
CALL GET(IMIST,AM,AM(1))	152670
IF(AM(4).NE.99999.)NEW=1	152680
DO 230 I=1,80	152690
IF(II.GT.3.AND.II.LT.11)AM(II)=0.	152700
AM(II+10)=ELMT(II,J)	152710
ELMT(II,J)=1.	152720
230 CONTINUE	152730
IF(NEW.EQ.0)CALL PUT(IMIST,AM,900,AM(1))	152740
IF(NEW.EQ.1)CALL REPLC(IMIST,AM,900,AM(1))	152750
IF(J.EQ.2)GO TO 240	152760
J=2	152770
AM(1)="RED START"	152780
GO TO 220	152790

240 RETURN	352900
250 FORMAT(A1)	352910
260 FORMAT(" INITIAL FORCEFILE--WILL BE ACCED TO HISTORYFILE AS C: ",A1)	352920
270 FORMAT(" ",A1)	352930
280 FORMAT(" INCOPRECT--MUST BE Y OR N")	352940
290 FORMAT(" PARENT- ",A10,5X)	352950
300 FORMAT(" SECTOR * CANNOT BE NEGATIVE--REENTER")	352960
310 FORMAT(" ",A10,5X)	352970
END	352980

SUBROUTINE CLEAR(IRQ)	152910
COMMON/ONE/LFIT(35),ARPA(90),YRUF(1024),D(80,2),ACI,ASCENE,ASECT	152920
COMMON/THREE/THIST(35),AM(90),YRUF(1024)	152930
DIMENSION RECS(15)	152940
DATA RECS/"BLUE END","RED END","CI AM","CI ARMOR","CI CANNON",	152950
"CI AO","CI AMC","CI LOSSES","INPUT 1","INPUT 2",	152960
"INPUT 3","INPUT 4","INPUT 5","BLUE START","RED START"/	152970
CALL OPENH(THIST,3LI-C,ILR)	152980
DO 170 I=1,15	152990
IF(RECS(I).EQ.0.)GO TO 190	153000
NN=30	153010
IF(I.EQ.7.OR.I.EQ.8)NN=20	153020
AM(1)=RECS(I)	153030
AM(2)=ACI	153040
AM(3)=ASECT	153050
AM(6)=99999.	153060
CALL GET(THIST,AM,AM(1),0,NN)	153070
IF(AM(4).EQ.99999.)GO TO 160	153080
IF(I.GT.1)GO TO 110	153090
IF(IRQ.EQ."Y")GO TO 110	153100
PRINT *, "SECTOR ",ASECT," WAS PREVIOUSLY GAMED."	153110
100 PRINT *, "IS THIS SECTOR BEING REGAMED NOW?"	153120
CALL BECCA(IRQ)	153130
IF(IRQ.EQ."Y")GO TO 110	153140
IF(IPG.EQ."N")GO TO 200	153150
PRINT *, "ANSWER MUST BE Y OR N"	153160
GO TO 120	153170
110 IF(NN.EQ.20)GO TO 150	153180
120 DO 130 I=6,9	153190
AM(I)=0.	153200
130 CONTINUE	153210
CALL PEFLC(THIST,AM,900,AM(1))	153220
IF(NN.EQ.30)GO TO 170	153230
140 CALL GETN(THIST,AM,AM(1))	153240
MTPE"CH(THIST,2LFP)	153250
IF(4.EQ.1)GO TO 170	153260
IF(AM(1).NE.RECS(I).OF.AM(2).NE.ACI)GO TO 170	153270
150 IF(AM(4).NE.ASECT)GO TO 140	153280
GO TO 120	153290
160 IF(I.LE.1)GO TO 200	153300
170 CONTINUE	153310
180 PRINT 190,ASECT,ACI	153320
190 FORMAT(" ALL STATS RECORDS FOR SECTOR ",F6.0," OF CI ",A10," HAVE	153330
"BEEN "/" ZEROED OUT ON HISTORYFILE.")	153340
200 CALL CLOSEH(THIST)	153350
IF(IPG.NE."N")IRQ="Y"	153360
RETURN	153370
210 FORMAT(A1)	153380
END	153390

SUBROUTINE APPORT	153411
COMMON IA,IC,IP,IEGAG,IERRN,IVIS,INCUNT,MINES,CPPR,FS=PP,FR,	153422
1ATIME,IFIRST,IRUN,MCRG,FYASS,IMAXKV,	153430
2SF(2),FSSF(2),PACK(2),	153440
3ELMT(80,2),ALGSS(66,66),SHOTS(55,2),CKTLL(53,2),SHOTSV(55,2)	153450
COMMON/REED/JDAY1,XINX(4),ICARD(20),TAP40F	053460
COMMON/DA/FPS(80,2),C3EWS(53,2),A1QS(12),CPOS(6),ITHEC(20,2),	053470
1 PSN(6,2,2),PLT(15),KEY(171)	053480
COMMON/ONE/LEIT(35),AORAY(90),VY8UF(1024),G(92,2),AGI,	053490
1ASCENE,ASECT	053500
COMMON/THREE/TWIST(75),AH(90),VY8UF(1024)	053510
DIMENSION XL(90),CIL(6),IHOLD(40,2),CUMLOS(80,2),CAIP(80,2)	053520
*,ICHECK(20),IHEAD(5),IFC(6),IPRNT(5),STATS(24,10),IEFF(5),PAK(2)	053530
DIMENSION IPRAY(2)	053540
EQUVALENCE(ARRAY,IPRAY)	053550
DATA(CIL(2),I=1,6)/1000.,5.,2.,1.33,1.,0./	053560
DATA(ICHECK(I),I=1,20)/16,19,21,26,27,29,32,46,47,50,51,53,54,	053570
51,52,640/	053580
DATA PAK/100.,100000000./	053590
DATA STATS/24000./	053600
DATA IHEAD/"(1H+," " "X,2X,2HUN," "3HIT=A10," "4X)"/	053610
DATA IFOR/"(1H+," " "X,2X,2HEL," "3X,4HLOST," "2X,4HREPA,"	053620
2,"2MIN,2X)"/	053630
DATA IPRNT/"(1H+," " "X,2X,12," "1X,F6.1," "2X,F6.1)"	053640
C/	053650
DATA IEFF/"(1H+," " "X,2X,2HEF," "2HEF=F4.3," "11X)"/	053660
C	053670
DO 120 I=1,IMAXKV	053680
ARRAY(I)=0	053690
XL(I)=0.	053700
DO 100 J=1,65	053710
100 XL(I)=XL(I)+ALOSS(J,I)	053720
DO 110 J=1,2	053730
IHOLD(I,J)=0	053740
CAIP(I,J)=0.	053750
110 O(I,J)=0.	053760
120 CONTINUE	053770
C	053780
130 PRINT *,"DO YOU WISH TO CHANGE COMBAT INTENSITY LEVEL FOR ANY UNIT"	053790
?"	053800
CALL REEDA(INX)	053810
140 FORMAT(1A1)	053820
IF(INX.EQ."Y")GO TO 150	053830
IF(INX.EQ."N")GO TO 270	053840
PRINT 1180	053850
GO TO 130	053860
C	053870
150 PRINT *,"ENTER PARENT ID -"	053880

CALL RECD8(ARRAY(1))	053890
160 FORMAT(1A10)	053900
170 F0PHAT(" ",1A10)	053910
C PRINT *, "ENTER UNIT ID -"	053920
CALL RECD8(UNIT)	053930
C	053940
PARENT=ARRAY(1)	053950
ARRAY(7)=99999	053960
CALL OPENH(LFIT,3LI-0,1LR)	053970
CALL GE(LFIT,ARRAY,ARRAY(1),0,10)	053980
IF(ARRAY(7).NE.99999)GO TO 190	053990
PRINT 1A, PARENT	054000
180 FORMAT(" UNIT ",A10," IS NOT ON FORCEFILE!")	054010
GO TO 250	054020
C	054030
190 IFLAG=0	054040
200 IF(UNIT.EQ."ALL")GO TO 220	054050
IF(ARRAY(2).EQ.UNIT)GO TO 220	054060
210 CALL GETNLFIT,ARRAY,ARRAY(1)	054070
N=IFETCH(LFIT,2LFP)	054080
IF(N.EQ.1009)GO TO 250	054090
IF(ARRAY(1).NE.PARENT)GO TO 250	054100
IF(UNIT.EQ."ALL")GO TO 240	054110
GO TO 280	054120
C	054130
220 PRINT *, "ENTER NEW COMBAT INTENSITY -"	054140
CALL RECD4	054150
ICIL=XINX(1)	054160
LHOLD=ICIL	054170
IF(ICIL.EQ.999)GO TO 230	054180
IF((ICIL.GE.0.AND.ICIL.LE.9).OR.(ICIL.GE.10.AND.ICIL.LE.19))GO TO	054190
Z 240	054200
PRINT *, "INVALID CBT INTENSITY LEVEL ENTERED."	054210
230 PRINT *, "COMBAT INTENSITY LEVELS"	054220
PRINT *, "ENTER 0 FOR UNCOMMITTED UNITS"	054230
PRINT *, " 1 FOR UNITS OUTSIDE CF DIRECT FIRE"	054240
PRINT *, " 2 FOR RESERVE UNITS COMMITTED LIFE"	054250
PRINT *, " 3 FOR UNITS ON PERIMETER CF M94"	054260
PRINT *, " 4 FOR UNITS IN MAIN BATTLE AREA"	054270
PRINT *, " 5 FOR UNITS HIT BY TACAIR"	054280
GO TO 130	054290
C	054300
240 IFLAG=1	054310
ICIL=LHOLD	054320
IF(ICIL.GE.10)ARRAY(9)=ICIL-ICIL/10*10	054330
ICIL=ICIL-ICIL/10*10	054340
ARRAY(7)=ICIL	054350

CALL REPLIC(LFIT,ARRAY,903,ARRAY(1))	054370
IF(UNIT.EQ."ALL")GO TO 210	054380
250 IF(IFLAG.EQ.3)PRINT 140,UNIT	054390
260 CALL CLOSEN(LFIT)	054400
GO TO 130	054410
C	054420
270 CALL OPENH(LFIT,3(I-0.1LR)	054430
280 CALL GETN(LFIT,ARRAY,ARRAY(1))	054440
M=IFETCH(LFIT,2LFR)	054450
IF(M.EQ.1000)GO TO 310	054460
IF(1SECT.NE.ARRAY(4).CP.ACT.NE.ARRAY(5))GO TO 290	054470
ICIL=ARRAY(7)	054480
J=ARRAY(3)	054490
DO 300 I=1,80	054500
IF(ICIL.NE.5)GO TO 290	054510
DATA(I,J)=DATA(I,J)+ARRAY(I+10)	054520
290 CBTINT=CTIL(ICIL+1)	054530
IF(CBTINT.EQ.0.)CBTINT=1.	054540
D(I,J)=0(I,J)+ARRAY(I+10)/CBTINT	054550
300 CONTINUE	054560
GO TO 240	054570
C	054580
310 CALL CLOSEN(LFIT)	054590
DO 320 I=1,IMAXV	054600
D(I,1)=2(I,1)-IFIY(ALOSS(66,I)/PACK(1))/10.	054610
IF(D(I,1).LT.0.10(I,1)=0.	054620
D(I,2)=0(I,2)-(ALOSS(66,I)-IFIY(ALOSS(66,I)/PACK(1))*PACK(1))/10.	054630
IF(D(I,2).LT.0.10(I,2)=3.	054640
320 CONTINUE	054650
C	054660
IFLAG=0	054670
DO 390 J=1,2	054680
DO 390 I=1,IMAXV	054690
IF(J.EQ.2)GO TO 340	054700
KIND="BLUE"	054710
XN=IFIY(XL(I)/PACK(1))/10.	054720
IF(D(I,J).GT.0.0R.XN.LE.C.)GO TO 350	054730
XL(I)=XL(I)-XN*10.*PACK(1)	054740
XN=0.	054750
PRINT 330,I,KIND	054760
330 FORMAT(" APPORTIONMENT OF ITEM ",I2," LOSSES TO ",I4," FORCE CANNON	054770
IT BE MADE")	054780
GO TO 350	054790
340 XN=(XL(I)-IFIY(XL(I)/PACK(1))*PACK(1))/10.	054800
KIND="RED"	054810
IF(D(I,J).GT.0.0R.XN.LE.C.)GO TO 350	054820
XL(I)=XL(I)-XN*10.	054830
XN=0.	054840

PRINT 330,I,KIND	154850
350 IF(XN.LE.D(I,J))GO TO 390	154860
IF(I.GT.3.AND.I.LT.16)GO TO 390	154870
D(I,J)=1.	154880
DO 370 II=1,65	154890
IF(J.EQ.2)GO TO 360	154900
D(I,J)=Q(I,J)+FIX(ALOSS(II,I)/PACK(1))/10.	154910
GO TO 370	154920
360 D(I,J)=Q(I,J)+(ALOSS(II,I)-FIX(ALOSS(II,I)/PACK(1))*PACK(1))/1.	154930
370 CONTINUE	154940
D(I,J)=Q(I,J)+ELMT(I,J)	154950
IFLAG=1	154960
PRINT 390,I,KIND	154970
390 FORMAT(" INSUFFICIENT CST INTENSITY LEVELS HAVE BEEN ASSIGNED FOR	154980
1 ITEM ",I2," OF ",A4," FORCE")	154990
IMOLD(I,J)=1	155000
390 CONTINUE	155010
IF(IFLAG.EQ.0)GO TO 430	155020
PRINT *, "AUTO-ALLOCATION OF ABOVE WPN SYSTEMS HAS BEEN INITIATED"	155030
C	155040
400 IF(IPUN.EQ.1) GO TO 1142	155050
JFLAG=0	155060
WRITE(6,410) ACI,ASECT	155070
410 FORMAT("1 UNIT STATUS FILE FOR CI ",A10," : SECTOR ",F6.3,")	155080
WRITE(6,600)	155090
WRITE(6,420)	155100
420 FORMAT(21X,2H--,5(21X,1=I))	155110
PARENT="ALL"	155120
CALL OPENM(LFIT,3LI=0,1LP)	155130
CALL OPENM(IHIST,3LI=0,1LP)	155140
430 CALL GETN(LFIT,ARRAY,ARRAY(1))	155150
MFLAG=0	155160
M=IFETCH(LFIT,2LFF)	155170
IF(M.EQ.1000) GO TO 450	155180
ICIL=ARRAY(7)	155190
IF(ARRAY(4).NE.ASECT)*FLAG=1	155200
IF(ARRAY(5).NE.ACI)*FLAG=1	155210
440 IF(PARENT.EQ."ALL") GO TO 610	155220
IF(ARRAY(1).EQ.PARENT) GO TO 630	155230
450 IF(MUNIT.EQ.0)GO TO 540	155240
IF(PARINIT.GT.0.) PAPEFF=PARFPS,PARINIT*100.	155250
PRINT 460,PARENT,PAPEFF	155260
460 FORMAT(" CUMULATIVE EFFECTIVENESS OF ",A10,"=",F4.0)	155270
WRITE(6,420)	155280
WRITE(6,420)	155290
WRITE(6,470)IP1	155300
470 FORMAT(" ",2X,"PARENT=",A10,1X,"")	155310
L1=1	155320

480 IF(JFLAG.EQ.1) L1=6	055330
L2=L1+4	055340
NPRNTS=0	055350
DO 490 IUN=L1,L2	055360
IF(ISTATS(1,IUN).EQ.0) GO TO 490	055370
INHEAD(2)=NUMB(IUN*22)	055380
IF(IUN.GT.5) INHEAD(2)=NUMB((IUN-5)*22)	055390
IF(JFLAG.EQ.1.AND.NPRNTS.EQ.0) WRITE(6,500)	055400
IF(JFLAG.EQ.1.AND.NPRNTS.EQ.0) WRITE(6,420)	055410
IF(JFLAG.EQ.1.AND.NPRNTS.EQ.0) WRITE(6,420)	055420
WRITE(6,INHEAD)STATS(1,IUN)	055430
NPRNTS=NPRNTS+1	055440
490 CONTINUE	055450
IF(NPRNTS.EQ.0) GO TO 570	055460
500 FORMAT(21X,11Z(' '))	055470
WRITE(6,420)	055480
IEFF(2)=NUMB(0)	055490
IF(JFLAG.NE.1) WRITE(6,IEFF) PAREFF	055500
DO 510 IUN=L1,L2	055510
IF(ISTATS(1,IUN).EQ.0) GO TO 510	055520
IEFF(2)=NUMB(IUN*22)	055530
IF(IUN.GT.5) IEFF(2)=NUMB((IUN-5)*22)	055540
WRITE(6,IEFF)STATS(2,IUN)	055550
510 CONTINUE	055560
WRITE(6,420)	055570
WRITE(6,420)	055580
IFOR(2)=NUMB(0)	055590
IF(JFLAG.NE.1) WRITE(6,IFOR)	055600
DO 520 IUN=L1,L2	055610
IF(ISTATS(1,IUN).EQ.0) GO TO 520	055620
IFOR(2)=NUMB(IUN*22)	055630
IF(IUN.GT.5) IFOR(2)=NUMB((IUN-5)*22)	055640
WRITE(6,IFOR)	055650
520 CONTINUE	055660
WRITE(6,420)	055670
WRITE(6,420)	055680
KNT=1	055690
DO 560 I=1,40	055700
IF(JFLAG.EQ.1.AND.IBLNK.EQ.5) GO TO 560	055710
IF(JFLAG.EQ.1) GO TO 530	055720
IF(CUMLOS(I,1).EQ.0.AND.CUMLOS(I,2).EQ.0) GO TO 560	055730
IPRNT(2)=NUMB(0)	055740
WRITE(6,IPRNT)I,CUMLOS(I,1),CUMLOS(I,2)	055750
530 IBLNK=0	055760
IF(KNT.GT.22) GO TO 560	055770
DO 550 IUN=L1,L2	055780
IF(ISTATS(2+KNT,IUN).EQ.0) GO TO 540	055790
IPRNT(2)=NUMB(IUN*22)	055800

IF(IUN.GT.5)IPRNT(2)=NU49((IUN-5)*22)	055910
IWPN=STATS(2*KNT,IUN)-IFIX(STATS(2*KNT,IUN)/PAK(1))*PAK(1)	055920
RLOST=IFIX(STATS(2*KNT,IUN)-IFIX(STATS(2*KNT,IUN)/PAK(2))*PAK(2)	055930
C =IWPN/1000.	055940
REMAIN=IFIX(STATS(2*KNT,IUN)/PAK(2))/10.	055950
WRITE(6,IPRNT)IWPN,RLOST,REMAIN	055960
540 IF(STATS(2*KNT,IUN).EQ.0.)IBLNK=IBLNK+1	055970
550 CONTINUE	055980
WRITE(6,420)	055990
IF(IBLNK.NE.5)KNT=KNT+1	055995
IF(JFLAG.EQ.1.AND.IBLNK.NE.5)GO TO 530	055999
560 CONTINUE	055920
IBLNK=1	055930
570 JFLAG=JFLAG+1	055940
IF(JFLAG.EQ.1)GO TO 490	055950
WRITE(6,600)	055960
580 DO 590 IUN=1,10	055970
DO 590 IWPN=1,24	055980
STATS(IWPN,IUN)=0.	055990
590 CONTINUE	056000
JFLAG=0	056010
600 FOPMAT(" ",132(" "))	056020
IF(N.EQ.1008)GO TO 999	056030
610 IF(ARRAY(1).NE.PARENT)NUNIT=0	056040
PARENT=ARRAY(1)	056050
IR1=IRRAY(1)	056060
CALL TRANS(IR1,IP1)	056070
PARFF=0.	056080
PARFOS=0.	056090
PARINIT=0.	056100
IMORT=0	056110
IWD=75	056120
DO 620 I=1,90	056130
DO 620 J=1,2	056140
620 CUMLOS(I,J)=0.	056150
630 TFPS=0.	056160
UEFF=0.	056170
JJ=ARRAY(3)	056180
IR2=IRRAY(2)	056190
CALL TRANS(IR2,IR2)	056200
IF(MFLAG.EQ.0)NUNIT=NUNIT+1	056210
IF(MFLAG.EQ.0)STATS(1,NUNIT)=IR2	056220
640 FOPMAT("PARENT=",A10,JX,"UNIT=",A10//29X	056230
1 "EL MTS",3X,"LOST",2X,"REPAIR")	056240
PERS=0.	056250
IF(ARRAY(3).NE.1)GO TO 860	056260
NWPN=3	056270
DO 950 I=1,65	056280

AIRKO=0.	056290
IF(MFLAG.EQ.1)CUMLOS(I,2)=CUMLOS(I,2)+ARRAY(I+10)	056300
IF(MFLAG.EQ.1)GO TO 650	056310
IF(ARRAY(I+10).EQ.0.)GO TO 350	056320
MOLO=ARRAY(I+10)	056330
IF(ICIL(ICIL+1).NE.0.)GO TO 650	056340
IF(DAIR(I,JJ).LE.0.)GO TO 650	056350
XN=FIX(ALOSS(66,I)/PACK(1))/10.	056360
IF(XN.GT.DAIR(I,JJ))XN=DAIR(I,JJ)	056370
AIPKO=ARRAY(I+10)*XN/DAIP(I,JJ)	056380
AIRKO=FIX(AIRKO*10+.5)/10.	056390
ARRAY(I+10)=ARRAY(I+10)-AIRKO	056400
IF(I.NE.2)GO TO 650	056410
PERS=PERS+AIRKO	056420
650 IF(I.EQ.2)GO TO 650	056430
IF(D(I,JJ).LE.0.)GO TO 650	056440
XN=FIX(X(I)/PACK(1))/10.	056450
IF(XN.GT.D(I,JJ))XN=D(I,JJ)	056460
CBTINT=CIL(ICIL+1)	056470
IF(CBTINT.EQ.0.)CBTINT=1.	056480
IF(MOLO(I,JJ).EQ.1)CBTINT=1.	056490
AKO=(ARRAY(I+10)*XN)/(CBTINT-C(I,JJ))	056500
AKO=FIX(AKO*10+.5)/10.	056510
ARRAY(I+10)=ARRAY(I+10)-AKO	056520
CUMLOS(I,2)=CUMLOS(I,2)+ARRAY(I+10)	056530
AKO=AKO+AIRKO	056540
STATS(NMNP,NUNIT)=I+FIX(AKO*10+.501)*PAK(1)+FIX(ARRAY(I+10)*10	056550
C+.501)*PAK(2)	056560
NMNP=NMPN+1	056570
CUMLOS(I,1)=CUMLOS(I,1)+AKO	056580
IF(ARRAY(I+10).LT.0)ARRAY(I+10)=0	056590
IF(I.LT.13)GO TO 650	056600
CLOST=(AKO-AIRKO)*CPEMS(I-12,JJ)	056610
ARRAY(12)=ARRAY(12)-CLOST	056620
PERS=PERS+CLOST	056630
GO 660 INX=1,20	056640
IF(I.EQ.ICHECK(INX))GO TO 670	056650
660 CONTINUE	056660
GO TO 650	056670
C	056680
670 AM(2)=ACI	056690
AM(3)=ASECT	056700
AM(4)="ERR09"	056710
IF(I.GE.16.AND.I.LE.30)GO TO 680	056720
IF(I.GE.31.AND.I.LE.42)GO TO 690	056730
IF(I.GE.43.AND.I.LE.58)GO TO 710	056740
IF(I.EQ.61.JR.I.EQ.62)GO TO 770	056750
GO TO 999	056760

C	680	AM(1)="CI ARMOR"	056770
		CAL GET (IHIST,AM,AM(1))	056780
		IF (AM(4).EQ."ERROR")GO TO 910	056790
		IWD=0	056800
		INC=(I-11)*4+11	056810
		FS=(AM(INC)*HOLD)/(CBTINT*D(I,JJ))	056820
		SHRT=AM(INC+3)	056830
		IF (I.GE.16.AND.I.LE.20)SHRT=SHRT+AM(INC+2)	056840
		IF (I.EQ.27)SHRT=AM(INC+2)	056850
		SHRT=(SHRT*HOLD)/(CBTINT*D(I,JJ))	056860
		IF (I.GE.16.AND.I.LE.20)IWD=76	056870
		IF (I.EQ.26.OR.I.EQ.27)IWD=77	056880
		IF (I.EQ.21.OR.I.EQ.28)IWD=78	056890
		IF (IWD.LI.76)GO TO 950	056900
		ARRAY(IWD)=ARRAY(IWD)+FS	056910
		ARRAY(IWD+9)=ARRAY(IWD+9)+SHRT	056920
		GO TO 950	056930
C	690	AM(1)="CI AD"	056940
		CAL GET (IHIST,AM,AM(1))	056950
		IF (AM(4).EQ."ERROR")GO TO 910	056960
		INC=(I-31)*4+11	056970
		NWD=76	056980
		IF (I.EQ.36)NWD=80	056990
		ARRAY(NWD)=(AM(INC)*HOLD)/(CBTINT*D(I,JJ))	057000
		ARRAY(NWD+9)=(AM(INC+2)*HOLD)/(CBTINT*D(I,JJ))	057010
		GO TO 950	057020
C	700	AM(1)="CI CANNON"	057030
		CAL GET (IHIST,AM,AM(1))	057040
		IF (AM(4).EQ."ERROR")GO TO 910	057050
		INC=(I-43)*4+10	057060
		IF (I.GE.43.AND.I.LE.47)GO TO 750	057070
		IF (I.EQ.50.OR.I.EQ.53)GO TO 730	057080
		IF (I.EQ.51)GO TO 720	057090
		DO 710 I=1,4	057100
		ARRAY(45)=ARRAY(45)+(AM(INC+II)*HOLD)/(CBTINT*D(I,JJ))	057110
	710	CONTINUE	057120
		GO TO 950	057130
	720	ARRAY(55)=(AM(INC+1)*HOLD)/(CBTINT*D(I,JJ))	057140
		ARRAY(56)=(AM(INC+3)*HOLD)/(CBTINT*D(I,JJ))	057150
		GO TO 850	057160
	730	IWD=84	057170
		DO 740 I=1,4	057180
		ARRAY(IWD+II)=ARRAY(IWD+II)+(AM(INC+II)*HOLD)/(CBTINT*D(I,JJ))	057190
	740	CONTINUE	057200
		GO TO 950	057210
			057220
			057230
			057240

750	IWD=88	157250
	DO 760 I1=1,4	157260
	ARRAY(ING)=ARRAY(IWD)+(AM(ING*I1)*MCLC)/(CATINT=0(I,JJ))	157270
760	CONTINUE	157280
	GO TO 950	157290
770	AM(1)="CI AM"	157300
	CALL GET(IHIST,AM,AM(1))	157310
	IF(ARRAY(4).EQ."EPROR")GO TO 810	157320
	IAM=13	157330
	IF(I.EQ.62)IAM=14	157340
	IWD=76	157350
	DO 790 INC=11,87,4	157360
	IF(AM(ING).NE.ARRAY(1))GO TO 730	157370
	AM(ING)=IAM	157380
	DO 780 I1=1,3	157390
	INX=INC*I1	157400
	ARRAY(IWD)=AM(INX)	157410
	IWD=IWD+1	157420
780	CONTINUE	157430
	IF(IWD.GE.90)GO TO 800	157440
790	CONTINUE	157450
800	CALL REPLC(IHIST,AM,900,AM(1))	157460
	GO TO 950	157470
C		157480
810	PRINT 820,AM(1)	157490
820	FORMAT(" ",A10," RECORD NOT FOUND")	157500
	GO TO 950	157510
C		157520
830	PRINT 840,I	157530
840	FORMAT(" AMMO RECORD SIZE FOR WEAPON TYPE ",I2," IS INSUFFICIENT")	157540
C		157550
950	CONTINUE	157560
	GO TO 890	157570
860	CONTINUE	157580
	JJ=2	157590
	NWPN=3	157600
	DO 880 I=1,65	157610
	AIRKO=0.	157620
	IF(MFLAG.EQ.1)CUMLOS(I,2)=CUMLOS(I,2)+ARRAY(I+10)	157630
	IF(MFLAG.EQ.1)GO TO 870	157640
	IF(ARRAY(I+10).EQ.0.)GO TO 880	157650
	IF(DAIR(I,JJ).LE.0.)GO TO 870	157660
	IF(CIL(ICIL+1).NE.0.)GO TO 870	157670
	XN=(ALOSS(66,I)-FIX(ALOSS(66,I)/PACK(1))*PACK(1))/10.	157680
	IF(XN.GT.DAIR(I,JJ))XN=DAIR(I,JJ)	157690
	AIRKO=ARRAY(I+10)*XN/DAIR(I,JJ)	157700
	AIRKO=FIX(AIRKO*10+.5)/10.	157710
	ARRAY(I+10)=ARRAY(I+10)-AIRKO	157720

IF (X.NE.2) GO TO 870	057730
PERS=PERS+AIRKO	057740
870 IF (I.EQ.2) GO TO 880	057750
IF (I(I,JJ).LE.0.) GO TO 880	057760
XN=(XL(I)-IFIX(XL(I)/PACK(1))*PACK(1))/10.	057770
IF (XN.GT.0(I,JJ)) XN=0(I,JJ)	057780
CBTINT=CIL(ICIL+1)	057790
IF (CBTINT.EQ.0.) CBTINT=1.	057800
IF (IHOLD(I,JJ).EQ.1) CBTINT=1.	057810
AKO=(ARRAY(I+10)*XN)/(CBTINT*C(I,JJ))	057820
AKO=IFIX(AKO*10+.5)/10.	057830
ARRAY(I+10)=ARRAY(I+10)-AKO	057840
CUMLOS(I,2)=CUMLOS(I,2)+ARRAY(I+10)	057850
AKO=AKO+4IRKO	057860
STATS(NWPN,NUNIT)=I+IFIX(AKO*10+.001)*PAK(1)+IFIX(ARRAY(I+10)*10+	057870
C+.001)*PAK(2)	057880
NWPN=NWPN+1	057890
CUMLOS(I,1)=CUMLOS(I,1)+AKO	057900
IF (ARRAY(I+10).LT.0) ARRAY(I+10)=0	057910
IF (I.LT.13) GO TO 880	057920
CLOST=(AKO-AIRKO)*CREWS(I-12,JJ)	057930
ARRAY(12)=ARRAY(12)-CLOST	057940
PERS=PERS+CLOST	057950
880 CONTINUE	057960
890 IF (NFLAG.EQ.1) GO TO 910	057970
IF (ARRAY(12).LT.0) ARRAY(12)=0	057980
CUMLOS(2,2)=CUMLOS(2,2)+ARRAY(12)	057990
I=2	058000
STATS(NWPN,NUNIT)=I+IFIX(PERS*10+.001)*PAK(1)+IFIX(ARRAY(12)*10+	058010
C+.001)*PAK(2)	058020
CUMLOS(2,1)=CUMLOS(2,1)+PERS	058030
900 FORMAT(2X,I3,4X,F5.1,2X,F5.1)	058040
910 DO 920 I=1,65	058050
920 TFPS=TFPS+ARRAY(I+10)*FPS(I,JJ)	058060
IF (NFLAG.EQ.1) GO TO 940	058070
IF (ARRAY(6).GT.0.) UEFF=TFPS/ARRAY(6)*100.	058080
STATS(2,NUNIT)=UEFF	058090
PRINT 930,ARRAY(2),UEFF	058100
930 FORMAT(' EFFECTIVENESS OF ',A10,'=',F4.0)	058110
ARRAY(4)=UEFF	058120
940 PARFPS=PARFPS+TFPS	058130
PARINIT=PARINIT+ARRAY(6)	058140
ARRAY(7)=ARRAY(9)	058150
IF (NFLAG.EQ.1) GO TO 430	058160
950 NEW=0	058170
AW(1)=ACI	058180
AW(2)=ARRAY(1)	058190
AW(3)=ARRAY(2)	058200

AM(4)=99999.	058210
CALL GET(IHIST,AM,AM(1))	058220
IF(AM(4).NE.99999.)INE=1	058230
AM(4)=ARRAY(4)	058240
AM(5)=ARRAY(5)	058250
DO 960 I=6,90	058260
AM(I)=ARRAY(I)	058270
960 CONTINUE	058280
IF(NEW.EQ.0)CALL PUT(IHIST,AM,900,AM(1))	058290
IF(NEW.EQ.1)CALL REPLC(IHIST,AM,900,AM(1))	058300
DO 970 I=76,90	058310
970 ARRAY(I)=0.	058320
CALL REPLC(LFIT,ARRAY,900,ARRAY(1))	058330
GO TO 439	058340
980 CALL CLOSEN(LFIT)	058350
DO 990 II=1,90	058360
AM(II)=0.	058370
990 CONTINUE	058380
AAH1="CI LOSSES"	058390
1000 AM(1)=AAH1	058400
AM(2)=ACI	058410
AM(4)=99999.	058420
CALL GET(IHIST,AM,AM(1),0.20)	058430
IF(AM(4).EQ.99999.)GO TO 1030	058440
1010 IF(AM(4).NE.ASECT)GO TO 1020	058450
CALL CLTE(IHIST)	058460
1020 CALL GETN(IHIST,AM,AM(1))	058470
M=IFETCH(IHIST,2LFP)	058480
IF(M.EQ.1(09))GO TO 1030	058490
IF(AM(1).EQ.AAH1.AND.AM(2).EQ.ACI)GO TO 1010	058500
1030 IF(AAH1.EQ."CI AMMO")GO TO 1040	058510
AAH1="CI AMMO"	058520
GO TO 1000	058530
1040 AM(1)="CI LOSSES"	058540
AM(2)=ACI	058550
AM(3)=ASECT*100.	058560
AM(5)=ASECT	058570
DO 1060 I=1,IMAXKV	058580
AM(4)=I	058590
IPUT=0	058600
DO 1050 J=1,IMAXKV	058610
AM(J+10)=ALOSS(I,J)	058620
IF(AM(J+10).NE.0.)IPUT=1	058630
1050 CONTINUE	058640
IF(IPUT.NE.0)CALL PUT(IHIST,AM,900,AM(1))	058650
AM(3)=AM(3)+IPUT	058660
1060 CONTINUE	058670
DO 1070 K=1,90	058680

AM(K)=0.	058690
1070 CONTINUE	058700
1080 AM(1)="CI AMMO"	058710
AM(2)=ACI	058720
AM(3)=ASECT*100.	058730
AM(5)=ASECT	058740
J=1	058750
1090 AM(4)=J	058760
DO 1100 I=1,29	058770
AM(I+13)=SHOTS(I,J)	058780
1100 CONTINUE	058790
CALL PUT(IHIST,AM,988,AM(1))	058800
IF(J.EQ.2)GO TO 1110	058810
J=2	058820
AM(3)=AM(3)+1.	058830
GO TO 1090	058840
1110 CALL CLOSE4(IHIST)	058850
AM1="BLUE END"	058860
J=1	058870
1120 INPUT=-1	058880
CALL CKSTOR(AM1,INPUT)	058890
DO 1130 I=1,I*AXKV	058900
AM(I+10)=EL4T(I,J)	058910
1130 CONTINUE	058920
CALL CKSTOR(AM1,INPUT)	058930
IF(J.EQ.2)GO TO 1140	058940
J=2	058950
AM1="RED END"	058960
GO TO 1120	058970
1140 PRINT "DISPLAY A UNIT? "	058980
1150 CALL REEWA(INT)	058990
1160 FORMAT(A1)	059000
1170 FORMAT(" ",1A1)	059010
IF(INT.EQ.1HW) GO TO 1200	059020
IF(INT.EQ.1HY) GO TO 1190	059030
PRINT 1180	059040
1180 FORMAT(" INCORRECT! RESPONSE MUST BE YES OR NO -TRY AGAIN")	059050
GO TO 1150	059060
C	059070
1190 CALL DISPLAY	059080
GO TO 1140	059090
1200 CONTINUE	059100
END	059110

SUBROUTINE BUILD	059130
COMMON IA,IO,IP,IENGAG,ITERM,IVIS,IMOUNT,MINES,CFFR,FSFRR,FBJ,	059140
LATIME,IFIRST,IQUN,NQPD,FMASS,IMAXV,	059150
ZSF(2),FSSF(2),PACK(2),	059160
IELMT(8),ALOSS(66,66),SMOTS(55,2),CKILL(53,2),SMOTSV(55,2)	059170
COMMON/REED/JDAY1,XINX(4),ICARD(23),IARMOF	059180
COMMON/DATA/FPS(80,2),CREWS(53,2),AFOS(12),DPJS(6),ITIME(21,2),	059190
1 PSN(6,2,2),PLT(15),KEY(171)	059200
COMMON/ONE/LFIT(35),ARRAY(90),MYGUF(1324),C(80,2),ACT,	059210
.ASCENE,ASECT,YSEC	059220
COMMON/TWO/LFIT(35),BARRAY(66),MYGUF(1024)	059230
100 BARRAY(1)="SRC"	059240
AMOLD=BARRAY(1)	059250
CALL OPENM(IFIT,3LI-0.1LR)	059260
DO 110 I=2,65	059270
BARRAY(I) = 0	059280
110 CONTINUE	059290
C	059300
C ABOVE DC LOOPS ZERO OUT WORK ARRAYS	059310
C	059320
C	059330
120 PRINT *,"ENTER SRC ACTION TYPE -"	059340
CALL REED4	059350
INX=XINX(1)	059360
IF(INX.EQ.999)GO TO 130	059370
IF(INX.GE.0.AND.INX.LE.4)GO TO 140	059380
PRINT *,"ACTION CODE ERROR - TRY AGAIN!"	059390
130 PRINT *,"VALID ACTION CODES."	059400
PRINT *,"ENTER 0 TO RETURN TO DECISION POINT"	059410
PRINT *," 1 TO ADD A NEW SRC"	059420
PRINT *," 2 TO DELETE A SRC"	059430
PRINT *," 3 TO DISPLAY A SPECIFIC SRC"	059440
PRINT *," 4 TO DISPLAY ALL SRC'S"	059450
GO TO 120	059460
C	059470
140 GO TO (460,270,360,150,400),INX+1	059480
150 PRINT *,"ENTER SRC TO BE DISPLAYED -"	059490
CALL PEECB(ASPC)	059500
BARRAY(2) = ASPC	059510
BARRAY(3) = 90909	059520
CALL GET(IFIT,BARRAY,BARRAY(1))	059530
IF(BARRAY(3).EQ.90909) GO TO 250	059540
PRINT 190 ,BARRAY(2)	059550
160 FORMAT(1X,"SRC=",A10.5X," ID QTY")	059560
DO 200 I=3,45,2	059570
IF (BARRAY(I).EQ. 0) GO TO 200	059580
IID=BARRAY(I)	059590
PRINT 190, (IID,BARRAY(I+1))	059600

190	FORMAT(20X,I3,F5.3)	059610
200	CONTINUE	059620
210	PRINT *, "DISPLAY ANOTHER SRC?"	059630
	CALL REECA(INX)	059640
	IF(INX.EQ."Y") GO TO 190	059650
	IF(INX.EQ."N") GO TO 120	059660
	GO TO 210	059670
250	PRINT 260, ASRC	059680
	BRRAY(3)=0	059690
260	FORMAT(1X,"SRC ",A10," NOT ON FILE")	059700
	GO TO 210	059710
270	PRINT *, "ENTER SRC TO BE ADDED -"	059720
	CALL REEDB(ASRC)	059730
	BRRAY(2)=ASRC	059740
	BRRAY(3) = 90909	059750
	CALL GET (IFIT,BRRAY,BRRAY(1))	059760
	IF(BRRAY(3).NE.90909) GO TO 320	059770
	NN=1	059780
	PRINT 290	059790
280	FORMAT(1X,"ENTER WPN ID,QTY-- (,1 IF CCNE ")	059800
290	CALL REED4	059810
	NN=INX(1)	059820
	AJ=INX(2)	059830
	IF(NN.EQ.0) GO TO 300	059840
	NN=NN+2	059850
	BRRAY(NN)=MM	059860
	BRRAY(NN+1)=AJ	059870
	PRINT *, "NEXT--"	059880
	GO TO 290	059890
300	CONTINUE	059900
	CALL PUT (IFIT,BRRAY,460,BRRAY(1))	059910
	M=IFETCH(IFIT,3LIRS)	059920
	IF (M.EQ.4460) GO TO 320	059930
310	FORMAT(1X,"SRC-",A10," ALREADY ON FILE")	059940
	GO TO 330	059950
320	PRINT 310, ASRC	059960
	BRRAY(3)=0	059970
330	DO 340 I=2,46	059980
	BRRAY(I)=0	059990
340	CONTINUE	060000
350	PRINT *, "ADD ANOTHER SRC?"	060010
	CALL REECA(INX)	060020
	IF(INX.EQ."Y") GO TO 270	060030
	IF(INX.EQ."N") GO TO 120	060040
	GO TO 350	060050
360	PRINT *, "ENTER SRC TO BE DELETED -"	060060
	CALL REED9(ASRC)	060070

ARRAY(2)=ASRC	160190
ARRAY(3)=90909	160193
CALL GET(IFIT,ARRAY,ARRAY(1))	160196
IF(ARRAY(3).EQ.90909) GO TO 380	160199
CALL OLTE(IFIT,ARRAY(1))	160202
GO TO 420	160205
380 PRINT *, "SRC ", ARRAY(2), " NOT ON FILE "	160208
ARRAY(3)=0	160211
390 PRINT *, "DELETE ANOTHER SRC?"	160214
CALL REEDA(INX)	160217
IF(INX.EQ."Y") GO TO 360	160220
IF(INX.EQ."N") GO TO 120	160223
GO TO 390	160226
400 CALL REWIND(IFIT)	160229
410 CONTINUE	160232
CALL GETN(IFIT,ARRAY,ARRAY(1))	160235
M=IFETCH(IFIT,2LEP)	160238
IF(M.EQ.1308) GO TO 420	160241
IF(AHOLD.EQ.ARRAY(1)) GO TO 430	160244
GO TO 410	160247
420 CALL CLOSEN(IFIT)	160250
GO TO 100	160253
430 PRINT 100,ARRAY(2)	160256
DO 440 I=3,45,2	160259
IF(ARRAY(I).EQ.0) GO TO 440	160262
IIO=ARRAY(I)	160265
PRINT 190,IIO,ARRAY(2,1))	160268
440 CONTINUE	160271
DO 450 I=2,46	160274
ARRAY(I)=0	160277
450 CONTINUE	160280
GO TO 410	160283
460 CONTINUE	160286
CALL CLOSEN(IFIT)	160289
END	160292

SUBROUTINE AMAD	360450
COMMON IA, ID, IP, IENGAG, ITERN, IVIS, IINCUNT, MINES, CFP, FSP, PPR,	360460
1 ATIME, IFTST, IRUL, NQD, FMASS, IMAXIV,	360470
2 SF(2), FSSF(2), PACK(2),	360480
3 FLNT(40,2), LOSS(66,66), SHOTS(55,2), CKILL(53,2), SHOTSV(55,2)	360490
COMMON/REED/JDAY1, XINX(4), ICARD(2), IAPMOR	360500
COMMON/DATA/APS(80,2), CREWS(53,2), APOS(12), DPOS(6), ITHEN(20,2),	360510
1 PSN(6,2,2), PLT(15), KEY(171)	360520
COMMON/ONE/LEIT(35), APRAY(90), MYBUE(1024), C(40,2), ACT,	360530
1 ASCENE, ASECT	360540
COMMON/THREE/THYST(35), IH(90), I19UF(1124)	360550
COMMON/EW/PCOGR(2,4), ICJUNT(4), ISMOKE(2), PCSMO(2), PCSSH(2), APD(2)	360560
1 NMU(2)	360570
DIMENSION ORD(7,2,8), PK(224), SSK(12,2,10), V(5), KING(2), WEAPC(3),	360580
1 SFACT(12,2), PKILL(12), PKILL(7), T*ASK(4), APP(8,2), SA(7), CELL(7),	360590
2 NPOP(7), POPORD(7,2), GNDLOS(8,42,2), AIRLOS(13,7,2), ACKILL(7,2),	360600
3 OPAY(42,2), GPKILL(42), ACLOST(7), ACCREW(2), ACAV(7,2), MAXPOP(7,2)	360610
DIMENSION IRAMO(12), I3A40(12), PAD(3), ACQ(5,7), PCNSM(2), PGAC(7,4)	360620
DIMENSION PF(12), SHTS(12), DUST(4,20,2)	360630
DATA ((ICUST(I,J,K), J=1,20), I=1,4), K=1,2)/	360640
1 4*1...92...35...9...75...71...66...3*1...38...9...67...51...2*...83...	360650
2 3*1...95...44...77...69...64...66...60...	360660
2 2*1...96...92...73...64...50...46...2*0...	360670
2 3*1...91...9...72...63...6...6...33...2*1...92...79...68...6...6...39...0...1...	360680
3 2*1...9...7...6...5*0...2*1...79...6...34...5*0...	360690
1 2*1...93...8...7...62...55...49...44...39...1...98...79...55...55...47...4...26...	360700
1 .05...0...	360710
2 2*1...98...74...61...50...42...23...2*0...	360720
2 1...94...74...58...46...38...4*0...	360730
2 2*1...85...69...56...44...26...3*0...1...03...7...54...41...39...4*1...	360740
3 1...97...67...43...6*0...1...02...52...04...6*0...	360750
DATA ((OPD(I,J,K), K=1,8), I=1,7), J=1,2)/	360760
1 9*0...	360770
2 8*0...	360780
3 16...5*0...1200...0...	360790
4 0...8...3*0...0...600...0...	360800
5 8*0...	360810
6 8*0...	360820
7 8*0...	360830
1 4*0...64...0...0...0...	360840
2 3*0...4...128...2000...0...0...	360850
3 0...0...6...0...4*0...	360860
4 0...4...0...0...128...2000...0...0...	360870
5 16...0...0...0...128...0...0...0...0...	360880
6 0...0...0...4...128...2000...0...0...	360890
7 8*0...	360900
DATA ((SFACT(I,J), I=1,12), J=1,2)/	360910
9 10*2.86, 3.42, 2.86,	360920

2 2*2.86,3*1.52,5*2.86,3.52,2.86/	360930
DATA((OPAV(I,J),I=1,42),J=1,2)/	360940
3 .83,9*1.,	360950
3 .95,.91,.93,.95,.95,	360960
3 .78,.74,.67,.78,.73,	360970
3 .95,.93,.93,.93,.93,.93,.93,.93,.93,	360980
3 .68,.75,5*.83,.68,.44,3*.73,	360990
R .83,9*1.,	361000
R .95,.95,.93,0,.86,	361010
Q 3*.79,.62,.0,	361020
R .95,.95,.93,.93,.93,.95,.93,.95,.95,.93,	361030
R 6*.85,3*.83,3*.85/	361040
DATA((ACAV(I,J),I=1,7),J=1,2)/2*.65,2*.81,.74,.76,.80,3*.76	361050
1,3*.81,.81/	361060
DATA(V(I),I=1,5)/1.,.85,.65,.45,.3/	361070
DATA(KIND(I),I=1,2)/"BLUE","RED"/	361080
DATA(WEAPC(I),I=1,3)/.9,.7,.1/	361090
DATA(TMASK(I),I=1,4)/4*.6/	361100
DATA((APOP(I,J),I=1,8),J=1,2)/1.,1.,3.,0.,0.,3.,0.,0.,	361110
A 1.,1.,1.,1.,12.,0.,0.,0.,0./	361120
DATA((ATFLOS(K,I,J),K=1,13),I=1,7),J=1,2)/182*0./	361130
DATA((IGNCLOS(K,I,J),K=1,8),I=1,42),J=1,2)/672*0./	361140
DATA(ACCREW(J),J=1,2)/2*0./	361150
DATA(TRAPO(I),I=1,12)/44.45,46.44,47.46,48.52,51.10,54/	361160
DATA(TRAPO(I),I=1,12)/44.45,0.1,47.48,49.51,52.53,53.54/	361170
C	361180
POINT(52,*)*****AH4C"	361190
DO 100 IEL=1,2	361200
PCNSH(IEI)=(1.-PCSMO(IEI))*(1.-PCSMO(13-IEI)/2.)*(1.-PCSS*(IEI))	361210
100 CONTINUE	361220
IV=IVIS	361230
IF(IV.EQ.5)IV=4	361240
I*(IV.EQ.6)IV=4	361250
NORO=1	361260
IF(NDAY1.EQ."DAY")NORO=2	361270
CALL OPENMS(3,KEY,171,0)	361280
CALL READMS(3,SSK,240,39)	361290
C DO 9999 I=1,240	361300
C SSK(I)=SSK(I)*.5	361310
C9999 CONTINUE	361320
CALL REACHS(3,MAXPOP,14,179)	361330
CALL CLCSMS(3)	361340
C	361350
DO 110 I=1,90	361360
ARRAY(I)=0	361370
110 CONTINUE	361380
130 FORMAT(" INCORRECT ENTRY--TRY AGAIN")	361390
150 PRINT *, "DO YOU WISH TO PROCESS AIR DEFENSE/ARMED HELICOPTER ASSESS"	361400

ISMENTS?	061410
CALL REECA(INX)	061420
IF(INX.EQ."Y")GO TO 160	061430
IF(INX.EQ."N")GO TO 1290	061440
GO TO 150	061450
160 PRINT "AIR DEFENSE/ARMED HELICOPTER ASSESSMENTS"	061460
JVIS=IVIS	061470
IF(JVIS.EQ.5) JVIS=4	061480
DO 1140 J=1,2	061490
IF(J.EQ.2)GO TO 165	061500
NCL=11	061510
NMAX=50	061520
NW=19	061530
GO TO 167	061540
165 IPUT=-1	061550
AM1="C1 AM"	061560
CALL CKSTOR(AM1,IPUT)	061570
NCL=51	061580
NMAX=90	061590
NW=59	061600
167 L=3-J	061610
SH=1.-FSSF(L)*2.96	061620
IF(SH.LE..2)SH=.2	061630
JPSN=1	061640
IF(J.EQ.10) JPSN=2	061650
DO 170 K=59,65	061660
IF(ELMT(K,L).GT.0.)GO TO 190	061670
170 CONTINUE	061680
GO TO 1120	061690
180 PRINT 190,KIND(J),KIND(L)	061700
190 FORMAT(" DO YOU WISH TO GAME ",A4," ADA AND ",A4," A/C? ")	061710
CALL REECA(INX)	061720
IF(INX.EQ."Y")GO TO 200	061730
IF(INX.EQ."N")GO TO 1140	061740
GO TO 180	061750
C SET AD ENVIRONMENT	061760
200 PRINT 210,KIND(J)	061770
210 FORMAT(// " THE FOLLOWING SETS PARAMETERS FOR ",A4," AD WEAPONS")	061780
220 PRINT 230,KIND(J)	061790
230 FORMAT(" ENTER ",A4," WEAPON CONTROL (STATUS) FACTOR- ")	061800
CALL REECA	061810
KTRL=XINX(1)	061820
ARRAY(INCL+1)=KTRL	061830
IF(KTRL.GE.1.AND.KTRL.LE.3)GO TO 250	061840
IF(KTRL.EQ.999)GO TO 240	061850
PRINT 130	061860
240 PRINT "FOR WEAPON FREE.....ENTER 1"	061870
	061880

PRINT *,	WEAPON TIGHT.....ENTER 2"	361990
PRINT *,	WEAPON HOLD.....ENTER 3"	361900
GO TO 220		361310
250 CONTINUE		361920
PRINT *,	"ENTER STANDOFF RANGE INCHES- "	361030
CALL REED4		361940
IR=XINX(1)		361950
ARRAY(NCL+2)=IR		361960
IF (IR.GE.1.AND.IR.LE.10) GO TO 270		361970
IF (IR.EQ.999) GO TO 260		361980
PRINT 130		361990
260 PRINT *,	"IF RANGE IS 500 ENTER 1"	362000
PRINT *,	1000 2"	362010
PRINT *,	1500 3"	362020
PRINT *,	2000 4"	362030
PRINT *,	2500 5"	362040
PRINT *,	3000 6"	362050
PRINT *,	3500 7"	362060
PRINT *,	4000 8"	362070
PRINT *,	4500 9"	362080
PRINT *,	5000 10"	362090
GO TO 250		362100
270 CONTINUE		362110
CALL OPENMS(3,KEY,171,0)		362120
INC=49+IP+((IVIS-1)*10)+((NORG-1)*40)		362130
CALL READMS(3,ACQ,35,INC)		362140
INC=129+INDEX5(IR,NORG,L,J,0,10,2,0,0)		362150
CALL READMS(3,PDAC,28,INC)		362160
INC=39+IR		362170
CALL READMS(3,PK,224,INC)		362180
C DO 9999 I=1,224		362190
C PK(I)=PK(I)*.5		362200
C9999 CONTINUE		362210
C		362220
CALL CLCSMS(3)		362230
280 PRINT *,	"ENTER PERCENT (DECIMAL) AD COMMITTED BY RANGE FACTORS."	362240
PRINT *,	"SHORT,MEDIUM, LONG-- "	362250
CALL REED4		362260
PAQ(1)=XINX(1)		362270
PAQ(2)=XINX(2)		362280
PAQ(3)=XINX(3)		362290
ARRAY(NCL+3)=PAQ(1)		362300
ARRAY(NCL+4)=PAQ(2)		362310
ARRAY(NCL+5)=PAQ(3)		362320
GO 290 N=1,3		362330
IF (PAQ(N).GE.0..AND.PAQ(N).LE.1.) GO TO 290		362340
PRINT *,	"INCORRECT ENTRY--MUST BE BETWEEN 0. AND 1. "	362350
GO TO 240		362360

290	CONTINUE	162370
300	PRINT 310,KIND(J)	162380
310	FORMAT(" ENTER PRIORITY WEIGHTING FACTOR FOR ".A4," ACA TARGETS (MAX=	162390
	1.0) - ")	162400
	CALL REEQ4	162410
	WEIGHT=XINX(1)	162420
	ARRAY(NCL+6)=WEIGHT	162430
	IF(WEIGHT.GE.1.0.AND.WEIGHT.LE.5.0)GO TO 320	162440
	PRINT 130	162450
	GO TO 300	162460
320	DO 330 I=31,42	162470
	FPS(I,J)=FPS(I,J)*WEIGHT	162480
330	CONTINUE	162490
C	SET FLYING TIME AND # SORTIES PER HOUR	162500
	PRINT 140,KIND(L)	162510
340	FORMAT("/" THE FOLLOWING SETS PARAMETERS FOR ".A4," HELICOPTERS"/)	162520
350	PRINT 360,KIND(L),ATIME	162530
360	FORMAT(" ENTER TOTAL FLYING TIME FOR ".A4," A/C THIS CI (MAX= ".F4	162540
	1.0," HOURS) - ")	162550
	CALL REEQ4	162560
	M=XINX(1)	162570
	ARRAY(NCL+7)=M	162580
	IF(M.LE.ATIME.AND.M.GE.0.0)GO TO 370	162590
	PRINT 130	162600
	GO TO 350	162610
370	PRINT 380,KIND(L)	162620
380	FORMAT(" ENTER SORTIES PER HOUR FOR THE FOLLOWING ".A4," A/C (MAX=	162630
	1.0) - ")	162640
	DO 400 K=59,65	162650
	IF(ELMT(K,L).LE.0.0)GO TO 400	162660
	KK=K-59	162670
390	PRINT *, "TYPE ".K," I "	162680
	CALL REEQ4	162690
	SA(KK)=XINX(1)	162700
	IF(SA(KK).GT.0.0.AND.SA(KK).LE.3.0)GO TO 400	162710
	PRINT 130	162720
	GO TO 390	162730
400	CONTINUE	162740
C	COMPUTE # SORTIES AVAILABLE	162750
	DO 410 K=59,65	162760
	KK=K-59	162770
410	SA(KK)=SA(KK)*M*ELMT(K,L)*ACAV(KK,L)	162780
C	BUILD CELL	162790
	PRINT 420,KIND(L),KIND(J)	162800
420	FORMAT("/" BEGIN BUILDING CELLS OF ".A4," A/C TO FLY AGAINST "	162810
	1A4," GROUND FORCES"/)	162820
	PRINT 430,KIND(L)	162830
430	FORMAT(" TOTAL ".A4," A/C AND SORTIES AVAILABLE THIS CI")	162840

GO TO 460	162950
440 PRINT 450,KIND(L)	162960
450 FORMAT(" ",44," A/C AND SORTIES REMAINING THIS CI")	162970
460 PRINT *, " A/C TYPE * A/C * SORTIES"	162980
DO 480 K=59,65	162990
KK=K-59	162900
CELL(KK)=0.	162910
ACKILL(KK,1)=ACKILL(KK,2)=0	162920
HELI=(FIX(ELMT(K,L)*ACAV(KK,L))-AIFLOS(13,KK,L))	162930
IF(SA(KK).LE.3..0*.HELI.LE.C.1)GO TO 480	162940
PRINT 470,K,HELI,SA(KK)	162950
470 FORMAT(9X,I2,3X,F4.0,9X,F5.0)	162960
480 CONTINUE	162970
PRINT *, "ENTER A/C ELMT #,NO. ADDED(+ OR -) TO CELL--C,0 TO STOP-	162980
2"	162990
490 CALL REEC4	163000
K=XINX(1)	163010
AC=XINX(2)	163020
IF(K.EQ.0)GO TO 590	163030
IF(K.LT.59.OR.K.GT.65)GO TO 510	163040
KK=K-59	163050
IF(CELL(KK)+AC.LT.0.)GO TO 540	163060
IF(CELL(KK)+AC.GT.ELMT(K,L)*ACAV(KK,L)-AIFLOS(13,KK,L))GO TO 530	163070
IF(CELL(KK)+AC.GT.SA(KK))GO TO 520	163080
CELL(KK)=CELL(KK)+AC	163090
500 PRINT *, "NEXT ENTRY- "	163100
GO TO 490	163110
510 PRINT *, "INVALID A/C ELMT #--ENTRY IGNORED"	163120
GO TO 500	163130
520 IF(IRUN.EQ.3)GO TO 550	163140
PRINT *, "A/C ENTERED REQUIRES MORE SORTIES THAN ARE AVAILABLE"	163150
PRINT *, "ENTRY IGNORED"	163160
GO TO 500	163170
530 IF(IRUN.EQ.3)GO TO 560	163180
PRINT *, "A/C ENTERED EXCEEDS # AVAILABLE--ENTRY IGNORED"	163190
GO TO 500	163200
540 IF(IRUN.EQ.3)GO TO 570	163210
PRINT *, "A/C FLYING CANNOT BE NEGATIVE--ENTRY IGNORED"	163220
GO TO 500	163230
550 CELL(KK)=SA(KK)	163240
GO TO 580	163250
560 CELL(KK)=ELMT(K,L)*ACAV(KK,L)-AIFLOS(13,KK,L)	163260
IF(CELL(KK).GT.SA(KK))CELL(KK)=SA(KK)	163270
GO TO 540	163280
570 CELL(KK)=0.	163290
580 PRINT *, "ENTRY ADJUSTED FOR TYPE ",K," A/C"	163300
GO TO 500	163310
590 ICF=0	163320

IF(I.EQ.1.OR.CELL(3).LE.0)GO TO 630	063330
600 PRINT 610	063340
610 FORMAT(" ENTER AN MISSILE FIRE LAUNCH METHOD- ")	063350
CALL REEC4	063360
IDF=INTX(1)	063370
IF(IDF.GE.1.AND.IDF.LE.2) GO TO 630	063380
IF(IDF.EQ.999) GO TO 620	063390
PRINT 130	063400
620 PRINT " FOR INDIRECT FIRE LAUNCH.....ENTER 1"	063410
PRINT " FOR DIRECT FIRE (AUTO OR REMOTE CONTROL)..ENTER 2"	063420
GO TO 600	063430
630 DO 480 KK=1,7	063440
NPOP(KK)=0	063450
GO 650 ITYP=1,8	063460
POP=PC(KK,ITYP)=0,	063470
IF(ITYP.GT.5)GO TO 650	063480
IF(KK.EQ.5.AND.L.EQ.2.AND.ITYP.GT.1)GO TO 650	063490
IF(IDF.EQ.1.AND.KK.EQ.3.AND.L.EQ.1.AND.ITYP.GT.1)GO TO 650	063500
IF(IDF.EQ.1.AND.KK.EQ.3.AND.L.EQ.1.AND.ITYP.EQ.1)GO TO 640	063510
DIV=APOP(ITYP,L)*POAC(KK,IV)	063520
IF(DIV.LE.0)GO TO 650	063530
NPOP(KK)=NPOP(KK)+ORD(KK,L,ITYP)/DIV	063540
IF(IDF.NE.1.OR.KK.NE.3.OR.L.NE.1)GO TO 650	063550
640 IF(ITYP.GT.1)GO TO 650	063560
DEN=CELL(5)+CELL(7)	063570
IF(DEN.LE.0)GO TO 650	063580
DIV=(CELL(5)*POAC(5,IV)+CELL(7)*POAC(7,IV))/DEN	063590
IF(DIV.LE.0)GO TO 650	063600
NPOP(KK)=NPOP(KK)+ORD(KK,L,ITYP)/DIV	063610
650 CONTINUE	063620
IF(IDF.NE.1.OR.L.NE.1)GO TO 660	063630
IF(KK.EQ.5.OR.KK.EQ.7)NPOP(KK)=NPOP(3)	063640
660 IF(NPOP(KK).LE.0)GO TO 640	063650
DO 670 ITYP=1,8	063660
IF(KK.EQ.3.AND.IDF.EQ.1.AND.L.EQ.1.AND.ITYP.GT.1)GO TO 680	063670
POPORD(KK,ITYP)=ORD(KK,L,ITYP)/NPOP(KK)	063680
670 CONTINUE	063690
680 CONTINUE	063700
C DO 685 KK=1,7	063710
IF(CELL(KK).LE.0)GO TO 685	063720
IF(NW.GT.NMAX-2)GO TO 685	063730
ARRAY(NW)=KK	063740
ARRAY(NW+1)=CELL(KK)	063750
IF(L.NE.1.OR.KK.NE.3)GO TO 683	063760
ARRAY(NW+2)=IDF	063770
NW=NW+1	063780
683 NW=NW+2	063790

685	CONTINUE	163910
	IF(NH.LE.NMAX)ARRAY(NH)=0	163920
	NH=NH+1	163930
	DO 690 I=1,42	163940
690	GFILL(I)=0.	163950
	TSTART=J.	163960
	NPOPUP=0	163970
	DO 700 KK=1,7	163980
	N=NPOP(KK)	163990
	IF(N.GT.MAXPOP(KK,L))N=MAXPOP(KK,L)	163900
	IF(CELL(KK).GT.0..AND.N.GT.NPOPUPINFQFUF=N	163910
	ACKILL(KK,1)=0	163920
	SA(KK)=SA(KK)-CELL(KK)	163930
	IF(KK.EQ.3.AND.IDF.EQ.1.AND.L.EQ.1)GO TO 700	163940
	TSTART=TSTART+CELL(KK)	163950
700	CONTINUE	163960
	TNOW=TSTART	163970
C	BEGIN LOOP TO FLY THE CELL	163980
C		163990
	IF(NPOPUP.EQ.0)NPOPUP=1	164000
	DO 1030 N=1,NPOPUP	164010
	PDF=0.	164020
	DO 710 I=1,42	164030
	ADD=(ELMT(I,J)*OPAV(I,J)-GNCLOS(0,I,J))-FFS(I,J)	164040
	IF(I.LT.13.AND.I.NE.3)ADD=0.	164050
	IF(J.EQ.1.AND.(I.EQ.24.0P.I.EQ.25.0P.I.EQ.33.0P.I.EQ.34.0P.	164060
	* I.EQ.35.1)ADD=0.	164070
	IF(J.EQ.2.AND.(I.EQ.14.0P.I.EQ.20.0P.I.EQ.37.0P.I.EQ.41.1)ADD=1.	164080
	PDF=PDF+ADD	164090
710	CONTINUE	164100
C	AD ASSESSMENT AGAINST A/C	164110
	DO 740 KK=1,7	164120
C		164130
	IF(KK.EQ.3.AND.IDF.EQ.1.AND.L.EQ.1) GO TO 740	164140
	IF(L.EQ.2.AND.(N.GT.MAXPOP(KK,L).OR.N.GT.NPOP(KK)))GO TO 740	164150
	IF(L.EQ.1.AND.(N.GT.MAXPOP(KK,L).OR.(KK.LE.4.AND.N.GT.NPOP(KK)))	164160
	*GO TO 740	164170
C		164180
	IF((CELL(KK)-ACKILL(KK,1)).LE.0.)GO TO 740	164190
	AKILL=1.	164200
	SKILL=3.	164210
	K=KK+55	164220
	DO 720 II=1,12	164230
	I=II+30	164240
	IF(J.EQ.1) ITP=IBAND(II)	164250
	IF(J.EQ.2) ITP=IBAND(II)	164260
	PKILL(II)=1.	164270
	IF(ITP.LE.3) GO TO 720	164280

TPA2=1	064290
IF(I.EQ.36.AND.J.EQ.1)TPA2=2	064300
IF(I.EQ.38)TPA2=2	064310
IF(I.GE.39)TPA2=3	064320
IF(I.EQ.39.AND.J.EQ.1)TPA2=3	064330
ELS=ELMT(I,J)-GMOLOS(I,I,J)	064340
IF(PA2(TPA2)*(ELMT(I,J)*OPAV(I,J)-GMOLOS(I,I,J)).LE.6.1GO TO 720	064350
S=1.-SSP(I)*SFAC(I,I,J)	064360
IF(I.LE.2)S=.2	064370
EXP=PA2(TPA2)*(ELMT(I,J)*OPAV(I,J)-GMOLOS(I,I,J))*T*AFK(ITER=4)	064380
IF(I.EQ.37.AND.J.EQ.1)EXP=EXP*.7	064390
IF(I.EQ.37.AND.J.EQ.2)EXP=EXP*.7	064400
IF(I.EQ.38.AND.J.EQ.2)EXP=EXP*.7	064410
C IF(I.EQ.38.AND.J.EQ.1)EXP=EXP*.7	064420
IF(I.EQ.37.AND.J.EQ.1A.AND.IMOUNT.EQ.2)EXP=EXP/3.	064430
IF(I.EQ.36.AND.J.EQ.1A.AND.IMOUNT.EQ.2.AND.J.EQ.1)EXP=EXP/3.	064440
IF(I.EQ.1.AND.I.GE.35.AND.I.LE.38)	064450
*EXP=EXP*PCNSH(I)	064460
IF(I.EQ.2.AND.I.EQ.37.OR.I.EQ.39)	064470
*EXP=EXP*PCNSH(I)	064480
AQ=1.	064490
IF(I.EQ.1.AND.I.GE.35.AND.I.LE.37)AQ=ACQ(2-34,KK)	064500
IF(I.EQ.2.AND.I.GE.37.AND.I.LE.38)AQ=ACQ(1-33,KK)	064510
EXP=EXP*AQ	064520
EXP=EXP*WEAPC(KTRL)*S*.9*(CELL(KK)-ACKILL(KK,1))/T*CH	064530
PF(II)=PF(II)*(1.-EXP/ELS)	064540
SMTS(II)=SMTS(II)*EXP	064550
IF(SSK(II,J,IR)/(CELL(KK)-ACKILL(KK,1)).GE.1.)GO TO 720	064560
PKILL(II)=(1.-SSK(II,J,IR)/(CELL(KK)-ACKILL(KK,1)))*EXP	064570
SKILL=SKILL*(1.-PKILL(II))	064580
SHOTSV(I,TYP,J)=SHOTSV(I,TYP,J)+EXP	064590
720 AKILL=AKILL*PKILL(II)	064600
PNSMK=1.	064610
IF(I.EQ.1.AND.I.GE.35.AND.I.LE.38)PNSMK=1.-(1.-PCNSH(I))/2.	064620
IF(I.EQ.2.AND.I.GE.37.AND.I.LE.39)PNSMK=1.-(1.-PCNSH(I))/2.	064630
ACLOST(KK)=(1.-AKILL)*(CELL(KK)-ACKILL(KK,1))*PNSMK	064640
IF(ACKILL(KK,1)*ACKILL(KK,2)*ACLOST(KK).GT.CELL(KK))ACLOST(KK)=	064650
1CELL(KK)-ACKILL(KK,1)-ACKILL(KK,2)	064660
C APPORTION LOSSES	064670
DO 730 II=1,12	064680
IF(PKILL(II).GE.1.)GO TO 730	064690
C IF(IAJ(II).EQ.3)GO TO 260	064700
I=II+30	064710
FRAC=(1.-PKILL(II))/SKILL	064720
AIRLOS(II,KK,L)=AIRLOS(II,KK,L)+ACLOST(KK)*FRAC	064730
AIRLOS(13,KK,L)=AIRLOS(13,KK,L)+ACLOST(KK)*FRAC	064740
ACCREW(L)=ACCREW(L)+ACLOST(KK)*CREWS(K-12,L)*FRAC	064750
730 CONTINUE	064760

M-183

	SHOTS(I,TYP,35,L)=SHOTS(I,TYP,35,L)+RCUNC3	365250
	IF(I,TYP,LE,4) ROUNDS=ROUNDS+.8	365260
	IPNG=IR	365270
	IF(I,TYP,NE,1) IRNG=IR+10	365280
C	CHECK FRIENDLY BARRAGE	365290
	FARD=DUST(JVIS,IPNG,2)	365300
	IF(IARD(J),LT,84) FARD=DUST(JVIS,IRNG,1)	365310
	IF(IARD(J),LT,21) FARD=1.	365320
C	CHECK THREAT BARRAGE	365330
	TARD=DUST(JVIS,IPNG,2)	365340
	IF(IARD(L),LT,84) TARD=DUST(JVIS,IPNG,1)	365350
	IF(IARD(L),LT,21) TARD=1.	365360
C	USE DOMINANT EFFECT	365370
	ADUST=FARD	365380
	IF(TARD,LT,FARD) ADUST=TARD	365390
	ABORT=1.	365400
	IF(I,TYP,LE,4) ABORT=1.-ADUST	365410
C	DEGRADE FOR FALSE TARGETS	365420
C	IF(L,EO,1) ABORT=ABORT*.7	365430
C	IF(L,EO,2) ABORT=ABORT*.3	365440
C		365450
	ROUNDS=ABORT*ROUNDS	365460
	PHKILL(KK)=PHKILL(KK)*(1.-POUR/VICTIM)*.P(CUNC3	365470
	938 CONTINUE	365480
	AMKILL=AMKILL+PHKILL(KK)	365490
	SMKILL=SMKILL+1.-PHKILL(KK)	365500
	940 CONTINUE	365510
	VKILL=(1.-AMKILL)*VICTIM	365520
	IF(GNCLOS(I,I,J)+VKILL.GT,ELMT(I,J)*OPAV(I,J)) VKILL=ELMT(I,J)*	365530
	10PAV(I,J)-GNCLOS(I,I,J)	365540
	IF(VKILL,LE,.01) GO TO 890	365550
C	APPROXIMATE LOSSES	365560
	CC 870 KK=1,7	365570
	IF(PHKILL(KK).GE,1) GO TO 870	365580
	GIS=1.	365590
	FRAC=(1.-PHKILL(KK))/SMKILL	365600
	IEL=1	365610
	IF(I,EO,3) GO TO 950	365620
	GNCLOS(KK,I,J)=GNCLOS(KK,I,J)+VKILL*FRAC	365630
	GNCLOS(I,I,J)=GNCLOS(I,I,J)+VKILL*FRAC	365640
	GFKILL(I)=GFKILL(I)+VKILL*FRAC	365650
	IF(I,LT,13) GO TO 970	365660
	GNCLOS(KK,2,J)=GNCLOS(KK,2,J)+VKILL*CREWS(I-12,J)*FRAC	365670
	GNCLOS(I,2,J)=GNCLOS(I,2,J)+VKILL*CREWS(I-12,J)*FRAC	365680
	GFKILL(2)=GFKILL(2)+VKILL*CREWS(I-12,J)*FRAC	365690
	IF(I,LT,21.OR,I,GT,29.OR,IMOUNT,EO,1.CP,J,EO,10) GO TO 870	365700
	GIS=6.	365710
	IEL=3	365720

850 GO 180 IMP=TEL,12	165730
GNLOS(KK,IMP,J)=GNLOS(KK,IMP,J)+VKILL*GIS*PLT(IMP)*FRAC	165740
GNLOS(A,IMP,J)=GNLOS(A,IMP,J)+VKILL*GIS*PLT(IMP)*FRAC	165750
GFKILL(IMP)=GFKILL(IMP)+VKILL*GIS*PLT(IMP)*FRAC	165760
960 CONTINUE	165770
870 CONTINUE	165780
880 CONTINUE	165790
TNOW=0.	165800
DO 890 KK=1,7	165810
ACKILL(KK,1)=ACKILL(KK,1)+ACKILL(KK,2)	165820
ACKILL(KK,2)=0	165830
ACLOST(KK)=0.	165840
IF(KK.EQ.1.AND.TOF.EQ.1.AND.L.EQ.1)GO TO 890	165850
TNOW=TNOW+(CELL(KK)-ACKILL(KK,1))	165860
990 CONTINUE	165870
C CHECK A/C LOSSES FOR ABORT	165880
900 NN=4	165890
IABORT=J	165900
IF(TNOW.GE.TSTART*.7.AND.NN.NE.NPOPOP)GO TO 1030	165910
IF(NN.EQ.NPOPOP)GO TO 920	165920
PRINT *, "LOSSES EXCEED 30% AFTER ",NN," FCPU'S"	165930
IF(IRUN.EQ.1.OR.IRUN.EQ.3)GO TO 910	165940
IABORT=1	165950
GO TO 930	165960
910 PRINT *, "SORTIE ABORTED"	165970
GO TO 930	165980
920 PRINT *, NN, " POPUP SORTIE COMPLETED."	165990
930 PRINT *, "DO YOU WISH TO SEE LOSSES?"	166000
CALL RECA(INX)	166010
IF(INX.EQ."Y")GO TO 940	166020
IF(INX.EQ."N")GO TO 1010	166030
GO TO 930	166040
940 PRINT 950,KIND(I)	166050
950 FORMAT(// " ",A4," HELICOPTERS KILLED"/ " TYPE " " KILLED")	166060
DO 970 KK=1,7	166070
IF(ACKILL(KK,1).LE..1)GO TO 970	166080
K=KK*50	166090
PRINT 960,K,ACKILL(KK,1)	166100
960 FORMAT(" ",3X,I2,10X,F5.1)	166110
970 CONTINUE	166120
PRINT 980,KIND(J)	166130
980 FORMAT(// " ",A4," GROUND FORCES KILLED"/ " TYPE " " KILLED")	166140
DO 1000 I=1,42	166150
IF(GFKILL(I).LE..1)GO TO 1000	166160
PRINT 990,I,GFKILL(I)	166170
990 FORMAT(" ",3X,I2,11X,F5.1)	166180
1000 CONTINUE	166190
1010 IF(IABORT.EQ.1)GO TO 1040	166200

1020 PRINT *, "DO YOU WISH TO ABORT THIS SORTIE?"	066210
C1020 PRINT *, "SORTIE ABORTED"	066220
GO TO 1040	066230
CALL DEEDA(INX)	066240
IF(INX.EQ."Y") GO TO 1040	066250
IF(INX.EQ."N") GO TO 1030	066260
GO TO 1020	066270
1030 CONTINUE	066280
1040 IF(L.EQ.2) GO TO 1049	066290
IF(CELL(3).EQ.0..OR.CELL(4).EQ.0.) GO TO 1042	066300
PRINT *, "THIS CELL HAS MIXED AM TYPES"	066310
GO TO 1049	066320
1042 IF(CELL(3).LE.0.) GO TO 1044	066330
N13=N13+1	066340
INC=(N13-1)*4+11	066350
KK=3	066360
GO TO 1044	066370
1044 IF(CELL(4).LE.0.) GO TO 1049	066380
N14=N14+1	066390
INC=91-N14*4	066400
KK=4	066410
1046 AM(INC)=KK*10	066420
AM(INC+1)=CELL(KK)	066430
AM(INC+2)=CELL(KK)-ACKILL(KK,1)	066440
DIV=AM(INC+2)+ACKILL(KK,1)/1.	066450
IF(DIV.LE.0.) GO TO 1048	066460
RCS=RCS/DIV	066470
AM(INC+3)=AM(INC+2)*RCS	066480
1048 AM(10)=N13+N14	066490
IF(AM(10).LT.20.) GO TO 1049	066500
PRINT *, "NO MORE BLUE CELLS CAN BE FLWVN IN THIS SECTOR."	066510
GO TO 1060	066520
1049 PRINT 1050,KIND(1)	066530
1050 FORMAT(" DO YOU WISH TO FLY ANOTHER CELL CF ".A4," A/C? ")	066540
CALL DEECA(INX)	066550
IF(INX.EQ."Y") GO TO 440	066560
IF(INX.EQ."N") GO TO 1060	066570
GO TO 1040	066580
1060 DO 1070 I=31,42	066590
1070 FPS(I,J)=FPS(I,J)/WEIGHT	066600
DO 1090 I=1,15	066610
IF(GNCLOS(8,I,J).LE.ELMT(I,J)*OPAV(I,J)) GO TO 1090	066620
IF(GNCLOS(8,I,J).LE.0.) GO TO 1090	066630
GO 1080 KK=1,7	066640
1080 GNCLOS(KK,I,J)=GNCLOS(KK,I,J)/GNCLOS(8,I,J)*ELMT(I,J)*OPAV(I,J)	066650
GNCLOS(8,I,J)=ELMT(I,J)*OPAV(I,J)	066660
1090 CONTINUE	066670
DO 1110 I=1,42	066680

GO 1110 K=99.65	366690
KK=K-59	366730
GNCLOS(KK,I,J)=IFIX(GNCLOS(KK,I,J)*10.+5)/10.	366710
ALOSS(K,I)=ALOSS(K,I)+IFIX(GNCLOS(KK,I,J)*10.+001)*PACK(J)	366720
IF(I.LT.3)GO TO 1110	366730
1100 II=I-30	366740
AIRLOS(II,KK,L)=IFIX(AIRLOS(II,KK,L)*10.+5)/10.	366750
ALOSS(I,K)=ALOSS(I,K)+IFIX(AIRLOS(II,KK,L)*10.+301)*PACK(L)	366760
ALOSS(I,2)=ALOSS(I,2)+IFIX(AIRLOS(II,KK,L)*CREWS(K-12,L)*10.+301)*PACK(L)	366770
I=PACK(L)	366780
1110 CONTINUE	366790
IF(ACREW(L).GT.ELMT(2,L))ACREW(L)=ELMT(2,L)	366800
GO TO 1140	366810
1120 PRINT 1130,KIND(L)	366820
1130 FORMAT(" THE ",A6," FORCE HAS NO HELICOPTERS")	366830
1140 CONTINUE	366840
IF(IRUN.EQ.3)GO TO 1200	366850
PRINT *, "DISPLAY LOSSES? "	366860
CALL REEDA(IYN)	366870
IF(IYN.EQ."N")GO TO 1200	366880
PRINT 1150	366890
1150 FORMAT("I")	366900
PRINT *, "-----AM/AD ASSESSMENTS-----"	366910
PRINT 1160	366920
1160 FORMAT(" I",34X,"I")	366930
GO 1270 J=1,2	366940
IFLAG=0	366950
DO 1260 I=1,65	366960
IF(I.GE.43.AND.I.LE.58)GO TO 1250	366970
CLOST=0.	366980
IF(I.LT.15.AND.I.NE.13)GO TO 1170	366990
IF(I.GE.59.AND.I.LE.65)GO TO 1250	367000
CLOST=GNCLOS(8,I,J)*CREWS(I-12,J)	367010
1170 AKILL=IFIX(GNCLOS(8,I,J)*10.+5)/10.	367020
IF(I.EQ.2)AKILL=AKILL+ACREW(J)	367030
1180 IF(AKILL.LT..1.AND.CLOST.LT..1)GO TO 1260	367040
IF(IFLAG.GT.0)GO TO 1210	367050
IFLAG=1	367060
IF(J.EQ.2)GO TO 1190	367070
PRINT *, "I TOTAL BLE LOSSES I"	367080
GO TO 1230	367090
1190 PRINT *, "I TOTAL REC LOSSES I"	367100
1200 PRINT *, "I ITEM LOST CREW I"	367110
PRINT 1160	367120
1210 IF(I.LT.15.AND.I.NE.13)GO TO 1230	367130
PRINT 1220,I,AKILL,CLOST	367140
1220 FORMAT(" I",9X,I2,F9.1,F7.1,7X,"I")	367150
GO TO 1260	367160

1230 PRINT 1240, I, AKILL	067170
1240 FORMAT(" I", 9X, I2, F9.1, 14X, "I")	067180
GO TO 1260	067190
1250 CLOST=4IFLOS(13, I-59, J)*CREWS(I-12, J)	067200
AKILL=IFIX(4IRLOS(13, I-58, J)*10.+.5)/10.	067210
GO TO 1190	067220
1260 CONTINUE	067230
IF(IFLAG.NE.0)PRINT 1160	067240
IF(L.EQ.2)GO TO 1270	067250
PRINT *, "*****"	067260
PRINT 1160	067270
1270 CONTINUE	067280
POINT *, "*****"	067290
1280 IF(IRUN.EQ.1)POINT *, "AN/AC ASSESSMENTS PRINTED HERE"	067300
INX=0	067310
CALL LOSS(59, 65, 1, 42, INX)	067320
CALL LOSS(31, 42, 59, 65, INX)	067330
IF(INX.EQ."NO")GO TO 1290	067340
CALL CKSTOR(AH1, IPUT)	067350
IPUT=-1	067360
AH1="CI AD"	067370
CALL CKSTOR(AH1, IPUT)	067380
DO 1295 I=1, 12	067390
INC=11+(I-1)*4	067400
AH(INC)=PF(I)	067410
AH(INC*2)=SHYS(I)	067420
1295 CONTINUE	067430
CALL CKSTOR(AH1, IPUT)	067440
C	067450
IPUT=-1	067460
AH1="INPUT 5"	067470
CALL CKSTOR(AH1, IPUT)	067480
DO 1297 I=11, 90	067490
1297 AH(I)=ARRAY(I)	067500
CALL CKSTOR(AH1, IPUT)	067510
PRINT 1150	067520
1290 END	067530

M-189

CALL XOUT(1)	072440
XCI=ACI	072450
XSECT=ASECT	072460
IOUT=1	072470
IF(XSECT.EQ.0.)IOUT=3	072480
IF(XCI.EQ.0.)IOUT=5	072490
IF(IRUN.EQ.3.OR.IRUN.GE.6)GO TO 210	072500
IF(XSECT.EQ.0.)GO TO 140	072510
C	072520
100 PRINT 110,XSECT,XCI	072530
110 FORMAT(" SECTOR OUTPUT FOR SECTOR ",F6.3," OF CI ",A12,"? ")	072540
CALL REEDA(IYN)	072550
120 FORMAT(A1)	072560
IF(IYN.EQ."Y")GO TO 220	072570
IF(IYN.EQ."N")GO TO 140	072580
PRINT 130	072590
130 FORMAT(" MUST BE Y OR N ")	072600
GO TO 100	072610
C	072620
140 PRINT "SELECT OUTPUT OPTION (0 IF DONE) - "	072630
CALL REED4	072640
IOUT=XINX(1)	072650
IF(IOUT.EQ.0)GO TO 230	072660
IF(IOUT.GE.1.AND.IOUT.LE.5)GO TO 170	072670
IF(IOUT.EQ.999)GO TO 160	072680
PRINT 150	072690
150 FORMAT(" INCORRECT ENTRY ")	072700
PRINT "LIST OPTIONS? "	072710
CALL REEDA(IYN)	072720
IF(IYN.EQ."N")GO TO 140	072730
160 PRINT " INPUT OUTPUT CREATED "	072740
PRINT " 1 SINGLE SECTOR "	072750
PRINT " 2 PARTIAL CI (MULTIPLE SECTOR) "	072760
PRINT " 3 FULL CI "	072770
PRINT " 4 PARTIAL GAME (MULTIPLE CI) "	072780
PRINT " 5 FULL GAME "	072790
GO TO 140	072800
C	072810
170 IF(IOUT.EQ.4)GO TO 200	072820
IF(IOUT.EQ.5)GO TO 210	072830
PRINT "CI NAME- "	072840
CALL REED8(ACI)	072850
180 FORMAT(A10)	072860
ASECT=0.	072870
IF(IOUT.EQ.3)GO TO 220	072880
PRINT "SECTOR #- "	072890
190 CALL REED4	072900
ASECT=XINX(1)	072910

IF(IQUT.EQ.1)GO TO 220	072920
WRITE(70,*)ASECT	072930
IF(ASECT.EQ.0)GO TO 220	072940
PRINT *, "NEXT (0 IF DONE) - "	072950
GO TO 190	072960
200 PRINT *, "OUTPUT FROM CI- "	072970
CALL REEBB(CCI(1))	072980
PRINT *, "THROUGH CI- "	072990
CALL REEBB(CCI(2))	073000
C	073010
C *** BEGIN CALLS TO SUBROUTINES WHICH CREATE FILE ***	073020
C	073030
210 IF(IQUT.GT.3)CALL GITNMS	073040
220 CALL ACCUM	073050
CALL HFCER	073060
CALL REPCRT	073070
CALL XOUT(1)	073080
IF(IPUN.EQ.3.OR.IRUN.GE.6)GO TO 230	073090
GO TO 140	073100
230 ACT=XCI	073110
ASECT=XSECT	073120
IFIRST=0	073130
END	073140

M-192

173 CONTINUE

NCI1=0

NCI2=0

180 RETURN

END

37364C

37365C

37366C

37367C

37368C

140 IF(NCI1.LE.NCI2)GO TO 200	374190
PRINT 150,CCI	374190
150 FOPWAI(" CI ".A10," COMES AFTER CI ".A10," IN GAME"/	374200
" - CORRECTION MADE.")	374210
NN=NCI1	374220
NCI1=NCI2	374230
NCI2=NN	374240
HOLD=CCI(1)	374250
CCI(1)=CCI(2)	374260
CCI(2)=HOLD	374270
GO TO 200	374280
C *** FULL GAME OUTPUT--SET NCI1, NCI2, CCI(1), CCI(2) ***	374290
C	374300
160 NCI1=4	374310
DO 170 I=4,89	374320
IF(ARRAY(I,1).NE.C.)GO TO 170	374330
NCI2=I	374340
GO TO 180	374350
170 CONTINUE	374360
NCI2=90	374370
180 CCI(1)=ARRAY(NCI1)	374380
CCI(2)=ARRAY(NCI2)	374390
IF(NCI2.GT.4)GO TO 190	374400
C ** ERROR--AT MOST ONE CI NAME ON RECCPD (NO GAMING) **	374410
C	374420
PRINT "HISTORYFILE SHOWS NO GAMING HAS BEEN COMPLETED."	374430
ICUT=0	374440
GO TO 230	374450
190 IF(ARRAY(NCI1).NE.0..AND.ARRAY(NCI2).NE.0.)GO TO 230	374460
C ** ERROR--ONE OF INDEXES NCI1,NCI2 IS INCORRECT **	374470
C	374480
PRINT "ERROR ON RETREIVING CI NAMES"	374490
ICUT=0	374500
200 RETURN	374510
END	374520

140 CONTINUE	075030
GO TO 170	075040
150 IF(AH(1).EQ."RED ENC")J=2	075050
GO 160 I=1,65	075060
ELMT(I,J)=ELMT(I,J)+AH(I+10)	075070
160 CONTINUE	075080
170 IF(AH(1).EQ."RED ENC")GO TO 180	075090
IF(AH(1).EQ."RED STAR")AH(1)="RED ENC"	075100
IF(AH(1).EQ."BLUE END")AH(1)="RED STAR"	075110
IF(AH(1).EQ."BLUE STAR")AH(1)="BLUE ENC"	075120
GO TO 110	075130
C	075140
C ** FOLLOWING FILLS "ALOSS" ARRAY	075150
C	075160
180 IF(IALOSS.GT.0)GO TO 230	075170
AH(1)="CI LOSSES"	075180
AH(2)=ACI	075190
AH(4)=99999.	075200
CALL GET(IHIST,AH,AH(1),0,20)	075210
IF(AH(4).NE.99999.)GO TO 200	075220
IALOSS=1	075230
PRINT 190,AH(1),ACI	075240
190 FORMAT(" ",A13," RECORDS FOR CI ",A10," ACT FOUND".	075250
1 " (ACCUM)")	075260
GO TO 230	075270
200 IF(AH(5).NE.ASECT)GO TO 220	075280
K=AH(4)	075290
GO 210 I=1,65	075300
ALOSS(K,I)=ALOSS(K,I)+AH(I+10)	075310
210 CONTINUE	075320
220 CALL GETN(IHIST,AH,AH(1))	075330
M=IFETCH(IHIST,2LFB)	075340
IF(M.EQ.1000)GO TO 230	075350
IF(AH(1).NE."CI LOSSES".OR.AH(2).NE.ACI)GO TO 230	075360
GO TO 200	075370
C	075380
C ** FOLLOWING FILLS "SHOTS" ARRAY	075390
C	075400
230 IF(IAMNO.GT.0)GO TO 270	075410
AH(1)="CI AMNO"	075420
AH(2)=ACI	075430
AH(4)=99999.	075440
CALL GET(IHIST,AH,AH(1),0,20)	075450
IF(AH(4).NE.99999.)GO TO 240	075460
IAMNO=1	075470
PRINT 190,AH(1),ACI	075480
GO TO 270	075490
240 IF(AH(5).NE.ASECT)GO TO 260	075500

J=AM(4)	175510
DO 250 I=1,55	175520
SHOTS(I,J)=SHOTS(I,J)+AM(I+10)	175530
250 CONTINUE	175540
260 CALL GETN(IHIST,AM,AM(1,1))	175550
M=IFETCH(IHIST,2LFP)	175560
IF(M.EQ.1009)GO TO 270	175570
IF(AM(1).NE."CI AMMO".OR.AM(2).NE.AC1)GO TO 270	175580
GO TO 240	175590
270 IF(IOUT.EQ.2)GO TO 290	175600
CALL CLOSEN(IHIST)	175610
GO TO 650	175620
C *** BEGIN FILLING ARRAYS FOR PARTIAL CI OUTPUT ***	175630
C ** USES LOGIC ABOVE (IE., FOR SINGLE SECTOR) TO ACTUALLY	175640
C RETRIEVE THE RECORDS AND FILL THE ARRAYS **	175650
C	175660
280 CALL OPENN(IHIST,3LI-0,1LR)	175670
RETN0 70	175680
290 READ(70,*)ASECT	175690
IF(ASECT.EQ.0.)GO TO 300	175700
AM(1)="BLUE START"	175710
GO TO 110	175720
300 CALL CLOSEN(IHIST)	175730
GO TO 650	175740
C *** BEGIN FILLING ARRAYS FOR FULL CI ***	175750
C	175760
C ** FOLLOWING FILLS "BEGIN" AND "ELMT" ARRAYS	175770
C	175780
310 CALL OPENN(IHIST,3LI-0,1LR)	175790
AM(1)="BLUE START"	175800
320 AM(2)=AC1	175810
AKEEP=AM(1)	175820
AM(6)=999999.	175830
CALL GET(IHIST,AM,AM(1),3,20)	175840
IF(AM(4).NE.999999.)GO TO 330	175850
PRINT 190,AM(1),AC1	175860
GO TO 340	175870
330 J=1	175880
IF(AM(1).EQ."BLUE END".OR.AM(1).EQ."RED END")GO TO 350	175890
IF(AM(1).EQ."RED START")J=2	175900
IF(AM(3).LE.0.)GO TO 370	175910
GO 340 I=1,65	175920
BEGIN(I,J)=BEGIN(I,J)+AM(I+10)	175930
340 CONTINUE	175940
GO TO 370	175950
350 IF(AM(1).EQ."RED END")J=2	175960

DO 360 I=1,65	175990
ELMT(I,J)=ELMT(I,J)+AM(I+10)	176000
360 CONTINUE	176010
370 CALL GETN(IHIST,A,AM(1))	176020
M=IFETCH(IHIST,2LFP)	176030
IF(M.EQ.1008)GO TO 380	176040
IF(AM(1).NE.AKEEP.OR.AM(2).NE.ACI)GO TO 380	176050
GO TO 330	176060
380 IF(AKEEP.EQ."EO END")GO TO 399	176070
IF(AKEEP.EQ."RED START")AM(1)="REC ENC"	176080
IF(AKEEP.EQ."BLUE END")AM(1)="REC START"	176090
IF(AKEEP.EQ."BLUE START")AM(1)="BLUE ENC"	176100
GO TO 320	176110
C	176120
C ** FOLLOWING FILLS "ALOSS" ARRAY	176130
C	176140
390 AM(1)="CI LOSSES"	176150
AM(2)=ACI	176160
AM(4)=99999.	176170
CALL GET(IHIST,AM,AM(1),0,20)	176180
IF(AM(4).NE.99999.)GO TO 400	176190
PRINT 190,AM(1),ACI	176200
GO TO 420	176210
400 K=AM(4)	176220
DO 410 I=1,65	176230
ALOSS(K,I)=ALOSS(K,I)+AM(I+10)	176240
410 CONTINUE	176250
CALL GETN(IHIST,A,AM(1))	176260
M=IFETCH(IHIST,2LFP)	176270
IF(M.EQ.1008)GO TO 420	176280
IF(AM(1).NE."CI LOSSES".OR.AM(2).NE.ACI)GO TO 420	176290
GO TO 400	176300
420 AM(1)="CI AMMO"	176310
C	176320
C ** FOLLOWING FILLS "SHOTS" ARRAY	176330
C	176340
AM(2)=ACI	176350
AM(4)=99999.	176360
CALL GET(IHIST,AM,AM(1),0,20)	176370
IF(AM(4).NE.99999.)GO TO 430	176380
PRINT 190,AM(1),ACI	176390
GO TO 450	176400
430 J=AM(4)	176410
DO 440 I=1,65	176420
SHOTS(I,J)=SHOTS(I,J)+AM(I+10)	176430
440 CONTINUE	176440
CALL GETN(IHIST,AM,AM(1))	176450
M=IFETCH(IHIST,2LFP)	176460

IF(N.EQ.1003)GO TO 450	37647C
IF(AH(1).NE."CI AMMO".OR.AH(2).NE.ACI)GO TO 450	3764AC
GO TO 430	37649C
450 CALL CLOSEM(IHIST)	37650C
GO TO 450	37651C
C	37652C
C *** BEGIN FILLING ARRAYS FOR PARTIAL GAME OR FULL GAME OUTPUTS ***	37653C
C	37654C
460 CALL OPENM(IHIST,3LI-C,1LR)	37655C
C	37656C
C **FOLLOWING FILLS "BEGIN" ARRAY	37657C
C	37658C
AM1="BLUE START"	37659C
J=1	37660C
ACT=ARRAY(NGI1)	37661C
470 AM(1)=AM1	37662C
AM(2)=ACT	37663C
AM(6)=99999.	37664C
CALL GET(IHIST,AM,AM(1),0,20)	37665C
IF(AH(4).NE.99999.)GO TO 500	37666C
PRINT 450,KIND(IJ)	37667C
480 FORMAT(" NO INITIAL WPN ARRAYS FOR ",A4," FORCE.")	37668C
GO TO 520	37669C
490 CALL GETN(IHIST,AM,AM(1))	37670C
M=IFETCH(IHIST,2LFP)	37671C
IF(N.EQ.1003)GO TO 520	37672C
IF(AH(1).NE.AM1.OR.AH(2).NE.ACI)GO TO 520	37673C
500 DO 510 I=1,45	37674C
BEGIN(I,J)=BEGIN(I,J)+AM(I+10)	37675C
510 CONTINUE	37676C
GO TO 490	37677C
520 IF(AH1.EQ."RED START")GO TO 530	37678C
AM1="RED START"	37679C
J=2	37680C
GO TO 470	37681C
C	37682C
C ** FOLLOWING FILLS "ALOSS" AND "SHOTS" ARRAYS	37683C
C	37684C
530 AM1="CI LOSSES"	37685C
540 AM(1)=AM1	37686C
AM(6)=99999.	37687C
CALL GET(IHIST,AM,AM(1),0,10)	37688C
IF(AH(4).NE.99999.)GO TO 570	37689C
PRINT 550,AM1	37690C
550 FORMAT(" NO ".A10," RECORDS ON FILE")	37691C
GO TO 630	37692C
560 CALL GETN(IHIST,A1,AM(1))	37693C
M=IFETCH(IHIST,2LFP)	37694C

IF(N.EQ.1000)GO TO 630	175950
IF(AH(1).NE.AH1)GO TO 630	176960
570 IF(IOUT.EQ.51)GO TO 590	176970
CO 580 NN=NCI1,NCI2	176980
IF(AH(2).EQ.ARRAY(NN))GO TO 590	176990
580 CONTINUE	177000
GO TO 560	177010
590 IF(AH1.EQ."CI AMMO")GO TO 610	177020
K=AH(4)	177030
DO 600 I=1,65	177040
ALOSS(K,I)=ALOSS(K,I)+AH(I+10)	177050
600 CONTINUE	177060
GO TO 560	177070
610 J=AH(4)	177080
DO 620 I=1,55	177090
SHOTS(I,J)=SHOTS(I,J)+AH(I+10)	177100
620 CONTINUE	177110
GO TO 560	177120
630 IF(AH1.EQ."CI AMMO")GO TO 640	177130
AH1="CI AMMO"	177140
GO TO 560	177150
640 CALL CLOSEN(IHIST)	177160
C ** THIS SUBROUTINE FILLS "ELMT" ARRAY	177170
C	177180
CALL COMMIT	177190
C ** THIS SUBROUTINE FILLS C, CATLOS, CATSUP ARRAYS	177200
C	177210
650 CALL SUMUP	177220
660 RETURN	177230
END	177240

140	CONTINUE	077740
	GO TO 120	077750
C	*** IN "UNCOMMITTED" UNIT--SUBTRACT WPNS FROM BEGIN ARFAY ***	077760
C		077770
C		077780
150	DO 160 I=1,65	077790
	BEGIN(I,J)=BEGIN(I,J)-AM(I+10)	077800
160	CONTINUE	077810
	GO TO 120	077820
170	CALL CLOSEM(INIST)	077830
180	RETURN	077840
	END	077850

N2=NR2
GO TO 110
130 RETURN
END

279350
379360
279370
379380

M-207

M-209

WRITE(6,120)	081050
WRITE(6,260)	081060
WRITE(6,130)	081070
WRITE(6,260)	081080
WRITE(6,140)	081090
WRITE(6,260)	081100
C	081110
C	081120
INX=0	081130
ITM=0	081140
IPP=2	081150
IF(IA.EQ.1)IPP=1	081160
NR=0	081170
NR=0	081180
150 RTKILL=0.	081190
RTKILL=0.	081200
BREP=0.	081210
RREP=0.	081220
J=1	081230
INUM=NR+1	081240
IF(INUM.GT.65)GO TO 190	081250
DO 160 I=INUM,65	081260
NR=I	081270
IF(I.EQ.1)GO TO 150	081280
RTKILL=0(I,1)	081290
IF(I.LE.15)GO TO 170	081300
IF(I.GE.59)GO TO 170	081310
IF(I.GE.15.AND.I.LE.20)INX=1	081320
IF(I.GE.21.AND.I.LE.30)INX=2	081330
IF(I.GE.31.AND.I.LE.58)INX=3	081340
DO 160 K=1,13	081350
ITM=2	081360
IF(K.EQ.8)ITM=1	081370
BREP=BREP+IFIX(RECOV(IPP,ITM,INX)*CATLCS(I,K,J)*.5)	081380
160 CONTINUE	081390
170 BNREP=RTKILL-BREP	081400
BTME=IFIX(BREP*TMER(IPP,ITM,INX)*.5)	081410
BCOS=IFIX(BREP*FOS(IPP,ITM,INX)*.5)	081420
BCIV=RTKILL-(BNREP+BTME+BCOS)	081430
GO TO 190	081440
180 CONTINUE	081450
190 INUM=NR+1	081460
IF(INUM.GT.65.AND.NR.GE.65)GO TO 230	081470
IF(INUM.GT.65)GO TO 150	081480
DO 210 I=INUM,65	081490
NR=I	081500
IF(2(I,2).LE.5)GO TO 210	081510
RTKILL=0(I,2)	081520

RREP=0(I,2)	001510
IF(I.LE.15)RREP=0	001510
IF(I.GE.59)RREP=0	001510
IF(I.LE.15.OR.I.GE.59)GO TO 200	001560
IF(I.GT.15.AND.I.LT.59)RREP=PREC-CATLOS(I,1,2)	001570
RREP2=IFIX(RREP*PREC(1)+.5)	001580
RREP5=IFIX(RREP*PREC(2)+.5)	001590
RREP10=IFIX(RREP*PREC(3)+.5)	001600
RREP2=RREP2+IFIX(CATLCS(I,1,2)*.275*PREC(1)+.5)	001610
RREP5=RREP5+IFIX(CATLCS(I,1,2)*.275*PREC(2)+.5)	001620
RREP10=RREP10+IFIX(CATLCS(I,1,2)*.275*PREC(3)+.5)	001630
RREP=RREP2+RREP5+RREP10	001640
200 CONTINUE	001650
RNREP=RTKILL-RREP	001660
GO TO 220	001670
210 CONTINUE	001680
220 IF(RTKILL.GT.0.)	001690
* WRITE(6,240) NR,RTKILL,RNREP,PREC,RTHEA,RCOS,BDIV	001700
IF(RTKILL.GT.0.)	001710
* WRITE(6,250) NR,RTKILL,RNREP,RREP,RREP2,RREP5,RREP10	001720
WRITE(6,260)	001730
GO TO 150	001740
230 WRITE(6,270)	001750
240 FORMAT(' ',6X,I2,2(3X,F6.0),4X,F6.0,3X,F6.0,2(2X,F6.0))	001760
250 FORMAT(' ',70X,I2,2(3X,F6.0),4X,F6.0,3X,F6.0,2(2X,F6.0))	001770
260 FORMAT(' ',I',63X,'I',63X,'I')	001780
270 FORMAT(' ',1X,I29(' '))	001790
RETURN	001800
END	001810

WRITE(6,115)	002310
WRITE(6,116)	002320
WRITE(6,119)	002330
DO 140 I=1,65	002340
IF (I(I,J).LE..1) GO TO 140	002350
WRITE(6,120) I	002360
DO 130 K=1,13	002370
IF (CATLOS(I,K,J).LE..1) GO TO 130	002380
IFOR(2)=NUMB((K-1)*9)	002390
WRITE(6,IFOR) CATLOS(I,K,J)	002400
130 CONTINUE	002410
IFOR(2)=NUMB(118)	002420
WRITE(6,IFOR) C(I,J)	002430
140 CONTINUE	002440
CALL NEWPAG	002450
WRITE(6,100) KIND(J)	002460
WRITE(6,110)	002470
WRITE(6,117)	002480
WRITE(6,118)	002490
WRITE(6,119)	002500
DO 147 KK=1,16	002510
YCAT=0	002520
DO 145 K=1,13	002530
XCAT(K)=0	002540
DO 141 I=1,65	002550
IF (MAP(I,J).EQ.KK) XCAT(K)=XCAT(K)+CATLOS(I,K,J)	002560
141 CONTINUE	002570
YCAT=YCAT+XCAT(K)	002580
145 CONTINUE	002590
IF (YCAT.LE..1) GO TO 147	002600
WRITE(6,125) ICAT(KK)	002610
DO 146 K=1,13	002620
IF (XCAT(K).LE..1) GO TO 146	002630
IFOR(2)=NUMB((K-1)*9)	002640
WRITE(6,IFOR) XCAT(K)	002650
146 CONTINUE	002660
IFOR(2)=NUMB(118)	002670
WRITE(6,IFOR) YCAT	002680
147 CONTINUE	002690
150 CONTINUE	002700
160 RETURN	002710
END	002720

CALL NEWPAG	043540
JJ=3-J	043590
WRITE(6,140)	043600
140 FORMAT(" *****KILLER-VICTIM MATRIX*****")	043610
150 FORMAT(8X,"",49X,A4,1X,A10)	043620
WRITE(6,150)KIND(JJ), CLASS(MI)	043630
160 FORMAT(" VICTIM ",47X,"KILLER")	043640
WRITE(6,160) (IK,IK=L,M)	043650
170 FORMAT(8X,"",5X,20(I2,4X))	043660
WRITE(6,170)	043670
180 FORMAT(" *****")	043680
200 220 I=LL,MM	043690
SUM=0.	043700
DO 190 II=1,IT	043710
SUM=SUM+XKILL(I,II,J)	043720
190 CONTINUE	043730
IF(SUM.LE.0.)GO TO 220	043740
WRITE(6,200)I	043750
200 FORMAT("G",2X,I2,3X," ")	043760
DO 210 II=1,IT	043770
IF(XKILL(I,II,J).LE.0.)GO TO 210	043780
IFOR(2)=NUMB((II-1)*6)	043790
WRITE(6,IFOR)XKILL(I,II,J)	043800
210 CONTINUE	043810
220 CONTINUE	043820
WRITE(6,190)	043830
230 CONTINUE	043840
240 CONTINUE	043850
C	043860
C TOTAL KILLER-VICTIM MATRIX	043870
C	043880
DO 350 J=1,2	043890
IT1=1	043900
DO 340 JJ=1,4	043910
IF(IT1.GT.66)GO TO 350	043920
I*=1	043930
DO 290 I=IT1,66	043940
II=I	043950
CTOF=0	043960
DO 270 K=1,65	043970
IF(J.EQ.1)GO TO 250	043980
XKILL(K,II,J)=(ALOSS(I,K)-IFIX(ALOSS(I,K)/PACK(1))*PACK(1))/1.	043990
GO TO 260	044000

250	XKILL(K,IT,J)=IFIX(ALOSS(I,K)/BACK(I))/10.	084060
260	CTOT=CTOT+XKILL(K,IT,J)	084070
270	CONTINUE	084080
	IF(CTOT.EQ.0.)GO TO 280	084090
	CHEAD(IT)=I	084100
	IT=IT+1	084110
	IF(IT.GT.20)GO TO 290	084120
280	CONTINUE	084130
290	IT=IT-1	084140
	IT=II+1	084150
	IF(IT.LE.0)GO TO 340	084160
	CALL NEWPAG	084170
	WRITE(6,370)	084180
	WRITE(6,300)KIND(3-J)	084190
300	FORMAT(2X,"*",42X,A4)	084200
	WRITE(6,160)	084210
	WRITE(6,170)(CHEAD(IK),IK=1,IT)	084220
	WRITE(6,180)	084230
	DO 330 K=1,65	084240
	SUM=0	084250
	DO 310 II=1,IT	084260
310	SUM=SUM+XKILL(K,II,J)	084270
	IF(SUM.LE.0)GO TO 330	084280
	WRITE(6,200)K	084290
	DO 320 II=1,IT	084300
	IF(XKILL(K,II,J).LE.0)GO TO 320	084310
	IFOR(2)=NUMB((II-1)*6)	084320
	WRITE(6,IFOR)XKILL(K,II,J)	084330
320	CONTINUE	084340
330	CONTINUE	084350
	WRITE(6,180)	084360
340	CONTINUE	084370
350	CONTINUE	084380
360	RETURN	084390
C		084400
370	FORMAT(" ",44(" "), "AGGREGATED KILLF-VICTIM MATRIX",54(" "))	084410
	END	084420

140 CALL REED4	084920
IMPW=XINX(1)	084930
IF(IMPW.EQ.0)GO TO 180	084940
IF(IMPW.GE.11.AND.IMPW.LE.65)GO TO 160	084950
PRINT *, "NOT A VALID WPN #--MUST BE BETWEEN 11 AND 65"	084960
150 PRINT *, "NEXT- "	084970
GO TO 140	084980
160 NSKP=NSKP+3	084990
IF(NSKP.GT.44)WRITE(6,220)	085000
IF(NSKP.GT.44)NSKP=20	085010
IFOR(2)=NUMB(NSKP)	085020
WRITE(6,IFOR)IMPW	085030
TOTAL(3,1,1)=TOTAL(3,1,1)+0(IMPW,1)	085040
TOTAL(3,1,2)=TOTAL(3,1,2)+0(IMPW,1)*FPS(IMPW,1)	085050
DO 170 IV=16,65	085060
AKILL=(ALOSS(IMPW,IV)-IFIX(ALOSS(IMPW,IV)/PACK(1))-PACK(1))/17.	085070
AKILL=IFIX(AKILL+.5)	085080
IF(AKILL.LE.0)GO TO 170	085090
TOTAL(3,2,1)=TOTAL(3,2,1)+AKILL	085100
TOTAL(3,2,2)=TOTAL(3,2,2)+AKILL*FPS(IV,2)	085110
170 CONTINUE	085120
GO TO 150	085130
180 CALL RATIOS(3)	085140
IF(NUM.EQ.0)WRITE(6,270)	085150
PRINT *, "ANOTHER LEFT "	085160
GO TO 110	085170
190 IF(NUM.NE.0)WRITE(6,270)	085180
200 RETURN	085190
210 FORMAT(A1)	085200
220 FORMAT(" ",T3,"I",T4,"I",T95,"I")	085210
230 FORMAT(" ",T3("= ")," OPTICAL LER'S ",T32("= ")))	085220
240 FORMAT(" ",T24,"WEIGHTED EQUALLY",T59,"WEIGHTED BY FPS")	085230
250 FORMAT(" ",T5,"INCLUDES BLUE: ")	085240
260 FORMAT(" ",T3,T33("= ")))	085250
270 FORMAT(" ",T3,T33("= ")))	085260
END	085270

120	IF(TOTAL(3,1,1).GT.0..AND.TOTAL(3,1,2).GT.0.)GO TO 130	085770
	PRINT *, "NO LER'S COMPUTED--BLUE LOSSES = 0"	085780
	ALER=0.	085790
	WLER=0.	085800
	GO TO 140	085810
130	ALER=TOTAL(3,2,1)/TOTAL(3,1,1)	085820
	WLER=TOTAL(3,2,2)/TOTAL(3,1,2)	085830
C		085840
C	** IFR **	085850
C		085860
140	IF(JFLAG.EQ.3)GO TO 230	085870
	IF(TOTAL(1,1,1).GT.0..AND.TOTAL(1,1,2).GT.0.)GO TO 150	085880
	PRINT *, "NO IFR'S COMPUTED--BLUE INITIAL = 0."	085890
	AIER=0.	085900
	WIER=0.	085910
	GO TO 160	085920
150	AIER=TOTAL(1,2,1)/TOTAL(1,1,1)	085930
	WIER=TOTAL(1,2,2)/TOTAL(1,1,2)	085940
C		085950
C	** FER **	085960
C		085970
160	IF(AIER.GT.0..AND.WIER.GT.0.)GO TO 170	085980
	PRINT *, "NO IFR'S COMPUTED--IFR = 0"	085990
	AFER=0.	086000
	WFER=0.	086010
	GO TO 180	086020
170	AFER=ALER/AIER	086030
	WFER=WLER/WIER	086040
C		086050
C	** RED SMFR **	086060
C		086070
180	IF(TOTAL(1,2,1).GT.0..AND.TOTAL(1,2,2).GT.0.)GO TO 190	086080
	PRINT *, "NO RED SMFR'S COMPUTED--REC INITIAL = 0."	086090
	RSMFR=0.	086100
	WRSNFR=0.	086110
	GO TO 200	086120
190	PSMFR=TOTAL(2,2,1)/TOTAL(1,2,1)	086130
	WRSNFR=TOTAL(2,2,2)/TOTAL(1,2,2)	086140
C		086150
C	** BLUE SMFR **	086160
C		086170
200	IF(TOTAL(1,1,1).GT.0..AND.TOTAL(1,1,2).GT.0.)GO TO 210	086180
	PRINT *, "NO BLUE SMFR'S COMPUTED--BLUE INITIAL = 0"	086190
	BSMFR=0.	086200
	WBSNFR=0.	086210
	GO TO 220	086220
210	BSMFR=TOTAL(2,1,1)/TOTAL(1,1,1)	086230
	WBSNFR=TOTAL(2,1,2)/TOTAL(1,1,2)	086240

C		246250
C	** SMFRD **	246260
C		246270
	220 SMFRD=9SMFR-RSMFR	246280
	WSMFRD=9WSMFR-WRSMFR	246290
C		246300
C	*** WRITE OUTPUT TO TAPES ***	246310
C		246320
	230 WRITE (6,330)	246330
	WRITE (6,340) TOTAL (3,2,1), TOTAL (3,2,2)	246340
	WRITE (6,330)	246350
	WRITE (6,370) ALER, WLER	246360
	WRITE (6,340) TOTAL (3,1,1), TOTAL (3,1,2)	246370
	WRITE (6,330)	246380
	IF (JFLAG.EQ.3) GO TO 260	246390
	WRITE (6,330)	246400
	WRITE (6,340) TOTAL (2,1,1), TOTAL (2,1,2)	246410
	WRITE (6,330)	246420
	WRITE (6,390) 8SMFR, W8SMFR	246430
	WRITE (6,340) TOTAL (1,1,1), TOTAL (1,1,2)	246440
	WRITE (6,330)	246450
	WRITE (6,330)	246460
	WRITE (6,340) TOTAL (2,2,1), TOTAL (2,2,2)	246470
	WRITE (6,330)	246480
	WRITE (6,390) RSMFR, WRSMFR	246490
	WRITE (6,340) TOTAL (1,2,1), TOTAL (1,2,2)	246500
	WRITE (6,330)	246510
	WRITE (6,330)	246520
	WRITE (6,330)	246530
	WRITE (6,400) 9SMFR, RSMFR, SMFRD, W9SMFR, W9SMFRD, WSMFRD	246540
	WRITE (6,330)	246550
	IF (JFLAG.GT.1) GO TO 240	246560
	WRITE (6,330)	246570
	WRITE (6,340) TOTAL (1,2,1), TOTAL (1,2,2)	246580
	WRITE (6,330)	246590
	WRITE (6,410) AIFR, WIFR	246600
	WRITE (6,340) TOTAL (1,1,1), TOTAL (1,1,2)	246610
	WRITE (6,330)	246620
	WRITE (6,330)	246630
	WRITE (6,340) ALEF, WLEF	246640
	WRITE (6,330)	246650
	WRITE (6,420) AFER, WFER	246660
	WRITE (6,340) AIFR, WIFR	246670
	WRITE (6,330)	246680
	240 NP=NP+1	246690
	IF (NP.GT.1) GO TO 250	246700
	IF (JFLAG.EQ.1) WRITE (6,350)	246710
	IF (JFLAG.EQ.2) WRITE (6,360)	246720

GO TO 100	106730
250 WRITE(6,350)	106740
260 RETURN	106750
270 FORMAT(" ",31(" "), " RATIO STATISTICS ",32(" "))	106760
280 FORMAT(" ",T3,"I",T48,"I",T85,"I",T5,"ALL MAJOR WEAPON SYSTEMS ")	106770
290 FORMAT(" ",T3,"I",T48,"I",T85,"I",T5,"ARMOR WEAPON SYSTEMS ONLY")	106780
300 FORMAT(" ",T3,"I",T48,"I",T85,"I",T24,"WEIGHTED EQUALLY",T59,	106790
* "WEIGHTED BY FOS")	106800
310 FORMAT(" ",33X,"(STANDARD OUTPUT)",T3,"I",T85,"I")	106810
320 FORMAT(" ",32X,"(GAME TRACE OUTPUT)",T3,"I",T85,"I")	106820
330 FORMAT(" ",T3,"I",T48,"I",T85,"I")	106830
340 FORMAT(" ",T3,"I",T48,"I",T85,"I",T20,F9.1,T58,F9.1)	106840
350 FORMAT(" ",T3,83(" "))	106850
360 FORMAT(" ",T3,83(" "))	106860
370 FORMAT(" ",T5,"LER",T20,10(" "),T35," ",F7.2,T53," ",	106870
* T58,10(" "),T73," ",F7.2)	106880
380 FORMAT(" ",T5,"SHEP(BLUE)",T20,10(" "),T35," ",F7.2,T53," ",	106890
* T58,10(" "),T73," ",F7.2)	106900
390 FORMAT(" ",T5,"SHEP(RED)",T20,10(" "),T35," ",F7.2,T53," ",	106910
* T58,10(" "),T73," ",F7.2)	106920
400 FORMAT(" ",T5,"SHEP",T20,10(" "),T35," ",F7.2,T53," ",	106930
* T58,10(" "),T73," ",F7.2)	106940
410 FORMAT(" ",T5,"IFR",T20,10(" "),T35," ",F7.2,T53," ",	106950
* T58,10(" "),T73," ",F7.2)	106960
420 FORMAT(" ",T5,"FER",T20,10(" "),T35," ",F7.2,T53," ",	106970
* T58,10(" "),T73," ",F7.2)	106980
END	106990

130 FORMAT("TABLE # " , I2 , " ") STATISTICS THROUGH (I " , A10)
END

197910
197920

APPENDIX N
DISTRIBUTION LIST

DISTRIBUTION LIST

<u>Organization</u>	<u>No. of Copies</u>
Commander Defense Documentation Center Cameron Station Alexandria, VA 22314	12
Commander USA Combined Arms Combat Developments Activity Fort Leavenworth, KS 66027	
ATZLCA-SW	10
ATZLCA-CA	10
Commandant Command and General Staff College ATTN: ATZLSW-TA CGSC - Library Fort Leavenworth, KS 66027	5

END

DATE
FILMED

1. 8/

DTIC