

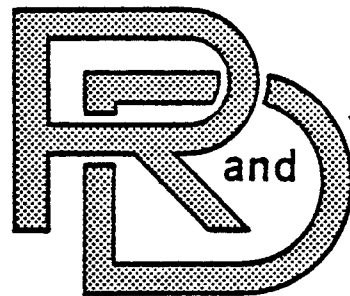
122X<sub>0</sub>

632

AD 074 535

TECHNICAL LIBRARY  
REFERENCE COPY

AD 407 45 35



TARADCOM

LABORATORY

TECHNICAL REPORT

No. 12444

20020725104

PARAMETRIC ENGINEERING  
SYSTEM DEFINITION MODEL



VOLUME II

APPENDIX C (FORTRAN LISTINGS)

July 1979

Contract DAAK30-78-C-0059

by S. Spaulding, A. Weintraub,  
F. Cioch, J. Lenz

Vector Research, Incorporated  
P.O. Box 1506  
Ann Arbor, Michigan 48106

TECHNICAL LIBRARY  
REFERENCE COPY

U.S. ARMY TANK-AUTOMOTIVE  
RESEARCH AND DEVELOPMENT COMMAND  
Warren, Michigan 48090

Approved for public release; distribution unlimited.

AN 42907

## APPENDIX C (FORTRAN LISTINGS)

## TABLE OF CONTENTS

|                                                             | <u>Page</u> |
|-------------------------------------------------------------|-------------|
| C.1. INTRODUCTION . . . . .                                 | C-3         |
| C.2. MAIN PROGRAM, LABELED COMMONS AND BLOCK DATA . . . . . | C-4         |
| Main Program . . . . .                                      | C-5         |
| Labeled Commons. . . . .                                    | C-7         |
| Block Data . . . . .                                        | C-10        |
| C.3. INITIALIZATION AND INPUT PROCESSING ROUTINES . . . . . | C-17        |
| CSCORE, Subroutine . . . . .                                | C-18        |
| INALTC, Subroutine . . . . .                                | C-21        |
| INATTR, Subroutine . . . . .                                | C-27        |
| INATR2, Subroutine . . . . .                                | C-33        |
| INFUNC, Subroutine . . . . .                                | C-38        |
| INIT, Subroutine . . . . .                                  | C-42        |
| INPROT, Subroutine . . . . .                                | C-45        |
| INSPEC, Subroutine . . . . .                                | C-52        |
| INVEHC, Subroutine . . . . .                                | C-55        |
| C.4. SOLUTION ROUTINES. . . . .                             | C-65        |
| COMPAT, Function . . . . .                                  | C-66        |
| DIMENS, Subroutine . . . . .                                | C-69        |
| FPOWER, Subroutine . . . . .                                | C-80        |
| GENVEH, Subroutine . . . . .                                | C-85        |
| MOBILE, Subroutine . . . . .                                | C-93        |
| OKCMP, Function . . . . .                                   | C-96        |
| PHIT, Function . . . . .                                    | C-100       |
| C.5. OUTPUT ROUTINES. . . . .                               | C-101       |
| OUTARR, Subroutine . . . . .                                | C-102       |
| OUTCOM, Subroutine . . . . .                                | C-105       |
| OUTENG, Subroutine . . . . .                                | C-108       |
| OUTPER, Subroutine . . . . .                                | C-112       |
| OUTVEH, Subroutine . . . . .                                | C-117       |
| TABVAL, Subroutine . . . . .                                | C-118       |

## TABLE OF CONTENTS (concluded)

|                                 | <u>Page</u> |
|---------------------------------|-------------|
| C.6. UTILITY ROUTINES . . . . . | C-123       |
| ARMULT, Subroutine . . . . .    | C-124       |
| COPY, Subroutine . . . . .      | C-125       |
| DVAL, Function . . . . .        | C-126       |
| DVAL2, Function. . . . .        | C-128       |
| ERR, Subroutine. . . . .        | C-130       |
| FVAL1, Function. . . . .        | C-142       |
| IATTR, Function. . . . .        | C-144       |
| IDVAL, Function. . . . .        | C-146       |
| IDVAL2, Function . . . . .      | C-148       |
| INITVL, Subroutine . . . . .    | C-150       |
| INLIST, Function . . . . .      | C-151       |
| IOKVAL, Function . . . . .      | C-152       |
| IRFIND, Function . . . . .      | C-155       |
| ITABLE, Function . . . . .      | C-157       |
| IVAL, Function . . . . .        | C-159       |
| IZERO, Subroutine. . . . .      | C-160       |
| LSERCH, Function . . . . .      | C-161       |
| NEWREC, Function . . . . .      | C-162       |
| NEWREL, Function . . . . .      | C-163       |
| NEWVEH, Function . . . . .      | C-166       |
| NLIST, Function. . . . .        | C-167       |
| NULIFY, Subroutine . . . . .    | C-168       |
| OKVAL, Function. . . . .        | C-169       |
| QUEUE, Subroutine. . . . .      | C-172       |
| STONAM, Subroutine . . . . .    | C-174       |
| STORE, Subroutine. . . . .      | C-175       |
| SUM, Function. . . . .          | C-176       |
| SUMCMP, Function . . . . .      | C-177       |
| SUMPD2, Function . . . . .      | C-179       |
| SUMPRD, Function . . . . .      | C-180       |
| TRANSF, Function . . . . .      | C-181       |
| UNHOOK, Subroutine . . . . .    | C-182       |
| ZERO, Subroutine . . . . .      | C-184       |
| Distribution List               |             |
| DD Form 1473                    |             |

## C.1: INTRODUCTION

This appendix contains the FORTRAN program listings for the COMPEND model. The listings are organized as follows:

- Section C-2 contains the main program, a listing of all labeled COMMON blocks (all other subprogram listings contain only the common block labels), and the BLOCKDATA subprogram.
- Section C-3 contains initialization and input processing routines.
- Section C-4 contains the main solution routines, including those used to estimate vehicle engineering and performance parameters.
- Section C-5 contains the output routines.
- Section C-6 contains utility routines.

## C.2: MAIN PROGRAM, LABELED COMMONS AND BLOCK DATA

This section contains the following listings:

- The main program.
- A consolidated listing of all labeled COMMON blocks (subprogram listings in other sections of this report contain only the block labels for COMMON blocks used in that subprogram).
- The BLOCK DATA subprogram.

# MAIN PROGRAM

COMBAT VEHICLE DEFINITION MODEL

THIS PROGRAM PRODUCES A DESCRIPTION OF AN ARMORED COMBAT VEHICLE OS A SPECIFIED CLASS WHICH MEETS THE USER'S SPECIFICATIONS. AS PART OF HIS SPECIFICATIONS THE USER MAY INDICATE VARIOUS PERFORMANCE CRITERIA THAT THE VEHICLE MUST MEET. HE MAY ALSO INDICATE VARIOUS ENGINEERING PARAMETERS DESCRIBING ITS COMPONENTS. FURTHERMORE HE MAY INDICATE WHICH SPECIFIC COMPONENTS ARE TO BE USED IN THE CONCEPT VEHICLE. THE PROGRAM USES A DESCRIPTION OF WHAT COMPONENT TYPES AND HOW MANY OF EACH ARE NEEDED TO PRODUCE A COMBAT VEHICLE IN THE SPECIFIED CLASS. IT ALSO READS IN A LIST OF ALTERNATIVE COMPONENTS OF EACH TYPE IT CAN USE IN ASSEMBLING THE CONCEPT VEHICLE. IT THEN USES A GENERATE-AND-TEST PROCEDURE TO ASSEMBLE COMPONENTS WHICH WILL PRODUCE A FINISHED VEHICLE MEETING THE USER'S SPECIFICATIONS.

FOR INPUT AND OUTPUT THE PROGRAM USES THE FOLLOWING DATA SETS:

| DSRN | I/O | FILE DESCRIPTION                         |
|------|-----|------------------------------------------|
| 1    | I   | DATA USED IN DESCRIBING FUNCTIONAL RELNS |
| 2    | I   | EXISTING VEHICLE FILE                    |
| 3    | I   | PROTOTYPE VEHICLE FILE                   |
| 4    | I   | ALTERNATIVE COMPONENT FILE               |
| 5    | I   | USER SPECIFICATION FILE                  |
| 6    | O   | ERROR MESSAGES                           |
| 7    | O   | OUTPUT FILE                              |
| 8    | O   | ECHO OF USER SPECS & DATA HEADERS        |

LOGICAL ALT V  
DATA ALT V/. FALSE./PERFORM INITIALIZATIONS  
CALL INIT

READ & STORE FUNCTIONAL RELATION DATA FILE  
CALL INFUNC

## MAIN PROGRAM

```

158 C
159 C READ & STORE PROTOTYPE VEHICLE DEFINITION FILE
160   CALL INPROT
161 C
162 C READ & STORE ALTERNATIVE COMPONENT FILE
163   CALL INALTC
164 C
165 C READ & STORE USER SPECIFICATIONS IMAGING EXTERNAL FORM
166   CALL INSPEC
167 C
168 C EXTRACT ANY INFORMATION NEEDED FROM EXISTING VEHICLE FILE TO
169   PRODUCE FINAL INTERNAL DESCRIPTION OF USER SPECIFICATIONS
170   CALL INVEHC
171 C
172 C GENERATE "BEST" CONCEPT VEHICLE USING A BACKTRACK PROGRAMMING
173   APPROACH
174   CALL GENVEH
175 C
176 C OUTPUT A DESCRIPTION OF THIS CONCEPT VEHICLE
177   CALL OUTVEH
178 C
179   STOP
180   END

```

## COMMON BLOCKS

```

COMMON /VEH/ VEHICLE(72,3), HULL(48,9), TURRET(42,9), MAINGN(44,9),
+   MACGUN(40,10), MISGUN(40,7), ADGUN(40,6), RANGER(28,7),
+   SENSOR(32,9), STABLE(30,6), GUNCTL(28,9), AMMO(30,30),
+   AMMOC(26,9), ENGINE(40,8), TRANSM(32,8), FINLDR(28,1),
+   ROADWH(36,10), SPRING(30,10), TRACK(34,8), SKIRT(28,8),
+   FUEL(28,10), FUEL(26,9), CREW(26,5), CARGO(26,5), ELECTR(26,6),
+   COMMO(28,9), FIREX(28,10), ENVIRC(26,9), DIAGNS(26,5),
+   SIGSUP(26,24), SMOKEG(26,10), EWSYS(26,10), RELN(3,50),
+   ISPECV, ICANDV, IALTV
DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
+   FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
+   ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),
+   IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),
+   IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
+   IFUEL(28,10), IFUEL(26,9), ICREW(26,5), ICARGO(26,5),
+   IFLECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),
+   IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
+   IRELN(3,50), VFILE(1), IVFILE(1)
EQUIVALENCE (VEFCLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
+   ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),
+   (MISGUN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
+   IRANGR(1)), (SENSOR(1), ISENSR(1)), (STABLE(1), ISTABL(1)),
+   (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
+   IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANSM(1), ITRANS(1)),
+   (FINLDR(1), IFINDR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),
+   ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),
+   (FUEL(1), IFUEL(1)), (FUEL(1), IFUEL(1)), (CREW(1),
+   ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
+   (COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),
+   IENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
+   (SMOKEG(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))
EQUIVALENCE (VEFCLE(1), VFILE(1), IVFILE(1))
COMMON/VPROTO/ FLDNAM(1049), RECNAM(33), DEFAULT(1049),
+   CMPCOD(31), ICMPTPT(31), ICMPTPK(31), IFILPT(33), IFLDPT(34),
+   IFLDTP(1049), IFLDVL(1049), LSTOUT, LSTREC(33), MAXNUM(31),
+   MINNUM(31), NALICS(31), NFLDS(33), NRECS(33), OUTNAM(7,450),

```

## COMMON BLOCKS

```

+   UNITS(1049), ICMATR, IEPARS, IPPARS, IDOUTF(450)
+   REAL*8 FLDNAM, RECNAM, DFAULT
+   COMMON /JPARS/
+   JACCEL, JACCUR, JACOST, JADGUN, JAI, JAMMO, JAMMOC, JAMOTP,
+   JANGLE, JARMTP, JBETAF, JBETAR, JBI, JBORE, JBRAKE, JCALIB,
+   JCARGO, JCMPLX, JCOMMO, JCOOLR, JCOVER, JCREW, JCTLSY, JDAMP,
+   JOELD, JDIAGN, JDPRES, JDTNG, JDWID, JDWUP, JDI, JD2, JD3,
+   JECOST, JEFFIC, JEPWR, JELEC, JELEVN, JEMOB, JENGIN, JENVR,
+   JEPROT, JERAMD, JEWSYS, JFCAP, JFGAL, JFINDR, JFIREX, JFRATE,
+   JFUEL, JFUELC, JFUELE, JFUELN, JGAMD, JGAMU, JGHPTN, JGPRES,
+   JGUINE, JHMCVE, JHP, JHULL, JKI, JLOAD, JLOC, JMACHG, JMAG,
+   JMATNG, JMANUF, JMATER, JMATUR, JMINWT, JMISL, JMODEL, JMUZLE,
+   JMXSPD, JNAT, JNCRFW, JNFWDG, JNRDS, JNRDWH, JNRETN, JNRVSG,
+   JNUM, JOCOST, JOCIAM, JOHT, JPERF, JPERSN, JPHITM, JPHITS,
+   JPK, JPSLOP, JRANGA, JRANGE, JRCOST, JRELIB, JRNGSY, JRADM,
+   JRTIME, JSADJ, JSNSR, JSHPIN, JSIGSP, JSKHI, JSKIRT, JSKTHK,
+   JSLOPE, JSMCKE, JSPD1, JSPD2, JSPRNG, JSTBSY, JSTEMP, JSTEM2
+   COMMON /JPARS1/
+   JSTIME, JSUSP, JT, JTBLN, JTFIRS, JTFIR1, JTHFL, JTHFU,
+   JTONFT, JTRACK, JTRAD, JTRANS, JTRATE, JTRNRQ, JTIB,
+   JTTF, JTTS, JTTU, JTURET, JTYPE, JTYPEE, JVACOS, JVCMPX, JVHT,
+   JVLEN, JVMATR, JVMBF, JVOCOS, JVQL, JVOPHR, JVVOL, JVWID,
+   JVWT, JWDEPN, JWDEPP, JWHTRV, JWT, JXH, JXTP, JX1, JX2,
+   JX3, JX4, JX5, JX6, JX7, JX8, JYEAR, JYGC, JYH, JY2, JY20, JY21,
+   JY22, JY23, JY3, JY4, JY5, JY6, JY7, JY8, JY9, JZH, JZ1, JZ10,
+   JZ11, JZ12, JZ13, JZ14, JZ15, JZ2, JZ3, JZ4, JZ5, JZ7, JZ8, JZ9,
+   JID, JNEXT, JRELOP, JSCORE, JVALUE, JKEY, JLT, JPENTR,
+   JSIGMA, JSIGM2, J30SPD
+   COMMON /NPARS/
+   NARMC, NARMC2, NARMTP, NADIMS, NARMTH, NARRI,
+   NARLAB, NARRS, NASPEC, NCCMAT, NCSPCS, NCMPTP,
+   NCMPT1, NCMP5, NCMPZL, NDAYNT, NDETP, NDISPR, NESPCS, NEVALC,
+   NFLDRG, NGEAR, NGRADE,
+   NWHTRV, NROWHL, NORDT2, NOROTP, NOUT, NOUTWD, NPSPCS,
+   NRRANGE, NRECTP, NRCTP1, NRELOP, NSHPTN, NSTABL, NSMGT, NTERIP,
+   NTKLEN, NENGTP, NTYPES, NUTRNI, NUTRNS, NVEHWT, NXY

```

## COMMON BLOCKS

```

COMMON /OUTPUT/ XNAT(32),XMANUF(32),XMODEL(32),NUMBER(32),
+ WEIGHT(32),PCWT(32),VOLUME(32),PCVOL(32),DWNUP(32),
+ MATUR(32),ICMPLX(32),RELIAB(32),INDX(32),NPROD(32),
+ RDTIME(32),KOUT(32),FMCWGT,DF(8,31)
LOGICAL*1 DF
REAL*8 XNAT,XMANUF,XMODEL
COMMON /AUX/ BL8,DASHES,BLS,IRELTP(15),ITYPES(4),MULT,NULL,
+ PLUS,RELNAM(15),SMINUS,TUNITS(3,50)
REAL*8 BL8,DASHES
EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
COMMON /ARRAYS/ DETRNG(3,2,9),NATION(160),MANUF(160),
+ MODEL(160),PHITM(3,2,3,3),PHITS(3,2,3,3),
+ PK(3,2,3,3),PPENET(3,2,3,3),TRNRAD(4,3),SIGMA(2,3,3,16),
+ SIGMA2(2,3,3,16),ARRLAB(5,75),IARDEF(9,28),IARDF2(6,13)
REAL*8 MANUF,MODEL,NATION
DIMENSION ARRAY(1)
EQUIVALENCE (ARRAY(1),DETRNG(1))
COMMON /JPARS2/ JARSIZ,JARLST,JARNUM,JARRPT,
+ JNAME,JFLDRC,JLEVPT,JNLEVL,JRECTP
COMMON /SPECS/ ATRENG(200),ATRP(34),VEHCAT,VEHCMP(31),
+ VEHENG(200),VFHPR(34),VEHWT,VEHEVL(5),NUMCMP(31),
+ TYPCMP(31),INDCMP(31),LSTCMP,TYPENG(200),RELENG(200),
+ MULENG(200),VLENG(200),ADDENG(200),VL2ENG(200),INDENG(200),
+ LSTENG,RELPER(34),MULPER(34),VL1PER(34),ADOPER(34),
+ VL2PER(34),LSTPER
REAL*8 ATRENG,ATRP,VEHCAT,VEHCMP,VEHENG,VEHPR
COMMON /SPECS2/ VEHS(10),ICMPS(31),ICOLS(10),IRECS(6,250),
+ JCOL,JDONE,JNXT,JSPEC,JTYP,JVEH,LSTVEH,LVEH,NIRECI,NVEHS,NVR
REAL*8 VEHS
COMMON /DATA/ ACCEL(4,5),CMPWTS(31),DENSITY(8),
+ MAXWHL,MINWHL,PHULL,PHULLP(4,3,2,2,4),PKDAT(3,3,2,3),
+ PUPRRD(4,3,2,2,4),RARMTH(4),RLTMAX,RLTMIN,
+ RSHPTN(4),RVEHWT(5),RTKLEN(4),RWHTRV(4),
+ SIGMVF(2,3,3,3),SIGMVT(2,3,3,2),
+ SLOPE(4,5),SPEED(4,2),VRIDE(4,5,4,3,2),WIDMAX,XMPG(4,5,4)
COMMON /DATA2/ RDISPR(3)

```

## BLOCK DATA

## BLOCK DATA

```

COMMON /VEH/ VEHICLE(72,3),HULL(48,9),TURRET(42,9),MAINING(44,9),
+   MACGUN(40,10),MISGUN(40,7),ADGUN(40,6),RANGER(28,7),
+   SENSOR(32,9),STABLE(30,6),GUNCTL(28,9),AMMO(30,30),
+   AMMOC(26,9),ENGINE(40,8),TRANSM(32,8),FINLDR(28,10),
+   ROADWH(36,10),SPRING(30,10),TRACK(34,8),SKIRT(28,8),
+   FUEL(28,10),FUELC(26,9),CREW(26,5),CARGO(26,5),ELECTR(26,5),
+   COMMD(28,9),FIREX(28,10),ENVIRC(26,9),DIAGNS(26,5),
+   SIGSUP(26,24),SMOKEG(26,10),EWSYS(26,10),RELN(3,50),
+   ISPECV,ICANDV,IALTV
DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
+   FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
+   ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
+   IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
+   IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
+   IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
+   IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
+   IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IOWSYS(26,10),
+   IRELN(3,50),VFILE(1),IVFILE(1)
EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
+   ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
+   (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
+   IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
+   (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
+   IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
+   (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
+   ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
+   (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
+   ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
+   (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
+   IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
+   (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IOWSYS(1)),(RELN(1),IRELN(1))
EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
COMMON/VPROTO/ FLDNAM(1049),RECNAM(33),DFALT(1049),
+   CMPCOD(31),ICMPPT(31),ICMPRK(31),IFILPT(33),IFLDPT(34),

```

## BLOCK DATA

```

+ IFLDTP(1049),IFLDVL(1049),LSTOUT,LSTREC(33),MAXNUM(31),
+ MINNUM(31),NALTCS(31),NFLDS(33),NRECS(33),OUTNAM(7,450),
+ UNITS(1049),ICMATR,IEPARS,IPPARS,IOUTF(450)
+ REAL*8 FLDNAM,RECNAME,DEFAULT
+ COMMON /JPARS/
+ JACCEL,JACCUR,JACOST,JADGUN,JAI,JAMMO,JAMMOC,JAMOP,
+ JANGLE,JARMT,P,JBTAF,JBTAR,JBI,JBORE,JBRAKE,JCALIB,
+ JCARGO,JCMPLX,JCCMC,JCOLR,JCOVER,JCREW,JCTLSY,JDAMP,
+ JDELD,JDIAGN,JDPRES,JDRNG,JDWID,JDNUP,JDI,JD2,JD3,
+ JECOST,JEFFIC,JEFPWR,JELC,JELEVN,JEMOB,JENGIN,JENVR,
+ JEPROT,JERAMD,JEWYS,JFCAP,JFGAL,JFINDR,JFIREX,JFRATE,
+ JFUEL,JFUELC,JFUELE,JFUELN,JGAMD,JGAMU,JGHPTN,JGPRES,
+ JGUIDE,JHMCVE,JHP,JHULL,JKI,JLOAD,JLOC,JMACHG,JMAG,
+ JMAING,JMANUF,JMATER,JMATUR,JMINWT,JMISL,JMODEL,JMUZLE,
+ JMXSPD,JNAT,JNCREW,JNFWDG,JNRDS,JNRDWH,JNRETN,JNRVSG,
+ JNUM,JOCOST,JOCIAM,JOHT,JPERF,JPERSON,JPHITM,JPHITS,
+ JPK,JPSLOP,JRANGA,JRANGE,JRCOST,JRELIB,JRNGSY,JROADW,
+ JRTIME,JSACJ,JSENSR,JSHPTN,JSIGSP,JSKHT,JSKIRT,JSKTHK,
+ JSLOPE,JSMOKE,JSPD1,JSPD2,JSPRNG,JSTBSY,JSTEMP,JSTEM2
+ COMMON /JPARS1/
+ JSTIME,JSUSP,JT,JTBLEN,JTFIRS,JTFIR1,JTHFL,JTHFU,
+ JTONFT,JTRACK,JRAD,JTRANS,JTRATE,JTRNRQ,JTIB,
+ JTTF,JTTS,JTTU,JTURET,JTYPE,JTYPEE,JVACOS,JVCMXP,JVHT,
+ JVLEN,JVMATR,JVMBF,JVOCOS,JVOL,JVOPHR,JVOL,JVVID,
+ JVWT,JWDEPN,JWDEPP,JWHTRV,JWT,JXH,JXTP,JX1,JX2,
+ JX3,JX4,JX5,JX6,JX7,JX8,JYEAR,JYGC,JYH,JY2,JY20,JY21,
+ JY22,JY23,JY3,JY4,JY5,JY6,JY7,JY8,JY9,JZH,JZ1,JZ10,
+ JZ11,JZ12,JZ13,JZ14,JZ15,JZ2,JZ3,JZ4,JZ5,JZ7,JZ8,JZ9,
+ JID,JNEXT,JRELCP,JSCORE,JVALUE,JKEY,JLT,JPENTR,
+ JSIGMA,JSIGM2,J30SPD
+ COMMON /NPARS/
+ NARMC,NARMC2,NARMT,P,NADIMS,NARMT,H,NARRI,
+ NARLAB,NARRS,NASPEC,NCOMAT,NCSPCS,NCMPTP,
+ NCMPT1,NCMPS,NCMPZL,NDAYNT,NDETP,NDISPR,NESPCS,NEVALC,
+ NFLOPC,NGEAR,NGRADE,
+ NWHTRV,NRDWHL,NRODT2,NORDTP,NOUT,NOUTWD,NPSPCS,

```

## BLOCK DATA

```

+   N RANGE,NRECTP,NRCTPI,NRELOP,NSHPTN,NSTABL,NSMTGT,VTERTP,
+   NTKLEN,NENGT,NTYPES,NUTRNI,NUTRNS,NVEHWT,NXY
COMMON /OUTPUT/ XNAT(32),XMANUF(32),XMODEL(32),NUMBER(32),
+   WEIGHT(32),PCWT(32),VOLUME(32),PCVOL(32),DWNUP(32),
+   MATUP(32),ICMPLX(32),RELIAB(32),INDX(32),NPRND(32),
+   RDTIME(32),KOUT(32),FMCWGT,DF(8,31)
LOGICAL*1 DF
REAL*8 XNAT,XMANUF,XMODEL
COMMON /AUX/ BL8,CASHES,BLS,IRELTP(15),ITYPES(4),MULT,NULL,
+   PLUS,RELNAM(15),SMINUS,TUNITS(3,5))
REAL*8 BL8,CASHES
EQUIVALENCE (FNULL,NULL),(IRLS,BLS)
COMMON /ARRAYS/ DETRNG(3,2,9),NATION(160),MANUF(160),
+   MODEL(160),PHITM(3,2,3,3),PHITS(3,2,3,3),
+   PK(3,2,3,3),PPENET(3,2,3,3),TRNRAD(4,3),SIGMA(2,3,3,16),
+   SIGMA2(2,3,3,16),APRLAB(5,75),IARDEF(9,28),IARDEF2(6,13)
REAL*8 MANUF,MODEL,NATION
DIMENSION ARRAY(1)
EQUIVALENCE (APRAY(1),DETRNG(1))
COMMON /JPARS2/ JARSIZ,JARLST,JARNUM,JARRPT,
+   JNAME,JFLCRC,JLEVPT,JNLEVL,JRECTP
COMMON /SPECS2/ VEHS(10),ICMPS(31),ICOLS(10),IRECS(6,250),
+   JCOL,JDONE,JNXT,JSPEC,JTYP,JVEH,LSTVEH,LVEH,NIRECI,NVEHS,NVR
REAL*8 VEHS
COMMON /DATA/ ACCEL(4,5),CMPWTS(31),DENSITY(8),
+   MAXWHL,MINWHL,PHHULL,PHULLP(4,3,2,2,4),PKDAT(3,3,2,3),
+   PTURRP(4,3,2,2,4),RARMTH(4),RLTMAX,RLTMIN,
+   PSHTPTN(4),RVEHWT(5),RTKLEN(4),RWHTRV(4),
+   SIGMVF(2,3,3,3),SIGMVT(2,3,3,2),
+   SLOPE(4,5),SPEED(4,2),VRIDE(4,5,4,3,2),WIDMAX,XMPS(4,5,4)
COMMON /DATA2/ ROISPR(3)
DIMENSION KARDEF(9,14)
EQUIVALENCE (KARDEF(1,1),IARDEF(1,14))
DATA

```

C

C

## BLOCK DATA

|   |                                                             |
|---|-------------------------------------------------------------|
| + | JACCEL/52/, JACCUR/26/, JACOST/4/, JADGUN/6/, JAI/25/,      |
| + | JAMMO/11/, JAMMO/12/, JAMOT/28/, JANGLE/27/, JARMP/28/,     |
| + | JRETAF/33/, JBETAR/34/, JBI/25/, JBORE/29/, JBRAKE/56/,     |
| + | JCALIB/27/, JCARGO/23/, JCMLX/9/, JCOMMO/25/, JCOOLR/27/,   |
| + | JCOVER/34/, JCREW/22/, JCILSY/10/, JDAMP/27/, JDELD/38/,    |
| + | JDIAGN/28/, JDPRES/37/, JDIRNG/29/, JDWID/45/, JDOWNUP/11/, |
| + | JD1/29/, JD2/28/, JD3/27/, JECOST/22/, JEFFIC/26/,          |
| + | JFEPWR/18/, JELEC/24/, JELEVN/36/, JEMOB/19/, JENGIN/13/,   |
| + | JENVR/27/, JEPRCT/20/, JERAMD/21/, JEWSYS/31/, JFCAP/26/,   |
| + | JFGAL/26/, JFINDR/15/, JFIREX/26/, JFRATE/33/, JFUEL/20/,   |
| + | JFUELC/21/, JFUELE/29/, JFUELN/28/, JGAMD/37/, JGAMJ/36/,   |
| + | JGHPTN/41/, JGPRES/43/, JGUIDE/28/, JHMOVE/35/, JHP/26/,    |
| + | JHULL/1/, JK1/31/, JLOAD/30/, JLOC/17/, JMACHG/4/,          |
| + | JMAG/28/, JMAING/3/, JMANUF/14/, JMATER/26/, JMATUR/8/,     |
| + | JMINWT/38/, JMISL/5/, JMODEL/15/, JMUZLE/25/, JMXSPD/48/,   |
| + | JNAT/13/, JNCREW/25/, JNFWDG/27/, JNRDS/26/, JNRDWH/25/,    |
| + | JNRETN/26/, JNRVSG/28/, JNUM/12/, JOCOST/5/, JODIAM/42/,    |
| + | JQHT/44/, JPERF/26/, JPERSN/26/, JPHITM/39/, JPHITS/38/,    |
| + | JPK/40/, JPSLOP/57/, JRANGA/50/, JRANGE/51/, JRCOST/6/      |
|   | DATA                                                        |
| + | JREL18/10/, JRNGSY/7/, JROADW/16/, JR TIME/7/, JSADJ/27/,   |
| + | JSENSR/8/, JSHPTN/42/, JSIGSP/29/, JSKHT/26/, JSKIRI/19/,   |
| + | JSKTHK/27/, JSLOPE/53/, JSMOKE/30/, JSPD1/58/, JSPD2/59/,   |
| + | JSPRNG/28/, JSTBSY/9/, JSTEMP/32/, JSTEM2/33/, JSTIME/31/,  |
| + | JSUSP/17/, JT/28/, JTBLN/26/, JTFIRS/32/,                   |
| + | JTFIR1/31/, JTHFL/32/, JTHFU/31/, JTONFI/69/, JTRACK/18/,   |
| + | JTRAD/55/, JTRANS/14/, JTRATE/54/, JTRNRQ/30/,              |
| + | JTTB/39/, JTTF/26/, JTTS/37/, JTU/38/, JTURET/2/,           |
| + | JTYPE/25/, JTYPEE/26/, JVACOS/65/, JVCMPX/62/, JVHT/35/,    |
| + | JVLFN/37/, JVMATR/61/, JVMMBF/63/, JVOCONS/66/, JVOL/3/,    |
| + | JVOPHR/64/, JVVOL/34/, JVVWID/36/, JVWT/33/, JWDEPN/46/,    |
| + | JWDEPP/47/, JWHTRV/25/, JWT/2/, JXH/27/,                    |
| + | JXTP/27/, JX1/39/, JX2/30/, JX3/29/, JX4/27/, JX5/35/,      |
| + | JX6/38/, JX7/29/, JX8/27/, JYEAR/16/, JYGC/68/, JYH/26/,    |
| + | JY2/39/, JY20/40/, JY21/41/, JY22/34/, JY23/35/, JY3/40/,   |
| + | JY4/41/, JY5/42/, JY6/34/, JY7/33/, JY8/31/, JY9/30/,       |

[illegible]

## BLOCK DATA

```

+ 1,39,2,0,'STAT',,VS,,MOVI,,NG T,'GT',
+ 1,39,3,0,'ORDN',,ANCE,,TYP,,E,, ,
+ 1,40,3,0,'RANG',,E,, , , ,
+ 1,40,2,0,'ASPE',,CT A,'NGLE',, , ,
+ 1,40,3,0,'CRDN',,ANCE,,TYP,,E,, ,
+ 1,55,4,0,'GEAR',, , , , ,
+ 1,60,3,0,'RANG',,E,, , , ,
+ 1,60,2,0,'ASPE',,CT A,'NGLE',, , ,
+ 1,60,2,0,'ORDN',,ANCE,,TYP,,E,, //
DATA KARDEF/
+ 4,34,2,0,'X CR',,Y S,'IGMA',, , ,
+ 4,34,3,0,'RANG',,E,, , , ,
+ 4,34,3,0,'CRDN',,ANCE,,TYP,,E,, ,
+ 6,34,2,0,'X OR',,Y S,'IGMA',, , ,
+ 6,34,3,0,'RANG',,E,, , , ,
+ 6,34,3,0,'CRDN',,ANCE,,TYP,,E,, ,
+ 8,27,2,0,'X OR',,Y S,'IGMA',, , ,
+ 8,27,3,0,'RANG',,E,, , , ,
+ 8,27,3,0,'CRDN',,ANCE,,TYP,,E,, ,
+ 9,29,3,0,'DETE',,CTIO,'N CA,'TEGO,'RY',
+ 9,29,2,0,'DAY',,VS N,'IGHT',, , ,
+ 11,27,2,0,'X CR',,Y S,'IGMA',, , ,
+ 11,27,3,0,'RANG',,E,, , , ,
+ 11,27,3,0,'ORDN',,ANCE,,TYP,,E,, //
DATA IARDF2/9,25,6,0,0,9,
+ 0,13,2,0,0,160,
+ 0,14,2,0,0,160,
+ 0,15,2,0,0,160,
+ 1,39,18,0,0,3,
+ 1,38,18,0,0,3,
+ 1,40,18,0,0,3,
+ 1,60,18,0,0,3,
+ 1,55,4,0,0,3,
+ 4,34,18,0,0,8,
+ 6,34,18,0,0,8,
+ 8,27,18,0,0,8,

```

## BLOCK DATA

```

+ 11,27,18,0,0,8/
+ DATA JNXT/1/,JTP/2/,JSPEC/3/,JVEH/4/,JCOL/5/,
+ NVEHS/10/,NVR/250/
+ DATA ICMPRK/3,5,7,8,9, 10,11,13,14,15, 20,16,17,18,19,
+ 22,23,21,12,4, 6,1,2,24,25, 26,27,28,29,30, 31/
+ DATA TUNITS/'KM','M',1000., 'FT','IN',12., 'M','FT',3.28,
+ 'M','IN',39.37, 'IN','MM',25.4, 'RAD','DEG',57.3,
+ 'MIN','SEC',60., 'HR','MIN',60., 'HR','SEC',3600.,
+ 'TON','LB',2000., 'TON','KG',907., 'KG','LB',2.2,
+ 'GAL','L',3.785, 'DEGC','DEGF',1.8, 'FT3','IN3',1728.,
+ 'M/S','FT/S',3.28, 'MPH','KPH',1.61, 'M/S','MPH',2.24,
+ 'MPH','FT/S',1.47/
+ DATA FMCWGT/0.1/
+ END

```

### C.3: INITIALIZATION AND INPUT PROCESSING ROUTINES

This section contains listings of initialization and input processing routines. See section C-2 for listings of labeled COMMON blocks referred to in these listings.

## INPUT ROUTINES - CSCORE SUBROUTINE

```

1  ISN
2  SUBROUTINE CSCORE
3  ROUTINE TO COMPUTE THE SCORE OF THE VARIOUS ALTERNATIVE COMPONENTS
4  AND TO ORDER THE COMPONENTS OF EACH TYPE IN DECREASING ORDER BY SCORE.
5
6  COMMON /VEH/
7  DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
8  + FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
9  + ISENSR(22,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),
10 + IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),
11 + IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
12 + IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,9),
13 + IELECT(26,6), ICCMMO(28,9), IFIREX(28,10), IENVIR(26,9),
14 + IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,1), IEWSYS(26,13),
15 + IRELN(3,50), VFILE(1), IVFILE(1)
16 EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
17 + ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),
18 + (MISGUN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
19 + IRANGR(1)), (SENSOR(1), ISENSR(1)), (STABLE(1), ISTABL(1)),
20 + (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
21 + IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANSM(1), ITRANS(1)),
22 + (FINLDR(1), IFINDR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),
23 + ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),
24 + (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
25 + ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
26 + (COMMO(1), ICCMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),
27 + IENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
28 + (SMOKEG(1), ISMOKE(1)), (IEWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))
29 EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))
30 COMMON /VPROTD/
31 REAL*8 FLDNAM, RECNAM, DFAULT
32 COMMON /JPARS/
33 COMMON /JPARS1/
34 COMMON /NPARS/
35 COMMON /AUX/

```

```

      INPUT ROUTINES - CSCORE SUBROUTINE

      REAL*8 BL8,DASHES
      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      COMMON /SPECS/
      REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER

      C
      DIMENSION IRS(50),NEXT(50),SCORES(50)

      C
      C
      C ITERATE OVER COMPONENT TYPES
      DO 50 ICOMP=1,NCMPTP

      C
      C GET ATTRIBUTES OF COMPONENT
      NF = NFELDS(ICOMP+1)
      IOFF = IFILPT(ICOMP+1)

      C
      C ITERATE OVER COMPONENTS OF THAT TYPE
      IREC = ICOMPPT(ICOMP)
      LIST = NULL
      J = 0
      CALL NULIFY(NEXT,50)
      CALL ZERO(SCORES,50)
      IF (IREC .EQ. NULL) GO TO 30
      IOFFR = IOFF + NF*(IREC-1) - 1
      SCORE = SUMPRC(VFILE(IOFFR+JEPWR),VEHEVL,5)
      VFILE(IOFFR+JSCORE) = SCORE

      C
      C INSERT NEW COMPONENT IN LIST IN DECREASING ORDER
      J = J + 1
      IRS(J) = IREC
      SCORES(J) = SCORE
      IF (J.NE. 1) GO TO 14
      NEXT(J) = LIST
      LIST = J
      GO TO 20

      C
      IF (SCORE .GT. SCORES(LIST)) GO TO 12
      K = LIST
      KP = K

```

```

      INPUT ROUTINES -  CSCORE SUBROUTINE

      K = NEXT(K)
      IF (K .NE. NULL .AND. SCORE .LE. SCORES(K)) GO TO 16
      NEXT(J) = K
      NEXT(KP) = J
      IREC = IVFILE(IOFFR+JNEXT)
      GO TO 10

20  C
44  C STRING TOGETHER COMPONENTS IN DECREASING ORDER BY THEIR SCORES
      K = LIST
      ICMPT(ICMP) = LIST
      IF (K .EQ. NULL) GO TO 50
      IOFFR = IOFF + NF*(K - 1) - 1
      IVFILE(IOFFR+JSCORE) = NEXT(K)
      K = NEXT(K)
      GO TO 32

      C
50  CONTINUE
      RETURN
      END
54

```

## INPUT ROUTINES - INALTC SUBROUTINE

```

1 SN
1 SUBROUTINE INALTC
C
C ROUTINE FOR READING AND STORING ALTERNATIVE COMPONENTS.
C
2 COMMON /VEH/
3 DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
+ FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
+ ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
+ IAMMOC(26,9),IENGINE(40,8),ITRANS(32,8),IFINDR(28,10),
+ IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
+ IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
+ IELECT(26,6),ICOMMD(28,9),IFIREX(28,10),IENVIR(26,9),
+ IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
+ IRELN(3,50),VFILE(1),IVFILE(1)
4 EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
+ ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
+ (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
+ IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
+ (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMJC(1),
+ IAMMOC(1)),(ENGINE(1),IENGINE(1)),(TRANSM(1),ITRANS(1)),
+ (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
+ ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
+ (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
+ ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
+ (COMMD(1),ICOMMD(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
+ IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
+ (SMOKEG(1),ISMOKE(1)),(IEWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
5 EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
6 COMMON /VPROTO/
7 REAL*8 FLDNAM,RECNAM,DEFAULT
8 COMMON /JPARS/
9 COMMON /JPARS1/
10 COMMON /NPARS/
11 COMMON /AUX/
12 REAL*8 BL8,DASHES

```

```

      INPUT ROUTINES -  INALTC SUBROUTINE

      EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
      COMMON /ARRAYS/
      REAL*8 MANUF, MODEL, NATION
      DIMENSION ARRAY(1)
      EQUIVALENCE (ARRAY(1), DETRNG(1))
      COMMON /JPARS2/
      COMMON /SPECS/
      REAL*8 ATRENG, ATRPER, VEHCAT, VEHCOMP, VEHENG, VEHPER

      C
      REAL*8 V, CMP, CMP2, ATTR
      DIMENSION INT(12), IRECS2(12), VALS(12), IVALS(12), CVALS(2,12)

      C
      C READ VEHICLE CATEGORY RECORD
      READ (4,100) V
      FORMAT (A8)
      IF (V .NE. VEHCAT) CALL ERR(4, 'INALTC', V, VEHCAT, 3)
      CMP2 = BL8

      C
      C READ COMPONENT HEADER
      READ (4,105, END=90) CMP, (INT(I), I=1, NCMPLZL)
      FORMAT (A8, 2X, I2(8X, I2))
      C CHECK WHETHER HAVE ENCOUNTERED A NEW TYPE OF COMPONENT
      IF (CMP .EQ. CMP2) GO TO 5
      N = 0
      CMP2 = CMP

      C
      C DETERMINE HOW MANY NON-ZERO ALTERNATIVES THERE ARE
      DO 10 I=1, NCMPLZL
      IF (INT(I) .EQ. 0) GO TO 15
      IF (INT(I) .NE. N+1) CALL ERR(5, 'INALTC', CMP, N+1, INT(I))
      CONTINUE
      N = N + NCMPLZL
      NA = NCMPLZL
      GO TO 20

      C
      15 I = I - 1

```

## INPUT ROUTINES - INALTC SUBROUTINE

```

15N  I SN
40    N = N + I
41    NA = I

C
C  IDENTIFY RECORD AND COMPONENT TYPES; STORE NUMBER OF ALTERNATIVES
C  OF THIS TYPE COMPONENT
20    DO 25 IRECTP=1,NRECTP
42      IF (CMP.EQ.RECNAM(IRECTP)) GO TO 27
43      CONTINUE
44    CALL ERR(9,'INALTC',CMP,0,0)
45    ICMP = IRECTP - 1
46    IF (ICMP.GE.1) NALTC$(ICMP) = N
47

C
C  DETERMINE OFFSET IN FILE IN WHICH RECORDS STORED
48    IOFF = IFILPT(IRECTP)
C  DETERMINE NUMBER OF FIELDS PER RECORD IN THIS FILE
49    NF = NFLOS(IRECTP)
C  GET NA NEW COMPONENT RECORDS AND QUEUE THEM ONTO THE CURRENT
C  LIST
50    DO 30 IREC=1,NA
51      IR = NEWREC(IRECTP)
52      CALL STORE(INT(IREC),VFILE(IOFF),NF,JID,IR)
53      IRECS2(IREC) = IR
54      CALL QUEUE(IR,ICMPPT(IRECTP-1),IRECTP)
55    CONTINUE
30

C
C  READ COMPONENT ATTRIBUTES
56    READ (4,110,END=90) ATTR
57    FORMAT (2X,A8)
C  IS THIS THE END OF THIS SET OF ATTRIBUTES ?
58    IF (ATTR.EQ.BL8.OR.ATTR.EQ.DASHES) GO TO 2
C  FIND FIELD INDEX AND RECORD TYPE
59    IFLD = IATTR(ATTR,CMP,IRECTP)
C  IF ATTRIBUTE NAME IS UNRECOGNIZED, HAVE AN ERROR
60    IF (IFLD.NE.NULL) GO TO 45
61    CALL ERR(6,'INALTC',CMP,ATTR,0)
62    GO TO 40

```

## INPUT ROUTINES - INALTC SUBROUTINE

```

ISN
C
C DETERMINE TYPE OF ATTRIBUTE
63 KFLD = IFLOPT(IRECTP) + IFLD
64 ITP = IFLOPT(KFLD)
C BACKSPACE RECORD AND READ VALUES USING FORMAT APPROPRIATE TO TYPE
65 BACKSPACE 4
66 DO 50 I=1,NTYPES
67 IF (ITYP.EQ. ITPES(I)) GO TO 55
68 CONTINUE
69 CALL ERR(10,'INALTC',CMP,ATTR,ITYP)
70 GO TO 2
C
C BRANCH APPROPRIATE TO TYPE
71 GO TO (60,65,70,80),I
C
C READ & STORE REAL VALUES
60 READ (4,115) (VALS(J),J=1,NA)
73 FORMAT (12X,12(F8.0,2X))
74 BACKSPACE 4
75 READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
76 DO 61 J = 1,NA
77 IF (CVALS(1,J) .EQ. BLS .AND. CVALS(2,J) .EQ. BLS)
+ VALS(J) = FNULL
61 CONTINUE
79 DO 62 J=1,NA
80 CALL STORE(VALS(J),VFILE(IOFF),NF,IFLD,IRES2(J))
81 CONTINUE
82 GO TO 40
C
C READ AND STORE INTEGER VALUES
65 READ (4,120) (IVALS(J),J=1,NA)
84 FORMAT (12X,12(I8,2X))
85 BACKSPACE 4
86 READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
87 DO 67 J = 1,NA
88 IF (CVALS(1,J) .EQ. BLS .AND. CVALS(2,J) .EQ. BLS)

```

```

      INPUT ROUTINES - INALTC SUBROUTINE

      + IVALS(J) = NULL
      67 CONTINUE
      90 DO 68 J=1,NA
      91 CALL STORE(IVALS(J),VFILE(IOFF),NF,IFLD,IRECS2(J))
      92 CONTINUE
      93 GO TO 40

      C READ & STORE AN ARRAY OF REAL VALUES
      C
      C DETERMINE WHICH ARRAY FILE INFORMATION WILL BE STORED IN
      70 DO 72 IARR=1,NARRS
      94 IF (IARDF2(JRECTP,IARR) .EQ. IRECTP .AND. IARDF2(JFLDRC,IARR)
      95 .EQ. IFLD) GO TO 73
      96 CONTINUE
      97 CALL ERR(16,'INALTC',IRECTP,IFLD,ITYP)

      C NUMBER OF ELEMENTS IN ARRAY
      73 NV = IARDF2(JARSIZ,IARR)
      C OFFSET IN FILE CONTAINING ARRAYS
      99 IOFFAR = IARDF2(JARRPT,IARR)
      C GET NA VACANT ARRAYS OF TYPE IARR & STORE POINTERS TO THEM FROM THE
      C CURRENT RECORD
      100 DO 75 J=1,NA
      101 K = IARDF2(JARLST,IARR) + 1
      102 IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INALTC',
      + ATTR,IARDF2(JARNUM,IARR),IARR)
      103 IARDF2(JARLST,IARR) = K
      104 CALL STORE(K,VFILE(IOFF),NF,IFLD,IRECS2(J))
      105 IRECS2(J) = K
      106 CONTINUE
      75 C READ NV VALUES FOR EACH OF THE NA ARRAYS AND STORE THEM
      107 DO 77 I=1,NV
      108 READ (4,115) (VALS(J),J=1,NA)
      109 BACKSPACE 4
      110 READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
      111 DO 76 J = 1,NA

```

```

      INPUT ROUTINES -  INALTC SUBROUTINE

      IF (CVALS(1,J) .EQ. BLS .AND. CVALS(2,J) .EQ. BLS)
      + VALS(J) = FNULL
      76 CONTINUE
      DO 77 J=1,NA
      113 CALL STORE(VALS(J),ARRAY(IOFFAR),NV,I,IRECS2(J))
      114 CONTINUE
      115 GO TO 40
      116
      117
      C
      C READ AND STORE CHARACTER INFORMATION (<= 8 CHARS)
      118 READ (4,125) (CVALS(1,J),CVALS(2,J),J=1,NA)
      119 FORMAT (12X,12(2A4,2X))
      C
      C DETERMINE WHICH ARRAY FILE INFORMATION WILL BE STORED IN
      120 DO 82 IARR=1,NARRS
      121 IF (IARDF2(JFLDRC,IARR) .EQ. IFLD) GO TO 83
      122 CONTINUE
      123 CALL ERR(16,'INALTC',IRECTP,IFLD,ITYP)
      C
      C OFFSET IN FILE CONTAINING CHARACTER STRINGS
      124 IOFFCH = IARDF2(JARRPT,IARR)
      C GET NA VACANT CHARACTER STRING RECORDS AND STORE POINTERS TO
      C THEM FROM THE CURRENT RECORD. ALSO STORE CHARACTER STRINGS IN
      C THE CHARACTER STRING RECORDS.
      125 DO 85 J=1,NA
      126 K = IARDF2(JARLST,IARR) + 1
      127 IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INALTC',
      + ATTR,IARDF2(JARNUM,IARR),IARR)
      128 IARDF2(JARLST,IARR) = K
      129 CALL STORE(K,VFILE(IOFF),NF,IFLD,IRECS2(J))
      130 CALL STORE(CVALS(1,J),ARRAY(IOFFCH),2,1,K)
      131 CALL STORE(CVALS(2,J),ARRAY(IOFFCH),2,2,K)
      132 CONTINUE
      133 GO TO 40
      C
      134 RETURN
      135 END

```

```

1SN
1  INPUT ROUTINES - INATTR SUBROUTINE

      SUBROUTINE INATTR(ATTR,IREC,ISPEC,IRECTP)

      ROUTINE TO READ THE VALUE(S) OF AN ATTRIBUTE OF INTEREST
      ASSOCIATED WITH A GIVEN VEHICLE IN THE EXISTING VEHICLE FILE.
      THE ATTRIBUTE'S VALUE MAY BE A REAL, INTEGER, OR CHARACTER
      SCALAR (<= 8 CHARS) OR AN ARRAY OF REAL NUMBERS.
      THIS ROUTINE PROCESSES ATTRIBUTES WHICH WILL BE USED AS
      CONSTRAINTS AND STORED IN A RELN RECORD. SUBROUTINE INATR2
      PROCESSES THOSE THAT ARE STORED AS CONSTANTS.

      C INPUT PARAMETERS:
      C ATTR  ATTRIBUTE OF INTEREST
      C IREC  RECORD OF IRECS CONTAINING INFORMATION ABOUT VEHICLE
      C       OF INTEREST
      C ISPEC RECORD DESCRIBING USER SPECIFICATION
      C IRECTP TYPE OF RECORD THE ATTRIBUTE IS A FIELD OF

      CCMON /VEH/
      DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
      + FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
      + ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
      + IAMMOC(26,9),IENGINE(40,8),ITRANS(32,8),IFINDR(28,10),
      + IROADW(36,10),ISPRNG(30,10),ITPACK(34,8),ISKIRT(28,8),
      + IFUEL(28,10),IFUEL(26,9),ICREW(26,5),ICARGN(26,5),
      + IELECT(26,6),ICOMMD(28,9),IFIREX(28,10),IENVIR(26,9),
      + IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
      + IRELN(3,50),VFILE(1),IVFILE(1)

      EQUIVALENCE (VEHICLE(1),IVECLE(1)),(IHULL(1),IHULL(1)),(TURRET(1),
      + TTURET(1)),(MAINGN(1),FMAING(1)),(MACGN(1),FMACGN(1)),
      + (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
      + IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
      + (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
      + IAMMOC(1)),(ENGINE(1),IENGINE(1)),(TRANSM(1),ITRANS(1)),
      + (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADWH(1)),(SPRING(1),
      + ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),

```

```

      INPUT ROUTINES -   INATTR SUBROUTINE

      + (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
      + ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
      + (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
      + IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
      + (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
      EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
      COMMON /VPROTO/
      REAL*8 FLDNAM,RECNAM,DEFAULT
      COMMON /JPARS/
      COMMON /JPARS1/
      COMMON /NPARS/
      COMMON /AUX/
      REAL*8 BL8,DASHES
      EQUIVALENCE (FNULL,NULL),(I8LS,BLS)
      COMMON /ARRAYS/
      REAL*8 MANUF,MODEL,NATION
      DIMENSION ARRAY(1)
      EQUIVALENCE (ARRAY(1),DETRNG(1))
      COMMON /JPARS2/
      COMMON /SPECS/
      REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
      COMMON /SPECS2/
      REAL*8 VEHS

      C
      REAL*8 ATTR
      DIMENSION CVALS(2,12),IVALS(12),VALS(12)

      C
      C MUST BACKSPACE TO POSITION RECORD POINTER APPROPRIATE FOR READING
      BACKSPACE 2

      C
      C GET TYPE OF ATTRIBUTE
      IFLD = IATTR2(ATTR,IRECTP)
      KFLD = IFLOPT(IRECTP) + IFLD
      ITYP = IFLDTP(KFLD)

      C
      C MARK ATTRIBUTE AS PROCESSED

```

```

      INPUT ROUTINES - INATTR SUBROUTINE

      IRECS(JDCNE,IREC) = 1

      C
      C GET COMPONENT IC IF THIS IS ENGINEERING PARAMETER CONSTRAINT
      ID = 0
      IF (IRECTP .GT. 1) ID = INDENG(ISPEC)

      C
      C TRANSLATE RELOP TO A CANONICAL VALUE FROM 1 TO 6
      DO 5 J=1,NRELOP
      IF (IRECTP .EQ. 1) REL = RELPER(ISPEC)
      IF (IRECTP .GT. 1) REL = RELENG(ISPEC)
      IF (REL .EQ. RELNAM(J)) GO TO 8
      CONTINUE
      IF (IRECTP .EQ. 1) CALL ERR(26,'INATTR','VEH',ATRPER(ISPEC),
      + RELPER(ISPEC))
      IF (IRECTP .GT. 1) CALL EFR(26,'INATTR',TYPENG(ISPEC),
      + ATRENG(ISPEC),RELENG(ISPEC))

      C
      KREL = IRELTF(J)

      C
      C GET THE COLUMN OF INTEREST IN THE EXISTING VEHICLE FILE
      IV = IRECS(JVEH,IREC)
      ICOL = ICOLS(IV)

      C
      C BRANCH ACCORDING TO DATATYPE
      DO 10 I=1,NTYPES
      IF (ITYP .EQ. ITPES(I)) GO TO 15
      CONTINUE
      CALL ERR(10,'INATTR',RECNAME(IRECTP),ATTR,ITYP)
      RETURN

      C
      15 GO TO (20,40,60,80),I

      C
      C READ & STORE A REAL VALUE
      20 READ (2,100) (VALS(I),I=1,ICOL)
      100 FORMAT (12X,12(F8.0,2X))
      VALUE = VALS(ICOL)

```

## INPUT ROUTINES - INATTR SUBROUTINE

```

15N
51 C TRANSFORM VALUE IF CALLED FOR BY USER SPECIFICATION
52 IF (IRECTP .GT. 1) GO TO 25
53 VAL = TRANSF(VALUE,MULPER(ISPEC),VL1PER(ISPEC),
54 + ADDPER(ISPEC),VL2PER(ISPEC))
55 GO TO 30
56
57 25 VAL = TRANSF(VALUE,MULENG(ISPEC),VLIENG(ISPEC),
58 + ADDENG(ISPEC),VL2ENG(ISPEC))
59 C STORE VALUE AS A CCNSTRANT OF THE RECORD
60 IR = NEWREL(IRECTP,IFLD,KREL,VAL,ID)
61 RETURN
62
63 C READ & STORE AN INTEGER VALUE
64 READ (2,105) (IVALS(I),I=1,ICOL)
65 FORMAT (12X,12(18,2X))
66 VALU = IVALS(ICOL)
67
68 C TRANSFORM VALUE IF CALLED FOR BY USER SPECIFICATION
69 IF (IRECTP .GT. 1) GO TO 45
70 IVALUE = TRANSF(VALUE,MULPER(ISPEC),VL1PER(ISPEC),
71 + ADDPER(ISPEC),VL2PER(ISPEC))
72 GO TO 50
73
74 45 IVALUE = TRANSF(VALUE,MULENG(ISPEC),VLIENG(ISPEC),
75 + ADDENG(ISPEC),VL2ENG(ISPEC))
76 IR = NEWREL(IRECTP,IFLD,KREL,IVALUE,ID)
77 RETURN
78
79 C READ & STORE AN ARRAY OF REAL VALUES
80
81 C DETERMINE WHICH ARRAY FILE THE INFORMATION WILL BE STORED IN
82 DO 62 IARR=1,NARRS
83 IF (IARDF2(JRECTP,IARR) .EQ. IRECTP .AND.
84 + IARDF2(JFLDRC,IARR) .EQ. IFLD) GO TO 63
85 CONTINUE
86 CALL ERR(16,'INATTR',IRECTP,IFLD,ITYP)
87
88 C NUMBER OF ELEMENTS IN ARRAY
89 NV = IARDF2(JARSIZ,IARR)
90

```

```

      INPUT ROUTINES -  INATTR SUBROUTINE

C  OFFSET IN FILE CONTAINING ARRAYS
71  IOFFAR = IARDF2(JARRPT,IARR)
C  GET A VACANT ARRAY OF TYPE IARR
72  K = IARDF2(JARLST,IARR) + 1
73  IF (K.GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INATTR',ATTR,
      + IARDF2(JARNUM,IARR),IARR)
74  IARDF2(JARLST,IARR) = K
C  GET A RELATION RECORD TO STORE THE CONSTRAINTS IN THIS FIELD IN
75  IR = NEWREL(IRECTP,IFLD,KREL,K,ID)
C  FOR EACH VALUE OF ARRAY SPECIFIED IN EXISTING VEHICLE FILE,
C  READ, TRANSFORM, & STORE IT.
76  DO 75 I=1,NV
77  READ (2,100) (VALS(J),J=1,ICOL)
78  VALUE = VALS(ICOL)
C  TRANSFORM VALUE IF CALLED FOR BY USER SPECIFICATION
79  IF (IRECTP.GT. 1) GO TO 65
80  VAL = TRANSF(VALUE,MULPER(ISPEC),VLI1PER(ISPEC),
      + ADDPER(ISPEC),VL2PER(ISPEC))
81  GO TO 70
82  VAL = TRANSF(VALUE,MULENG(ISPEC),VLIENG(ISPEC),
      + ADDENG(ISPEC),VL2ENG(ISPEC))
83  CALL STORE(VAL,ARRAY(IOFFAR),NV,I,K)
84  CONTINUE
85  RETURN
C
C  READ & STORE A CHARACTER VALUE (<= 8 CHARS)
86  READ (2,110) (CVALS(1,J),CVALS(2,J),J=1,ICOL)
87  FORMAT (12X,12(2A4,2X))
C  DETERMINE WHICH ARRAY FILE INFORMATION WILL BE STORED IN
88  DO 82 IARR=1,NARRS
89  IF (IARDF2(JFLDRC,IARR).EQ. IFLD) GO TO 83
90  CONTINUE
91  CALL ERR(16,'INATTR',IRECTP,IFLD,ITYP)
C
C  OFFSET IN FILE CONTAINING CHARACTER STRINGS
92  IOFFCH = IARDF2(JARRPT,IARR)

```

```

      INPUT ROUTINES -  INATTR SUBROUTINE

15N  C  GET A VACANT CHARACTER STRING RECORD
93    K = IARDF2(JARLST,IARR) + 1
94    IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INATTR',ATTR,
      + IARDF2(JARNUM,IARR),IARR)
95    IARDF2(JARLST,IARR) = K
      C  CHECK THAT RELATION IS EQUALITY OR INEQUALITY (OTHER RELNS NOT
      C  APPROPRIATE FOR CHARACTER DATA).
96    IF (KREL .NE. 3 .AND. KREL .NE. 4) CALL ERR(27,'INATTR',
      + KREL,ATTR,IRECTP)
      C  GET A RELATION RECORD TO STORE THE CONSTRAINT ON THIS FIELD IN
97    IR = NEWREL(IRECTP,IFLD,KREL,K,1D)
      C  STORE CHARACTER DATA IN CHARACTER RECORD
98    CALL STORE(CVALS(1,ICOL),ARRAY(IOFFCH),2,1,K)
99    CALL STORE(CVALS(2,ICOL),ARRAY(IOFFCH),2,2,K)
100   RETURN
101   END

```

```

ISN      INPUT ROUTINES -   INATR2 SUBROUTINE
1      SUBROUTINE INATR2(ATTR,IREC,ISPEC,IRECTP)
C
C      ROUTINE TO READ THE VALUE(S) OF AN ATTRIBUTE OF INTEREST
C      ASSOCIATED WITH A GIVEN VEHICLE IN THE EXISTING VEHICLE FILE.
C      THE ATTRIBUTE'S VALUE MAY BE A REAL, INTEGER, OR CHARACTER
C      SCALAR (<= 8 CHARS) OR AN ARRAY OF REAL NUMBERS.
C      THIS ROUTINE PROCESSES ATTRIBUTES WHICH WILL BE
C      STORED AS CONSTANTS IN A COMPONENT RECORD. SUBROUTINE INATTR
C      PROCESSES THOSE THAT ARE STORED AS CONSTRAINTS.
C
C      INPUT PARAMETERS:
C      ATTR    ATTRIBUTE OF INTEREST
C      IREC    RECORD OF IRECS CONTAINING INFORMATION ABOUT VEHICLE
C              OF INTEREST
C      ISPEC   RECORD DESCRIBING USER SPECIFICATION
C      IRECTP  TYPE OF RECORD THE ATTRIBUTE IS A FIELD OF
C
C      COMMON /VEH/
2      DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
3      FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
      + ISENSOR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
      + IAMMOC(26,9),IENGINE(40,8),ITRANS(32,8),IFINDR(28,10),
      + IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
      + IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
      + IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
      + IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
      + IRELN(3,50),VFILE(1),IVFILE(1)
4      EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
      + ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
      + (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
      + IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
      + (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
      + IAMMOC(1)),(ENGINE(1),IENGINE(1)),(TRANSM(1),ITRANS(1)),
      + (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
      + ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),

```

```

      INPUT ROUTINES - INATR2 SUBROUTINE

      + (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
      + ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
      + (COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),
      + IENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
      + (SMOKEG(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))
      EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))
      COMMON /VPROTO/
      REAL*8 FLDNAM, RECNAM, DEFAULT
      COMMON /JPARS/
      COMMON /JPARS1/
      COMMON /NPARS/
      COMMON /AUX/
      REAL*8 BL8, CASHES
      EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
      COMMON /ARRAYS/
      REAL*8 MANUF, MODEL, NATICN
      DIMENSION ARRAY(1)
      EQUIVALENCE (APRAY(1), DETRNG(1))
      COMMON /JPARS2/
      COMMON /SPECS/
      REAL*8 ATRENG, ATRPER, VEHCAT, VEHCMP, VEHENG, VEHPER
      COMMON /SPECS2/
      REAL*8 VEHS

      C
      REAL*8 ATTR
      DIMENSION CVALS(2, 12), IVALS(12), VALS(12)

      C
      C MUST BACKSPACE TO POSITION RECORD POINTER APPROPRIATE FOR READING
      BACKSPACE 2

      C
      C GET TYPE OF ATTRIBUTE
      IFLD = IATTR2(ATTR, IRECTP)
      KFLD = IFLOPT(IRECTP) + IFLD
      ITYP = IFLOTP(KFLD)

      C
      C GET COMPONENT IC AND COMPONENT TYPE

```

## INPUT ROUTINES - INATR2 SUBROUTINE

```

1 SN
29 ID = INDCMP(ISPEC)
30 ICMF = IRECTP - 1
31 C
32 C MARK SPECIFICATION AS PROCESSED
33 IRECS(JDONE, IREC) = 1
34 C
35 C GET NUMBER OF FIELDS IN RECORD
36 NF = NFIDS(IRECTP)
37 C GET OFFSET OF RECORD FILE
38 IOFF = IFILPT(IRECTP)
39 C
40 C GET POINTER TO RECORD HAVING THE APPROPRIATE ID TO STORE THE DATA
41 C IN FOR THIS COMPONENT; IF ONE DOESN'T EXIST, CREATE ONE.
42 IR = IRFIND(IRECTP, ID, ISPECV)
43 IF (IR .NE. NULL) GO TO 8
44 IR = NEWREC(IRECTP)
45 CALL STORE(IVECLE(ICMP, ISPECV), VFILE(IOFF), NF, JID, IR)
46 IVECLE(ICMP, ISPECV) = IR
47 C STORE THE IC
48 CALL STORE(ID, VFILE(IOFF), NF, JID, IR)
49 C
50 C GET THE COLUMN OF INTEREST IN THE EXISTING VEHICLE FILE
51 IV = IPECS(JVEH, IREC)
52 ICOL = ICOLS(IV)
53 C
54 C BRANCH ACCORDING TO DATATYPE
55 DO 10 I=1, NTYPES
56 IF (ITYP .EQ. ITYPES(I)) GO TO 15
57 CONTINUE
58 CALL ERR(10, 'INATR2', RECNAM(IRECTP), ATTR, ITYP)
59 RETURN
60 C
61 15 GO TO (20, 40, 60, 80), I
62 C
63 C READ & STORE A REAL VALUE
64 20 READ (2, 100) (VALS(I), I=1, ICOL)

```

```

      INPUT ROUTINES -   INATR2 SUBROUTINE

ISN  49      100  FORMAT (12X,12(F8.0,2X))
      50  C   STORE VALUE IN THE RECCRD
      51  CALL STORE(VALS(ICOL),VFILE(IOFF),NF,IFLD,IR)
      RETURN
C
C  52  C   READ & STORE AN INTEGER VALUE
      53  40  READ (2,105) (IVAL5(I),I=1,ICOL)
      54  105  FORMAT (12X,12(18,2X))
      55  CALL STORE(IVAL5(ICOL),VFILE(IOFF),NF,IFLD,IR)
      RETURN
C
C   C   READ & STORE AN ARRAY OF REAL VALUES
C
C   C   DETERMINE WHICH ARRAY FILE THE INFORMATION WILL BE STORED IN
      56  60  DO 62 IARR=1,NARRS
      57  IF (IARDF2(JRECTP,IARR) .EQ. IRECTP .AND.
      58  +   IARDF2(JFLDRC,IARR) .EQ. IFLD) GO TO 63
      59  CONTINUE
      CALL ERR(16,'INATR2',IRECTP,IFLD,ITYP)
C
C   C   NUMBER OF ELEMENTS IN ARRAY
      60  63  NV = IARDF2(JARSIZ,IARR)
C   C   OFFSET IN FILE CONTAINING ARRAYS
      61  IOFFAR = IARDF2(JAPRPT,IARR)
C   C   GET A VACANT ARRAY OF TYPE IARR
      62  K = IARDF2(JARLST,IARR) + 1
      63  IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INATR2',ATTR,
      64  +   IARDF2(JARNUM,IARR),IARR)
      IARDF2(JARLST,IARR) = K
C   C   STOPS POINTER TO THIS ARRAY
      65  CALL STORE(K,VFILE(IOFF),NF,IFLD,IR)
C   C   FOR EACH VALUE OF ARRAY SPECIFIED IN EXISTING VEHICLE FILE,
C   C   READ, TRANSFORM, & STORE IT.
      66  DO 75 I=1,NV
      67  READ (2,100) (VALS(J),J=1,ICOL)
      68  CALL STORE(VALS(ICOL),ARRAY(IOFFAR),NV,I,K)

```

```

      INPUT ROUTINES -  INATR2 SUBROUTINE

      ISN
69      75      CONTINUE
70      RETURN

      C
71      C READ & STORE A CHARACTER VALUE (<= 8 CHARS)
72      READ (2,110) (CVALS(1,J),CVALS(2,J),J=1,ICOL)
73      FORMAT (12X,12(2A4,2X))
74      C DETERMINE WHICH ARRAY FILE INFORMATION WILL BE STORED IN
75      DO 82 IARR=1,NARRS
76      IF (IARDF2(JFLDRC,IARR) .EQ. IFLD) GO TO 83
      CONTINUE
      CALL ERR(16,'INATR2',IRECTP,IFLD,ITYP)

      C
77      C OFFSET IN FILE CONTAINING CHARACTER STRINGS
78      IOFFCH = IARDF2(JARRPT,IARR)
79      C GET A VACANT CHARACTER STRING RECORD
      K = IARDF2(JARLST,IARR) + 1
      IF (K .GT. IARDF2(JARNUM,IARR)) CALL ERR(17,'INATR2',ATTR,
      + IARDF2(JARNUM,IARR),IARR)
      IARDF2(JARLST,IARR) = K
      C STORE POINTER TO THE CHARACTER RECORD
      CALL STORE(K,VFILE(IOFF),NF,IFLD,IR)
      C STORE CHARACTER DATA IN CHARACTER RECORD
      CALL STORE(CVALS(1,ICOL),ARRAY(IOFFCH),2,1,K)
      CALL STORE(CVALS(2,ICOL),ARRAY(IOFFCH),2,2,K)
      RETURN
      END

```

## INPUT ROUTINES - INFUNC SUBROUTINE

## SUBROUTINE INFUNC

SUBROUTINE READS A FORMATTED FILE OF DATA VARIABLE  
HEADERS FOLLOWED BY DATA FOR EACH VARIABLE. THE DATA VARIABLES  
CONSIST OF SEQUENTIAL VARIABLES IN A COMMON BLOCK. EACH VARIABLE  
HEADER RECORD HAS THE FOLLOWING FIELDS:

| NAME       | COL 1-8 | VARIABLE NAME                                 |
|------------|---------|-----------------------------------------------|
| DIM 1      | 9-11    | 1ST DIMENSION OF VARIABLE IN PROGRAM          |
| DIM 2      | 12-14   | 2ND DIMENSION OF VARIABLE IN PROGRAM          |
| DIM 3      | 15-17   | 3RD DIMENSION OF VARIABLE IN PROGRAM          |
| STAR       | 19      | '*' IF DATA FOR VARIABLE IS ALL ZEROS         |
| RED. DIM 1 | 20-22   | (OPTIONAL) REDUCED DIMENSION APPLYING TO DATA |

| REPL 1 | 23 | (OPTIONAL) DATA GIVEN ALONG FIRST SECTION |
|--------|----|-------------------------------------------|
|--------|----|-------------------------------------------|

OF DIMENSION IS TO BE REPLICATED ALONG  
LATER SECTIONS IF FIELD CONTAINS "##"

|            |       |                                   |
|------------|-------|-----------------------------------|
| RED DIM 2  | 24-25 | (OPTIONAL) REDUCED DIMENSION 2    |
| REPL 2     | 26    | (OPTIONAL) REPLICATION SYMBOL (#) |
| RED DIM 3  | 27-28 | (OPTIONAL) REDUCED DIMENSION 3    |
| REPL 3     | 29    | (OPTIONAL) REPLICATION FACTOR 3   |
| FTN FORMAT | 31-78 | FORTRAN FORMAT (IN PARENTHESES)   |

THE DATA VALUES FOR A VARIABLE FOLLOW THE HEADER CARD. FOR  
ARRAYS THE NUMBER OF DATA VALUES EXPECTED IS THE PRODUCT OF THE  
NON-ZERO DIMENSIONS GIVEN ON THE HEADER CARD. IF REDUCED DIMENSIONS  
ARE SPECIFIED THEY ARE USED INSTEAD OF THE ACTUAL DIMENSIONS IN  
COMPUTING THE NUMBER OF EXPECTED DATA VALUES. FOR SCALARS, VARIABLES  
WITH NO DIMENSIONS, A SINGLE VALUE IS EXPECTED. IF A '\*' IS GIVEN  
IN THE STAR FIELD, ZERO DATA VALUES ARE EXPECTED & THE NEXT HEADER  
CARD SHOULD FOLLOW IMMEDIATELY. THE FORMAT THAT THE DATA VALUES  
APPEAR IN IS UP TO THE USER. THE FORMAT IS INDICATED ON THE HEADER  
CARD & MUST BE A VALID FORTRAN FORMAT SPECIFICATION.

A BLOCK OF DATA VARIABLES MAY BE TERMINATED BY A RECORD WITH  
"ENDBLK" IN COLS 1-6. ALL VARIABLES WITHIN A BLOCK MUST BE IN PROPER

ISN  
1

## INPUT ROUTINES - INFUNC SUBROUTINE

```

1 SN      C SEQUENCE & IN 1:1 CORRESPONDENCE WITH THE VARIABLES OF THE ASSOCIATED
          C COMMON BLOCK USED IN THE MAIN PROGRAM. SEE THE USER'S MANUAL FOR A
          C LIST OF THE INPUT VARIABLES EXPECTED, THEIR ORDER, & THEIR
          C DIMENSIONS.
          C THE DATA SET REFERENCE NUMBERS USED ARE AS FOLLOWS:
          C 1 INPUT
          C 8 OUTPUT ECHO OF DATA FILE HEADER CARDS
          C
          C COMMON /DATA/
          C COMMON /DATA2/
          C EQUIVALENCE (VEC(1),ACCEL(1))
          C
          C REAL VEC(2),V(1000)
          C REAL*8 VAR,ENDBLK,ENDFIL
          C INTEGER FMT(12)
          C DATA STAR1/1H*/ ,ENDBLK/6HENDBLK/,ENDFIL/6HENDFIL/
          C DATA SHARP/1H#/
          C
          C CALL FTNCMD('SET MODECHECK=OFF',17)
          C JJO = 0
          C READ (1,100,END=44) VAR,I,J,K,STAR,I2,S1,J2,S2,K2,S3,FMT
          C WRITE (8,101) JJO,VAR,I,J,K,STAR,I2,S1,J2,S2,K2,S3,FMT
          C FORMAT (A8,3I3,1X,A1,I3,2(A1,I2),A1,1X,12A4)
          C FORMAT (I6,2X,A8,3I3,1X,A1,I3,2(A1,I2),A1,1X,12A4)
          C IF (VAR .EQ. ENDBLK .OR. VAR .EQ. ENDFIL) GO TO 44
          C JSIZE = 1
          C IF (I .NE. 0) JSIZE = JSIZE*I
          C IF (J .NE. 0) JSIZE = JSIZE*J
          C IF (K .NE. 0) JSIZE = JSIZE*K
          C IF(JSIZE .GT. 5000) STOP 100
          C JJ2 = JJO + JSIZE
          C IF(JJ2 .GT. 5000) STOP 200
          C JJ1 = JJO + 1
          C IF (STAR .EQ. STAR1) GO TO 40
          C *** IF ALL ZEROS, THEN SKIP THIS VARIABLE
          C IF (I .NE. 0) GO TO 31

```

## INPUT ROUTINES - INFUNC SUBROUTINE

```

15N
27 C ***      IS THIS A SCALAR
28   READ (1,FMT) VEC(JJ1)
29   GO TO 40
30 C
31   IF (J .NE. 0) GO TO 32
32   IS IT A VECTOR ?
33   IF (I2 .EQ. 0) I2 = 1
34   JJ3 = JJ0 + I2
35   READ (1,FMT) (VEC(M),M=JJ1,JJ3)
36   GO TO 40
37 C
38   IF (K .NE. 0) GO TO 36
39   IS IT A 2-D ARRAY
40   IF (I2 .NE. 0) GO TO 33
41   READ (1,FMT) (VEC(M),M=JJ1,JJ2)
42   GO TO 40
43 C
44   IJ=I2*J2
45   READ (1,FMT) (V(M),M=1,IJ)
46   DO 34 I1=1,I2
47     DO 34 J1=1,J2
48       J11 = J1-1
49       M1 = JJ0+I1+I*J11
50       M2 = I1+I2*J11
51       VEC(M1) = V(M2)
52   GO TO 40
53 C
54   IF (I2 .NE. 0) GO TO 38
55   READ (1,FMT) (VEC(M),M=JJ1,JJ2)
56   GO TO 40
57 C
58   IJK = I2*J2*K2
59   READ (1,FMT) (V(M),M=1,IJK)
60   DO 39 I1=1,I2
61     DO 39 J1=1,J2
62       J11 = J1-1
63       DO 39 K1=1,K2
64         K11 = K1-1

```

## INPUT ROUTINES - INFUNC SUBROUTINE

```

ISN
57      M1=JJ0+I1+I*J11+I*J*K11
58      M2=I1+I2*J11+I2*J2*K11
59      VEC(M1) = V(M2)

39
C
C
40
60      IREP = 1
61      IF (S1.EQ. SHARP) IREP = IREP+1
62      IF (S2.EQ. SHARP) IREP = IREP+J
63      IF (S3.EQ. SHARP) IREP = IREP*K
64      IF (IREP.EQ. 1) GO TO 43
65      J2SIZE = JSIZE/IREP
66      L1 = JJ0
67      DO 42 ICT=1, IREP
68          L1 = L1 + J2SIZE
69          DO 42 L2=1, J2SIZE
70              VEC(L1+L2)=VEC(JJ0+L2)
71      JJ0 = JJ0 + JSIZE
72      GO TO 11
73      RETURN
74      END

```

```

ISN      INPLT RCUTINES -   INIT SUBROUTINE
1
C
C
2
3
SUBROUTINE INIT
ROUTINE TO INITIALIZE VARIABLES AT BEGINNING OF RUN.
C
COMMON /VEH/
DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
+ FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANSR(28,7),
+ ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
+ IAMMOC(26,9),IENGINE(40,8),ITRANS(32,8),IFINDR(28,10),
+ IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
+ IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
+ IFLECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
+ IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
+ IRELN(3,50),VFILE(1),IVFILE(1)
EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
+ ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
+ (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
+ TRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
+ (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
+ IAMMOC(1)),(ENGINE(1),IENGINE(1)),(TRANSM(1),ITRANS(1)),
+ (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
+ ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
+ (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
+ ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
+ (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
+ IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
+ (SMOKEG(1),ISMOKE(1)),(IEWSYS(1),IEWSYS(1)),(IRELN(1),IRELN(1))
EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
COMMON /VPROTO/
REAL*8 FLDNAM,RECNAM,DEFAULT
COMMON /NPARS/
COMMON /OUTPUT/
LOGICAL*1 DF
REAL*8 XNAT,XMANUF,XMODEL
COMMON /AUX/
REAL*8 BL8,DASHES

```

```

13N      INPUT ROUTINES -  INIT SUBROUTINE
14      EQUIVALENCE (FNJLL, NULL), (IBLS, BLS)
15      COMMON /ARRAYS/
16      REAL*8 MANUF, MODEL, NATION
17      DIMENSION ARRAY(1)
18      EQUIVALENCE (ARRAY(1), DETRNG(1))
19      COMMON /JPARS2/
20      COMMON /SPECS/
21      REAL*8 ATRENG, ATRPER, VEHCAT, VEHCMP, VEHENG, VEHPER

C      INITIALIZE IFLDP based on number of fields in each type of
C      record
C
22      N = 0
23      NN = 1
24      IFLDP(1) = N
25      DO 10 I=1, NRECTP
26          N = N + NFLDS(I)
27          IFILPT(I) = NN
28          NN = NN + NFLDS(I) * NRECS(I)
29          IFLDP(I+1) = N
C
C      DETERMINE NUMBER OF CELLS IN ALL RECORDS COMBINED
C
30      NCELL = 0
31      DO 15 I=1, NRECTP
32          NCELL = NCELL + NFLDS(I) * NRECS(I)
C
C      INITIALIZE THE FIELDS OF ALL RECORDS TO NULL
C
33      CALL NULIFY(VEHICLE, NCELL)
C
C      INITIALIZE VARIABLES THAT WILL CONTAIN TEXT TO BLANKS
C
34      CALL INITVL(FLCDNAM, NFLDRC*2, BLS)
35      CALL INITVL(RECDNAM, NRECTP*2, BLS)
36      CALL INITVL(CMPCOD, NCMPTP, BLS)

```

## INPUT ROUTINES -- INIT SUBROUTINE

```

ISN
37      CALL INITVL(UNITS,NFLDRC,BLS)
      C
      C INITIALIZE COUNTERS TO ZERO
      C
      LSTCMP = 0
      LSTENG = 0
      LSTPER = 0
      CALL IZERO(LSTREC,NRECTP)
      CALL IZERO(MAXNUM,NCMPPT)
      CALL IZERO(MINNUM,NCMPPT)
      CALL IZERO(NALTC,NCMPPT)
      C
      C INITIALIZE FIELD TYPES AND VALUES TO NULL
      C
      CALL NULIFY(IFLDT, NFLDRC)
      CALL NULIFY(IFLDVL, NFLDRC)
      CALL NULIFY(ICMPPT, NCMPPT)
      C
      C INITIALIZE POINTERS TO ARRAYS
      C
      NN = 1
      DO 20 I=1,NARRS
        IARDF2(JARLST,I) = 0
        IARDF2(JARRPT,I) = NN
        NA = IARDF2(JARSIZ,I) * IARDF2(JARNUM,I)
        NN = NN + NA
      20 C
      RETURN
      END
54
55

```

```

      INPUT ROUTINES - INPROT SUBROUTINE

      SUBROUTINE INPROT

C
C  ROUTINE FOR READING THE PROTOTYPE VEHICLE DESCRIPTION FILE AND
C  STORING THIS INFORMATION IN INTERNAL RECORDS AND ARRAYS.
C

      COMMON /VPROTO/
      REAL*8 FLDNAM,RECNAME,DEFAULT
      COMMON /NPARS/
      COMMON /OUTPUT/
      LOGICAL*1 DF
      REAL*8 XNAT,XMANUF,XMODEL
      COMMON /AUX/
      REAL*8 BL8,DASHES
      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      COMMON /ARRAYS/
      REAL*8 MANUF,MODEL,NATION
      DIMENSION ARRAY(1)
      EQUIVALENCE (ARRAY(1),DETRNG(1))
      COMMON /JPARS2/
      COMMON /SPECS/
      REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER

C
      COMMON //
      DIMENSION CNAME(7)
      EQUIVALENCE (CMPTYP,CNAME(1)), (CMPTYP,CMPTP2)
      REAL*8 V,VH,CMPTYP,RWNAME,PPARS,COMATR,COMEND,DFVAL
      DATA STAR/'*','/','VH','VEHICLE' /
      DATA KPA/'RA'/'PPARS'/'I. PERFO'/'COMATR'/'A. COMM'/'
      + CMEND/'B. SPEC'/'KC8'/'CR'/'

C
      INITIALIZATIONS
      LSTOUT = 0
      IARRAY = 0
      ICOM = 0
      KMP=0

```

## INPUT ROUTINES - INPROT SUBROUTINE

```

ISN
28      K = 10
29      LARDEF = 0
30      LEVEL = 0

      C
      C READ VEHICLE HEADER; SKIP OVER COMMENT LINES
31      10 READ (3,100) C,V,VEHCAT
32      100 FORMAT (A1,A7,3X,A6)
33      IF (C.EQ. STAR) GO TO 10
34      IF (V.NE. VH) CALL ERR(1,'INPROT',V,0,0)
35      IREC = 1
36      IRECPT = IFLCPT(IREC)
37      RECNAM(IREC) = VH
38      ISTATE = 1

      C
      C READ A NEW RECORD
39      15 READ(3,105,END=90) C,CMPYTP,FILL,CCOD, ITYPE
40      105 FORMAT(A1,A8,5A4,23X,A2)
      C
      C IS THIS A COMMENT? IF SO, SKIP INPUT DATA PROCESSING.
      C
      C      IF(C.EQ.STAR) GO TO 30
41
      C
      C IF THIS IS DATA, BACKSPACE AND READ INPUT RECORD AGAIN.
      C
      C      BACKSPACE 3
42
      C
      C READ DATA ON APPROPRIATE FORMAT AS INDICATED BY ITYPE;
      C ITYPE.EQ.'C8' IMPLIES CHARACTER DFVAL, .NE.'C8' IMPLIES NUMERIC.
      C
      C      IF(ITYPE.EQ.KC8) READ(3,107) CCOD,RWNAME,IFLDV,MIN,MAX,UNIT,
      C          + DFVAL
43
      C      107 FORMAT(25X,A4,A8,10X,I2,5X,2I3,2X,A4,2X,A8)
44
      C      IF(ITYPE.NE.KC8) READ(3,109) CCOD,RWNAME,IFLDV,MIN,MAX,UNIT,
      C          + DFVAL
45
      C      109 FORMAT(25X,A4,A8,10X,I2,5X,2I3,2X,A4,2X,F8.0)
      C

```

```

C CHECK FOR NEW COMPONENT, INDICATED BY BLANK RENAME AND IFLDV.
C IF TRUE, SKIP TC NEW COMPONENT SECTION.
C
47 IF (RNAME .EQ. BL8 .AND. IFLDV .EQ. 0) GO TO 20
C
C PROCESS NEW ATTRIBUTE OF CURRENT RECORD
C IF (IFLDV .GT. NFLOS(IREC)) CALL ERR(2,'INPROT',RNAME,IFLDV,IREC)
48 IFLDRC = IRECT + IFLDV
49 FLDNAM(IFLDR) = RNAME
50 IFLOVL(IFLDR) = IFLDV
51 IFLOTP(IFLDR) = ITYPE
52 UNITS(IFLDR) = UNIT
53 DEFAULT(IFLDR) = DFVAL
54 IF (ISTATE.NE.1) GO TO 18
55 MINNUM(IFLDV) = MIN
56 MAXNUM(IFLDV) = MAX
57 KMP = KMP + 1
58 KOUT(KMP) = IFLDV
59 IF (ISTATE .EQ. 3 .OR. ICOM .EQ. 0) CALL STONAM(ONAME,IFLDR)
60 IARRAY = 0
61 IF (ITYPE .EC. KRA) IARRAY = 1
62 GO TO 15
63
C
C PROCESS A NEW COMPONENT RECORD
64 IPEC = IREC + 1
65 ICOMP = IREC - 1
66 IF (IREC .GE. NRECTP) CALL ERR(3,'INPROT',IREC,NRECTP,0)
67 IRECT = IFLOTP(IREC)
68 RECNAM(IREC) = CMPTYP
69 CMPCOD(ICMP) = CCOD
70 CALL STONAM(CNAME,0)
71 GO TO 15
C
30 CONTINUE
C
C COMMENT PROCESSING SECTION

```

```

      INPUT ROUTINES -   INPROT SUBROUTINE

      C   END-OF-ARRAY
      C
      C   IARRAY = 0
      C
      C   38   CONTINUE
      C
      C   PROCESSING FOR COMMENTS OTHER THAN ARRAY LABELS
      C
      C   IS THIS THE DEFINITION OF AN ATTRIBUTE LEVEL ? IF SO SKIP IT
      C   IF (CMPTYP .EQ. BL8) GO TO 15
      C
      C   BRANCH TO CODE ACCORDING TO STATE
      C   STATE = 1: READING VEHICLE COMPONENT HEADERS; CHECK FOR
      C   BEGINNING OF PERFORMANCE PARAMETERS
      C   2: READING PERFORMANCE PARS; CHECK FOR END OF PERF PARS
      C   3: READING COMPONENTS & THEIR ATTRIBUTES; CHECK FOR
      C   BEGINNING OF COMMON ATTRIBUTES
      C   4: READING COMMON ATTRIBUTES; CHECK WHETHER HAVE
      C   REACHED END OF SECTION OR HAVE ALREADY READ THEM
      C
      C   GO TO (40,50,60,70),ISTATE
      C
      C   STATE 1. IS THIS BEGINNING OF PERFORMANCE PARS ?
      C   IF (CMPTYP .NE. PPARS) GO TO 45
      C   IPPARS = LSTOUT + 1
      C   ISTATE = 2
      C   GO TO 15
      C   STORE NAME IN OUTNAM
      C   45   CALL STONAM(CNAME,0)
      C   GO TO 15
      C
      C   STATE 2. IS THIS THE END OF THE PERFORMANCE PARS ?
      C   IF (CMPTYP .NE. DASHES) GO TO 55
      C   ICMATR = LSTCUT + 1
      C   ISTATE = 3
      C   GO TO 15

```

```

      INPUT ROUTINES -  INPROT SUBROUTINE

      C
      C CHECK WHETHER CURRENTLY PROCESSING ARRAY LABLES; IF NOT SKIP
      C ARRAY LABEL SECTION
      C   IF (IARRAY .NE. 1) GO TO 38
      C
      C ARRAY LABEL PROCESSING SECTION
      C
      C CHECK FOR END OF CURRENT ARRAY LABLES; IF TRUE,
      C SKIP TO END-OF-ARRAY
      C   IF (CMPTP2.NE.BLS.OR.FILL(1).EQ.BLS)GO TO 37
      C
      C IF THIS IS LABEL OF ANOTHER LEVEL, STORE IT
      C   IF (CMPTYP .NE. BL8) GO TO 34
      C     LEVEL = LEVEL + 1
      C   IF (LEVEL .GT. NARLAB) CALL ERR(36,'INPROT',IREC,IFLDV,LEVEL)
      C     ILEVEL = ILEVEL + 1
      C   CALL COPY(ONAME(3),ARRLAB(1,LEVEL),5)
      C   GO TO 15
      C
      C THIS IS A NEW DIMENSION NAME. CHECK THAT PREVIOUS DIMENSION
      C HAD RIGHT NUMBER OF LEVELS
      C   IF (LARDEF .EQ. 0) GO TO 36
      C     IF (ILEVEL .NE. IARDEF(JNLEVL,LARDEF)) CALL ERR(34,'INPROT',
      C       + IREC,IFLDV,ILEVEL)
      C
      C CHECK THAT CURRENT DIMENSION IS DEFINED IN IARDEF
      C   LARDEF = LARDEF + 1
      C   IF (IARDEF(JRECTP,LARDEF) .NE. IREC .OR. IARDEF(JFLDRC,LARDEF)
      C     + .NE. IFLDV) CALL ERR(35,'INPROT',IREC,IFLDV,LARDEF)
      C     ILEVEL = 0
      C   STORE POINTER TO LEVEL
      C   IARDEF(JLEVPT,LARDEF) = LEVEL + 1
      C   GO TO 15
      C
      C   CONTINUE
      C

```

```

      INPUT ROUTINES -  INPROT SUBROUTINE

ISM
103 C   SAVE NAME IN OUTNAM
104 C   CALL STONAM(CNAME,0)
      GO TO 15

105 C   STATE 3.  IS THIS THE BEGINNING OF COMMON ATTRIBUTES ?
106 C   IF (CMPTYP .NE. COMATR) GO TO 65
107 C   ISTATE = 4
      GO TO 15

108 C   HAS THE COMMON ATTRIBUTE SECTION BEEN PROCESSED YET ?
109 C   IF (ICOM .NE. 0) GO TO 68
110 C   IF NOT, STORE THESE FIRST FEW LINES IN A TEMPORARY LOCATION
111 C   AND LATER MOVE THEM TO THE END OF THE COMMON ATTRIBUTE SECTION.
      CALL COPY(ONAME,OUTNAM(1,NOUT-K),NOUTWD)
      K = K - 1
      GO TO 15

112 C   STORE NAME IN OUTNAM
113 C   IF (CMPTYP .EQ. DASHES) GO TO 15
114 C   CALL STONAM(CNAME,0)
      GO TO 15

115 C   STATE 4.  HAS ONE ALREADY READ THESE ATTRIBUTES ONCE ?
      CONTINUE
116 C   IS THIS THE END OF THE SECTION OF COMMON ATTRIBUTES ?
117 C   IF (CMPTYP .NE. COMEND .AND. CMPTYP .NE. DASHES) GO TO 75
118 C   ISTATE=3
      IF(ICOM.NE.0) GO TO 15

119 C   AT THIS POINT ALL COMMENTS ASSOCIATED WITH COMMON ATTRIBUTES
120 C   HAVE BEEN READ ONCE; WE SET POINTER AND FLAGS TO INDICATE THIS
      IEPARS = LSTOUT + 1
      ICOM=1
121 C   INSERT THE TEMPORARILY STORED NAMES INTO THE LIST AT THIS POINT
      NR = 10 - K

```

```

      INPUT ROUTINES -  INPROT SUBROUTINE

      I SN
122      CALL COPY(OUTNAM(1,NOUT-10),OUTNAM(1,LSTOUT+1),NR#NOUTWD)
123      CALL IZERO(IOUTF(LSTOUT+1),NR)
124      LSTOUT = LSTOUT + NR
125      GO TO 15

      C  STORE NAME IN OUTNAM
126      75  IF(ICOM.EQ.0)CALL STGNAM(ONAME,0)
127      GO TO 15

      C
128      90  RETURN
129      END

```

```

      INPUT ROUTINES -  INSPC SUBROUTINE

      SUBROUTINE INSPC

C
C ROUTINE FOR READING AND INITIALLY STORING USER SEPCIFICATIONS
C IN INTERNAL FORM CLCSELY PARALLELING EXTERNAL FORM.
C
      CCMCN /NPARS/
      COMMON /AUX/
      REAL*8 BL8,DASHES
      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      COMMON /SPECS/
      REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER

C
      DIMENSION CMPHD(4)
      REAL*8 V
      DATA VH/'VEH'/,WT/'WEIG'/,GWT/'GROS'/,EVAL/'EVAL'/,
      + DESIR/'DESI'/,CMPHD/'C','E','P','V'/,VEH/'VEH'/

C
C READ & ECHO VEHICLE CLASS
      READ (5,100) HEAD,V
      WRITE (8,100) HEAD,V
      100 FORMAT (A4,16X,A8)
      IF (HEAD.NE. VH) CALL ERR(11,'INSPC',HEAD,VH,0)
      IF (V.NE. VEHCAT) CALL ERR(12,'INSPC',V,VEHCAT,0)

C
C READ & ECHO GROSS WEIGHT ESTIMATE FOR VEHICLE
      READ (5,105) HEAD,VEHWT
      WRITE (8,105) HEAD,VEHWT
      105 FORMAT (A4,16X,F8.0)
      IF (HEAD.NE. WT .AND. HEAD.NE. GWT) CALL ERR(11,'INSPC',
      + HEAD,WT,0)

C
C READ & ECHO SET OF EVALUATION WEIGHTS FOR VARIOUS CATEGORIES
      READ (5,110) HEAD,VEHEVL
      WRITE (8,110) HEAD,VEHEVL
      110 FORMAT (A4,16X,5F8.0)

```

```

1  INPUT ROUTINES - INSPEC SUBROUTINE
2
3  IF (HEAD.NE.EVAL.AND.HEAD.NE.DESIR) CALL ERR(11,'INSPEC',
4  + HEAD,EVAL,0)
5
6  C READ HEADER
7
8  READ (5,115,END=90) HEAD
9  FORMAT (A1)
10
11 C
12
13 C BRANCH ACCORDING TO HEADER
14
15 DO 10 I=1,4
16   IF (CMPHD(I).EQ.HEAD) GO TO (20,30,40,40),I
17   CONTINUE
18   CALL ERR(14,'INSPEC',HEAD,3,0)
19
20 C
21 C READ & STORE COMPONENT SPECIFICATIONS
22
23 READ (5,115,END=90) HEAD
24 IF (HEAD.NE.BLS) GO TO 5
25 BACKSPACE 5
26 I = LSTCMP + 1
27 IF (I.GT. NCSPCS) CALL ERR(13,'INSPEC','COMPONENT',NCSPCS,0)
28 READ (5,120,END=90) HEAD,NUMCMP(I),TYPCMP(I),INDCMP(I),VEHCMP(I)
29 WRITE (8,120) HEAD,NUMCMP(I),TYPCMP(I),INDCMP(I),VEHCMP(I)
30 FORMAT (A1,I2,1X,A3,1X,12,1X,A6)
31 LSTCMP = I
32 GO TO 20
33
34 C
35 C READ & STORE ENGINEERING PARAMETER SPECIFICATIONS
36
37 I = LSTENG + 1
38 IF (I.GT. NESPCS) CALL ERR(13,'INSPEC','ENG PAR',NESPCS,0)
39 READ (5,125,END=90) HEAD,TYPENG(I),ATRENG(I),RELENG(I),VEHENG(I),
40 + MULENG(I),VLIENG(I),ADDENG(I),VL2ENG(I),INDENG(I)
41 WRITE (8,125) HEAD,TYPENG(I),ATRENG(I),RELENG(I),VEHENG(I),
42 + MULENG(I),VLIENG(I),ADDENG(I),VL2ENG(I),INDENG(I)
43
44 C
45 C
46 C
47 C
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

```

## INPUT ROUTINES - INSPEC SUBROUTINE

```

ISN      C
48      C READ & STORE PERFORMANCE PARAMETERS
49      I = LSTPER + 1
50      IF (I .GT. NPSPCS) CALL ERR(13,'INSPEC', 'PERF PAR',VPSPCS,0)
51      READ (5,130,END=90) HEAD,TYPPE,ATRPER(I),RELPER(I),
+      VEHPER(I),MULPER(I),VL1PER(I),ADDPER(I),VL2PER(I)
52      +      WRITE (8,130) HEAD,TYPPE,ATRPER(I),RELPER(I),
+      VFHPER(I),MULPER(I),VL1PER(I),ADDPER(I),VL2PER(I)
53      FORMAT (A1,A3,1X,A8,1X,A2,1X,A6,1X,A1,1X,F8.0,1X,A1,1X,F8.0)
54      IF (HEAD .NE. BLS) GO TO 5
55      IF (TYPPE .NE. BLS .AND. TYPPE .NE. VEH) CALL ERR(15,'INSPEC',
+      TYPPE,0,0)
56      LSTPER = I
57      GO TO 40
58      C
59      RETURN
60      END

```

```

      INPUT ROUTINES -   INVEHC SUBROUTINE

      SUBROUTINE INVEHC

      C
      C ROUTINE TO PRODUCE INTERNAL VEHICLE SPECIFICATIONS FROM THE USER'S
      C INPUT SPECIFICATIONS AND A FILE OF EXISTING VEHICLES.
      C

      COMMON /VEH/
      DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
      + FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
      + ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),
      + IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),
      + IPOADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
      + IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
      + IELECT(26,6), ICCMOC(28,9), IFIREX(28,10), IENVIR(26,9),
      + IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
      + IRELN(3,50), VFILE(1), IVFILE(1)
      EQUIVALENCE (VEHICLE(1),IVECLE(1)), (HULL(1),IHULL(1)), (TURRET(1),
      + ITURET(1)), (MAINGN(1),FMAING(1)), (MACGUN(1),FMACGN(1)),
      + (MISGUN(1),FMISGN(1)), (ADGUN(1),IADGUN(1)), (RANGER(1),
      + IRANGR(1)), (SENSOR(1),ISENSR(1)), (STABLE(1),ISTABL(1)),
      + (GUNCTL(1),IGNCTL(1)), (AMMO(1),IAMMO(1)), (AMMOC(1),
      + IAMMOC(1)), (ENGINE(1),IENGIN(1)), (TRANSM(1),ITRANS(1)),
      + (FINLDR(1),IFINDR(1)), (ROADWH(1),IROADW(1)), (SPRING(1),
      + ISPRNG(1)), (TRACK(1),ITRACK(1)), (SKIRT(1),ISKIRT(1)),
      + (FUEL(1),IFUEL(1)), (FUELC(1),IFUELC(1)), (CREW(1),
      + ICREW(1)), (CARGO(1),ICARGO(1)), (ELECTR(1),IELECT(1)),
      + (COMMOC(1),ICMOC(1)), (FIREX(1),IFIREX(1)), (ENVIRC(1),
      + IENVIR(1)), (DIAGNS(1),IDIAGN(1)), (SIGSUP(1),ISIGSP(1)),
      + (SMOKEG(1),ISMOKE(1)), (IEWSYS(1),IEWSYS(1)), (RELN(1),IRELN(1))
      EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
      COMMON /VPROT0/
      REAL*8 FLONAM, RECNAM, DFAULT
      COMMON /JPARS/
      COMMON /JPARS1/
      COMMON /NPARS/
      COMMON /AUX/

```

## INPUT ROUTINES - INVEHC SUBROUTINE

```

11 SN
12 REAL*8 BL8,DASHES
13 EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
14 COMMON /ARRAYS/
15 REAL*8 MANUF,MODEL,NATION
16 DIMENSION ARRAY(1)
17 EQUIVALENCE (ARRAY(1),DETRNG(1))
18 COMMON /JPARS2/
19 COMMON /SPECS/
20 REAL*8 ATRENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
21 COMMON /SPECS2/
22 REAL*8 VEHS
23
24 DIMENSION IALTS(12),VS(12)
25 REAL*8 ATTR,CMP,V,VS,VEHC
26 DATA VEHIC/'VEHICLE'/,VEH/'VEH'/
27
28 CALL NULIFY(ICMPS,NCMPTP)
29 CALL INITVL(VEHS,NVEHS*2,BLS)
30 CALL NULIFY(IRECS,NIRECI#NVR)
31 LSTVEH = 0
32 LVEH = 0
33 IVEHS = NULL
34
35 C
36 C GO THROUGH INTERNAL FILE OF USER SPECIFICATIONS, NOTING THOSE THAT
37 C REFER TO AN EXISTING VEHICLE AND ARRANGING THEM SO THAT THE REQUIRED
38 C INFORMATION CAN BE EXTRACTED DURING ONE PASS THROUGH THE EXISTING
39 C VEHICLE FILE. AT THE SAME TIME, PROCESS ALL SPECIFICATIONS THAT
40 C DON'T REQUIRE ACCESS TO THE EXISTING VEHICLE FILE.
41
42 C
43 C IF (LSTCMP .LE. 0) GO TO 50
44 C ITERATE OVER COMPONENT SPECIFICATION RECORDS
45 C DO 40 I=1,LSTCMP
46 C CHECK THAT COMPONENT TYPE CODE IS LEGAL
47 C IF (IYPCMP(I) .NE. VEH) GO TO 10
48 C THIS RECORD SPECIFIES A BASIC VEHICLE TO START WITH
49 C IF (VEHCMP(I) .EQ. BL8) CALL ERR(19,'INVEHC',I,

```

```

      INPUT ROUTINES - INVEHC SUBROUTINE

      +
      0,0)
C  GET INDEX OF VEHICLE IN VEHICLE NAME TABLE
      K = ITABLE(VEHS,VEHCMP(I),NVEHS,2)
      IF (K.EQ.NULL) CALL ERR(20,'INVEHC',VEHCMP(I),NVEHS,0)
C  FOR EACH COMPONENT OF VEHICLE,
C  GET NEW RECORD TO STORE INFORMATION TO GUIDE LATER FILE RETRIEVAL
C  & STORE INFORMATION IN THIS RECORD
      DO 5 ICMPT=1,NCMPTP
      ICMPS(ICMP) = NEWVEH(ICMPS(ICMP),0,I,K)
      GO TO 40
C
C  GET COMPONENT INDEX
C  1)
      ICMP = LSEARCH(CMPCOD,TPCMP(I),NCMPTP,I)
      IF (ICMP.NE.NULL) GO TO 15
      CALL ERR(22,'INVEHC',TPCMP(I),I,1)
      GO TO 40
C  DOES THIS RECORD INDICATE THE NUMBER OF COMPONENTS OF A PARTICULAR
C  TYPE DESIRED ?
      15
      IF (NUMCMP(I) .LE. 0) GO TO 20
      IF SO, ADJUST MIN & MAXNUM TO THIS NUMBER
      IF (NUMCMP(I) .LT. MINNUM(ICMP) .OR. NUMCMP(I) .GT.
      +
      MAXNUM(ICMP)) CALL ERR(23,'INVEHC',NUMCMP(I),ICMP,I)
      MINNUM(ICMP) = NUMCMP(I)
      MAXNUM(ICMP) = NUMCMP(I)
C
C  IS AN EXISTING VEHICLE REFERRED TO ?
      2)
      IF (VEHCMP(I) .EQ. BL8) GO TO 30
C  GET INDEX OF VEHICLE IN VEHICLE NAME TABLE
      K = ITABLE(VEHS,VEHCMP(I),NVEHS,2)
      IF (K.EQ.NULL) CALL ERR(20,'INVEHC',VEHCMP(I),NVEHS,0)
C  IF SO, GET NEW RECORD TO STORE INFORMATION TO GUIDE LATER FILE
C  RETRIEVAL & STORE INFO IN RECORD
      ICMPS(ICMP) = NEWVEH(ICMPS(ICMP),1,I,K)
      GO TO 40
C
C  COMPONENT SPEC SHOULD REFER TO AN ALTERNATIVE COMPONENT

```

```

      INPUT ROUTINES -   INVEHC SUBROUTINE

ISN 30      IF (INDCMP(I) .GE. 1 .AND. INDCMP(I) .LE. NALYCS(ICMP))
54      +      GO TO 32
55      CALL ERR(24,'INVEHC',INDCMP(I),TYPCMP(I),I)
56      GO TO 40
57      C CREATE DUPLICATE RECORD OF THE DESIRED ALTERNATIVE COMPONENT
32      IRECTP = ICMP + 1
58      IOFF = IFILPT(IIRECTP)
59      IR = NEWREC(IIRECTP)
60      CALL COPY(VFILE(ICFF + (INDCMP(I)-1)*NFLDS(IIRECTP)),
+      VFILE(IOFF + (IR-1)*NFLDS(IIRECTP)),NFLDS(IIRECTP))
C SET NEXT FIELD OF NEW COMPONENT TO THE CURRENT LIST OF
C COMPONENTS
61      CALL STORE(IVECLE(ICMP,ISPECV),VFILE(IOFF),NFLDS(IIRECTP),
+      JNEXT,IR)
C INDICATE COMPONENT TO BE SELECTED
62      IVECLE(ICMP,ISPECV) = IR
63      CONTINUE
C
50      IF (LSTENG .LE. 0) GO TO 100
C ITERATE OVER ENGINEERING PARAMETER CONSTRAINTS SPECIFIED BY USER
65      DO 90 I=1,LSTENG
C
C VERIFY THAT COMPONENT TYPE CODE SPECIFIED IS OK & GET CORRESPONDING
C COMPONENT INDEX
66      ICMP = LSEARCH(CMPCOD,TYPENG(I),NCMPTP,1)
67      IF (ICMP .NE. NULL) GO TO 55
68      CALL ERR(22,'INVEHC',TYPENG(I),I,2)
69      GO TO 90
C
C VERIFY THAT ATTRIBUTE IS SPECIFIED CORRECTLY; GET CORRESPONDING
C INDEX
55      IFLD = IATTR2(ATRENG(I),ICMP + 1)
70      IF (IFLD .NE. NULL) GO TO 60
71      CALL ERR(25,'INVEHC',TYPENG(I),ATRENG(I),I)
72      GO TO 90
73      C

```

## INPUT ROUTINES - INVEHC SUBROUTINE

```

84      C  VERIFY THAT RELCP IS APPROPRIATE
85      DO 65 J=1,NRELOC
86      IF (RELENG(I)) .EQ. RELNAM(J)) GO TO 70
87      CONTINUE
88      CALL ERR(26,'INVEHC',TYPENG(I),ATRENG(I),RELENG(I))
89      GO TO 90
90      KREL = IRELTP(J)
91      C
92      C  IS AN EXISTING VEHICLE REFERRED TO ?
93      IF (VEHENG(I)) .EQ. BL8) GO TO 80
94      C  GET INDEX OF VEHICLE IN VEHICLE NAME TABLE
95      K = ITABLE(VEHS,VEHENG(I),NVEHS,2)
96      IF (K .EQ. NULL) CALL ERR(20,'INVEHC',VEHENG(I),NVEHS,0)
97      IF (K .GT. LVEH) LVEH = K
98      C  IF SO, GET NEW RECORD AND STORE INFO IN IT TO GUIDE LATER FILE
99      C  RETRIEVAL
100     ICMP(1) = NEWVEH(ICMPS(ICMP),2,I,K)
101     GO TO 90
102     C
103     C  ENGINEERING PARAMETER SPEC SHOULD SPECIFY A VALUE. STORE THIS
104     C  AS A CONSTRAINT
105     IR = NEWREL(ICMP+1,IFLD,KREL,VLENG(I),INDENG(I))
106     CONTINUE
107     C
108     C  PROCESS PERFORMANCE PARAMETER SPECIFICATIONS
109     IF (LSTPER .LE. 0) GO TO 150
110     C  ITERATE OVER PERFORMANCE CONSTRAINTS SPECIFIED BY USER
111     DO 140 I=1,LSTPER
112     C
113     C  VERIFY THAT ATTRIBUTE IS SPECIFIED CORRECTLY; GET CORRESPONDING INDEX
114     IFLD = IATTR2(ATRPER(I),1)
115     IF (IFLD .NE. NULL) GO TO 110
116     CALL ERR(25,'INVEHC',VEH,ATRPER(I),0)
117     GO TO 140

```

```

      INPUT ROUTINES -   INVEHC SUBROUTINE

      C
      C  VERIFY THAT RELCP IS APPROPRIATE
104  DO 115 J=1,NRELOP
105  IF (RELPER(I) .EQ. RELNAM(J)) GO TO 120
106  CONTINUE
107  CALL ERR(26,'INVEHC',VEH,ATRP(I),RELPER(I))
108  GO TO 140
109
110  KREL = IRELTP(J)
111
112  C  IS AN EXISTING VEHICLE REFERRED TO ?
113  IF (VEHPER(I) .EQ. BL8) GO TO 130
114  C  GET INDEX OF VEHICLE IN VEHICLE NAME TABLE
115  K = ITABLE(VEHS,VEHPER(I),NVEHS,2)
116  IF (K .EQ. NULL) CALL ERR(20,'INVEHC',VEHPER(I),NVEHS,0)
117  C  IF SO, GET NEW RECORD AND STORE INFO IN IT TO GUIDE LATER FILE
118  C  RETRIEVAL
119  IVEHS = NEWVEH(IVEHS,3,I,K)
120  GO TO 140
121
122  C  PERFORMANCE PARAMETER SPEC SHOULD SPECIFY A VALUE.  STORE THIS AS
123  C  A CONSTRAINT.
124  IR = NEWREL(1,IFLD,KREL,VLLPER(I),0)
125  CONTINUE
126
127  C  READ & EXTRACT INFORMATION FROM EXISTING VEHICLE FILE
128
129  IF (VEHS(1) .EQ. BL8) GO TO 500
130
131  C  READ VEHICLE CATEGORY HEADER AND CHECK AGAINST EXPECTATION
132
133  READ (2,900,END=500) V
134  FORMAT (A8)
135  IF (V .NE. VEHCAT) CALL ERR(18,'INVEHC',V,VEHCAT,0)
136

```

## INPUT ROUTINES - INVEHC SUBROUTINE

```

111      C      DETERMINE THE LAST COMPONENT THAT NEED TO READ INFORMATION ABOUT
112      C
113      C      LCMP = 0
114      C      DO 155 ICMPT=1,NCMPTP
115      C      IF (ICMPS(ICMP) .NE. NULL) LCMP = ICMPT
116      C      155      CONTINUE
117      C
118      C      READ VEHICLE RECORD
119      C      READ (2,910,END=380) V,VS
120      C      910      FORMAT (A8,4X,12(A8,2X))
121      C
122      C      IDENTIFY THE VEHICLES WHICH WANT TO EXTRACT INFORMATION ABOUT
123      C      DO 180 IV=1,LVEH
124      C      DO 165 ICCL=1,NCMPZL
125      C      IF (VS(ICOL) .EQ. VEH$(IV)) GO TO 170
126      C      165      CONTINUE
127      C      GO TO 180
128      C
129      C      FOUND A VEHICLE OF INTEREST; MARK IT FOUND & REMEMBER ITS COLUMN
130      C      170      ICOLS(IV) = ICOL
131      C      180      CONTINUE
132      C
133      C      READ VEHICLE COMPONENT POINTERS
134      C      READ (2,920) CMP,IATLS
135      C      920      FORMAT (2X,A8,12(8X,I2))
136      C      IF (CMP .EQ. BL8 .OR. CMP .EQ. DASHES) GO TO 220
137      C
138      C      IDENTIFY WHETHER THIS IS A COMPONENT OF INTEREST
139      C      DO 200 ICMPT=1,NCMPTP
140      C      IF (ICMPS(ICMP) .NE. NULL .AND. RECNAME(ICMP+1) .EQ. CMP)
141      C      +      GO TO 205
142      C      200      CONTINUE
143      C      GO TO 190
144      C
145      C      COMPONENT IS OF INTEREST; REMEMBER WHICH COLUMN
146      C      EACH COMPONENT OF INTEREST OF THIS TYPE IS IN.

```

```

      INPUT ROUTINES - INVEHC SUBROUTINE

      C  ITERATE OVER CONSTRAINTS INVOLVING THIS COMPONENT & STORE COLUMN
      C  INFORMATION
131      IREC = ICMPS(ICMP)
132      IF (IREC .EQ. NULL) GO TO 190
133      IV = IRECS(JVEH,IREC)
134      ICOL = ICOLS(IV)
135      IF (IRECS(JCOL,IREC) .EQ. NULL) IRECS(JCOL,IREC) = IALTS(ICOL)
136      IREC = IRECS(JNXT,IREC)
137      GO TO 208

      C  READ VEHICLE PERFORMANCE PARAMETERS
138      READ (2,930) ATTR
139      FORMAT (2X,A8)
140      IF (ATTR .EQ. BL8 .OR. ATTR .EQ. DASHES) GO TO 240
      C  IS THIS ATTRIBUTE OF INTEREST ?
141      IREC = IVEHS
142      IF (IREC .EQ. NULL) GO TO 220
143      ISPEC = IRECS(JSPEC,IREC)
144      IF (ATTRPER(ISPEC) .NE. ATTR) GO TO 230
      C  THIS IS ATTRIBUTE OF INTEREST; READ IT
145      CALL INATTR(ATTR,IREC,ISPEC,1)
146      IREC = IRECS(JNXT,IREC)
147      GO TO 225

      C  READ VEHICLE COMPONENT HEADER
148      READ (2,940,END=300) CMP,IALTS
149      FORMAT (A8,2X,12(8X,12))
      C  DETERMINE COMPONENT TYPE
150      DO 250 ICMP=1,NCMP
151      IF (RECNAME(ICMP+1) .EQ. CMP) GO TO 255
152      CONTINUE
153      CALL ERR(28,'INVEHC',CMP,0,0)

      C  IS THIS A COMPONENT OF INTEREST ?
154      IF (ICMP .GT. LCOMP) GO TO 300
155      IF (ICMPS(ICMP) .NE. NULL) GO TO 260

```

## INPUT ROUTINES - INVEHC SUBROUTINE

```

156      C IF NOT, SKIP OVER ITS ATTRIBUTE RECORDS
157      258 READ (2,930,END=300) ATTR
158      IF (ATTR.EQ. BL8 .OR. ATTR.EQ. DASHES) GO TO 240
        GO TO 258

159      C ARE INTERESTED IN THIS COMPONENT. ITERATE OVER USER
160      C SPECIFICATIONS FOR EACH ATTRIBUTE OF THE COMPONENT TO SEE WHETHER
161      C THE ATTRIBUTE IS OF INTEREST. FIRST READ THE NEXT ATTRIBUTE.
        C
        260 IRECTP = ICMP + 1
        262 READ (2,930,END=300) ATTR
        IF (ATTR.EQ. BL8 .OR. ATTR.EQ. DASHES) GO TO 240

162      C IS THIS AN ATTRIBUTE OF INTEREST ?
        IREC = ICMP5(ICMP)
163      IF (IREC.EQ. NULL) GO TO 262
164      ISPEC = IRECS(JSPEC,IREC)
165      ITYP = IRECS(JTYP,IREC)
166      IF (ITYP.EQ. 2) GO TO 270

        C SPECIFICATION CALLS FOR THIS ATTRIBUTE TO BE USED AND STORED
        C AS A CONSTANT
        CALL INATR2(ATTR,IREC,ISPEC,IRECTP)
        GO TO 275

167      C CHECK WHETHER ATTRIBUTE IS OF INTEREST
168      270 IF (ATTR.NE. ATRENG(ISPEC)) GO TO 275

169      C SPECIFICATION CALLS FOR THIS ATTRIBUTE TO BE READ AND STORED AS
        C A CONSTRAINT
        CALL INATR(ATTR,IREC,ISPEC,IRECTP)
        GO TO 275

170      C GET NEXT SPECIFICATION RELEVANT TO THIS COMPONENT
171      275 IREC = IRECS(JNXT,IREC)
172      GO TO 265

173      C CHECK THAT ALL USER SPECIFICATIONS REFERRING TO THE
        C EXISTING VEHICLE FILE HAVE BEEN PROCESSED.
        300 IPREV = NULL

```

```

      ISN
174      INPUT ROUTINES -  INVEHC SUBROUTINE
175
176      DO 310 IV=1,LSTVEH
177      IF (IRECS(JDONE,IV) .NE. NULL) GO TO 310
178      IRECS(JDONE,IV) = IPREV
179      IPREV = IV
180      CONTINUE
181      IF (IPREV .NE. NULL) CALL ERR(32,'INVEHC',IPREV,0,0)
182      RETURN
183      END
184
185      C 380 CALL ERR(33,'INVEHC',LSTVEH,0,0)
186      C 500 RETURN
187      END

```

## C.4: SOLUTION ROUTINES

This section contains listings of the solution routines including the main "backtracking" algorithm (GENVEH), the function OKCOMP used to check compatibility of components with user specifications, the function COMPAT, used to check component compatibility and the routines used to estimate vehicle engineering and performance parameters (DIMENS, FPOWER, MOBILE, and PAIT). See section C-2 for listings of the labeled COMMON blocks referred to in these routines.

# SOLUTION ROUTINES - FUNCTION COMPAT

ISI

1

LOGICAL FUNCTION COMPAT (ICMP,IR)

ROUTINE TO TEST WHETHER A TRIAL COMPONENT OF A GIVEN TYPE IS  
COMPATIBLE WITH PREVIOUS COMPONENTS SELECTED. RETURNS .TRUE. IF SO,  
.FALSE. OTHERWISE.

### INPUT PARAMETERS:

ICMP COMPONENT TYPE

| I.R.      | COMPONENT       |
|-----------|-----------------|
| 1000-1600 | C-H stretching  |
| 1700-1800 | C=O stretching  |
| 1600-1450 | C=C stretching  |
| 1500-1400 | N-H bending     |
| 1300-1200 | C-N stretching  |
| 1100-1000 | C-O stretching  |
| 900-800   | C-H bending     |
| 700-600   | C-F stretching  |
| 600-500   | C-Cl stretching |
| 500-400   | C-Br stretching |
| 400-300   | C-I stretching  |

/HEAD/ COMMON/

```

DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
+ FMACGN(40,10), FWISGN(40,7), IADGUN(40,6), IRANGR(28,7),
+ ISENSR(32,9), ISTABL(30,6), IGNCYL(28,9), IAMMO(30,30),
+ IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),
+ IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
+ IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
+ IELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),
+ IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
+ IRELN(3,50), VFILE(1), IVFILE(1)

```

```

EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
+ ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),
+ (MISGUN(1), FMISGN(1)), (ADGUN(1), TADGUN(1)), (RANGER(1),
+ IRANGER(1)), (SENSOR(1), ISENSOR(1)), (STABLE(1), ISTABL(1)),
+ (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
+ IAMMOC(1)), (ENGINE(1), IENGINE(1)), (TRANSM(1), ITRANS(1)),
+ (PINLDR(1), IFINDR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),
+ ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),
+ (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
+ ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
+ (COMTO(1), ICOMNO(1)), (PIREX(1), IFIREX(1)), (ENVIRC(1),
+ IENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
+ (SMOKEG(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))

```

EQUIVALENCE (VEHICLE(1),VFILF(1),IVFILE(1))

COMMON /VPROTO/

10

15

## SOLUTION ROUTINES - FUNCTION COMPAT

```

15      REAL*8 FLDNAM,RECNAM,DEFAULT
16      COMMON /JPARS/
17      COMMON /JPARS1/
18      COMMON /NPARS/
19      COMMON /AUX/
20      REAL*8 BL8,DASHES
21      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
22      COMMON /SPECS/
23      REAL*8 ATRENG,ATRPER,VEHCAT,VEHCOMP,VEHENG,VEHPER
24
25      COMPAT = .TRUE.
26      GO TO (90,90,10,90,90, 90,90,90,90,90, 20,90,90,30,90,
27      + 90,90,90,90,90, 90,90,90,90,90, 90,90,90,90,90),ICMP
28
29      C CHECK THAT MAIN GUN IS NOT TOO LARGE FOR WEIGHT OF VEHICLE
30      1) WTMIN = FMAING(JMINWT,IR)
31      IF (WTMIN.EQ. FNULL) GO TO 90
32      IF (WTMIN.GT. VEHWT) COMPAT = .FALSE.
33      GO TO 90
34
35      C CHECK THAT ORDNANCE HAS SAME CALIBER AS AT LEAST ONE OF THE GUNS
36      C SELECTED
37      20 CAL = DVAL2(AMMO(JCALIB,IR),ICMP,JCALIB,'MM ')
38      DO 26 IC=JMAING,JADGUN
39      IGR = IVECLE(IC,ICANDV)
40      IF (IGR.EQ. NULL) GO TO 26
41      IRECTP = IC + 1
42      IOFF = IFILPT(IRECTP)
43      GCAL = DVAL(VFILE(IOFF),NFLDS(IRECTP),JCALIB,IGR,IRECTP,'MM ',D)
44      IF (GCAL-CAL.LT..001.AND. CAL-GCAL.LT..001) GO TO 90
45      IGR = IDVAL(VFILE(IOFF),NFLDS(IRECTP),JNEXT,IGR,IRECTP,'- ',D)
46      GO TO 22
47      26 CONTINUE
48      COMPAT = .FALSE.
49      GO TO 90
50
51      C

```

## SOLUTION ROUTINES - FUNCTION COMPAT

```

ISN
35      C  CHECK THAT TRANSMISSION IS COMPATIBLE WITH ENGINE
36      KEYE = IDVAL(ENGINE,NFLDS(JENGINE+1),JTRNRQ,IVECLE(JENGINE,ICANDV),
37      +      JENGINE+1,'-',DF)
38      KEYT = IDVAL2(TRANSM(JKEY,IR),JTRANS,JKEY,'- ')
39      IF (KEYE .NE. KEYT) COMPAT = .FALSE.
40      GO TO 9)

C
39      RETURN
40      END

```

```

1      TSN
2
3      SOLUTION ROUTINES - DIMENS SUBROUTINE
4
5      SUBROUTINE DIMENS(IV,IRETN)
6
7      ROUTINE TO CALCULATE THE DIMENSIONS OF A TANK BASED ON AN ASSUMED
8      GEOMETRICAL CONFIGURATION (SEE ARTICLE BY R. LAWSON IN THE TANK
9      WEAPON SYSTEM, SYSTEM RESEARCH GROUP, OSU, SEPT. 1968)
10     ALSO CALCULATES SELECTED MOBILITY CHARACTERISTICS.
11
12     INPUT PARAMETERS:
13     IV      SPECIFIC VEHICLE OF INTEREST
14
15     OUTPUT PARAMETERS:
16     IRETN   COMPONENT TYPE TO TRY NEW SELECTION IF BACKUP IS NEEDED;
17     SET TO ZERO OTHERWISE.
18
19     COMMON /VEH/
20     DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
21     + FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
22     + ISNSR(32,9), ISTABL(30,6), IGUNCTL(28,9), IAMMO(30,30),
23     + IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),
24     + IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
25     + IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
26     + IELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),
27     + IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
28     + IRELN(3,50), VFULE(1), IVFILE(1)
29
30     EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
31     + ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),
32     + (MISSGN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
33     + IRANGR(1)), (SENSR(1), ISNSR(1)), (STABLE(1), ISTABL(1)),
34     + (GUNCTL(1), IGUNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
35     + IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANSM(1), ITRANS(1)),
36     + (FINLDR(1), IFINDR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),
37     + ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),
38     + (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
39     + ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
40     + (COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),

```

```

      SOLUTION ROUTINES - DIMENS SUBROUTINE

      + IENVIR(1), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
      + (SMOKEG(1), ISMOKE(1)), (EWSYS(1), IENSYS(1)), (RELN(1), IRELN(1))
      EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))
      COMMON /VPROTO/
      REAL*8 PLDVAM, RECNAM, DEFAULT
      COMMON /JPARS/
      COMMON /JPARS1/
      COMMON /NPARS/
      COMMON /AUX/
      REAL*8 BL8, DASHES
      EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
      COMMON /ARRAYS/
      REAL*8 RANGE, MODEL, NATION
      DIMENSION ARRAY(1)
      EQUIVALENCE (ARRAY(1), DETRNG(1))
      COMMON /JPARS2/
      COMMON /DATA/
      COMMON /DATA2/

      C DATA PI/3.14159/

      C GET POINTERS TO COMPONENTS OF VEHICLE
      LTURRET = IVECLE(JTURRET, IV)
      LHULL = IVECLE(JHULL, IV)
      LMAING = IVECLE(JMAING, IV)
      LENGIN = IVECLE(JENGIN, IV)
      LTRANS = IVECLE(JTRANS, IV)
      LFINDR = IVECLE(JFINDR, IV)
      LROADW = IVECLE(JROADW, IV)
      LTRACK = IVECLE(JTRACK, IV)
      LSKIRT = IVECLE(JSKIRT, IV)
      LAMMO = IVECLE(JAMMO, IV)

      C TURRET DIAMETER
      X1 = DVAL2(MAINGN(JX1, LMAING), JMAING, JX1, 'IN ')
      X2 = DVAL2(TURRET(JX2, LTURRET), JTURRET, JX2, 'IN ')

```

## SOLUTION ROUTINES - DIMENS SUBROUTINE

```

15N
34 XTP = 2. * (X1+X2)
35 TURRET(JXTP,LTURET) = XTP

C
C HULL INSIDE WIDTH & VEHICLE WIDTH
36 X5 = DVAL2(ENGINE(JX5,LENGIN),JENGIN,JX5,'IN ')
37 X6 = DVAL2(ENGINE(JX6,LENGIN),JENGIN,JX6,'IN ')
38 X7 = DVAL2(TRANSM(JX7,LTRANS),JTRANS,JX7,'IN ')
39 X8 = DVAL2(FINDER(JX8,LFINDER),JFINDER,JX8,'IN ')
40 XHPT = AMAX1(X5+2.*X6,X7+2.*X8)
41 XH = AMAX1(TURRET(JXTP,LTURET),XHPT)
42 HULL(JXH,LHULL) = XH
43 X3 = DVAL2(HULL(JX3,LHULL),JHULL,JX3,'IN ')
44 X4 = DVAL2(TRACK(JX4,LTRACK),JTRACK,JX4,'IN ')
45 XJ = 2.*X3 + 2.*X4 + XH
46 VEHICLE(JVWID,IV) = XJ

C
C LENGTH OF HULL
47 THFU = DVAL2(HULL(JTHFU,LHULL),JHULL,JTHFU,'IN ')
48 THFL = DVAL2(HULL(JTHFL,LHULL),JHULL,JTHFL,'IN ')
49 GAMMAU = DVAL2(HULL(JGAMU,LHULL),JHULL,JGAMU,'RAD ')
50 COSGU = COS(GAMMAU)
51 TANGH = TAN(GAMMAU)
52 Y1 = THFU / COSGU
53 Y2 = DVAL2(HULL(JY2,LHULL),JHULL,JY2,'IN ')
54 Y3 = DVAL2(HULL(JY3,LHULL),JHULL,JY3,'IN ')
55 K1 = DVAL2(TURRET(JK1,LTURET),JTURET,JK1,'IN ')
56 XTP = DVAL2(TURRET(JXTP,LTURET),JTURET,JXTP,'IN ')
57 Y4 = XTP + K1
58 Y5 = DVAL2(HULL(JY5,LHULL),JHULL,JY5,'IN ')
59 Y6 = DVAL2(ENGINE(JY6,LENGIN),JENGIN,JY6,'IN ')
60 Y7 = DVAL2(TRANSM(JY7,LTRANS),JTRANS,JY7,'IN ')
61 Y8 = DVAL2(TRANSM(JY8,LTRANS),JTRANS,JY8,'IN ')
62 Y9 = DVAL2(HULL(JY9,LHULL),JHULL,JY9,'IN ')
63 YH = Y2 + Y3 + Y4 + Y5 + Y6 + Y7 + Y8
64 HULL(JYH,LHULL) = YH
65 Y0 = Y1 + YH + Y9

```

## SOLUTION ROUTINES - DIMENS SUBROUTINE

```

ISN
66      VEHICLE(JVLEN,IV) = VJ

C
C      TRACK GROUND CONTACT LENGTH
67      D1 = DVAL2(ROADWH(JD1,LOADW),JROADW,JD1,'IN')
68      D2 = DVAL2(ROADWH(JD2,LOADW),JROADW,JD2,'IN')
69      D3 = DVAL2(ROADWH(JD3,LOADW),JROADW,JD3,'IN')
70      T = DVAL2(TRACK(JT,LTRACK),JTRACK,JT,'IN')
71      Z1 = DVAL2(ROADWH(JZ1,LOADW),JROADW,JZ1,'IN')
72      Z14 = DVAL2(ROADWH(JZ14,LOADW),JROADW,JZ14,'IN')
73      Z15 = DVAL2(ROADWH(JZ15,LOADW),JROADW,JZ15,'IN')
74      BF = DVAL2(ROADWH(JBETAF,LOADW),JROADW,JBETAF,'RAD')
75      BR = DVAL2(ROADWH(JBETAR,LOADW),JROADW,JBETAR,'RAD')
76      SINFF = SIN(BF)
77      TANBF = TAN(BF)
78      COSBF = SINBF / TANBF
79      SINBR = SIN(BR)
80      TANBR = TAN(BR)
81      COSBR = SINBR / TANBR
82      TD3Z1 = T + D3/2. - Z1
83      ZZ14 = (SQRT(((D1+D3)/2.))**2 - ((D1-D3)/2.))**2)
      + (D1-D3) / (2.*TANBF)) * SINBF + TD3Z1
84      ZZ15 = (SQRT(((D2+D3)/2.))**2 - ((D2-D3)/2.))**2)
      + (D2-D3) / (2.*TANBR)) * SINBR + TD3Z1
85      IF (Z14 .GE. ZZ14) GO TO 5
86      CALL ERR(40,'DIMENS','Z14','IDLER',Z14)
87      Z14 = ZZ14
88      IF (Z15 .GE. ZZ15) GO TO 10
89      CALL ERR(40,'DIMENS','Z15','SPROCKET',Z15)
90      Z15 = ZZ15
91      C1 = D1/2. + T
92      C2 = D2/2. + T
93      Z1M = Z1 - (T + D3/2.)
94      YGC = Y0 - (C1 + C2 +
      + (2.*COSBF*(Z14+Z1M) - (D1-D3)) / (2.*SINBF) +
      + (2.*COSBR*(Z15+Z1M) - (D1-D3)) / (2.*SINBR))
95      VEHICLE(JYGC,IV) = YGC / 12.

```

## SOLUTION ROUTINES - DIMENS SUBROUTINE

```

104 C
105 C
106 C
107 C
108 C
109 C
110 C
111 C
112 C
113 C
114 C
115 C
116 C
117 C
118 C
119 C
120 C
121 C
122 C
123 C
124 C
125 C
126 C
127 C
128 C

ISN
96 C
97 C
98 C
99 C
100 C
101 C
102 C
103 C

HEIGHTS
Z2 = DVAL2(HULL(JZ2,LHULL),JHULL,JZ2,'IN ')
Z3 = DVAL2(HULL(JZ3,LHULL),JHULL,JZ3,'IN ')
Z4 = DVAL2(HULL(JZ4,LHULL),JHULL,JZ4,'IN ')
Z5 = DVAL2(TURRET(JZ5,LTURET),JTURET,JZ5,'IN ')
Z10 = DVAL2(ENGINE(JZ10,LENGIN),JENGIN,JZ10,'IN ')
Z11 = DVAL2(ENGINE(JZ11,LENGIN),JENGIN,JZ11,'IN ')
Z12 = DVAL2(HULL(JZ12,LHULL),JHULL,JZ12,'IN ')
Z13 = DVAL2(TURRET(JZ13,LTURET),JTURET,JZ13,'IN ')

ALPHAF = DVAL2(MAINGN(JDPRES,LMAING),JMAING,JDPRES,'RAD ')
B = DVAL2(MAINGH(JODIAM,LMAING),JMAING,JODIAM,'IN ')
Y22 = Y1 + Y2 + Y3 + Y4/2. - (Z1+Z2+Z4)*TANGU/2.
Y23 = DVAL2(TURRET(JY23,LTURET),JTURET,JY23,'IN ')
Y24 = Y4/2. + Y5 + Y6 + Y7 + Y8
SINAF = SIN(ALPHAF)
B2SAF = B/(2. * SINAF)
TANAF = TAN(ALPHAF)
COSAF = SINAF / TANAF
X02 = (X0/2.) ** 2
ZZ5 = (SQRT(Y22**2 + X02) + B2SAF - Y23)*TANAF
IF (TURRET(JZ5,LTURET) .EQ. FNULL) GO TO 12
IF (Z5 .GE. ZZ5) GO TO 15
CALL WARN(2,'DIMENS','FRONT',ZZ5,Z5)
Z5 = ZZ5
TURRET(JZ5,LTURET) = Z5
ZZ13 = (SQRT(Y24**2 + X02) + B2SAF - Y23)*TANAF
IF (TURRET(JZ13,LTURET) .EQ. FNULL) GO TO 18
IF (Z13 .GE. ZZ13) GO TO 20
CALL WARN(2,'DIMENS','REAR',ZZ13,Z13)
Z13 = ZZ13
TURRET(JZ13,LTURET) = Z13
Y20 = DVAL2(MAINGN(JY20,LMAING),JMAING,JY20,'IN ')
Y21 = DVAL2(MAINGN(JY21,LMAING),JMAING,JY21,'IN ')
Z6 = (Y20+Y21)*SINAF + B/(2.*COSAF)

```

## SOLUTION ROUTINES - DIMENS SUBROUTINE

```

ISN      C
129      Z7 = DVAL2(TURRET(JZ7,LTURET),JTURET,JZ7,'IN ')
130      Z8 = DVAL2(HULL(JZ8,LHULL),JHULL,JZ8,'IN ')
131      Z9 = DVAL2(HULL(JZ9,LHULL),JHULL,JZ9,'IN ')
132      ZMAX2 = AMAX1(Z3+Z4+Z5+Z6,Z10+Z11+Z12+Z13+Z6,Z8+Z9)
133      Z0 = Z1 + Z2 + ZMAX2 + Z7
134      VEHICLE(JVHT,IV) = Z0

C
C AREA & WEIGHT OF HULL
135      ZH = AMAX1(Z3,Z10+Z11)
136      HULL(JZ0,LHULL) = ZH
137      IARMHL = IHULL(JARMTTP,LHULL)
138      D = DENSTY(IARMHL)
139      AHUF = XH * ZH / (2. * COSGU)
140      WHUF = AHUF * THFU * D

C
141      AUFD = XH * (Y2 + Y3 + Y4/2. - ZH*TANGU/2.) - PI*Y4**2/8.
142      WHFD = AHFD * Z4 * D

C
143      AHRD = XH*(Y4/2. + Y5 + Y6 + Y7) - PI*Y4**2/8.
144      WHRD = AHRD * Z12 * D

C
145      AHUB = XH * ZH / 2.
146      WHUB = AHUB * Y9 * D

C
147      DELTAD = DVAL2(HULL(JDELD,LHULL),JHULL,JDELD,'RAD ')
148      COSDD = COS(DELTAD)
149      TANDD = TAN(DELTAD)
150      AHLB = XH * ZH / (2. * COSDD)
151      WHLB = AHLB * Y9 * D

C
152      GAMMAD = DVAL2(HULL(JGAMD,LHULL),JHULL,JGAMD,'RAD ')
153      TANGD = TAN(GAMMAD)
154      COSGD = COS(GAMMAD)
155      AHB = XH * (YH - ZH*TANDD/2. - ZH*TANGD/2.)
156      WHB = AHB * Z2 * D

```

## SOLUTION ROUTINES - DIMENS SUBROUTINE

```

ISN      C
157      AHS = 2. * YH * ZH - (ZH**2/2.) * (TANDD + TANGD + TANGU)
158      WHS = AHS * X3 * D
159      AHLF = XH * ZH / (2. * COSGD)
160      WHLF = AHLF * THFL * D
161      WH = WHUF + WHFD + WHRD + WHUB + WULB + WHB + WHS + WHLF
162      HULL(JWT,LHULL) = WH
163      C
164      C WEIGHT OF TURRET ARMOR
165      AI = .717 * Y4
166      TURRET(JAI,LTURET) = 2. * AI
167      CI = .59 * Y4
168      BI = Z0 - (Z1 + Z2 + Z3 + Z4 + Z7)
169      TURRET(JBI,LTURET) = 2. * BI
170      IARMTU = IDVAL2(TURRET(JARMTU,LTURET),JTURET,JARMTU,'- ')
171      DT = DENSTY(IARMTU)
172      TTF = DVAL2(TURRET(JTTF,LTURET),JTURET,JTTF,'IN ')
173      TTS = DVAL2(TURRET(JTTS,LTURET),JTURET,JTTS,'IN ')
174      TTU = DVAL2(TURRET(JTTU,LTURET),JTURET,JTTU,'IN ')
175      TTB = DVAL2(TURRET(JTTB,LTURET),JTURET,JTTB,'IN ')
176      WT = .6667*PI*((AI + TTF/2. + TTB/2.) * (BI + Z7) * (CI + TTS)
177      + - (AI*BI*CI))*DT + (4.*PI*AI*CI - PI*Y4**2/4.)*TTU*DT
178      TURRET(JWT,LTURET) = WT
179      C
180      C COMPUTE WEIGHT OF TRACK SKIRTS, IF ANY
181      IF (LSKIRT.EQ.NULL) GO TO 30
182      IARMTU = IDVAL2(SKIRT(JTYPE,LSKIRT),JSKIRT,JTYPE,'- ')
183      DS = DENSTY(IARMTU)
184      HS = DVAL2(SKIRT(JSKIRT,LSKIRT),JSKIRT,JSKIRT,'IN ')
185      TS = DVAL2(SKIRT(JSKTHK,LSKIRT),JSKIRT,JSKTHK,'IN ')
186      WS = 2.*DS*TS*(Y0*HS + Y0*X4 + 2.*X4*HS)
187      SKIRT(JWT,LSKIRT) = WS
188      C
189      C

```

## SOLUTION ROUTINES - DIMENS SUBROUTINE

```

183 C SUM WEIGHT OF ALL COMPONENTS
184   W = SUMCMP(IV,JWT,'LB ')
185   WTON = W / 2000.
      VEHICLE(JVWT,IV) = W

186 C SUM INTERNAL VOLUMES OF COMPONENTS
      VSUM = SUMCMP(IV,JVOL,'FT3 ')

187 C CALCULATE INTERNAL VOLUME OF THE VEHICLE ENVELOPE
      VH = AHS * XH / 2.
      HULI(JVOL,LHULL) = VH / 1728.
      VT = .6667 * PI * AI * BI * CI
      TURRET(JVOL,LTURET) = VT / 1728.
      VOLENV = VH + VT
      VOLFT3 = VOLENV / 1728.
      VEHICLE(JVVOL,IV) = VOLFT3

188 C
189 C MOBILITY PARAMETERS
190 C
191 C TREAD & L/T RATIO
192   TREAD = VEHICLE(JVWID,IV) - X4
193   RLT = YGC / TREAD
      VEHICLE(JLT,IV) = ELT

194 C
195 C GROSS HP / TON
196   HP = DVAL2(ENGINE(JHP,LENGIN),JENGIN,JHP,'HP ')
      HPTON = HP / WTON
      VEHICLE(JGHPTN,IV) = HPTON

197 C
198 C SPROCKET HP / TON
199   EFFTR = DVAL2(TRANSM(JEFFIC,LTRANS),JTRANS,JEFFIC,'- ')
      EFFPD = DVAL2(FINLDR(JEFFIC,LFINDR),JFINDR,JEFFIC,'- ')
      SHPTON = HP * (EFFTR/100.) * (EFFPD/100.) / WTON
      VEHICLE(JSHPTN,IV) = SHPTON

200 C
201 C
202 C
203 C

```

## SOLUTION ROUTINES - DIMENS SUBROUTINE

```

ISN
204 C AVERAGE GROUND PRESSURE
205   P = W / (2.*X4*YGC)
      VEHICLE(JGPRE,IV) = P
C
206 C WT PER LINEAL FOOT
207   WFT = WTON / (YGC/12.)
      VEHICLE(JTONFT,IV) = WFT
C
208 C MAX DITCH THAT CAN CROSS
209   DITCH = (Y0 - (D1/2.)) - (D2/2.) - 2.*T) / 24.
      VEHICLE(JDWID,IV) = DITCH
C
210 C MAX SPEED ON LEVEL ROAD
211   SPD1 = FVAL1(SPEED(1,1),RSHPTN,NSHPTN,SHPTON)
      VEHICLE(JMXSPD,IV) = SPD1
C
212 C MAX SPEED ON 30 DEG GRADE (NO SLIPPAGE)
213   SPD2 = FVAL1(SPEED(1,2),RSHPTN,NSHPTN,SHPTON)
      VEHICLE(J30SPD,IV) = SPD2
C
214 C MAX SLOPE THAT CAN CLIMB
215   SLOPMX = FVAL2(SLOPE,RSHPTN,NSHPTN,SHPTON,RVEHWT,NVEHWT,WTON)
      VEHICLE(JSLOPE,IV) = SLOPMX
C
216 C ACCELERATION SEC FROM 0-20MPH ON LEVEL ROAD
217   ACC = FVAL2(ACCEL,RSHPTN,NSHPTN,SHPTON,RVEHWT,NVEHWT,WTON)
      VEHICLE(JACCEL,IV) = ACC
C
218 C RANGE. FIRST DETERMINE THE TOTAL VOLUME OF FUEL TANKS & THE TOTAL
219   VOLUME OF PROTECTED FUEL TANKS
220   IREC = IVECLE(JFUELC,IV)
221   TFUEL = 0.
222   FUEIP = 0.
223   IF (IREC.EQ. NULL) GO TO 50
      GAL = DVAL2(FUELC(JFCAP,IREC),JFUELC,JFCAP,'GAL ')
      LOC = IDVAL2(IFEUCL(JLOC,IREC),JFUELC,JLOC,' ')

```

## SOLUTION ROUTINES - DIMENS SUBROUTINE

```

ISN
224      IF (LOC .LE. 2) FUEL = FUEL + GAL
225      TFUEL = TFUEL + GAL
226      IREC = IFUEL(JNEXT,IREC)
227      GO TO 45
228      IENGTP = IDVAL2(IENGIN(JTYPE,LENGIN),JENGIN,JTYPE,'-')
229      XNMPG = FVAL2(XMPG(1,1,LENGTP),RSHPTN,NSHPTN,SHPTON,
+      RVEHWT,NVEHWT,WTON)
230      RANGE = XNMPG * TFUEL
231      RANGP = XNMPG * FUEL
232      VEHICLE(JRANGF,IV) = RANGE
233      VEHICLE(JRANGA,IV) = RANGE
C
C      ROUGH TERRAIN VELOCITY
234      NUMWHL = IDVAL2(IROADW(JNRDWH,LROADW),JROADW,JNRDWH,'-')
235      LSPDMP = IVECLE(JSUSP,IV)
236      WHTRV = DVAL2(SPRING(JWHTRV,LSPDMP),JSUSP,JWHTRV,'IN')
237      IP (NUMWHL .LT. MINWHL .OR. NUMWHL .GT. MAXWHL) CALL ERR(41,
+      'DIMENS',NUMWHL,0,0)
238      NW = NUMWHL - MINWHL + 1
239      VRIDE1 = FVAL3(VRIDE(1,1,1,NW,1),RWHTRV,NWHTRV,WHTRV,
+      RVEHWT,NVEHWT,WTON,RTKLEN,NTKLEN,YGC)
240      VRIDE2 = FVAL3(VRIDE(1,1,1,NW,2),RWHTRV,NWHTRV,WHTRV,
+      RVEHWT,NVEHWT,WTON,RTKLEN,NTKLEN,YGC)
C
C      PROBABILITY OF VEHICLE BEING PENETRATED BY A HIT
C
241      CALL INITVL(PPENET(1,1,1,IV),18,FNULL)
242      DO 60 IORD=1,NORDT2
243      DO 60 IASPEC=1,NASPEC
244      DO 60 IRANGE=1,NRANGE
245      PHULL = FVAL1(PHULLP(1,IRANGE,IASPEC,IORD,IARWHL),RARMTH,
+      NARMTH,THFU)
246      PPTURR = FVAL1(PTURRP(1,IRANGE,IASPEC,IORD,IARMTU),RARMTH,
+      NARMTH,TTF)
247      PPENET(IRANGE,IASPEC,IORD,IV) = PHULL*PPHULL + (1.-PPHULL)*
+      PPTURR

```

## SOLUTION ROUTINES - DIMENS SUBROUTINE

ISN

248 60

CONTINUE

I VECLE(JPENTR,IV) = IV

RETURN

END

251

## SOLUTION ROUTINES - PPOWER SUBROUTINE

ISN  
1

SUBROUTINE PPOWER(IV,IKETN)

C  
C ROUTINE TO CALCULATE THE ACCURACY AND LETHALITY OF THE ARMAMENT  
C SYSTEM SELECTED. THIS INCLUDES THE PROBABILITY OF HIT AS A FUNCTION  
C OF RANGE, STATIONARY VS. MOVING FIRER & TARGET, AND ORDNANCE TYPE  
C FOR THE SAME STANDARD TARGET. IF THERE ARE ANY CONSTRAINTS ON THESE  
C VARIABLES SPECIFIED BY THE USER, THE PROGRAM TESTS FOR THEM AND IF  
C THEY ARE NOT SATISFIED RETURNS AN INDICATOR OF WHICH COMPONENT TYPE  
C TO BACK UP TO FOR SELECTING AN ALTERNATIVE CHOICE.

C  
C INPUT PARAMETERS:  
C IV VEHICLE

C  
C OUTPUT PARAMETERS:

C IRETN RETURN CODE: 0 IF ALL CONSTRAINTS SATISFIED; OTHERWISE  
C SET TO THE INDEX OF THE COMPONENT TYPE TO BACK UP TO.

2  
3

COMMON /VEH/

C DIMENSION IVECLE(72,3),IHULL(48,9),ITURF(42,9),FPAING(44,9),  
C FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),  
C ISENSK(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),  
C IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),  
C IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),  
C IFUEL(28,10),IFUEL(26,9),ICREW(26,5),ICARGO(26,5),  
C IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),  
C IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),  
C IREH(3,50),VFILE(1),IVFILE(1)  
C EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),  
C ITURF(1)),(FPAING(1),FPAING(1)),(MACGN(1),FMACGN(1)),  
C (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),  
C IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),  
C (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),  
C IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),  
C (FINDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),  
C ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),

4

## SOLUTION ROUTINES - FPOWER SUBROUTINE

ISW

```

+      (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
+      ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
+      (COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),
+      IENVIR(1)), (DIAGHS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
+      (SMOKEG(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (REIN(1), IREIN(1))
+      EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))
COMMON /JPARS/
COMMON /JPARS1/
COMMON /NPARS/
COMMON /AUX/
REAL*8 BL8, DASHES
EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
COMMON /ARRAYS/
REAL*8 MANUF, MODEL, NATION
DIMENSION ARRAY(1)
EQUIVALENCE (ARRAY(1), DETRNG(1))
COMMON /JPARS2/
COMMON /DATA/
COMMON /DATA2/

C
19      DIMENSION SS(2,3,3)
20      LOGICAL NOMISL
21      DATA KSTGT/1/, KMTGT/2/

C
22      IRETN = 0

C
C      COMPUTE DISPERSION IN THE STATIONARY FIRER - STATIONARY TARGET CASE
C      BY OBTAINING THE ROOT SUM OF SQUARES OF THE DISPERSIONS ASSOCIATED
C      WITH EACH OF THE INDIVIDUAL COMPONENTS INVOLVED.
C
C      FIRST SFT POINTERS TO COMPONENTS INVOLVED.
23      LMAING = IVECLE(JMAING, IV)
24      LMISL = IVECLE(JMISL, IV)
25      LRNGSY = IVECLE(JRNGSY, IV)
26      LSTESY = IVECLE(JSTBSY, IV)
27      LCTLSY = IVECLE(JCTLSY, IV)

```

## SOLUTION ROUTINES - FPOWER SUBROUTINE

```

ISN
28      LAMMO = IVECLE(JAMMO,IV)

C
C      GET POINTERS TO DISPERSION ARRAYS ASSOCIATED WITH EACH COMPONENT
29      IARRMG = MAINGN(JSIGMA,LMAING)
30      IF (LAJSL.NE. NULL) IARRML = MISGUN(JSIGMA,LMISL) + 8
31      IARRRG = IRANGK(JSIGM2,LKNGSY)
32      IARRCL = IGNCPL(JSIGM2,LCTLSY) + 3

C
C      DETERMINE PERFORMANCE CLASS OF STABILIZATION SYSTEM
33      ISTEPF = 1
34      IF (LSTBSY.NE. NULL) ISTEPF = ISTABL(JPERF,LSTBSY) + 1
35      IF (ISTEPF.LT. 1 .OR. ISTEPF.GT. 3) CALL ERR(45,'FPOWER',
+      ISTEPF,1,3)

C
C      DETERMINE WHETHER VEHICLE CAN FIRE MISSILES
36      NOMISL = .TRUE.
C      IS THERE A MISSILE LAUNCHER ON THE VEHICLE ?
37      IF (LMISL.EQ. NULL) GO TO 3
38      NOMISL = .FALSE.
39      GO TO 10

C      IS THERE A GUIDED MISSILE ORDNANCE THAT CAN BE FIRED FROM THE
C      MAIN GUN ?
40      CAL = FMAING(JCALIB,LMAING)
41      IR = LAMMO
42      IF (IR.EQ. NULL) GO TO 10
43      IF (ABS(AMMO(JCALIB,IR)-CAL).GT..001) GO TO 8
44      IF (IAMMO(JGUIDE,IR).LE. 1) GO TO 8
45      NOMISL = .FALSE.
46      GO TO 10
47      IR = IAMMO(JNEXT,IR)
48      GO TO 5

C
C      ITERATE OVER RELEVANT ORDNANCE TYPES AND RANGES, COMPUTING ROOT
C      SUM OF SQUARES OF COMPONENTS' DISPERSIONS.
49      DO 20 IORDTP=1,NORDTP

```

## SOLUTION ROUTINES - FPOWER SUBROUTINE

```

58      ISN
59      IF (IORDTP.EQ. NORDTP .AND. NOMISL) GO TO 20
60      IARR = IARRMG
61      IF (IORDTP.EQ. NORDTP .AND. IMISL.NE. NULL) IARR = IARRML
62      DO 15 IRANGE=1,NRANGE
63      DO 15 IXY=1,NXY
64      SS(IXY,IRANGE,IORDTP) = SIGMA(IXY,IRANGE,IORDTP,IARR)**2
65      + SIGMA2(IXY,IRANGE,IORDTP,IARRG)**2
66      + SIGMA2(IXY,IRANGE,IORDTP,IARRCL)**2
67      CONTINUE
68      CONTINUE
69      C
70      C CALCULATE THE PROBABILITY OF HIT AGAINST A STANDARD TARGET FOR
71      C VARIOUS RANGE, ORDNANCE, AND MOVING / STATIONARY CONDITIONS.
72      C
73      CALL INITVL(PHITS(1,1,1,IV),18,FNULL)
74      CALL INITVL(PHITM(1,1,1,IV),18,FNULL)
75      CALL INITVL(PK(1,1,1,IV),18,FNULL)
76      DO 40 IORDTP=1,NORDTP
77      IF (IORDTP.EQ. NORDTP .AND. NOMISL) GO TO 40
78      DO 35 IRANGE=1,NRANGE
79      XSS2 = SS(1,IRANGE,IORDTP)
80      YSS2 = SS(2,IRANGE,IORDTP)
81      XSS = SQRT(XSS2)
82      YSS = SQRT(YSS2)
83      XMF2 = SIGMVF(1,IRANGE,IORDTP,ISTBPF)
84      YMF2 = SIGMVF(2,IRANGE,IORDTP,ISTBPF)
85      XMS = SQRT(XSS2 + XMF2)
86      YMS = SQRT(YSS2 + YMF2)
87      XMT2 = SIGMVT(1,IRANGE,IORDTP,1)
88      YMT2 = SIGMVT(2,IRANGE,IORDTP,1)
89      XSM = SQRT(XSS2 + XMT2)
90      YSM = SQRT(YSS2 + YMT2)
91      XMM = SQRT(XSS2 + XMF2 + XMT2)
92      YMM = SQRT(YSS2 + YMF2 + YMT2)
93      PHITS(IRANGE,KSTGT,IORDTP,IV) = PHIT(XSS,YSS)
94      PHITS(IRANGE,KMTGT,IORDTP,IV) = PHIT(XSM,YSM)

```

```

      SOLUTION ROUTINES - FPOWER SUBROUTINE

      PHITM(IRANGE,KSTGT,IORDTP,IV) = PHIT(XMS,YMS)
      PHITM(IRANGE,KMTGT,IORDTP,IV) = PHIT(XMN,YMN)

C
C  CALCULATE THE PROBABILITY OF A KILL GIVEN A HIT AGAINST THE TARGET
      DO 30 IASPEC = 1,NASPEC
      PK(IRANGE,IASPEC,IORDTP,IV) = FVAL1(PKDAT(1,IRANGE,
      +   IASPEC,IORDTP),RDISPR,NDISPR,(XSS+YSS)/2.)
      CONTINUE
      CONTINUE
      CONTINUE
      IVECLE(JPHITS,IV) = IV
      IVECLE(JPHITM,IV) = IV
      IVECLE(JPK,IV) = IV
      RETURN
      END

```

## SOLUTION ROUTINES - GENVEH SUBROUTINE

ISN

1

SUBROUTINE GENVEH

C

ROUTINE TO GENERATE THE "BEST" CONCEPT VEHICLE FROM THE COMPONENTS  
C AVAILABLE AND MATCHING THE CONSTRAINTS SPECIFIED.

C

COMMON /VEH/

2

DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),

3

+ FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),

+ ISENSOR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),

+ IAMMOC(26,9), IENGINE(40,8), ITRANS(32,8), IFINDR(28,10),

+ IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),

+ IFUEL(28,13), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),

+ IELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),

+ IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),

+ IRELN(3,50), VFILE(1), IVFILE(1)

4

EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURET(1),

+ ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),

+ (HISGUN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),

+ IRANGR(1)), (SENSOR(1), ISENSOR(1)), (STABLE(1), ISTABL(1)),

+ (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),

+ IAMMOC(1)), (ENGINE(1), IENGINE(1)), (TRANSM(1), ITRANS(1)),

+ (FINLDR(1), IFINDR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),

+ ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),

+ (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),

+ ICRFW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),

+ (COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),

+ IENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),

+ (SMOKEG(1), ISMOKE(1)), (IEWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))

EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))

COMMON /VPROTO/

REAL\*8 FLDNAM, RECNAM, DEFAULT

COMMON /JPARS/

COMMON /JPARS1/

COMMON /NPARS/

COMMON /AUX/

11

```

11          SOLUTION ROUTINES - GENVEH SUBROUTINE
12
13      REAL*8 EL8,DASHES
14      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
15
16      DIMENSION JJ(35),JCT(35),NUM(35),MATCH(35),NUMSPC(35)
17      LOGICAL OKCMP,COMPAT
18
19      ORDER THE COMPONENTS OF EACH TYPE IN DECREASING ORDER BY SCORE
20      CALL CSCORE
21
22      DETERMINE NUMBER OF USER SPECIFICATIONS FOR EACH COMPONENT
23
24      CALL IZERO(NUMSPC,NCMP)
25      DO 2 ICMP=1,NCMP
26      IF (IVECLE(ICMP,ISPECV) .EQ. NULL) GO TO 2
27      IOFF = IFILPT(ICMP+1)
28      NUMSPC(ICMP) = NLIST(VFILE(IOFF),NFLDS(ICMP+1),JNEXT,
29      + IVECLE(ICMP,ISPECV))
30      CONTINUE
31
32      -- VEHICLE SPECIFICATION FILE CONTAINS INFORMATION
33      -- ABOUT THE VEHICLE MODEL WHICH INTERFERES WITH
34      -- THE MATCHING OF SPECIFICATION TO ALTERNATIVES COMPONENTS.
35      -- THEREFORE, THE JMODEL FIELD MUST BE NULL BEFORE
36      -- THE BEST VEHICLE CAN BE GENERATED.
37
38      DO 4 ICMP=1,NCMP
39      IR = IVECLE(ICMP,ISPECV)
40      3 IF (IR .EQ. NULL) GO TO 4
41      IOFF = IFILPT(ICMP+1)
42      NF = NFLDS(ICMP+1)
43      CALL STORE(NULL,IVFILE(IOFF),NF,JMODEL,IR)
44      IR = IVAL(IVFILE(IOFF),NF,JNEXT,IR)
45      GO TO 3
46      4 CONTINUE
47
48      C
49      C
50      C
51      C
52      C
53      C
54      C
55      C
56      C
57      C
58      C
59      C
60      C
61      C
62      C
63      C
64      C
65      C
66      C
67      C
68      C
69      C
70      C
71      C
72      C
73      C
74      C
75      C
76      C
77      C
78      C
79      C
80      C
81      C
82      C
83      C
84      C
85      C
86      C
87      C
88      C
89      C
90      C
91      C
92      C
93      C
94      C
95      C
96      C
97      C
98      C
99      C
100     C

```



# SOLUTION ROUTINES - GENVEH SUBROUTINE

```

C
C CALCULATE ANY PERFORMANCE PARS APPROPRIATE AFTER SELECTION OF THIS
C COMPONENT TYPE
C
47 IRETN = IPFORM(ICMP)
48 IF (IRETN .EQ. 0) GO TO 10
C
C IF BACKTRACKING IS INDICATED, DO SO
C
C IF (IRETN .GE. 1) CALL ERR(44,'GENVEH',I,IRETN,ICMP)
50 DO 11 K=IRETN,1
51 CALL UNHOOK(ICMPRK(K))
52 CONTINUE
53 I = IRETN
54 GO TO 22
C
C PROCESS NEXT COMPONENT TYPE IF THERE IS ONE
C
55 I = I + 1
56 IF (I .GT. NCMPTP) RETURN
57 GO TO 5
C
C COMPONENT DIDN'T SATISFY USER SPECS. KEEP NOTE OF NUMBER OF SUCH
C COMPONENTS.
C
58 JCT(I) = JCT(I) + 1
C
C GET NEXT COMPONENT OF CURRENT TYPE, IF THERE IS ONE
15 JI = IVAL(VFILE(IOFF),NF,JSCORE,JI)
60 JJ(I) = JI
61 IF (JJ .NE. NULL) GO TO 8
C
C CAN FAILURE TO FIND A SUITABLE COMPONENT POSSIBLY BE AVOIDED BY
C BACKTRACKING ?
C
C IF (JCT(I)+MINNUM(ICMP) .LE. NALTC5(ICMP)) GO TO 20

```

## SOLUTION ROUTINES - GENVEH SUBROUTINE

```

ISN
63      CALL ERR(42,'GENVEH',ICMP,JI,JCT(I))
64      PAUSE

C
C      BACKTRACK TO PREVIOUS COMPONENT TYPE
C
20      CALL UNHOOK(ICMP)
65      I = I - 1
66      IF (I .GE. 1) GO TO 25
67      CALL ERR(43,'GENVEH',ICMP,0,0)
68      PAUSE
69

C
C      TRY NEXT CHOICE OF PREVIOUS COMPONENT TYPE
C
C
25      ICMP = ICMPRK(I)
70      IRECTP = ICMP + 1
71      NF = NFIDS(IRECTP)
72      IOFF = IFILPT(IRECTP)
73      JI = IVAL(VFILE(IOFF),NF,JSCORE,JJ(I))
74      IF (JI .EQ. NULL) GO TO 20
75      NUM(I) = J
76      CALL UNHOOK(ICMP)
77      MATCH(I) = 0
78      JJ(I) = JI
79      GO TO 8
80

C
81      END
82

C
C      FUNCTION IPFORM(ICMP)
C
C      ROUTINE TO SUPERVISE THE CALCULATION OF PERFORMANCE VARIABLES.
C      THE ROUTINE IS CALLED AFTER EACH COMPONENT OF A GIVEN TYPE IS
C      SELECTED FOR THE CANDIDATE VEHICLE. IF THIS IS AN APPROPRIATE TIME
C      TO CALCULATE A GIVEN PERFORMANCE PARAMETER (I.E., ALL RELEVANT
C      INDEPENDENT VARIABLES ARE KNOWN AT THIS POINT OR CAN BE ESTIMATED
C      SUFFICIENTLY WELL TO TEST A CONSTRAINT ON A PERFORMANCE PARAMETER),
C      THE ROUTINE DOES SO. WHENEVER THE ROUTINE CALCULATES A PERFORMANCE
C      PARAMETER IT CHECKS WHETHER IT SATISFIES ANY CONSTRAINTS THAT MAY BE

```

## SOLUTION ROUTINES - GENVEH SUBROUTINE

```

83      C IMPOSED ON IT, EITHER BY THE USER OR FROM THE PROGRAM'S KNOWLEDGE OF
84      C VEHICLE DESIGN. IF A PERFORMANCE PARAMETER DOES NOT MEET AN IMPOSED
      C CONSTRAINT, THE CODE INDICATES HOW FAR TO BACKUP TO TRY TO OBTAIN A
      C BETTER MATCH (I.E., WHAT COMPONENT TYPE TO RETURN TO TO TRY TO SELECT
      C A DIFFERENT CHOICE). THE FUNCTION RETURNS THE INDEX OF SUCH A
      C COMPONENT IF ANY TESTED CONSTRAINTS FAIL, OR IT RETURNS ZERO
      C OTHERWISE.
      C
      C INPUT PARAMETERS:
      C   ICMP   TYPE OF COMPONENT JUST SELECTED FROM CANDIDATE VEHICLE
      C
      C   COMMON /VEH/
      C   DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
      C   + FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
      C   + ISENSR(32,9), ISTABL(30,6), IGHCTL(28,9), IAMMO(30,3),
      C   + IAMMOC(26,9), IENGINE(40,8), ITRANS(32,8), IFINDR(28,10),
      C   + IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
      C   + IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
      C   + IELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),
      C   + IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
      C   + IRELN(3,50), VFILE(1), IVFILE(1)
      C   EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURET(1),
      C   + ITURET(1)), (MAINGN(1), FMAING(1)), (MACGN(1), FMACGN(1)),
      C   + (MISGN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
      C   + IRANGR(1)), (SENSOR(1), ISENSR(1)), (STABLE(1), ISTABL(1)),
      C   + (GUNCTL(1), IGUNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
      C   + IAMMOC(1)), (ENGINE(1), IENGINE(1)), (TRANSM(1), ITRANS(1)),
      C   + (FINDR(1), IFINDR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),
      C   + ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),
      C   + (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
      C   + ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
      C   + (COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),
      C   + IENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
      C   + (SMOKEG(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))
      C   EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))
      C   COMMON /JPARS/

```

## SOLUTION ROUTINES - GENVEH SUBROUTINE

```

15N
88 COMMON /JPARS1/
89 COMMON /AUX/
90 REAI*8 EL8,DASHES
91 EQUIVALENCE (FNULL,NULL), (JELS,BLS)
92 COMMON /SPECS/
93 REAI*8 ATRENG,ATRPER,VEHCAT,VEHCOMP,VEHENG,VEHPER
      C
94 LOGICAL OKVAL
      C
95 IPFCRM = 0
96 GO TO (10,20,10,10,10, 10,10,10,10,10, 60,10,80,10,100,
      + 10,10,10,10,10, 10,10,10,10,10, 10,10,10,10,10, 160),ICMP
      C
97 RETURN
      C
98 HAVE JUST SELECTED TURRET. CAN NOW COMPUTE THE GEOMETRY, WEIGHT,
99 PROTECTION, AND MANY OF THE MOBILITY PARAMETERS IN FINAL FORM.
      C
      C 20 CALI DIMENS(ICANDV,IPFORM)
      C RETURN
      C
      C HAVE JUST SELECTED ARMAMENT COMPONENTS. COMPUTE ACCURACY &
      C LETHALITY PARAMETERS AND CHECK THAT THEY SATISFY ANY USER-PROVIDED
      C CONSTRAINTS.
      C
100 CALI FPOWER (ICANDV,IPFORM)
101 RETURN
      C
      C HAVE JUST SELECTED THE ENGINE. CAN MAKE QUICK TEST THAT HAVE
      C SUFFICIENT GROSS HP / TON.
102 IF (VEHICLE(JGHPTN,ISPECV) .EQ. FNULL) RETURN
103 LENGIN = IVECLE(JJENGIN,ISPECV)
104 HP = ENGINE(JHP,LENGIN)
105 IF (OKVAL(HP/VEHWT,1,JGHPTN,NULL)) RETURN
106 IPFORM = ICMP
107 RETURN
      C
      C HAVE JUST SELECTED THE FINAL DRIVE. CAN MAKE INITIAL ESTIMATE OF

```

## SOLUTION ROUTINES - GENVEH SUBROUTINE

```

ISN
108 C POWER LIMITED MOBILITY PARAMETERS & BACKTRACK IF THESE DON'T MEET
109 C CONSTRAINTS.
109 100 CALL MOBILE(ICANDV,IPFORM)
      RETURN
      C
110 C HAVE SELECTED ALL COMPONENTS. CHECK RAM/D & COST CONSTRAINTS.
111 C (THIS PORTION NOT IMPLEMENTED).
      160 RETURN
      END

```

## SOLUTION ROUTINES - MOBILE SUBROUTINE

ISN

1

SUBROUTINE MOBILE(IV,IRETN)

C ROUTINE TO CALCULATE SELECTED POWER-CONSTRAINED MOBILITY  
 C PARAMETERS TO SEE WHETHER THE CHOICE OF POWER TRAIN COMPONENTS IS  
 C SATISFACTORY IF THE USER HAS SPECIFIED ANY CONSTRAINTS ON THESE  
 C PARAMETERS.

C INPUT PARAMETERS:

C IV VEHICLE OF INTEREST

C OUTPUT PARAMETERS:

C IRETN COMPONENT RANK INDEX TO BACK UP TO IF THE USER'S  
 C CONSTRAINTS ARE NOT SATISFIED; 0 OTHERWISE.

2

COMMON /VEH/

3

DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),  
 + FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),  
 + ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),  
 + IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),  
 + TROADW(36,10), ISPENG(30,10), ITRACK(34,8), ISKIRT(28,8),  
 + IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),  
 + IELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),  
 + IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),  
 + IRELN(3,50), VFILE(1), IVFILE(1)  
 EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),  
 + ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),  
 + (MISGUN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),  
 + IRANGR(1)), (SENSOR(1), ISENSR(1)), (STABLE(1), ISTABL(1)),  
 + (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),  
 + IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANSM(1), ITRANS(1)),  
 + (FINLDR(1), IFINDR(1)), (ROADWH(1), TROADW(1)), (SPRING(1),  
 + ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),  
 + (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),  
 + ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),  
 + (COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),

4

```

      SCLUTION ROUTINES - MOBILE SUBROUTINE

      + IENVIR(1), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
      + (SMOKEG(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))
      EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))
      COMMON /VPROTO/
      REAL*8 PLDNAM, RECNAM, DEFAULT
      COMMON /JPARS/
      COMMON /JPARS1/
      COMMON /HPARS/
      COMMON /AUX/
      REAL*3 EL3, DASHES
      EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
      COMMON /SPECS/
      REAL*8 ATRFNG, ATRPER, VEHCAT, VEHCMP, VEHENG, VEHPER
      COMMON /DATA/
      COMMON /DATA2/

      LOGICAL OKVAL
      IRETN = 0

      LENGIN = IVECLE(JENGIN, IV)
      LTPANS = IVECLE(JTRANS, IV)
      LFINDR = IVECLE(JFINDR, IV)

      CHECK GROSS HP / TON
      HP = DVAL2(ENGINE(JHP, LENGIN), JENGIN, JHP, 'HP ')
      HPTON = HP / VEHWT
      IF (.NOT.OKVAL(HPTON, 1, JGHPTN, NULL)) GO TO 999

      CHECK SPROCKET HP / TON
      EFFTR = DVAL2(TRANSM(JEFFIC, LTRANS), JTRANS, JEFFIC, '---')
      EFFFD = DVAL2(FINDR(JEFFIC, LFINDR), JFINDR, JEFFIC, '---')
      SHPTON = HP * EFFTR * EFFFD / VEHWT
      IF (.NOT.OKVAL(SHPTON, 1, JSHTPN, NULL)) GO TO 999

      CHECK SPEED ON LEVEL ROAD IF CONSTRAINED

```

## SOLUTION ROUTINES - MOBILE SUBROUTINE

```

ISN
30      IF (IVECLE(JMXSPD,ISPECV) .EQ. NULL) GO TO 10
31      SPD1 = FVAL1(SPEED(1,1),RSHPTN,NSHPTN,SHPTON)
32      IF (.NOT.OKVAL(SPD1,1,JMXSPD,NULL)) GO TO 999
C
33      C CHECK SPEED ON 30 DEGREE GRADE, IF CONSTRAINED
34      IF (IVECLE(J30SPD,ISPECV) .EQ. NULL) GO TO 20
35      SPD2 = FVAL1(SPEED(1,2),RSHPTN,NSHPTN,SHPTON)
36      IF (.NOT.OKVAL(SPD2,1,J30SPD,NULL)) GO TO 999
C
37      C CHECK MAX SLOPE THAT TANK CAN CLIMB, IF THIS IS CONSTRAINED
38      IF (IVECLE(JSLOPE,ISPECV) .EQ. NULL) GO TO 30
39      SLOPMX = FVAL2(SLOPE,RSHPTN,NSHPTN,SHPTON,RVEHWT,NVEHWT,VEHWT)
40      IF (.NOT.OKVAL(SLOPMX,1,JSLOPE,NULL)) GO TO 999
C
41      C CHECK MAX ACCELERATION THAT VEHICLE CAN ACHIEVE, IF CONSTRAINED
42      IF (IVECLE(JACCEL,ISPECV) .EQ. NULL) RETURN
43      ACC = FVAL2(ACCEL,RSHPTN,NSHPTN,SHPTON,RVEHWT,NVEHWT,VEHWT)
44      IF (.NOT.OKVAL(ACC,1,JACCEL,NULL)) GO TO 999
45      RETURN
C
46      C HAVE FOUND AT LEAST ONE POWER-LIMITED MOBILITY PARAMETER CONSTRAINT
47      C THAT WAS NOT SATISFIED. BACKUP ONE. TRY NEW FINAL DRIVE.
999      DO 1000 IRETN=1,NCMPTP
1000      IF (ICMPRK(IRETN) .EQ. JFINDR) RETURN
1000      CONTINUE
1000      PAUSE
1000      END

```

## SOLUTION ROUTINES - FUNCTION OKCMP

ISN  
1

LOGICAL FUNCTION OKCMP(ICMP,JI,IFILE,NF,MATCH,NUM,NUMSPC)

ROUTINE TO TEST WHETHER A SELECTED COMPONENT IS COMPATIBLE WITH THE USER SPECIFICATIONS. THE SELECTED COMPONENT IS DEEMED COMPATIBLE WITH THE USER SPECIFICATIONS IF EITHER OF THE FOLLOWING CONDITIONS ARE MET: (1) THE USER HAS INDICATED A SPECIFIC COMPONENT TO BE SELECTED AND THE COMPONENT IN QUESTION IS THAT COMPONENT, AND (2) THERE IS AT LEAST ONE COMPONENT OF THE CURRENT TYPE THAT MAY BE SELECTED WITHOUT EXCEEDING THE MAX NUMBER OF COMPONENT CONSTRAINTS AND IF THERE ARE ANY SETS OF ENGINEERING CONSTRAINTS SPECIFIED WHICH HAVEN'T YET BEEN PAIRED WITH A SPECIFIC COMPONENT SELECTION, ONE SUCH SET CAN BE PAIRED WITH THE CURRENT SELECTION SUCH THAT NO CONSTRAINTS ARE VIOLATED. IN THESE CASES THE FUNCTION RETURNS A .TRUE. VALUE; OTHERWISE IT RETURNS .FALSE.

## INPUT PARAMETERS:

ICMP COMPONENT TYPE  
JI TRIAL COMPONENT TO BE TESTED FOR COMPATIBILITY  
IFILE FILE IN WHICH COMPONENTS ARE LOCATED  
NF NUMBER OF FIELDS IN RECORDS OF THIS FILE  
MATCH NUMBER OF MATCHES OBTAINED BETWEEN COMPONENTS OF THIS TYPE SPECIFIED BY THE USER AND COMPONENTS SELECTED SO FAR  
NUM NUMBER OF COMPONENTS OF THIS TYPE ALREADY SELECTED  
NUMSPC NUMBER OF COMPONENTS OF THIS TYPE REFERRED TO BY THE USER SPECIFICATIONS

## OUTPUT PARAMETERS:

MATCH UPDATED BY 1 IF THE CANDIDATE COMPONENT CAN BE PAIRED WITH AN AS YET UNMATCHED USER SPECIFICATION  
NUM UPDATED BY 1 IF THE CANDIDATE COMPONENT IS ACCEPTABLE.

## COMMON /VEH/

DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),  
+ FMACGN(40,10), FMISSGN(40,7), IADGUN(40,6), IRANGR(28,7),  
+ ISENSF(32,9), ISTAEL(30,6), IGUNCTL(28,9), IAMMO(30,30),

2  
3

## SOLUTION ROUTINES - FUNCTION OKCMP

```

ISN      4      IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),
+      IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
+      IFUEL(28,10), IFUEL(26,9), ICREW(26,5), ICARGO(26,5),
+      IELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),
+      IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
+      IRELN(3,50), VFILE(1), IVFILE(1)
+      EQUIVALENCE (VEHICLE(1), FVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
+      ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),
+      (MISGUN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
+      IRANGR(1)), (SENSOR(1), ISENSE(1)), (STABLE(1), ISTABL(1)),
+      (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
+      IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANSM(1), ITRANS(1)),
+      (FINLDR(1), IFINDR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),
+      ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),
+      (FUEL(1), IFUEL(1)), (FUEL(1), IFUEL(1)), (CREW(1),
+      ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
+      (COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),
+      IENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
+      (SMOKFG(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))
+      EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))
+      COMMON /VPROTC/
+      REAL*8 FLDNAM, RECNAM, DEFAULT
+      COMMON /JPARS/
+      COMMON /JPARS1/
+      COMMON /NPAES/
+      COMMON /AUX/
+      REAL*8 BL8, DASHES
+      EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
C
C      EQUIVALENCE (JMATCH, JMODEL)
C      USE AN UNUSED FIELD TO POINT FROM A USER SPEC RECORD TO THE
C      ACTUAL COMPONENT PAIRED WITH IT.
C
C      DIMENSION IFILE(NF,1)
C      LOGICAL OKVAL, IOKVAL
C      DATA NCMPMX/50/

```

## SOLUTION ROUTINES - FUNCTION OKCMP

```

18      C      OKCMP = .TRUE.
19      C      IRECTP = ICMP + 1
20      C      C ARE THERE ANY USER SPECIFICATIONS ?
21      C      IF (IVECLE(ICMP,ISPECV) .EQ. NULL) NUM=NUM+1
22      C      IF (IVECLE(ICMP,ISPECV) .EQ. NULL) RETURN
23      C      C IS THE CURRENT COMPONENT SPECIFICALLY REQUESTED BY THE USER ?
24      C      IR = INLIST(IFILE(JID,JI),IFILE,NF,JID,JNEXT,IVECLE(ICMP,ISPECV))
25      C      IF (IR .EQ. NULL) GO TO 10
26      C      MATCH = MATCH + 1
27      C      NUM = NUM + 1
28      C      IFILE(JMATCH,IR) = JI
29      C      RETURN
30      C      C ARE THERE ALREADY TOO MANY COMPONENTS ?
31      C      IF (NUM + (NUMSPC - MATCH) .LE. MAXNUM(ICMP)) GO TO 12
32      C      CKCMP = .FALSE.
33      C      RETURN
34      C      C ITERATE OVER ANY UNMATCHED SETS OF ENGINEERING PARAMETER CONSTRAINTS
35      C      IR = IVECLE(ICMP,ISPECV)
36      C      C SKIP OVER ANY SPECIFICATIONS THAT REFER TO CONSTANT COMPONENTS
37      C      IF (IFILE(JID,IR) .LT. NCMPX) GO TO 30
38      C      C SKIP OVER ANY SPECIFICATIONS THAT HAVE ALREADY BEEN MATCHED TO
39      C      C ANOTHER COMPONENT
40      C      IF (IFILE(JMATCH,IR) .NE. NULL) GO TO 30
41      C      C DOES THE TRIAL COMPONENT SATISFY THIS SET ?
42      C      IOFFR = IFILPT(ICMP+1)
43      C      DO 20 IFLD=2,NF
44      C      IF (IFLD .EQ. JID) GO TO 20
45      C      IVF = IVAL(IVFILE(IOFFR),NF,IFLD,IP)
46      C      IF (IVF .EQ. NULL) GO TO 20

```

## SOLUTION ROUTINES - FUNCTION OKCMP

```

ISM
39 C HAVE FOUND A CONSTRAINT. TEST IT. FIRST DETERMINE TYPE OF FIELD.
40 KFLD = IFIDPT(IRECTP) + IFLD
    ITPP = IFIDTP(KFLD)
41 C CONSTRAINTS ON ARRAYS OR CHARACTER FIELDS ARE NOT IMPLEMENTED
    IF (ITYP .EQ. ITPPES(3) .OR. ITPP .EQ. ITPPES(4)) GO TO 20
42 C IS TYPE REAL ?
    IF (ITYP .NE. ITPPES(1)) GO TO 15
43 C IF SO GET VALUE
    V = DVAL2(IFILE(IFLD,JI),ICMP,IFLD,UNITS(KFLD))
44 IF (.NOT.OKVAL(V,IRECTP,IFLD,IR)) GO TO 30
45 GO TO 20
46 C TYPE IS INTEGER
    IV = IFILE(IFLD,JI)
47 IF (IV .EQ. NULL) IV = DEFAULT(KFLD)
48 IF (.NOT.IOKVAL(IV,IRECTP,IFLD,IR)) GO TO 30
49 CONTINUE
50 C
51 C COMPONENT DOES INDEED SATISFY THIS SET OF ENGINEERING CONSTRAINTS
    NUM = NUM + 1
52 MATCH = MATCH + 1
53 IFILE(JMATCH,IR) = JI
    RETURN
54 C
55 C GET NEXT USER SPECIFICATION
    IR = IFILE(JNEXT,IR)
    IF (IR .NE. NULL) GO TO 14
56 C
57 C COMPONENT INCOMPATIBLE WITH ENGINEERING CONSTRAINTS
    OKCMP = .FALSE.
58 RETURN
    END

```

```

1      ISN
2      SOLUTION ROUTINES - PHIT FUNCTION
3      FUNCTION PHIT(XSIG,YSIG)
4
5      ROUTINE TO CALCULATE THE PROBABILITY OF A HIT AGAINST A TARGET OF
6      A STANDARD SIZE AND SHAPE HAVING A SPECIFIED AIM POINT, GIVEN A
7      STANDARD DEVIATION OF THE FIRE IN THE X AND Y DIRECTIONS.  THE
8      DISPERSION OF FIRE IS ASSUMED TO FOLLOW A BIVARIATE NORMAL
9      DISTRIBUTION.  THIS VERSION OF THE ROUTINE ASSUMES A SQUARE TARGET
10     2.34 M ON A SIDE WITH THE AIM POINT IN THE MIDDLE.  THE STANDARD
11     DEVIATIONS OF THE FIRING DISTRIBUTION IS ALSO ASSUMED TO BE
12     EXPRESSED IN METERS.
13
14     DATA RSQRT2/.70711/,XFGT/1.17/,YFGT/1.17/
15
16     IF (XSIG .LT. 1.E-6) XSIG = 1.E-6
17     IF (YSIG .LT. 1.E-6) YSIG = 1.E-6
18     PHIT = ERF(RSQRT2*XTGT/XSIG) * ERF(RSQRT2*YTGT/YSIG)
19     RETURN
20     END

```

#### C.5: OUTPUT ROUTINES

This section contains listings of routines used to format and write the output tables which display the results of a model run.

```

ISN
1
      OUTPUT ROUTINES - OUTARR SUBROUTINE

      SUBROUTINE OUTARR(ARR, IRECTP, IFLD)

      ROUTINE TO OUTPUT THE VALUES ASSOCIATED WITH AN ARRAY WHICH
      IS THE VALUE OF A RECORD'S FIELD

      INPUT PARAMETERS:
      ARR      ARRAY OF INTEREST
      IRECTP   RECORD TYPE OF RECORD POINTING TO ARRAY
      IREC     RECORD POINTING TO ARRAY
      IFLD     FIELD OF RECORD POINTING TO ARRAY

      COMMON /VPROTO/
      REAL*8 FLDNAM, RECNAM, DEFAULT
      COMMON /NPARS/
      COMMON /AUX/
      REAL*8 BL8, CASHES
      EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
      COMMON /ARRAYS/
      REAL*8 MANUF, MODEL, NATION
      DIMENSION ARRAY(1)
      EQUIVALENCE (ARRAY(1), DETPNG(1))
      COMMON /JPARS2/

      DIMENSION ARR(1), ILEVS(3), NLEVS(3)

      GET PARAMETERS OF INTEREST ASSOCIATED WITH ARRAY
      ND = 0
      DO 10 I=1, NADIMS
         IF (IARDEF(JRECTP, I) .NE. IRECTP .OR. IARDEF(JFLDRC, I) .NE.
           +   IFLD) GO TO 10
         ND = ND + 1
         NLEVS(ND) = IARDEF(JNLEVL, I)
         ILEVS(ND) = IARDEF(JLEVPT, I)
         CONTINUE
      10   IFLDRC = IFLDPT(IRECTP) + IFLD
21

```

## OUTPUT ROUTINES - OUTARR SUBROUTINE

```

1SN
22      UNIT = UNITS(IFLCRC)
23      C  BRANCH ACCORDING TO NUMBER OF DIMENSIONS OF ARRAY
24      IF (ND.LE.0.OR. ND.GT.3) CALL ERR(38,'OUTARR',ND,0,0)
25      GO TO (20,40,60), ND
26
27      C  1-DIMENSIONAL ARRAY.  ITERATE OVER LEVELS OF ARRAY, OUTPUTTING
28      C  EACH WITH ASSOCIATED VALUES
29      N = NLEVS(1)
30      DO 30 I=1,N
31          K = ILEVS(1) + I - 1
32          IF(ARR(I).EQ. FNULL) GO TO 31
33          WRITE (7,100) (ARRLAB(J,K),J=1,5),ARR(I),UNIT
34          GO TO 30
35          FORMAT (4X,5A4,10X,F10.2,4X,A4)
36          WRITE(7,101) (ARRLAB(J,K),J=1,5)
37          FORMAT(4X,5A4,16X,'N/A')
38          CONTINUE
39          RETURN
40
41      C  2-DIMENSIONAL ARRAY.  ITERATE OVER DIMENSIONS IN REVERSE ORDER.
42      C  ITERATE OVER LEVELS OF ARRAY, OUTPUTTING EACH WITH ASSOCIATED VALUE
43      L = 0
44      N2 = NLEVS(2)
45      DO 50 JD=1,N2
46          K2 = ILEVS(2) + JD - 1
47          WRITE(7,100) (ARRLAB(J,K2),J=1,5)
48          N1 = NLEVS(1)
49          DO 50 ID=1,N1
50              L = L + 1
51              K1 = ILEVS(1) + ID - 1
52              IF(ARR(L).EQ. FNULL) GO TO 51
53              WRITE (7,105) (ARRLAB(J,K1),J=1,5),ARR(L),UNIT
54              GO TO 50
55              FORMAT (7X,5A4,7X,F10.2,4X,A4)
56              WRITE(7,106) (ARRLAB(J,K1),J=1,5)

```

```

106  FORMAT(7X,5A4,13X,'N/A')
50    CONTINUE
51    RETURN
52
C
C 3-DIMENSIONAL ARRAY.  ITERATE OVER DIMENSIONS IN REVERSE ORDER.
C  ITERATE OVER LEVELS OF ARRAY, FOR EACH OUTPUTTING LABEL & VALUE
60    L = 0
      N3 = NLEVS(3)
      DO 70 KD=1,N3
        K3 = ILEVS(3) + KD - 1
        WRITE (7,100) (ARRLAB(J,K3),J=1,5)
        N2 = NLEVS(2)
        DO 70 JD=1,N2
          K2 = ILEVS(2) + JD - 1
          WRITE (7,105) (ARRLAB(J,K2),J=1,5)
          N1 = NLEVS(1)
          DO 70 ID=1,N1
            L = L + 1
            IF(ARR(L).EQ.FNULL) GO TO 71
            K1 = ILEVS(1) + ID - 1
            WRITE (7,110) (ARRLAB(J,K1),J=1,5),ARR(L),UNIT
            GO TO 70
          END DO
        END DO
      END DO
      FORMAT (10X,5A4,4X,F10.2,4X,A4)
      WRITE(7,111) (ARRLAB(J,K1),J=1,5)
      FORMAT(10X,5A4,10X,'N/A')
      CONTINUE
      RETURN
      END

```

## OUTPUT ROUTINES - OUTCOM SUBROUTINE

```

ISN
1  SUBROUTINE OUTCOM
C
C  ROUTINE TO OUTPUT THE ATTRIBUTES IN COMMON TO ALL COMPONENTS OF
C  THE VEHICLE AS WELL AS COMBINED MEASURES OFR THE VEHICLE AS A WHOLE.
C
2  COMMON /VPROIO/
3  REAL*8 FLDNAM, RECNAM, DEFAULT
4  COMMON /JPARS/
5  COMMON /JPARS1/
6  COMMON /NPARS/
7  COMMON /OUTPLT/
8  LOGICAL*1 DF
9  REAL*8 XNAT, XMANUF, XMODEL
C
10  DIMENSION ISUB(3), CV(5)
11  DATA ISUB/'A','B','C'/, CV/' C','ONCE','PT V','EHIC','LE'/
C
C  PRODUCE TABLE 1A: VEHICLE COMPONENT IDENTIFICATIONS & PRODUCIBILITY
C
C  OUTPUT TABLE TITLE
12  WRITE (7,100) ISUB(1)
13  100  FORMAT ('1',23X,'TABLE 1',A1,'.  CONCEPT VEHICLE:',
+        ' ATTRIBUTES IN COMMON TO ALL COMPONENTS.')

```

```

ISN          OUTPUT ROUTINES - OUTCOM SUBROUTINE

20  C  WRITE SPACER OR SUBSYSTEM HEADER WITHOUT CORRESPONDING VALUES
21  WRITE (7,110) (OUTNAM(J,I),J=1,7)
22  FORMAT (1X,7A4)
      GO TO 20
23  C  WRITE LABEL & VALUES
24  KK = KK + 1
25  K = KOUT(KK)
      WRITE (7,115) (OUTNAM(J,I),J=1,7),XNAT(K),XMANUF(K),XMODEL(K),
26  +      INDX(K),NPROD(K),DF(1,K),RDTIME(K),DF(2,K)
27  FORMAT (1X,7A4,3X,A8,2(2X,A8),4X,I2,6X,I6,A1,2X,F6.1,A1)
      CONTINUE
28  C
      C  OUTPUT SEPARATOR LINE & VEHICLE SUMMARY
      C  WRITE (7,120) CV,NPROD(NCMPT1)
      C  FORMAT (82X,'_____/1X,5A4,60X,F6.1)
      C
      C  NOW DO TABLE IB:  NUMBER, WEIGHT, VOLUME, & RAM/D
      C
      C  OUTPUT TABLE TITLE
      C  WRITE (7,100) ISUB(2)
      C
      C  OUTPUT TABLE COLUMN HEADERS
      C  WRITE (7,125)
      C  FORMAT ('- ',47X,'WEIGHT',9X,'INTERNAL VOLUME',6X,'-----',
30  +      'R A M / D -----',1X,' SUBSYSTEM / COMPONENT',10X,
31  +      'NUMBER',7X,'LBS PERCENT',6X,'CU FT PERCENT',5X,
      +      'MAINT/OP MATURITY CMPLXITY MMBF')
      C
      C  ITERATE OVER COMPONENT LABELS, WRITING LABELS & ASSOCIATED VARIABLES
      C  KK = 0
      C  DO 40 I=1,LAST
      C  IF (IOUTF(I) .NE. 0) GO TO 30
32  C  WRITE SPACER OR SUBSYSTEM HEADER WITHOUT CORRESPONDING VALUES
33  WRITE (7,110) (OUTNAM(J,I),J=1,7)
34  GO TO 40
35  C  WRITE LABEL & VALUES
36

```

## OUTPUT ROUTINES - OUTCOM SUBROUTINE

```

15N
37 30      KK = KK + 1
38      K = KOUT(KK)
39      IF( K.EQ. JHULL .OR. K.EQ. JTURET) GO TO 35
40      WRITE (7,130) (OUTNAM(J,I),J=1,7),NUMBER(K),WEIGHT(K),
+      DF(3,K),PCWT(K),VOLUME(K),DF(4,K),PCVOL(K),DWNUP(K),
+      DF(5,K),MATUR(K),DF(6,K),ICMPLX(K),DF(7,K),RELIAB(K),DF(8,K)
41      GO TO 40
42 35      WRITE(7,131) (OUTNAM(J,I),J=1,7),NUMBER(K),WEIGHT(K),
+      DF(3,K),PCWT(K),VOLUME(K),DF(4,K),DWNUP(K),
+      DF(5,K),MATUR(K),DF(6,K),ICMPLX(K),DF(7,K),RELIAB(K),DF(8,K)
43 130      FORMAT (1X,7A4,5X,12,4X,F9.2,A1,3X,F5.1,5X,F6.1,A1,3X,F5.1,5X,
+      F6.1,A1,6X,I1,A1,8X,I1,A1,3X,F7.1,A1)
44 131      FORMAT (1X,7A4,5X,12,4X,F9.2,A1,3X,F5.1,5X,F6.1,A1,3X,
+      F6.1,A1,6X,I1,A1,8X,I1,A1,3X,F7.1,A1)
45 40      CONTINUE
C
C      WRITE MISC SYSTEM LINE
C
46      WGTM = FMCWGT * WEIGHT(NCMPT1)
47      VOLLM = SUM(VOLUME(JMAING),NCMPT1-3)
48      VOLLM = VOLUME(NCMPT1) - VOLLM
49      IF(VOLLM.LE. 0.0) VOLLM = 0.0
50      PCT11 = VOLLM / VOLUME(NCMPT1) * 100.
51      WRITE(7,132) WGTM,FMCWGT,VOLLM,PCT11
52 132      FORMAT(1X,
+      3X,F5.1,5X,F6.1,1X,3X,F5.1,5X,
+      3X,
+      0.0)
C      OUTPUT SEPARATOR LINE & VEHICLE SUMMARY
53      WRITE (7,135) CV,WEIGHT(NCMPT1),PCWT(NCMPT1),VOLUME(NCMPT1),
+      PCVOL(NCMPT1),DWNUP(NCMPT1),MATUR(NCMPT1),ICMPLX(NCMPT1),
+      RELIAB(NCMPT1)
54 135      FORMAT (34X,
+      ,6X,
+      ,9X,
+      ,4X,
+      ,6X,
+      ,8X,
+      ,6X,
+      ,1X,5A4,14X,
+      ,1X,5X,F6.1,4X,F5.1,5X,F6.1,7X,I1,9X,I1,4X,F7.1)
55      RETURN
56      END

```

```

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



## OUTPUT ROUTINES - OUTENG SUBROUTINE

```

I SN      C  WRITE SPACER OR HEADER WITHOUT CORRESPONDING VALUE
34         5  WRITE (7,105) (OUTNAM(J,1),J=1,7)
35         105  FORMAT (1X,7A4)
36         IFRST = I
37         GO TO 80

C
C  GET VALUE ASSOCIATED WITH LABEL
38         8  IFLDRC = IOUTF(I)
39         ITYPE = IFLOTP(IFLDRC)
40         IFLO = IFLOVL(IFLDRC)
41         IRECTP = IFLDRC - IFLO
42         IRECTP = ITABLE(IFLOTP,IRECTP,NRECTP,1)
43         NF = NFLODS(IRECTP)
44         IOFF = IFILPT(IRECTP)
45         ICMP = IRECTP - I
46         IREC = IVECLE(ICMP,IV)
47         IF (IREC.EQ. NULL) GO TO 80
48         DO 10 K=1,NTYPES
49             IF (ITYPES(K).EQ. ITYPE) GO TO 15
50             CONTINUE
51         CALL ERR(10,'OUTENG', 'VEHIC',FLDNAM(IFLDRC), ITYPE)
52         15  UNIT = UNITS(IFLDRC)
53         DF = BL
54         C  BRANCH ACCORDING TO CATATYPE
55             GO TO (20,30,40,50),K

C
C  REAL VALUE
55         20  V = DVAL(VFILE(IOFF),NF,IFLO,IREC,IRECTP,UNIT,DF)
56         WRITE (7,110) (OUTNAM(J,1),J=1,7),V,DF,UNIT
57         110  FORMAT (1X,7A4,1X,F10.2,A1,3X,A4)
58         GO TO 80

C
C  INTEGER VALUE
59         30  IVL = IDVAL(IVFILE(ICFF),NF,IFLO,IREC,IRECTP,UNIT,DF)
60         WRITE (7,115) (OUTNAM(J,1),J=1,7),IVL,DF,UNIT
61         115  FORMAT (1X,7A4,3X,I8,A1,3X,A4)

```

## OUTPUT ROUTINES - OUTENG SUBROUTINE

```

1 SN
62      GO TO 80
C
C   REAL ARRAY
40      IOFF = IFILPT(IRECTP)
63      IVL = IVAL(VFILE(IOFF),NF,IFLD,IREC)
64      IF (IVL.EQ.NULL) GO TO 20
65      C GET POINTER TO ARRAY DESCRIPTION
66      DO 42 IARR=1,NARRS
67      IF (IARDF2(JRECTP,IARR).EQ.IRECTP.AND.IARDF2(JFLDRC,
+      IARR).EQ.IFLD) GO TO 44
68      CONTINUE
C   NUMBER OF ELEMENTS IN ARRAY
69      NV = IARDF2(JARSIZ,IARR)
C   OFFSET IN FILE CONTAINING ARRAYS
70      IOFFAR = IARDF2(JARRPT,IARR)
C   GET OFFSET OF ARRAY TO BASE
71      IOFF = IOFFAR + NV*(IVL-1)
C   WRITE CONTENTS OF ARRAY
72      WRITE(7,105) (OUTNAM(J,I),J=1,7)
73      CALL OUTARR(ARRAY(IOFF),IRECTP,IFLD)
74      GO TO 80
C
C   CHARACTER VALUE (SHOULDN'T BE ANY)
50      CONTINUE
80      I = I + 1
75      IF (I.LE.LAST) GO TO 2
76      RETURN
77      END
78
79

```

# OUTPUT ROUTINES - OUTER SUBROUTINE

ISN  
1

SUBROUTINE OUTER(IV)

C

ROUTINE TO OUTPUT PERFORMANCE PARAMETERS FOR A SPECIFIED VEHICLE

C

INPUT PARAMETERS:

C

IV DESCRIPTION OF VEHICLE OF INTEREST

C

COMMON /VEH/

2 3

DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),

+ FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),

+ ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),

+ IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),

+ IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),

+ IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),

+ IELECT(26,6), ICCMOC(28,9), IFIREX(28,10), IENVIR(26,9),

+ IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),

+ IRELN(3,50), VFILE(1), IVFILE(1)

4 EQUIVALENCE (VEHICLE(1),IVECLE(1)), (HULL(1),IHULL(1)), (TURRET(1),

+ ITURET(1)), (MAINGN(1),FMAING(1)), (MACGUN(1),FMACGN(1)),

+ (MISGUN(1),FMISGN(1)), (ADGUN(1),IADGUN(1)), (RANGER(1),

+ IRANGR(1)), (SENSOR(1),ISENSR(1)), (STABLE(1),ISTABL(1)),

+ (GUNCTL(1),IGNCTL(1)), (AMMO(1),IAMMO(1)), (AMMOC(1),

+ IAMMOC(1)), (ENGINE(1),IENGIN(1)), (TRANSM(1),ITRANS(1)),

+ (FINDR(1),IFINDR(1)), (ROADWH(1),IROADW(1)), (SPRING(1),

+ ISPRNG(1)), (TPACK(1),ITRACK(1)), (SKIRT(1),ISKIRT(1)),

+ (FUEL(1),IFUEL(1)), (FUELC(1),IFUELC(1)), (CREW(1),

+ ICREW(1)), (CARGO(1),ICARGO(1)), (ELECTR(1),IELECT(1)),

+ (CCMOC(1),ICCMOC(1)), (FIREX(1),IFIREX(1)), (ENVIRC(1),

+ IENVIR(1)), (DIAGNS(1),IDIAGN(1)), (SIGSUP(1),ISIGSP(1)),

+ (SMOKEG(1),ISMOKE(1)), (IEWSYS(1),IEWSYS(1)), (RELN(1),IRELN(1))

EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))

COMMON /VPROTD/

REAL\*8 FLDNAM, RECNAM, DFAULT

COMMON /NPARS/

COMMON /AUX/

5 6 7 8 9

```

      OUTPUT ROUTINES - OUTPER SUBROUTINE

      REAL*8 BL8,CASHES
      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
      COMMON /ARRAYS/
      REAL*8 MANUF,MODEL,NATION
      DIMENSION ARRAY(1)
      EQUIVALENCE (ARRAY(1),DETRNG(1))
      COMMON /JPARS2/

      LOGICAL*1 BL,DF
      DATA BL/' '/

      IRECTP = 1

      C
      C OUTPUT TABLE TITLE
      WRITE (7,100)
      100  FORMAT ('1',' TABLE 2.  CONCEPT VEHICLE, PERFORMANCE',
      +      ' PARAMETERS.'/')

      C
      C ITERATE OVER PERFORMANCE PAR LABELS
      LAST = ICNATR - 1
      NF = NFIDS(1)
      DO 80 I=1,PPARS,LAST
      IF (IOUTF(I)) .NE. 0) GO TO 8

      C WRITE SPACER OR HEADER WITHOUT CORRESPONDING VALUE
      5    WRITE (7,105) (OUTNAM(J,I),J=1,7)
      105  FORMAT (1X,7A4)
      GO TO 80

      C
      C GET VALUE ASSOCIATED WITH LABEL
      8    IFLD = IOUTF(I)
      ITYPE = IFLDTP(IFLD)
      DO 10 K=1,NTYPES
      IF (ITYPES(K)) .EQ. ITYPE) GO TO 15
      10    CONTINUE
      CALL EPR(10,'OUTPER','VEHIC',FLDNAM(IFLD),ITYPE)
      15    UNIT = UNITS(IFLD)

```

## OUTPUT ROUTINES - OUTER SUBROUTINE

```

15N
36      DF = RI
37      C  BRANCH ACCORDING TO DATATYPE
          GO TO (20,30,40,50),K
38      C
39      C  REAL VALUE
40      V = DVAL(IVECLE,NF,IFLD,IV,1,UNIT,DF)
41      WRITE (7,110) (OUTNAM(J,I),J=1,7),V,DF,UNIT
          FORMAT (1X,7A4,1X,F10.2,A1,3X,A4)
          GO TO 80
42      C
43      C  INTEGER VALUE
44      IVL = IDVAL(IVECLE,NF,IFLD,IV,1,UNIT,DF)
45      WRITE (7,115) (OUTNAM(J,I),J=1,7),IVL,DF,UNIT
          FORMAT (1X,7A4,3X,I8,A1,3X,A4)
          GO TO 80
46      C
47      C  REAL ARRAY
48      IOFF = IFILPT(IRECTP)
          IVL = IVAL(VFILE(IOFF),NF,IFLD,IV)
          IF (IVL.EQ. NULL) GO TO 20
49      C  GET POINTER TO ARRAY DESCRIPTION
          DO 42 IARR=1,NARRS
          IF (IARDF2(JRECTP,IARR).EQ. IRECTP .AND. IARDF2(JFLDRC,
50      + IARR).EQ. IFLD) GO TO 44
          CONTINUE
51      C  NUMBER OF ELEMENTS IN ARRAY
          NV = IARDF2(JARSIZ,IARR)
52      C  OFFSET IN FILE CONTAINING ARRAYS
          IOFFAR = IARDF2(JARRPT,IARR)
53      C  GET OFFSET OF ARRAY TO BASE
          IOFF = IOFFAR + NV*(IVL-1)
54      C  WRITE CONTENTS OF ARRAY
          WRITE(7,105) (OUTNAM(J,I),J=1,7)
          CALL CUTARR(ARRAY(IOFF),IRECTP,IFLD)
          GO TO 80
55      C
56
57

```

## OUTPUT ROUTINES - OUTER SUBROUTINE

|     |    |                                    |
|-----|----|------------------------------------|
| ISN | C  | CHARACTER VALUE (SHOULDN'T BE ANY) |
| 58  | 50 | CONTINUE                           |
| 59  | 80 | CONTINUE                           |
| 60  |    | RETURN                             |
| 61  |    | END                                |

## OUTPUT ROUTINES - OUTVEH SUBROUTINE

```

1 SN
1 SUBROUTINE OUTVEH
C
C ROUTINE TO OUTPUT BEST CANDIDATE VEHICLE
C
2 CCMCN /VEH/
3 DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
+ FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
+ ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),
+ IAMMOC(26,9), IENGINE(40,8), ITRANS(32,8), IFINDR(28,10),
+ IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
+ IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
+ IELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),
+ IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
+ IRELN(3,50), VFIL(1), IVFILE(1)
4 EQUIVALENCE (VEHICLE(1),IVECLE(1)), (HULL(1),IHULL(1)), (TURET(1),
+ ITURET(1)), (MAINGN(1),FMAING(1)), (MACGUN(1),FMACGN(1)),
+ (MISGUN(1),FMISGN(1)), (ADGUN(1),IADGUN(1)), (RANGER(1),
+ IRANGR(1)), (SENSOR(1),ISENSR(1)), (STABLE(1),ISTABL(1)),
+ (GNCTL(1),IGNCTL(1)), (AMMO(1),IAMMO(1)), (AMMOC(1),
+ IAMMOC(1)), (ENGINE(1),IENGINE(1)), (TRANSM(1),ITRANS(1)),
+ (FINLDR(1),IFINDR(1)), (ROADWH(1),IROADW(1)), (SPRING(1),
+ ISPRNG(1)), (TRACK(1),ITRACK(1)), (SKIRT(1),ISKIRT(1)),
+ (FUEL(1),IFUEL(1)), (FUELC(1),IFUELC(1)), (CREW(1),
+ ICREW(1)), (CAPGO(1),ICARGO(1)), (ELECTR(1),IELECT(1)),
+ (COMMO(1),ICOMMO(1)), (FIREX(1),IFIREX(1)), (ENVIRC(1),
+ IENVIR(1)), (DIAGNS(1),IDIAGN(1)), (SIGSUP(1),ISIGSP(1)),
+ (SMOKEG(1),ISMOKE(1)), (EWSYS(1),IEWSYS(1)), (RELN(1),IRELN(1))
5 EQUIVALENCE (VEHICLE(1),VFIL(1),IVFILE(1))
6 CALL TABVAL(ICANDV)
7 CALL OUTCOM
8 CALL OUTPER(ICANDV)
C

```

OUTPUT ROUTINES - OUTVEH SUBROUTINE

ISN

9

C

CALL OUTENG(ICANDV)

RETURN

END

10

11

# OUTPUT ROUTINES - TABVAL SUBROUTINE

## SUBROUTINE TABVAL( IV)

ROUTINE TO PUT VALUES OF INTEREST INTO ARRAYS PRIOR TO PRINTING.  
ALSO CALCULATES SUMS, AVERAGES, & PERCENTS OF APPROPRIATE ITEMS.

### INPUT PARAMETERS:

TV  
VEHICLE CF INTEREST

COMMON /VFH/

```

DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
+ FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
+ ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),
+ IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),
+ IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
+ IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
+ IELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),
+ IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
+ IRELN(3,50), VFILE(1), IVFILE(1)

```

```

EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
+ ITURRET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
+ (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
+ IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
+ (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMDC(1),
+ IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
+ (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
+ ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
+ (FUEL(1),IFUEL(1)),(FUELCL(1),IFUELCL(1)),(CREW(1),
+ ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
+ (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
+ IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
+ (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
COMMON /VPRN10/
REAL*8 FLDNAM,RECNAM,DEFAULT
COMMON /JPARS/

```

OUTPUT ROUTINES - TABVAL SUBROUTINE

```

1 ISN
9
10 COMMON /JPARS1/
11 COMMON /NPARS/
12 COMMON /OUTPUT/
13 LOGICAL*1 OF
14 REAL*8 XNAT,XMANUF,XMODEL
15 COMMON /AUX/
16 REAL*8 BL8,DASHES
17 EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
18 COMMON /ARRAYS/
19 REAL*8 MANUF,MODEL,NATION
20 DIMENSION ARRAY(1)
21 EQUIVALENCE (ARRAY(1),DETRNG(1))
22 COMMON /JPARS2/
23 COMMON /DATA/
24 COMMON /DATA2/
25
26 C INITIALIZE ARRAYS
27 CALL IZERO(NUMBER,NCMPT1*12)
28 CALL INITVL(CF,2*NCMPTP,' ')
29 CALL INITVL(XMANUF,64,BLS)
30 CALL INITVL(XMODEL,64,BLS)
31 CALL INITVL(XNAT,64,BLS)
32
33 C ITERATE OVER COMPONENT TYPES
34 DO 100 ICMP=1,NCMPTP
35
36 C GET INFORMATION ABOUT THIS COMPONENT TYPE
37 IRECTP = ICMP + 1
38 IDFF = IFILPT(IRECTP)
39 NF = NFLDS(IRECTP)
40
41 C ITERATE OVER COMPONENTS OF THIS VEHICLE
42 IREC = IVECLE(ICMP,IV)
43 IF (IREC.EQ. NULL) GO TO 20
44
45 C INCREMENT NUMBER OF COMPONENTS OF THIS TYPE

```

## OUTPUT ROUTINES - TABVAL SUBROUTINE

```

37      ISN
38      35      C      NUMBER(ICMP) = NUMBER(ICMP) + 1
39      36      C      INCREMENT WEIGHT OF COMPONENTS OF THIS TYPE
40      37      C      WEIGHT(ICMP) = WEIGHT(ICMP) + DVAL(VFILE(IOFF),NF,JWT,IREC,
41      38      C      + IRECTP,'LB ',DF(3,ICMP))
42      39      C      INCREMENT INTERNAL VOLUME OF COMPONENTS OF THIS TYPE
43      40      C      TEST LOCATION OF COMPONENT
44      41      C      ICLOC = IVAL(IVFILE(IOFF),NF,JLOC,IREC)
45      42      C      IF( ICLOC .EQ. 3) GO TO 12
46      43      C      VOLUME(ICMP) = VOLUME(ICMP) + DVAL(VFILE(IOFF),NF,JVOL,IREC,
47      44      C      + IRECTP,'FT3 ',DF(4,ICMP))
48      45      C      INCREMENT MAINTENANCE / OPERATIONAL HRS
49      46      C      12 DWNUP(ICMP) = DWNUP(ICMP) + DVAL(VFILE(IOFF),NF,JDOWNUP,IREC,
50      47      C      + IRECTP,'- ',DF(5,ICMP))
51      48      C      INCREMENT MATURITY INDEX
52      49      C      MATUR(ICMP) = MATUR(ICMP) + IDVAL(VFILE(IOFF),NF,JMATUR,IREC,
53      50      C      + IRECTP,'- ',DF(6,ICMP))
54      51      C      INCREMENT COMPLEXITY INDEX
55      52      C      ICMPLEX(ICMP) = ICMPLEX(ICMP) + IDVAL(VFILE(IOFF),NF,JCMPLEX,IREC,
56      53      C      + IRECTP,'- ',DF(7,ICMP))
57      54      C      INCREMENT RELIABILITY INDEX
58      55      C      RELIAB(ICMP) = RELIAB(ICMP) + DVAL(VFILE(IOFF),NF,JRELIB,IREC,
59      60      C      + IRECTP,'- ',DF(8,ICMP))
61      62      C      DO FOLLOWING OPERATIONS ONLY FOR FIRST COMPONENT OF THIS TYPE
62      63      C      IF (NUMBER(ICMP) .NE. 1) GO TO 15
63      64      C      NUMBER OF ITEMS PRODUCED
64      65      C      NPROD(ICMP) = IDVAL(VFILE(IOFF),NF,JNUM,IREC,

```

```

      OUTPUT ROUTINES - TABVAL SUBROUTINE

      + IRECTP, '- ', DF(1, ICOMP))
C
C  R&D TIME
46  RDTIME(ICMP) = DVAL(VFILE(IOFF), NF, JRTIME, IREC,
      + IRECTP, 'YR ', DF(2, ICOMP))
C
C  NOTE COMPONENT INDEX
47  INDX(ICMP) = IVAL(VFILE(IOFF), NF, JID, IREC)
C
C  NATION
48  K = IVAL(VFILE(IOFF), NF, JNAT, IREC)
49  XNAT(ICMP) = NATCN(K)
C
C  MANUFACTURER
50  K = IVAL(VFILE(IOFF), NF, JMANUF, IREC)
51  XMANUF(ICMP) = MANUF(K)
C
C  MODEL
52  K = IVAL(VFILE(IOFF), NF, JMODEL, IREC)
53  XMODEL(ICMP) = MODEL(K)
C
C  GET NEXT COMPONENT OF THIS TYPE
54  IREC = IVAL(VFILE(IOFF), NF, JNEXT, IREC)
55  GO TO 10
C
C  IF THERE IS MORE THAN ONE COMPONENT OF THIS TYPE, COMPUTE
C  AVERAGES FOR CERTAIN OF THE ATTRIBUTES
C
56  IF (NUMBER(ICMP) .LE. 1) GO TO 100
C
57  X = NUMBER(ICMP)
C
C  AVERAGE COMPLEXITY INDEX
58  ICMPLEX(ICMP) = (ICMPLEX(ICMP) / X) + 0.5
C
C  AVERAGE MATURITY INDEX

```

# OUTPUT ROUTINES - TABVAL SUBROUTINE

C.6: UTILITY ROUTINES

This section contains listings of utility routines used for a variety of purposes. Comment cards in the listing of each routine indicate its purpose.

```

UTILITY ROUTINES - ARMULT SUBROUTINE

SUBROUTINE ARMULT(Y,X,N,W)

    ROUTINE TO MULTIPLY EACH ELEMENT OF A VECTOR BY A CONSTANT ,
    STORING THE RESULT OF EACH MULTIPLICATION IN THE CELLS OF A
    VECTOR PARALLEL TO THE FIRST.

    INPUT PARAMETERS:
        Y      RESULTANT VECTOR
        X      INITIAL VECTOR
        N      NUMBER OF CELLS IN X & Y
        W      THE CONSTANT THAT EACH ELEMENT OF X IS TO MULTIPLIED BY

    DIMENSION X(N),Y(N)

    DO 10 I=1,N
        Y(I) = X(I) * W
    RETURN
END
```



## UTILITY ROUTINES - DVAL FUNCTION

```

1  FUNCTION DVAL(FILE,NFLD,IFLD,IREC,I RECTP,UNIT,DF)
2
3  C
4  C FUNCTION TO RETRIEVE A VALUE OF A SPECIFIED FIELD OF A RECORD.
5  C IF THE VALUE IS NULL A DEFAULT VALUE IS SUBSTITUTED FOR IT. AN
6  C INPUT TO THE FUNCTION ALSO INDICATES WHAT UNITS OF MEASURE ARE
7  C DESIRED. IF THE VALUE IS NOT ALREADY IN THESE UNITS IT IS CONVERTED
8  C TO THEM.
9  C
10 C
11 C INPUT PARAMETERS:
12 C FILE FILE IN WHICH VALUE IS LOCATED
13 C NFLD NUMBER OF FIELDS IN EACH RECORD OF FILE
14 C IFLD FIELD OF INTEREST
15 C IREC RECORD OF INTEREST
16 C I RECTP RECORD TYPE
17 C UNIT UNIT OF MEASURE DESIRED
18 C
19 C OUTPUT PARAMETERS:
20 C DF SET TO "*" IF DEFAULT VALUE IS USED
21 C
22 C
23 C COMMON /VPROTO/
24 C REAL*8 FLDNAM,RECNAME,DEFAULT
25 C COMMON /NPARS/
26 C COMMON /AUX/
27 C REAL*8 BL8,DASHES
28 C EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
29 C
30 C
31 C LOGICAL*1 STAR,DF
32 C DIMENSION FILE(NFLD,1)
33 C DATA DASH/'-','/',STAR/'*','/'
34 C
35 C
36 C OVAL = FILE(IFLD,I REC)
37 C IFLDRC = IFLDPT(I RECTP) + IFLD
38 C IF (OVAL .NE. FNULL) GO TO 5
39 C OVAL = DEFAULT(IFLDRC)
40 C DF = STAR

```

## UTILITY ROUTINES - DVAL FUNCTION

```

15N
16      5      IF (UNIT.EQ. DASH .OR. UNIT.EQ. UNITS(IFLDRC) .OR. UNITS(IFLDRC)
      +      .EQ. CASH) RETURN
17      C      SEARCH TRANSLATION TABLE FOR UNIT TRANSLATION FACTOR
18      DO 10 I=1,NUTRNS
19      IF (UNIT.NE. TUNITS(1,I) .OR. UNITS(IFLDRC) .NE. TUNITS(2,I))
      +      GO TO 8
20      DVAL = DVAL * TUNITS(3,I)
21      RETURN
22      8      IF (UNIT.NE. TUNITS(2,I) .OR. UNITS(IFLDRC) .NE. TUNITS(1,I))
      +      GO TO 10
23      DVAL = DVAL / TUNITS(3,I)
24      RETURN
      10      CONTINUE
      C
      C      WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
25      CALL ERR(37,'DVAL ',IRECTP,UNIT,UNITS(IFLDRC))
26      RETURN
27      END

```

## 1

٤

6

c

6

۷

٤

C

C



## UTILITY ROUTINES - DVAL2 FUNCTION

```

ISN
18      +      GO TO 10
19      DVAL2 = DVAL2 / TUNITS(3,I)
20      RETURN
21      CONTINUE
22      C
23      C WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
      CALL ERR(37,'DVAL2 ',ICMP+1,UNIT,UNITS(IFLDRC))
      RETURN
      END

```

## UTILITY ROUTINES - ERR SUBROUTINE

```

1  ISN
1  SUBROUTINE ERR(IERR,SUBR,PAR1,PAR2,PAR3)
C
C  ROUTINE TO PROCESS ERRORS AND WRITE OUT ERROR COMMENTS.
C  ALSO HAS AN ENTRY "WARN" TO PRINT OUT WARNING MESSAGES.
C
C  INPUT PARAMETERS:
C  IERR  POSITIVE INTEGER IDENTIFYING THE ERROR CONDITION
C  SUBR  SUBROUTINE IN WHICH THE ERROR OCCURRED
C  PAR1  THE FIRST PARAMETER OF THE ERROR CONDITION
C  PAR2  THE SECOND PARAMETER OF THE ERROR CONDITION
C  PAR3  THE THIRD PARAMETER OF THE ERROR CONDITION
C
C  COMMON /VFH/
2  DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
3  + FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
+ ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
+ IAMMOC(26,9),IENGINE(40,8),ITRANS(32,8),IFINDR(28,10),
+ IROADWH(26,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
+ IFUEL(28,10),IFUELCL(26,9),ICREW(26,5),ICARGO(26,5),
+ IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
+ IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
+ IRELN(3,50),VFILE(1),IVFILE(1)
4  EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
+ ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
+ (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
+ IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
+ (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
+ IAMMOC(1)),(ENGINE(1),IENGINE(1)),(TRANSM(1),ITRANS(1)),
+ (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADWH(1)),(SPRING(1),
+ ISPFNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
+ (FUEL(1),IFUEL(1)),(FUELCL(1),IFUELCL(1)),(CREW(1),
+ ICREW(1)),(CARGO(1),ICARGO(1)),(ELECT(1),IELECT(1)),
+ (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
+ IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
+ (SMOKEG(1),ISMOKE(1)),(IEWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))

```

## UTILITY ROUTINES - ERR SUBROUTINE

```

TSN
5  EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
6  COMMON /VPRG TO/
7  REAL*8 FLDNAM,RECNAM,DEFAULT
8  COMMON /JPARS/
9  COMMON /JPARS1/
10 COMMON /NPARS/
11 COMMON /OUTPUT/
12 LOGICAL*1 DF
13 REAL*8 XNAT,XMANUF,XMODEL
14 COMMON /AUX/
15 REAL*8 BL8,DASHES
16 EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
17 COMMON /ARRAYS/
18 REAL*8 MANUF,MODEL,NATION
19 DIMENSION ARRAY(1)
20 EQUIVALENCE (ARRAY(1),DETRNG(1))
21 COMMON /JPARS2/
22 COMMON /SPECS/
23 REAL*8 ATPENG,ATRPER,VEHCAT,VEHCMP,VEHENG,VEHPER
24 COMMON /SPECS2/
25 REAL*8 VEHS
26 COMMON /DATA/
27 COMMON /DATA2/
C
28 REAL*8 SUBR
29 DIMENSION PAR1(2),PAR2(2)
30 EQUIVALENCE (XPARI,IPARI),(XPAR2,IPAR2),(XPAR3,IPAR3)
C  EQUATE ARGS SO THAT INTEGER PARAMETERS ARE DEFINED.
31 XPARI = PAR1(1)
32 XPAR2 = PAR2(1)
33 XPAR3 = PAR3
C
34 WRITE (6,2) IERR,SUBR
35 FORMAT ('0 *** ERROR ',I2,' DETECTED IN SUBROUTINE ',A6,'.')
C
C  BRANCH TO APPROPRIATE PIECE OF CODE TO HANDLE ERROR.

```

## UTILITY ROUTINES - ERR SUBROUTINE

```

ISN      C
36      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,
+        170,180,190,200,210,220,230,240,250,260,270,280,290,300,
+        310,320,330,340,350,360,370,380,390,400,410,420,430,440,450),
+        IERR

37      C      WRITE (6,15) XPAR1,XPAR2
38      10      FORMAT (' THE FIRST NON-COMMENT RECORD OF THE PROTOTYPE FILE' /
15      15      ' WAS NOT A VALID VEHICLE DEFINITION RECORD. IT CONTAINED',
+        ' A4/' IN COLUMNS 2-5, AND 'A4,' IN COLUMNS 12-15.')
39      STOP 1

40      C      WRITE (6,25) PAR1,IPAR3,IPAR2
41      20      FORMAT (' THE ATTRIBUTE ',2A4,' OF RECORD TYPE ',I2,' SPECIFIED' /
25      25      ' IN THE VEHICLE PROTOTYPE FILE HAS AN OFFSET WITHIN ITS' /
+        ' RECORD OF ',I4,' WHICH EXCEEDS THE NUMBER OF FIELDS' /
+        ' DEFINED FOR THIS RECORD TYPE.')
42      PAUSE
43      GO TO 9999

44      C      WRITE (6,35) IPAR2,IPAR3,PAR1
45      30      FORMAT (' THERE WERE ',I3,' RECORD TYPES ENCOUNTERED IN THE' /
35      35      ' VEHICLE PROTOTYPE FILE AND ONLY ',I3,' DEFINED.' /
+        ' THE LATEST RECODE READ CONTAINS ',2A4,' IN COLS 2-9.')
46      PAUSE
47      GO TO 9999

48      C      WRITE (6,45) PAR1,PAR2
49      40      FORMAT (' THE ALTERNATIVE COMPONENT FILE SPECIFIED THE VEHICLE' /
45      45      ' CATEGORY ',2A4,' WHICH DOES NOT MATCH THE VEHICLE CATEGORY' /
+        ' SPECIFIED BY THE PROTOTYPE FILE: ',2A4)
50      PAUSE
51      GO TO 9999

52      C      WRITE (6,55) PAR1,IPAR3
53      50      FORMAT (' A HEADER WAS ENCOUNTERED IN THE ALTERNATIVE COMPONENT' /
55      55

```

## UTILITY ROUTINES - ERR SUBROUTINE

```

ISN
54      + ' FILE IN WHICH THE COMPONENTS WERE NOT SEQUENTIALLY'//
55      + ' NUMBERED. THE OFFENDING COMPONENT TYPE IS ',2A4,'. THE'//
56      + ' OFFENDING COMPONENT SEQUENCE NUMBER IS ',I3)
57      PAUSE
58      GO TO 9999
59      C
60      WRITE (6,65) PAR2,PAR1
61      FORMAT (' THE ATTRIBUTE ',2A4,' APPEARING UNDER COMPONENT ',2A4//
62      + ' IN THE ALTERNATIVE VEHICLE FILE COULD NOT BE FOUND IN THE'//
63      + ' LIST OF LEGAL ATTRIBUTES FOR THIS COMPONENT.'//)
64      PAUSE
65      GO TO 9999
66      C
67      WRITE (6,75) IPAR1
68      FORMAT (' THE RECORD TYPE SPECIFIED, ',I3,' WAS OUT OF BOUNDS.'//)
69      PAUSE
70      GO TO 9999
71      C
72      WRITE (6,85) IPAR1,IPAR2
73      FORMAT (' HAVE RUN OUT OF RECORDS OF TYPE ',I3,'.'//
74      + ' MORE THAT ',I4,' RECORDS OF THIS TYPE WERE NEEDED.'//
75      + ' NEED TO REDIMENSION APPROPRIATE ARRAYS, CHANGE BLOCKDATA.'//
76      + ' AND RECOMPILE.'//)
77      PAUSE
78      GO TO 9999
79      C
80      WRITE (6,95) PAR1
81      FORMAT (' THE HEADER ',2A4,' READ FROM THE ALTERNATIVE COMPONENT'//
82      + ' FILE WAS NOT RECOGNIZED AS A LEGAL COMPONENT.'//)
83      PAUSE
84      GO TO 9999
85      C
86      WRITE (6,105) XPAR3,PAR2,PAR1
87      FORMAT (' THE DATATYPE ',A4,' STORED FOR ATTRIBUTE ',2A4,' OF'//
88      + ' COMPONENT ',2A4,' DOES NOT MATCH ANY OF THE LEGAL'//
89      + ' ALTERNATIVES.'//)

```

## UTILITY ROUTINES - ERR SUBROUTINE

```

1 SN
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93

```

```

      PAUSE
      GO TO 9999

C
110 WRITE (6,115) XPAR1,XPAR2
115 FORMAT (' THE IDENTIFYING HEADER ',A4,' READ FROM THE USER ',/
+ ' SPECIFICATION FILE DID NOT MATCH THAT EXPECTED: ',A4)
      PAUSE
      GO TO 9999

C
120 WRITE (6,125) PAR1,PAR2
125 FORMAT (' THE USER SPECIFICATION FILE DESIGNATED A VEHICLE',/
+ ' CATEGORY ',2A4,' WHICH DOES NOT MATCH THE VEHICLE CATEGORY',/
+ ' SPECIFIED BY THE PROTOTYPE FILE: ',2A4)
      PAUSE
      GO TO 9999

C
130 WRITE (6,135) PAR1,IPAR2
135 FORMAT (' AN INSUFFICIENT NUMBER OF ',2A4,' RECORDS, ',I3,' ',/
+ ' WERE SET ASIDE TO STORE THE USERS SPECS. ARRAYS',/
+ ' DIMENSIONED ON THIS SUBSCRIPT SHOULD BE ENLARGED, BLOCK',/
+ ' DATA CHANGED, AND THE PROGRAM RECOMPILED.')
```

```

      PAUSE
      GO TO 9999

C
140 WRITE (6,145) XPAR1
145 FORMAT (' THE FIRST CHARACTER OF A HEADER RECORD FOR',/
+ ' (1) COMPONENTS, (2) ENGINEERING PARAMETERS, OR (3) (VEHICLE)',/
+ ' PERFORMANCE PARAMETERS WAS "',A1,"", NOT "C", "E",',/
+ ' "P", OR "V",')
```

```

      PAUSE
      GO TO 9999

C
150 WRITE (6,155) XPAR1
155 FORMAT (' THE TYPE FIELD OF A RECORD DESCRIBING A',/
+ ' VEHICLE PERFORMANCE PARAMETER CONSTRAINT CONTAINED',/
+ ' ",A3,"", NOT " " OR "VEH",')
```

## UTILITY ROUTINES - ERR SUBROUTINE

```

15N
94      PAUSE
95      GO TO 9999
C
96      WRITE (6,165) IPAR2,IPAR1
97      FORMAT (' THE PROTOTYPE FILE SPECIFIED THAT FIELD ',I3,
+           ' OF RECORD TYPE ',I3,' HAD A DATATYPE "RA".')
+           ' THIS DOES NOT MATCH WITH THE ARRAY IARDF2 STORING'
+           ' INFORMATION ABOUT FIELDS WHICH MAY HAVE VALUES'
+           ' WHICH ARE ARRAYS.')
98      PAUSE
99      GO TO 9999
C
100     WRITE (6,175) IPAR3,PAR1,IPAR2
101     FORMAT (' HAVE RUN OUT OF SPACE IN STORING ARRAY TYPE ',I2,
+           ' AS A VALUE OF ATTRIBUTE ',2A4,'. NEED TO ENLARGE'
+           ' NUMBER OF SUCH ARRAYS BEYOND THE ',I4,' CURRENTLY'
+           ' ALLOCATED. CHANGE AFFECTED ARRAY SIZES & BLOCKDATA'
+           ' AND RECOMPILE.')
102     PAUSE
103     GO TO 9999
C
104     WRITE (6,185) PAR2,PAR1
105     FORMAT (' THE USER SPECIFICATION FILE DESIGNATED A VEHICLE '
+           ' CATEGORY ',2A4,' WHICH DOES NOT MATCH THE VEHICLE CATEGORY'
+           ' SPECIFIED BY THE EXISTING VEHICLE FILE: ',2A4)
106     PAUSE
107     GO TO 9999
C
108     WRITE (6,195) IPAR1
109     FORMAT (' USER COMPONENT SPECIFICATION # ',I3,' CALLED FOR THE'
+           ' USE OF AN EXISTING VEHICLE BY HAVING "VEH" IN COLUMNS 2-4'
+           ' BUT FAILED TO INDICATE WHICH EXISTING VEHICLE.')
110     PAUSE
111     GO TO 9999
C
112     WRITE (6,205) PAR1,IPAR2

```

## UTILITY ROUTINES - ERR SUBROUTINE

```

113 205      FORMAT (' VEHICLE NAME ',2A4,' DID NOT FIT IN THE VEHS ARRAY.'/
114      +      ' MUST EXPAND THE SIZE FOR THIS ARRAY BEYOND ',I3)
115      PAUSE
116      GO TO 9999
117
118      C
119      WRITE (6,215) IPAR1,IPAR2
120      FORMAT (' ARRAY CVERFLOW FOR ARRAY IRECS. MUST EXPAND ITS SIZE.'/
121      +      ' BEYOND ',I3,'. THE VEHICLE NAME BEING PROCESSED WHEN'/
122      +      ' OVERFLOW OCCURRED WAS #',I2)
123      PAUSE
124      GO TO 9999
125
126      C
127      WRITE (6,225) XPAR1,PAR2
128      FORMAT (' THE COMPONENT CODE "',A2,'" SPECIFIED BY THE '/
129      +      ' ENGINEERING PARAMETER SPECIFICATION REFERRING TO ATTRIBUTE',
130      +      ' 2A4/' COULD NOT BE LOCATED IN THE LIST OF VALID COMPONENT ',
131      +      ' CODES. THIS SPECIFICATION WILL BE SKIPPED.')
132      RETURN
133
134      C
135      WRITE (6,235) IPAR3,IPAR2,IPAR1
136      FORMAT (' THE NUMBER OF PARAMETERS CALLED FOR BY JSER '/
137      +      ' SPECIFICATION # ',I3,' FOR A COMPONENT OF TYPE ',I2/
138      +      ' WAS ',I3,' WHICH IS OUTSIDE THE RANGE DEFINED BY THE '/
139      +      ' MINIMUM AND MAXIMUM FOR THIS COMPONENT TYPE SPECIFIED '/
140      +      ' IN THE PROTOTYPE FILE. HOWEVER THE USER SPEC WILL BE'/
141      +      ' USED ANYWAY.')
142      RETURN
143
144      C
145      WRITE (6,245) IPAR3,IPAR2,IPAR1
146      FORMAT (' USER SPECIFICATION # ',I3,' FOR COMPONENT OF TYPE ',A3/
147      +      ' REFERRED TO COMPONENT INDEX ',I3,' WHICH IS OUT OF BOUNDS'/
148      +      ' OF THE LEGAL RANGE OF SUCH INDEGES.')
149      PAUSE
150      GO TO 9999
151
152      C
153      WRITE (6,255) PAR2,IPAR3,XPAR1

```

## UTILITY ROUTINES - ERR SUBROUTINE

```

131      ISN      255      FORMAT (' THE ATTRIBUTE ',2A4,' SPECIFIED IN THE USER ENGINEERING'//
+      ' PARAMETER CONSTRAINT # ',I3,', COULD NOT BE FOUND AMONGST'//
+      ' THE LEGAL LIST OF ATTRIBUTES FOR COMPONENT ',A2,', THIS'//
+      ' SPECIFICATION WILL BE IGNORED.')
132      RETURN
      C
133      260      WRITE (6,265) XPAR3,XPAR1,PAR2
134      265      FORMAT (' THE RELOP '"',A4,'" SPECIFIED BY THE USER '//
+      ' SPECIFICATION REFERRING TO COMPONENT ',A2,' AND ATTRIBUTE ',
+      ' 2A4/' WAS NOT AMONGST THE LIST OF LEGAL RELATIONAL'//
+      ' OPERATORS. THIS SPECIFICATION WILL BE IGNORED.')
135      RETURN
      C
136      270      WRITE (6,275) PAR2,IPAR3,RELNAM(IPAR1)
137      275      FORMAT (' A CONSTRAINT ASSOCIATED WITH ATTRIBUTE ',2A4/
+      ' OF RECORD TYPE ',I2,' SPECIFIED A '"',A2,'" RELATION'//
+      ' INVOLVING CHARACTER DATA.')
138      PAUSE
139      GO TO 9999
      C
140      280      WRITE (6,285) PAR1
141      285      FORMAT (' THE COMPONENT NAME ',2A4,' SPECIFIED IN THE EXISTING'//
+      ' VEHICLE FILE DOSE NOT MATCH ANY OF THE LEGAL COMPONENT'//
+      ' NAMES STORED.')
142      PAUSE
143      GO TO 9999
      C
144      290      WRITE (6,295) IPAR2,IPAR3,IPAR1
145      295      FORMAT (' A CONSTRAINT IN THE USER SPECIFICATION FILE INDICATED '//
+      ' THAT FIELD ',I3,' OF RECORD ID ',I3,' HAVING RECORD TYPE',
+      ' I4,' WAS TO BE NOT EQUAL A SPECIFIED VALUE. THIS IS NOT'//
+      ' A LEGAL CONSTRAINT TO SPECIFY IN THIS SITUATION.')
146      PAUSE
147      GO TO 9999
      C
148      300      WRITE (6,305) IPAR1

```

## UTILITY ROUTINES - ERR SUBROUTINE

```

149      ISN
150      149
151      150
152      151
153      152
154      153
155      154
156      155
157      156
158      157
159      158
160      159
161      160
162      161
163      162
164      163
165      164
166      165
167      166
168      167
169      168
170      169
171      170
172      171
173      172
174      173
175      174
176      175
177      176
178      177
179      178
180      179
181      180
182      181
183      182
184      183
185      184
186      185
187      186
188      187
189      188
190      189
191      190
192      191
193      192
194      193
195      194
196      195
197      196
198      197
199      198
200      199
201      200
202      201
203      202
204      203
205      204
206      205
207      206
208      207
209      208
210      209
211      210
212      211
213      212
214      213
215      214
216      215
217      216
218      217
219      218
220      219
221      220
222      221
223      222
224      223
225      224
226      225
227      226
228      227
229      228
230      229
231      230
232      231
233      232
234      233
235      234
236      235
237      236
238      237
239      238
240      239
241      240
242      241
243      242
244      243
245      244
246      245
247      246
248      247
249      248
250      249
251      250
252      251
253      252
254      253
255      254
256      255
257      256
258      257
259      258
260      259
261      260
262      261
263      262
264      263
265      264
266      265
267      266
268      267
269      268
270      269
271      270
272      271
273      272
274      273
275      274
276      275
277      276
278      277
279      278
280      279
281      280
282      281
283      282
284      283
285      284
286      285
287      286
288      287
289      288
290      289
291      290
292      291
293      292
294      293
295      294
296      295
297      296
298      297
299      298
300      299
301      300
302      301
303      302
304      303
305      304
306      305
307      306
308      307
309      308
310      309
311      310
312      311
313      312
314      313
315      314
316      315
317      316
318      317
319      318
320      319
321      320
322      321
323      322
324      323
325      324
326      325
327      326
328      327
329      328
330      329
331      330
332      331
333      332
334      333
335      334
336      335
337      336
338      337
339      338
340      339
341      340
342      341
343      342
344      343
345      344
346      345
347      346
348      347
349      348
350      349
351      350
352      351
353      352
354      353
355      354
356      355
357      356
358      357
359      358
360      359
361      360
362      361
363      362
364      363
365      364
366      365
367      366
368      367
369      368
370      369
371      370
372      371
373      372
374      373
375      374
376      375
377      376
378      377
379      378
380      379
381      380
382      381
383      382
384      383
385      384
386      385
387      386
388      387
389      388
390      389
391      390
392      391
393      392
394      393
395      394
396      395
397      396
398      397
399      398
400      399
401      400
402      401
403      402
404      403
405      404
406      405
407      406
408      407
409      408
410      409
411      410
412      411
413      412
414      413
415      414
416      415
417      416
418      417
419      418
420      419
421      420
422      421
423      422
424      423
425      424
426      425
427      426
428      427
429      428
430      429
431      430
432      431
433      432
434      433
435      434
436      435
437      436
438      437
439      438
440      439
441      440
442      441
443      442
444      443
445      444
446      445
447      446
448      447
449      448
450      449
451      450
452      451
453      452
454      453
455      454
456      455
457      456
458      457
459      458
460      459
461      460
462      461
463      462
464      463
465      464
466      465
467      466
468      467
469      468
470      469
471      470
472      471
473      472
474      473
475      474
476      475
477      476
478      477
479      478
480      479
481      480
482      481
483      482
484      483
485      484
486      485
487      486
488      487
489      488
490      489
491      490
492      491
493      492
494      493
495      494
496      495
497      496
498      497
499      498
500      499
501      500
502      501
503      502
504      503
505      504
506      505
507      506
508      507
509      508
510      509
511      510
512      511
513      512
514      513
515      514
516      515
517      516
518      517
519      518
520      519
521      520
522      521
523      522
524      523
525      524
526      525
527      526
528      527
529      528
530      529
531      530
532      531
533      532
534      533
535      534
536      535
537      536
538      537
539      538
540      539
541      540
542      541
543      542
544      543
545      544
546      545
547      546
548      547
549      548
550      549
551      550
552      551
553      552
554      553
555      554
556      555
557      556
558      557
559      558
560      559
561      560
562      561
563      562
564      563
565      564
566      565
567      566
568      567
569      568
570      569
571      570
572      571
573      572
574      573
575      574
576      575
577      576
578      577
579      578
580      579
581      580
582      581
583      582
584      583
585      584
586      585
587      586
588      587
589      588
590      589
591      590
592      591
593      592
594      593
595      594
596      595
597      596
598      597
599      598
600      599
601      600
602      601
603      602
604      603
605      604
606      605
607      606
608      607
609      608
610      609
611      610
612      611
613      612
614      613
615      614
616      615
617      616
618      617
619      618
620      619
621      620
622      621
623      622
624      623
625      624
626      625
627      626
628      627
629      628
630      629
631      630
632      631
633      632
634      633
635      634
636      635
637      636
638      637
639      638
640      639
641      640
642      641
643      642
644      643
645      644
646      645
647      646
648      647
649      648
650      649
651      650
652      651
653      652
654      653
655      654
656      655
657      656
658      657
659      658
660      659
661      660
662      661
663      662
664      663
665      664
666      665
667      666
668      667
669      668
670      669
671      670
672      671
673      672
674      673
675      674
676      675
677      676
678      677
679      678
680      679
681      680
682      681
683      682
684      683
685      684
686      685
687      686
688      687
689      688
690      689
691      690
692      691
693      692
694      693
695      694
696      695
697      696
698      697
699      698
700      699
701      700
702      701
703      702
704      703
705      704
706      705
707      706
708      707
709      708
710      709
711      710
712      711
713      712
714      713
715      714
716      715
717      716
718      717
719      718
720      719
721      720
722      721
723      722
724      723
725      724
726      725
727      726
728      727
729      728
730      729
731      730
732      731
733      732
734      733
735      734
736      735
737      736
738      737
739      738
740      739
741      740
742      741
743      742
744      743
745      744
746      745
747      746
748      747
749      748
750      749
751      750
752      751
753      752
754      753
755      754
756      755
757      756
758      757
759      758
760      759
761      760
762      761
763      762
764      763
765      764
766      765
767      766
768      767
769      768
770      769
771      770
772      771
773      772
774      773
775      774
776      775
777      776
778      777
779      778
780      779
781      780
782      781
783      782
784      783
785      784
786      785
787      786
788      787
789      788
790      789
791      790
792      791
793      792
794      793
795      794
796      795
797      796
798      797
799      798
800      799
801      800
802      801
803      802
804      803
805      804
806      805
807      806
808      807
809      808
810      809
811      810
812      811
813      812
814      813
815      814
816      815
817      816
818      817
819      818
820      819
821      820
822      821
823      822
824      823
825      824
826      825
827      826
828      827
829      828
830      829
831      830
832      831
833      832
834      833
835      834
836      835
837      836
838      837
839      838
840      839
841      840
842      841
843      842
844      843
845      844
846      845
847      846
848      847
849      848
850      849
851      850
852      851
853      852
854      853
855      854
856      855
857      856
858      857
859      858
860      859
861      860
862      861
863      862
864      863
865      864
866      865
867      866
868      867
869      868
870      869
871      870
872      871
873      872
874      873
875      874
876      875
877      876
878      877
879      878
880      879
881      880
882      881
883      882
884      883
885      884
886      885
887      886
888      887
889      888
890      889
891      890
892      891
893      892
894      893
895      894
896      895
897      896
898      897
899      898
900      899
901      900
902      901
903      902
904      903
905      904
906      905
907      906
908      907
909      908
910      909
911      910
912      911
913      912
914      913
915      914
916      915
917      916
918      917
919      918
920      919
921      920
922      921
923      922
924      923
925      924
926      925
927      926
928      927
929      928
930      929
931      930
932      931
933      932
934      933
935      934
936      935
937      936
938      937
939      938
940      939
941      940
942      941
943      942
944      943
945      944
946      945
947      946
948      947
949      948
950      949
951      950
952      951
953      952
954      953
955      954
956      955
957      956
958      957
959      958
960      959
961      960
962      961
963      962
964      963
965      964
966      965
967      966
968      967
969      968
970      969
971      970
972      971
973      972
974      973
975      974
976      975
977      976
978      977
979      978
980      979
981      980
982      981
983      982
984      983
985      984
986      985
987      986
988      987
989      988
990      989
991      990
992      991
993      992
994      993
995      994
996      995
997      996
998      997
999      998
1000     999

```

## UTILITY ROUTINES - ERR SUBROUTINE

```

174      ISN
175      C
176      WRITE (6,375) IPARI,XPAR3,XPAR2
177      FORMAT (' AN ATTEMPT TO TRANSLATE THE VALUE IN A FIELD OF A' /
+ ' RECORD OF RECORD TYPE ',I3,' FROM ',A4,' TO ',A4,' FAILED' /
+ ' BECAUSE THIS PAIR OF UNITS WAS NOT FOUND IN "TUNITS".')
      PAUSE
      GO TO 9999

178      C
179      WRITE (6,385) IPARI
180      FORMAT (' AN OUTPUT ARRAY WAS ENCOUNTERED WITH ',I3,' DIMENSIONS' /
+ ' BUT THE PROGRAM WAS WRITTEN TO HANDLE ONLY ARRAYS HAVING' /
+ ' 1 TO 3 DIMENSIONS.')
      PAUSE
      GO TO 9999

181      C
182      CONTINUE

183      C
184      WRITE (6,405) XPAR3,XPAR1,PAR2,PAR2
185      FORMAT (' THE DISTANCE (',F8.1,') SPECIFIED BY VARIABLE ',A4 /
+ ' ASSOCIATED WITH THE SELECTED ROAD WHEEL COMPONENT AND' /
+ ' GIVING THE VERTICAL DISTANCE BETWEEN THE ROAD WHEELS AND ' /
+ A8,',' WAS TOO SMALL TO KEEP THE ROAD WHEELS FROM INTERFERING' /
+ ' WITH THE ',A8)
      PAUSE
      GO TO 9999

186      C
187      WRITE (6,415) IPARI,MINNUM,MAXNUM
188      FORMAT (' THE NUMBER OF ROAD WHEELS SPECIFIED, ',I2,' WAS NOT' /
+ ' BETWEEN THE ALLOWED MINIMUM, ',I2,' AND MAXIMUM, ',I3)
      PAUSE
      GO TO 9999

189      C
190      WRITE (6,425) IPAR3,IPARI
191      FORMAT (' A SATISFACTORY CANDIDATE VEHICLE COULD NOT BE FOUND' /
+ ' BECAUSE NONE OF THE ',I2,' ALTERNATIVES OF COMPONENT TYPE' /
+ I3,' WERE FOUND TO BE SUITABLE.')

```

## UTILITY ROUTINES - ERR SUBROUTINE

```

15N
193      PAUSE
194      GO TO 9999
195      C
196      WRITE (6,435)
197      FORMAT (' NONE OF THE COMBINATIONS OF ALTERNATIVE COMPONENTS'//
198      + ' WERE BOTH CONSISTENT WITH EACH OTHER AND WITH THE '//
199      + ' USERS SPECIFICATIONS. THEREFORE A CANDIDATE VEHICLE'//
200      + ' SOLUTION WAS NOT FOUND. THE USER SHOULD RUN THE PROGRAM'//
201      + ' AGAIN, REMOVING SOME CONSTRAINTS.')
202      PAUSE
203      GO TO 9999
204      C
205      CONTINUE
206      I = IPAR1 - 1
207      WRITE (6,455) I
208      FORMAT (' THE STABILIZATION PERFORMANCE INDEX SPECIFIED IN THE '//
209      + ' SELECTED "STABLE" COMPONENT RECORD WAS OUT-OF-BOUNDS (NOT'//
210      + ' 1 OR 2). IT WAS: ',I3)
211      PAUSE
212      GO TO 9999
213      C
214      ENTRY WARN(IWARN,SUBR,PAR1,PAR2,PAR3)
215      C
216      EQUATE ARGS SO THAT INTEGER PARAMETERS ARE DEFINED
217      C
218      XPAR1 = PAR1(1)
219      XPAR2 = PAR2(1)
220      XPAR3 = PAR3
221      C
222      BRANCH TO APPROPRIATE PIECE OF CODE TO HANDLE WARNING MESSAGE
223      C
224      GO TO (510,520),IWARN
225      C
226      ICMP = IPAR1 - 1
227      WRITE (6,515) IPAR2,IPAR3,ICMP

```

## UTILITY ROUTINES - ERR SUBROUTINE

```

I SN      212      515      FORMAT (' A CONSTRAINT SPECIFIED IN THE USER SPECIFICATION FILE' /
+          ' HAS BEEN USED TO CHANGE THE VALUE OF FIELD ',I3,' OF ' /
+          ' RECCRE ID ',I3,' OF COMPONENT TYPE ',I3,'. IF THIS' /
+          ' MODIFICATION OF THE COMPONENT REQUIRES DEVELOPMENT TIME' /
+          ' OR COST, OR IF THE COMPONENT HAS BEEN USED TO DEVELOP' /
+          ' PERFORMANCE DATA FOR THIS RUN, ONE SHOULD TAKE THESE' /
+          ' FACTORS INTO ACCOUNT IN INTERPRETING THE OUTPUT OF THIS' ,
+          ' RUN.')
          213      RETURN
C
214      520      WRITE (6,525)
215      525      FORMAT (' THE GUN DEPRESSION ANGLE SPECIFIED IN THE SELECTED' /
+          ' MAIN GUN COULD NOT BE ACHIEVED WITH THE HULL & TURRET' /
+          ' DIMENSIONS GIVEN. THE MAIN GUNS HEIGHT ABOVE THE ',A8 /
+          ' OF THE HULL WAS THEREFORE INCREASED FROM ',F6.0,' INCHES' /
+          ' TO ',F6.0,' INCHES.')
          216      RETURN
C
217      9999      RETURN
218      END

```

## UTILITY ROUTINES - FVAL1 FUNCTION

```

1      ISN
1      FUNCTION FVAL1(X,R,N,V)
C
C      FUNCTION TO INTERPOLATE OR EXTRAPOLATE ON AN ARRAY AND RETRIEVE THE
C      FUNCTIONAL VALUE CORRESPONDING TO AN INDEPENDENT VARIABLE. THIS
C      VALUE IS RETURNED BY THE FUNCTION. THERE ARE TWO OTHER ENTRY POINTS,
C      FVAL2 & FVAL3, FOR FUNCTIONS OF TWO OR THREE VARIABLES. THIS
C      FUNCTION USES A ROUGH APPROXIMATION METHOD OF INTERPOLATION IN THESE
C      LATTER CASES.
C
C      INPUT PARAMETERS:
C      X      ARRAY STORING THE FUNCTION
C      R      VECTOR STORING VALUES OF DEPENDENT VARIABLES
C      N      NUMBER OF VALUES TABULATED IN R & N
C      V      INDEPENDENT VALUE TO USE IN INTERPOLATION
C
C      DIMENSION X(N),R(N)
C      DIMENSION XX(N1,N2),R1(N1),R2(N2)
C      DIMENSION XXX(N1,N2,N3),R3(N3)
C
C      DO 10 I=2,N
C      IF (V .GT. R(I) .AND. I .NE. N) GO TO 10
C      SLOPE = (X(I) - X(I-1)) / (R(I) - R(I-1))
C      FVAL1 = X(I-1) + (V - R(I-1))*SLOPE
C      RETURN
C      CONTINUE
C      PAUSE
C
C      ENTRY FVAL2(XX,R1,N1,V1,R2,N2,V2)
C
C      DO 30 I=2,N1
C      IF (V1 .GT. R1(I) .AND. I .NE. N1) GO TO 30
C      DO 20 J=2,N2
C      IF (V2 .GT. R2(J) .AND. J .NE. N2) GO TO 20
C      SLOPE1 = (XX(I,J) - XX(I-1,J)) / (R1(I) - R1(I-1))
C      SLOPE2 = (XX(I-1,J) - XX(I-1,J-1)) / (R2(J) - R2(J-1))

```

## UTILITY ROUTINES - FVAL1 FUNCTION

```

ISN
19      FVAL2 = XX(I-1,J-1) + (V1 - R1(I-1))*SLOPE1
      + (V2 - R2(J-1)) * SLOPE2
20      RETURN
21      CONTINUE
22      CONTINUE
23      PAUSE
      C
24      ENTRY FVAL3(XXX,R1,N1,V1,R2,N2,V2,R3,N3,V3)
      C
25      DO 60 I=2,N1
26      IF (V1 .GT. R1(I) .AND. I .NE. N1) GO TO 60
27      DO 50 J=2,N2
28      IF (V2 .GT. R2(J) .AND. J .NE. N2) GO TO 50
29      DO 40 K=2,N3
30      IF (V3 .GT. R3(K) .AND. K .NE. N3) GO TO 40
31      SLOPE1 = (XXX(I,J,K) - XXX(I-1,J,K)) / (R1(I) - R1(I-1))
32      SLOPE2 = (XXX(I-1,J,K) - XXX(I-1,J-1,K)) / (R2(J) - R2(J-1))
33      SLOPE3 = (XXX(I-1,J-1,K) - XXX(I-1,J-1,K-1)) /
      (R3(K) - R3(K-1))
      FVAL3 = XXX(I-1,J-1,K-1) + (V1 - R1(I-1))*SLOPE1
      + (V2 - R2(J-1)) * SLOPE2
      + (V3 - R3(K-1)) * SLOPE3
      RETURN
34      CONTINUE
      CONTINUE
      CONTINUE
      CONTINUE
      PAUSE
      END
      C
35
36      40
37      50
38      60
39
40

```

# UTILITY ROUTINES - IATTR FUNCTION

**FUNCTION IATTR(ATTR,CMP,I RECTP)**

FUNCTION FOR COMPUTING THE INDEX OF AN ATTRIBUTE WITHIN A COMPONENT RECORD. FUNCTION RETURNS THIS INDEX AS ITS VALUE OR RETURNS A NULL VALUE IF THE SPECIFIED ATTRIBUTE NAME DOES NOT IDENTIFY A LEGAL ATTRIBUTE OF THIS COMPONENT. FUNCTION ALSO RETURNS THE INDEX OF THE RECORD TYPE CORRESPONDING TO CMP.

### INPUT PARAMETERS:

| ATTR | ATTRIBUTE NAME |
|------|----------------|
|------|----------------|

[illegible]

### OUTPUT PARAMETERS:

| IRECTP | RECORD | TYPE | INDEX | OF | CMP |
|--------|--------|------|-------|----|-----|
| 1      | 1      | 1    | 1     | 1  | 1   |
| 2      | 2      | 2    | 2     | 2  | 2   |
| 3      | 3      | 3    | 3     | 3  | 3   |
| 4      | 4      | 4    | 4     | 4  | 4   |
| 5      | 5      | 5    | 5     | 5  | 5   |
| 6      | 6      | 6    | 6     | 6  | 6   |
| 7      | 7      | 7    | 7     | 7  | 7   |
| 8      | 8      | 8    | 8     | 8  | 8   |
| 9      | 9      | 9    | 9     | 9  | 9   |
| 10     | 10     | 10   | 10    | 10 | 10  |
| 11     | 11     | 11   | 11    | 11 | 11  |
| 12     | 12     | 12   | 12    | 12 | 12  |
| 13     | 13     | 13   | 13    | 13 | 13  |
| 14     | 14     | 14   | 14    | 14 | 14  |
| 15     | 15     | 15   | 15    | 15 | 15  |
| 16     | 16     | 16   | 16    | 16 | 16  |
| 17     | 17     | 17   | 17    | 17 | 17  |
| 18     | 18     | 18   | 18    | 18 | 18  |
| 19     | 19     | 19   | 19    | 19 | 19  |
| 20     | 20     | 20   | 20    | 20 | 20  |
| 21     | 21     | 21   | 21    | 21 | 21  |
| 22     | 22     | 22   | 22    | 22 | 22  |
| 23     | 23     | 23   | 23    | 23 | 23  |
| 24     | 24     | 24   | 24    | 24 | 24  |
| 25     | 25     | 25   | 25    | 25 | 25  |
| 26     | 26     | 26   | 26    | 26 | 26  |
| 27     | 27     | 27   | 27    | 27 | 27  |
| 28     | 28     | 28   | 28    | 28 | 28  |
| 29     | 29     | 29   | 29    | 29 | 29  |
| 30     | 30     | 30   | 30    | 30 | 30  |
| 31     | 31     | 31   | 31    | 31 | 31  |
| 32     | 32     | 32   | 32    | 32 | 32  |
| 33     | 33     | 33   | 33    | 33 | 33  |
| 34     | 34     | 34   | 34    | 34 | 34  |
| 35     | 35     | 35   | 35    | 35 | 35  |
| 36     | 36     | 36   | 36    | 36 | 36  |
| 37     | 37     | 37   | 37    | 37 | 37  |
| 38     | 38     | 38   | 38    | 38 | 38  |
| 39     | 39     | 39   | 39    | 39 | 39  |
| 40     | 40     | 40   | 40    | 40 | 40  |
| 41     | 41     | 41   | 41    | 41 | 41  |
| 42     | 42     | 42   | 42    | 42 | 42  |
| 43     | 43     | 43   | 43    | 43 | 43  |
| 44     | 44     | 44   | 44    | 44 | 44  |
| 45     | 45     | 45   | 45    | 45 | 45  |
| 46     | 46     | 46   | 46    | 46 | 46  |
| 47     | 47     | 47   | 47    | 47 | 47  |
| 48     | 48     | 48   | 48    | 48 | 48  |
| 49     | 49     | 49   | 49    | 49 | 49  |
| 50     | 50     | 50   | 50    | 50 | 50  |
| 51     | 51     | 51   | 51    | 51 | 51  |
| 52     | 52     | 52   | 52    | 52 | 52  |
| 53     | 53     | 53   | 53    | 53 | 53  |
| 54     | 54     | 54   | 54    | 54 | 54  |
| 55     | 55     | 55   | 55    | 55 | 55  |
| 56     | 56     | 56   | 56    | 56 | 56  |
| 57     | 57     | 57   | 57    | 57 | 57  |
| 58     | 58     | 58   | 58    | 58 | 58  |
| 59     | 59     | 59   | 59    | 59 | 59  |
| 60     | 60     | 60   | 60    | 60 | 60  |
| 61     | 61     | 61   | 61    | 61 | 61  |
| 62     | 62     | 62   | 62    | 62 | 62  |
| 63     | 63     | 63   | 63    | 63 | 63  |
| 64     | 64     | 64   | 64    | 64 | 64  |
| 65     | 65     | 65   | 65    | 65 | 65  |
| 66     | 66     | 66   | 66    | 66 | 66  |
| 67     | 67     | 67   | 67    | 67 | 67  |
| 68     | 68     | 68   | 68    | 68 | 68  |
| 69     | 69     | 69   | 69    | 69 | 69  |
| 70     | 70     | 70   | 70    | 70 | 70  |
| 71     | 71     | 71   | 71    | 71 | 71  |
| 72     | 72     | 72   | 72    | 72 | 72  |
| 73     | 73     | 73   | 73    | 73 | 73  |
| 74     | 74     | 74   | 74    | 74 | 74  |
| 75     |        |      |       |    |     |

ALSO THERE IS AN ENTRY POINT IATTR2 WHERE BOTH ATTR AND IRECTP ARE INPUT AND THE FIELD INDEX COMPUTED.

COMMON /VPROTO/

REAL#8 FLDNAM,RECNAM,DEFAULT

COMMON /NPARS/

COMMON /AUX/

REAL#8 BL8,DASHFS

EQUIVALENCE (FNULL, NULL), (IBLS, BLS)

REAL\*8 ATTR,CMP

DETERMINE INDEX OF RECORD STORING COMPONENT

ON 10 IRECTP=1,NRECTP

IF (RECNAME(I) .EQ. 'CMT') GO TO 15

CONTINUE

CALL ERR(6, 'IATTR ', CMP, ATTR, 0)

YATP = NULL

IRECTP = NULL

**RETURN**

NSI

1

2

3.

4

5.

6

၄

2

C

1

2

3

4

16

## UTILITY ROUTINES - IATTR FUNCTION

```

15N      C
16      C      ENTRY IATTR2(ATTR,I RECTP)
17      C
18      C      DETERMINE INDEX OF ATTRIBUTE WITHIN COMPONENT RECORD
19      C      J1 = IFLDPT(I RECTP) + 1
20      C      JN = IFLDPT(I RECTP+1)
21      C      DO 20 J=J1,JN
22      C      IF (ATTR.EQ. FLDNAM(J)) GO TO 25
23      C      CONTINUE
24      C      IATTR = NULL
25      C      IATTR2 = IATTR
26      C      RETURN
27      C      END
28
25      C
26      C      IATTR = IFLDVL(J)
27      C      IATTR2 = IATTR
28      C      RETURN
29      C      END

```

## UTILITY ROUTINES - IDVAL FUNCTION

```

15N 1
C
C      FUNCTION IDVAL(IFILE,NFLD,IFLD,IREF,IREFTP,UNIT,DF)
C
C      FUNCTION TO RETRIEVE A VALUE OF A SPECIFIED FIELD OF A RECORD.
C      IF THE VALUE IS NULL A DEFAULT VALUE IS SUBSTITUTED FOR IT. AN
C      INPUT TO THE FUNCTION ALSO INDICATES WHAT UNITS OF MEASURE ARE
C      DESIRED. IF THE VALUE IS NOT ALREADY IN THESE UNITS IT IS CONVERTED
C      TO THEM.
C
C      INPUT PARAMETERS:
C      IFILE  FILE IN WHICH VALUE IS LOCATED
C      NFLD   NUMBER OF FIELDS IN EACH RECORD OF FILE
C      IFLD   FIELD OF INTEREST
C      IREF   RECORD OF INTEREST
C      IREFTP RECORD TYPE
C      UNIT   UNIT OF MEASURE DESIRED
C
C      OUTPUT PARAMETERS:
C      DF     SET TO "*" IF DEFAULT VALUE IS USED
C
C      COMMON /VPROTO/
C      REAL*8 FLDNAM,RECNUM,DEFAULT
C      COMMON /NPARS/
C      COMMON /AUX/
C      REAL*8 BL8,DASHES
C      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
C
C      LOGICAL*1 STAR,DF
C      DIMENSION IFILE(NFLD,1)
C      DATA CASH/'-','/',STAR/'*'/
C
C      IDVAL = IFILE(IFLD,IREF)
C      IFLDRC = IFLDPT(IREFTP) + IFLD
C      IF (IDVAL .NE. NULL) GO TO 5
C      IDVAL = DEFAULT(IFLDRC)
C      DF = STAR

```

## UTILITY ROUTINES - IDVAL FUNCTION

```

15N 5      IF (UNIT.EQ. DASH .OR. UNIT.EQ. UNITS(IFLDRC) .OR. UNITS(IFLDRC)
16      +   .EQ. DASH) RETURN
17  C SEARCH TRANSLATION TABLE FOR UNIT TRANSLATION FACTOR
18      DO 10 I=1,NUTRNS
19      IF (UNIT.NE. TUNITS(1,I) .OR. UNITS(IFLDRC) .NE. TUNITS(2,I))
20      +   GO TO 8
21      IDVAL = IDVAL * TUNITS(3,I)
22      RETURN
23  8      IF (UNIT.NE. TUNITS(2,I) .OR. UNITS(IFLDRC) .NE. TUNITS(1,I))
24      +   GO TO 10
25      IDVAL = IDVAL / TUNITS(3,I)
26      RETURN
27  10     CONTINUE
      C
      C WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
      CALL ERR(37,'IDVAL ',IRECTP,UNIT,UNITS(IFLDRC))
      RETURN
      END

```

## UTILITY ROUTINES - IDVAL2 FUNCTION

```

1      ISN
2      1
3
4      FUNCTION IDVAL2(IX,ICMP,IFLD,UNIT)
5
6      C
7      C
8      C
9      C
10     C
11     C
12     C
13     C
14     C
15     C
16     C
17     C
18     C
19     C
20     C
21     C
22     C
23     C
24     C
25     C
26     C
27     C
28     C
29     C
30     C
31     C
32     C
33     C
34     C
35     C
36     C
37     C
38     C
39     C
40     C
41     C
42     C
43     C
44     C
45     C
46     C
47     C
48     C
49     C
50     C
51     C
52     C
53     C
54     C
55     C
56     C
57     C
58     C
59     C
60     C
61     C
62     C
63     C
64     C
65     C
66     C
67     C
68     C
69     C
70     C
71     C
72     C
73     C
74     C
75     C
76     C
77     C
78     C
79     C
80     C
81     C
82     C
83     C
84     C
85     C
86     C
87     C
88     C
89     C
90     C
91     C
92     C
93     C
94     C
95     C
96     C
97     C
98     C
99     C
100    C
101    C
102    C
103    C
104    C
105    C
106    C
107    C
108    C
109    C
110    C
111    C
112    C
113    C
114    C
115    C
116    C
117    C
118    C
119    C
120    C
121    C
122    C
123    C
124    C
125    C
126    C
127    C
128    C
129    C
130    C
131    C
132    C
133    C
134    C
135    C
136    C
137    C
138    C
139    C
140    C
141    C
142    C
143    C
144    C
145    C
146    C
147    C
148    C
149    C
150    C
151    C
152    C
153    C
154    C
155    C
156    C
157    C
158    C
159    C
160    C
161    C
162    C
163    C
164    C
165    C
166    C
167    C
168    C
169    C
170    C
171    C
172    C
173    C
174    C
175    C
176    C
177    C
178    C
179    C
180    C
181    C
182    C
183    C
184    C
185    C
186    C
187    C
188    C
189    C
190    C
191    C
192    C
193    C
194    C
195    C
196    C
197    C
198    C
199    C
200    C
201    C
202    C
203    C
204    C
205    C
206    C
207    C
208    C
209    C
210    C
211    C
212    C
213    C
214    C
215    C
216    C
217    C
218    C
219    C
220    C
221    C
222    C
223    C
224    C
225    C
226    C
227    C
228    C
229    C
230    C
231    C
232    C
233    C
234    C
235    C
236    C
237    C
238    C
239    C
240    C
241    C
242    C
243    C
244    C
245    C
246    C
247    C
248    C
249    C
250    C
251    C
252    C
253    C
254    C
255    C
256    C
257    C
258    C
259    C
260    C
261    C
262    C
263    C
264    C
265    C
266    C
267    C
268    C
269    C
270    C
271    C
272    C
273    C
274    C
275    C
276    C
277    C
278    C
279    C
280    C
281    C
282    C
283    C
284    C
285    C
286    C
287    C
288    C
289    C
290    C
291    C
292    C
293    C
294    C
295    C
296    C
297    C
298    C
299    C
300    C
301    C
302    C
303    C
304    C
305    C
306    C
307    C
308    C
309    C
310    C
311    C
312    C
313    C
314    C
315    C
316    C
317    C
318    C
319    C
320    C
321    C
322    C
323    C
324    C
325    C
326    C
327    C
328    C
329    C
330    C
331    C
332    C
333    C
334    C
335    C
336    C
337    C
338    C
339    C
340    C
341    C
342    C
343    C
344    C
345    C
346    C
347    C
348    C
349    C
350    C
351    C
352    C
353    C
354    C
355    C
356    C
357    C
358    C
359    C
360    C
361    C
362    C
363    C
364    C
365    C
366    C
367    C
368    C
369    C
370    C
371    C
372    C
373    C
374    C
375    C
376    C
377    C
378    C
379    C
380    C
381    C
382    C
383    C
384    C
385    C
386    C
387    C
388    C
389    C
390    C
391    C
392    C
393    C
394    C
395    C
396    C
397    C
398    C
399    C
400    C
401    C
402    C
403    C
404    C
405    C
406    C
407    C
408    C
409    C
410    C
411    C
412    C
413    C
414    C
415    C
416    C
417    C
418    C
419    C
420    C
421    C
422    C
423    C
424    C
425    C
426    C
427    C
428    C
429    C
430    C
431    C
432    C
433    C
434    C
435    C
436    C
437    C
438    C
439    C
440    C
441    C
442    C
443    C
444    C
445    C
446    C
447    C
448    C
449    C
450    C
451    C
452    C
453    C
454    C
455    C
456    C
457    C
458    C
459    C
460    C
461    C
462    C
463    C
464    C
465    C
466    C
467    C
468    C
469    C
470    C
471    C
472    C
473    C
474    C
475    C
476    C
477    C
478    C
479    C
480    C
481    C
482    C
483    C
484    C
485    C
486    C
487    C
488    C
489    C
490    C
491    C
492    C
493    C
494    C
495    C
496    C
497    C
498    C
499    C
500    C
501    C
502    C
503    C
504    C
505    C
506    C
507    C
508    C
509    C
510    C
511    C
512    C
513    C
514    C
515    C
516    C
517    C
518    C
519    C
520    C
521    C
522    C
523    C
524    C
525    C
526    C
527    C
528    C
529    C
530    C
531    C
532    C
533    C
534    C
535    C
536    C
537    C
538    C
539    C
540    C
541    C
542    C
543    C
544    C
545    C
546    C
547    C
548    C
549    C
550    C
551    C
552    C
553    C
554    C
555    C
556    C
557    C
558    C
559    C
560    C
561    C
562    C
563    C
564    C
565    C
566    C
567    C
568    C
569    C
570    C
571    C
572    C
573    C
574    C
575    C
576    C
577    C
578    C
579    C
580    C
581    C
582    C
583    C
584    C
585    C
586    C
587    C
588    C
589    C
590    C
591    C
592    C
593    C
594    C
595    C
596    C
597    C
598    C
599    C
600    C
601    C
602    C
603    C
604    C
605    C
606    C
607    C
608    C
609    C
610    C
611    C
612    C
613    C
614    C
615    C
616    C
617    C
618    C
619    C
620    C
621    C
622    C
623    C
624    C
625    C
626    C
627    C
628    C
629    C
630    C
631    C
632    C
633    C
634    C
635    C
636    C
637    C
638    C
639    C
640    C
641    C
642    C
643    C
644    C
645    C
646    C
647    C
648    C
649    C
650    C
651    C
652    C
653    C
654    C
655    C
656    C
657    C
658    C
659    C
660    C
661    C
662    C
663    C
664    C
665    C
666    C
667    C
668    C
669    C
670    C
671    C
672    C
673    C
674    C
675    C
676    C
677    C
678    C
679    C
680    C
681    C
682    C
683    C
684    C
685    C
686    C
687    C
688    C
689    C
690    C
691    C
692    C
693    C
694    C
695    C
696    C
697    C
698    C
699    C
700    C
701    C
702    C
703    C
704    C
705    C
706    C
707    C
708    C
709    C
710    C
711    C
712    C
713    C
714    C
715    C
716    C
717    C
718    C
719    C
720    C
721    C
722    C
723    C
724    C
725    C
726    C
727    C
728    C
729    C
730    C
731    C
732    C
733    C
734    C
735    C
736    C
737    C
738    C
739    C
740    C
741    C
742    C
743    C
744    C
745    C
746    C
747    C
748    C
749    C
750    C
751    C
752    C
753    C
754    C
755    C
756    C
757    C
758    C
759    C
760    C
761    C
762    C
763    C
764    C
765    C
766    C
767    C
768    C
769    C
770    C
771    C
772    C
773    C
774    C
775    C
776    C
777    C
778    C
779    C
780    C
781    C
782    C
783    C
784    C
785    C
786    C
787    C
788    C
789    C
790    C
791    C
792    C
793    C
794    C
795    C
796    C
797    C
798    C
799    C
800    C
801    C
802    C
803    C
804    C
805    C
806    C
807    C
808    C
809    C
810    C
811    C
812    C
813    C
814    C
815    C
816    C
817    C
818    C
819    C
820    C
821    C
822    C
823    C
824    C
825    C
826    C
827    C
828    C
829    C
830    C
831    C
832    C
833    C
834    C
835    C
836    C
837    C
838    C
839    C
840    C
841    C
842    C
843    C
844    C
845    C
846    C
847    C
848    C
849    C
850    C
851    C
852    C
853    C
854    C
855    C
856    C
857    C
858    C
859    C
860    C
861    C
862    C
863    C
864    C
865    C
866    C
867    C
868    C
869    C
870    C
871    C
872    C
873    C
874    C
875    C
876    C
877    C
878    C
879    C
880    C
881    C
882    C
883    C
884    C
885    C
886    C
887    C
888    C
889    C
890    C
891    C
892    C
893    C
894    C
895    C
896    C
897    C
898    C
899    C
900    C
901    C
902    C
903    C
904    C
905    C
906    C
907    C
908    C
909    C
910    C
911    C
912    C
913    C
914    C
915    C
916    C
917    C
918    C
919    C
920    C
921    C
922    C
923    C
924    C
925    C
926    C
927    C
928    C
929    C
930    C
931    C
932    C
933    C
934    C
935    C
936    C
937    C
938    C
939    C
940    C
941    C
942    C
943    C
944    C
945    C
946    C
947    C
948    C
949    C
950    C
951    C
952    C
953    C
954    C
955    C
956    C
957    C
958    C
959    C
960    C
961    C
962    C
963    C
964    C
965    C
966    C
967    C
968    C
969    C
970    C
971    C
972    C
973    C
974    C
975    C
976    C
977    C
978    C
979    C
980    C
981    C
982    C
983    C
984    C
985    C
986    C
987    C
988    C
989    C
990    C
991    C
992    C
993    C
994    C
995    C
996    C
997    C
998    C
999    C
1000   C

```

## UTILITY ROUTINES - IDVAL2 FUNCTION

```

18      +      GO TO 10
19      IDVAL2 = IDVAL2 / TUNITS(3,I)
20      RETURN
21      CONTINUE
22      C
23      C WRITE ERROR MESSAGE THAT UNIT TRANSLATION NOT FOUND
24      CALL ERR(37,'IDVAL2',ICMP+1,UNIT,UNITS(IFLDRC))
25      RETURN
26      END

```

## UTILITY ROUTINES - INITVL SUBROUTINE

```

TSN
1  C
   C SURROUTINE INITVL(IVEC,N,IVAL)
   C
   C SUBROUTINE TO INITIALIZE ALL THE ELEMENTS OF AN INTEGER ARRAY TO
   C A SPECIFIED VALUE.
   C
   C INPUT PARAMETERS:
   C IVEC A BLOCK OF CONTIGUOUS STORAGE TO BE INITIALIZED
   C      (IT NEED NOT BE A SINGLE I-DIM ARRAY IN THE CALLING
   C ROUTINE, THOUGH IT IS TREATED AS SUCH IN INITVL).
   C N THE NUMBER OF COMPUTER WORDS IN IVEC TO BE INITIALIZED.
   C IVAL THE VALUE TO SET EACH OF THE WORDS OF IVEC TO.
   C
   C DIMENSION IVEC(N)
   C DO 10 I=1,N
   C     IVEC(I) = IVAL
   C RETURN
   C END
2
3
4 10
5
6

```

## UTILITY ROUTINES - INLIST FUNCTION

```

1  ISM
2
3  FUNCTION INLIST(IVAL,IFILE,NF,IFLD,JNEXT,IREC)
4
5  ROUTINE SEARCHES A LINKED LIST OF RECORDS FOR A RECORD WITH A
6  SPECIFIED VALUE IN A SPECIFIED FIELD. FUNCTION RETURNS A POINTER
7  TO THE RECORD IF IT FINDS ONE, NULL OTHERWISE.
8
9  INPUT PARAMETERS:
10  IVAL  VALUE TO LOOK FOR
11  IFILE  FILE IN WHICH RECORDS CHAINED TOGETHER ARE LOCATED
12  NF     NUMBER OF FIELDS IN EACH RECORD
13  IFLD   FIELD IN WHICH TO SEARCH FOR VALUE
14  JNEXT  FIELD USED FOR CHAINING RECORDS
15  IREC   FIRST RECORD IN LIST
16
17  COMMON /AUX/
18  REAL*8 BL8,DASHES
19  EQUIVALENCE (FNULL,NULL),(IRLS,BLS)
20
21  DIMENSION IFILE(NF,1)
22
23  INLIST = IREC
24  IF (INLIST.EQ. NULL) RETURN
25  IF (IFILE(IFLD,INLIST).EQ. IVAL) RETURN
26  INLIST = IFILE(JNEXT,INLIST)
27  GO TO 5
28  END

```

## UTILITY ROUTINES - FUNCTION IOKVAL

ISN

1

LOGICAL FUNCTION IOKVAL(IVALUE,IRECTP,IFLD,ISAV)

C

FUNCTION TO TEST WHETHER THE USER HAS SPECIFIED ANY CONSTRAINTS  
ON A PARTICULAR ATTRIBUTE'S VALUE AND IF SO WHETHER THE SPECIFIED  
VALUE SATISFIES THESE CONSTRAINTS. THE FUNCTION RETURNS .TRUE. IF NO  
CONSTRAINTS ARE SPECIFIED OR IF THE INDICATED VALUE SATISFIES ALL  
CONSTRAINTS.

C

INPUT PARAMETERS:

IVALUE VALUE TO BE TESTED

IRECTP RECORD TYPE IN WHICH VALUE IS LOCATED

IFLD FIELD OF RECORD IN WHICH VALUE LOCATED

ISAV RECORD NUMBER WHICH RELATION REFERENCE STORED

C

2 COMMON /VEH/

3 DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),

+ FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),

+ ISENSR(32,9), ISTABL(30,6), IGUNCTL(28,9), IAMMO(30,30),

+ IAMMOC(26,9), IENGINE(40,8), ITRANS(32,8), IFINDR(28,10),

+ IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),

+ IFUEL(28,10), IFUELCL(26,9), ICREW(26,5), ICARGO(26,5),

+ IELECT(26,6), ICCMMO(28,9), IFIREX(28,10), IENVIR(26,9),

+ IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),

+ IRELN(3,50), VFILE(1), IVFILE(1)

4 EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),

+ ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),

+ (MISGUN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),

+ IRANGR(1)), (SENSOR(1), ISENSR(1)), (STABLE(1), ISTABL(1)),

+ (GUNCTL(1), IGUNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),

+ IAMMOC(1)), (ENGINE(1), IENGINE(1)), (TRANSM(1), ITRANS(1)),

+ (FINLDR(1), IFINDR(1)), (ROADWH(1), IROADWH(1)), (SPRING(1),

+ ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),

+ (FUEL(1), IFUEL(1)), (FUELCL(1), IFUELCL(1)), (CREW(1),

+ ICREW(1)), (CAPGO(1), ICARGO(1)), (ELECTP(1), IELECT(1)),

+ (COMMO(1), ICCMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),

## UTILITY ROUTINES - FUNCTION IOKVAL

```

13      IOKVAL = .TRUE.
14
15      C
16      C GET CONSTRAINT RECORD IF ONE EXISTS
17      NF = NFLDS(IIRECTP)
18      IOFF = IFILPT(IIRECTP)
19
20      C
21      C TEST IF RELATION REFERENCE PASSED
22      C
23      IF (ISAV .EQ. NULL) GO TO 3
24      IREC=ISAV
25      GO TO 4
26
27      3      IREC = ISPECV
28      IF (IRECTP .GT. 1) IREC = IVECLE(IIRECTP-1, ISPECV)
29      IF (IRECTP .EQ. NULL) RETURN
30      IR = IVAL(VFILE(IOFF),NF,IFLD,IREC)
31      IF (IR .EQ. NULL) RETURN
32      IV = IRELN(JVALUE,IR)
33      IRTYP = IRELN(JRELOP,IR)
34      GO TO (10,15,20,25,30,35),IRTP
35      IF (IVALUE .LT. IV) GO TO 40
36      IOKVAL = .FALSE.
37      RETURN
38
39      10      IF (IVALUE .LE. IV) GO TO 40
40      IOKVAL = .FALSE.
41      RETURN
42
43      + IENVIR(1),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
44      + (SMOKEG(1),ISMOKE(1)),(EWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
45      EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
46      COMMON /VPROTIN/
47      REAL*8 FLDNAM,RECNAM,DEFAULT
48      COMMON /JPARS/
49      COMMON /JPARS1/
50      COMMON /AUX/
51      REAL*8 BL8,DASHES
52      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)

```

## UTILITY ROUTINES - FUNCTION IOKVAL

```

ISN      20      IF ( IVALUE .EQ. IV ) GO TO 40
33      IOKVAL = .FALSE.
34      RETURN
35
36      25      IF ( IVALUE .NE. IV ) GO TO 40
37      IOKVAL = .FALSE.
38      RETURN
39
40      30      IF ( IVALUE .GE. IV ) GO TO 40
41      IOKVAL = .FALSE.
42      RETURN
43
44      35      IF ( IVALUE .GT. IV ) GO TO 40
45      IOKVAL = .FALSE.
46      RETURN
47
C
C GET NEXT CCNSTRANT
45      40      IR = IRELN(JNEXT,IR)
46      GO TO 5
47      END

```

## UTILITY ROUTINES - IRFIND FUNCTION

```

ISN
1  FUNCTION IRFIND(IRECTP, ID, IVEH)
C
C  ROUTINE TO SEARCH FOR A COMPONENT RECORD HAVING A GIVEN ID
C  STORED IN THE USER SPECIFICATION VEHICLE DESCRIPTION. RETURNS A
C  POINTER TO THIS RECORD OR NULL IF SUCH A RECORD DOESN'T EXIST.
C
C  INPUT PARAMETERS:
C  IRECTP  RECORD TYPE
C  ID      ID TO LOOK FOR
C  IVEH    VEHICLE DESCRIPTION IN WHICH TO SEARCH FOR RECORD
C
2  COMMON /VEH/
3  DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
+    FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
+    ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),
+    IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),
+    IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
+    IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
+    IELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),
+    IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
+    IRELN(3,50), VFILE(1), IVFILE(1)
4  EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),
+    ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),
+    (MISGUN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
+    IRANGR(1)), (SENSOR(1), ISENSR(1)), (STABLE(1), ISTABL(1)),
+    (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
+    IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANSM(1), ITRANS(1)),
+    (FINLDR(1), IFINDR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),
+    ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),
+    (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
+    ICREW(1)), (CAPGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
+    (COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),
+    IENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
+    (SMCKEG(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))
5  EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))

```

## UTILITY ROUTINES - IRFIND FUNCTION

```

1 SN
6      COMMON /VPROTIO/
7      REAL*8 FLDNAM,RECNAME,DEFAULT
8      COMMON /JPARS/
9      COMMON /JPARS1/
10     COMMON /NPARS/
11     COMMON /OUTPLT/
12     LOGICAL*1 DF
13     REAL*8 XNAT,XMANUF,XMODEL
14     COMMON /AUX/
15     REAL*8 BL8,CASHES
16     EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
C
17     ICMP = IRECTP - 1
18     IRFIND = IVECLE(ICMP,IVEH)
C   ARE THERE ANY COMPONENT RECORDS IN LIST ?
19     IF (IRFIND.EQ.NULL) RETURN
20     IOFF = IFILPT(IRECTP)
21     NF = NFIDS(IRECTP)
C   TEST WHETHER ID MATCHES THAT OF CURRENT RECORD
22     10   KID = IVAL(VFILE(IOFF),NF,JID,IRFIND)
23     IF (ID.EQ.KID) RETURN
C   TRY NEXT RECCPD
24     IRFIND = IVAL(VFILE(IOFF),NF,JNEXT,IRFIND)
25     IF (IRFIND.NE.NULL) GO TO 10
26     RETURN
27     END

```



UTILITY ROUTINES - ITABLE FUNCTION

ISM  
20

END

```

1SN
1      UTILITY ROUTINES - IVAL FUNCTION
      FUNCTION IVAL(IFILE,NFLD,IFLD,IREC)
      C
      C      FUNCTION TO RETRIEVE AND RETURN THE VALUE OF A SPECIFIED
      C      INTEGER FIELD OF A SPECIFIED RECORD IN A SPECIFIED FILE (ARRAY).
      C
      C      INPUT PARAMETERS:
      C      IFILE  FILE IN WHICH VALUE IS LOCATED
      C      NFLD   NUMBER OF FIELDS IN EACH RECORD OF FILE
      C      IFLD   FIELD OF INTEREST
      C      IREC   RECORD OF INTEREST
      C
      C      DIMENSION IFILE(NFLD,1)
      C
      IVAL = IFILE(IFLD,IREC)
      RETURN
      END
2
3
4
5

```

## UTILITY ROUTINES - IZERO SUBROUTINE

```
1 SN  
1  
C SUBROUTINE IZERO(IARRAY,N)  
C  
C INITIALIZES AN INTEGER ARRAY TO ZERO  
C  
2 DIMENSION IARRAY(N)  
3 DO 10 I=1,N  
4 IARRAY(I) = 0  
5 RETURN  
6 END
```

```

13      ISN
1      1
2
3
4
5
6
7
8
9
10
11
12
13

```

UTILITY ROUTINES - LSERCH FUNCTION

```

      FUNCTION LSERCH(LIST,KEY,NLIST,NKEY)
      C
      C      FUNCTION TO SEARCH A LIST (VECTOR) FOR A SPECIFIED KEY. THE
      C      FUNCTION RETURNS THE INDEX OF THE KEY IN THE LIST IF SUCCESSFUL, OR
      C      A NULL VALUE IF NOT.
      C
      C      INPUT PARAMETERS:
      C      LIST      LIST OF ITEMS TO SEARCH
      C      KEY       KEY TO LOOK FOR
      C      NLIST     SIZE OF LIST
      C      NKEY      NUMBER OF WORDS CONTAINED BY EACH ITEM OF LIST & BY KEY
      C
      C      DIMENSION KEY(NKEY),LIST(NKEY,NLIST)
      C      DATA NULL/-1/
      C
      C      DO 10 I=1,NLIST
      C      DO 5 J=1,NKEY
      C      IF (KEY(J) .NE. LIST(J,I)) GO TO 10
      C      CONTINUE
      C      LSERCH = I
      C      RETURN
      C      CONTINUE
      C      LSERCH = NULL
      C      RETURN
      C      END

```

## UTILITY ROUTINES - NEWREC FUNCTION

```

1 SN
1  FUNCTION NEWREC(IRECTP)
C
C  ROUTINE FOR OBTAINING THE NEXT AVAILABLE RECORD OF THE
C  SPECIFIED TYPE FROM THE AVAILABLE SPACE LIST AND RETURNING
C  A POINTER TO IT.
C
C  INPUT PARAMETERS:
C  IRECTP  RECORD TYPE OF INTEREST
C
C  COMMON /VPROTO/
C  REAL*8 FLONAM,RECNAM,DEFAULT
C  COMMON /NPARS/
C
C  IF (IRECTP .LE. 1 .OR. IRECTP .GT. NRECTP) CALL ERR(7,
C  + 'NEWREC',IRECTP,0,0)
C  IF (LSTREC(IRECTP) .GE. NRECS(IRECTP)) CALL ERR(8,'NEWREC',
C  + IRECTP,NRECS(IRECTP),0)
C  LSTREC(IRECTP) = LSTREC(IRECTP) + 1
C  NEWREC = LSTREC(IRECTP)
C  RETURN
C  END
10

```

```

1      ISN
1      UTILITY ROUTINES - NEWREL FUNCTION
      FUNCTION NEWREL(IRECTP,IFLD,KREL,IVALUE,ID)
      C
      C      FUNCTION TO GET A NEW RELATION RECORD AND STORE IT AS
      C      THE VALUE (OR ONE OF A LIST OF VALUES) OF A SPECIFIED FIELD
      C      IN A SPECIFIED COMPONENT RECORD OF THE USER SPECIFICATION VEHICLE.
      C
      C      INPUT PARAMETERS
      C      IRECTP  RECORD TYPE
      C      IFLD   FIELD INDEX OF THE COMPONENT TYPE
      C      KREL   AN INTEGER FROM 1 TO 6 INDICATING THE TYPE OF RELATION
      C      IVALUE THE VALUE FIELD OF THE RELATION
      C      ID     COMPONENT IDENTIFIER IF RELATION REFERS TO A COMPONENT
      C
      C      COMMON /VEH/
      C      DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
      C      +      FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
      C      +      ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
      C      +      IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
      C      +      IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
      C      +      IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
      C      +      IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
      C      +      IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
      C      +      IRELN(3,50),VFILE(1),IVFILE(1)
      C      EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
      C      +      ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
      C      +      (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
      C      +      IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
      C      +      (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
      C      +      IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
      C      +      (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
      C      +      ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
      C      +      (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
      C      +      ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
      C      +      (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
      C      +      IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),

```

## UTILITY ROUTINES - NEWREL FUNCTION

```

15N      + (SMCKEG(1), ISMOKE(1)), (IEWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))
5      EQUIVALENCE (VEHICLE(1), VFILE(1), IVFILE(1))
6      COMMON /VPROTO/
7      REAL*8 FLDNAM, RECNAM, DEFAULT
8      COMMON /JPARS/
9      COMMON /JPARS1/
10     COMMON /NPARS/
11     COMMON /AUX/
12     REAL*8 BL8, DASHES
13     EQUIVALENCE (FNULL, NULL), (IBLS, BLS)

14C      C GET A NEW RELATION RECORD, IF THERE IS ONE.
15      IF (LSTREC(NRECTP) .GE. NRECS(NRECTP)) CALL ERR(30, 'NEWREL',
16      + LSTREC(NRECTP), 0, 0)
17      LSTREC(NRECTP) = LSTREC(NRECTP) + 1
18      NEWREL = LSTREC(NRECTP)
19      IOFF = IFILPT(IIRECTP)
20      NF = NFLDS(IIRECTP)
21      ICMP = IIRECTP - 1
22      IREC = ISPECV
23      IF (IRECTP .EQ. 1) GO TO 13
24      IREC = IVECLE(ICMP, ISPECV)
25      IF (IRECTP .NE. NULL) GO TO 5
26      C NO COMPONENT RECORD OF THIS TYPE MATCHING THE APPROPRIATE ID
27      C CURRENTLY EXISTS FOR THE USER
28      C SPECIFICATION VEHICLE. CREATE ONE TO ATTACH RELATION RECORD TO.
29      IREC = NEWREL(IIRECTP)
30      CALL STORE(IVECLE(ICMP, ISPECV), VFILE(IOFF), NF, JNEXT, IREC)
31      IVECLE(ICMP, ISPECV) = IREC
32      C STORE COMPONENT RECORD ID
33      CALL STORE(ID, VFILE(IOFF), NF, JID, IREC)
34      GO TO 15

35C      C IS THIS RECORD APPROPRIATE FOR ATTACHING CONSTRAINT TO ?
36      KID = IVAL(VFILE(IOFF), NF, JID, IREC)
37      IF (KID .NE. ID) GO TO 8

```

## UTILITY ROUTINES - NEWREL FUNCTION

```

ISN
C IF SO, CREATE A NEW RELATION RECORD AND STORE THE POINTER IN THE
C APPROPRIATE FIELD. IF A PREDEFINED ALTERNATE COMPONENT IS TO BE
C CHANGED, WARN THE USER.
    IF (ID.GT. NALTCS(ICMP)) GO TO 15
    CALL WARN(1,'NEWREL',IRECTP,IFLD,ID)
    CALL STORE(IVALUE,VFILE(IOFF),NF,IFLD,IREC)
    RETURN
C IF NOT, TRY OTHER COMPONENT RECORDS
    IREC = IVAL(VFILE(IOFF),NF,JNEXT,IREC)
    GO TO 2
    IRELN(JNEXT,NEWREL) = IVECLE(IFLD,ISPECV)
    IF (IRECTP.NE. 1) IRELN(JNEXT,NEWREL) =
+   IVALUE(IOFF + (NF * (IRECT-1)) + (IFLD-1))
    CALL STORE(NEWREL,VFILE(IOFF),NF,IFLD,IREC)
    IRELN(JRELCP,NEWREL) = KREL
    IRELN(JVALUE,NEWREL) = IVALUE
    RETURN
END

```

```

1SN
1      UTILITY ROUTINES - NEWVEH FUNCTION
      FUNCTION NEWVEH(I1,I2,I3,I4)
      FUNCTION TO GET A NEW RECCRD IN IRECS AND STORE
      INFORMATION IN IT. THE FUNCTION RETURNS A POINTER TO THE
      RECORD OBTAINED.
      C
      C INPUT PARAMETERS:
      C   I1  VALUE OF 1ST FIELD
      C   I2  VALUE OF 2ND FIELD
      C   I3  VALUE OF 3RD FIELD
      C   I4  VALUE OF 4TH FIELD
      C
      COMMON /SPECS2/
      REAL*8 VEHS
      C
      LSTVEH = LSTVEH + 1
      IF (LSTVEH .GT. NVR) CALL ERR(21,'NEWVEH',NVR,I4,0)
      NEWVEH = LSTVEH
      IRECS(JNXT,LSTVEH) = I1
      IRECS(JTYP,LSTVEH) = I2
      IRECS(JSPEC,LSTVEH) = I3
      IRECS(JVEH,LSTVEH) = I4
      RETURN
      END
2
3
4
5
6
7
8
9
10
11
12

```

```

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

## UTILITY ROUTINES - NULIFY SUBROUTINE

```

1  ISN
1  SUBROUTINE NULIFY(IBLOCK,N)
C
C  ROUTINE TO NULL OUT THE CONTENTS OF A BLOCK OF STORAGE.
C
C  INPUT PARAMETERS:
C  IBLOCK  LOCATION OF THE BEGINNING OF THE BLOCK TO BE NULLED.
C  N       SIZE OF THE BLOCK TO BE NULLED
C
C  DIMENSION IBLOCK(N)
C  DATA NULL/-1/
C
C  DO 10 I=1,N
C      IBLOCK(I) = NULL
C  RETURN
C  END
2
3
4
5
6
7

```

## UTILITY ROUTINES - FUNCTION OKVAL

ISN

1

LOGICAL FUNCTION OKVAL(VAL,IRECTP,IFLD,ISAV)

FUNCTION TO TEST WHETHER THE USER HAS SPECIFIED ANY CONSTRAINTS ON A PARTICULAR ATTRIBUTE'S VALUE AND IF SO WHETHER THE SPECIFIED VALUE SATISFIES THESE CONSTRAINTS. THE FUNCTION RETURNS .TRUE. IF NO CONSTRAINTS ARE SPECIFIED OR IF THE INDICATED VALUE SATISFIES ALL CONSTRAINTS.

## INPUT PARAMETERS:

VAL VALUE TO BE TESTED  
IRECTP RECORD TYPE IN WHICH VALUE IS LOCATED  
IFLD FIELD OF RECORD IN WHICH VALUE LOCATED  
ISAV RECORD NUMBER IF RELATION RECORD PASSED

## COMMON /VEH/

2

3

DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),  
FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),  
ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),  
IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),  
IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),  
IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),  
IELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),  
IDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),  
IRELN(3,50), VFILE(1), IVFILE(1)

4

EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURRET(1),  
ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),  
(MISGUN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),  
IRANGR(1)), (SENSOR(1), ISENSR(1)), (STABLE(1), ISTABL(1)),  
(GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),  
IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANSM(1), ITRANS(1)),  
(FINLDR(1), IFINDR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),  
ISPRNG(1)), (TPACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),  
(FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),  
ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),  
(COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),

## UTILITY ROUTINES - FUNCTION OKVAL

```

15      IENVIR(1)), (DIAGNS(1), IDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
16      +      (SMOKEG(1), ISMOKE(1)), (EWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))
17      EQUIVALENCE (VEFCLE(1), VFFILE(1), IVFILE(1))
18      COMMON /VPROTD/
19      REAL*8 FLDNAM, RECNAM, DEFAULT
20      COMMON /JPARS/
21      COMMON /JPARS1/
22      COMMON /AUX/
23      REAL*8 BL8, CASHES
24      EQUIVALENCE (FNULL, NULL), (IBLS, BLS)
25
26      OKVAL = .TRUE.
27
28      C
29      GET CONSTRAINT RECORD IF ONE EXISTS
30      NF = NFIDS(IIRECTP)
31      IOFF = IFILPT(IIRECTP)
32
33      C
34      TEST IF RELATION REFERENCE PASSED
35
36      C
37      IF (ISAV .EQ. NULL) GO TO 3
38      IREC=ISAV
39      GO TO 4
40
41      3      IREC = ISPCV
42      IF (IRECTP .GT. 1) IREC = IVECLE(IIRECTP-1, ISPECV)
43      IF (IRECTP .EQ. NULL) RETURN
44      IR = IVAL(VFFILE(IOFF), NF, IFLD, IREC)
45      IF (IR .EQ. NULL) RETURN
46      V = RELN(JVALUE, IR)
47      IRTYP = IRELN(JRELOP, IR)
48      GO TO (10, 15, 20, 25, 30, 35), IRTYP
49      IF (VAL .LT. V) GO TO 40
50      OKVAL = .FALSE.
51      RETURN
52
53      10      IF (VAL .LE. V) GO TO 40
54      OKVAL = .FALSE.
55      RETURN

```

## UTILITY ROUTINES - FUNCTION OKVAL

```

ISN  20  IF (VAL .EQ. V) GO TO 40
33      OKVAL = .FALSE.
34      RETURN
35
36  25  IF (VAL .NE. V) GO TO 40
37      OKVAL = .FALSE.
38      RETURN
39
40  30  IF (VAL .GE. V) GO TO 40
41      OKVAL = .FALSE.
42      RETURN
43
44  35  IF (VAL .GT. V) GO TO 40
      OKVAL = .FALSE.
      RETURN

C
C  GET NEXT CONSTRAINT
45  40  IR = IRELN(JNEXT,IR)
46      GO TO 5
47      END

```

## UTILITY ROUTINES - QUEUE SUBROUTINE

```

1  I SN
2
3  SUBROUTINE QUEUE(IR,LISTPT,IRECTP)
4
5  ROUTINE FOR QUEUEING A RECCRD ONTO A LIST.
6
7  INPUT PARAMETERS:
8  IR      THE RECORD TO BE QUEUED
9  LISTPT  A POINTER TO THE HEAD OF THE LIST
10 IRECTP  THE TYPE OF RECORD
11
12 COMMON /VEH/
13 DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
14 + FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
15 + ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
16 + IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
17 + IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
18 + IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,9),
19 + IELECT(26,6),ICCMC(28,9),IFIREX(28,10),IENVIR(26,9),
20 + IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
21 + IRELN(3,50),VFILE(1),IVFILE(1)
22 EQUIVALENCE (VEHICLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
23 + ITURET(1)),(MAINGN(1),FMAING(1)),(MACGUN(1),FMACGN(1)),
24 + (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
25 + IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
26 + (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
27 + IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
28 + (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
29 + ISPRNG(1)),(TPACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
30 + (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
31 + ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
32 + (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
33 + IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
34 + (SMCKEG(1),ISMOKE(1)),(IEWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
35 EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
36 COMMON /VPROTN/
37 REAL*8 FLDNAM,RECNAM,DFAULT

```

## UTILITY ROUTINES - QUEUE SUBROUTINE

```

15N      CCMON /JPARS/
8         COMMON /JPARS1/
9         CCMON /AUX/
10        REAL*8 BL8,DASHES
11        EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
12
13        C
14        IOFF = IFILPT(IRECTP)
15        NF = NFLOS(IRECTP)
16        IF (LISTPT.NE. NULL) GO TO 10
17        LISTPT = IR
18        CALL STORE(NULL,VFILE(IOFF),NF,JNEXT,IR)
19        RETURN
20
21        C
22        IO IREC = LISTPT
23        C GET VALUE OF NEXT FIELD OF RECORD
24        IRECL = IREC
25        IREC = IVAL(VFILE(IOFF),NF,JNEXT,IREC)
26        IF (IREC.NE. NULL) GO TO 15
27        CALL STORE(IR,VFILE(IOFF),NF,JNEXT,IRECL)
28        CALL STORE(NULL,VFILE(IOFF),NF,JNEXT,IR)
29        RETURN
30        END

```

```

UTILITY ROUTINES - STONAM SUBROUTINE

      C
      C     SUBROUTINE STONAM(CNAME,IVFLAG)
      C
      C     ROUTINE TO STORE AN OUTPUT NAME READ FROM THE PROTOTYPE
      C     DEFINITION FILE IN THE NEXT AVAILABLE RECORD OF THE CJTNAM ARRAY.
      C
      C     INPUT PARAMETERS:
      C       CNAME    NAME TO BE STORED
      C       IVFLAG   AN INDICATOR WHICH HAS THE VALUE (0) IF THERE IS NO VALUE
      C                 TO PRINT. IF THERE IS A VALUE TO PRINT THEN IFLDRC IS THE
      C                 INDEX OF THE FIELD IN THE RECORD.
      C
      C     COMMON /VPROTO/
      C       REAL*8 FLDNAM,RECNAME,DFAULT
      C       COMMON /NPARS/
      C
      C     DIMENSION CNAME(NOUTWD)
      C
      C       LSTOUT = LSTOUT + 1
      C       IF (LSTOUT.GT. NOUT) CALL ERR(31,'STONAM',NOUT,0,0)
      C       CALL COPY(CNAME,OUTNAM(1,LSTOUT),NOUTWD)
      C       IOUTF(LSTOUT) = IVFLAG
      C       RETURN
      C
      C

```

```

      UTILITY ROUTINES - STORE SUBROUTINE
      SUBROUTINE STORE(VALUE,FILE,NFLD,IFLD,IREC)
      ROUTINE TO STORE A VALUE IN A PARTICULAR FIELD OF A
      PARTICULAR RECORD OF A SPECIFIED FILE (ARRAY).
      INPUT PARAMETERS:
      VALUE  VALUE TO BE STORED
      FILE   FILE IN WHICH TO STORE VALUE
      NFLD   NUMBER OF FIELDS IN EACH RECORD OF FILE
      IFLD   FIELD IN WHICH TO STORE VALUE
      IREC   RECORD IN WHICH TO STORE VALUE
      DIMENSION FILE(NFLD,1)
      FILE(IFLD,IREC) = VALUE
      RETURN
      END

```

ISM

1

C

C

C

C

C

C

C

C

C

C

C

C

2

C

3

4

5



## UTILITY ROUTINES - SUMCMP FUNCTION

```

1  ISN
1  FUNCTION SUMCMP(IV,IFLD,UNIT)
C
C  FUNCTION TO SUM A COMMON ATTRIBUTE SUCH AS WEIGHT, VOLUME, OR
C  COST ACROSS ALL COMPONENTS OF A SPECIFIED VEHICLE.
C
C  INPUT PARAMETERS:
C  IV      VEHICLE
C  IFLD    FIELD OF RECORD TO SUM
C  UNIT    UNITS OF MEASURE DESIRED
C
2  COMMON /VEH/
3  DIMENSION IVECLE(72,3),IHULL(48,9),ITURET(42,9),FMAING(44,9),
+  FMACGN(40,10),FMISGN(40,7),IADGUN(40,6),IRANGR(28,7),
+  ISENSR(32,9),ISTABL(30,6),IGNCTL(28,9),IAMMO(30,30),
+  IAMMOC(26,9),IENGIN(40,8),ITRANS(32,8),IFINDR(28,10),
+  IROADW(36,10),ISPRNG(30,10),ITRACK(34,8),ISKIRT(28,8),
+  IFUEL(28,10),IFUELC(26,9),ICREW(26,5),ICARGO(26,5),
+  IELECT(26,6),ICOMMO(28,9),IFIREX(28,10),IENVIR(26,9),
+  IDIAGN(26,5),ISIGSP(26,24),ISMOKE(26,10),IEWSYS(26,10),
+  IRELN(3,50),VFILE(1),IVFILE(1)
4  EQUIVALENCE (VECLE(1),IVECLE(1)),(HULL(1),IHULL(1)),(TURRET(1),
+  ITURET(1)),(MAINGN(1),FMAING(1)),(MACGN(1),FMACGN(1)),
+  (MISGUN(1),FMISGN(1)),(ADGUN(1),IADGUN(1)),(RANGER(1),
+  IRANGR(1)),(SENSOR(1),ISENSR(1)),(STABLE(1),ISTABL(1)),
+  (GUNCTL(1),IGNCTL(1)),(AMMO(1),IAMMO(1)),(AMMOC(1),
+  IAMMOC(1)),(ENGINE(1),IENGIN(1)),(TRANSM(1),ITRANS(1)),
+  (FINLDR(1),IFINDR(1)),(ROADWH(1),IROADW(1)),(SPRING(1),
+  ISPRNG(1)),(TRACK(1),ITRACK(1)),(SKIRT(1),ISKIRT(1)),
+  (FUEL(1),IFUEL(1)),(FUELC(1),IFUELC(1)),(CREW(1),
+  ICREW(1)),(CARGO(1),ICARGO(1)),(ELECTR(1),IELECT(1)),
+  (COMMO(1),ICOMMO(1)),(FIREX(1),IFIREX(1)),(ENVIRC(1),
+  IENVIR(1)),(DIAGNS(1),IDIAGN(1)),(SIGSUP(1),ISIGSP(1)),
+  (SMOKEG(1),ISMOKE(1)),(IEWSYS(1),IEWSYS(1)),(RELN(1),IRELN(1))
5  EQUIVALENCE (VEHICLE(1),VFILE(1),IVFILE(1))
6  COMMON /VPROTO/

```

## UTILITY ROUTINES - SUMCMP FUNCTION

```

15M 7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

      REAL*8 FLDNAM,RECNAM,DEFAULT
      COMMON /JPARS/
      COMMON /JPARS1/
      COMMON /NPARS/
      COMMON /AUX/
      REAL*8 BLR,DASHES
      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)

      SUMCMP = 0.
      DO 20 ICMPT=1,NCMPTP
        IRECTP = ICMPT + 1
        NF = NFIDS(IRECTP)
        IOFF = IFILPT(IRECTP)
        IREC = IVECLE(ICMP,IV)
        IF (IREC .EQ. NULL) GO TO 20
        SUMCMP = SUMCMP + DVAL(VFILE(IOFF),NF,IFLD,IREC,IRECTP,UNIT,DF)
        IREC = IVAL(VFILE(IOFF),NF,JNEXT,IREC)
        GO TO 10
      CONTINUE
      RETURN
      END

```

FUNCTION SUMPD2(IX,Y,N)

FUNCTION TO CALCULATE THE SUM OF THE PRODUCT OF CORRESPONDING ELEMENTS IN TWO EQUAL SIZED ARRAYS.

EMENTS IN TWO EQUAL SIZED ARRAYS.

COMPUTES THE VECTOR PRODUCT IX\*Y.

**DIMENSION IX(N), Y(N)**

SUMP02 = 0.

00 10 1=1,N

$$\text{SUMPD2} = \text{SUMPD2} + \text{IX(I)} * \text{Y(I)}$$

RETIPN

END

```

1)  ISN
2)
3)
4)
5)
6)
7)

      UTILITY ROUTINES - SUMPRD FUNCTION

      FUNCTION SUMPRD(X,Y,N)
      C
      C  FUNCTION TO CALCULATE THE SUM OF THE PRODUCT OF CORRESPONDING
      C  ELEMENTS IN TWO EQUAL SIZED ARRAYS.
      C
      C
      C  COMPUTES THE VECTOR PRODUCT X*Y.
      C
      C
      C  DIMENSION X(N),Y(N)
      C  SUMPRD = 0.
      C  DO 10 I=1,N
      C      SUMPRD = SUMPRD + X(I)*Y(I)
      C  1)  RETURN
      C      END

```

```

ISN
1
    UTILITY ROUTINES - TRANSF FUNCTION
    FUNCTION TRANSF(VALUE,MULOP,FACTOR,ADDOP,TERM)
    ROUTINE TO TRANSFORM A VALUE BY MULTIPLYING IT BE A CONSTANT
    AND/OR BY ADDING OR SUBTRACTING A CONTANT FROM IT, IF SUCH
    OPERATIONS ARE SPECIFIED BY THE USER. THE FUNCTION RETURNS THE
    TRANSFORMED VALUE.
    C
    C INPUT PARAMETERS:
    C VALUE VALUE TO BE TRANSFORMED (POSSIBLY)
    C MULOP MULTIPLY OPERATOR (OR BLANK)
    C FACTOR CONSTANT TO BE MULTIPLIED
    C ADDOP + OR - OPERATOR (OR BLANK)
    C TERM CONSTANT TO BE ADDED OR SUBTRACTED
    C
    COMMON /AUX/
    REAL*8 BL8,DASHES
    EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
    TRANSF = VALUE
    IF (MULOP .NE. MULT) GO TO 10
    TRANSF = TRANSF * FACTOR
    C
    IF (ADDOP .NE. PLUS) GO TO 20
    TRANSF = TRANSF + TERM
    RETURN
    C
    IF (ADDOP .NE. SMINUS) RETURN
    TRANSF = TRANSF - TERM
    RETURN
    END

```

## UTILITY ROUTINES - UNHOOK SUBROUTINE

```

1      JSN
2      SUBROUTINE UNHOOK(ICMP)
3
4      ROUTINE TO UNCHAIN COMPONENT RECORDS OF A GIVEN TYPE CHAINED
5      TOGETHER AS PART OF THE SPECIFICATION VEHICLE DESCRIPTION WHENEVER
6      BACKTRACKING IS NECESSARY.
7
8      INPUT PARAMETERS:
9      ICMP      COMPONENT TYPE
10
11      COMMON /VEH/
12      DIMENSION IVECLE(72,3), IHULL(48,9), ITURET(42,9), FMAING(44,9),
13      + FMACGN(40,10), FMISGN(40,7), IADGUN(40,6), IRANGR(28,7),
14      + ISENSR(32,9), ISTABL(30,6), IGNCTL(28,9), IAMMO(30,30),
15      + IAMMOC(26,9), IENGIN(40,8), ITRANS(32,8), IFINDR(28,10),
16      + IROADW(36,10), ISPRNG(30,10), ITRACK(34,8), ISKIRT(28,8),
17      + IFUEL(28,10), IFUELC(26,9), ICREW(26,5), ICARGO(26,5),
18      + IELECT(26,6), ICOMMO(28,9), IFIREX(28,10), IENVIR(26,9),
19      + ICDIAGN(26,5), ISIGSP(26,24), ISMOKE(26,10), IEWSYS(26,10),
20      + IRELN(3,50), VFIL(1), IVFIL(1)
21
22      EQUIVALENCE (VEHICLE(1), IVECLE(1)), (HULL(1), IHULL(1)), (TURET(1),
23      + ITURET(1)), (MAINGN(1), FMAING(1)), (MACGUN(1), FMACGN(1)),
24      + (MISGUN(1), FMISGN(1)), (ADGUN(1), IADGUN(1)), (RANGER(1),
25      + IRANGR(1)), (SENSOR(1), ISENSR(1)), (STABLE(1), ISTABL(1)),
26      + (GUNCTL(1), IGNCTL(1)), (AMMO(1), IAMMO(1)), (AMMOC(1),
27      + IAMMOC(1)), (ENGINE(1), IENGIN(1)), (TRANSM(1), ITRANS(1)),
28      + (FINLDR(1), IFINDR(1)), (ROADWH(1), IROADW(1)), (SPRING(1),
29      + ISPRNG(1)), (TRACK(1), ITRACK(1)), (SKIRT(1), ISKIRT(1)),
30      + (FUEL(1), IFUEL(1)), (FUELC(1), IFUELC(1)), (CREW(1),
31      + ICREW(1)), (CARGO(1), ICARGO(1)), (ELECTR(1), IELECT(1)),
32      + (COMMO(1), ICOMMO(1)), (FIREX(1), IFIREX(1)), (ENVIRC(1),
33      + IENVIR(1)), (DIAGNS(1), ICDIAGN(1)), (SIGSUP(1), ISIGSP(1)),
34      + (SMOKEG(1), ISMOKE(1)), (IEWSYS(1), IEWSYS(1)), (RELN(1), IRELN(1))
35
36      EQUIVALENCE (VEHICLE(1), VFIL(1), IVFIL(1))
37      COMMON /VPROTO/
38      REAL #8 FLDNAM, RECNAM, DEFAULT

```

## UTILITY ROUTINES - UNHOOK SUBROUTINE

```

10      COMMON /JPARS/
11      COMMON /JPARS1/
12      COMMON /AUX/
13      REAL*8 BL8,CASHES
14      EQUIVALENCE (FNULL,NULL),(IBLS,BLS)
15
16      C
17
18      IR = IVECLE(ICMP,ISPECV)
19      IF (IR.EQ. NULL) RETURN
20      IVECLE(ICMP,ISPECV) = NULL
21      IRECTP = ICMP + 1
22      NF = NFLDS(IRECTP)
23      IOFF = IFILPT(IRECTP)
24      NEXT = IVAL(VFILE(IOFF),NF,JNEXT,IR)
25      CALL STORE(NULL,VFILE(IOFF),NF,JNEXT,IR)
26      IF (NEXT.EQ. NULL) RETURN
27      IR = NEXT
28      GO TO 10
29      END

```

## UTILITY ROUTINES - ZERO SUBROUTINE

|   |     |    |  |
|---|-----|----|--|
| 1 | ISN |    |  |
| 1 |     |    |  |
|   |     | C  |  |
|   |     | C  |  |
|   |     | C  |  |
| 2 |     |    |  |
| 3 |     |    |  |
| 4 |     | 10 |  |
| 5 |     |    |  |
| 6 |     |    |  |

```

SUBROUTINE ZERO(V,N)
INITIALIZES AN ARRAY TO ZERO
DIMENSION V(N)
DO 10 I=1,N
  V(I) = 0.
RETURN
END

```

DISTRIBUTION LIST

## DISTRIBUTION LIST

Please notify USATARADCOM, DROTA-ZC, Warren, MI 48090, of corrections and/or changes in address.

No. of  
Copies

|    |                                                                                                                                                                   |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13 | Commander<br>US Army Tank-Automotive Research and Development Command<br>Attention: DRDTA-UL<br>Warren, MI 48090                                                  |
| 17 | Commander<br>US Army Tank-Automotive Research and Development Command<br>Attention: DRDTA-Z<br>Warren, MI 48090                                                   |
| 3  | Director<br>US Army Material Systems Analysis Activity<br>Attention: DRXS-DM (Messrs. D. Woomert, W. Niemeyer, W. Criswell)<br>Aberdeen Proving Grounds, MD 21005 |
| 1  | Director<br>US Army Human Engineering Laboratory<br>Attention: Mr. A. Eckels<br>Aberdeen Proving Grounds, MD 21005                                                |
| 2  | Director<br>US Army Ballistic Research Laboratory<br>Attention: Dr. A. Rainis, Mr. G. Bowers<br>Aberdeen Proving Grounds, MD 21005                                |
| 2  | Director<br>Defense Documentation Center<br>Cameron Station<br>Alexandria, VA 22314                                                                               |

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                       | READ INSTRUCTIONS<br>BEFORE COMPLETING FORM                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|--------------------------------------------------------------------------|
| 1. REPORT NUMBER<br>12444                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER                                            |
| 4. TITLE (and Subtitle)<br>PARAMETRIC ENGINEERING SYSTEM DEFINITION MODEL<br>VOLUME I--MAIN REPORT, APPENDICES A & B<br>VOLUME II--APPENDIX C                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                       | 5. TYPE OF REPORT & PERIOD COVERED<br>Final Report, 4/19/78 -<br>9/29/78 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                       | 6. PERFORMING ORG. REPORT NUMBER<br>VRI-TARADCOM-1 FR78-1                |
| 7. AUTHOR(s)<br>S. Spaulding, A. Weintraub, F. Cioch, J. Lenz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                       | 8. CONTRACT OR GRANT NUMBER(s)<br>DAAK30-78-C-0059                       |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Vector Research, Incorporated<br>P.O. Box 1506<br>Ann Arbor, Michigan 48106                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                       | 10. PROGRAM ELEMENT, PROJECT, TASK<br>AREA & WORK UNIT NUMBERS           |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>US Army Tank-Automotive Research and<br>Development Command, Attn: DRDTA-PAV<br>Warren, Michigan 48090                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                       | 12. REPORT DATE<br>August 1979                                           |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                       | 13. NUMBER OF PAGES<br>360                                               |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)<br>US Army Tank-Automotive Research and<br>Development Command, Attn: DRDTA-NK<br>Warren, Michigan 48090                                                                                                                                                                                                                                                                                                                                                                                                                       |                       | 15. SECURITY CLASS. (of this report)<br>Unclassified                     |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                       | 15a. DECLASSIFICATION/DOWNGRADING<br>SCHEDULE<br>N/A                     |
| 16. DISTRIBUTION STATEMENT (of this Report)<br><br>Approved for public release;<br>distribution unlimited.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                       |                                                                          |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                       |                                                                          |
| 18. SUPPLEMENTARY NOTES<br><br>N/A                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                       |                                                                          |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number)<br>Armored Combat Vehicles<br>Tanks<br>System Performance Estimation<br>Parametric Design Models<br>R&D Planning Models                                                                                                                                                                                                                                                                                                                                                                                                 |                       |                                                                          |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number)<br>This report describes a parametric engineering system definition model developed for use in planning tank-automotive research and development. The model can be used for (1) estimating the performance of a conceptual armored combat vehicle consisting of a specified set of components or (2) "sizing" a vehicle to meet a set of performance specifications. Key elements of the model include a structure for a data base to contain descriptions of components which might be incorporated into future armored |                       |                                                                          |

Unclassified

combat vehicles, (2) a solution algorithm which uses a combinatorial approach to search over alternative combinations of components to find one which meets specifications input by the model user, (3) a variety of engineering relationships and "look-up table" functions for estimating system engineering parameters and performance characteristics, and (4) routines which output a description of the concept vehicle generated by the model.