INTERACTIVE HIGH RESOLUTION FORCE-ON-FORCE MODEL

MAIN BATTLE PROCESSOR SOURCE CODE LISTING

SPC 94-282-1A

APRIL 1994

PREPARED FOR:

U.S. ARMY TANK - AUTOMOTIVE COMMAND
ADVANCED SYSTEMS CONCEPTS DIVISION
WARREN, MICHIGAN 48397-5000

CONTRACT NO. DAAE07-92-C-R044

PREPARED BY:

STEVEN M. BUC

SYSTEM PLANNING CORPORATION
1500 WILSON BOULEVARD
ARLINGTON, VIRGINIA 22209

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1994	Technical October 1993-March94

4. TITLE AND SUBTITLE
Interactive High Resolution Force-on-Force Model -- Main Battle Processor Source Code Listing

5. FUNDING NUMBERS
C

6. AUTHOR(S)
Steven M. Buc

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
System Planning Corporation
1500 Wilson Boulevard
Arlington, Virginia 22209

8. PERFORMING ORGANIZATION REPORT NUMBER
SPC 94-282-1A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Tank - Automotive Command
Advanced Systems Concepts Division
Warren, Michigan 48397-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Contract No. DAAE07-92-C-R044

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document provides the source code listing for the main battle processor for a unique personal computer based interactive force-on-force model. The programming language used is FORTRAN, and this code has been configured to run on 386+ compatible personal computers. The code is currently dimensioned to permit the operator to simulate armored warfare with up to 200 direct fire units on each side. Up to 10 different battlefield systems are permitted within these 200 units, with each system employing up to different on-board weapons.

(continued on following page.)

14. SUBJECT TERMS	15. NUMBER OF PAGES
force-on-force model, interactive, personal computer, artillery, vehicles, direct fire, guided weapons, terrain, obscurants, mine fields, graphics	74
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

(Abstract continued)

Ten indirect fire support systems may also be modeled to round out the combined arms team. Each artillery system may employ up to 5 different types of ammunition, to include high explosive, submunition dispensing, and guided munitions. The effects of terrain masking are included through the use of a digitized terrain board. Battlefield obscurants, target detection probabilities, and the effects of moving targets on weapon accuracy are also modeled. Mine fields and other obstacles can also be simulated.

The operator directs unit movements and fire orders. The code operates with a variable time step, which can be set as low as one second. The operator can interact with the code operation every sixty cycles. Battlefield activity is displayed as it occurs on the graphics screen. This software was developed in support of studies and analyses of the sequential mission duty cycle/mission profile of a future main battle tank for the year 2015. However, the software is sufficiently flexible to permit the analysis of tradeoffs involving many weapon systems and subsystems on the modern battlefield.

```
C
C   RUN.FOR          March 22, 1994
C
C   MAIN ROUTINE for:
C                Interactive High-Resolution Force-on-Force
C                              Model
C
C   Written and Developed by:
C
C                    Steven M. Suc
C                  System Planning Corporation
C                  1500 Wilson Boulevard
C                  Arlington, Virginia  22209
C
C   Developed for:
C                The United States Army Tank-Automotive Command
C                Research, Development & Engineering Center
C                  AMSTA-ZED, Norma Tallorin
C                Warren, Michigan  48397-5000
C
C   With special acknowledgements to:
C
C                    Dr. Herbert H. Dobbs
C                    Dobbs Associates
C                    448 W. Maryknoll Road
C                  Rochester Hills, Michigan  48309
C
C   Dr. Dobbs provided unique insights into the mechanical functions
C   of numerious armor vehicle sub-systems modeled with this software.
C
C                    Stephen Griggs
C                  System Planning Corporation
C
C   Mr. Griggs provided invaluable assistance in modeling the finer
C   details of armored warfare and reducing the nature of human behavior
C   in this environment to its basic elements, which is the essence
C   of this simulation.
C
C                    Dave Watson
C                    Department of Mathematics
C                    The University of Western Australia
C                    Nedlands, WA 6009  Australia
C
C   Mr. Watson provided Program ACORD (through Internet), used here as
C   a subroutine to automatically contour raw elevation data, based on
C   Delaunay triangulation, reference: "Davis, J.C., 1986 Statistics and
C   data anlaysis in geology, 2ed., Wiley."  Without this subroutine the
C   simulation of real terrain and its graphical representation would
C   have been greatly complicated.
C
C    INPUT PHASE
C    READ:
C          GRID
C          CONTOURS
C          FORCE STRUCTURE
C          UNIT MOVEMENT
C          WEAPON SYSTEMS
C
C    MAP PHASE
C    PLOT:
C          GRID
C          CONTOURS
C          UNIT LOCATIONS
C    ZOOM FEATURES
C
C    STATUS PHASE
C    DISPLAY:
C          UNIT LOCATIONS, MOVEMENT, LOSSES
C
C    COMMAND PHASE
C    GIVE ORDERS AFFECTING:
C          UNIT MOVEMENT
C          UNIT FIRES
C
C    BATTLE PHASE
C    BY TIME INCREMENT:
C          MOVE UNITS
C          SHOOT UNITS
C
C    OUTPUT PHASE
C    SELECT OUTPUTS:
C          UNIT STATUS
C          LOSSES
C          CHRONOLOGY OF EVENTS
C
C    CHANGE TIME STEP
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
     COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
     & NC(15),NCOL(15),JCON
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
     COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),ISSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)
      COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
      COMMON/FIRE/IBPRIME(200,2),IBSECOND(200,2),IRPRIME(200,2),
     & IRSECOND(200,2),WRNGB(200,2),WRNGR(200,2)
      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
     COMMON/ARTTGT/ARTGTB(200,2),ARTGTR(200,2),ITLB,ITLR

      INTEGER ARTGTB,ARTGTR

      CHARACTER*8 BFNAME,RFNAME
C     CHARACTER*80 LINE
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C     CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
      RJUNK=RRAND()
       ITIME=0
       IDTIME=1
     IP1=0
     ITLB=0
     ITLR=0
C
CC     OPEN DUMP FILES FOR DATA ANALYSIS
CC
       OPEN (10,FILE='BATTLE.DMP')
       WRITE (10,*) 'TIME SEQUENCE OF BATTLEFIELD EVENTS'
       WRITE (10,*)
     OPEN (11,FILE='SHOTS.OUT')
     WRITE (11,*) 'SHOTS.OUT'
     WRITE (11,*) 'APS 1-INTERCEPT 0-MISS -1-N/A 2-NO AMMO 3-NO TIME
     & -3-PREVIOUSLY KILLED'
     WRITE (11,*) 'KILL 1-KILLED 0-MISS -1-PREVIOUSLY KILLED'
     WRITE (11,*) '  TIME  SIDE    #  SYS#   WP#  ENGY   RNG  TGT#
     &TSYS   APS  KILL  HULL  TUR.  AZSHOT'
     OPEN (12,FILE='STATUS.OUT')
     WRITE (12,*) 'STATUS.OUT'
     WRITE (12,*) ' DEF = 1 IN-DEFILADE   KILL = 1 ALIVE'
     WRITE (12,*) '   T  SD   # SYS SPD #WP AM1 AM2 AM3 EGY FMD MMD
     &DEF KILL'
     OPEN (13,FILE='MODES.OUT')
     WRITE (13,*) 'BLUE SYSTEM #1 MODE ACCUMULATED TIME'
     WRITE (13,*) '  TIME    DT    #  MODE   TIME'
     OPEN (14,FILE='OTHERS.OUT')
     WRITE (14,*) 'BLUE #1 SPEED ACCELERATION EM ENERGY AZIMUTH GRADE'
     WRITE (14,*) '  TIME    DT    #  SPD  ACEL  ENGY  AZ   GRD'
     OPEN (15,FILE='MINEKLS.OUT')
     WRITE (15,*) ' MINEFIELD KILLS'
     WRITE (15,*) '  TIME   SIDE   VEH#  SYS#  FIELD#'
     OPEN (16,FILE='ARTYKLS.OUT')
     WRITE (16,*) ' ARTILLERY FIRE KILLS'
     WRITE (16,*) '  TIME TGT SIDE VEH#  SYS# STR SIDE  VEH#  SYS#
     &AMMO#'
20     CALL PHASE (IP)
       IF (IP.EQ.1.AND.IP1.EQ.0) CALL INPUT
       IF (IP.EQ.1) IP1=1
       IF (IP.EQ.2) CALL MAPIT
       IF (IP.EQ.3) CALL STATUS
       IF (IP.EQ.4) CALL COMMAND
     IF (IP.EQ.5) CALL BATTLE2
C      IF (IP.EQ.6) CALL
       IF (IP.EQ.7) CALL TIME
     IF (IP.EQ.0) THEN
      CALL WEND
      GO TO 1000
     ENDIF
       GO TO 20
1000   CONTINUE
     CLOSE (10)
     CLOSE (11)
     CLOSE (12)
     CLOSE (13)
     CLOSE (14)
     CLOSE (15)
     CLOSE (16)
     END

       SUBROUTINE PHASE (IP)
1      WRITE (*,*)
       WRITE (*,*) '1-INPUT PHASE'
C       READ:
CC           GRID
CC           CONTOURS
CC           FORCE STRUCTURE
CC           UNIT MOVEMENT
CC           WEAPON SYSTEMS
C
       WRITE (*,*) '2-MAP PHASE'
CC      PLOT:
CC           GRID
CC           CONTOURS
C            UNIT LOCATIONS
C      ZOOM FEATURES
C
       WRITE (*,*) '3-STATUS PHASE'
C      DISPLAY:
```

```
C          UNIT SPEED (TRUE), LOSSES, AMMO REMAINING
C
      WRITE (*,*) '4-COMMAND PHASE'
C     GIVE ORDERS AFFECTING:
C           UNIT MOVEMENT
C           UNIT FIRES
C           OBSCURANTS
C
      WRITE (*,*) '5-BATTLE PHASE'
C     BY TIME INCREMENT:
C           MOVE UNITS
C           SHOOT UNITS
C
      WRITE (*,*) '6-OUTPUT PHASE'
C     SELECT OUTPUTS:
C           UNIT STATUS
C           LOSSES
C           CHRONOLOGY OF EVENTS
C
      WRITE (*,*) '7-CHANGE TIME STEP'
C
      WRITE (*,*) '0-QUIT'
      WRITE (*,*)
      WRITE (*,*) 'ENTER PHASE SELECTION'
      WRITE (*,*)
      READ (*,*,ERR=1) IP
      RETURN
      END

      SUBROUTINE INPUT
C     INPUT PHASE
C     READ:
C           GRID
C           CONTOURS
C           FORCE STRUCTURE
C           UNIT MOVEMENT
C           WEAPON SYSTEMS
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
     & NC(15),NCOL(15),JCON
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACQB(10,5),ACQR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)
      COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMMR(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
      COMMON/ELEVA/CONT(20),IELEV(10000)
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
      COMMON/OBSCURE/NGRENB(200,2),NGRENR(200,2),IBESMK(200),
     & IRESMK(200)
      COMMON/TRACKING/BOREB(10,3),BORER(10,3),IRNGB(30,4),IRNGR(30,4),
     & ITRKB(10,4),ITRKR(10,4),TOFB(10,3),TOFR(10,3)
      COMMON/PRIOTY/KPRIB(200,10),KPRIR(200,10)
      COMMON/ACEL/ACELB(10,5),ACELR(10,5),GRADEB(200),GRADER(200)
      COMMON/MENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
     & ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2)
           COMMON/FLAGS/ITER
           COMMON/TURRET/AZTB(200),AZTR(200),TRATEB(10),TRATER(10)
           COMMON/COOKERS/NCOOK,XYCOOK(400,2),PBURNB(10),PBURNR(10),
     & KBURNB(200),KBURNR(200)

      CHARACTER*9 BFNAME,RFNAME
      CHARACTER*80 LINE
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*13 FORM
C     CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
1002  WRITE (*,*)
           WRITE (*,*) 'TERRAIN  ON = 1  OFF = 0'
      WRITE (*,*)
      READ (*,*,ERR=1002) ITER

      OPEN (1,FILE='GRID.IN')
      READ (1,*) NX
      READ (1,*) NY
      READ (1,*) SCALE
      READ (1,*) SCFACT
      READ (1,*) PRFACT
      READ (1,*) ZFACT
      CLOSE (1)
      NY=NX
C
      OPEN (1,FILE='COLORS.IN')
311   FORMAT (2I5)
      DO 151 I=1,15
151   READ (1,311) NC(I),NCOL(NC(I))
      CLOSE (1)
C
C     INITIALIZE PLOTTING FACTORS
C
C     XUNIT=11.0/NX*SCFACT
C     YUNIT=8.5/NY
      GXUNIT=11.0/NX*SCFACT
      GYUNIT=8.5/NY
      XUNIT=GXUNIT/SCALE
      YUNIT=GYUNIT/SCALE
211   FORMAT (2F10.4,A5,2F10.4,A5,I5)
      IF (ITER.EQ.1) THEN
      OPEN (1,FILE='CONTSEGM.DAT')
      I=1
101   READ (1,211,ERR=1001) XA,YA,I1,XB,YB,I2,IC(I)
C     WRITE (*,211) XA,YA,I1,XB,YB,I2,IC(I)
C
      X1(I)=XA*XUNIT
      X2(I)=XB*XUNIT
      Y1(I)=YA*YUNIT
      Y2(I)=YB*YUNIT
      I=I+1
      JCON=I-1
      GO TO 101
C
1001  CONTINUE
      CLOSE (1)
C
C     READ CONTOUR AND GRID ELEVATION INFORMATION
C
      OPEN (1,FILE='JDAVIS.DAT')
C.....READ NO. OF DATA POINTS, NO. OF CONTOURS, DATA FORMAT
      READ(1,510) NDATA, LC, FORM
C     WRITE (*,510) NDATA, LC, FORM
510   FORMAT(I8,I5,A13)
C.....READ THE CONTOUR VALUES
      READ(1,FORM) (CONT(I),I=1,LC)
C     WRITE (*,FORM) (CONT(I),I=1,LC)
      CLOSE (1)
C     CONVERT CONTOUR INPUT DATA FROM ELEVATIONS IN FEET
C     TO ELEVATIONS IN KM
      DO 514 I=1,LC
      CONT(I)=CONT(I)*.3048/1000.
514   CONTINUE
      ELSE
      LC=1
      CONT(1)=0.0
      ENDIF
C
      OPEN (1,FILE='GRID.DAT')
511   FORMAT (3I6)
      DO 513 IX=1,NX
      DO 512 IY=1,NY
      I=NY*(IX-1)+IY
      READ (1,511) NUMX,NUMY,IELEV(I)
      IF (ITER.NE.1) IELEV(I)=1
512   CONTINUE
513   CONTINUE
      CLOSE (1)
C
C     READ FORCE STRUCTURE AND MOVEMENT
C
C     BHEIR  BLUE FORCE HEIRARCHY
C     RHEIR  RED  FORCE HEIRARCHY
C     BUNIT  BLUE UNIT NAMES
C     RUNIT  RED  UNIT NAMES
C     IBSYS  BLUE SYSTEM NUMBER AND NUMBER OF SYSTEMS FOR EACH UNIT
C     IRSYS  RED SYSTEM NUMBER AND NUMBER OF SYSTEMS FOR EACH UNIT
C     NB     NUMBER OF BLUE UNITS
C     NR     NUMBER OF RED UNITS
C
C     BXYA   BLUE UNIT ABSOLUTE X AND Y POSITION
C     RXYA   RED UNIT ABSOLUTE X AND Y POSITION
C     XG     X POSITION IN GRID UNITS
C     YG     Y POSITION IN GRID UNITS
C     IBSA   BLUE UNIT SPEED AND AZIMUTH (ROUNDED TO INTEGER)
C     IRSA   RED UNIT SPEED AND AZIMUTH (ROUNDED TO INTEGER)
C
C     COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
C    & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      DO 11 I=1,200
      BHIER(I)=' '
      RHIER(I)=' '
      BUNIT(I)=' '
      RUNIT(I)=' '
      IBSYS(I,1)=0
      IBSYS(I,2)=0
      IRSYS(I,1)=0
      IRSYS(I,2)=0
11    CONTINUE

C     OPEN FILES AND READ CURRENT STATUS
C     READ FORCE STRUCTURES
      OPEN (1,FILE='BFORCE')
      READ (1,3) L1BF
      READ (1,3) L2F
```

```
      READ (1,3) L3F
C     WRITE (*,31) L1BF
C     WRITE (*,31) L2F
C     WRITE (*,31) L3F
2     FORMAT (A16,A24,2I8)
3     FORMAT (A80)
21    FORMAT (1X,A16,A24,2I8)
31    FORMAT (1X,A80)
      I=1
10    READ (1,2,ERR=100) BHIER(I),BUNIT(I),IBSYS(I,1),IBSYS(I,2)
            IF (IBSYS(I,1).EQ.0) GO TO 100
C     WRITE (*,21) BHIER(I),BUNIT(I),IBSYS(I,1),IBSYS(I,2)
      I=I+1
      GO TO 10
100   CONTINUE
      NB=I-1
      CLOSE (1)
C
      OPEN (1,FILE='RFORCE')
      READ (1,3) L1RF
      READ (1,3) L2F
      READ (1,3) L3F
C     WRITE (*,31) L1RF
C     WRITE (*,31) L2F
C     WRITE (*,31) L3F
      I=1
20    READ (1,2,ERR=200) RHIER(I),RUNIT(I),IRSYS(I,1),IRSYS(I,2)
            IF (IRSYS(I,1).EQ.0) GO TO 200
C     WRITE (*,21) RHIER(I),RUNIT(I),IRSYS(I,1),IRSYS(I,2)
      I=I+1
      GO TO 20
200   CONTINUE
      NR=I-1
      CLOSE (1)
C
C     READ MOVEMENT STATUS
      OPEN (1,FILE='BMOVE')
      READ (1,3) L1BM
      READ (1,3) L2M
      READ (1,3) L3M
C     WRITE (*,31) L1BM
C     WRITE (*,31) L2M
C     WRITE (*,31) L3M
12    FORMAT (A16,A24,4F8.3,2I8)
121   FORMAT (1X,A16,A24,4F8.3,2I8)
      DO 110 I=1,NB
      READ (1,12) BHIER(I),BUNIT(I),BXYA(I,1),BXYA(I,2),BXYO(I,1),
     & BXYO(I,2),IBSA(I,1),IBSA(I,2)
C     WRITE (*,121) BHIER(I),BUNIT(I),BXYA(I,1),BXYA(I,2),BXYO(I,1),
C    & BXYO(I,2),IBSA(I,1),IBSA(I,2)
      ISPOS(I)=IBSA(I,1)
      MTYPEB(I)=0
      BXYP(I,1)=0.
      BXYP(I,2)=0.
110   CONTINUE
      CLOSE (1)
C
      OPEN (1,FILE='RMOVE')
      READ (1,3) L1RM
      READ (1,3) L2M
      READ (1,3) L3M
C     WRITE (*,31) L1RM
C     WRITE (*,31) L2M
C     WRITE (*,31) L3M
      DO 115 I=1,NR
      READ (1,12) RHIER(I),RUNIT(I),RXYA(I,1),RXYA(I,2),RXYO(I,1),
     & RXYO(I,2),IRSA(I,1),IRSA(I,2)
C     WRITE (*,121) RHIER(I),RUNIT(I),RXYA(I,1),RXYA(I,2),RXYO(I,1),
C    & RXYO(I,2),IRSA(I,1),IRSA(I,2)
      ISPDR(I)=IRSA(I,1)
      MTYPER(I)=0
      RXYP(I,1)=0.
      RXYP(I,2)=0.
115   CONTINUE
      CLOSE (1)
C
CC
C     READ WEAPON SYSTEMS CHARACTERISTICS
C

C     WEAPONS.FOR
CC
CC    INPUTS WEAPONS CHARACTERISTICS FROM DATA FILES
CC
CC    FILE STRUCTURE IS DEFINED IN FILE "WEAPONS.IN"
CC
CC    NBF    # OF BLUE SYSTEM FILES                    (10 MAX)
CC    NBR    # OF RED SYSTEM FILES                     (10 MAX)
CC    BFNAME  BLUE SYSTEM FILE NAMES                   (10 MAX)
CC    RFNAME  RED SYSTEM FILE NAMES                    (10 MAX)
CC    NWB    # OF BLUE WEAPONS FOR EACH SYSTEM         (3 MAX)
CC    NWR    # OF RED WEAPONS FOR EACH SYSTEM          (3 MAX)
C     NRWB   # OF RANGES FOR EACH BLUE WEAPON PER SYSTEM   (6 MAX)
C     NRWR   # OF RANGES FOR EACH RED WEAPON PER SYSTEM    (6 MAX)
C     RB     BLUE WEAPON RANGES
C     RR     RED WEAPON RANGES
CC    PKB    BLUE Pk FOR EACH RED TARGET (EXPOSED/DEFILADE)  (10 MAX)
CC    PKR    RED Pk FOR EACH BLUE TARGET (EXPOSED/DEFILADE)  (10 MAX)
CC    NRNDB  # SHOTS FOR EACH BLUE WEAPON
CC    NRNDR  # SHOTS FOR EACH RED WEAPON
C     TEMAXB  MAX RANGE ENGAGEMENT TIME FOR EACH BLUE WEAPON
C     TEMAXR  MAX RANGE ENGAGEMENT TIME FOR EACH RED WEAPON
C     TEMINB  MIN RANGE ENGAGEMENT TIME FOR EACH BLUE WEAPON
C     TEMINR  MIN RANGE ENGAGEMENT TIME FOR EACH RED WEAPON
CC    MILB   BLUE WEAPON FIRE-AND-FORGET (0) OR MAN-IN-LOOP (1)
CC    MILR   RED  WEAPON FIRE-AND-FORGET (0) OR MAN-IN-LOOP (1)
CC    BOREB  BLUE BATTLESIGHT RANGES
CC    BORER  RED  BATTLESIGHT RANGES
CC    TOFB   BLUE TIME OF FLIGHT TO BATTLESIGHT
CC    TOFR   RED  TIME OF FLIGHT TO BATTLESIGHT
CC    IRNGB  BLUE WEAPON RANGING/TRACKING THROUGH OBSCURANTS
CC    IRNGR  RED WEAPON RANGING/TRACKING THROUGH OBSCURANTS
CC    ITRKB  BLUE SYSTEM SEEING THROUGH OBSCURANTS
CC    ITRKR  RED SYSTEM SEEING THROUGH OBSCURANTS
CC    SRCHB  SEARCH RATE BLUE SYSTEM
CC    SRCHR  SEARCH RATE RED SYSTEM
CC    ACELB  BLUE SYSTEM ACCELERATION PARAMETERS
CC           1 - MAX SPEED ON FLAT GRADE
CC           2 - TIME FROM ZERO TO MAX SPEED
CC           3 - MAX SPEED ON A GRADE
CC           4 - GRADE FOR THAT MAX SPEED (DEGREES)
CC           5 - TIME FROM ZERO TO THAT MAX SPEED
CC    ACELR  RED SYSTEM ACCELERATION PARAMETERS
CC
C     INITIALIZE ARRAYS
      DO 116 JJ=1,180
      RB(JJ)=0.0
      RR(JJ)=0.0
      DO 117 II=1,20
      PKB(JJ,II)=0.0
      PKR(JJ,II)=0.0
117   CONTINUE
116   CONTINUE
      DO 118 II=1,30
      NRNDB(II)=0
      TEMAXB(II)=0.0
      TEMINB(II)=0.0
      NRNDR(II)=0
      TEMAXR(II)=0.0
      TEMINR(II)=0.0
118   CONTINUE
      DO 119 II=1,10
      DO 120 JJ=1,3
      NWB(II)=0
      NWR(II)=0
120   CONTINUE
      NRWB(II,JJ)=0
      NRWR(II,JJ)=0
      MILB(II,JJ)=-1
      MILR(II,JJ)=-1
119   CONTINUE

C
301   FORMAT (I2)
302   FORMAT (A9)
303   FORMAT (A80)
501   FORMAT (1X,I2)
502   FORMAT (1X,A9)
503   FORMAT (1X,A80)
      OPEN (1,FILE='WEAPONS.IN')
      READ (1,301) NBF
C     WRITE (*,501) NBF
      DO 310 I=1,NBF
      READ (1,302) BFNAME(I)
C     WRITE (*,502) BFNAME(I)
310   CONTINUE
      READ (1,301) NRF
C     WRITE (*,501) NRF
      DO 312 I=1,NRF
      READ (1,302) RFNAME(I)
C     WRITE (*,502) RFNAME(I)
312   CONTINUE
      CLOSE (1)
C
C     READ BLUE FILES
C
      DO 400 I=1,NBF
      OPEN (1,FILE=BFNAME(I))
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
320   FORMAT (I8,3I8)
520   FORMAT (1X,I8,3I8)
      READ (1,320) NWB(I),(NRWB(I,J),J=1,3)
C     WRITE (*,520) NWB(I),(NRWB(I,J),J=1,3)
      DO 350 J=1,NWB(I)
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
321   FORMAT (F8.0,20F8.6)
```

```
521   FORMAT (1X,F7.0,1X,20(F6.3,3X))
      DO 340 K=1,NRWB(I,J)
      ISR=(I-1)*18+(J-1)*6+K
C      WRITE (*,*) 'ISR',ISR
C      WHERE I=SYSTEM FILE (10 MAX); J=SYSTEM WEAPON (3 MAX);
C           K=WEAPON RANGE (6 MAX); N=TARGET (2*10 MAX)
      READ (1,321) RB(ISR),(PKB(ISR,IST),IST=1,20)
C      WRITE (*,321) RB(ISR),(PKB(ISR,IST),IST=1,20)
C      WRITE (*,*) 'RB(22)',RB(22)
340   CONTINUE
360   CONTINUE
C      WRITE (*,*)
C      WRITE (*,*) (RB(ISR),ISR=1,36)
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
322   FORMAT (I8,I8,2F8.2,I8,2F8.2)
522   FORMAT (1X,I8,I8,2X,2(F5.2,3X),1X,I8,1X,2F8.2)
      DO 380 L=1,NWB(I)
      ISROF=(I-1)*3+L
C      WHERE I=SYSTEM FILE (10 MAX); L=SYSTEM WEAPON (3 MAX);
C           READ (1,322) M,NRNDB(ISROF),TEMINB(ISROF),TEMAXB(ISROF),
     &  MILB(I,L),BOREB(I,L),TOFB(I,L)
C      WRITE (*,522) M,NRNDB(ISROF),TEMINB(ISROF),TEMAXB(ISROF),
C     &  MILB(I,L),BOREB(I,L),TOFB(I,L)
380   CONTINUE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
381   FORMAT (5I8)
382   FORMAT (1X,5I8)
      DO 361 L=1,NWB(I)
      ISROF=(I-1)*3+L
C      WHERE I=SYSTEM FILE (10 MAX); L=SYSTEM WEAPON (3 MAX);
C      J=CONCEALMENT TYPE (4 MAX)
      READ (1,381) IL,(IRNGB(I,J),J=1,4)
C      WRITE (*,382) IL,(IRNGB(I,J),J=1,4)
361   CONTINUE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
362   FORMAT (4I8)
363   FORMAT (1X,4I8)
      READ (1,362) (ITRKB(I,L),L=1,4)
C      WRITE (*,363) (ITRKB(I,L),L=1,4)
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
364   FORMAT (2I8)
365   FORMAT (1X,2I8)
           READ (1,364) SRCHB(I),TRATEB(I)
C      WRITE (*,365) SRCHB(I),TRATEB(I)
C
C      ACCELERATION PARAMETERS
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,461) (ACELB(I,L),L=1,5)
C      WRITE (*,462) (ACELB(I,L),L=1,5)
      CLOSE (1)

400   CONTINUE
C
C      READ RED FILES
C
      DO 500 I=1,NRF
      OPEN (1,FILE=RFNAME(I))
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,320) NWR(I),(NRWR(I,J),J=1,3)
C      WRITE (*,520) NWR(I),(NRWR(I,J),J=1,3)
      DO 450 J=1,NWR(I)
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      DO 440 K=1,NRWR(I,J)
      ISR=(I-1)*18+(J-1)*6+K
C      WHERE I=SYSTEM FILE (10 MAX); J=SYSTEM WEAPON (3 MAX);
C           K=WEAPON RANGE (6 MAX); N=TARGET (2*10 MAX)
      READ (1,321) RR(ISR),(PKR(ISR,IST),IST=1,20)
C      WRITE (*,521) RR(ISR),(PKR(ISR,IST),IST=1,20)
440   CONTINUE
450   CONTINUE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      DO 460 L=1,NWR(I)
      ISROF=(I-1)*3+L
C      WHERE I=SYSTEM FILE (10 MAX); L=SYSTEM WEAPON (3 MAX);
      READ (1,322) M,NRNDR(ISROF),TEMINR(ISROF),TEMAXR(ISROF),
     &  MILR(I,L),BORER(I,L),TOFR(I,L)
C      WRITE (*,522) M,NRNDR(ISROF),TEMINR(ISROF),TEMAXR(ISROF),
C     &  MILR(I,L),BORER(I,L),TOFR(I,L)
460   CONTINUE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      DO 451 L=1,NWR(I)
      ISROF=(I-1)*3+L
C      WHERE I=SYSTEM FILE (10 MAX); L=SYSTEM WEAPON (3 MAX);
C      J=CONCEALMENT TYPE (4 MAX)
      READ (1,381) IL,(IRNGR(I,J),J=1,4)
C      WRITE (*,382) IL,(IRNGR(I,J),J=1,4)
451   CONTINUE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,352) (ITRKR(I,L),L=1,4)
C      WRITE (*,353) (ITRKR(I,L),L=1,4)
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,364) SRCHR(I),TRATER(I)
C      WRITE (*,365) SRCHR(I),TRATER(I)
C
C      ACCELERATION PARAMETERS
461   FORMAT (5F8.1)
462   FORMAT (1X,5F8.1)
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,303) LINE
C      WRITE (*,503) LINE
      READ (1,461) (ACELR(I,L),L=1,5)
C      WRITE (*,462) (ACELR(I,L),L=1,5)
      CLOSE (1)
500   CONTINUE

C      READ ELECTROMAGNETIC ENERGY PARAMETERS
      OPEN (1,FILE='EMGUN.IN')
      READ (1,3) L1BF
C      WRITE (*,31) L1BF
      READ (1,3) L2F
C      WRITE (*,31) L2F
531   FORMAT (I8)
532   FORMAT (1X,I8)
      READ (1,531) NBLUE
C      WRITE (*,532) NBLUE
      READ (1,3) L1BF
C      WRITE (*,31) L1BF
      READ (1,531) NEMW
C      WRITE (*,532) NEMW
      READ (1,3) L1BF
C      WRITE (*,31) L1BF
533   FORMAT (2I8)
504   FORMAT (1X,2I8)
      READ (1,533) (NEMWN(IEM),IEM=1,NEMW)
C      WRITE (*,504) (NEMWN(IEM),IEM=1,NEMW)
      READ (1,3) L1BF
C      WRITE (*,31) L1BF
      READ (1,3) L2F
C      WRITE (*,31) L2F
505   FORMAT (2F8.3)
506   FORMAT (1X,2F8.3)
      DO 507 IEM=1,NEMW
      READ (1,505) (EWP(IEM,JEM),JEM=1,2)
C      WRITE (*,506) (EWP(IEM,JEM),JEM=1,2)
```

```
507   CONTINUE
      READ (1,3) L1BF
C     WRITE (*,31) L1BF
      READ (1,3) L1BF
C     WRITE (*,31) L1BF
      READ (1,3) L1BF
C     WRITE (*,31) L1BF
508   FORMAT (5F8.3)
509   FORMAT (1X,5F8.3)
      READ (1,508) (EG(IEM),IEM=1,5)
C     WRITE (*,509) (EG(IEM),IEM=1,5)

      CLOSE (1)
C

C     INITIALIZE BLUE EM WEAPON ENERGIES
C     AND NUMBER OF SHOTS FOR EACH BLUE AND RED WEAPON
C
C     IBSYS  BLUE SYSTEM NUMBER AND NUMBER OF SYSTEMS FOR EACH UNIT
C     IRSYS  RED SYSTEM NUMBER AND NUMBER OF SYSTEMS FOR EACH UNIT
C     NB     NUMBER OF BLUE UNITS
C     NR     NUMBER OF RED UNITS
C     NRNDB  # SHOTS FOR EACH BLUE WEAPON
C     NRNDR  # SHOTS FOR EACH RED WEAPON
C     NWB    # OF BLUE WEAPONS FOR EACH SYSTEM          (3 MAX)
C     NWR    # OF RED WEAPONS FOR EACH SYSTEM           (3 MAX)
C    & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
C     ISROF=(I-1)*3+J
C
      DO 600 I=1,NB
      NSYS=IBSYS(I,1)

      IF (NSYS.EQ.NBLUE) THEN
        ELSYS(I)=EG(1)
        TE(I)=0.0
        IF (ELSYS(I).LE.EG(3)) THEN
          MODEEM(I)=2
C         START CHARGING SYSTEM
        ELSE
          MODEEM(I)=1
C         SYSTEM IS IDLE
        ENDIF
      ENDIF
      NWEAP=NWB(NSYS)
      IAMB(I)=0
C     SMOKE GRENADES (2 VOLLEYS MAX)
      NGRENB(I,1)=2
      NGRENB(I,2)=0
      IBESMK(I)=0
      DO 580 J=1,NWEAP
      ISROF=(NSYS-1)*3+J
      NSHOTB(I,J)=NRNDB(ISROF)
580   CONTINUE
600   CONTINUE
C
      DO 700 I=1,NR
      NSYS=IRSYS(I,1)
      NWEAP=NWR(NSYS)
      IAMR(I)=0
C     SMOKE GRENADES (2 VOLLEYS MAX)
      NGRENR(I,1)=2
      NGRENR(I,2)=0
      IRESMK(I)=0
      DO 680 J=1,NWEAP
      ISROF=(NSYS-1)*3+J
      NSHOTR(I,J)=NRNDR(ISROF)
680   CONTINUE
700   CONTINUE
C
C     INITIALIZE MOVEMENT AND FIRE MODES
C
C     MANEUVER AND FIRE MODES.
C     MODEL TRACKS THE CURRENT MODE EACH SYSTEM IS IN,
C     THE TIME WHEN THE SYSTEM CAN RE-MODE,
C     AND THE ACCUMULATED TIME IN EACH MODE DURING EACH BATTLE CYCLE
C
C     THIS MODE INFORMATION IS DUMPED AFTER EACH BATTLE CYCLE
C     AND TOTAL ACCUMULATED TIME IS DUMPED AT END OF GAME
C
C     MODEMB  CURRENT MANEUVER MODE FOR EACH BLUE SYSTEM (3 MODES)
C           MODE 1 -- DRIVING
C           MODE 2 -- ACCELERATING
C           MODE 3 -- STOPPED
C     MODEMR  CURRENT MANEUVER MODE FOR EACH RED SYSTEM
C     TCMMB   CYCLE TIME IN EACH MANEUVER MODE FOR BLUE SYSTEM
C     TCMMR   CYCLE TIME IN EACH MANEUVER MODE FOR RED SYSTEM
C     TTMMB   TOTAL TIME IN EACH MANEUVER MODE FOR BLUE SYSTEM
C     TTMMR   TOTAL TIME IN EACH MANEUVER MODE FOR RED SYSTEM
C     MODEFB  CURRENT FIRE MODE FOR EACH BLUE SYSTEM (9 MODES)
C           MODE 1 -- SEARCHING FOR TARGETS
C           MODE 2 -- ENGAGING A TARGET
C           MODE 3 -- INSUFFICIENT EM ENERGY (HOLD FIRE)
C           MODE 4 -- CREW/GUNNER BALKS AND HOLDS FIRE
C           MODE 5 -- (HOLD FIRE) NO TARGETS WITHIN EFFECTIVE RANGE
C           MODE 6 -- MISSED TARGET RE-ENGAGE
C           MODE 7 -- SURVEILLANCE
C           MODE 8 -- RE-DESIGNATE MAN-IN-LOOP WEAPON
C           MODE 9 -- SACRIFICE SHOT (FIRE-AND-FORGET WEAPON)
C     MODEFR  CURRENT FIRE MODE FOR EACH RED SYSTEM
C     KEWB    CURRENT ENGAGEMENT WEAPON ON BLUE SYSTEM
C     KEWR    CURRENT ENGAGEMENT WEAPON ON RED SYSTEM
C     TCFMB   CYCLE TIME IN EACH FIRE MODE FOR BLUE SYSTEM
C     TCFMR   CYCLE TIME IN EACH FIRE MODE FOR RED SYSTEM
C     TTFMB   TOTAL TIME IN EACH FIRE MODE FOR BLUE SYSTEM
C     TTFMR   TOTAL TIME IN EACH FIRE MODE FOR RED SYSTEM
C     TRMMB   TIME AT WHICH BLUE SYSTEM CAN RE-MODE MANEUVER
C     TRMFB   TIME AT WHICH BLUE SYSTEM CAN RE-MODE FIRE
C     TRMMR   TIME AT WHICH RED SYSTEM CAN RE-MODE MANEUVER
C     TRMFR   TIME AT WHICH RED SYSTEM CAN RE-MODE FIRE
C
C
C     COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
C    & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
C    & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
C    & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
C    & TRMFR(200),TRDESB(200),TRDESR(200)
C
      DO 800 I=1,NB
      MODEMB(I)=1
        IF (IBSA(I,1).EQ.0) MODEMB(I)=3
      TTMMB(I,1)=0.0
      TTMMB(I,2)=0.0
      TTMMB(I,3)=0.0
      MODEFB(I)=7
      AZTB(I)=IBSA(I,2)
      TTFMB(I,1)=0.0
      TTFMB(I,2)=0.0
      TTFMB(I,3)=0.0
      TTFMB(I,4)=0.0
      TTFMB(I,5)=0.0
      TTFMB(I,6)=0.0
      TTFMB(I,7)=0.0
      TRMMB(I)=0.0
      TRMFB(I)=0.0
C     TOTAL TIMES FOR EACH CYCLE ARE INITIALIZED GOING INTO
C     BATTLE PHASE
C
C     INITIALIZE TERRAIN GRADE FOR EACH VEHICLE FOR ACCELERATION
C     DETERMINATION
      KSIDE=1
      CALL GRADE (KSIDE,I)

C     INITIALIZE ACTUAL VEHICLE SPEED
      ISPDB(I)=IBSA(I,1)
C
800   CONTINUE
C
      DO 850 I=1,NR
      MODEMR(I)=1
        IF (IRSA(I,1).EQ.0) MODEMR(I)=3
      TTMMR(I,1)=0.0
      TTMMR(I,2)=0.0
      TTMMR(I,3)=0.0
      MODEFR(I)=7
      AZTR(I)=IRSA(I,2)
      TTFMR(I,1)=0.0
      TTFMR(I,2)=0.0
      TTFMR(I,3)=0.0
      TTFMR(I,4)=0.0
      TTFMR(I,5)=0.0
      TTFMR(I,6)=0.0
      TTFMR(I,7)=0.0
      TRMMR(I)=0.0
      TRMFR(I)=0.0
C     INITIALIZE TERRAIN GRADE FOR EACH VEHICLE FOR ACCELERATION
C     DETERMINATION
      KSIDE=2
      CALL GRADE (KSIDE,I)

C     INITIALIZE ACTUAL VEHICLE SPEED
      ISPDR(I)=IRSA(I,1)
C
850   CONTINUE
C
C      WRITE (*,*) '1'
C      WRITE (*,*) (RB(ISR),ISR=1,36)
C      WRITE (*,*) (RR(ISR),ISR=1,36)
C
C     WIND CONDITIONS
C
855   WRITE (*,*)
      WRITE (*,*) 'WIND CONDITIONS'
      WRITE (*,*) '1 -- RANDOM WIND -- 15 KNOTS MAX, ANY DIRECTION'
      WRITE (*,*) '2 -- ENTER WIND CONDITIONS'
      WRITE (*,*)
      READ (*,*,ERR=855) IWIND
       IF (IWIND.LT.1.OR.IWIND.GT.2) GO TO 855
       IF (IWIND.EQ.1) THEN
       WRAND=RND()
       IWSPEED=NINT(15.0*WRAND)
       DRAND=RND()
       IWDIR=NINT(DRAND*360.0)
        WRITE (*,*)
        WRITE (*,*) 'RANDOM WIND'
        WRITE (*,*) IWSPEED,' KNOTS going towards AZIMUTH ',IWDIR
        WRITE (*,*)
       ELSE
        WRITE (*,*)
        WRITE (*,*) 'ENTER WIND SPEED (KNOTS max 15) and AZIMUTH'
        WRITE (*,*)
       READ (*,*,ERR=855) WRAND,DRAND
        IF (WRAND.GT.15.) WRAND=15.
       IF (DRAND.LT.0.0) DRAND=DRAND+360.
       IF (DRAND.GT.360.0) DRAND=DRAND-360.0
```

```
      IWSPEED=INT(WRAND)
      IWDIR=INT(ORAND)
      WRITE (*,*)
      WRITE (*,*) IWSPEED,' KNOTS going towards AZIMUTH ',IWDIR
      WRITE (*,*)
      ENDIF
C
C     DUST CONDITIONS
C
885   WRITE (*,*)
      WRITE (*,*) 'DUST CONDITIONS'
      WRITE (*,*) '1 -- YES -- PLAY DUST'
      WRITE (*,*) '2 -- NO  -- DO NOT PLAY DUST'
      WRITE (*,*)
      READ (*,*,ERR=885) IDUST
      IF (IDUST.LT.1.OR.IDUST.GT.2) GO TO 885
C
C     INITIALIZE SMOKE AND DUST ARRAYS
C
C     DUST AND ENGINE SMOKE CLOUDS ARE ACCOUNTED FOR AS DESCRETE
C     PUFFS.  GRENADE AND ARTILLERY SMOKE ARE MODELED AS CURTAINS.
C     PUFFS ARE HEMISPHERICAL FOR PURPOSES OF EXPANSION AND WIND DRIFT.
C     CURTAINS DO NOT EXPAND, BUT DO DRIFT ON THE WIND.
C     PUFFS CAN BE GENERATED IN 10 SECOND INCREMENTS.  PUFFS AND
C     CURTRAINS ARE MOVED ON THE WIND ONLY EVERY 10 SECONDS.
C     PUFFS AND CURTAINS DISSIPATE EVERY 60 SECONDS.  PUFFS EXPAND
C     AT A RATE OF 5 FEET PER SECOND.
      EXPUFF=5.0*.3048/1000.
      DTPUFF=10.0
      TENDPUFF=60.0
C
C     DUST ARRAYS
C     ACTIVE CLOUDS:  BEGINNING NUMBER = 1 ; ENDING NUMBER = 1
C     PTDUST (2400,3) -- POSITION X,Y AND TIME FOR DUST PUFFS
      IDUST1=0
      IDUST2=0
      NDUST=0
      NDUSTMX=2400
C     ENGINE SMOKE ARRAYS
C     ACTIVE CLOUDS:  BEGINNING NUMBER = 1 ; ENDING NUMBER = 1
C     PTENG (2400,3)  -- POSITION X,Y AND TIME FOR ENGINE SMOKE PUFFS
      IENG1=0
      IENG2=0
      NENG=0
      NENGMX=2400
C     SMOKE GRENADE CURTAINS
C     ACTIVE CLOUDS:  BEGINNING NUMBER = 1 ; ENDING NUMBER = 1
C     PSMK (400,2)  -- POSITION X,Y FOR SMOKE GRENADE CURTAIN
C     TASMK (400,2) -- TIME AND AZIMUTH FOR SMOKE GRENADE CURTAIN
      ISMK1=0
      ISMK2=0
      NSMK=0
      NSMKMX=400
C     ARTILLERY SMOKE CURTAINS
C     ACTIVE CLOUDS:  BEGINNING NUMBER = 1 ; ENDING NUMBER = 1
C     PART (99,2)  -- POSTION X,Y FOR ARTILLERY SMOKE CURTAIN
C     TAART (99,2) -- TIME AND RADIUS  FOR ARTILLERY SMOKE CURTAIN
      IART1=0
      IART2=0
      NART=0
            NARTMX=99
C
C     INITIALIZE TARGET PRIORITIZATION
C
      DO 901 KI=1,200
      DO 902 KJ=1,10
      KPRIB(KI,KJ)=1
      KPRIR(KI,KJ)=1
902   CONTINUE
901   CONTINUE

C     READ TARGET DETECTION SIGNATURE DATA
      CALL RSIGNAT
C     READ ACTIVE PROTECTION SYSTEM DATA
      CALL READAPS
C     READ PROBABILITY OF VEHICLE BURNING DATA AND INITIALIZE ARRAYS
            CALL READCOOK
C     READ MINEFIELD DATA
            CALL READMINES
C     READ ARTILLERY SYSTEM DATA
            CALL READARTY
C     INITIALIZE ARTILLERY AVAILABILITY
            CALL ARTINIT

            RETURN
      END



      SUBROUTINE MAPIT
C
C     MAP.FOR
C     PLOTS GRID
C     THEN OVERLAYS CONTOURS
C

      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
     COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
     & NC(15),NCOL(15),JCON
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
     COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
     COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
     COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACQB(10,5),ACQR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)
      COMMON/FLAGS/ITER

      CHARACTER*8 BFNAME,RFNAME
C     CHARACTER*80 LINE
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C     CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
      CHARACTER*1 WELL

C
120   CALL ZOOMIN (WELL)
      IF (WELL.EQ.'O') GO TO 200
121   WRITE (*,*)
      WRITE (*,*) 'ENTER MAPPING OPTIONS'
      WRITE (*,*) ' GRID,CONTOURS,UNITS,OBSCURANTS (1=YES,0-NO)'
      WRITE (*,*)
      READ (*,*,ERR=121) IGRID,ICONT,IUNIT,IOBSC
      CALL PLOTS (0,1,0)
     IF (ICONT.EQ.1.AND.ITER.EQ.1) CALL CONTPLOT (JCON)
     IF (IGRID.EQ.1) CALL GRID2
     IF (IUNIT.EQ.1) CALL PLOTMINES
     IF (IUNIT.EQ.1) CALL UNITS
      IF (IOBSC.EQ.1) THEN
      CALL PDUST
C     PLOT ENGINE SMOKE CLOUDS
      CALL PENG
C     PLOT SMOKE GRENADE CURTAINS
     CALL PSMKCURTAINS
C     PLOT ARTILLERY SMOKE CURTAINS
     CALL PARTSMK
      ENDIF
      READ (*,*)
      CALL PLOT (0.,0.,999)
      GO TO 120
200   RETURN
      END

      SUBROUTINE GRID2
C     GRID.FOR
C
C     GRIDS MAP WITH SQUARES
C     OPENS GRID.DAT FILE FOR ELEVATION DATA AT EACH GRID CENTER
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
C
C     NX          NUMBER OF X GRIDS
C     NY          NUMBER OF Y GRIDS (SHOULD EQUAL NX)
C     SCALE       REAL WORLD SCALE FACTOR FOR EACH GRID
C     SCFACT      SCREEN SCALING FACTOR FOR X DIRECTION
C     PRFACT      PRINTER SCALING FACTOR FOR X DIRECTION
C     ZFACT       ZOOM FACTOR FOR SIZE OF GRID ON SCREEN
C
C
      ISKIP=0
C
C     DRAW A SQUARE FOR EACH GRID LOCATION
C
C     CALL PLOTS (0,1,0)
      XUNIT=11.0/NX*SCFACT
      YUNIT=8.5/NY
      CALL FACTOR (ZFACT)
      IF (XLEN.GT.0.) CALL FACTOR (ZFACTN)
      DO 100 IX=1,NX
      X=IX*XUNIT-XUNIT/2.
      X0=X
      DO 50 IY=1,NY
      X=X0
      CALL NEWPEN (7)
      Y=IY*YUNIT-YUNIT/2.
C     MOVE TO GRID SQUARE CENTER BOTTOM LEFT CORNER
      IF (XLEN.GT.0.) CALL ZOOMIT (X,Y,ISKIP)
      IF (ISKIP.EQ.1) GO TO 50
      CALL PLOT (X,Y,3)
C     DRAW FOUR SIDES OF SQUARE
      CALL PLOT (0.,YUNIT,12)
      CALL PLOT (XUNIT,0.,12)
      CALL PLOT (0.,-YUNIT,12)
      CALL PLOT (-XUNIT,0.,12)
50    CONTINUE
100   CONTINUE
C     READ (*,*)
      RETURN
      END


      SUBROUTINE CONTPLOT (K)
C     CONTPLOT.FOR
C
C     PLOTS CONTOUR MAP AS GENERATED BY SOFTWARE
C     OPENS CONTSEGM.DAT FILE FOR CONTOUR SEGMENTS
C
```

```
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
     & NC(15),NCOL(15),JCON
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
C
C     NX          NUMBER OF X GRIDS
C     NY          NUMBER OF Y GRIDS (SHOULD EQUAL NX)
C     SCALE       REAL WORLD SCALE FACTOR FOR EACH GRID
C     SCFACT      SCREEN SCALING FACTOR FOR X DIRECTION
C     PRFACT      PRINTER SCALING FACTOR FOR X DIRECTION
C     ZFACT       ZOOM FACTOR FOR SIZE OF GRID ON SCREEN
C
      ISKIP=0
      CALL NEWPEN (7)
      CALL FACTOR (ZFACT)
      IF (XLEN.GT.0.) CALL FACTOR (ZFACTN)
C
      J=K-1
      DO 200 I=1,J
      CALL NEWPEN (NCOL(IC(I)))
      X=X1(I)
      Y=Y1(I)
      IF (XLEN.GT.0.) CALL ZOOMIT (X,Y,ISKIP)
      IF (ISKIP.EQ.1) GO TO 200
      CALL PLOT (X,Y,3)
      X=X2(I)
      Y=Y2(I)
      IF (XLEN.GT.0.) CALL ZOOMIT (X,Y,ISKIP)
      IF (ISKIP.EQ.1) GO TO 200
      CALL PLOT (X,Y,2)
200   CONTINUE
C     READ (*,*)
C     CALL PLOT (0.,0.,999)

      RETURN

      END


      SUBROUTINE ZOOMIN (WELL)
C
C     ZOOM WINDOW INPUTS
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
C
C     BLX    BOTTOM LEFT X COORDINATE
C     BLY    BOTTOM LEFT Y COOFDINATE
C     XLEN   ZOOM WINDOW SIDE LENGTH
C     ZFACTN  NEW ZOOM FACTOR TO INCREASE SCALE
C
      CHARACTER*1 WELL
      XUNIT=11.0/NX*SCFACT
      YUNIT=8.5/NY

1     WRITE (*,*) 'ZOOM OPTION ? (enter Y or N) (Q quits this phase)'
      WRITE (*,*)
      READ (*,*,ERR=1) WELL
      IF (WELL.EQ.'Y') THEN
2     WRITE (*,*) 'ENTER GRID NX,NY OF WINDOW BOTTOM LEFT CORNER'
      WRITE (*,*)
      READ (*,*,ERR=2) BLX,BLY
      BLX=BLX*XUNIT
      BLY=BLY*YUNIT
3     WRITE (*,*) 'ENTER WINDOW EDGE LENGTH IN GRID UNITS'
      WRITE (*,*)
      READ (*,*,ERR=3) XLEN
      IF (XLEN.LT.4.1) XLEN=4.1
C     CALCULATE NEW ZOOM FACTOR
      XLEN1=NX*XUNIT
      ZFACTN=XLEN1/XLEN/XUNIT*ZFACT
C       ZFACTN=ZFACT
      XMAX=XLEN*XUNIT
      YMAX=XLEN*YUNIT
      ELSE
      BLX=-1.
      BLY=-1.
      XLEN=-1.
      ENDIF
      RETURN
      END


      SUBROUTINE ZOOMIT (X,Y,ISKIP)
C
C     EXERCISE COORDINATE ZOOM
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
C
C     BLX    BOTTOM LEFT X COORDINATE
C     BLY    BOTTOM LEFT Y COORDINATE
C     XLEN   ZOOM WINDOW SIDE LENGTH

      ISKIP=0
      X=X-BLX
      Y=Y-BLY
      IF (X.LT.0.0) ISKIP=1
      IF (Y.LT.0.0) ISKIP=1
      IF (X.GT.XMAX) ISKIP=1
      IF (Y.GT.YMAX) ISKIP=1
      RETURN
      END


      SUBROUTINE UNITS
C
C     PLOTS UNIT LOCATIONS (VEHICLES) ON MAP
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/COOKERS/NCOOK,XYCOOK(400,2),PBURNB(10),PBURNR(10),
     & KBURNB(200),KBURNR(200)
      CHARACTER*16 BHIER,RHIER,HHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*1 BLANK
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      DIMENSION XARRAY(4),YARRAY(4)
C
C
      ISKIP=0
C
C     DRAWS AND FILLS A SQUARE FOR EACH UNIT LOCATION
C     SIDE LENGTH IS 5 METERS SINCE LARGEST UNIT IS A SQUAD VEHICLE
C     WHEN AT MAX ZOOM
C     AT MIN ZOOM, BOX IS DRAWN LARGER SO YOU CAN SEE IT BETTER
C
C     CALL PLOTS (0,1,0)
      CALL FACTOR (ZFACT)
C       XUNIT=11.0/NX*SCFACT
C       YUNIT=8.5/NY
      GXUNIT=11.0/NX*SCFACT
      GYUNIT=8.5/NY
      XUNIT=GXUNIT/SCALE
      YUNIT=GYUNIT/SCALE
      XL=XUNIT*.005/SCALE**2.
      YL=YUNIT*.005/SCALE**2.
C     XL=XUNIT*.150/SCALE
C     YL=YUNIT*.150/SCALE
      IF (XLEN.GT.0.) THEN
            CALL FACTOR (ZFACTN)
C             XL=XUNIT*.150/SCALE*ZFACT/ZFACTN
C             YL=YUNIT*.150/SCALE*ZFACT/ZFACTN
      ENDIF
      DO 200 N=1,2
      DO 100 I=1,200
      ISKIP=0
      IF (N.EQ.1) THEN
            IHIER=BHIER(I)
            BLANK=IHIER(1:1)
      ELSE
            IHIER=RHIER(I)
            BLANK=IHIER(1:1)
      ENDIF
      IF (BLANK.EQ.'B') GO TO 50
      IF (BLANK.EQ.'R') GO TO 50
      GO TO 100
50    IF (N.EQ.1) THEN
            X=BXYA(I,1)
            Y=BXYA(I,2)
            CALL NEWPEN (9)
                  IF (IBSYS(I,2).EQ.0) CALL NEWPEN(7)
                  IF (KBURNB(I).EQ.1) CALL NEWPEN(14)
      ELSE
            X=RXYA(I,1)
            Y=RXYA(I,2)
            CALL NEWPEN (12)
                  IF (IRSYS(I,2).EQ.0) CALL NEWPEN(7)
                  IF (KBURNR(I).EQ.1) CALL NEWPEN(14)
      ENDIF
      X=X*XUNIT
      Y=Y*YUNIT
      IF (X.LE.0.0) ISKIP=1
      IF (Y.LE.0.0) ISKIP=1
      IF (XLEN.GT.0.) THEN
            CALL ZOOMIT (X,Y,ISKIP)
                  IF (Y.GT.(.95*YMAX)) ISKIP=1
                  IF (X.LT.(.05*XMAX)) ISKIP=1
      ENDIF
      IF (ISKIP.EQ.1) GO TO 100
C     DRAW AND FILL FOUR SIDED SQUARE
      XARRAY(1)=X+XL/2.
      XARRAY(2)=X+XL/2.
      XARRAY(3)=X-XL/2.
      XARRAY(4)=X-XL/2.
      YARRAY(1)=Y-YL/2.
      YARRAY(2)=Y+YL/2.
      YARRAY(3)=Y+YL/2.
      YARRAY(4)=Y-YL/2.
      CALL FILL (4,XARRAY,YARRAY)
100   CONTINUE
200   CONTINUE
C     READ (*,*)
      RETURN
      END

      SUBROUTINE MOVEORS
C     ORDERS.FOR
C
```

```
C     ENTER MOVEMENT AND FIRE CONTROL ORDERS FOR UNITS
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
C     COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
C    & NC(15),NCOL(15),JCON
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),PKB(180),PKR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),TEMINR(30),
     & TEMINF(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPOB(200),ISPOR(200)
      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMMR(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
            COMMON/DEFILADE/IDEFB(200),IDEFR(200)
            COMMON/TURRET/AZTB(200),AZTR(200),TRATEB(10),TRATER(10)
            COMMON/MINEBREECH/MBREECH(200,2),MINEON(100)
            COMMON/AAVAIL/IBAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)
C
      CHARACTER*9 BFNAME,RFNAME
C     CHARACTER*80 LINE
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*1 C2
C     CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
C     MOVEMENT ORDERS
C
      PI=3.14159
      KSIDE=0
10     WRITE (*,*)
      WRITE (*,*) 'enter BLUE or RED, or QUIT'
       WRITE (*,*)
      READ (*,*,ERR=10) C2
       IF (C2(1:1).EQ.'B') KSIDE=1
       IF (C2(1:1).EQ.'R') KSIDE=2
       IF (C2(1:1).EQ.'Q') GO TO 1000
       IF (KSIDE.EQ.0) GO TO 10
C
5      WRITE (*,*)
      WRITE (*,*) 'CURRENT STATUS'
11     FORMAT (1X,A80)
       IF (C2(1:1).EQ.'B') WRITE (*,11) L1BM
       IF (C2(1:1).EQ.'R') WRITE (*,11) L1RM
       WRITE (*,*) ' UNIT HIERARCHY      IDENTITY              POSITION
     &     OBJECTIVE    SPD   AZM'
       IF (C2(1:1).EQ.'B') THEN
C       BLUE
        KK=0
        DO 130 K=1,NB
121      FORMAT (1X,I3,1X,A16,A24,4(F6.3,1X),I3,1X,I3)
            IF (IBSYS(K,2).GE.1) THEN
         WRITE (*,121) K,BHIER(K),BUNIT(K),BXYA(K,1),BXYA(K,2),BXYO(K,1),
     &  BXYO(K,2),IBSA(K,1),IBSA(K,2)
         KK=KK+1
         ENDIF
         IF (KK.EQ.20) THEN
              READ (*,*)
              KK=0
         ENDIF
130      CONTINUE
C
       ELSE
C       RED
        KK=0
        DO 140 K=1,NR
            IF (IRSYS(K,2).GE.1) THEN
         WRITE (*,121) K,RHIER(K),RUNIT(K),RXYA(K,1),RXYA(K,2),RXYO(K,1),
     &  RXYO(K,2),IRSA(K,1),IRSA(K,2)
         KK=KK+1
         ENDIF
         IF (KK.EQ.20) THEN
              READ (*,*)
              KK=0
         ENDIF
140      CONTINUE
       ENDIF

141     WRITE (*,*) 'ENTER COMMAND UNIT (0-RELISTS -1-QUIT)'
        WRITE (*,*)
       READ (*,*,ERR=141) ICOMMAND
        IF (ICOMMAND.EQ.0) GO TO 5
        IF (ICOMMAND.EQ.-1) GO TO 10

142     WRITE (*,*) 'ENTER NUMBER RANGE OF SUBORDINATES TO FOLLOW'
       WRITE (*,*) 'THIS ORDER (#START #END) (0 0 FOR NONE)'
        WRITE (*,*)
       READ (*,*,ERR=142) NSUB1,NSUB2
        IF (NSUB1.LE.0) THEN
             NSUB1=0
             NSUB2=0
        ENDIF
        IF (NSUB2.LE.0) THEN
             NSUB1=0
             NSUB2=0
        ENDIF
C
150     WRITE (*,*)
       WRITE (*,*) 'MOVEMENT TYPE'
       WRITE (*,*) '1-FOLLOW ME'
       WRITE (*,*) ' -- ALL UNITS TURN AT A PIVOT POINT AND THEN HEAD'
       WRITE (*,*) '    TO A NEW OBJECTIVE WHERE THEY WILL STOP'
       WRITE (*,*) '2-COME ON LINE'
       WRITE (*,*) ' -- ALL UNITS FORM UP AND STOP ON A LINE DESCRIBED'
       WRITE (*,*) '    BY A PIVOT POINT AND LINE AZIMUTH'
       WRITE (*,*) '3-TURN'
       WRITE (*,*) ' -- ALL UNITS TURN TO THE NEW AZIMUTH AND PROCEED'
       WRITE (*,*) '    TO NEW OBJECTIVE POINT DESCRIBED BY THE'
       WRITE (*,*) '    DISTANCE ALONG THAT NEW AZIMUTH'
       WRITE (*,*) '4-SPEED CHANGE'
       WRITE (*,*) ' -- ALL UNITS GIVEN NEW SPEED'
       WRITE (*,*) '5-DEFILADE'
       WRITE (*,*) ' -- VEHICLES TAKE UP PREPARED DEFENSIVE POSITIONS'
      WRITE (*,*) '6-BREECH OBSTACLES'
      WRITE (*,*) ' -- VEHICLES ADVANCE SLOWLY AND CLEAR OBSTACLES'
      WRITE (*,*) '7-QUIT'
       WRITE (*,*)
      READ (*,*,ERR=150) MTYPE
       WRITE (*,*)
      IF (MTYPE.LE.0.OR.MTYPE.GE.8) GO TO 150
      IF (MTYPE.EQ.7) GO TO 5
      IF (MTYPE.EQ.5) CALL DIGIN
      IF (MTYPE.EQ.6) CALL BREECH
C
       IF (MTYPE.EQ.1) THEN
C       FOLLOW ME -- ALL UNITS TURN AT THEIR RELATIVE PIVOT POINTS
C       AND THEN HEAD TO NEW X,Y OBJECTIVE, WHERE THEY WILL STOP
151     WRITE (*,*) 'ENTER THE RANGE TO THE PIVOT POINT ON THE CURRENT'
       WRITE (*,*) 'LINE OF ADVANCE AND THE RANGE AND AZIMUTH TO THE'
       WRITE (*,*) 'NEW OBJECTIVE POINT FOR THE COMMAND UNIT'
       WRITE (*,*)
       READ (*,*,ERR=151) RANGEP,RANGEO,AZO
       IF (KSIDE.EQ.1) THEN
       MTYPEB(ICOMMAND)=1
       IDEFB(ICOMMAND)=0
       ELSE
       MTYPER(ICOMMAND)=1
       IDEFR(ICOMMAND)=0
       ENDIF
       IF (KSIDE.EQ.1) THEN
       IAZ=IBSA(ICOMMAND,2)
       X0=BXYA(ICOMMAND,1)
       Y0=BXYA(ICOMMAND,2)
       ELSE
       IAZ=IRSA(ICOMMAND,2)
       X0=RXYA(ICOMMAND,1)
       Y0=RXYA(ICOMMAND,2)
       ENDIF
       AZIN0=IAZ
       CALL FDIR (AZIN0,AZIN)
C
C       FIND COMMAND UNIT PIVOT POINT
       XP=X0+RANGEP*COS(AZIN*PI/180.)
       YP=Y0+RANGEP*SIN(AZIN*PI/180.)
C       FIND COMMAND UNIT OBJECTIVE POINT
       CALL FDIR (AZO,AZOUT)
       XO=XP+RANGEO*COS(AZOUT*PI/180.)
       YO=YP+RANGEO*SIN(AZOUT*PI/180.)
C       STORE THIS INFORMATION SO WHEN PIVOT POINT IS REACHED
C       UNIT WILL TURN AND HEAD TO NEW OBJECTIVE
       IF (KSIDE.EQ.1) THEN
       BXYP(ICOMMAND,1)=XP
       BXYP(ICOMMAND,2)=YP
       BXYO(ICOMMAND,1)=XO
       BXYO(ICOMMAND,2)=YO
       ELSE
       RXYP(ICOMMAND,1)=XP
       RXYP(ICOMMAND,2)=YP
       RXYO(ICOMMAND,1)=XO
       RXYO(ICOMMAND,2)=YO
       ENDIF
C
C       FIND RELATIVE PIVOT POINT AZIMUTH FOR DETERMINING SUBORDINATE
C       UNITS' RANGE TO THEIR PIVOT POINT
C       BISECT ANGLE BETWEEN OLD AZIMUTH AND NEW AZIMUTH AT PIVOT POINT
       AZINCL=(-AZIN-AZOUT)/2.
       AZBISECT=AZOUT+AZINCL
       SLOPE=TAN(AZBISECT/180.*PI)
       B=YP-SLOPE*XP
C       FOR EACH SUBORDINATE, PIVOT POINT IS ON THIS LINE
C       FIND INTERSECTION OF CURRENT AZIMUTH AND LOCATION WITH
C       THIS PIVOT AZIMUTH AND PIVOT POINT
C
       DO 170 I=NSUB1,NSUB2
       NSKIP=0
       IF (KSIDE.EQ.1) THEN
       IDEFB(I)=0
       MTYPEB(I)=1
       X1=BXYA(I,1)
       Y1=BXYA(I,2)
       IAZ=IBSA(I,2)
      IF (IBSYS(I,2).LT.1) NSKIP=1
       ELSE
       IDEFR(I)=0
       MTYPER(I)=1
```

```
      X1=RXYA(I,1)
      Y1=RXYA(I,2)
      IAZ=IRSA(I,2)
            IF (IRSYS(I,2).LT.1) NSKIP=1
      ENDIF
      IF (NSKIP.EQ.1) GO TO 170
C     FIND EACH UNITS' ORIGINAL LONGITUDINAL DISTANCE FROM THE
C     COMMAND UNIT ADD OR SUBTRACT THIS DISTANCE FROM RANGE TO
C     NEW OBJECTIVE
      DELX=X1-X0
      DELY=Y1-Y0
      IF (AZIN.GE.360) AZIN=AZIN-360.
      IF (AZIN.LT.0.) AZIN=AZIN+360.
C     FIND DISTANCE BETWEEN UNIT LOCATION AND AZIMUTH LINE THROUGH
C     COMMAND UNIT
      SLOPEL=TAN(AZIN/180.*PI)
      BL=Y0-SLOPEL*X0
      SLOPEP=-SLOPEL
      BP=Y1-SLOPEP*X1
C     FIND INTERSECTION POINT
      XINT=(BP-BL)/(SLOPEL-SLOPEP)
      YINT=BL+SLOPEL*XINT
C     FIND DISTANCES
      DISTL2=(XINT-X1)**2+(YINT-Y1)**2
      DIST02=(DELX)**2+(DELY)**2
      IF (DIST02.LT.DISTL2) THEN
            DIST=0.0
      ELSE
            DIST=(DIST02-DISTL2)**.5
      ENDIF
C
      IF (AZIN.EQ.90.) DIST0=DELY
      IF (AZIN.EQ.270.) DIST0=-DELY
      IF (AZIN.EQ.0.) DIST0=DELX
      IF (AZIN.EQ.180.) DIST0=-DELX
      IF (AZIN.GT.0.0.AND.AZIN.LT.90.0) THEN
            IF (DELX.GT.0.) THEN
                  DIST0=DIST
            ELSE
                  DIST0=-DIST
            ENDIF
      ENDIF
      IF (AZIN.GT.90.0.AND.AZIN.LT.270.0) THEN
            IF (DELX.GT.0.) THEN
                  DIST0=-DIST
            ELSE
                  DIST0=DIST
            ENDIF
      ENDIF
      IF (AZIN.GT.270.0.AND.AZIN.LT.360.0) THEN
            IF (DELX.GT.0.) THEN
                  DIST0=DIST
            ELSE
                  DIST0=-DIST
            ENDIF
      ENDIF
C
      RANGE=RANGEO+DIST0
      AZIN1=IAZ
      CALL FDIR (AZIN1,AZOUT1)
      SLOPE1=TAN(AZOUT1/180.*PI)
      B1=Y1-SLOPE1*X1
C     INTERSECTION IS PIVOT POINT
      XINT=(B1-BL)/(SLOPEL-SLOPE1)
      YINT=B1+SLOPE1*XINT
      IF (KSIDE.EQ.1) THEN
      BXYP(I,1)=XINT
      BXYP(I,2)=YINT
      XO=XINT+RANGE*COS(AZOUT/180.*PI)
      YO=YINT+RANGE*SIN(AZOUT/180.*PI)
      BXYO(I,1)=XO
      BXYO(I,2)=YO
      ELSE
      RXYP(I,1)=XINT
      RXYP(I,2)=YINT
      XO=XINT+RANGE*COS(AZOUT/180.*PI)
      YO=YINT+RANGE*SIN(AZOUT/180.*PI)
      RXYO(I,1)=XO
      RXYO(I,2)=YO
      ENDIF
170   CONTINUE

      ENDIF
C
      IF (MTYPE.EQ.2) THEN
C     COME ON LINE -- ALL UNITS FORM UP AND STOP ON A LINE DESCRIBED
C     BY THE PIVOT POINT AND LINE AZIMUTH
171     WRITE (*,*) 'ENTER THE RANGE TO THE PIVOT POINT ON THE CURRENT'
      WRITE (*,*) 'LINE OF ADVANCE AND AN AZIMUTH'
      WRITE (*,*) 'DESCRIBING THE LINE TO FORM UP ON'
      WRITE (*,*)
      READ (*,*,ERR=171) RANGEO,AZIMUTH
      CALL FDIR (AZIMUTH,AZOUT)
      AZIMUTH=AZOUT
C     FIND THE DISTANCE EACH UNIT IS FROM THE PIVOT POINT
C     FIND NEW POSITION ON THE LINE AT THE SAME DISTANCE
C     TAKE STRAIGHT PATH TO THIS POINT
      IF (KSIDE.EQ.1) THEN
      IDEFB(ICOMMAND)=0
      XO=BXYA(ICOMMAND,1)
      YO=BXYA(ICOMMAND,2)
      IAZ=IBSA(ICOMMAND,2)
      AZIN0=IAZ
      CALL FDIR (AZIN0,AZIN)
C     FIND PIVOT POINT
      XP=XO+RANGEO*COS(AZIN/180.*PI)
      YP=YO+RANGEO*SIN(AZIN/180.*PI)
      X2=XP+RANGEO*COS(AZIMUTH/180.*PI)
      Y2=YP+RANGEO*SIN(AZIMUTH/180.*PI)
      BXYO(ICOMMAND,1)=X2
      BXYO(ICOMMAND,2)=Y2
      ELSE
      IDEFR(ICOMMAND)=0
      XO=RXYA(ICOMMAND,1)
      YO=RXYA(ICOMMAND,2)
      IAZ=IRSA(ICOMMAND,2)
      AZIN0=IAZ
      CALL FDIR (AZIN0,AZIN)
C     FIND PIVOT POINT
      XP=XO+RANGEO*COS(AZIN/180.*PI)
      YP=YO+RANGEO*SIN(AZIN/180.*PI)
      X2=XP+RANGEO*COS(AZIMUTH/180.*PI)
      Y2=YP+RANGEO*SIN(AZIMUTH/180.*PI)
      RXYO(ICOMMAND,1)=X2
      RXYO(ICOMMAND,2)=Y2
      ENDIF
      TOP=Y2-Y0
      BOT=X2-X0
      IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
      IF (BOT.LT.0.0) THEN
            IF (TOP.LT.0.0) DIR=DIR+180.
            IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
            IF (TOP.LT.0.0) DIR=360.-DIR
      ENDIF
      CALL FAZ (DIR,AZIN)
      IF (KSIDE.EQ.1) THEN
      IBSA(ICOMMAND,2)=INT(AZIN)
      MTYPEB(ICOMMAND)=2
      ELSE
      IRSA(ICOMMAND,2)=INT(AZIN)
      MTYPER(ICOMMAND)=2
      ENDIF
      DO 180 I=NSUB1,NSUB2
      NSKIP=0
      IF (KSIDE.EQ.1) THEN
      IDEFB(I)=0
      IF (IBSYS(I,2).LT.1) NSKIP=1
      X1=BXYA(I,1)
      Y1=BXYA(I,2)
      ELSE
      IDEFR(I)=0
      IF (IRSYS(I,2).LT.1) NSKIP=1
      X1=RXYA(I,1)
      Y1=RXYA(I,2)
      ENDIF
      IF (NSKIP.EQ.1) GO TO 180
C     FIND EACH UNITS' ORIGINAL LONGITUDINAL DISTANCE FROM THE
C     COMMAND UNIT ADD OR SUBTRACT THIS DISTANCE FROM RANGE TO
C     PIVOT POINT
      DELX=X1-X0
      DELY=Y1-Y0
      IF (AZIN.GE.360) AZIN=AZIN-360.
      IF (AZIN.LT.0.) AZIN=AZIN+360.
C     FIND DISTANCE BETWEEN UNIT LOCATION AND AZIMUTH LINE THROUGH
C     COMMAND UNIT
      SLOPEL=TAN(AZIN/180.*PI)
      BL=Y0-SLOPEL*X0
      SLOPEP=-SLOPEL
      BP=Y1-SLOPEP*X1
C     FIND INTERSECTION POINT
      XINT=(BP-BL)/(SLOPEL-SLOPEP)
      YINT=BL+SLOPEL*XINT
C     FIND DISTANCES
      DISTL2=(XINT-X1)**2+(YINT-Y1)**2
      DIST02=(DELX)**2+(DELY)**2
      IF (DIST02.LT.DISTL2) THEN
            DIST=0.0
      ELSE
            DIST=(DIST02-DISTL2)**.5
      ENDIF
C
      IF (AZIN.EQ.90.) DIST0=DELY
      IF (AZIN.EQ.270.) DIST0=-DELY
      IF (AZIN.EQ.0.) DIST0=DELX
      IF (AZIN.EQ.180.) DIST0=-DELX
      IF (AZIN.GT.0.0.AND.AZIN.LT.90.0) THEN
            IF (DELX.GT.0.) THEN
                  DIST0=DIST
            ELSE
                  DIST0=-DIST
            ENDIF
      ENDIF
      IF (AZIN.GT.90.0.AND.AZIN.LT.270.0) THEN
            IF (DELX.GT.0.) THEN
                  DIST0=-DIST
            ELSE
                  DIST0=DIST
            ENDIF
      ENDIF
      IF (AZIN.GT.270.0.AND.AZIN.LT.360.0) THEN
            IF (DELX.GT.0.) THEN
                  DIST0=DIST
```

```
            ELSE
                  DIST0=-DIST
            ENDIF
      ENDIF
C
      RANGE=RANGEO+DIST0
      IF (KSIDE.EQ.1) THEN
      X2=XP+RANGE*COS(AZIMUTH/180.*PI)
      Y2=YP+RANGE*SIN(AZIMUTH/180.*PI)
      BXYO(I,1)=X2
      BXYO(I,2)=Y2
      ELSE
      X2=XP+RANGE*COS(AZIMUTH/180.*PI)
      Y2=YP+RANGE*SIN(AZIMUTH/180.*PI)
      RXYO(I,1)=X2
      RXYO(I,2)=Y2
      ENDIF
C
      TOP=Y2-Y1
      BOT=X2-X1
       IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
      IF (BOT.LT.0.0) THEN
            IF (TOP.LT.0.0) DIR=DIR+180.
            IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
            IF (TOP.LT.0.0) DIR=360.-DIR
      ENDIF
      CALL FAZ (DIR,AZIN)
       IF (KSIDE.EQ.1) THEN
       IBSA(I,2)=INT(AZIN)
       MTYPEB(I)=2
       ELSE
       IRSA(I,2)=INT(AZIN)
       MTYPER(I)=2
       ENDIF
180    CONTINUE

      ENDIF
      IF (MTYPE.EQ.3) THEN
C     TURN - ALL UNITS TURN TO THE NEW AZIMUTH AND PROCEED TO NEW
C     OBJECTIVE POINT DESCRIBED BY THE DISTANCE ALONG THAT NEW AZIMUTH
181    WRITE (*,*) 'ENTER NEW AZIMUTH AND RANGE TO OBJECTIVE FOR'
      WRITE (*,*) 'COMMAND UNIT'
       WRITE (*,*)
      READ (*,*,ERR=181) AZIMUTH,RANGE
      CALL FDIR (AZIMUTH,AZOUT)
       IF (KSIDE.EQ.1) THEN
       IDEFB(ICOMMAND)=0
       X1=BXYA(ICOMMAND,1)
       Y1=BXYA(ICOMMAND,2)
       X2=X1+RANGE*COS(AZOUT/180.*PI)
       Y2=Y1+RANGE*SIN(AZOUT/180.*PI)
       BXYO(ICOMMAND,1)=X2
       BXYO(ICOMMAND,2)=Y2
       IBSA(ICOMMAND,2)=INT(AZIMUTH)
       MTYPEB(ICOMMAND)=3
       ELSE
       IDEFR(ICOMMAND)=0
       X1=RXYA(ICOMMAND,1)
       Y1=RXYA(ICOMMAND,2)
       X2=X1+RANGE*COS(AZOUT/180.*PI)
       Y2=Y1+RANGE*SIN(AZOUT/180.*PI)
       RXYO(ICOMMAND,1)=X2
       RXYO(ICOMMAND,2)=Y2
       IRSA(ICOMMAND,2)=INT(AZIMUTH)
       MTYPER(ICOMMAND)=3
       ENDIF
       DO 190 I=NSUB1,NSUB2
       IF (KSIDE.EQ.1) THEN
       IDEFB(I)=0
                        IF (IBSYS(I,2).GE.1) THEN
            X1=BXYA(I,1)
            Y1=BXYA(I,2)
            X2=X1+RANGE*COS(AZOUT/180.*PI)
            Y2=Y1+RANGE*SIN(AZOUT/180.*PI)
            BXYO(I,1)=X2
            BXYO(I,2)=Y2
            IBSA(I,2)=INT(AZIMUTH)
            MTYPEB(I)=3
            ENDIF
      ELSE
      IDEFR(I)=0
                        IF (IRSYS(I,2).GE.1) THEN
            X1=RXYA(I,1)
            Y1=RXYA(I,2)
            X2=X1+RANGE*COS(AZOUT/180.*PI)
            Y2=Y1+RANGE*SIN(AZOUT/180.*PI)
            RXYO(I,1)=X2
            RXYO(I,2)=Y2
            IRSA(I,2)=INT(AZIMUTH)
            MTYPER(I)=3
            ENDIF
      ENDIF
190    CONTINUE
      ENDIF
C
      IF (MTYPE.EQ.4) THEN
C     SPEED CHANGE
191    WRITE (*,*) 'ENTER DESIRED SPEED'
       WRITE (*,*)
      READ (*,*,ERR=191) SPEED
       IF (SPEED.LT.0.0) GO TO 191
       IF (KSIDE.EQ.1) THEN

C      TO ACCOUNT FOR ACCELERATION PARAMETERS OF VEHICLE,
C      THIS DESIRED SPEED IS CONTINUOUSLY MATCHED AGAINST THE
C      MOBILITY PARAMETERS ENTERED FOR THIS SYSTEM AND AN ACTUAL
C      SPEED IS DETERMINED FOR EACH TIME STEP.
C      VARIABLES ISPDB(200) AND ISPDR(200) ARE THE ACTUAL SPEEDS
      IF (MBREECH(ICOMMAND,1).EQ.1) SPEED=4.
       IBSA(ICOMMAND,1)=INT(SPEED)
      IF (IBAVL(ICOMMAND).EQ.0) IBSA(ICOMMAND,1)=0
      IDEFB(ICOMMAND)=0
C        MTYPEB(ICOMMAND)=4
       IF (SPEED.LE.0.0) MODEMB(ICOMMAND)=3
       ELSE
      IDEFR(ICOMMAND)=0
      IF (MBREECH(ICOMMAND,2).EQ.1) SPEED=4.
      IRSA(ICOMMAND,1)=INT(SPEED)
      IF (IRAVL(ICOMMAND).EQ.0) IRSA(ICOMMAND,1)=0
C        MTYPER(ICOMMAND)=4
       IF (SPEED.LE.0.0) MODEMR(ICOMMAND)=3
       ENDIF
       DO 200 I=NSUB1,NSUB2
       IF (KSIDE.EQ.1) THEN
      IDEFB(I)=0
      IF (MBREECH(I,1).EQ.1) SPEED=4.
                  IF (IBSYS(I,2).GE.1) THEN
                  IBSA(I,1)=INT(SPEED)
                  IF (IBAVL(I).EQ.0) IBSA(I,1)=0
C        MTYPEB(I)=4
       IF (SPEED.LE.0.0) MODEMB(I)=3
            ENDIF
       ELSE
      IDEFR(I)=0
      IF (MBREECH(I,2).EQ.1) SPEED=4.
                  IF (IRSYS(I,2).GE.1) THEN
                  IRSA(I,1)=INT(SPEED)
                  IF (IRAVL(I).EQ.0) IRSA(I,1)=0
C        MTYPER(I)=4
       IF (SPEED.LE.0.0) MODEMR(I)=3
            ENDIF
       ENDIF
200    CONTINUE
      ENDIF

      IF (MTYPE.EQ.5) GO TO 10
      GO TO 150

1000   CONTINUE
      RETURN
      END


      SUBROUTINE FDIR (AZIN,AZOUT)
C
CC    TAKES AZIMUTH ANGLE IN AND CONVERTS TO CARTESIAN ANGLE OUT
C
      AZOUT=-AZIN+90.
       IF (AZOUT.LT.0.0) AZOUT=AZOUT+360.
      RETURN
      END


      SUBROUTINE FAZ (AZOUT,AZIN)
C
CC    TAKES CARTESIAN ANGLE OUT AND GIVES AZIMUTH ANGLE IN
C
      AZIN=-(AZOUT-90.)
       IF (AZIN.LT.0.0) AZIN=AZIN+360.
      RETURN
      END


      SUBROUTINE BATTLE2
C     BATTLE.FOR
C
C     MOVE AND SHOOT UNITS
C
C     CYCLES THROUGH SIXTY MOVEMENT AND SHOOTING TIME STEPS
C     THEN RETURNS TO COMMAND LEVEL
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
C     COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
C    & NC(15),NCOL(15),JCON
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
     COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
    & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
     COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
    & L1BM,L2M,L3M,L1RM
     COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
    & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
    & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
    & TEMINR(30),ACQB(10,5),ACQR(10,5),SRCHB(10),SRCHR(10),
    & MILB(10,3),MILR(10,3)
     COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
     COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
    & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
    & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
    & IAMB(200)
     COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
    & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
```

```
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
       COMMON/FIRE/IBPRIME(200,2),IBSECOND(200,2),IRPRIME(200,2),
     & IRSECOND(200,2),WRNGB(200,2),WRNGR(200,2)
       COMMON/MASKV/VMSK(400,2)
       COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
       COMMON/EDSMKG/ISGERASE,PSGERASE(400,2),ASGERASE(400)
       COMMON/EENPUFFS/IEERASE,PTERASE(2400,3)
       COMMON/EDPUFFS/IDERASE,PTDRASE(2400,3)
       COMMON/TRACKING/BOREB(10,3),BORER(10,3),IRNGB(30,4),IRNGR(30,4),
     & ITRKB(10,4),ITRKR(10,4),TOFB(10,3),TOFR(10,3)
       COMMON/ALIVE/NBA,NRA,BTHREAT,RTHREAT
       COMMON/PRIOTY/KPRIB(200,10),KPRIR(200,10)
       COMMON/ACEL/ACELB(10,5),ACELR(10,5),GRADEB(200),GRADER(200)
       COMMON/EMENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
     & ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2)
             COMMON/FLAGS/ITER
             COMMON/TURRET/AZTB(200),AZTR(200),TRATEB(10),TRATER(10)
             COMMON/SIMAPS/ISIMB(200),ISIMR(200)
             COMMON/OBSCURE/NGRENB(200,2),NGRENR(200,2),IBESMK(200),
     & IRESMK(200)
             COMMON/ESMOKE/TESMK(200)
             COMMON/COOKERS/NCOOK,XYCOOK(400,2),PBURNB(10),PBURNR(10),
     & KBURNB(200),KBURNR(200)
             COMMON/MINES/NMINES,XYMINE(100,2),WLMINE(100,2),NAZMINE(100,3)
             COMMON/MINEKILLS/MINEK(200,2)
             COMMON/MINEBREECH/MBREECH(200,2),MINEON(100)

             COMMON/ARTTGT/ARTGTB(200,2),ARTGTR(200,2),ITLB,ITLR
             COMMON/AMISH/NFMIS(2),MFUB(200,5),MFUR(200,5),XYFUB(200,2),
     & XYFUR(200,2),MILAB(200,10),MILAR(200,10)
             COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(1
0)
     & ,NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5)
     & ,IAFAFB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
     & ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
     & ISCOOTB(10,4),ISCOOTR(10,4)
             COMMON/AAVAIL/IBAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)
             COMMON/SCOOT/ITSC(200,2),ITAVL(200,2)

             INTEGER ARTGTB,ARTGTR
              DIMENSION BACEL(200)
C
             CHARACTER*9 BFNAME,RFNAME,BANAME,RANAME
C     CHARACTER*80 LINE
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*1 C2
C     CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
C      WRITE (*,*) '2'
C      WRITE (*,*) (RB(ISR),ISR=1,36)
C      WRITE (*,*) (RR(ISR),ISR=1,36)
      PI=3.14159
       DT=IDTIME
             ITLB=0
             ITLR=0

      CALL LIVING
C
C      INITIALIZE MOVEMENT AND FIRE MODES FOR THIS CYCLE
C
C      TCMMB  CYCLE TIME IN EACH MANEUVER MODE FOR BLUE SYSTEM
C      TCMMR  CYCLE TIME IN EACH MANEUVER MODE FOR RED SYSTEM
C      TCFMB  CYCLE TIME IN EACH FIRE MODE FOR BLUE SYSTEM
C      TCFMR  CYCLE TIME IN EACH FIRE MODE FOR RED SYSTEM
C
      DO 800 I=1,NB
       TCMMB(I,1)=0.0
       TCMMB(I,2)=0.0
       TCMMB(I,3)=0.0
       TCFMB(I,1)=0.0
       TCFMB(I,2)=0.0
       TCFMB(I,3)=0.0
       TCFMB(I,4)=0.0
       TCFMB(I,5)=0.0
       TCFMB(I,6)=0.0
       TCFMB(I,7)=0.0
       TCFMB(I,8)=0.0
       TCFMB(I,9)=0.0
       TEM(I,1)=0.0
             TEM(I,2)=0.0
             TESMK(I)=0.0
       IKILLB(I)=0
800    CONTINUE
C
      DO 850 I=1,NR
       TCMMR(I,1)=0.0
       TCMMR(I,2)=0.0
       TCMMR(I,3)=0.0
       TCFMR(I,1)=0.0
       TCFMR(I,2)=0.0
       TCFMR(I,3)=0.0
       TCFMR(I,4)=0.0
       TCFMR(I,5)=0.0
       TCFMR(I,6)=0.0
       TCFMR(I,7)=0.0
       TCFMR(I,8)=0.0
       TCFMR(I,9)=0.0
       IKILLR(I)=0
850    CONTINUE
C
       CALL PLOTS (0,1,0)
C      ZOOM TO ACTIVE GRID
       CALL BZOOM
C      PLOT CONTOURS
       IF (ITER.EQ.1) CALL BCONT (0)
C      PLOT GRID
      CALL GRID2
      CALL PLOTMINES
      TGREN=DTPUFF+ITIME
      TPUFF=DTPUFF+ITIME

       DO 1000 ICYCLE=1,60

C      INITIALIZE ACTIVE PROTECTION SIMULTANEITY CHECK
      DO 851 ISIM=1,200
      ISIMB(ISIM)=0
851     ISIMR(ISIM)=0

C      PLOT CURRENT UNIT POSITIONS
      CALL UNITS
C      SAVE THESE POSITIONS TO ERASE LATER
      CALL OLDPOS
C
       IF (ITIME.GE.TPUFF) THEN
       ISGERASE=0
       IEERASE=0
       IDERASE=0
C      PLOT DUST CLOUDS
C       WRITE (*,*) 'NDUST ',NDUST
C       WRITE (*,*) 'NENG ',NENG
       IF (IDUST.EQ.1.AND.NDUST.GT.0) CALL PDUST
C      PLOT ENGINE SMOKE CLOUDS
       IF (NENG.GT.0) CALL PENG
C      PLOT SMOKE GRENADE CURTAINS
      IF (NSMK.GT.0) CALL PSMKCURTAINS
C      PLOT ARTILLERY SMOKE CURTAINS
      IF (NART.GT.0) CALL PARTSMK
      ENDIF
       IF (ITIME.GE.TGREN) THEN
C      FIRE SMOKE GRENADES AS ORDERED
      CALL GRENADES
       TGREN=1.E6
       ENDIF
C
C      RESOLVE ARTILLERY IN-DIRECT FIRE MISSIONS
      CALL FIREMISH (ITIME,IDTIME)
C
C      EXECUTE MOVEMENT ORDERS
C
C      BLUE
       DO 130 K=1,NB
C
C      CHECK TO SEE IF THIS UNIT IS KILLED
      IF (IBSYS(K,2).EQ.0) GO TO 130
C
C      ACCUMULATE CYCLE TIME FOR ENGINE SMOKE IF TURNED ON
      IF (IBESMK(K).EQ.1) TESMK(K)=TESMK(K)+DT

C
C      CHECK TO SEE IF THIS IS AN ARTILLERY UNIT AND SHOULD BEGIN
C       SCOOTING
      KASYS=IBSYS(K,1)
      IF (KASYS.GT.10) THEN
       KASYS=KASYS-10
       IF (ISCOOTB(KASYS,1).EQ.1) THEN
C        THIS UNIT SHOOTS AND SCOOTS
       IF (IBAVL(K).EQ.0) THEN
C        THIS UNIT IS CURRENTLY INVOLVED IN A FIRE MISSION
        IF (ITIME.GE.ITSC(K,1)) THEN
C          ITS TIME TO SCOOT
        IBAVL(K)=-1
        IBSA(K,1)=50
C          DEFINE NEW MOVEMENT OBJECTIVE BASED ON SCOOT DISTANCE
        AZIMUTH=REAL(IBSA(K,2))
        CALL FDIR (AZIMUTH,AZOUT)
        OBJR=REAL(ISCOOTB(KASYS,3))/1000.
        X1=BXYA(K,1)
        Y1=BXYA(K,2)
        X2=X1+OBJR*COS(AZOUT/180.*PI)
        Y2=Y1+OBJR*SIN(AZOUT/180.*PI)
        BXYO(K,1)=X2
        BXYO(K,2)=Y2
       ENDIF
       ENDIF
       IF (IBAVL(K).EQ.-1) THEN
C        THIS UNIT IS CURRENTLY SCOOTING
C        SEE IF IT IS TIME TO UN-SCOOT IT AND MAKE IT AVAILABLE FOR
C        A FIRE MISSION
       IF (ITIME.GE.ITAVL(K,1)) IBAVL(K)=1
       ENDIF

       ELSE

C        THIS ARTILLERY UNIT DOES NOT SCOOT
C        SEE IF IT IS INVOLVED IN A FIRE MISSION AND IS NOW TIME TO
C        MAKE AVAILABLE AGAIN
```

```
            IF (IBAVL(K).EQ.0) THEN
C       THIS UNIT IS CURRENTLY INVOLVED IN A FIRE MISSION
C       SEE IF IT IS TIME TO MAKE IT AVAILABLE AGAIN
                IF (ITIME.GE.ITAVL(K,1)) IBAVL(K)=1
              ENDIF
             ENDIF
            ENDIF
C
C     UPDATE TERRAIN GRADE FOR THIS VEHICLE
      KSIDE=1
      CALL GRADE (KSIDE,K)

            BACEL(K)=0.0
C     COMPARE DESIRED SPEED AGAINST ACTUAL SPEED ON THIS GRADE
C     AND DETERMINE CURRENT ACCELERATION
      CALL FINDACEL (KSIDE,K,ARATE)
            BACEL(K)=ARATE
C
C     CHECK ACCELERATION AND SELECT MANEUVER MODE
      IF (ARATE.EQ.0.0) MODEMB(K)=1
C       VEHICLE DRIVES AT CURRENT SPEED
      IF (ARATE.LE.0.0.AND.ISPDB(K).EQ.0) THEN
C       VEHICLE IS STOPPED AND HAS NO DESIRE TO MOVE
       MODEMB(K)=3
       ARATE=0.0
      ENDIF
      IF (ARATE.LT.0.0.AND.IBSA(K,1).NE.0) THEN
C       DECELERATE ACCORDINGLY
       MODEMB(K)=1
      ENDIF
      IF (ARATE.LT.0.0.AND.IBSA(K,1).EQ.0) THEN
C       STOP VEHICLE
       MODEMB(K)=3
       ISPDB(K)=IBSA(K,1)
       ARATE=0.0
      ENDIF
      IF (ARATE.GT.0.0) MODEMB(K)=2
C       ACCELERATE ACCORDINGLY

C     INCREASE CYCLE TIME IN THIS MODE
      TCMMB(K,MODEMB(K))=TCMMB(K,MODEMB(K))+DT
C
      IF (MTYPEB(K).NE.1) THEN
C       ALL OF THESE MOVEMENT TYPES SIMPLY INVOLVE CONTINUING IN
C       CURRENT DIRECTION UNTIL OBJECTIVE IS REACHED
       AZIN=IBSA(K,2)
       CALL FDIR (AZIN,AZOUT)
C       CHECK CURRENT DISTANCE TO OBJECTIVE
      ROBJ1=((BXYA(K,1)-BXYO(K,1))**2+(BXYA(K,2)-BXYO(K,2))**2)**.5
C       MOVE UNIT
      DELPOS=(ISPDB(K)*DT+ARATE/2.*DT*DT)/3600.
CI
C       before moving, evaluate minefields
              IF (NMINES.GE.1.AND.DELPOS.GT.0.0) THEN
              CALL MINEFIELD (1,K,DELPOS,AZOUT,INNOW,INNEXT)
               IF (INNOW.NE.0.OR.INNEXT.NE.0) THEN
C         IN OR ENTERING A MINEFIELD, DOES HE KNOW IT?
                INKNOW=0
                NXKNOW=0
                IF (INNOW.NE.0.AND.NAZMINE(INNOW,3).EQ.1) INKNOW=1
                IF (INNEXT.NE.0.AND.NAZMINE(INNEXT,3).EQ.1) NXKNOW=1
                IF (INKNOW.EQ.1.OR.NXKNOW.EQ.1) THEN
                IF (MBREECH(K,1).EQ.0) THEN
C         STOP AND AWAIT ORDERS, YOU'RE IN OR GOING INTO A KNOWN MINEFIELD
                 DELPOS=0.0
                 ARATE=0.0
                 ISPDB(K)=0
                 MODEMB(K)=3
                 IBSA(K,1)=0
                ENDIF
                IF (MBREECH(K,1).EQ.-1)
     &             CALL MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)
                ELSE
C         YOU'RE MOVING THROUGH A MINEFIELD AND DON'T KNOW IT
C         OR YOU'RE BREECHING IT.
C         EVALUATE PROBABILITY OF BEING KILLED
                 IF (MBREECH(K,1).NE.1)
     &             CALL MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)
               ENDIF
              ENDIF
             ENDIF
CI
            BXYA(K,1)=BXYA(K,1)+DELPOS*COS(AZOUT*PI/180.)
       BXYA(K,2)=BXYA(K,2)+DELPOS*SIN(AZOUT*PI/180.)
C       CHANGE SPEED
       ISPDB(K)=ISPDB(K)+ARATE*DT
C       CHECK NEW DISTANCE TO OBJECTIVE
      ROBJ2=((BXYA(K,1)-BXYO(K,1))**2+(BXYA(K,2)-BXYO(K,2))**2)**.5
C       IF THIS DISTANCE HAS GOTTEN BIGGER, THEN OBJECTIVE HAS JUST
C       BEEN PASSED, STOP VEHICLE
            IF (ROBJ2.GE.ROBJ1) THEN
              IBSA(K,1)=0
              ISPDB(K)=0
              MODEMB(K)=3
            ENDIF
      ELSE
C       FOLLOW-ME MOVEMENT TYPE, CHECK TO SEE IF PIVOT POINT IS
C       NOW REACHED.  IF REACHED, CHANGE MOVEMENT TYPE AND AZIMUTH
C       AND OBJECTIVE
       AZIN=IBSA(K,2)
       CALL FDIR (AZIN,AZOUT)
C       CHECK CURRENT DISTANCE TO PIVOT POINT
      ROBJ1=((BXYA(K,1)-BXYP(K,1))**2+(BXYA(K,2)-BXYP(K,2))**2)**.5
C       MOVE UNIT
      DELPOS=(ISPDB(K)*DT+ARATE/2.*DT*DT)/3600.
CI
C       before moving, evaluate minefields
      IF (NMINES.GE.1.AND.DELPOS.GT.0.0) THEN
      CALL MINEFIELD (1,K,DELPOS,AZOUT,INNOW,INNEXT)
       IF (INNOW.NE.0.OR.INNEXT.NE.0) THEN
C         IN OR ENTERING A MINEFIELD, DOES HE KNOW IT?
        INKNOW=0
        NXKNOW=0
        IF (INNOW.NE.0.AND.NAZMINE(INNOW,3).EQ.1) INKNOW=1
        IF (INNEXT.NE.0.AND.NAZMINE(INNEXT,3).EQ.1) NXKNOW=1
        IF (INKNOW.EQ.1.OR.NXKNOW.EQ.1) THEN
        IF (MBREECH(K,1).EQ.0) THEN
C         STOP AND AWAIT ORDERS, YOU'RE IN OR GOING INTO A KNOWN MINEFIELD
         DELPOS=0.0
         ARATE=0.0
         ISPDB(K)=0
         MODEMB(K)=3
         IBSA(K,1)=0
        ENDIF
        IF (MBREECH(K,1).EQ.-1)
     &       CALL MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)
        ELSE
C         YOU'RE MOVING THROUGH A MINEFIELD AND DON'T KNOW IT
C         OR YOU'RE BREECHING IT.
C         EVALUATE PROBABILITY OF BEING KILLED
         IF (MBREECH(K,1).NE.1)
     &       CALL MINEKILL (1,K,INNOW,INNEXT,DELPOS)
        ENDIF
       ENDIF
      ENDIF
CI
      BXYA(K,1)=BXYA(K,1)+DELPOS*COS(AZOUT*PI/180.)
       BXYA(K,2)=BXYA(K,2)+DELPOS*SIN(AZOUT*PI/180.)
C       CHANGE SPEED
       ISPDB(K)=ISPDB(K)+ARATE*DT
C       CHECK NEW DISTANCE TO PIVOT POINT
      ROBJ2=((BXYA(K,1)-BXYP(K,1))**2+(BXYA(K,2)-BXYP(K,2))**2)**.5
C       IF THIS DISTANCE HAS GOTTEN BIGGER, THEN PIVOT POINT HAS JUST
C       BEEN PASSED, TURN VEHICLE
       IF (ROBJ2.GE.ROBJ1) THEN
            MTYPEB(K)=0
            X2=BXYO(K,1)
            Y2=BXYO(K,2)
            X1=BXYA(K,1)
            Y1=BXYA(K,2)
               TOP=Y2-Y1
            BOT=X2-X1
             IF (BOT.EQ.0.0) BOT=BOT+.001
            DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
                IF (BOT.LT.0.0) THEN
                    IF (TOP.LT.0.0) DIR=DIR+180.
                    IF (TOP.GT.0.0) DIR=180.-DIR
                ELSE
                IF (TOP.LT.0.0) DIR=360.-DIR
                ENDIF
            AZOUT=DIR
            CALL FAZ (AZOUT,AZIN)
            IAZ=INT(AZIN)
            IBSA(K,2)=IAZ
       ENDIF
      ENDIF
C       SET TURRET POSITION IF STILL IN SURVEILLANCE MODE
      IF (MODEFB(K).EQ.7) AZTB(K)=IBSA(K,2)
130    CONTINUE
C
C       RED
       DO 140 K=1,NR
C       CHECK TO SEE IF THIS UNIT IS KILLED
       IF (IRSYS(K,2).EQ.0) GO TO 140
C
C       CHECK TO SEE IF THIS IS AN ARTILLERY UNIT AND SHOULD BEGIN
C       SCOOTING
      KASYS=IRSYS(K,1)
      IF (KASYS.GT.10) THEN
       KASYS=KASYS-10
       IF (ISCOOTR(KASYS,1).EQ.1) THEN
C         THIS UNIT SHOOTS AND SCOOTS
        IF (IRAVL(K).EQ.0) THEN
C         THIS UNIT IS CURRENTLY INVOLVED IN A FIRE MISSION
        IF (ITIME.GE.ITSC(K,2)) THEN
C         ITS TIME TO SCOOT
         IRAVL(K)=-1
         IRSA(K,1)=50
C           DEFINE NEW MOVEMENT OBJECTIVE BASED ON SCOOT DISTANCE
         AZIMUTH=REAL(IRSA(K,2))
         CALL FDIR (AZIMUTH,AZOUT)
         OBJR=REAL(ISCOOTR(KASYS,3))/1000.
         X1=RXYA(K,1)
         Y1=RXYA(K,2)
         X2=X1+OBJR*COS(AZOUT/180.*PI)
         Y2=Y1+OBJR*SIN(AZOUT/180.*PI)
         RXYO(K,1)=X2
         RXYO(K,2)=Y2
        ENDIF
       ENDIF
       IF (IRAVL(K).EQ.-1) THEN
C         THIS UNIT IS CURRENTLY SCOOTING
C         SEE IF IT IS TIME TO UN-SCOOT IT AND MAKE IT AVAILABLE FOR
C         A FIRE MISSION
```

```
            IF (ITIME.GE.ITAVL(K,2)) IRAVL(K)=1
           ENDIF

           ELSE

C     THIS ARTILLERY UNIT DOES NOT SCOOT
C     SEE IF IT IS INVOLVED IN A FIRE MISSION AND IS NOW TIME TO
C     MAKE AVAILABLE AGAIN
           IF (IRAVL(K).EQ.0) THEN
C     THIS UNIT IS CURRENTLY INVOLVED IN A FIRE MISSION
C     SEE IF IT IS TIME TO MAKE IT AVAILABLE AGAIN
            IF (ITIME.GE.ITAVL(K,2)) IRAVL(K)=1
           ENDIF
          ENDIF
         ENDIF

C    UPDATE TERRAIN GRADE FOR THIS VEHICLE
     KSIDE=2
     CALL GRADE (KSIDE,K)

C    COMPARE DESIRED SPEED AGAINST ACTUAL SPEED ON THIS GRADE
C    AND DETERMINE CURRENT ACCELERATION
     CALL FINDACEL (KSIDE,K,ARATE)
C
C    CHECK ACCELERATION AND SELECT MANEUVER MODE
     IF (ARATE.EQ.0.0) MODEMR(K)=1
C     VEHICLE DRIVES AT CURRENT SPEED
     IF (ARATE.LE.0.0.AND.ISPDR(K).EQ.0) THEN
C     VEHICLE IS STOPPED AND HAS NO DESIRE TO MOVE
      MODEMR(K)=3
      ARATE=0.0
     ENDIF
     IF (ARATE.LT.0.0.AND.IRSA(K,1).NE.0) THEN
C     DECELERATE ACCORDINGLY
      MODEMR(K)=1
     ENDIF
     IF (ARATE.LT.0.0.AND.IRSA(K,1).EQ.0) THEN
C     STOP VEHICLE
      MODEMR(K)=3
      ISPDR(K)=IRSA(K,1)
      ARATE=0.0
     ENDIF
     IF (ARATE.GT.0.0) MODEMR(K)=2
C     ACCELERATE ACCORDINGLY
C
C    INCREASE CYCLE TIME IN THIS MODE
     TCMMR(K,MODEMR(K))=TCMMR(K,MODEMR(K))+DT
C
     IF (MTYPER(K).NE.1) THEN
C     ALL OF THESE MOVEMENT TYPES SIMPLY INVOLVE CONTINUING IN
C     CURRENT DIRECTION UNTIL OBJECTIVE IS REACHED
      AZIN=IRSA(K,2)
      CALL FDIR (AZIN,AZOUT)
C     CHECK CURRENT DISTANCE TO OBJECTIVE
     ROBJ1=((RXYA(K,1)-RXYO(K,1))**2+(RXYA(K,2)-RXYO(K,2))**2)**.5
C     MOVE UNIT
     DELPOS=(ISPDR(K)*DT+ARATE/2.*DT*DT)/3600.
CI
C     before moving, evaluate minefields
           IF (NMINES.GE.1.AND.DELPOS.GT.0.0) THEN
           CALL MINEFIELD (2,K,DELPOS,AZOUT,INNOW,INNEXT)
            IF (INNOW.NE.0.OR.INNEXT.NE.0) THEN
C       IN OR ENTERING A MINEFIELD, DOES HE KNOW IT?
             INKNOW=0
             NXKNOW=0
             IF (INNOW.NE.0.AND.NAZMINE(INNOW,3).EQ.1) INKNOW=1
             IF (INNEXT.NE.0.AND.NAZMINE(INNEXT,3).EQ.1) NXKNOW=1
             IF (INKNOW.EQ.1.OR.NXKNOW.EQ.1) THEN
              IF (MBREECH(K,2).EQ.0) THEN
C       STOP AND AWAIT ORDERS, YOU'RE IN OR GOING INTO A KNOWN MINEFIELD
               DELPOS=0.0
               ARATE=0.0
               ISPDR(K)=0
               MODEMR(K)=3
               IRSA(K,1)=0
              ENDIF
              IF (MBREECH(K,2).EQ.-1)
     &            CALL MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)
             ELSE
C       YOU'RE MOVING THROUGH A MINEFIELD AND DON'T KNOW IT
C       EVALUATE PROBABILITY OF BEING KILLED
              IF (MBREECH(K,2).NE.1)
     &            CALL MINEKILL (2,K,INNOW,INNEXT,DELPOS)
             ENDIF
            ENDIF
           ENDIF
CI
           RXYA(K,1)=RXYA(K,1)+DELPOS*COS(AZOUT*PI/180.)
     RXYA(K,2)=RXYA(K,2)+DELPOS*SIN(AZOUT*PI/180.)
C     CHANGE SPEED
     ISPDR(K)=ISPDR(K)+ARATE*DT
C     CHECK NEW DISTANCE TO OBJECTIVE
     ROBJ2=((RXYA(K,1)-RXYO(K,1))**2+(RXYA(K,2)-RXYO(K,2))**2)**.5
C     IF THIS DISTANCE HAS GOTTEN BIGGER, THEN OBJECTIVE HAS JUST
C     BEEN PASSED, STOP VEHICLE
          IF (ROBJ2.GE.ROBJ1) THEN
            IRSA(K,1)=0
            MODEMR(K)=3
          ENDIF
     ELSE
C     FOLLOW-ME MOVEMENT TYPE, CHECK TO SEE IF PIVOT POINT IS
C     NOW REACHED.  IF REACHED, CHANGE MOVEMENT TYPE AND AZIMUTH
C     AND OBJECTIVE
      AZIN=IRSA(K,2)
      CALL FDIR (AZIN,AZOUT)
C     CHECK CURRENT DISTANCE TO PIVOT POINT
     ROBJ1=((RXYA(K,1)-RXYP(K,1))**2+(RXYA(K,2)-RXYP(K,2))**2)**.5
C     MOVE UNIT
CI
C     before moving, evaluate minefields
      IF (NMINES.GE.1.AND.DELPOS.GT.0.0) THEN
      CALL MINEFIELD (2,K,DELPOS,AZOUT,INNOW,INNEXT)
       IF (INNOW.NE.0.OR.INNEXT.NE.0) THEN
C       IN OR ENTERING A MINEFIELD, DOES HE KNOW IT?
        INKNOW=0
        NXKNOW=0
        IF (INNOW.NE.0.AND.NAZMINE(INNOW,3).EQ.1) INKNOW=1
        IF (INNEXT.NE.0.AND.NAZMINE(INNEXT,3).EQ.1) NXKNOW=1
        IF (INKNOW.EQ.1.OR.NXKNOW.EQ.1) THEN
         IF (MBREECH(K,2).EQ.0) THEN
C       STOP AND AWAIT ORDERS, YOU'RE IN OR GOING INTO A KNOWN MINEFIELD
          DELPOS=0.0
          ARATE=0.0
          ISPDR(K)=0
          MODEMR(K)=3
          IRSA(K,1)=0
         ENDIF
         IF (MBREECH(K,2).EQ.-1)
     &       CALL MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)
        ELSE
C       YOU'RE MOVING THROUGH A MINEFIELD AND DON'T KNOW IT
C       EVALUATE PROBABILITY OF BEING KILLED
         IF (MBREECH(K,2).NE.1)
     &       CALL MINEKILL (2,K,INNOW,INNEXT,DELPOS)
        ENDIF
       ENDIF
      ENDIF
CI
      DELPOS=(ISPDR(K)*DT+ARATE/2.*DT*DT)/3600.
       RXYA(K,1)=RXYA(K,1)+DELPOS*COS(AZOUT*PI/180.)
       RXYA(K,2)=RXYA(K,2)+DELPOS*SIN(AZOUT*PI/180.)
C     CHANGE SPEED
       ISPDR(K)=ISPDR(K)+ARATE*DT
C     CHECK NEW DISTANCE TO PIVOT POINT
     ROBJ2=((RXYA(K,1)-RXYP(K,1))**2+(RXYA(K,2)-RXYP(K,2))**2)**.5
C     IF THIS DISTANCE HAS GOTTEN BIGGER, THEN PIVOT POINT HAS JUST
C     BEEN PASSED, TURN VEHICLE
      IF (ROBJ2.GE.ROBJ1) THEN
          MTYPER(K)=0
          X2=RXYO(K,1)
          Y2=RXYO(K,2)
          X1=RXYA(K,1)
          Y1=RXYA(K,2)
          TOP=Y2-Y1
          BOT=X2-X1
          IF (BOT.EQ.0.0) BOT=BOT+.001
          DIR=ATAN(ABS(TOP/ABS(BOT))*180./PI
               IF (BOT.LT.0.0) THEN
                    IF (TOP.LT.0.0) DIR=DIR+180.
                    IF (TOP.GT.0.0) DIR=180.-DIR
               ELSE
                    IF (TOP.LT.0.0) DIR=360.-DIR
               ENDIF
          AZOUT=DIR
          CALL FAZ (AZOUT,AZIN)
          IAZ=INT(AZIN)
          IRSA(K,2)=IAZ
      ENDIF
      ENDIF
C     SET TURRET POSITION IF STILL IN SURVEILLANCE MODE
  IF (MODEFR(K).EQ.7) AZTR(K)=IRSA(K,2)
140   CONTINUE
C
CC    EXECUTE DIRECT-FIRE ORDERS
CC
CC    ALL ACTIVITY OCCURS AT THE END OF THE MODE TIME
CC    THAT WAY WASTED TIME AND WASTED ENGAGEMENTS HAVE
CC    AN EFFECT ON THE BATTLE
CC
CC    BLUE
C
      DO 150 K=1,NB
                                   LOOP 1
C     CHECK TO SEE IF THIS UNIT IS KILLED AND SKIP
CC     WRITE (*,*) 'KILL STATUS BLUE ',K,' ALIVE'
CC     BUT BEFORE SKIPPING IT, SEE IF IT WAS A FIRE-AND-FORGET
CC     WEAPON AND IN THE LAST SECONDS OF ENGAGMENT MODE OR
CC     RE-ENGAGEMENT MODE.
      MODEF=MODEFB(K)
      IF (IBSYS(K,2).EQ.0) THEN
       NSYS=IBSYS(K,1)
        IF (MODEF.EQ.2.OR.MODEF.EQ.6) THEN
         NWP=ITGTB(K,2)
         MILTYPE=MILB(NSYS,NWP)
         IF (MILTYPE.EQ.0) THEN
C         CALCULATE TIME TO BULLET IMPACT
          TCHANGE=TRMFB(K)
          BORE=BOREB(NSYS,NWP)/1000.
          TOF=TOFB(NSYS,NWP)
          TRNG=PKTB(K,2)
        TTRNG=TRNG/BORE*TOF+1.0
         IF (ITIME.GE.(TCHANGE-TTRNG)) THEN
C          THIS WEAPON IS NOW IN SACRIFICE MODE SINCE IT IS DEAD AND
C          HAPPENED TO GET A FIRE-AND-FORGET SHOT OFF FIRST
```

```
            MODEF=9
                  MODEFB(K)=MODEF
                  NSHOTB(K,NWP)=NSHOTB(K,NWP)-1
                  CALL OUTPUT2 (ITIME,1,K,0)
C           WRITE (*,*) ' BLUE ',K,' MAKES SACRIFICE SHOT'
            WRITE (10,*) ' BLUE ',K,' MAKES SACRIFICE SHOT'
           ELSE
            MODEFB(K)=0
C           FIRE MODE SET TO 0 -- KILLED WHILE ENGAGING A TARGET
           ENDIF
          ENDIF
         ENDIF
            ENDIF

      IF (IBSYS(K,2).EQ.0.AND.MODEF.NE.9) GO TO 150
C
            IF (MODEF.EQ.-9) THEN
C       RE-EVALUATE RE-ENGAGEMENT MODE TO SEE IF THIS RED TARGET
C       WAS JUST KILLED
                  IF (IBSYS(ITGTB(K,1),2).EQ.1) THEN
                  MODEF=9
                  MODEFB(K)=9
                  WRITE (10,*) 'BLUE ',K,' RE-ENGAGES RED ',ITGTB(K,1)
                 ELSE
C       THIS TARGET WAS KILLED LAST GO AROUND SO FIND ANOTHER
                  MODEF=1
                  MODEFB(K)=1
                  NSYS=IBSYS(K,1)
                  ALFT1=IBPRIME(K,1)
                  ARGT1=IBPRIME(K,2)
                  IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
                  ALFT2=IBSECOND(K,1)
                  ARGT2=IBSECOND(K,2)
                  IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
                  IF (IBPRIME(K,1).EQ.IBSECOND(K,1).AND.IBPRIME(K,2).
     &      EQ.IBSECOND(K,2)) THEN
                  ALFT2=-1.0
                  ARGT2=-1.0
                  ENDIF
C       DETERMINE END TIME OF SEARCH MODE CYCLE
                  STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHB(NSYS)
                  TRMFB(K)=ITIME+STIME
                 ENDIF
                ENDIF

C       WRITE (*,*) 'FIRE MODE BLUE ',MODEF
C       INCREASE CYCLE TIME IN THIS MODE
        TCFMB(K,MODEF)=TCFMB(K,MODEF)+DT
C       WRITE (*,*) 'CYCLE TIME BLUE ',TCFMB(K,MODEF)
C
        IF (MODEF.EQ.1.OR.MODEF.EQ.4.OR.MODEF.EQ.5.OR.MODEF.EQ.3) THEN
C                                       LOOP 2
CCC     SEARCH MODE (1) OR BALK MODE (4) OR WAIT MODE (5)
CCC     EM ENERGY WAIT MODE (3)
CCC     BALK MODE IS ACTUALLY RE-SEARCHING, BUT IS BROKEN OUT
CCC     FOR DUTY CYCLE ACCOUNTABILITY
CCC     WAIT MODE IS ACTUALLY RE-SEARCHING, BUT IS BROKEN OUT
CCC     FOR DUTY CYCLE ACCOUNTABILITY
CCC     EM ENERGY WAIT MODE IS ACTUALLY RE-SEARCHING, BUT IS BROKEN OUT
CCC     FOR DUTY CYCLE ACCOUNTABILITY
CCC
C       CHECK FOR MODE CHANGE AT END OF SUB-CYCLE
        TCHANGE=TRMFB(K)
C       WRITE (*,*) 'FIRE MODE CHANGE BLUE ',TCHANGE
        TNEXT=ITIME+IDTIME
        IF (TCHANGE.LE.TNEXT) THEN
                                LOOP 3
CCC     IT'S TIME TO FIND A TARGET
CCC
CCC     FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
CCC     SO THAT MASKING CAN BE EVALUATED
CCC
        KSIDE=1
        MYSIDE=1
        IBLUE=K
        CALL VMASK (KSIDE,IBLUE)
C
CCC     FIND ALL RED TARGETS IN PRIMARY AND SECONDARY ZONES
CCC     OUT TO MAX RANGE FOR EACH ON-BOARD WEAPON
        NSYS=IBSYS(K,1)
        NWEAP=NWB(NSYS)
        NRNG1=NRWB(NSYS,1)
        RMAX1=0.0
        RMAX2=0.0
        RMAX3=0.0
        NUM1=0
        NUM2=0
        NUM3=0
C       ISR=(I-1)*18+(J-1)*6+K
CCC     WHERE I=SYSTEM FILE (10 MAX); J=SYSTEM WEAPON (3 MAX);
CCC           K=WEAPON RANGE (6 MAX); N=TARGET (2*10 MAX)
        NUM1=(NSYS-1)*18+(1-1)*6+NRNG1
        RMAX1=RB(NUM1)
C       WRITE (*,*) 'RMAX1',RMAX1,'RB(22)',RB(22)
C       WRITE (*,*) 'NSYS',NSYS,'NWEAP',NWEAP,'NRNG1',NRNG1,'NUM1',NUM1
C       WRITE (*,*) 'NSHOTB',NSHOTB(K,1)
            IF (NSHOTB(K,1).LE.0) THEN
            RMAX1=0.0
            NUM1=0
            ENDIF
        IF (NWEAP.GE.2) THEN
        NRNG2=NRWB(NSYS,2)
        NUM2=(NSYS-1)*18+(2-1)*6+NRNG2
        RMAX2=RB(NUM2)
C       WRITE (*,*) 'NSYS',NSYS,'NWEAP',NWEAP,'NRNG2',NRNG2,'NUM2',NUM2
C       WRITE (*,*) 'NSHOTB',NSHOTB(K,2)
            IF (NSHOTB(K,2).LE.0) THEN
            RMAX2=0.0
            NUM2=0
            ENDIF
        ENDIF
        IF (NWEAP.EQ.3) THEN
        NRNG3=NRWB(NSYS,3)
        NUM3=(NSYS-1)*18+(3-1)*6+NRNG3
        RMAX3=RB(NUM3)
C       WRITE (*,*) 'NSYS',NSYS,'NWEAP',NWEAP,'NRNG3',NRNG3,'NUM3',NUM3
C       WRITE (*,*) 'NSHOTB',NSHOTB(K,3)
            IF (NSHOTB(K,3).LE.0) THEN
            RMAX3=0.0
            NUM3=0
            ENDIF
        ENDIF
C
C       SEARCH ALL ZONES BASED ON ZONE AZIMUTHS
        IOVER1=0
        IOVER2=0
          ALFT1=IBPRIME(K,1)
          ARGT1=IBPRIME(K,2)
        IF (ALFT1.GT.ARGT1) IOVER1=1
        IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
        ALFT2=IBSECOND(K,1)
          ARGT2=IBSECOND(K,2)
        IF (ALFT2.GT.ARGT2) IOVER2=1
        IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
        IF (IBPRIME(K,1).EQ.IBSECOND(K,1).AND.IBPRIME(K,2).
        &       EQ.IBSECOND(K,2)) THEN
          ALFT2=-1.0
          ARGT2=1.0
          ENDIF
C       GET X,Y COORDINATES FOR THIS BLUE UNIT
        XB=BXYA(K,1)
        YB=BXYA(K,2)
C       INITIALIZE PRIORITIES
        NTGT1=0
        PRI=0.0
        MYWP1=0
        NTGT2=0
        SEC=0.0
        MYWP2=0
        NTGT3=0
        THIRD=0.0
        MYWP3=0
        TRNG1=0.0
        TRNG2=0.0
        TRNG3=0.0
        PKWP1=0.0
        PKWP2=0.0
        PKWP3=0.0
C
CCC     IF THIS BLUE SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
CCC     TRACKING OR RANGING,
CCC     FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C       EVALUATED
        ITRK1=ITRKB(NSYS,1)
        ISROF=(NSYS-1)*3+1
        IRNG11=1
        IRNG12=1
        IRNG13=1
        IRNG11=IRNGB(ISROF,1)
          IF (NWEAP.GE.2) IRNG12=IRNGB((ISROF+1),1)
          IF (NWEAP.GT.2) IRNG13=IRNGB((ISROF+2),1)
        ICANRD=1
        IF (IRNG11.EQ.0.OR.IRNG12.EQ.0.OR.IRNG13.EQ.0) ICANRD=0
          IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
      & CALL DMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C
CCC     FIND ANGLE AND RANGE TO ALL BURNING VEHICLES, SINCE THEY COMPLETELY
C       MASK VISUAL AND IR SYSTEMS
        CALL COOKMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C
CCC     IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
CCC     TRACKING OR RANGING,
CCC     FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C       CAN BE EVALUATED
        ITRK2=ITRKB(NSYS,2)
        ISROF=(NSYS-1)*3+1
        IRNG21=1
        IRNG22=1
        IRNG23=1
        IRNG21=IRNGB(ISROF,2)
          IF (NWEAP.GE.2) IRNG22=IRNGB((ISROF+1),2)
          IF (NWEAP.GT.2) IRNG23=IRNGB((ISROF+2),2)
        ICANRE=1
          IF (IRNG21.EQ.0.OR.IRNG22.EQ.0.OR.IRNG23.EQ.0) ICANRE=0
          IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
      & CALL ESMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C
CCC     IF THIS BLUE SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
CCC     TRACKING OR RANGING,
CCC     FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C       CAN BE EVALUATED
        ITRK3=ITRKB(NSYS,3)
        ISROF=(NSYS-1)*3+1
```

```
      IRNG31=1
      IRNG32=1
      IRNG33=1
      IRNG31=IRNGB(ISROF,3)
      IF (NWEAP.GE.2) IRNG32=IRNGB((ISROF+1),3)
      IF (NWEAP.GT.2) IRNG33=IRNGB((ISROF+2),3)
      ICANRS=1
      IF (IRNG31.EQ.0.OR.IRNG32.EQ.0.OR.IRNG33.EQ.0) ICANRS=0
      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
     & CALL GRMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C
C     IF THIS BLUE SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C     TRACKING OR RANGING,
C     FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
MASKING
C     CAN BE EVALUATED
            ITRK4=ITRKB(NSYS,4)
            ISROF=(NSYS-1)*3+1
            IRNG41=1
            IRNG42=1
            IRNG43=1
            IRNG41=IRNGB(ISROF,4)
            IF (NWEAP.GE.2) IRNG42=IRNGB((ISROF+1),4)
            IF (NWEAP.GT.2) IRNG43=IRNGB((ISROF+2),4)
            ICANRA=1
            IF (IRNG41.EQ.0.OR.IRNG42.EQ.0.OR.IRNG43.EQ.0) ICANRA=0
            IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
     & CALL ASMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C
            IFIND=0
      DO 149 J=1,NR
C                                   LOOP 4
C     CHECK TO SEE IF THIS RED UNIT IS KILLED
            IF (IRSYS(J,2).EQ.0.OR.IRSYS(J,1).GT.10) GO TO 149
C     FIND THIS RED UNIT'S X,Y COORDINATES
      XR=RXY(J,1)
      YR=RXY(J,2)
C     FIND RANGE TO THIS TARGET
      RANGE=((XR-XB)**2+(YR-YB)**2)**.5
C       WRITE (*,*) 'RANGE TO RED ',J,RANGE

C     CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
C     NOT DETECTED
      CALL CHECKSIG (1,K,J,RANGE,JSKIP)
       IF (JSKIP.EQ.0) GO TO 149

C     IF THIS TARGET IS OUTSIDE THE RANGE OF ALL WEAPONS SKIP IT
C      WRITE (*,*) 'RANGE',RANGE,'RMAX1',RMAX1,'RMAX2',RMAX2,
C     &'RMAX3',RMAX3
      JSKIP=0
            IF ((RANGE*1000.).LE.RMAX1) JSKIP=1
            IF ((RANGE*1000.).LE.RMAX2) JSKIP=1
            IF ((RANGE*1000.).LE.RMAX3) JSKIP=1
C       WRITE (*,*) 'RANGE SKIP ',JSKIP,' 0=YES'
      IF (JSKIP.EQ.0) GO TO 149
C     TARGET IS WITHIN RANGE OF ONE OF THESE WEAPONS
C     IF THIS TARGET IS OUTSIDE THE FIRE ORDERS RANGE WINDOW SKIP IT
      IF (RANGE.LT.WRNGB(K,1).OR.RANGE.GT.WRNGB(K,2)) JSKIP=0
      IF (JSKIP.EQ.0) GO TO 149
C     FIND ANGLE TO THIS TARGET
      TOP=YR-YB
      BOT=XR-XB
       IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
      IF (BOT.LT.0.0) THEN
            IF (TOP.LT.0.0) DIR=DIR+180.
            IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
            IF (TOP.LT.0.0) DIR=360.-DIR
      ENDIF
C     CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
      CALL FAZ (DIR,AZIN)
C
C     AT THIS POINT EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE
C     MASKING AND OBSCURANTS AND DROP THIS
C     TARGET FROM ZONE IF UNOBSERVABLE
C
C     TERRAIN MASKING
      IF (ITER.EQ.1) CALL TERRMASK (XB,YB,XR,YR,JSKIP)
       IF (JSKIP.EQ.0) GO TO 149
C
C     VEHICLE MASKING
      CALL VSEE (MYSIDE,K,J,RANGE,AZIN,JSKIP,NB,NR)
C
C     BURNING VEHICLE MASKING
            CALL COOKSEE (XB,YB,XR,YR,AZIN,JSKIP)

            IF (JSKIP.EQ.0) GO TO 149
C
C     DUST MASKING
C     IF THIS BLUE SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET
      JDUST=1
      IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
     & CALL DSEE (XB,YB,XR,YR,AZIN,JDUST)
C     IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE THROUGH IT
C     SKIP THIS TARGET
      IF (JDUST.EQ.0.AND.ITRK1.EQ.0) GO TO 149
C     THERE MAY BE DUST IN THE WAY, BUT BLUE CAN SEE THROUGH IT.
C     THE QUESTION REMAINS WHETHER BLUE CAN RANGE OR TRACK THROUGH
C     THE DUST.  IF JDUST=0 THERE IS DUST IN THE WAY.
C
C     ENGINE SMOKE MASKING
C     IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
      JENG=1
      IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
     & CALL ESSEE (XB,YB,XR,YR,AZIN,JENG)
      IF (JENG.EQ.0.AND.ITRK2.EQ.0) GO TO 149
C
C     SMOKE GRENADE CURTAIN MASKING
C     IF THIS BLUE SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     EVALUATE ALL SMOKE GRENADE CUHTAINS WITHIN RANGE OF THIS TARGET
      JSMK=1
      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
     & CALL GRSEE (XB,YB,XR,YR,AZIN,JSMK)
      IF (JSMK.EQ.0.AND.ITRK3.EQ.0) GO TO 149
C
C     ARTILLERY SMOKE MASKING
C     IF THIS BLUE SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C     TRACKING OR RANGING,
C     EVALUATE ALL ARTILLERY SMOKE PUFFS WITHIN RANGE OF THIS TARGET
      JART=1
      IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
     & CALL ASSEE (XB,YB,XR,YR,AZIN,JART)
      IF (JART.EQ.0.AND.ITRK4.EQ.0) GO TO 149
C
      INZONE=3
      IF (AZIN.GE.ALFT1.AND.AZIN.LE.ARGT1) INZONE=1
      IF (IOVER1.EQ.1.AND.AZIN.LE.(ARGT1-360.0)) INZONE=1
C       TARGET IS IN PRIMARY ZONE
      IF (AZIN.GE.ALFT2.AND.AZIN.LE.ARGT2) INZONE=2
      IF (IOVER2.EQ.1.AND.AZIN.LE.(ARGT2-360.0)) INZONE=2
C       TARGET IS IN SECONDARY ZONE
C
C       WRITE (*,*) 'RED ',J,' INZONE ? ',INZONE
      IF (INZONE.GE.1) THEN
C                                   LOOP 5
C     GIVE THIS TARGET A PRIORITY NUMBER
C     AND IN THE END SELECT ONLY THE HIGHEST PRIORITY TARGET
C      TARGET PRIORITY IS A COMPOSITE OF THE PROBABILITY OF THIS
C      TARGET KILLING YOU TIMES YOU'RE PROBABILITY OF KILLING IT.
      NITSYS=IRSYS(J,1)
C     CYCLE THROUGH ALL WEAPONS ON YOUR SYSTEM FOR THE ONE WITH THE
C     HIGHEST PK AGAINST THIS TARGET AND EVALUATE WHETHER YOU WILL
C     SHOOT IT BASED ON NUMBER OF AVAILABLE ROUNDS OF AMMUNITION
C     THE PK AGAINST THIS TARGET, AND THE GENERAL THREAT PICTURE
      PKIT1=0.0
      PKIT2=0.0
      PKIT3=0.0
      KSIDE=1
      PKITM=0.0
      IWEAP=0

      IFIND=IFIND+1
      IF (IFIND.EQ.1) THEN

C     BLUE HAS FOUND ITS FIRST POTENTIAL TARGET

C     EVALUATE EM ENERGY AVAILABILITY FOR THIS BLUE SYSTEM
C     (POTENTIAL MODE 3)
C
      IF (NSYS.EQ.NBLUE) THEN
       EL1=1.0
       EL2=1.0
       EL3=1.0
       ILOST1=0
       ILOST2=0
       ILOST3=0
       IF (NEMW.GE.1) THEN
       EL1=ELSYS(K)/EWP(1,1)
       IF (EL1.LT.EWP(1,2)) EL1=0.0
       IF (EL1.LT.1.0) THEN
        WRITE (10,*) ' BLUE ',K,' WEAPON ',1,' INSUFFICIENT ENERGY
     &FOR FULL EM SHOT'
       ELSE
        EL1=1.0
       ENDIF
       ENDIF
       IF (NEMW.GE.2) THEN
       EL2=ELSYS(K)/EWP(2,1)
       IF (EL2.LT.EWP(2,2)) EL2=0.0
       IF (EL2.LT.1.0) THEN
        WRITE (10,*) ' BLUE ',K,' WEAPON ',2,' INSUFFICIENT ENERGY
     &FOR FULL EM SHOT'
       ELSE
        EL2=1.0
       ENDIF
       ENDIF
       IF (NEMW.EQ.3) THEN
       EL3=ELSYS(K)/EWP(3,1)
       IF (EL3.LT.EWP(3,2)) EL3=0.0
       IF (EL3.LT.1.0) THEN
        WRITE (10,*) ' BLUE ',K,' WEAPON ',3,' INSUFFICIENT ENERGY
     &FOR FULL EM SHOT'
       ELSE
        EL3=1.0
       ENDIF
       ENDIF
```

```
      ENDIF
      ENDIF
C     END OF EM ENERGY EVALUATION

      DO 141 NUM=1,3
C                                   LOOP 6
      IF (NUM.EQ.1.AND.NUM1.EQ.0) GO TO 141
      IF (NUM.EQ.2.AND.NUM2.EQ.0) GO TO 141
      IF (NUM.EQ.3.AND.NUM3.EQ.0) GO TO 141
      IF (NUM.EQ.1) THEN
      CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG1,RANGE,PKIT1,J)
            IF (ISPDB(K).GT.0.0.OR.ISPDR(J).GT.0) THEN
             MILTYPE=MILB(NSYS,NUM)
             IF (MILTYPE.EQ.0) THEN
              BORE=BOREB(NSYS,NUM)
              TOF=TOFB(NSYS,NUM)
              CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT1)
             ENDIF
            ENDIF
            IF (JDUST.EQ.0.0.OR.JENG.EQ.0) THEN
C     EVALUATE DUST EFFECTS ON WEAPON RANGING AND TRACKING
C     IF THIS WEAPON IS F-A-F THEN DUST DEGRADES HIT PROBABILITY
C     IF THIS WEAPON CANNOT RANGE THROUGH THE DUST
C     IF THIS WEAPON IS M-I-L THEN DUST DEGRADES TRACKING
C     IF THIS WEAPON CANNOT TRACK THROUGH THE DUST
      IF (IRNG11.EQ.0.OR.IRNG21.EQ.0) THEN
C      THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
C      EVALUATE M-I-L OR F-A-F
       MILTYPE=MILB(NSYS,NUM)
        IF (MILTYPE.EQ.0) THEN
C        THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
         BORE=BOREB(NSYS,NUM)
         TOF=TOFB(NSYS,NUM)
         CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT1)
        ELSE
C        SKIP THIS WEAPON SINCE CAN'T TRACK THE TARGET
         PKIT1=0.0
        ENDIF
       ENDIF
      ENDIF
      NSHOTS=NSHOTB(K,NUM)
C      WRITE (*,*) 'LOG BOT1',LOG(1.-PKIT1),'PKIT1',PKIT1
       IF (PKIT1.NE.0.0) THEN
        IF (NSYS.EQ.NBLUE) PKIT1=PKIT1*EL1
        IF (PKIT1.NE.0.0) THEN
         SKILL=LOG(.01)/LOG(1.-PKIT1)
        ELSE
         SKILL=9999.
         ILOST1=1
        ENDIF
       ELSE
        SKILL=9999.
       ENDIF
            IF (NSHOTS.EQ.0) PKIT1=0.0
C            IF (SKILL.GT.16.0) PKIT1=0.0
C           IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
C            IF (SKILL.GT.7.0) THEN
            IF (SKILL.GT.45.0) THEN
                 PSHOOT=NSHOTS/SKILL
                PNO=RND()
                 IF (PNO.GE.PSHOOT) PKIT1=0.0
            ENDIF
            IF (PKIT1.GT.PKITM) THEN
                 PKITM=PKIT1
                 IWEAP=1
            ENDIF
      ENDIF
      IF (NUM.EQ.2) THEN
      CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG2,RANGE,PKIT2,J)
            IF (ISPDB(K).GT.0.0.OR.ISPDR(J).GT.0) THEN
             MILTYPE=MILB(NSYS,NUM)
             IF (MILTYPE.EQ.0) THEN
              BORE=BOREB(NSYS,NUM)
              TOF=TOFB(NSYS,NUM)
              CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT2)
             ENDIF
            ENDIF
            IF (JDUST.EQ.0.0.OR.JENG.EQ.0) THEN
       IF (IRNG12.EQ.0.OR.IRNG22.EQ.0) THEN
C      THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
C      EVALUATE M-I-L OR F-A-F
       MILTYPE=MILB(NSYS,NUM)
        IF (MILTYPE.EQ.0) THEN
C        THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
         BORE=BOREB(NSYS,NUM)
         TOF=TOFB(NSYS,NUM)
         CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT2)
        ELSE
C        SKIP THIS WEAPON SINCE CAN'T TRACK THE TARGET
         PKIT2=0.0
        ENDIF
       ENDIF
      ENDIF
      NSHOTS=NSHOTB(K,NUM)
C      WRITE (*,*) 'LOG BOT2',LOG(1.-PKIT2),'PKIT2',PKIT2
       IF (PKIT2.NE.0.0) THEN
        IF (NSYS.EQ.NBLUE) PKIT2=PKIT2*EL2
        IF (PKIT2.NE.0.0) THEN
         SKILL=LOG(.01)/LOG(1.-PKIT2)
        ELSE
         SKILL=9999.
         ILOST2=1
        ENDIF
       ELSE
        SKILL=9999.
       ENDIF
            IF (NSHOTS.EQ.0) PKIT2=0.0
C            IF (SKILL.GT.16.0) PKIT2=0.0
C           IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
            IF (SKILL.GT.45.0) THEN
                 PSHOOT=NSHOTS/SKILL
                PNO=RND()
                 IF (PNO.GE.PSHOOT) PKIT2=0.0
            ENDIF
            IF (PKIT2.GT.PKITM) THEN
                 PKITM=PKIT2
                 IWEAP=2
            ENDIF
      ENDIF
      IF (NUM.EQ.3) THEN
      CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG3,RANGE,PKIT3,J)
      IF (ISPDB(K).GT.0.0.OR.ISPDR(J).GT.0) THEN
       MILTYPE=MILB(NSYS,NUM)
       IF (MILTYPE.EQ.0) THEN
       BORE=BOREB(NSYS,NUM)
       TOF=TOFB(NSYS,NUM)
       CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT3)
      ENDIF
      ENDIF
       IF (JDUST.EQ.0.0.OR.JENG.EQ.0) THEN
        IF (IRNG13.EQ.0.0.OR.IRNG23.EQ.0) THEN
C        THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
C        EVALUATE M-I-L OR F-A-F
         MILTYPE=MILB(NSYS,NUM)
          IF (MILTYPE.EQ.0) THEN
C          THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
           BORE=BOREB(NSYS,NUM)
           TOF=TOFB(NSYS,NUM)
           CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT3)
          ELSE
C          SKIP THIS WEAPON SINCE CAN'T TRACK THE TARGET
           PKIT3=0.0
          ENDIF
         ENDIF
        ENDIF
        NSHOTS=NSHOTB(K,NUM)
C        WRITE (*,*) 'LOG BOT3',LOG(1.-PKIT3),'PKIT3',PKIT3
         IF (PKIT3.NE.0.0) THEN
          IF (NSYS.EQ.NBLUE) PKIT3=PKIT3*EL3
          IF (PKIT3.NE.0.0) THEN
           SKILL=LOG(.01)/LOG(1.-PKIT3)
          ELSE
           SKILL=9999.
           ILOST3=1
          ENDIF
         ELSE
          SKILL=9999.
         ENDIF
              IF (NSHOTS.EQ.0) PKIT3=0.0
C              IF (SKILL.GT.16.0) PKIT3=0.0
C             IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
C              IF (SKILL.GT.7.0) THEN
              IF (SKILL.GT.45.0) THEN
                   PSHOOT=NSHOTS/SKILL
                  PNO=RND()
                   IF (PNO.GE.PSHOOT) PKIT3=0.0
              ENDIF
              IF (PKIT3.GT.PKITM) THEN
                   PKITM=PKIT3
                   IWEAP=3
              ENDIF
        ENDIF

141   CONTINUE
C                                   END LOOP 6
C     END OF BLUE WEAPON SELECTION LOOP
C      WRITE (*,*) 'BLUE WEAPON SELECTION',IWEAP,'PK',PKITM

C     IF TARGET IS AT A QUESTIONABLE EFFECTIVE RANGE SKIP IT
      IF (PKITM.LE.0.0) GO TO 143
      IF (IWEAP.EQ.0.0) GO TO 143

C     FIND PROBABILITY OF THIS TARGET KILLING YOU AND APPLY PRIORITY
C     FACTOR
C
      KPFACT=KPRIB(K,NITSYS)
      PKYOU1=0.0
      PKYOU2=0.0
      PKYOU3=0.0
      KSIDE=2
      NUMIT=NWR(NITSYS)
      IRNG1=NRWR(NITSYS,1)
      IF (NUMIT.GE.2) IRNG2=NRWR(NITSYS,2)
      IF (NUMIT.EQ.3) IRNG3=NRWR(NITSYS,3)
      DO 142 NUM=1,NUMIT
      IF (NUM.EQ.1) CALL
     &      FINDPK (KSIDE,NITSYS,NSYS,NUM,IRNG1,RANGE,PKYOU1,K)
      IF (NUM.EQ.2) CALL
     &      FINDPK (KSIDE,NITSYS,NSYS,NUM,IRNG2,RANGE,PKYOU2,K)
      IF (NUM.EQ.3) CALL
     &      FINDPK (KSIDE,NITSYS,NSYS,NUM,IRNG3,RANGE,PKYOU3,K)
142   CONTINUE
C     FIND MAXIMUM PK AGAINST YOU
      PKYMAX=0.0
```

```
      IF (PKYOU1.GT.PKYMAX) PKYMAX=PKYOU1
      IF (PKYOU2.GT.PKYMAX) PKYMAX=PKYOU2
      IF (PKYOU3.GT.PKYMAX) PKYMAX=PKYOU3

C     IF TARGET'S PK ON YOU IS ZERO, GIVE IT A SMALL VALUE SO THAT
C     YOU AT LEAST HAVE A TARGET TO SHOOT AT IF NO OTHERS ARE BETTER
C     AFTER ALL THIS TROUBLE OF DECIDING IF YOU CAN KILL IT

      IF (PKYMAX.LE.0.0) PKYMAX=0.01

C     FIND COMPOSITE KILL PRIORITY

C     WRITE (*,*) 'PK IT ON YOU',PKYMAX

      PRIOR=PKITM*PKYMAX*KPFACT
C     ADD A SMALL RANDOM NUMBER SO NO TWO ARE EVER THE SAME
      SMALL=RND()
      SMALL=SMALL/25.
      PRIOR=PRIOR+SMALL
C
C     PRIORITIZE THIS TARGET WITHIN PRIMARY OR SECONDARY ZONE
C
      IF (INZONE.EQ.1) THEN
C     THIS TARGET IS IN PRIMARY ZONE
       IF (PRIOR.GT.PRI) THEN
          PRI=PRIOR
          NTGT1=J
          MYWP1=IWEAP
          TRNG1=RANGE
           IF (IWEAP.EQ.1) PKWP1=PKIT1
           IF (IWEAP.EQ.2) PKWP1=PKIT2
           IF (IWEAP.EQ.3) PKWP1=PKIT3
       ENDIF
      ENDIF
      IF (INZONE.EQ.2) THEN
C     THIS TARGET IS IN SECONDARY ZONE
       IF (PRIOR.GT.SEC) THEN
          SEC=PRIOR
          NTGT2=J
          MYWP2=IWEAP
          TRNG2=RANGE
           IF (IWEAP.EQ.1) PKWP2=PKIT1
           IF (IWEAP.EQ.2) PKWP2=PKIT2
           IF (IWEAP.EQ.3) PKWP2=PKIT3
       ENDIF
      ENDIF
           IF (INZONE.EQ.3) THEN
C     THIS TARGET IS IN 360 DEGREE ZONE
       IF (PRIOR.GT.THIRD) THEN
          THIRD=PRIOR
          NTGT3=J
          MYWP3=IWEAP
          TRNG3=RANGE
           IF (IWEAP.EQ.1) PKWP3=PKIT1
           IF (IWEAP.EQ.2) PKWP3=PKIT2
           IF (IWEAP.EQ.3) PKWP3=PKIT3
       ENDIF
      ENDIF

143   CONTINUE
C     TARGET WAS SKIPPED TO LINE 143 SINCE IT WAS AT A QUESTIONABLE
C     EFFECTIVE RANGE OR INSUFFICIENT AMMO TO RISK MISSING THE SHOT

C     WRITE (*,*) 'PRI',PRI,'NTGT1',NTGT1

      ENDIF
C                                 END LOOP 5
C     END OF INZONE

149   CONTINUE
C                                 END LOOP 4
C     END OF FINDING A TARGET FOR THIS BLUE SYSTEM
C
C     IF THE SECONDARY TARGET HAS 125% OF THE PRIORITY
C     OF THE PRIMARY TARGET, SHOOT AT THE SECONDARY TARGET
      ITGTB(K,1)=0
      ITGTB(K,2)=0
      PKTB(K,1)=0.0
      PKTB(K,2)=0.0
      TRNG=0.0
      MYWP=0
      IF (SEC.GE.(1.25*PRI)) THEN
C     SHOOT AT TARGET IN SECONDARY ZONE SINCE IT IS COMPELLING
      ITGTB(K,1)=NTGT2
      ITGTB(K,2)=MYWP2
      PKTB(K,1)=PKWP2
      PKTB(K,2)=TRNG2
      TRNG=TRNG2
      MYWP=MYWP2
      ELSE
C     STAY WITH PRIMARY ZONE TARGET
      ITGTB(K,1)=NTGT1
      ITGTB(K,2)=MYWP1
      PKTB(K,1)=PKWP1
      PKTB(K,2)=TRNG1
      TRNG=TRNG1
      MYWP=MYWP1
      ENDIF
C     IF THERE ARE NO TARGETS IN PRIMARY AND SECONDARY,
C     SHOOT AT THIRD ZONE TARGET (360 DEGREES)
      IN3=0
      IF (NTGT1.EQ.0.AND.NTGT2.EQ.0) THEN
       ITGTB(K,1)=NTGT3
       ITGTB(K,2)=MYWP3
       PKTB(K,1)=PKWP3
       PKTB(K,2)=TRNG3
       TRNG=TRNG3
       MYWP=MYWP3
       IF (NTGT3.NE.0) IN3=1
      ENDIF
C
C     FOR BATTLE MANAGEMENT, CHECK TO SEE IF MORE THAN 4 BLUE SYSTEMS
C     ARE ALREADY SERVICING THIS TARGET. IF SO, SKIP IT.
C
      IF (IAMR(ITGTB(K,1)).EQ.4) THEN
       ITGTB(K,1)=0
       ITGTB(K,2)=0
       PKTB(K,1)=0.0
       PKTB(K,2)=0.0
       TRNG=0.0
       MYWP=0
      ELSE
       IAMR(ITGTB(K,1))=IAMR(ITGTB(K,1))+1
      ENDIF
C

C     WRITE (*,*) 'PRIMARY ZONE TARGET',ITGTB(K,1)

C     EVALUATE TARGET SEARCH RESULTS AND
C     CHANGE MODE FOR THIS BLUE SYSTEM ACCORDINGLY
      STIME=0.0
      MODEN=0
      IF (ITGTB(K,1).EQ.0) THEN
C                                 LOOP 7
C     NO TARGETS, WAIT MODES  (EM ENERGY OR SIMPLY NO TARGETS)

       ILOSTEM=0
       IF (ILOST1.EQ.1.OR.ILOST2.EQ.1.OR.ILOST3.EQ.1) ILOSTEM=1

       IF (ILOSTEM.EQ.1.AND.NSYS.EQ.NBLUE) THEN

C      A TARGET WAS LOST DUE TO INSUFFICIENT EM ENERGY ALL AROUND
C      GO TO EM ENERGY WAIT MODE (MODE 3)
       WRITE (10,*) ' BLUE ',K,' INSUFFICIENT ENERGY TO FIRE ANY WPN'
C      WAIT A COMPLETE SEARCH CYCLE TIME FOR DUTY CYCLE ACCOUNTABILITY
C      BUT CREW IS STILL SEARCHING DURING THIS TIME
       MODEN=3
       MODEFB(K)=MODEN
C      DETERMINE END TIME OF ENERGY WAIT/SEARCH MODE CYCLE
       STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHB(NSYS)
        TRMFB(K)=ITIME+IDTIME+STIME

       ELSE

C      SIMPLY NO TARGETS WITHIN EFFECTIVE RANGE
C      WAIT A COMPLETE SEARCH CYCLE TIME FOR DUTY CYCLE ACCOUNTABILITY
C      BUT CREW IS STILL SEARCHING DURING THIS TIME
       MODEN=5
       MODEFB(K)=MODEN
C      DETERMINE END TIME OF SEARCH MODE CYCLE
       STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHB(NSYS)
        TRMFB(K)=ITIME+IDTIME+STIME
C       WRITE (*,*) 'BLUE ',K,' NO TARGETS '

       ENDIF

      ELSE

C      THERE IS A TARGET WITHIN EFFECTIVE RANGE

C     SET TURRET AZIMUTH
      CALL TURAZ (1,K,ITGTB(K,1),AZTB(K))

C     SINCE A TARGET WAS FOUND, CREW GOES INTO QUICK SEARCH
C     MODE FOR NEXT SEARCH CYCLE, FOCUSING INITIALLY ON TARGETS
C     WITHIN A .5 SECOND TRAVERSE FROM THIS TARGET.
C     RESET PRIMARY SEARCH ZONE AND SET SECONDARY TO 0,0
      NSYS=IBSYS(K,1)
      IBSECOND(K,1)=0
      IBSECOND(K,2)=0
      ALFT1=AZTB(K)-TRATEB(NSYS)/2.0
      ARGT1=AZTB(K)+TRATEB(NSYS)/2.0
      IF (ALFT1.LT.0.0) ALFT1=ALFT1+360.0
      IF (ARGT1.GT.360.0) ARGT1=ARGT1-360.0
      IBPRIME(K,1)=ALFT1
      IBPRIME(K,2)=ARGT1
      ALFT2=0.0
      ARGT2=0.0
      IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
      IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.

C     CHECK MODES FOR ENGAGEMENT, AND BALKING
       BALK=RND()
       IF (BALK.GT.0.975) THEN
C       BALKING, WAIT SEARCH TIME, BUT STILL SEARCH
        MODEN=4
        MODEFB(K)=MODEN
C       DETERMINE END TIME OF BALKING MODE CYCLE
        STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHB(NSYS)
         TRMFB(K)=ITIME+IDTIME+STIME
131    FORMAT (' BLUE ',I3,' BALKS AT RED ',I3)
C         WRITE (*,131) K,ITGTB(K,1)
          WRITE (10,131) K,ITGTB(K,1)
       ENDIF
```

```
C     IF NOTHING ELSE, THEN ENGAGE THE TARGET
      IF (MODEN.EQ.0) THEN
       MODEN=2
       MODEFB(IK)=MODEN
C      DETERMINE END TIME OF ENGAGMENT MODE CYCLE
C      BASED ON INTERPOLATING BETWEEN MIN RANGE AND MAX RANGE
C       ENGAGEMENT TIMES
       IST=(NSYS-1)*3+MYWP
       NRG=NRWB(NSYS,MYWP)
       ISRM=(NSYS-1)*18+(MYWP-1)*6+NRG
       ISR1=(NSYS-1)*18+(MYWP-1)*6+1
       RMIN=RB(ISR1)
       RMAX=RB(ISRM)
       TMIN=TEMING(IST)
       TMAX=TEMAXB(IST)
       CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)

C     IF TARGET TO BE ENGAGED IS IN ZONE 3 (360 DEGREES),
C     MUST ADD THIS EXTRA SEARCH TIME TO THE ENGAGEMENT TIME
       STIME3=360.0/SRCHB(NSYS)*IN3
        TRMFB(IK)=ITIME+IDTIME+STIME+STIME3
132   FORMAT (' BLUE ',I3,' ENGAGES RED ',I3,' WEAPON ',I3,
     &  ' RANGE ',F6.0)
C      WRITE (*,132) K,ITGTB(K,1),MYWP,TRNG*1000.
      WRITE (10,132) K,ITGTB(K,1),MYWP,TRNG*1000.
C
CC
C     EVALUATE EM ENERGY AVAILABILITY FOR THIS BLUE
      NSYS=IBSYS(K,1)
      IF (NSYS.EQ.NBLUE) THEN
       IF (MYWP.EQ.1.AND.EL1.LT.1.0) THEN
       IENL=EL1*100
        WRITE (10,*) ' BLUE ',K,' WEAPON ',MYWP,' FIRES AT ',IENL,
     &'% ENERGY'
       ENDIF
       IF (MYWP.EQ.2.AND.EL2.LT.1.0) THEN
       IENL=EL2*100
        WRITE (10,*) ' BLUE ',K,' WEAPON ',MYWP,' FIRES AT ',IENL,
     &'% ENERGY'
       ENDIF
       IF (MYWP.EQ.3.AND.EL3.LT.1.0) THEN
       IENL=EL3*100
        WRITE (10,*) ' BLUE ',K,' WEAPON ',MYWP,' FIRES AT ',IENL,
     &'% ENERGY'
       ENDIF
      ENDIF
       ENDIF
      ENDIF
C                                  END LOOP 7
C     END OF TARGET SEARCH EVALUATION RESULTS

     ENDIF
C                                END LOOP 3
C     END OF IT'S TIME TO SELECT A TARGET
C
     ENDIF
C                                END LOOP 2
C     END OF SEARCH MODE
C
      IF (MODEF.EQ.2.OR.MODEF.EQ.6.OR.MODEF.EQ.9.OR.
     &  MODEF.EQ.8) THEN
C                                     LOOP 9
C     ENGAGEMENT MODE
C     CHECK FOR MODE CHANGE AT END OF SUB-CYCLE
      TCHANGE=TRMFB(IK)
       TNEXT=ITIME+IDTIME
      IF (TCHANGE.LE.TNEXT) THEN
C                                  LOOP 9
C      IT'S TIME TO KILL THE TARGET
C
C     BLUE EXPENDS ONE SHOT
      JKILL=0
      IAMOUT=0
      NWP=ITGTB(IK,2)
      NTG=ITGTB(K,1)
      TRNG=PKTB(IK,2)
      NSHOTB(K,NWP)=NSHOTB(K,NWP)-1
      IF (NSHOTB(K,NWP).EQ.0) IAMOUT=1
      IAMR(NTG)=IAMR(NTG)-1
      IF (IAMR(NTG).LT.0) IAMR(NTG)=0
C  EM ENERGY
      BENERGY=0.0
      NSYS=IBSYS(K,1)
      IF (NSYS.EQ.NBLUE) THEN
       ELSYS(K)=ELSYS(K)-EWP(NWP,1)
       BENERGY=1.0
       IF (ELSYS(K).LT.0.0) BENERGY=1.0-ABS(ELSYS(K)/EWP(NWP,1))
       IF (ELSYS(K).LT.0.0) ELSYS(K)=0.0
      IENERGY=NINT(BENERGY*100)
      ENDIF
C
C     FIND SEARCH AREAS FOR RE-MODING INFORMATION
      NSYS=IBSYS(K,1)
      ALFT1=IBPRIME(K,1)
      ARGT1=IBPRIME(K,2)
      ALFT2=IBSECOND(K,1)
      ARGT2=IBSECOND(K,2)
      IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
      IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
      IF (IBPRIME(K,1).EQ.IBSECOND(K,1).AND.IBPRIME(K,2).
     &    EQ.IBSECOND(K,2)) THEN
       ALFT2=-1.0
       ARGT2=-1.0
       ENDIF
C     DETERMINE END TIME OF SEARCH MODE CYCLE
      STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHB(NSYS)
      PROB=RND()
     PKIT=PKTB(K,1)
C
C      CHECK RED TARGET FOR ACTIVE PROTECTION
     CALL CHECKAPS (1,NSYS,NTG,NWP,PKIT,IFAPS,TRNG)
           XR=RXYA(NTG,1)
           YR=RXYA(NTG,2)
           XB=BXYA(K,1)
           YB=BXYA(K,2)
C          DRAW LINE FROM SHOOTER TO TARGET
          CALL PLINE (XR,YR,XB,YB,1)
       IF (PROB.LE.PKIT) THEN
C                                  LOOP 10
C                        KILL TARGET AND RE-MODE
C     RE-EVALUATE MASKING AND OBSCURANT
C     TERRAIN MASKING
      JSKIP=1
      IF (ITER.EQ.1) CALL TERRMASK (XB,YB,XR,YR,JSKIP)

C     IF THIS IS A MAN IN THE LOOP (MIL) WEAPON, THE TARGET
C     MUST STILL BE ABLE TO BEEN SEEN AND TRACKED IN ORDER TO
C     BE HIT.  EVALUATE CURRENT DUST AND ENGINE SMOKE SITUATION.
C
C     EVALUATE M-I-L OR F-A-F
      MILTYPE=MILB(NSYS,NWP)
      IF (MILTYPE.EQ.1) THEN
C     THIS WEAPON IS M-I-L, RE-EVALUATE OBSCURANTS FOR MASKING

C     IF THIS BLUE SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION
C     OR TRACKING
C     FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C     RE-EVALUATED
      RANGE=TRNG*1000.
      ITRK1=ITRKB(NSYS,1)
      ISROF=(NSYS-1)*3+NWP
      IRNG1=IRNGB(ISROF,1)
      ICANRD=1
      IF (IRNG1.EQ.0) ICANRD=0
      IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
     & CALL DMASK (XB,YB,RANGE,0.0,0.0)
C
C     FIND ANGLE AND RANGE TO ALL BURNING VEHICLES, SINCE THEY COMPLETELY
C     MASK VISUAL AND IR SYSTEMS
     CALL COOKMASK (XB,YB,RANGE,0.0,0.0)
C
C      IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C     CAN BE RE-EVALUATED
      ITRK2=ITRKB(NSYS,2)
      ISROF=(NSYS-1)*3+NWP
      IRNG2=IRNGB(ISROF,2)
      ICANRE=1
      IF (IRNG2.EQ.0) ICANRE=0
      IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
     & CALL ESMASK (XB,YB,RANGE,0.0,0.0)
C
C      IF THIS BLUE SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C     CAN BE RE-EVALUATED
      ITRK3=ITRKB(NSYS,3)
      ISROF=(NSYS-1)*3+NWP
      IRNG3=IRNGB(ISROF,3)
      ICANRE=1
      IF (IRNG3.EQ.0) ICANRS=0
      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
     & CALL GRMASK (XB,YB,RANGE,0.0,0.0)
C
C      IF THIS BLUE SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C      TRACKING OR RANGING,
C      FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
MASKING
C      CAN BE RE-EVALUATED
     ITRK4=ITRKB(NSYS,4)
     ISROF=(NSYS-1)*3+NWP
     IRNG4=IRNGB(ISROF,4)
     ICANRA=1
     IF (IRNG4.EQ.0) ICANRA=0
     IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
     & CALL ASMASK (XB,YB,RANGE,0.0,0.0)
C
C      DUST MASKING
C     RE-EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET

C     FIND ANGLE TO THIS TARGET
      TOP=YR-YB
      BOT=XR-XB
       IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
      IF (BOT.LT.0.0) THEN
           IF (TOP.LT.0.0) DIR=DIR+180.
           IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
           IF (TOP.LT.0.0) DIR=360.-DIR
      ENDIF
```

```
C     CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
      CALL FAZ (DIR,AZIN)
C
      JDUST=1
      IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
     & CALL DSEE (XB,YB,XR,YR,AZIN,JDUST)

C     IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE OR
C     TRACK THROUGH IT, THE TARGET IS MISSED
      IF (JDUST.EQ.0.AND.ITRK1.EQ.0) JSKIP=0

C     IF JSKIP=1 THERE MAY STILL BE DUST IN THE WAY, BUT BLUE CAN SEE
C     THROUGH IT.  THE QUESTION REMAINS WHETHER BLUE CAN TRACK THROUGH
C     THE DUST.

      IF (JDUST.EQ.0.AND.ICANRD.EQ.0) JSKIP=0

C     IF JSKIP=1, THERE IS NO DUST OR TRACKING PROBLEM
C
C     BURNING VEHICLE MASKING
            CALL COOKSEE (XB,YB,XR,YR,AZIN,JSKIP)

C     ENGINE SMOKE MASKING
C     RE-EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
      JENG=1
      IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
     & CALL ESSEE (XB,YB,XR,YR,AZIN,JENG)
      IF (JENG.EQ.0.AND.ITRK2.EQ.0) JSKIP=0
      IF (JENG.EQ.0.AND.ICANRE.EQ.0) JSKIP=0

C     SMOKE GRENADE CURTAIN MASKING
C     RE-EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
      JSMK=1
      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
     & CALL GRSEE (XB,YB,XR,YR,AZIN,JSMK)
      IF (JSMK.EQ.0.AND.ITRK3.EQ.0) JSKIP=0
      IF (JSMK.EQ.0.AND.ICANRS.EQ.0) JSKIP=0

C     ARTILLERY SMOKE CURTAIN MASKING
C     RE-EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS
TARGET
            JART=1
            IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
     & CALL ASSEE (XB,YB,XR,YR,AZIN,JART)
            IF (JART.EQ.0.AND.ITRK4.EQ.0) JSKIP=0
            IF (JART.EQ.0.AND.ICANRA.EQ.0) JSKIP=0

      ENDIF
C     END OF OBSCURANT RE-EVALUATION FOR M-H-L WEAPON

       IF (JSKIP.EQ.1) THEN
C
C     RED TARGET KILLED
C     TALLY RED KILLS FOR RESOLUTION AFTER THIS SUB-CYCLE
C     JUST IN CASE THIS RED TARGET IS ABOUT TO KILL A BLUE
C     TARGET
       IKILB(NTG)=1
            JKILL=1
C      RE-MODE TO SEARCH
       MODEN=1
133    FORMAT (' BLUE ',I3,' KILLS RED ',I3)
C         IF (IRSYS(NTG,2).NE.0) WRITE (*,133) K,NTG
C       IF (IAMOUT.EQ.1.AND.MODEF.NE.9)
C     & WRITE (*,*) 'BLUE ',K,' WEAPON ',NWP,' OUT OF AMMO'
         IF (IRSYS(NTG,2).NE.0) WRITE (10,133) K,NTG
       IF (IAMOUT.EQ.1.AND.MODEF.NE.9)
     & WRITE (10,*) 'BLUE ',K,' WEAPON ',NWP,' OUT OF AMMO'
            IAMOUT=0
C      IF (IRSYS(NTG,2).NE.0) CALL OUTPUT2 (ITIME,2,NTG,0)

       ELSE

C        SOME SORT OF MASKING GOT IN THE WAY OF M-H-L SYSTEM
C        OR TERRAIN GOT IN THE WAY FOR F-A-F WEAPON WHILE TARGET
C        WAS MOVING
         IF (IRSYS(NTG,2).NE.0) WRITE (10,134) K,NTG
       IF (MODEF.EQ.9) GO TO 136
       IF (IAMOUT.EQ.1.AND.MODEF.NE.9)
     & WRITE (10,*) 'BLUE ',K,' WEAPON ',NWP,' OUT OF AMMO'
       IAMOUT=0
            WRITE (10,*) 'BLUE ',K,' SEARCHES FOR A BETTER TARGET'
        MODEN=1

       ENDIF

      ELSE

134    FORMAT (' BLUE ',I3,' MISSES RED ',I3)
C         IF (IRSYS(NTG,2).NE.0) WRITE (*,134) K,NTG
C       IF (IAMOUT.EQ.1.AND.MODEF.NE.9)
C     & WRITE (*,*) 'BLUE ',K,' WEAPON ',NWP,' OUT OF AMMO'
         IF (IRSYS(NTG,2).NE.0) WRITE (10,134) K,NTG
       IF (IAMOUT.EQ.1.AND.MODEF.NE.9)
     & WRITE (10,*) 'BLUE ',K,' WEAPON ',NWP,' OUT OF AMMO'
       IAMOUT=0
       IF (MODEF.EQ.9) GO TO 136
C      RE-MODE BLUE SYSTEM AFTER MISS
C      THERE ARE TWO OPTIONS -- 1 RE-ENGAGE
C                            - 2 RE-SEARCH
C      RE-ENGAGE BASED ON AVAILABLE AMMUNITION, AND THE Pk
C      AND AVAILABLE EM ENERGY
C      DO NOT RE-ENGAGE IF EM ENERGY IS BELOW FULL PK SHOT
C      RE-SEARCH IF DECISION IS TO NOT RE-ENGAGE
      PKIT=PKTB(K,1)
       NSHOTS=NSHOTB(K,NWP)
       SKILL=LOG(.01)/LOG(1.-PKIT)
C      EM ENERGY
       NSYS=IBSYS(K,1)
       IF (NSYS.EQ.NBLUE) THEN
        ENEED=EWP(NWP,1)
        IF (ELSYS(K).LT.ENEED) THEN
         NSHOTS=0.0
         MODENEM=3
        IF (IRSYS(NTG,2).NE.0)
     &   WRITE (10,*) ' BLUE ',K,' INSUFFICIENT EM ENERGY TO RE-ENGAGE'
        ENDIF
       ENDIF
            IF (NSHOTS.EQ.0) MODEN=1
C           IF (SKILL.GT.16.0) MODEN=1
C           IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
C           IF (SKILL.LE.16.0.AND.NSHOTS.NE.0) THEN
C           IF (SKILL.GT.7.0.AND.NSHOTS.NE.0) THEN
           IF (NSHOTS.NE.0) THEN
               PSHOOT=NSHOTS/SKILL
               PNO=RNO()
C               IF (SKILL.LE.7) PNO=0.0
                  IF (SKILL.LE.45) PNO=0.0
              IF (PNO.GE.PSHOOT.OR.MODEF.EQ.8) THEN
                  MODEN=1
C         WRITE (*,*) 'BLUE ',K,' SEARCHES FOR A BETTER TARGET'
         WRITE (10,*) 'BLUE ',K,' SEARCHES FOR A BETTER TARGET'
               ELSE
                IF (IRSYS(NTG,2).NE.0) THEN
C                  RE-ENGAGE THE TARGET
                                MODEN=6
C         WRITE (*,*) 'BLUE ',K,' RE-ENGAGES RED ',NTG
C         WRITE (10,*) 'BLUE ',K,' RE-ENGAGES RED ',NTG
C        DETERMINE END TIME OF ENGAGMENT MODE CYCLE
C        BASED ON INTERPOLATING BETWEEN MIN RANGE AND MAX RANGE
C         ENGAGEMENT TIMES
         TRNG=PKTB(K,2)
         IST=(NSYS-1)*3+NWP
         NRG=NRWB(NSYS,NWP)
         ISRM=(NSYS-1)*18+(NWP-1)*6+NRG
         ISR1=(NSYS-1)*18+(NWP-1)*6+1
         RMIN=RB(ISR1)
         RMAX=RB(ISRM)
         TMIN=TEMINB(IST)
         TMAX=TEMAXB(IST)
         CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
                ELSE
                MODEN=1
                ENDIF
               ENDIF
            ENDIF
135    CONTINUE
      ENDIF
C                              END LOOP 10
C     END OF TARGET KILLED AND SYSTEM RE-MODED

       TRMFB(K)=ITIME+IDTIME+STIME
      IF (MODENEM.EQ.3) MODEN=3
      MODEFB(K)=MODEN
      IF (MODEF.EQ.9) MODEFB(K)=-9

C     WRITE SHOT OUTPUT INFORMATION
    IF (IRSYS(NTG,2).EQ.0) THEN
     JKILL=-1
     IF (IFAPS.NE.-1) IFAPS=-3
    ENDIF
      CALL OUTPUT1 (ITIME,1,K,NSYS,NWP,IENERGY,TRNG,NTG,IRSYS(NTG,1),
     &              IFAPS,JKILL)
C

      ELSE
C      END OF IT'S TIME TO KILL THE TARGET
C
C     CHECK TO SEE IF TARGET WAS KILLED BY ANOTHER WEAPON DURING
C      THIS ENGAGEMENT PERIOD.  IF IT WAS, CHECK TO SEE IF THIS WEAPON
C     CAN REDESIGNATE.
C
      NTG=ITGTB(K,1)
      IF (IRSYS(NTG,2).EQ.0) THEN
C      THIS TARGET HAS JUST BEEN KILLED, TRY TO RE-DESIGNATE
C       IF THIS SYSTEM JUST RE-DESIGNATED, HOWEVER, MUST WAIT
C       3 SECOND RE-DESIGNATION TIME PERIOD BEFORE DOING IT AGAIN
       IF (MODEF.EQ.8) THEN
        TCAN=TRDESB(K)
       ELSE
        TCAN=0.0
       ENDIF
       NWP=ITGTB(K,2)
       NSYS=IBSYS(K,1)
       MILTYPE=MILB(NSYS,NWP)
        IF (MILTYPE.EQ.1.AND.ITIME.GE.TCAN) THEN
C        THIS WEAPON IS MAN-IN-LOOP.  IT CAN RE-DESIGNATE, BUT NEEDS
C        AT LEAST 3 SECONDS FOR GUNNER TO OVERCOME HIS SURPRISE AND
C        QUICKLY ASSESS THE SITUATION, THEN FIND ANOTHER TARGET.
C         FIND TIME TO IMPACT
         TRDESB(K)=ITIME+3.0
         THIT=TRMFB(K)-ITIME
          IWANT=0
          IF (THIT.GT.3.0) THEN
C          THERE IS ENOUGH TIME
```

```
      THIT=THIT-3.0
      TRNG=PKTB(K,2)
C      DETERMINE ENGAGMENT MODE CYCLE TIME TO THIS TARGET
C      BASED ON INTERPOLATING BETWEEN MIN RANGE AND MAX RANGE
C      ENGAGEMENT TIMES AND TARGET RANGE
      MYWP=NWP
      IST=(NSYS-1)*3+MYWP
      NRG=NRWB(NSYS,MYWP)
      ISRM=(NSYS-1)*18+(MYWP-1)*6+NRG
      ISR1=(NSYS-1)*18+(MYWP-1)*6+1
      RMIN=RB(ISR1)
      RMAX=RB(ISRM)
      TMIN=TEMINB(IST)
      TMAX=TEMAXB(IST)
      CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
C      FIND DISTANCE MISSILE IS FROM TARGET
      DMSL=TRNG/STIME*THIT
C      MISSILE CAN BE RE-DESIGNATED TO FIND ANOTHER RED
C      TARGET WITHIN THIS DISTANCE FROM THE OLD TARGET
C      BUT NOT BEYOND SYSTEM MAXIMUM RANGE.
C
C      FIND CLOSEST OTHER RED TARGET
C      CURRENT TARGET X,Y POSITION
      XT=RXYA(NTG,1)
      YT=RXYA(NTG,2)
      XB=BXYA(K,1)
      YB=BXYA(K,2)
C      SEARCH ALL RED TARGETS FOR CLOSEST
      DRMIN=1.0E12
C    FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
C    SO THAT MASKING CAN BE EVALUATED
C
      KSIDE=1
      MYSIDE=1
      IBLUE=K
      CALL VMASK (KSIDE,IBLUE)

C    EVALUATE OBSCURANTS ALSO
C    IF THIS BLUE SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION
C    OR TRACKING
C    FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C    RE-EVALUATED
     RANGE=TRNG*1000.*1.1
     ITRK1=ITRKB(NSYS,1)
     ISROF=(NSYS-1)*3+NWP
     IRNG1=IRNGB(ISROF,1)
     ICANRD=1
     IF (IRNG1.EQ.0) ICANRD=0
     IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
    & CALL DMASK (XB,YB,RANGE,0.0,0.0)
C
C            CALL COOKMASK (XB,YB,RANGE,0.0,0.0)
C
C    IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C    TRACKING OR RANGING,
C    FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C    CAN BE RE-EVALUATED
     ITRK2=ITRKB(NSYS,2)
     ISROF=(NSYS-1)*3+NWP
     IRNG2=IRNGB(ISROF,2)
     ICANRE=1
     IF (IRNG2.EQ.0) ICANRE=0
     IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
    & CALL ESMASK (XB,YB,RANGE,0.0,0.0)
C
C    IF THIS BLUE SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C    TRACKING OR RANGING,
C    FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C    CAN BE RE-EVALUATED
     ITRK3=ITRKB(NSYS,3)
     ISROF=(NSYS-1)*3+NWP
     IRNG3=IRNGB(ISROF,3)
     ICANRS=1
     IF (IRNG3.EQ.0) ICANRS=0
     IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
    & CALL GRMASK (XB,YB,RANGE,0.0,0.0)
C
C    IF THIS BLUE SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C    TRACKING OR RANGING,
C    FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
MASKING
C    CAN BE RE-EVALUATED
          ITRK4=ITRKB(NSYS,4)
          ISROF=(NSYS-1)*3+NWP
          IRNG4=IRNGB(ISROF,4)
          ICANRA=1
          IF (IRNG4.EQ.0) ICANRA=0
          IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
    & CALL ASMASK (XB,YB,RANGE,0.0,0.0)
C
      DO 11 JRED=1,NR
       IF (IRSYS(JRED,2).EQ.0) GO TO 11
      XNT=RXYA(JRED,1)
      YNT=RXYA(JRED,2)
      DNT=((XNT-XT)**2+(YNT-YT)**2)**.5
      DNB=((XNT-XB)**2+(YNT-YB)**2)**.5
       RANGE=DNB
C       CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
C       NOT DETECTED
       CALL CHECKSIG (1,K,JRED,RANGE,JSKIP)
        IF (JSKIP.EQ.0) GO TO 11
          IF (DNB.LE.RMAX.AND.DNT.LE.DMSL) THEN
C           DIVIDE BY A RANDOM FACTOR SO EVERYONE DOES NOT
C           TAKE THE SAME ALTERNATE TARGET
          RRR=RND()
          DNT=DNT/RRR
            IF (DNT.LT.DRMIN) THEN
C             EVALUATE MASKING
C     FIND ANGLE TO THIS TARGET
          TOP=YNT-YB
          BOT=XNT-XB
           IF (BOT.EQ.0.0) BOT=BOT+.001
          DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
          IF (BOT.LT.0.0) THEN
              IF (TOP.LT.0.0) DIR=DIR+180.
              IF (TOP.GT.0.0) DIR=180.-DIR
          ELSE
              IF (TOP.LT.0.0) DIR=360.-DIR
          ENDIF
C     CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
          CALL FAZ (DIR,AZIN)
C
C     AT THIS POINT EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE
C     MASKING AND OBSCURANTS AND DROP THIS
C     TARGET IF UNOBSERVABLE
          JSKIP=1
C     TERRAIN MASKING
          IF (ITER.EQ.1) CALL TERRMASK (XB,YB,XNT,YNT,JSKIP)
          IF (JSKIP.EQ.0) GO TO 11
          CALL VSEE (MYSIDE,K,JRED,RANGE,AZIN,JSKIP,NB,NR)
C
C     BURNING VEHICLE MASKING
          CALL COOKSEE (XB,YB,XNT,YNT,AZIN,JSKIP)

          IF (JSKIP.EQ.0) GO TO 11

C     DUST MASKING
C     RE-EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET

          JDUST=1
          IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
         & CALL DSEE (XB,YB,XNT,YNT,AZIN,JDUST)

C     IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE OR
C     TRACK THROUGH IT, THE TARGET IS MISSED
          IF (JDUST.EQ.0.AND.ITRK1.EQ.0) JSKIP=0

C     IF JSKIP=1 THERE MAY STILL BE DUST IN THE WAY, BUT BLUE CAN SEE
C     THROUGH IT.  THE QUESTION REMAINS WHETHER BLUE CAN TRACK THROUGH
C     THE DUST.

          IF (JDUST.EQ.0.AND.ICANRD.EQ.0) JSKIP=0

C     IF JSKIP=1, THERE IS NO DUST OR TRACKING PROBLEM

C     ENGINE SMOKE MASKING
C     RE-EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
          JENG=1
          IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
         & CALL ESSEE (XB,YB,XNT,YNT,AZIN,JENG)
          IF (JENG.EQ.0.AND.ITRK2.EQ.0) JSKIP=0
          IF (JENG.EQ.0.AND.ICANRE.EQ.0) JSKIP=0

C     SMOKE GRENADE CURTAIN MASKING
C     RE-EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
          JSMK=1
          IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
         & CALL GRSEE (XB,YB,XNT,YNT,AZIN,JSMK)
          IF (JSMK.EQ.0.AND.ITRK3.EQ.0) JSKIP=0
          IF (JSMK.EQ.0.AND.ICANRS.EQ.0) JSKIP=0

C     ARTILLERY SMOKE CURTAIN MASKING
C     RE-EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS
TARGET
          JART=1
          IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
         & CALL ASSEE (XB,YB,XNT,YNT,AZIN,JART)
          IF (JART.EQ.0.AND.ITRK4.EQ.0) JSKIP=0
          IF (JART.EQ.0.AND.ICANRA.EQ.0) JSKIP=0

C     END OF OBSCURANT RE-EVALUATION FOR M-I-L WEAPON

          IF (JSKIP.EQ.0) GO TO 11

C     TARGET CAN BE SEEN AND IS CLOSER TO OLD TARGET
                DRMIN=DNT
                IWANT=JRED
                RNG=DNB
              ENDIF
            ENDIF
11        CONTINUE
          ENDIF
            IF (IWANT.NE.0) THEN
C             THERE IS ANOTHER TARGET TO DESIGNATE
C  FIND PROBABILTY OF KILLING THIS NEW RED TARGET
              NITSYS=IRSYS(IWANT,1)
          CALL FINDPK (KSIDE,NSYS,NITSYS,NWP,NRNG,RNG,PKIT,IWANT)
           PKTB(K,1)=PKIT
           PKTB(K,2)=RNG
              MODEN=8
              STIME=THIT+3.0
              ITGTB(K,1)=IWANT
C               TRMFB(K)=ITIME+IDTIME+STIME
```

```
C          MODEFB(K)=MODEN
C          WRITE (*,*) 'BLUE ',K,' RE-DESIGNATES AT RED ',IWANT
                   WRITE (10,*) 'BLUE ',K,' RE-DESIGNATES AT RED ',IWANT
                   ELSE
C          NO TARGETS, FINISH OUT CURRENT ENGAGEMENT MODE TIME
C          THEN RE-SEARCH
           MODEN=1
C          BLUE EXPENDS ONE SHOT
           NWP=ITGTB(K,2)
           NTG=ITGTB(K,1)
           TRNG=PKTB(K,2)
           NSHOTB(K,NWP)=NSHOTB(K,NWP)-1
           IAMR(NTG)=IAMR(NTG)-1
           IF (IAMR(NTG).LT.0) IAMR(NTG)=0
C
C          FIND SEARCH AREAS FOR RE-MODING INFORMATION
           NSYS=IBSYS(K,1)
           ALFT1=IBPRIME(K,1)
           ARGT1=IBPRIME(K,2)
           ALFT2=IBSECOND(K,1)
           ARGT2=IBSECOND(K,2)
           IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
           IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
           IF (IBPRIME(K,1).EQ.IBSECOND(K,1).AND.IBPRIME(K,2).
     &      EQ.IBSECOND(K,2)) THEN
           ALFT2=-1.0
           ARGT2=-1.0
           ENDIF
C          DETERMINE END TIME OF SEARCH MODE CYCLE
           STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHB(NSYS)
           ENDIF

           TRMFB(K)=ITIME+IDTIME+STIME
           MODEFB(K)=MODEN

C     WRITE SHOT OUTPUT INFORMATION
           JKILL=-1
           IFAPS=-1
      CALL OUTPUT1 (ITIME,1,K,NSYS,NWP,0,TRNG,NTG,IRSYS(NTG,1),
     &          IFAPS,JKILL)
C
        ELSE
       IF (IAMOUT.EQ.1)
C     & WRITE (*,*) 'BLUE ',K,' WEAPON ',NWP,' OUT OF AMMO'
     & WRITE (10,*) 'BLUE ',K,' WEAPON ',NWP,' OUT OF AMMO'
       IAMOUT=0
        ENDIF
C       END OF IT CAN BE RE-DESIGNATED
      ENDIF
C     END OF THIS TARGET HAS JUST BEEN KILLED

      ENDIF
C                              END LOOP 9
C     END OF REDESIGNATING MODE

      ENDIF
C                              END LOOP 8
C     END OF ENGAGEMENT MODE

150   CONTINUE
C                          END LOOP 1
C     END OF BLUE SYSTEM FIRE MODE LOOP
C
C     NOW DO RED
C
C     RED
C
      DO 1150 K=1,NR
C                              LOOP 1
C     CHECK TO SEE IF THIS UNIT IS KILLED AND SKIP
C      WRITE (*,*) 'KILL STATUS RED ',K,' ALIVE'
C     BUT BEFORE SKIPPING IT, SEE IF IT WAS A FIRE-AND-FORGET
C     WEAPON AND IN THE LAST SECONDS OF ENGAGMENT MODE OR
C     RE-ENGAGEMENT MODE.
      MODEF=MODEFR(K)
      IF (IRSYS(K,2).EQ.0) THEN
       NSYS=IRSYS(K,1)
       IF (MODEF.EQ.2.OR.MODEF.EQ.6) THEN
        NWP=ITGTR(K,2)
        MILTYPE=MILR(NSYS,NWP)
        IF (MILTYPE.EQ.0) THEN
C        CALCULATE TIME TO BULLET IMPACT
         TCHANGE=TRMFR(K)
         BORE=BORER(NSYS,NWP)/1000.
         TOF=TOFR(NSYS,NWP)
         TRNG=PKTR(K,2)
         TTRNG=TRNG/BORE*TOF+1.0
         IF (ITIME.GE.(TCHANGE-TTRNG)) THEN
C         THIS WEAPON IS NOW IN SACRIFICE MODE SINCE IT IS DEAD AND
C         HAPPENED TO GET A FIRE-AND-FORGET SHOT OFF FIRST
          MODEF=9
                MODEFR(K)=MODEF
                NSHOTR(K,NWP)=NSHOTR(K,NWP)-1
                CALL OUTPUT2 (ITIME,2,K,0)
C         WRITE (*,*) ' RED ',K,' MAKES SACRIFICE SHOT'
          WRITE (10,*) ' RED ',K,' MAKES SACRIFICE SHOT'
         ELSE
          MODEFR(K)=0
C         FIRE MODE SET TO 0 -- KILLED WHILE ENGAGING A TARGET
         ENDIF
          ENDIF
         ENDIF
        ENDIF
      IF (IRSYS(K,2).EQ.0.AND.MODEF.NE.9) GO TO 1150
C
      IF (MODEF.EQ.-6) THEN
C        RE-EVALUATE RE-ENGAGEMENT MODE TO SEE IF THIS RED TARGET
C        WAS JUST KILLED
       IF (IBSYS(ITGTR(K,1),2).EQ.1) THEN
        MODEF=6
        MODEFR(K)=6
        WRITE (10,*) 'RED ',K,' RE-ENGAGES BLUE ',ITGTR(K,1)
       ELSE
C        THIS TARGET WAS KILLED LAST GO AROUND SO FIND ANOTHER
        MODEF=1
        MODEFR(K)=1
        NSYS=IRSYS(K,1)
        ALFT1=IRPRIME(K,1)
        ARGT1=IRPRIME(K,2)
        IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
        ALFT2=IRSECOND(K,1)
        ARGT2=IRSECOND(K,2)
        IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
        IF (IRPRIME(K,1).EQ.IRSECOND(K,1).AND.IRPRIME(K,2).
     &      EQ.IRSECOND(K,2)) THEN
        ALFT2=-1.0
        ARGT2=-1.0
        ENDIF
C       DETERMINE END TIME OF SEARCH MODE CYCLE
        STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHR(NSYS)
        TRMFR(K)=ITIME+STIME
       ENDIF
      ENDIF
C
       MODEF=MODEFR(K)
C       WRITE (*,*) 'FIRE MODE RED ',MODEF
C      INCREASE CYCLE TIME IN THIS MODE
       TCFMR(K,MODEF)=TCFMR(K,MODEF)+DT
C       WRITE (*,*) 'CYCLE TIME RED ',TCFMR(K,MODEF)
C
       IF (MODEF.EQ.1.OR.MODEF.EQ.4.OR.MODEF.EQ.5) THEN
C                                    LOOP 2
C
C      SEARCH MODE (1) OR BALK MODE (4) OR WAIT MODE (5)
C      BALK MODE IS ACTUALLY RE-SEARCHING, BUT IS BROKEN OUT
C      FOR DUTY CYCLE ACCOUNTABILITY
C      WAIT MODE IS ACTUALLY RE-SEARCHING, BUT IS BROKEN OUT
C      FOR DUTY CYCLE ACCOUNTABILITY
C
C      CHECK FOR MODE CHANGE AT END OF SUB-CYCLE
       TCHANGE=TRMFR(K)
C       WRITE (*,*) 'FIRE MODE CHANGE RED ',TCHANGE
       TNEXT=ITIME+IDTIME
       IF (TCHANGE.LE.TNEXT) THEN
C                                      LOOP 3
C      IT'S TIME TO FIND A TARGET
C
C      FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
C      SO THAT MASKING CAN BE EVALUATED
C
       KSIDE=2
       IRED=K
       MYSIDE=2
       CALL VMASK (KSIDE,IRED)
C
C      FIND ALL BLUE TARGETS IN PRIMARY AND SECONDARY ZONES
C      OUT TO MAX RANGE FOR EACH ON-BOARD WEAPON
       NSYS=IRSYS(K,1)
       NWEAP=NWR(NSYS)
       NRNG1=NRWR(NSYS,1)
       RMAX1=0.0
       RMAX2=0.0
       RMAX3=0.0
       NUM1=0
       NUM2=0
       NUM3=0
C      ISR=(I-1)*18+(J-1)*6+K
C       WHERE I=SYSTEM FILE (10 MAX); J=SYSTEM WEAPON (3 MAX);
C             K=WEAPON RANGE (6 MAX); N=TARGET (2*10 MAX)
       NUM1=(NSYS-1)*18+(1-1)*6+NRNG1
       RMAX1=RR(NUM1)
C       WRITE (*,*) 'RMAX1',RMAX1,'RR(22)',RR(22)
C       WRITE (*,*) 'NSYS',NSYS,'NWEAP',NWEAP,'NRNG1',NRNG1,'NUM1',NUM1
C       WRITE (*,*) 'NSHOTR',NSHOTR(K,1)
            IF (NSHOTR(K,1).LE.0) THEN
            RMAX1=0.0
            NUM1=0
            ENDIF
       IF (NWEAP.GE.2) THEN
       NRNG2=NRWR(NSYS,2)
       NUM2=(NSYS-1)*18+(2-1)*6+NRNG2
       RMAX2=RR(NUM2)
C       WRITE (*,*) 'NSYS',NSYS,'NWEAP',NWEAP,'NRNG2',NRNG2,'NUM2',NUM2
C       WRITE (*,*) 'NSHOTR',NSHOTR(K,2)
            IF (NSHOTR(K,2).LE.0) THEN
            RMAX2=0.0
            NUM2=0
            ENDIF
       ENDIF
       IF (NWEAP.EQ.3) THEN
       NRNG3=NRWR(NSYS,3)
       NUM3=(NSYS-1)*18+(3-1)*6+NRNG3
```

```
      RMAX3=RR(NUM3)
C     WRITE (*,*) 'NSYS',NSYS,'NWEAP',NWEAP,'NRNG3',NRNG3,'NUM3',NUM3
C     WRITE (*,*) 'NSHOTR',NSHOTR(K,3)
            IF (NSHOTR(K,3).LE.0) THEN
            RMAX3=0.0
            NUM3=0
            ENDIF
      ENDIF
C
C     SEARCH ALL ZONES BASED ON ZONE AZIMUTHS
            IOVER1=0
            IOVER2=0
      ALFT1=IRPRIME(K,1)
      ARGT1=IRPRIME(K,2)
      ALFT2=IRSECOND(K,1)
            ARGT2=IRSECOND(K,2)
            IF (ALFT1.GT.ARGT1) IOVER1=1
            IF (ALFT2.GT.ARGT2) IOVER2=1
      IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
      IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
      IF (IRPRIME(K,1).EQ.IRSECOND(K,1).AND.IRPRIME(K,2).
     &    EQ.IRSECOND(K,2)) THEN
      ALFT2=-1.0
      ARGT2=-1.0
      ENDIF
C     GET X,Y COORDINATES FOR THIS RED  UNIT
      XB=RXYA(K,1)
      YB=RXYA(K,2)
C     INITIALIZE PRIORITIES
      NTGT1=0
      PRI=0.0
      MYWP1=0
      NTGT2=0
      SEC=0.0
      MYWP2=0
      NTGT3=0
      THIRD=0.0
      MYWP3=0
      TRNG1=0.0
      TRNG2=0.0
      PKWP1=0.0
      PKWP2=0.0
      PKWP3=0.0
C
C     IF THIS RED  SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C     EVALUATED
      ITRK1=ITRKR(NSYS,1)
      ISROF=(NSYS-1)*3+1
      IRNG11=1
      IRNG12=1
      IRNG13=1
      IRNG11=IRNGR(ISROF,1)
      IF (NWEAP.GE.2) IRNG12=IRNGR((ISROF+1),1)
      IF (NWEAP.GT.2) IRNG13=IRNGR((ISROF+2),1)
      ICANRD=1
      IF (IRNG11.EQ.0.OR.IRNG12.EQ.0.OR.IRNG13.EQ.0) ICANRD=0
      IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
     & CALL DMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C
C           CALL COOKMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C
C     IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C     CAN BE EVALUATED
      ITRK2=ITRKR(NSYS,2)
      ISROF=(NSYS-1)*3+1
      IRNG21=1
      IRNG22=1
      IRNG23=1
      IRNG21=IRNGR(ISROF,2)
      IF (NWEAP.GE.2) IRNG22=IRNGR((ISROF+1),2)
      IF (NWEAP.GT.2) IRNG23=IRNGR((ISROF+2),2)
      ICANRE=1
      IF (IRNG21.EQ.0.OR.IRNG22.EQ.0.OR.IRNG23.EQ.0) ICANRE=0
      IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
     & CALL ESMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C
C     IF THIS RED  SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C     CAN BE EVALUATED
      ITRK3=ITRKR(NSYS,3)
      ISROF=(NSYS-1)*3+1
      IRNG31=1
      IRNG32=1
      IRNG33=1
      IRNG31=IRNGR(ISROF,3)
      IF (NWEAP.GE.2) IRNG32=IRNGR((ISROF+1),3)
      IF (NWEAP.GT.2) IRNG33=IRNGR((ISROF+2),3)
      ICANRS=1
      IF (IRNG31.EQ.0.OR.IRNG32.EQ.0.OR.IRNG33.EQ.0) ICANRS=0
      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
     & CALL GRMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C
C     IF THIS RED  SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C     TRACKING OR RANGING,
C     FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
MASKING
C     CAN BE EVALUATED
      ITRK4=ITRKR(NSYS,4)
      ISROF=(NSYS-1)*3+1
      IRNG41=1
      IRNG42=1
      IRNG43=1
      IRNG41=IRNGR(ISROF,4)
      IF (NWEAP.GE.2) IRNG42=IRNGR((ISROF+1),4)
      IF (NWEAP.GT.2) IRNG43=IRNGR((ISROF+2),4)
      ICANRA=1
      IF (IRNG41.EQ.0.OR.IRNG42.EQ.0.OR.IRNG43.EQ.0) ICANRA=0
      IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
     & CALL ASMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C
      DO 1149 J=1,NB
C                                       LOOP 4
C     CHECK TO SEE IF THIS BLUE UNIT IS KILLED
      IF (IBSYS(J,2).EQ.0.OR.IBSYS(J,1).GT.10) GO TO 1149
C     FIND THIS BLUE UNIT'S X,Y COORDINATES
      XR=BXYA(J,1)
      YR=BXYA(J,2)
C     FIND RANGE TO THIS TARGET
      RANGE=(XR-XB)**2+(YR-YB)**2)**.5
C     WRITE (*,*) 'RANGE TO BLUE ',J,RANGE

C     CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
C     NOT DETECTED
      CALL CHECKSIG (2,K,J,RANGE,JSKIP)
      IF (JSKIP.EQ.0) GO TO 1149

C     IF THIS TARGET IS OUTSIDE THE RANGE OF ALL WEAPONS SKIP IT
C     WRITE (*,*) 'RANGE',RANGE,'RMAX1',RMAX1,'RMAX2',RMAX2,
C    &'RMAX3',RMAX3
      JSKIP=0
      IF ((RANGE*1000.).LE.RMAX1) JSKIP=1
      IF ((RANGE*1000.).LE.RMAX2) JSKIP=1
      IF ((RANGE*1000.).LE.RMAX3) JSKIP=1
C     WRITE (*,*) 'RANGE SKIP ',JSKIP,' 0=YES'
      IF (JSKIP.EQ.0) GO TO 1149
C     TARGET IS WITHIN RANGE OF ONE OF THESE WEAPONS
C     IF THIS TARGET IS OUTSIDE THE FIRE ORDERS RANGE WINDOW SKIP IT
      IF (RANGE.LT.WRNGR(K,1).OR.RANGE.GT.WRNGR(K,2)) JSKIP=0
      IF (JSKIP.EQ.0) GO TO 1149
C     FIND ANGLE TO THIS TARGET
      TOP=YR-YB
      BOT=XR-XB
C     WRITE (*,*) 'BOT',BOT
      IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
      IF (BOT.LT.0.0) THEN
            IF (TOP.LT.0.0) DIR=DIR+180.
            IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
            IF (TOP.LT.0.0) DIR=360.-DIR
      ENDIF
C     CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
      CALL FAZ (DIR,AZIN)
C
C     AT THIS POINT EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE
C     MASKING AND OBSCURIANTS AND DROP THIS
C     TARGET FROM ZONE IF UNOBSERVABLE (SMOKE NOT DONE YET)
C
C     TERRAIN MASKING
      IF (ITER.EQ.1) CALL TERRMASK (XR,YR,XB,YB,JSKIP)
      IF (JSKIP.EQ.0) GO TO 1149
C
C     VEHICLE MASKING
      CALL VSEE (MYSIDE,K,J,RANGE,AZIN,JSKIP,NB,NR)
C
C     BURNING VEHICLE MASKING
      CALL COOKSEE (XB,YB,XR,YR,AZIN,JSKIP)

      IF (JSKIP.EQ.0) GO TO 1149
C
C     DUST MASKING
C     IF THIS RED  SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET
      JDUST=1
      IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
     & CALL DSEE (XB,YB,XR,YR,AZIN,JDUST)
C     IF THERE IS DUST IN THE WAY AND RED  CANNOT SEE THROUGH IT
C     SKIP THIS TARGET
      IF (JDUST.EQ.0.AND.ITRK1.EQ.0) GO TO 1149
C     THERE MAY BE DUST IN THE WAY, BUT RED  CAN SEE THROUGH IT.
C     THE QUESTION REMAINS WHETHER RED  CAN RANGE OR TRACK THROUGH
C     THE DUST.  IF JDUST=0 THERE IS DUST IN THE WAY.
C
C     ENGINE SMOKE MASKING
C     IF THIS RED  SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
      JENG=1
      IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
     & CALL ESSEE (XB,YB,XR,YR,AZIN,JENG)
      IF (JENG.EQ.0.AND.ITRK2.EQ.0) GO TO 1149
C
C     SMOKE GRENADE CURTAIN MASKING
C     IF THIS RED  SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
      JSMK=1
```

```
      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
     & CALL GRSEE (XB,YB,XR,YR,AZIN,JSMK)
      IF (JSMK.EQ.0.AND.ITRK3.EQ.0) GO TO 1149
C
C     ARTILLERY SMOKE MASKING
C     IF THIS RED SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C     TRACKING OR RANGING,
C     EVALUATE ALL ARTILLERY SMOKE PUFFS WITHIN RANGE OF THIS TARGET
            JART=1
            IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
     & CALL ASSEE (XB,YB,XR,YR,AZIN,JART)
            IF (JART.EQ.0.AND.ITRK4.EQ.0) GO TO 1149
C
            INZONE=3
      IF (AZIN.GE.ALFT1.AND.AZIN.LE.ARGT1) INZONE=1
            IF (IOVER1.EQ.1.AND.AZIN.LE.(ARGT1-360.0)) INZONE=1
C     TARGET IS IN PRIMARY ZONE
      IF (AZIN.GE.ALFT2.AND.AZIN.LE.ARGT2) INZONE=2
            IF (IOVER2.EQ.1.AND.AZIN.LE.(ARGT2-360.0)) INZONE=2
C     TARGET IS IN SECONDARY ZONE
C     AT THIS POINT EVALUATE TERRAIN MASKING AND OBSCURANTS AND DROP
C     TARGET FROM ZONE IF UNOBSERVABLE (NOT DONE YET)
C
C     WRITE (*,*) 'BLUE ',J,' INZONE ? ',INZONE
      IF (INZONE.GE.1) THEN
C                                   LOOP 5
C     GIVE THIS TARGET A PRIORITY NUMBER
C     AND IN THE END SELECT ONLY THE HIGHEST PRIORITY TARGET
C     TARGET PRIORITY IS A COMPOSITE OF THE PROBABILITY OF THIS
C     TARGET KILLING YOU TIMES YOU'RE PROBABILITY OF KILLING IT.
      NITSYS=ISSYS(J,1)
C     CYCLE THROUGH ALL WEAPONS ON YOUR SYSTEM FOR THE ONE WITH THE
C     HIGHEST PK AGAINST THIS TARGET AND EVALUATE WHETHER YOU WILL
C     SHOOT IT BASED ON NUMBER OF AVAILABLE ROUNDS OF AMMUNITION
C     AND THE PK AGAINST THIS TARGET
      PKIT1=0.0
      PKIT2=0.0
      PKIT3=0.0
      KSIDE=2
      PKITM=0.0
      IWEAP=0
      DO 1141 NUM=1,3
C                                   LOOP 6
      IF (NUM.EQ.1.AND.NUM1.EQ.0) GO TO 1141
      IF (NUM.EQ.2.AND.NUM2.EQ.0) GO TO 1141
      IF (NUM.EQ.3.AND.NUM3.EQ.0) GO TO 1141
      IF (NUM.EQ.1) THEN
      CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG1,RANGE,PKIT1,J)
            IF (ISPDR(K).GT.0.OR.ISPDB(J).GT.0) THEN
             MILTYPE=MILR(NSYS,NUM)
             IF (MILTYPE.EQ.0) THEN
             BORE=BORER(NSYS,NUM)
             TOF=TOFR(NSYS,NUM)
             CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT1)
            ENDIF
           ENDIF
            IF (JDUST.EQ.0.OR.JENG.EQ.0) THEN
       IF (IRNG11.EQ.0.OR.IRNG21.EQ.0) THEN
C      THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
C      EVALUATE M-I-L OR F-A-F
      MILTYPE=MILR(NSYS,NUM)
       IF (MILTYPE.EQ.0) THEN
C       THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
       BORE=BORER(NSYS,NUM)
       TOF=TOFR(NSYS,NUM)
       CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT1)
       ELSE
C       SKIP THIS WEAPON SINCE CAN'T TRACK THE TARGET
       PKIT1=0.0
       ENDIF
      ENDIF
      ENDIF
      NSHOTS=NSHOTR(K,NUM)
C      WRITE (*,*) 'LOG BOT1',LOG(1.-PKIT1),'PKIT1',PKIT1
       IF (PKIT1.NE.0.0) THEN
       SKILL=LOG(.01)/LOG(1.-PKIT1)
       ELSE
       SKILL=9999.
       ENDIF
            IF (NSHOTS.EQ.0) PKIT1=0.0
C           IF (SKILL.GT.16.0) PKIT1=0.0
C           IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
C           IF (SKILL.GT.7.0) THEN
            IF (SKILL.GT.45.0) THEN
                PSHOOT=NSHOTS/SKILL
                PNO=RND()
                 IF (PNO.GE.PSHOOT) PKIT1=0.0
            ENDIF
            IF (PKIT1.GT.PKITM) THEN
                 PKITM=PKIT1
                 IWEAP=1
            ENDIF
      ENDIF
      IF (NUM.EQ.2) THEN
      CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG2,RANGE,PKIT2,J)
            IF (ISPDR(K).GT.0.OR.ISPDB(J).GT.0) THEN
             MILTYPE=MILR(NSYS,NUM)
             IF (MILTYPE.EQ.0) THEN
             BORE=BORER(NSYS,NUM)
             TOF=TOFR(NSYS,NUM)
             CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT2)
            ENDIF
           ENDIF
            IF (JDUST.EQ.0.OR.JENG.EQ.0) THEN
       IF (IRNG12.EQ.0.OR.IRNG22.EQ.0) THEN
C      THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
C      EVALUATE M-I-L OR F-A-F
      MILTYPE=MILR(NSYS,NUM)
       IF (MILTYPE.EQ.0) THEN
C       THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
       BORE=BORER(NSYS,NUM)
       TOF=TOFR(NSYS,NUM)
       CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT2)
       ELSE
C       SKIP THIS WEAPON SINCE CAN'T TRACK THE TARGET
       PKIT2=0.0
       ENDIF
      ENDIF
      ENDIF
      NSHOTS=NSHOTR(K,NUM)
C      WRITE (*,*) 'LOG BOT2',LOG(1.-PKIT2),'PKIT2',PKIT2
       IF (PKIT2.NE.0.0) THEN
       SKILL=LOG(.01)/LOG(1.-PKIT2)
       ELSE
       SKILL=9999.
       ENDIF
            IF (NSHOTS.EQ.0) PKIT2=0.0
C           IF (SKILL.GT.16.0) PKIT2=0.0
C           IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
C           IF (SKILL.GT.7.0) THEN
            IF (SKILL.GT.45.0) THEN
                PSHOOT=NSHOTS/SKILL
                PNO=RND()
                 IF (PNO.GE.PSHOOT) PKIT2=0.0
            ENDIF
            IF (PKIT2.GT.PKITM) THEN
                 PKITM=PKIT2
                 IWEAP=2
            ENDIF
      ENDIF
      IF (NUM.EQ.3) THEN
      CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG3,RANGE,PKIT3,J)
      IF (ISPDR(K).GT.0.OR.ISPDB(J).GT.0) THEN
      MILTYPE=MILR(NSYS,NUM)
       IF (MILTYPE.EQ.0) THEN
       BORE=BORER(NSYS,NUM)
       TOF=TOFR(NSYS,NUM)
       CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT3)
      ENDIF
      ENDIF
       IF (JDUST.EQ.0.OR.JENG.EQ.0) THEN
          IF (IRNG13.EQ.0.OR.IRNG23.EQ.0) THEN
C      THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
C       EVALUATE M-I-L OR F-A-F
       MILTYPE=MILR(NSYS,NUM)
        IF (MILTYPE.EQ.0) THEN
C       THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
        BORE=BORER(NSYS,NUM)
        TOF=TOFR(NSYS,NUM)
        CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT3)
        ELSE
C       SKIP THIS WEAPON SINCE CAN'T TRACK THE TARGET
        PKIT3=0.0
        ENDIF
       ENDIF
      ENDIF
      NSHOTS=NSHOTR(K,NUM)
C      WRITE (*,*) 'LOG BOT3',LOG(1.-PKIT3),'PKIT3',PKIT3
       IF (PKIT3.NE.0.0) THEN
       SKILL=LOG(.01)/LOG(1.-PKIT3)
       ELSE
       SKILL=9999.
       ENDIF
            IF (NSHOTS.EQ.0) PKIT3=0.0
C           IF (SKILL.GT.16.0) PKIT3=0.0
C           IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
C           IF (SKILL.GT.7.0) THEN
            IF (SKILL.GT.45.0) THEN
                PSHOOT=NSHOTS/SKILL
                PNO=RND()
                 IF (PNO.GE.PSHOOT) PKIT3=0.0
            ENDIF
            IF (PKIT3.GT.PKITM) THEN
                 PKITM=PKIT3
                 IWEAP=3
            ENDIF
      ENDIF

1141  CONTINUE
C                                   END LOOP 6
C     END OF RED  WEAPON SELECTION LOOP
C      WRITE (*,*) ' RED WEAPON SELECTION',IWEAP,'PK',PKITM

C     IF TARGET IS AT A QUESTIONABLE EFFECTIVE RANGE SKIP IT
      IF (PKITM.LE.0.0) GO TO 1143
      IF (IWEAP.EQ.0.0) GO TO 1143

C     FIND PROBABILITY OF THIS TARGET KILLING YOU AND APPLY PRIORITY
C     FACTOR
C
      KPFACT=KPRIR(K,NITSYS)
      PKYOU1=0.0
      PKYOU2=0.0
```

```
      PKYOU3=0.0
      KSIDE=1
      NUMIT=NWB(NITSYS)
      IRNG1=NRWB(NITSYS,1)
      IF (NUMIT.GE.2) IRNG2=NRWB(NITSYS,2)
      IF (NUMIT.EQ.3) IRNG3=NRWB(NITSYS,3)
      DO 1142 NUM=1,NUMIT
      IF (NUM.EQ.1) CALL
     &        FINDPK (KSIDE,NITSYS,NSYS,NUM,IRNG1,RANGE,PKYOU1,K)
      IF (NUM.EQ.2) CALL
     &        FINDPK (KSIDE,NITSYS,NSYS,NUM,IRNG2,RANGE,PKYOU2,K)
      IF (NUM.EQ.3) CALL
     &        FINDPK (KSIDE,NITSYS,NSYS,NUM,IRNG3,RANGE,PKYOU3,K)
1142  CONTINUE
C     FIND MAXIMUM PK AGAINST YOU
      PKYMAX=0.0
      IF (PKYOU1.GT.PKYMAX) PKYMAX=PKYOU1
      IF (PKYOU2.GT.PKYMAX) PKYMAX=PKYOU2
      IF (PKYOU3.GT.PKYMAX) PKYMAX=PKYOU3

C     IF TARGET'S PK ON YOU IS ZERO, GIVE IT A SMALL VALUE SO THAT
C     YOU AT LEAST HAVE A TARGET TO SHOOT AT IF NO OTHERS ARE BETTER
C     AFTER ALL THIS TROUBLE OF DECIDING IF YOU CAN KILL IT

      IF (PKYMAX.LE.0.0) PKYMAX=0.01

C     FIND COMPOSITE KILL PRIORITY

C     WRITE (*,*) 'PK IT ON YOU',PKYMAX

      PRIOR=PKITM*PKYMAX*KPFACT
C     ADD A SMALL RANDOM NUMBER SO NO TWO ARE EVER THE SAME
      SMALL=RND()
      SMALL=SMALL/1000.
      PRIOR=PRIOR+SMALL
C
C     PRIORITIZE THIS TARGET WITHIN PRIMARY OR SECONDARY ZONE
C
      IF (INZONE.EQ.1) THEN
C     THIS TARGET IS IN PRIMARY ZONE
       IF (PRIOR.GT.PRI) THEN
          PRI=PRIOR
          NTGT1=J
          MYWP1=IWEAP
          TRNG1=RANGE
          IF (IWEAP.EQ.1) PKWP1=PKIT1
          IF (IWEAP.EQ.2) PKWP1=PKIT2
          IF (IWEAP.EQ.3) PKWP1=PKIT3
       ENDIF
      ENDIF
      IF (INZONE.EQ.2) THEN
C     THIS TARGET IS IN SECONDARY ZONE
       IF (PRIOR.GT.SEC) THEN
          SEC=PRIOR
          NTGT2=J
          MYWP2=IWEAP
          TRNG2=RANGE
          IF (IWEAP.EQ.1) PKWP2=PKIT1
          IF (IWEAP.EQ.2) PKWP2=PKIT2
          IF (IWEAP.EQ.3) PKWP2=PKIT3
       ENDIF
      ENDIF
          IF (INZONE.EQ.3) THEN
C     THIS TARGET IS IN 360 DEGREE ZONE
       IF (PRIOR.GT.THIRD) THEN
          THIRD=PRIOR
          NTGT3=J
          MYWP3=IWEAP
          TRNG3=RANGE
          IF (IWEAP.EQ.1) PKWP3=PKIT1
          IF (IWEAP.EQ.2) PKWP3=PKIT2
          IF (IWEAP.EQ.3) PKWP3=PKIT3
       ENDIF
      ENDIF

1143  CONTINUE
C     TARGET WAS SKIPPED TO LINE 1143 SINCE IT WAS AT A QUESTIONABLE
C     EFFECTIVE RANGE OR INSUFFICIENT AMMO TO RISK MISSING THE SHOT

C     WRITE (*,*) 'PRI',PRI,'NTGT1',NTGT1

      ENDIF
C                              END LOOP 5
C     END OF INZONE

1149  CONTINUE
C                              END LOOP 4
C     END OF FINDING A TARGET FOR THIS RED  SYSTEM
C
C     IF THE SECONDARY TARGET HAS 125% OF THE PRIORITY
C     OF THE PRIMARY TARGET, SHOOT AT THE SECONDARY TARGET
      ITGTR(K,1)=0
      ITGTR(K,2)=0
      PKTR(K,1)=0.0
      PKTR(K,2)=0.0
      TRNG=0.0
      MYWP=0
      IF (SEC.GE.(1.25*PRI)) THEN
C     SHOOT AT TARGET IN SECONDARY ZONE SINCE IT IS COMPELLING
      ITGTR(K,1)=NTGT2
      ITGTR(K,2)=MYWP2
      PKTR(K,1)=PKWP2
      PKTR(K,2)=TRNG2
      TRNG=TRNG2
      MYWP=MYWP2
      ELSE
C     STAY WITH PRIMARY ZONE TARGET
      ITGTR(K,1)=NTGT1
      ITGTR(K,2)=MYWP1
      PKTR(K,1)=PKWP1
      PKTR(K,2)=TRNG1
      TRNG=TRNG1
      MYWP=MYWP1
      ENDIF
C     IF THERE ARE NO TARGETS IN PRIMARY AND SECONDARY,
C     SHOOT AT THIRD ZONE TARGET (360 DEGREES)
      IN3=0
      IF (NTGT1.EQ.0.AND.NTGT2.EQ.0) THEN
      ITGTR(K,1)=NTGT3
      ITGTR(K,2)=MYWP3
      PKTR(K,1)=PKWP3
      PKTR(K,2)=TRNG3
      TRNG=TRNG3
      MYWP=MYWP3
      IF (NTGT3.NE.0) IN3=1
      ENDIF
C
C     FOR BATTLE MANAGEMENT, CHECK TO SEE IF MORE THAN 4 RED  SYSTEMS
C     ARE ALREADY SERVICING THIS TARGET, IF SO, SKIP IT.
C
      IF (IAMB(ITGTR(K,1)).EQ.4) THEN
       ITGTR(K,1)=0
       ITGTR(K,2)=0
       PKTR(K,1)=0.0
       PKTR(K,2)=0.0
       TRNG=0.0
       MYWP=0
      ELSE
       IAMB(ITGTR(K,1))=IAMB(ITGTR(K,1))+1
      ENDIF
C

C     WRITE (*,*) 'PRIMARY ZONE TARGET',ITGTR(K,1)

C     EVALUATE TARGET SEARCH RESULTS AND
C     CHANGE MODE FOR THIS BLUE SYSTEM ACCORDINGLY
      STIME=0.0
      MODEN=0
      IF (ITGTR(K,1).EQ.0) THEN
C                                    LOOP 7
C     NO TARGETS, WAIT MODE
C     WAIT A COMPLETE SEARCH CYCLE TIME FOR DUTY CYCLE ACCOUNTABILITY
C     BUT CREW IS STILL SEARCHING DURING THIS TIME
       MODEN=5
       MODEFR(K)=MODEN
C      DETERMINE END TIME OF SEARCH MODE CYCLE
C     WRITE (*,*) 'SEARCH RATE RED',SRCHR(NSYS)
       STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHR(NSYS)
       TRMFR(K)=ITIME+IDTIME+STIME
C       WRITE (*,*) 'RED  ',K,' NO TARGETS '
      ELSE

C     THERE IS A TARGET WITHIN EFFECTIVE RANGE

C     SET TURRET AZIMUTH
      CALL TURAZ (2,K,ITGTR(K,1),AZTR(K))

C     SINCE A TARGET WAS FOUND, CREW GOES INTO QUICK SEARCH
C     MODE FOR NEXT SEARCH CYCLE, FOCUSING INITIALLY ON TARGETS
C     WITHIN A .5 SECOND TRAVERSE FROM THIS TARGET.
C     RESET PRIMARY SEARCH ZONE AND SET SECONDARY TO 0,0
      NSYS=IRSYS(K,1)
      IRSECOND(K,1)=0
      IRSECOND(K,2)=0
      ALFT1=AZTR(K)-TRATER(NSYSY2.0
      ARGT1=AZTR(K)+TRATER(NSYS)/2.0
      IF (ALFT1.LT.0.0) ALFT1=ALFT1+360.0
      IF (ARGT1.GT.360.0) ARGT1=ARGT1-360.0
      IRPRIME(K,1)=ALFT1
      IRPRIME(K,2)=ARGT1
      ALFT2=0.0
      ARGT2=0.0
      IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
      IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.

C     CHECK MODES FOR ENGAGEMENT, BALKING AND ENERGY WAIT
       BALK=RND()
       IF (BALK.GT.0.975) THEN
C      BALKING, WAIT SEARCH TIME, BUT STILL SEARCH
        MODEN=4
        MODEFR(K)=MODEN
C       DETERMINE END TIME OF BALKING MODE CYCLE
       STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHR(NSYS)
        TRMFR(K)=ITIME+IDTIME+STIME
1131   FORMAT (' RED  ',I3,' BALKS AT BLUE ',I3)
C        WRITE (*,1131) K,ITGTR(K,1)
        WRITE (10,1131) K,ITGTR(K,1)

       ENDIF
C     EVALUATE EM ENERGY REQUIREMENTS FOR BLUE SYSTEM (MODE 3)
C     (NOT DONE YET)
C
C     IF NOTHING ELSE, THEN ENGAGE THE TARGET
        IF (MODEN.EQ.0) THEN
```

```
      MODEN=2
      MODEFR(IK)=MODEN
C      DETERMINE END TIME OF ENGAGMENT MODE CYCLE
C      BASED ON INTERPOLATING BETWEEN MIN RANGE AND MAX RANGE
C      ENGAGEMENT TIMES
      IST=(NSYS-1)*3+MYWP
      NRG=NRWR(NSYS,MYWP)
      ISRM=(NSYS-1)*18+(MYWP-1)*6+NRG
      ISR1=(NSYS-1)*18+(MYWP-1)*6+1
      RMIN=RR(ISR1)
      RMAX=RR(ISRM)
      TMIN=TEMINR(IST)
      TMAX=TEMAXR(IST)
      CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
C     IF TARGET TO BE ENGAGED IS IN ZONE 3 (360 DEGREES),
C     MUST ADD THIS EXTRA SEARCH TIME TO THE ENGAGEMENT TIME
      STIME3=360.0/SRCHR(NSYS)*IN3
      TRMFR(K)=ITIME+IDTIME+STIME+STIME3
1132  FORMAT (' RED ',I3,' ENGAGES BLUE ',I3,' WEAPON ',I3,
     & ' RANGE ',F6.0)
C      WRITE (*,1132) K,ITGTR(K,1),MYWP,TRNG*1000.
      WRITE (10,1132) K,ITGTR(K,1),MYWP,TRNG*1000.
      ENDIF
     ENDIF
C                                  END LOOP 7
C     END OF TARGET SEARCH EVALUATION RESULTS

     ENDIF
C                                  END LOOP 3
C     END OF IT'S TIME TO SELECT A TARGET
C
     ENDIF
C                                  END LOOP 2
C     END OF SEARCH MODE
C
      IF (MODEF.EQ.2.OR.MODEF.EQ.6.OR.MODEF.EQ.9.OR.
     & MODEF.EQ.8) THEN
C                                  LOOP 8
C     ENGAGEMENT MODE
C     CHECK FOR MODE CHANGE AT END OF SUB-CYCLE
      TCHANGE=TRMFR(IK)
      TNEXT=ITIME+IDTIME
      IF (TCHANGE.LE.TNEXT) THEN
C                                  LOOP 9
C      IT'S TIME TO KILL THE TARGET
C
C     RED EXPENDS ONE SHOT
      IAMOUT=0
      JKILL=0
      NWP=ITGTR(K,2)
      NTG=ITGTR(K,1)
      TRNG=PKTR(K,2)
      NSHOTR(K,NWP)=NSHOTR(K,NWP)-1
      IF (NSHOTR(K,NWP).EQ.0) IAMOUT=1
      IAMB(NTG)=IAMB(NTG)-1
      IF (IAMB(NTG).LE.0) IAMB(NTG)=0
C
C     FIND SEARCH AREAS FOR RE-MODING INFORMATION
      NSYS=IRSYS(K,1)
      ALFT1=IRPRIME(K,1)
      ARGT1=IRPRIME(K,2)
      ALFT2=IRSECOND(K,1)
      ARGT2=IRSECOND(K,2)
      IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
      IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
      IF (IRPRIME(K,1).EQ.IRSECOND(K,1).AND.IRPRIME(K,2).
     &    EQ.IRSECOND(K,2)) THEN
      ALFT2=-1.0
      ARGT2=-1.0
      ENDIF
C     DETERMINE END TIME OF SEARCH MODE CYCLE
      STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHR(NSYS)
C     RE-EVALUATE MASKING AND OBSCURANT
      PROB=RND()
           PKIT=PKTR(K,1)
C
C     CHECK BLUE TARGET FOR ACTIVE PROTECTION
           CALL CHECKAPS (2,NSYS,NTG,NWP,PKIT,IFAPS,TRNG)
        XR=RXYA(K,1)
        YR=RXYA(K,2)
        XB=BXYA(NTG,1)
        YB=BXYA(NTG,2)
C        DRAW LINE FROM SHOOTER TO TARGET
        CALL PLINE (XR,YR,XB,YB,2)
      IF (PROB.LE.PKIT) THEN
C                                  LOOP 10
C                        KILL TARGET AND RE-MODE
C     RE-EVALUATE MASKING AND OBSCURANTS
C     TERRAIN MASKING
      JSKIP=1
      IF (ITER.EQ.1) CALL TERRMASK (XB,YB,XR,YR,JSKIP)

C     IF THIS IS A MAN IN THE LOOP (MIL) WEAPON, THE TARGET
C     MUST STILL BE ABLE TO BEEN SEEN AND TRACKED IN ORDER TO
C     BE HIT.  EVALUATE CURRENT DUST AND ENGINE SMOKE SITUATION.
C
C     EVALUATE M-I-L OR F-A-F
      MILTYPE=MILR(NSYS,NWP)
      IF (MILTYPE.EQ.1) THEN
C     THIS WEAPON IS M-I-L, RE-EVALUATE OBSCURANTS FOR MASKING

C     IF THIS RED  SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION
C      OR TRACKING
C      FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C      RE-EVALUATED
      RANGE=TRNG*1000.
      ITRK1=ITRKR(NSYS,1)
      ISROF=(NSYS-1)*3+NWP
      IRNG1=IRNGR(ISROF,1)
      ICANRD=1
      IF (IRNG1.EQ.0) ICANRD=0
      IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
     & CALL DMASK (XR,YR,RANGE,0.0,0.0)
C
     CALL COOKMASK (XR,YR,RANGE,0.0,0.0)
C
C      IF THIS RED  SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C      TRACKING OR RANGING,
C      FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C      CAN BE RE-EVALUATED
      ITRK2=ITRKR(NSYS,2)
      ISROF=(NSYS-1)*3+NWP
      IRNG2=IRNGR(ISROF,2)
      ICANRE=1
      IF (IRNG2.EQ.0) ICANRE=0
      IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
     & CALL ESMASK (XR,YR,RANGE,0.0,0.0)
C
C      IF THIS RED  SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C      TRACKING OR RANGING,
C      FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C      CAN BE RE-EVALUATED
      ITRK3=ITRKR(NSYS,3)
      ISROF=(NSYS-1)*3+NWP
      IRNG3=IRNGR(ISROF,3)
      ICANRE=1
      IF (IRNG3.EQ.0) ICANRS=0
      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
     & CALL GRMASK (XR,YR,RANGE,0.0,0.0)
C
C     IF THIS RED  SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C      TRACKING OR RANGING,
C      FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
MASKING
C      CAN BE RE-EVALUATED
     ITRK4=ITRKR(NSYS,4)
     ISROF=(NSYS-1)*3+NWP
     IRNG4=IRNGR(ISROF,4)
     ICANRA=1
     IF (IRNG4.EQ.0) ICANRA=0
     IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
     & CALL ASMASK (XR,YR,RANGE,0.0,0.0)
C
C      DUST MASKING
C     RE-EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET

C      FIND ANGLE TO THIS TARGET
      TOP=YB-YR
      BOT=XB-XR
       IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
      IF (BOT.LT.0.0) THEN
           IF (TOP.LT.0.0) DIR=DIR+180.
           IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
           IF (TOP.LT.0.0) DIR=360.-DIR
      ENDIF
C      CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
      CALL FAZ (DIR,AZIN)
C
      JDUST=1
      IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
     & CALL DSEE (XR,YR,XB,YB,AZIN,JDUST)
C
C     BURNING VEHICLE MASKING
     CALL COOKSEE (XR,YR,XB,YB,AZIN,JSKIP)

C      IF THERE IS DUST IN THE WAY AND RED  CANNOT SEE OR
C      TRACK THROUGH IT, THE TARGET IS MISSED
      IF (JDUST.EQ.0.AND.ITRK1.EQ.0) JSKIP=0

C     IF JSKIP=1 THERE MAY STILL BE DUST IN THE WAY, BUT RED  CAN SEE
C     THROUGH IT.  THE QUESTION REMAINS WHETHER RED  CAN TRACK THROUGH
C     THE DUST.

      IF (JDUST.EQ.0.AND.ICANRD.EQ.0) JSKIP=0

C      IF JSKIP=1, THERE IS NO DUST OR TRACKING PROBLEM

C      ENGINE SMOKE MASKING
C     RE-EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
      JENG=1
      IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
     & CALL ESSEE (XR,YR,XB,YB,AZIN,JENG)
      IF (JENG.EQ.0.AND.ITRK2.EQ.0) JSKIP=0
      IF (JENG.EQ.0.AND.ICANRE.EQ.0) JSKIP=0

C     SMOKE GRENADE CURTAIN MASKING
C     RE-EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
      JSMK=1
      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
     & CALL GRSEE (XR,YR,XB,YB,AZIN,JSMK)
      IF (JSMK.EQ.0.AND.ITRK3.EQ.0) JSKIP=0
```

```
      IF (JSMK.EQ.0.AND.ICANRS.EQ.0) JSKIP=0

C     ARTILLERY SMOKE CURTAIN MASKING
C     RE-EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS
TARGET
            JART=1
            IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
     & CALL ASSEE (XR,YR,XB,YB,AZIN,JART)
            IF (JART.EQ.0.AND.ITRK4.EQ.0) JSKIP=0
            IF (JART.EQ.0.AND.ICANRA.EQ.0) JSKIP=0

      ENDIF
C     END OF OBSCURANT RE-EVALUATION FOR M-H-L WEAPON

      IF (JSKIP.EQ.1) THEN
C
C     BLUE TARGET KILLED
C     TALLY BLUE KILLS FOR RESOLUTION AFTER THIS SUB-CYCLE
C     JUST IN CASE THIS BLUE TARGET IS ABOUT TO KILL A RED
C     TARGET
       IKILLR(NTG)=1
       JKILL=1
C     RE-MODE TO SEARCH
       MODEN=1
1133   FORMAT (' RED ',I3,' KILLS BLUE ',I3)
C        IF (IBSYS(NTG,2).NE.0) WRITE (*,1133) K,NTG
C      IF (IAMOUT.EQ.1.AND.MODEF.NE.9)
C     & WRITE (*,*) 'RED ',K,' WEAPON ',NWP,' OUT OF AMMO'
        IF (IBSYS(NTG,2).NE.0) WRITE (10,1133) K,NTG
      IF (IAMOUT.EQ.1.AND.MODEF.NE.9)
     & WRITE (10,*) 'RED ',K,' WEAPON ',NWP,' OUT OF AMMO'
      IAMOUT=0
C     IF (IBSYS(NTG,2).NE.0) CALL OUTPUT2 (ITIME,1,NTG,0)

      ELSE

C       SOME SORT OF MASKING GOT IN THE WAY OF M-H-L SYSTEM
C       OR TERRAIN GOT IN THE WAY FOR F-A-F WEAPON WHILE TARGET
C       WAS MOVING
        IF (IBSYS(NTG,2).NE.0) WRITE (10,1134) K,NTG
      IF (MODEF.EQ.9) GO TO 1135
      IF (IAMOUT.EQ.1.AND.MODEF.NE.9)
     & WRITE (10,*) 'RED ',K,' WEAPON ',NWP,' OUT OF AMMO'
      IAMOUT=0
        WRITE (10,*) 'RED ',K,' SEARCHES FOR A BETTER TARGET'
       MODEN=1

      ENDIF

     ELSE

1134    FORMAT (' RED ',I3,' MISSES BLUE ',I3)
C        IF (IBSYS(NTG,2).NE.0) WRITE (*,1134) K,NTG
        IF (IBSYS(NTG,2).NE.0) WRITE (10,1134) K,NTG
C      IF (IAMOUT.EQ.1.AND.MODEF.NE.9)
C     & WRITE (*,*) 'RED ',K,' WEAPON ',NWP,' OUT OF AMMO'
      IF (IAMOUT.EQ.1.AND.MODEF.NE.9)
     & WRITE (10,*) 'RED ',K,' WEAPON ',NWP,' OUT OF AMMO'
      IAMOUT=0
      IF (MODEF.EQ.9) GO TO 1135
C     RE-MODE RED SYSTEM AFTER MISS
C     THERE ARE TWO OPTIONS -- 1 RE-ENGAGE
C                           -- 2 RE-SEARCH
C     RE-ENGAGE BASED ON AVAILABLE AMMUNITION, AND THE Pk
C     RE-SEARCH IF DECISION IS TO NOT RE-ENGAGE
            PKIT=PKTR(K,1)
      NSHOTS=NSHOTR(K,NWP)
      SKILL=LOG(.01)/LOG(1.-PKIT)
           IF (NSHOTS.EQ.0) MODEN=1
C          IF (SKILL.GT.16.0) MODEN=1
C           IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
C           IF (SKILL.LE.16.0.AND.NSHOTS.NE.0) THEN
C           IF (SKILL.GT.7.0.AND.NSHOTS.NE.0) THEN
           IF (NSHOTS.NE.0) THEN
               PSHOOT=NSHOTS/SKILL
               PNO=RND()
C               IF (SKILL.LE.7) PNO=0.0
                                  IF (SKILL.LE.45) PNO=0.0
               IF (PNO.GE.PSHOOT.OR.MODEF.EQ.8) THEN
                     MODEN=1
C       WRITE (*,*) 'RED ',K,' SEARCHES FOR A BETTER TARGET'
        WRITE (10,*) 'RED ',K,' SEARCHES FOR A BETTER TARGET'
                ELSE
                  IF (IBSYS(NTG,2).NE.0) THEN
C                   RE-ENGAGE THE TARGET
                                                MODEN=6
C       WRITE (*,*) 'RED ',K,' RE-ENGAGES BLUE ',NTG
C        WRITE (10,*) 'RED ',K,' RE-ENGAGES BLUE ',NTG
C       DETERMINE END TIME OF ENGAGMENT MODE CYCLE
C       BASED ON INTERPOLATING BETWEEN MIN RANGE AND MAX RANGE
C        ENGAGEMENT TIMES
        TRNG=PKTR(K,2)
        IST=(NSYS-1)*3+NWP
        NRG=NRWR(NSYS,NWP)
        ISRM=(NSYS-1)*18+(NWP-1)*6+NRG
        ISR1=(NSYS-1)*18+(NWP-1)*6+1
        RMIN=RR(ISR1)
        RMAX=RR(ISRM)
        TMIN=TEMINR(IST)
        TMAX=TEMAXR(IST)
        CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
                  ELSE
                    MODEN=1
                  ENDIF
                ENDIF
           ENDIF
1135   CONTINUE
      ENDIF
C                              END LOOP 10
C     END OF TARGET KILLED AND SYSTEM RE-MODED

      TRMFR(K)=ITIME+IDTIME+STIME
      MODEFR(K)=MODEN
      IF (MODEF.EQ.9) MODEFR(K)=-9

C     WRITE SHOT OUTPUT INFORMATION
    IF (IBSYS(NTG,2).EQ.0) THEN
      JKILL=-1
      IF (IFAPS.NE.-1) IFAPS=-3
    ENDIF
      CALL OUTPUT1 (ITIME,2,K,NSYS,NWP,0,TRNG,NTG,IBSYS(NTG,1),
     &              IFAPS,JKILL)
C

      ELSE
C      END OF IT'S TIME TO KILL THE TARGET
C
C     CHECK TO SEE IF TARGET WAS KILLED BY ANOTHER WEAPON DURING
C      THIS ENGAGEMENT PERIOD.  IF IT WAS, CHECK TO SEE IF THIS WEAPON
C     CAN REDESIGNATE.
C
      NTG=ITGTR(K,1)
      IF (IBSYS(NTG,2).EQ.0) THEN
C       THIS TARGET HAS JUST BEEN KILLED, TRY TO RE-DESIGNATE
C       IF THIS SYSTEM JUST RE-DESIGNATED, HOWEVER, MUST WAIT
C       3 SECOND RE-DESIGNATION TIME PERIOD BEFORE DOING IT AGAIN
       IF (MODEF.EQ.8) THEN
        TCAN=TRDESR(K)
       ELSE
        TCAN=0.0
       ENDIF
       NWP=ITGTR(K,2)
       NSYS=IRSYS(K,1)
       MILTYPE=MILR(NSYS,NWP)
        IF (MILTYPE.EQ.1.AND.ITIME.GE.TCAN) THEN
C        THIS WEAPON IS MAN-IN-LOOP, IT CAN RE-DESIGNATE, BUT NEEDS
C       AT LEAST 3 SECONDS FOR GUNNER TO OVERCOME HIS SURPRISE AND
C        QUICKLY ASSESS THE SITUATION, THEN FIND ANOTHER TARGET.
C        FIND TIME TO IMPACT
        TRDESR(K)=ITIME+3.0
        THIT=TRMFR(K)-ITIME
         IWANT=0
         IF (THIT.GT.3.0) THEN
C         THERE IS ENOUGH TIME
          THIT=THIT-3.0
          TRNG=PKTR(K,2)
C         DETERMINE ENGAGMENT MODE CYCLE TIME TO THIS TARGET
C         BASED ON INTERPOLATING BETWEEN MIN RANGE AND MAX RANGE
C         ENGAGEMENT TIMES AND TARGET RANGE
          MYWP=NWP
          IST=(NSYS-1)*3+MYWP
          NRG=NRWR(NSYS,MYWP)
          ISRM=(NSYS-1)*18+(MYWP-1)*6+NRG
          ISR1=(NSYS-1)*18+(MYWP-1)*6+1
          RMIN=RR(ISR1)
          RMAX=RR(ISRM)
          TMIN=TEMINR(IST)
          TMAX=TEMAXR(IST)
          CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
C         FIND DISTANCE MISSILE IS FROM TARGET
          DMSL=TRNG/STIME*THIT
C         MISSILE CAN BE RE-DESIGNATED TO FIND ANOTHER RED
C         TARGET WITHIN THIS DISTANCE FROM THE OLD TARGET
C         BUT NOT BEYOND SYSTEM MAXIMUM RANGE.
C
C         FIND CLOSEST OTHER RED TARGET
C         CURRENT TARGET X,Y POSITION
          XT=BXYA(NTG,1)
          YT=BXYA(NTG,2)
          XB=RXYA(K,1)
          YB=RXYA(K,2)
C         SEARCH ALL RED TARGETS FOR CLOSEST
          DRMIN=1.0E12
C       FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
C       SO THAT MASKING CAN BE EVALUATED
C
          KSIDE=2
          MYSIDE=2
          IRED=K
         CALL VMASK (KSIDE,IRED)

C       EVALUATE OBSCURANTS ALSO
C        IF THIS RED SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION
C        OR TRACKING
C       FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C       RE-EVALUATED
        RANGE=TRNG*1000.*1.1
        ITRK1=ITRKR(NSYS,1)
        ISROF=(NSYS-1)*3+NWP
        IRNG1=IRNGR(ISROF,1)
        ICANRD=1
        IF (IRNG1.EQ.0) ICANRD=0
       IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
     & CALL DMASK (XB,YB,RANGE,0.0,0.0)
```

```
C
            CALL COOKMASK (XB,YB,RANGE,0.0,0.0)
C
C     IF THIS RED  SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C     CAN BE RE-EVALUATED
      ITRK2=ITRKR(NSYS,2)
      ISROF=(NSYS-1)*3+NWP
      IRNG2=IRNGR(ISROF,2)
      ICANRE=1
      IF (IRNG2.EQ.0) ICANRE=0
      IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
     & CALL ESMASK (XB,YB,RANGE,0.0,0.0)
C
C     IF THIS RED  SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C     TRACKING OR RANGING,
C     FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C     CAN BE RE-EVALUATED
      ITRK3=ITRKR(NSYS,3)
      ISROF=(NSYS-1)*3+NWP
      IRNG3=IRNGR(ISROF,3)
      ICANRS=1
      IF (IRNG3.EQ.0) ICANRS=0
      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
     & CALL GRMASK (XB,YB,RANGE,0.0,0.0)
C
C     IF THIS RED  SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C     TRACKING OR RANGING,
C     FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C     CAN BE RE-EVALUATED
            ITRK4=ITRKR(NSYS,4)
            ISROF=(NSYS-1)*3+NWP
            IRNG4=IRNGR(ISROF,4)
            ICANRA=1
            IF (IRNG4.EQ.0) ICANRA=0
            IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
     & CALL ASMASK (XB,YB,RANGE,0.0,0.0)
C
         DO 111 JBLUE=1,NR
          IF (IBSYS(JBLUE,2).EQ.0) GO TO 111
          XNT=BXYA(JBLUE,1)
          YNT=BXYA(JBLUE,2)
         DNT=((XNT-XT)**2+(YNT-YT)**2)**.5
         DNB=((XNT-XB)**2+(YNT-YB)**2)**.5
          RANGE=DNB
C         CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
C         NOT DETECTED
         CALL CHECKSIG (2,K,JBLUE,RANGE,JSKIP)
          IF (JSKIP.EQ.0) GO TO 111
           IF (DNB.LE.RMAX.AND.DNT.LE.DMSL) THEN
C           DIVIDE BY A RANDOM FACTOR SO EVERYONE DOES NOT
C           TAKE THE SAME ALTERNATE TARGET
           RRR=RND()
           DNT=DNT/RRR
            IF (DNT.LT.DRMIN) THEN
C            EVALUATE MASKING
C     FIND ANGLE TO THIS TARGET
      TOP=YNT-YB
      BOT=XNT-XB
       IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
      IF (BOT.LT.0.0) THEN
            IF (TOP.LT.0.0) DIR=DIR+180.
            IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
            IF (TOP.LT.0.0) DIR=360.-DIR
      ENDIF
C     CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
      CALL FAZ (DIR,AZIN)
C
C     AT THIS POINT EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE
C     MASKING AND OBSCURANTS AND DROP THIS
C     TARGET IF UNOBSERVABLE (TERRAIN / SMOKE NOT DONE YET)
      JSKIP=1
C     TERRAIN MASKING
      IF (ITER.EQ.1) CALL TERRMASK (XNT,YNT,XB,YB,JSKIP)
      IF (JSKIP.EQ.0) GO TO 111
            CALL VSEE (MYSIDE,K,JBLUE,RANGE,AZIN,JSKIP,NB,NR)
            CALL COOKSEE (XB,YB,XNT,YNT,AZIN,JSKIP)
      IF (JSKIP.EQ.0) GO TO 111

C     DUST MASKING
C     RE-EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET

      JDUST=1
      IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
     & CALL DSEE (XB,YB,XNT,YNT,AZIN,JDUST)

C     IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE OR
C     TRACK THROUGH IT, THE TARGET IS MISSED
      IF (JDUST.EQ.0.AND.ITRK1.EQ.0) JSKIP=0

C     IF JSKIP=1 THERE MAY STILL BE DUST IN THE WAY, BUT BLUE CAN SEE
C     THROUGH IT.  THE QUESTION REMAINS WHETHER BLUE CAN TRACK THROUGH
C     THE DUST.

      IF (JDUST.EQ.0.AND.ICANRD.EQ.0) JSKIP=0

C     IF JSKIP=1, THERE IS NO DUST OR TRACKING PROBLEM

C     ENGINE SMOKE MASKING
C     RE-EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
      JENG=1
      IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
     & CALL ESSEE (XB,YB,XNT,YNT,AZIN,JENG)
      IF (JENG.EQ.0.AND.ITRK2.EQ.0) JSKIP=0
      IF (JENG.EQ.0.AND.ICANRE.EQ.0) JSKIP=0

C     SMOKE GRENADE CURTAIN MASKING
C     RE-EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
      JSMK=1
      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
     & CALL GRSEE (XB,YB,XNT,YNT,AZIN,JSMK)
      IF (JSMK.EQ.0.AND.ITRK3.EQ.0) JSKIP=0
      IF (JSMK.EQ.0.AND.ICANRS.EQ.0) JSKIP=0

C     ARTILLERY SMOKE CURTAIN MASKING
C     RE-EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS
TARGET
      JART=1
      IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
     & CALL ASSEE (XB,YB,XNT,YNT,AZIN,JART)
      IF (JART.EQ.0.AND.ITRK4.EQ.0) JSKIP=0
      IF (JART.EQ.0.AND.ICANRA.EQ.0) JSKIP=0

C     END OF OBSCURANT RE-EVALUATION FOR M-I-L WEAPON

      IF (JSKIP.EQ.0) GO TO 111

C     TARGET CAN BE SEEN AND IS CLOSER TO OLD TARGET
               DRMIN=DNT
               IWANT=JBLUE
               RNG=DNB
             ENDIF
            ENDIF
111        CONTINUE
          ENDIF
            IF (IWANT.NE.0) THEN
C            THERE IS ANOTHER TARGET TO DESIGNATE
C FIND PROBABILTY OF KILLING THIS NEW BLUE TARGET
            NITSYS=IBSYS(IWANT,1)
       CALL FINDPK (KSIDE,NSYS,NITSYS,NWP,NRNG,RNG,PKIT,IWANT)
        PKTR(K,1)=PKIT
        PKTR(K,2)=RNG
             MODEN=8
             STIME=THIT+3.0
             ITGTR(K,1)=IWANT
C              TRMFR(K)=ITIME+IDTIME+STIME
C             MODEFR(K)=MODEN
C             WRITE (*,*) 'RED  ',K,' RE-DESIGNATES AT BLUE ',IWANT
             WRITE (10,*) 'RED  ',K,' RE-DESIGNATES AT BLUE ',IWANT
C
                ELSE

C            NO TARGETS, FINISH OUT CURRENT ENGAGEMENT MODE TIME
C            THEN RE-SEARCH
             MODEN=1
C            RED  EXPENDS ONE SHOT
             NWP=ITGTR(K,2)
             NTG=ITGTR(K,1)
             TRNG=PKTR(K,2)
            NSHOTR(K,NWP)=NSHOTR(K,NWP)-1
             IAMB(NTG)=IAMB(NTG)-1
             IF (IAMB(NTG).LT.0) IAMB(NTG)=0
C
C            FIND SEARCH AREAS FOR RE-MODING INFORMATION
            NSYS=IRSYS(K,1)
            ALFT1=IRPRIME(K,1)
            ARGT1=IRPRIME(K,2)
            ALFT2=IRSECOND(K,1)
            ARGT2=IRSECOND(K,2)
             IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
             IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
            IF (IRPRIME(K,1).EQ.IRSECOND(K,1).AND.IRPRIME(K,2).
     &       EQ.IRSECOND(K,2)) THEN
             ALFT2=-1.0
             ARGT2=-1.0
             ENDIF
C            DETERMINE END TIME OF SEARCH MODE CYCLE
            STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHR(NSYS)
            ENDIF

             TRMFR(K)=ITIME+IDTIME+STIME
            MODEFR(K)=MODEN

C     WRITE SHOT OUTPUT INFORMATION
      JKILL=-1
      IFAPS=-1
       CALL OUTPUT1 (ITIME,2,K,NSYS,NWP,0,TRNG,NTG,IBSYS(NTG,1),
     &             IFAPS,JKILL)
C
          ELSE
C         IF (IAMOUT.EQ.1)
C     &  WRITE (*,*) 'RED ',K,' WEAPON ',NWP,' OUT OF AMMO'
           IF (IAMOUT.EQ.1)
     &  WRITE (10,*) 'RED ',K,' WEAPON ',NWP,' OUT OF AMMO'
           IAMOUT=0
            ENDIF
C           END OF IT CAN BE RE-DESIGNATED
          ENDIF
C          END OF THIS TARGET HAS JUST BEEN KILLED
C
```

```
      ENDIF
C                                END LOOP 9
C     END OF IT'S TIME TO KILL THE TARGET

      ENDIF
C                                END LOOP 8
C     END OF ENGAGEMENT MODE

1150  CONTINUE
C                              END LOOP 1
C     END OF RED  SYSTEM FIRE MODE LOOP
C
C     RESOLVE ENGAGEMENTS TO SEE WHO GOT KILLED THIS 60-CYCLE PERIOD
C
      DO 900 KILLS=1,NR
            IF (IKILLB(KILLS).EQ.1) THEN
             IRSYS(KILLS,2)=0
             CALL OUTPUT2 (ITIME,2,KILLS,0)
C       EVALUATE WHETHER THIS VEHICLE BURNS AND BECOMES AN OBSCURANT
             CALL COOKS (2,KILLS)
             IKILLB(KILLS)=0
            ENDIF
900   CONTINUE
      DO 901 KILLS=1,NB
            IF (IKILLR(KILLS).EQ.1) THEN
             IBSYS(KILLS,2)=0
             CALL OUTPUT2 (ITIME,1,KILLS,0)
             CALL COOKS (1,KILLS)
             IKILLR(KILLS)=0
            ENDIF
901   CONTINUE
C
C     EVALUATE MINEFIELD KILLS
            IF (NMINES.GT.0) THEN
            DO 902 MKILLS=1,NR
             IF (MINEK(MKILLS,2).GT.0) THEN
              IF (IRSYS(MKILLS,2).NE.0) CALL OUTPUT2 (ITIME,2,MKILLS,0)
              IRSYS(MKILLS,2)=0
              CALL WMINE (ITIME,2,MKILLS,MINEK(MKILLS,2))
              CALL OUTPUT2 (ITIME,2,MKILLS,0)
              NAZMINE(MINEK(MKILLS,2),3)=1
              WRITE (10,*) ' RED ',MKILLS,' KILLED BY MINEFIELD ',
     &                          MINEK(MKILLS,2)
              MINEK(MKILLS,2)=0
             ENDIF
902   CONTINUE
            DO 903 MKILLS=1,NB
             IF (MINEK(MKILLS,1).GT.0) THEN
              IF (IBSYS(MKILLS,2).NE.0) CALL OUTPUT2 (ITIME,1,MKILLS,0)
              IBSYS(MKILLS,2)=0
              CALL WMINE (ITIME,1,MKILLS,MINEK(MKILLS,1))
              CALL OUTPUT2 (ITIME,1,MKILLS,0)
              NAZMINE(MINEK(MKILLS,1),3)=1
              WRITE (10,*) ' BLUE ',MKILLS,' KILLED BY MINEFIELD ',
     &                          MINEK(MKILLS,1)
              MINEK(MKILLS,1)=0
             ENDIF
903   CONTINUE
            ENDIF

C     END OF RESOLVING ENGAGEMENTS
C
      IF (ITIME.GE.TPUFF) THEN
C     MOVE OBSCURANTS
      CALL MPUFFS
C     DELETE DISSIPATED OBSCURANTS
      CALL DELPUFFS
C     CREATE NEW OBSCURANTS
      CALL NEWPUFFS
C     ERASE OLD OBSCURANT POSITIONS
      CALL ERASEPUFFS
C     ERASE OLD SMOKE CURTAIN POSITIONS
      CALL ERSMKCURT
       TPUFF=DTPUFF+ITIME
      ENDIF
C
C     CHARGE EM SYSTEMS
            CALL EMCHARGE
C
            ITIME=ITIME+IDTIME
C       WRITE (*,*) 'TIME ',ITIME
      WRITE (10,*) 'TIME ',ITIME
C
C     ERASE OLD UNIT POSITIONS
             IF (ICYCLE.NE.60) CALL EUNITS
             CALL WOTHERS (BACEL)
1000  CONTINUE
C     PLOT ONE LAST TIME THE OBSCURANT POSITIONS
      IF (IDUST.EQ.1.AND.NDUST.GT.0) CALL PDUST
C     PLOT ENGINE SMOKE CLOUDS
      IF (NENG.GT.0) CALL PENG
C     PLOT GRENADE SMOKE CURTAINS
            IF (NSMK.GT.0) CALL PSMKCURTAINS
C     ARTILLERY SMOKE
            IF (NART.GT.0) CALL PARTSMK
C
C     ACCUMULATE MODE CYCLE TIMES
C
C     DO 1800 I=1,NB
C      TTMMB(I,1)=TTMMB(I,1)+TCMMB(I,1)
C      TTMMB(I,2)=TTMMB(I,2)+TCMMB(I,2)
C      TTMMB(I,3)=TTMMB(I,3)+TCMMB(I,3)
C      TTFMB(I,1)=TTFMB(I,1)+TCFMB(I,1)
C      TTFMB(I,2)=TTFMB(I,2)+TCFMB(I,2)
C      TTFMB(I,3)=TTFMB(I,3)+TCFMB(I,3)
C      TTFMB(I,4)=TTFMB(I,4)+TCFMB(I,4)
C      TTFMB(I,5)=TTFMB(I,5)+TCFMB(I,5)
C      TTFMB(I,6)=TTFMB(I,6)+TCFMB(I,6)
C      TTFMB(I,7)=TTFMB(I,7)+TCFMB(I,7)
C      TTFMB(I,8)=TTFMB(I,8)+TCFMB(I,8)
C      TTFMB(I,9)=TTFMB(I,9)+TCFMB(I,9)
C     IF (IBSYS(I,1).EQ.NBLUE) THEN
C      TTEM(I,1)=TTEM(I,1)+TEM(I,1)
C      TTEM(I,2)=TTEM(I,2)+TEM(I,2)
C     ENDIF

C1800  CONTINUE
C
C     DO 1850 I=1,NR
C      TTMMR(I,1)=TTMMR(I,1)+TCMMR(I,1)
C      TTMMR(I,2)=TTMMR(I,2)+TCMMR(I,2)
C      TTMMR(I,3)=TTMMR(I,3)+TCMMR(I,3)
C      TTFMR(I,1)=TTFMR(I,1)+TCFMR(I,1)
C      TTFMR(I,2)=TTFMR(I,2)+TCFMR(I,2)
C      TTFMR(I,3)=TTFMR(I,3)+TCFMR(I,3)
C      TTFMR(I,4)=TTFMR(I,4)+TCFMR(I,4)
C      TTFMR(I,5)=TTFMR(I,5)+TCFMR(I,5)
C      TTFMR(I,6)=TTFMR(I,6)+TCFMR(I,6)
C      TTFMR(I,7)=TTFMR(I,7)+TCFMR(I,7)
C      TTFMR(I,8)=TTFMR(I,8)+TCFMR(I,8)
C      TTFMR(I,9)=TTFMR(I,9)+TCFMR(I,9)
C1850  CONTINUE

C     WRITE BLUE SYSTEM #1 (EM TANK) MODE TIMES FOR THIS
C     SIXTY CYCLE PERIOD
      CALL WMODES

      READ (*,*)
      CALL PLOT (0.,0.,999)
      RETURN
      END


      SUBROUTINE TIME
       COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
1     WRITE (*,*)
      WRITE (*,*) 'CURRENT TIME STEP ',IDTIME
      WRITE (*,*) 'ENTER NEW TIME STEP (INTEGER)'
      WRITE (*,*)
      READ (*,*,ERR=1) DTIME
       IDTIME=INT(DTIME)
       WRITE (*,*) 'NEW TIME STEP ',IDTIME
      WRITE (*,*) '(RETURN TO CONTINUE)'
      READ (*,*)
      RETURN
      END


      SUBROUTINE FIREORS
C      FIREORS.FOR
C
C     ENTER FIRE CONTROL ORDERS FOR UNITS
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
C     COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
C    & NC(15),NCOL(15),JCON
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)
      COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/FIRE/IBPRIME(200,2),IBSECOND(200,2),IRPRIME(200,2),
     & IRSECOND(200,2),WRNGB(200,2),WRNGR(200,2)
      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
C      COMMON/TYPES/NBAT,NCOMP,ICOMP(10),NPLAT(10),IPLAT(10,10),
C     & NSQD(10,10),JPLAT,JSQD
C
      CHARACTER*9 BFNAME,RFNAME
C      CHARACTER*80 LINE
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*1 C2
C      CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
C      FIRE CONTROL ORDERS
C       WRITE (*,*) '3'
```

```
C      WRITE (*,*) (RB(ISR),ISR=1,36)
C      WRITE (*,*) (RR(ISR),ISR=1,36)
C
       KSIDE=0
10     WRITE (*,*)
       WRITE (*,*) 'FIRE CONTROL ORDERS'
       WRITE (*,*) 'enter BLUE or RED, or QUIT'
       WRITE (*,*)
       READ (*,*,ERR=10) C2
       IF (C2(1:1).EQ.'B') KSIDE=1
       IF (C2(1:1).EQ.'R') KSIDE=2
       IF (C2(1:1).EQ.'Q') GO TO 1000
       IF (KSIDE.EQ.0) GO TO 10
C
5      WRITE (*,*)
       WRITE (*,*) 'CURRENT STATUS'
11     FORMAT (1X,A80)
C      IF (C2(1:1).EQ.'B') WRITE (*,11) L1BM
C      IF (C2(1:1).EQ.'R') WRITE (*,11) L1RM
       WRITE (*,*) ' UNIT HIERARCHY    IDENTITY              PRIMARY
     &ZONE  SECONDARY ZONE '
       WRITE (*,*) '                                          left
     &right   left   right'
       IF (C2(1:1).EQ.'B') THEN
C      BLUE
       KK=0
            DO 130 K=1,NB
            IF (IBSYS(K,1).GT.10) GO TO 130
121    FORMAT (1X,I3,1X,A16,A24,4(I6,1X))
       IF (IBSYS(K,2).EQ.1) THEN
       WRITE (*,121) K,BHIER(K),BUNIT(K),IBPRIME(K,1),IBPRIME(K,2),
     & IBSECOND(K,1),IBSECOND(K,2)
       KK=KK+1
       ENDIF
       IF (KK.EQ.20) THEN
            READ (*,*)
            KK=0
       ENDIF
130    CONTINUE
C
       ELSE
C      RED
       KK=0
            DO 140 K=1,NR
            IF (IRSYS(K,1).GT.10) GO TO 140
       IF (IRSYS(K,2).EQ.1) THEN
       WRITE (*,121) K,RHIER(K),RUNIT(K),IRPRIME(K,1),IRPRIME(K,2),
     & IRSECOND(K,1),IRSECOND(K,2)
       KK=KK+1
       ENDIF
       IF (KK.EQ.20) THEN
            READ (*,*)
            KK=0
       ENDIF
140    CONTINUE
       ENDIF

141    WRITE (*,*) 'ENTER COMMAND UNIT (0-RELISTS -1-QUIT)'
       WRITE (*,*)
       READ (*,*,ERR=141) ICOMMAND
       IF (ICOMMAND.EQ.0) GO TO 5
       IF (ICOMMAND.EQ.-1) GO TO 10
C
142    WRITE (*,*) 'ENTER NUMBER RANGE OF SUBORDINATES FOR'
       WRITE (*,*) 'SUB-UNIT ALLOCATION (#START #END) (0 0 FOR NONE)'
       WRITE (*,*)
       READ (*,*,ERR=142) NSUB1,NSUB2
       NSUB=1
       IF (NSUB1.LE.0) THEN
            NSUB1=0
            NSUB2=0
            NSUB=0
       ENDIF
       IF (NSUB2.LE.0) THEN
            NSUB1=0
            NSUB2=0
            NSUB=0
       ENDIF
       IF (NSUB.EQ.0) GO TO 145
C
143    WRITE (*,*) 'SUB-UNIT ALLOCATIONS'
       WRITE (*,*) '1-ALL UNITS SEQUENTIAL INCLUDING COMMANDERS'
       WRITE (*,*) '  XOs  PLATOON LEADERS  AND SQUADS'
       WRITE (*,*) '2-ALL SUBORDINATES HAVE SAME ENTIRE SECTOR AS'
       WRITE (*,*) '  THE COMMANDER (EVERYONE SHOOTS IN SAME PLACE)'
C      WRITE (*,*) '2-UNIT COMMANDERS AND PLATOON LEADERS HAVE'
C      WRITE (*,*) '  THEIR WHOLE SUBORDINATE UNIT SECTOR'
       WRITE (*,*)
       READ (*,*,ERR=143) ISUB
       IF (ISUB.NE.1.AND.ISUB.NE.2) GO TO 143
C
145    WRITE (*,*)
       WRITE (*,*) 'ENTER PRIMARY ZONE AZIMUTHS (LEFT TO RIGHT)'
       WRITE (*,*) 'FOR COMMAND UNIT AND SECONDARY ZONE AZIMUTHS'
       WRITE (*,*) 'ENTER 0 0 IF NO SECONDARY ZONE (integers)'
       WRITE (*,*)
       READ (*,*,ERR=145) IPL,IPR,ISL,ISR
       ISEC=1
       IF (IPL.LT.0) IPL=IPL+360
       IF (IPR.LT.0) IPR=IPR+360
       IF (ISL.LT.0) ISL=ISL+360
       IF (ISR.LT.0) ISR=ISR+360
       IF (ISL.EQ.0.AND.ISR.EQ.0) ISEC=0
       PL=IPL
       PR=IPR
       SL=ISL
       SR=ISR
       IF (ISEC.EQ.0) GO TO 155
C
150    WRITE (*,*)
       WRITE (*,*) 'SECONDARY SECTOR ALLOCATION'
       IF (IPL.NE.ISL.AND.IPR.NE.ISR) THEN
       WRITE (*,*) '1-UNIQUE COMMAND UNIT ANGLES, SAME SUB-UNIT ALLOC.'
       ISTYPE=1
       ENDIF
       IF (IPL.EQ.ISL.AND.IPR.EQ.ISR) THEN
151    WRITE (*,*) 'ENTER '
       WRITE (*,*) '2-EVERYONE SHIFT ONE SUB-UNIT ZONE LEFT'
       WRITE (*,*) '3-EVERYONE SHIFT ONE SUB-UNIT ZONE RIGHT'
       WRITE (*,*)
       READ (*,*,ERR=151) ISTYPE
       IF (ISTYPE.EQ.1) ISTYPE=0
       ENDIF
       WRITE (*,*)
       IF (ISTYPE.LE.0.OR.ISTYPE.GE.4) GO TO 150
C
155    CONTINUE
C
C      ESTABLISH RANGE WINDOWS
C
200    WRITE (*,*)
       WRITE (*,*) 'RANGE WINDOWS'
       WRITE (*,*) '1 -- UNRESTRICTED'
       WRITE (*,*) '2 -- ESTABLISH MIN and MAX ENGAGEMENT RANGES'
       WRITE (*,*)
       READ (*,*,ERR=200) IRNG
       IF (IRNG.LT.1.OR.IRNG.GT.2) GO TO 200
       IF (IRNG.EQ.2) THEN
       WRITE (*,*)
       WRITE (*,*) 'ENTER MIN RANGE AND MAX RANGE meters'
       READ (*,*,ERR=200) XMN,XMX
       WRITE (*,*) XMN,XMX
       XMN=XMN/1000.
       XMX=XMX/1000.
       ELSE
       XMN=0.0
       XMX=1.0E8
       ENDIF
C
C      BLUE
       IF (KSIDE.EQ.1) THEN
       WRNGB(ICOMMAND,1)=XMN
       WRNGB(ICOMMAND,2)=XMX
       IBPRIME(ICOMMAND,1)=IPL
       IBPRIME(ICOMMAND,2)=IPR
       IF (IPL.EQ.0.AND.IPR.EQ.0) THEN
       MODEFB(ICOMMAND)=7
       ELSE
         IF (MODEFB(ICOMMAND).EQ.7.OR.MODEFB(ICOMMAND).EQ.1) THEN
C        DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
C        SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
         ALFT1=IPL
         ARGT1=IPR
         IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
         ALFT2=ISL
         ARGT2=ISR
         IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
         NSYS=IBSYS(ICOMMAND,1)
         STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHB(NSYS)
         TSHOOT=0.0
         IF (MODEFB(ICOMMAND).EQ.7) TSHOOT=STIME*RND()
         IF (MODEFB(ICOMMAND).EQ.1) TSHOOT=TRMFB(ICOMMAND)-ITIME
         TRMFB(ICOMMAND)=ITIME+STIME+TSHOOT
         MODEFB(ICOMMAND)=1
         ENDIF
       ENDIF
       IF (ISEC.EQ.0) THEN
       IBSECOND(ICOMMAND,1)=0
       IBSECOND(ICOMMAND,2)=0
       ELSE
       IBSECOND(ICOMMAND,1)=ISL
       IBSECOND(ICOMMAND,2)=ISR
       ENDIF
       IF (NSUB.NE.0) THEN
         IF (ISUB.EQ.1) THEN
C          ALLOCATE SEQUENTIALLY TO ALL SUBORDINATES
           PANGLE=PR-PL
           SANGLE=SR-SL
           MSUB=0
           DO 157 I=NSUB1,NSUB2
             IF (IBSYS(I,2).EQ.1.AND.IBSYS(I,1).LE.10) THEN
             WRNGB(I,1)=XMN
             WRNGB(I,2)=XMX
               MSUB=MSUB+1
               IF (IPL.EQ.0.AND.IPR.EQ.0) THEN
               MODEFB(I)=7
               ELSE
                 IF (MODEFB(I).EQ.7.OR.MODEFB(I).EQ.1) THEN
C        DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
C        SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
               ALFT1=IPL
               ARGT1=IPR
               IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
               ALFT2=ISL
```

```
      ARGT2=ISR
      IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
      NSYS=IBSYS(I,1)
      STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHB(NSYS)
      TSHOOT=0.0
      IF (MODEFB(I).EQ.7) TSHOOT=STIME*RND()
       IF (MODEFB(I).EQ.1) TSHOOT=TRMFB(I)-ITIME
       TRMFB(I)=ITIME+STIME+TSHOOT
            MODEFB(I)=1
            ENDIF
           ENDIF
      ENDIF
167    CONTINUE
      DPANGLE=PANGLE/MSUB
      DSANGLE=SANGLE/MSUB
C
      IC=0
      DO 170 I=NSUB1,NSUB2
                  IF (IBSYS(I,2).NE.1.OR.IBSYS(I,1).GT.10) GO TO 170
       IC=IC+1
       IBPRIME(I,1)=INT(PL+(IC-1)*DPANGLE)
       IBPRIME(I,2)=INT(PL+IC*DPANGLE)
       IF (ISEC.EQ.0) THEN
       IBSECOND(I,1)=0
       IBSECOND(I,2)=0
       ELSE
       IBSECOND(I,1)=INT(SL+(IC-1)*DSANGLE)
       IBSECOND(I,2)=INT(SL+IC*DSANGLE)
       ENDIF
170   CONTINUE
      IF (ISTYPE.EQ.2) THEN
      DO 171 I=NSUB1,NSUB2
                  IF (IBSYS(I,2).NE.1.OR.IBSYS(I,1).GT.10) GO TO 171
            IF (I.NE.NSUB1) THEN
       IBSECOND(I,1)=IBPRIME(ILAST,1)
       IBSECOND(I,2)=IBPRIME(ILAST,2)
            ELSE
       IBSECOND(I,1)=INT(SL-DSANGLE)
      IF (IBSECOND(I,1).LT.0) IBSECOND(I,1)=IBSECOND(I,1)+360
       IBSECOND(I,2)=ISL
            ENDIF
       ILAST=I
171    CONTINUE
      ENDIF
      IF (ISTYPE.EQ.3) THEN
      IDEL=NSUB2-NSUB1+1
      DO 172 J=1,IDEL
      I=NSUB2-(J-1)
            IF (IBSYS(I,2).NE.1.OR.IBSYS(I,1).GT.10) GO TO 172
            IF (I.NE.NSUB2) THEN
       IBSECOND(I,1)=IBPRIME(ILAST,1)
       IBSECOND(I,2)=IBPRIME(ILAST,2)
            ELSE
       IBSECOND(I,1)=ISR
       IBSECOND(I,2)=INT(SL+DSANGLE)
      IF (IRSECOND(I,2).GT.360) IRSECOND(I,2)=IRSECOND(I,2)-360
            ENDIF
       ILAST=I
172    CONTINUE
      ENDIF

      ENDIF
C
      IF (ISUB.EQ.2) THEN
C       EVERY ONE GETS THE SAME ZONES
       DO 175 I=NSUB1,NSUB2
       WRNGB(I,1)=XMN
       WRNGB(I,2)=XMX
       IBPRIME(I,1)=IBPRIME(ICOMMAND,1)
       IBPRIME(I,2)=IBPRIME(ICOMMAND,2)
                  IF (IBSYS(I,2).EQ.1.AND.IBSYS(I,1).LE.10) THEN
          IF (IPL.EQ.0.AND.IPR.EQ.0) THEN
          MODEFB(I)=7
          ELSE
            IF (MODEFB(I).EQ.7.OR.MODEFB(I).EQ.1) THEN
C     DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
C     SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
      ALFT1=IPL
      ARGT1=IPR
      IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
      ALFT2=ISL
      ARGT2=ISR
      IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
      NSYS=IBSYS(I,1)
      STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHB(NSYS)
      TSHOOT=0.0
       IF (MODEFB(I).EQ.7) TSHOOT=STIME*RND()
       IF (MODEFB(I).EQ.1) TSHOOT=TRMFB(I)-ITIME
       TRMFB(I)=ITIME+STIME+TSHOOT
            MODEFB(I)=1
            ENDIF
           ENDIF
          ENDIF
         ENDIF
       IF (ISEC.EQ.0) THEN
       IBSECOND(I,1)=0
       IBSECOND(I,2)=0
       ELSE
       IBSECOND(I,1)=IBSECOND(ICOMMAND,1)
       IBSECOND(I,2)=IBSECOND(ICOMMAND,2)
       ENDIF
175   CONTINUE
      ENDIF
      ENDIF
C
C     RED
      IF (KSIDE.EQ.2) THEN
      WRNGR(ICOMMAND,1)=XMN
      WRNGR(ICOMMAND,2)=XMX
      IRPRIME(ICOMMAND,1)=IPL
      IRPRIME(ICOMMAND,2)=IPR
            IF (IPL.EQ.0.AND.IPR.EQ.0) THEN
            MODEFR(ICOMMAND)=7
            ELSE
       IF (MODEFR(ICOMMAND).EQ.7.OR.MODEFR(ICOMMAND).EQ.1) THEN
C     DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
C     SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
      ALFT1=IPL
      ARGT1=IPR
      IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
      ALFT2=ISL
      ARGT2=ISR
      IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
      NSYS=IRSYS(ICOMMAND,1)
      STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHR(NSYS)
      TSHOOT=0.0
      IF (MODEFR(ICOMMAND).EQ.7) TSHOOT=STIME*RND()
       IF (MODEFR(ICOMMAND).EQ.1) TSHOOT=TRMFR(ICOMMAND)-ITIME
       TRMFR(ICOMMAND)=ITIME+STIME+TSHOOT
       MODEFR(ICOMMAND)=1
        ENDIF
            ENDIF
      IF (ISEC.EQ.0) THEN
      IRSECOND(ICOMMAND,1)=0
      IRSECOND(ICOMMAND,2)=0
      ELSE
      IRSECOND(ICOMMAND,1)=ISL
      IRSECOND(ICOMMAND,2)=ISR
      ENDIF
      IF (NSUB.NE.0) THEN
        IF (ISUB.EQ.1) THEN
C         ALLOCATE UNIFORMLY TO ALL SUBORDINATES
         PANGLE=PR-PL
         SANGLE=SR-SL
         MSUB=0
         DO 167 I=NSUB1,NSUB2
              IF (IRSYS(I,2).EQ.1.AND.IRSYS(I,1).LE.10) THEN
         WRNGR(I,1)=XMN
         WRNGR(I,2)=XMX
         MSUB=MSUB+1
           IF (IPL.EQ.0.AND.IPR.EQ.0) THEN
           MODEFR(I)=7
           ELSE
             IF (MODEFR(I).EQ.7.OR.MODEFR(I).EQ.1) THEN
C     DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
C     SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
      ALFT1=IPL
      ARGT1=IPR
      IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
      ALFT2=ISL
      ARGT2=ISR
      IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
      NSYS=IRSYS(I,1)
      STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHR(NSYS)
      TSHOOT=0.0
      IF (MODEFR(I).EQ.7) TSHOOT=STIME*RND()
       IF (MODEFR(I).EQ.1) TSHOOT=TRMFR(I)-ITIME
       TRMFR(I)=ITIME+STIME+TSHOOT
            MODEFR(I)=1
            ENDIF
           ENDIF
         ENDIF
167      CONTINUE
         DPANGLE=PANGLE/MSUB
         DSANGLE=SANGLE/MSUB
C
         IC=0
         DO 180 I=NSUB1,NSUB2
              IF (IRSYS(I,2).NE.1.OR.IRSYS(I,1).GT.10) GO TO 180
          IC=IC+1
          IRPRIME(I,1)=INT(PL+(IC-1)*DPANGLE)
          IRPRIME(I,2)=INT(PL+IC*DPANGLE)
          IF (ISEC.EQ.0) THEN
          IRSECOND(I,1)=0
          IRSECOND(I,2)=0
          ELSE
          IRSECOND(I,1)=INT(SL+(IC-1)*DSANGLE)
          IRSECOND(I,2)=INT(SL+IC*DSANGLE)
          ENDIF
180   CONTINUE
        IF (ISTYPE.EQ.2) THEN
        DO 181 I=NSUB1,NSUB2
             IF (IRSYS(I,2).NE.1.OR.IRSYS(I,1).GT.10) GO TO 181
                  IF (I.NE.NSUB1) THEN
           IRSECOND(I,1)=IRPRIME(ILAST,1)
           IRSECOND(I,2)=IRPRIME(ILAST,2)
                ELSE
           IRSECOND(I,1)=INT(SL-DSANGLE)
           IRSECOND(I,2)=ISL
                ENDIF
         ILAST=I
181      CONTINUE
        ENDIF
```

```
          IF (ISTYPE.EQ.3) THEN
          IDEL=NSUB2-NSUB1+1
          DO 182 J=1,IDEL
          I=NSUB2-(J-1)
                    IF (IRSYS(I,2).NE.1.OR.IRSYS(I,1).GT.10) GO TO 182
                  IF (I.NE.NSUB2) THEN
               IRSECOND(I,1)=IRPRIME(ILAST,1)
               IRSECOND(I,2)=IRPRIME(ILAST,2)
                    ELSE
               IRSECOND(I,1)=ISR
               IRSECOND(I,2)=INT(SL+DSANGLE)
                    ENDIF
          ILAST=I
182       CONTINUE
          ENDIF

        ENDIF
C
        IF (ISUB.EQ.2) THEN
C         EVERY ONE GETS THE SAME ZONES
          DO 186 I=NSUB1,NSUB2
          WRNGR(I,1)=XMN
          WRNGR(I,2)=XMX
          IRPRIME(I,1)=IRPRIME(ICOMMAND,1)
          IRPRIME(I,2)=IRPRIME(ICOMMAND,2)
                    IF (IRSYS(I,2).EQ.1.AND.IRSYS(I,1).LE.10) THEN
            IF (IPL.EQ.0.AND.IPR.EQ.0) THEN
            MODEFR(I)=7
            ELSE
              IF (MODEFR(I).EQ.7.OR.MODEFR(I).EQ.1) THEN
C     DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
C     SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
            ALFT1=IPL
            ARGT1=IPR
            IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.
            ALFT2=ISL
            ARGT2=ISR
            IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.
            NSYS=IRSYS(I,1)
            STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHR(NSYS)
            TSHOOT=0.0
            IF (MODEFR(I).EQ.7) TSHOOT=STIME*RND()
            IF (MODEFR(I).EQ.1) TSHOOT=TRMFR(I)-ITIME
            TRMFR(I)=ITIME+STIME+TSHOOT
                MODEFR(I)=1
              ENDIF
            ENDIF
          ENDIF
          IF (ISEC.EQ.0) THEN
          IRSECOND(I,1)=0
          IRSECOND(I,2)=0
          ELSE
          IRSECOND(I,1)=IRSECOND(ICOMMAND,1)
          IRSECOND(I,2)=IRSECOND(ICOMMAND,2)
          ENDIF
186     CONTINUE
        ENDIF

      ENDIF
      ENDIF
C
      GO TO 141

1000  CONTINUE
      RETURN
      END

      SUBROUTINE COMMAND
10     WRITE (*,*)
      WRITE (*,*) '1 -- MOVEMENT ORDERS'
              WRITE (*,*) '2 -- DIRECT-FIRE ORDERS'
              WRITE (*,*) '3 -- VEHICLE OBSCURANTS ORDERS'
              WRITE (*,*) '4 -- DIRECT-FIRE TARGET PRIORITIES'
              WRITE (*,*) '5 -- INDIRECT-FIRE ORDERS'
              WRITE (*,*) '6 -- QUIT'
      WRITE (*,*)
      READ (*,*,ERR=10) IWANT
      IF (IWANT.EQ.1) CALL MOVEORS
      IF (IWANT.EQ.2) CALL FIREORS
      IF (IWANT.EQ.3) CALL OBSCURANTS
              IF (IWANT.EQ.4) CALL PRIORITIES
              IF (IWANT.EQ.5) CALL IDFIREORS
              IF (IWANT.EQ.6) RETURN
      GO TO 10
      END


      SUBROUTINE FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG,RT,PKIT,NUMT)
C      INTERPOLATES PK TABLES TO GIVE PK AT DESIGNATED RANGE
C
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACQB(10,5),ACQR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)
      COMMON/DEFILADE/IDEFB(200),IDEFR(200)

      DIMENSION PK(6),R(6)

      CHARACTER*9 BFNAME,RFNAME
C       WRITE (*,*) '4'
C       WRITE (*,*) (RB(ISR),ISR=1,36)
C       WRITE (*,*) (RR(ISR),ISR=1,36)
C
C       KSIDE  1 - BLUE      2 - RED
C       NSYS   ATTACKING SYSTEM NUMBER
C       NITSYS  TARGET SYSTEM NUMBER
C       NUM    ATTACKING WEAPON NUMBER
C       NRNG   NUMBER OF RANGES IN RANGE TABLE
C       RT     TARGET RANGE
C       PKIT   WEAPON PK ON THIS TARGET AT THIS RANGE
C       NUMT   TARGET NUMBER
C
      PKIT=0.0

      IF (NSYS.GT.10) RETURN
      IF (NITSYS.GT.10) THEN
       PKIT=1.0
       RETURN
      ENDIF

       RANGE=RT*.999*1000.
       DO 10 NR=1,6
       PK(NR)=0.0
       R(NR)=0.0
10     CONTINUE
C
       DO 100 NR=1,NRNG

        IF (KSIDE.EQ.1) THEN
       N=(NSYS-1)*18+(NUM-1)*6+NR
       R(NR)=RB(N)
C       DETERMINE IF TARGET IS EXPOSED OR IN DEFILADE
       JDEF=IDEFR(NUMT)

        IF (JDEF.EQ.0) THEN
C         TARGET IS EXPOSED
         NIT=(NITSYS-1)*2+1
         PK(NR)=PKB(N,NIT)

        ELSE

C         TARGET IN DEFILADE
         NIT=(NITSYS-1)*2+2
         PK(NR)=PKB(N,NIT)
        ENDIF

        ELSE

       N=(NSYS-1)*18+(NUM-1)*6+NR
       R(NR)=RR(N)

C       DETERMINE IF TARGET IS EXPOSED OR IN DEFILADE
       JDEF=IDEFB(NUMT)
        IF (JDEF.EQ.0) THEN
C         TARGET IS EXPOSED
         NIT=(NITSYS-1)*2+1
         PK(NR)=PKR(N,NIT)
        ELSE
C         TARGET IN DEFILADE
         NIT=(NITSYS-1)*2+2
         PK(NR)=PKR(N,NIT)
        ENDIF

        ENDIF

C       WRITE (*,*) 'PK ',PK(NR),'R ',R(NR)
100    CONTINUE
C
       DO 200 I=1,(NRNG-1)
        IF (RANGE.GT.R(I).AND.RANGE.LT.R(I+1)) THEN
C         ITS BETWEEN THESE TWO
         BOT=R(I+1)-R(I)
         TOP=RANGE-R(I)
         DELPK=PK(I+1)-PK(I)
         FRACPK=DELPK*TOP/BOT
         PKIT=FRACPK+PK(I)
        ENDIF
200    CONTINUE
       IF (RANGE.GT.R(NRNG)) PKIT=0.0
       RETURN
       END


       SUBROUTINE TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
C
CCC     INTERPOLATES MIN AND MAX ENGAGEMENT TIMES TO GIVE
CC      TIME IN ENGAGEMENT MODE
C
      RNG=TRNG*1000.

       RMIN=0.0
       BOT=RMAX-RMIN
      TOP=RNG-RMIN
       DELT=TMAX-TMIN
       FRACT=DELT*TOP/BOT
       STIME=FRACT+TMIN
      RETURN
      END


       SUBROUTINE VMASK (KSIDE,IVEHICLE)
C
```

```
C     FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
C     SO THAT MASKING CAN BE EVALUATED
C     VEHICLE MASKING IS ONLY A POTENTIAL PROBLEM FOR VEHICLES
C     WITHIN 100 METERS OF THE SHOOTER, SINCE TERRAIN DEVIATIONS
C     WILL MAKE VEHICLE MASKING LARGELY
C     IRRELEVANT NEAR THE TARGET AND OVER THE INTERVENING DISTANCES.
C
      COMMON/MASKV/VMSK(400,2)
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT


C     KSIDE   - 1 - BLUE   2 - RED
C     IVEHICLE  -  VEHICLE WHICH IS SHOOTING
C     VMSK(I,1) -- RANGE TO EACH VEHICLE IN FORCE STRUCTURE
C     VMSK(I,2) -- ANGLE TO EACH VEHICLE IN FORCE STRUCTURE
C
      PI=3.14159
C     INITALIZE ARRAYS
      DO 10 I=1,400
      VMSK(I,1)=1.0
      VMSK(I,2)=0.0
10    CONTINUE
C     FIND SHOOTING VEHICLE'S LOCATION
      IF (KSIDE.EQ.1) THEN
      XS=BXYA(IVEHICLE,1)
      YS=BXYA(IVEHICLE,2)
      ELSE
      XS=RXYA(IVEHICLE,1)
      YS=RXYA(IVEHICLE,2)
      ENDIF
C     CYCLE THROUGH ALL BLUE AND RED VEHICLES TO FILL MASKING ARRAY
      IALL=NB+NR
      DO 20 I=1,IALL
      IF (I.GT.NB) THEN
        IWANT=I-NB
        XT=RXYA(IWANT,1)
        YT=RXYA(IWANT,2)
      ELSE
        IWANT=I
        XT=BXYA(IWANT,1)
        YT=BXYA(IWANT,2)
      ENDIF
      RANGE=((XT-XS)**2+(YT-YS)**2)**.5
      IF (RANGE.GT.0.1) GO TO 20
C     FIND ANGLE TO THIS TARGET
      TOP=YT-YS
      BOT=XT-XS
      IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
      IF (BOT.LT.0.0) THEN
          IF (TOP.LT.0.0) DIR=DIR+180.
          IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
          IF (TOP.LT.0.0) DIR=360.-DIR
      ENDIF
C     CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
      CALL FAZ (DIR,AZIN)
      VMSK(I,1)=RANGE
      VMSK(I,2)=AZIN
20    CONTINUE
      RETURN
      END


      SUBROUTINE VSEE (MYSIDE,K,J,RANGE,AZIN,JSKIP,NB,NR)
C
C     EVALUATES VEHICLE MASKING TO SEE IF TARGET CAN BE SEEN
C
      COMMON/MASKV/VMSK(400,2)
C
C     MYSIDE  SHOOTING VEHICLE SIDE (1 - BLUE, 2 - RED)
C     K       SHOOTING VEHICLE NUMBER
C     J       THE TARGET (IT IS OF COURSE IN THE FIELD OF VIEW)
C     RANGE   TARGET RANGE (DON'T LOOK FOR MASKING BEYOND THIS RANGE)
C     AZIN    AZIMUTH TO TARGET
C     JSKIP   FLAG TO SKIP THIS TARGET IF MASKED (=0 TO SKIP)
C     VMSK    VEHICLE RANGE AND AZIMUTH DATA FROM SHOOTING VEHICLE
C
      IF (MYSIDE.EQ.1) THEN
        ITSKIP=J
        MESKIP=K
      ELSE
        ITSKIP=J+NB
        MESKIP=K+NB
      ENDIF
      IALL=NB+NR
      PI=3.14159
C     AVERAGE WIDTH OF AN ARMORED VEHICLE
      WIDTH=4./1000.
      DO 10 I=1,IALL
      IF (JSKIP.EQ.0) GO TO 10
      IF (I.EQ.ITSKIP) GO TO 10
      IF (I.EQ.MESKIP) GO TO 10
      IF (VMSK(I,1).GT.0.1) GO TO 10
      AZLFT=VMSK(I,2)-180./PI*ATAN(WIDTH/2./VMSK(I,1))
      AZRGT=VMSK(I,2)+180./PI*ATAN(WIDTH/2./VMSK(I,1))
      IF (AZLFT.LT.0.0) AZLFT=AZLFT+360.
      IF (AZRGT.GT.360.0) AZRGT=AZRGT-360.
      IF (AZLFT.GT.AZRGT) AZRGT=AZRGT+360.
      IF (AZIN.GE.AZLFT.AND.AZIN.LE.AZRGT) JSKIP=0
10    CONTINUE
      RETURN
      END


      SUBROUTINE TERRMASK (XS,YS,XT,YT,JSKIP)
C     EVALUATES TERRAIN MASKING BETWEEN SHOOTER AND TARGET.
C     TO ENSURE EQUIVALENT TERRAIN MASKING CALCULATIONS BETWEEN
C     RED AND BLUE OVER THE SAME TERRAIN, MASKING IS ALWAYS
C     CALCULATED FROM BLUE TO RED, EVEN IF RED IS SHOOTER
C
C     XS,YS   SHOOTER REAL WORLD COORDINATES
C     XT,YT   TARGET  REAL WORLD COORDINATES
C     JSKIP   FLAG TO SKIP THIS TARGET IF MASKED (=0 TO SKIP)

      COMMON/ELEVA/CONT(20),IELEV(10000)
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT

      DOUBLE PRECISION ANGLE,ANGLEI

C     JSKIP=1


C     WRITE (10,*) 'ANGLE, ANGLEI'


C     FIND SHOOTER AND TARGET GRID COORDINATES
C     TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
C     CENTER TO CENTER ABOUT THE SCALE.
      IXS=NINT(XS/SCALE)
      IYS=NINT(YS/SCALE)
      IXT=NINT(XT/SCALE)
      IYT=NINT(YT/SCALE)
      IF (IXS.LT.1) IXS=1
      IF (IXT.LT.1) IXT=1
      IF (IYS.LT.1) IYS=1
      IF (IYT.LT.1) IYT=1
      IF (IXS.GT.NX) IXS=NX
      IF (IXT.GT.NX) IXT=NX
      IF (IYS.GT.NY) IYS=NY
      IF (IYT.GT.NY) IYT=NY
      X1=IXS*SCALE
      X2=IXT*SCALE
      Y1=IYS*SCALE
      Y2=IYT*SCALE
      RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5
C
C     FIND SHOOTER AND TARGET ELEVATION
      IS=NY*(IXS-1)+IYS
      SELEV=CONT(IELEV(IS))
      IT=NY*(IXT-1)+IYT
      TELEV=CONT(IELEV(IT))
C
C     ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
C     FOR VEHICLE HEIGHT
      SELEV=SELEV+.002
      TELEV=TELEV+.002
C
C     FIND ELEVATION ANGLE TO THIS TARGET
      TOP=TELEV-SELEV
      BOT=RANGE
      ANGLE=ATAN(TOP/BOT)
C
C     TO EVALUATE MASKING, FIND THE ELEVATION ANGLE AT SCALE
C     INCREMENTS ALONG THE LINE OF SIGHT.  IF ELEVATION ANGLE IS GREATER
C     THAN THIS ANGLE, SKIP THE TARGET, SINCE IT IS MASKED BY
C     TERRAIN
C
      DY=Y2-Y1
      DX=X2-X1
C
      IF (ABS(DY).GT.ABS(DX)) THEN
C     STEP THROUGH GRID ALONG Y AXIS
      SLOPE=(X2-X1)/(Y2-Y1)
      B=X2-SLOPE*Y2
      YI=Y1
20    CONTINUE
      IF (JSKIP.EQ.0) GO TO 200
      YI=YI+SCALE*DY/ABS(DY)
      XI=B+SLOPE*YI
      RI=((YI-Y1)**2+(XI-X1)**2)**.5
      IF (RI.GE.RANGE) GO TO 200
      IXI=NINT(XI/SCALE)
      IYI=NINT(YI/SCALE)
      ITI=NY*(IXI-1)+IYI
C       XIXI=IXI*SCALE
C       YIYI=IYI*SCALE
C     RI=((YIYI-Y1)**2+(XIXI-X1)**2)**.5
      SELEVI=CONT(IELEV(ITI))
C     FIND ELEVATION ANGLE TO THIS GRID SQUARE
      TOP=SELEVI-SELEV
      BOT=RI
      ANGLEI=ATAN(TOP/BOT)

C     WRITE (10,*) I,ANGLE,ANGLEI

    IF (ANGLEI.GT.ANGLE) JSKIP=0
      GO TO 20
```

```
200   CONTINUE
      ELSE
C     STEP THROUGH GRID ALONG X AXIS
      SLOPE=(Y2-Y1)/(X2-X1)
      B=Y2-SLOPE*X2
      XI=XS
10    CONTINUE
      IF (JSKIP.EQ.0) GO TO 100
      XI=XI+SCALE*DX/ABS(DX)
      YI=B+SLOPE*XI
      RI=((YI-YS)**2+(XI-XS)**2)**.5
      IF (RI.GE.RANGE) GO TO 100
      IXI=NINT(XI/SCALE)
      IYI=NINT(YI/SCALE)
      IT=NY*(IXI-1)+IY
C       XIXI=IXI*SCALE
C       YIYI=IYI*SCALE
C       RI=((YIYI-Y1)**2+(XIXI-X1)**2)**.5
      SELEVI=CONT(IELEV(IT))
C     FIND ELEVATION ANGLE TO THIS GRID SQUARE
      TOP=SELEVI-SELEV
      BOT=RI
            ANGLEI=ATAN(TOP/BOT)
C     WRITE (10,*) 2,ANGLE,ANGLEI
            IF (ANGLEI.GT.ANGLE) JSKIP=0
            GO TO 10
100   CONTINUE
            ENDIF
      RETURN
      END

      SUBROUTINE BZOOM
C
C     ZOOM WINDOW FOR BATTLE PHASE
C     ZOOMS IN ON ACTIVE GRID
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C
C     BLX    BOTTOM LEFT X COORDINATE
C     BLY    BOTTOM LEFT Y COOFDINATE
C     XLEN   ZOOM WINDOW SIDE LENGTH
C     ZFACTN  NEW ZOOM FACTOR TO INCREASE SCALE
C
      XUNIT=11.0/NX*SCFACT
      YUNIT=8.5/NY
      XMIN=1.0E6
      XMAX=0.0
      YMIN=1.0E6
      YMAX=0.0
            DO 10 I=1,NB
               IF (IBSYS(I,1).GT.10) GO TO 10
        IF (BXYA(I,1).LT.XMIN) XMIN=BXYA(I,1)
        IF (BXYA(I,1).GT.XMAX) XMAX=BXYA(I,1)
        IF (BXYA(I,2).LT.YMIN) YMIN=BXYA(I,2)
        IF (BXYA(I,2).GT.YMAX) YMAX=BXYA(I,2)
10    CONTINUE
            DO 11 I=1,NR
               IF (IRSYS(I,1).GT.10) GO TO 11
        IF (RXYA(I,1).LT.XMIN) XMIN=RXYA(I,1)
        IF (RXYA(I,1).GT.XMAX) XMAX=RXYA(I,1)
        IF (RXYA(I,2).LT.YMIN) YMIN=RXYA(I,2)
        IF (RXYA(I,2).GT.YMAX) YMAX=RXYA(I,2)
11    CONTINUE
      XMIN=XMIN-SCALE
      XMAX=XMAX+SCALE
      XAVG=XMIN+XMAX
      XAVG=XAVG/2.
      DX=XMAX-XMIN
      YMIN=YMIN-SCALE
      YMAX=YMAX+SCALE
      YAVG=YMIN+YMAX
      YAVG=YAVG/2.
      DY=YMAX-YMIN
C     WINDOW EDGE LENGTH IN GRID UNITS
        IF (DY.GT.DX) THEN
           XLEN=NINT(DY/SCALE)
           IF (XLEN.LT.4.1) XLEN=4.1
           XMIN=XAVG-DY/2.
           YMIN=YAVG-DY/2.
        ELSE
           XLEN=NINT(DX/SCALE)
           IF (XLEN.LT.4.1) XLEN=4.1
           XMIN=XAVG-DX/2.
           YMIN=YAVG-DX/2.
        ENDIF
C     GRID NX,NY OF WINDOW BOTTOM LEFT CORNER
      BLX=NINT(XMIN/SCALE)*XUNIT
      BLY=NINT(YMIN/SCALE)*YUNIT
C     CALCULATE NEW ZOOM FACTOR
      XLEN1=NX*XUNIT
      ZFACTN=XLEN1/XLEN/XUNIT*ZFACT
C       ZFACTN=ZFACT
      XMAX=XLEN*XUNIT
      YMAX=XLEN*YUNIT

      RETURN
      END

      SUBROUTINE EUNITS
C
C     ERASES UNIT LOCATIONS (VEHICLES) ON MAP
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/OLDMOVE/BXYOLD(200,2),RXYOLD(200,2)
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*1 BLANK
      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      DIMENSION XARRAY(4),YARRAY(4)
C
C     ISKIP=0
C
C     DRAWS AND FILLS A SQUARE FOR EACH UNIT LOCATION
C     SIDE LENGTH IS 5 METERS SINCE LARGEST UNIT IS A SQUAD VEHICLE
C      WHEN AT MAX ZOOM
C      AT MIN ZOOM, BOX IS DRAWN LARGER SO YOU CAN SEE IT
C
      CALL PLOTS (0,1,0)
      CALL FACTOR (ZFACT)
C       XUNIT=11.0/NX*SCFACT
C       YUNIT=8.5/NY
      GXUNIT=11.0/NX*SCFACT
      GYUNIT=8.5/NY
      XUNIT=GXUNIT/SCALE
      YUNIT=GYUNIT/SCALE
      XL=XUNIT*.005/SCALE*2.
      YL=YUNIT*.005/SCALE*2.
C       XL=XUNIT*.150/SCALE
C       YL=YUNIT*.150/SCALE
C       XL=XUNIT*.150/SCALE
C       YL=YUNIT*.150/SCALE
       IF (XLEN.GT.0.) THEN
            CALL FACTOR (ZFACTN)
C             XL=XUNIT*.150/SCALE*ZFACT/ZFACTN
C             YL=YUNIT*.150/SCALE*ZFACT/ZFACTN
       ENDIF
      DO 200 N=1,2
      DO 100 I=1,200
       ISKIP=0
       IF (N.EQ.1) THEN
            IHIER=BHIER(I)
            BLANK=IHIER(1:1)
       ELSE
            IHIER=RHIER(I)
            BLANK=IHIER(1:1)
       ENDIF
       IF (BLANK.EQ.'B') GO TO 50
       IF (BLANK.EQ.'R') GO TO 50
       GO TO 100
50     IF (N.EQ.1) THEN
            X=BXYOLD(I,1)
            Y=BXYOLD(I,2)
           CALL NEWPEN (0)
            IF (IBSYS(I,2).EQ.0) CALL NEWPEN(0)
       ELSE
            X=RXYOLD(I,1)
            Y=RXYOLD(I,2)
           CALL NEWPEN (0)
            IF (IRSYS(I,2).EQ.0) CALL NEWPEN(0)
       ENDIF
       X=X*XUNIT
       Y=Y*YUNIT
       IF (X.LE.0.0) ISKIP=1
       IF (Y.LE.0.0) ISKIP=1
       IF (XLEN.GT.0.) THEN
            CALL ZOOMIT (X,Y,ISKIP)
                 IF (Y.GT.(.95*YMAX)) ISKIP=1
                 IF (X.LT.(.05*XMAX)) ISKIP=1
       ENDIF
       IF (ISKIP.EQ.1) GO TO 100
C      DRAW AND FILL FOUR SIDED SQUARE
       XARRAY(1)=X+XL/2.
       XARRAY(2)=X+XL/2.
       XARRAY(3)=X-XL/2.
       XARRAY(4)=X-XL/2.
       YARRAY(1)=Y-YL/2.
       YARRAY(2)=Y+YL/2.
       YARRAY(3)=Y+YL/2.
       YARRAY(4)=Y-YL/2.
```

```
      CALL FILL (4,XARRAY,YARRAY)
100   CONTINUE
200   CONTINUE
C     READ (*,*)
      RETURN
      END


      SUBROUTINE SCONT (K)
C     CONTPLOT.FOR FOR BATTLE PHASE
C
C     PLOTS CONTOUR MAP AS GENERATED BY SOFTWARE
C     OPENS CONTSEGM.DAT FILE FOR CONTOUR SEGMENTS
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
     & NC(15),NCOL(15),JCON
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
C
C     NX          NUMBER OF X GRIDS
C     NY          NUMBER OF Y GRIDS (SHOULD EQUAL NX)
C     SCALE       REAL WORLD SCALE FACTOR FOR EACH GRID
C     SCFACT      SCREEN SCALING FACTOR FOR X DIRECTION
C     PRFACT      PRINTER SCALING FACTOR FOR X DIRECTION
C     ZFACT       ZOOM FACTOR FOR SIZE OF GRID ON SCREEN
C
      ISKIP=0
      CALL NEWPEN (7)
      CALL FACTOR (ZFACT)
      IF (XLEN.GT.0.) CALL FACTOR (ZFACTN)
C
      K=JCON
      J=K-1
      DO 200 I=1,J
      CALL NEWPEN (NCOL(IC(I)))
      X=X1(I)
      Y=Y1(I)
      IF (XLEN.GT.0.) CALL ZOOMIT (X,Y,ISKIP)
      IF (ISKIP.EQ.1) GO TO 200
      CALL PLOT (X,Y,3)
      X=X2(I)
      Y=Y2(I)
      IF (XLEN.GT.0.) CALL ZOOMIT (X,Y,ISKIP)
      IF (ISKIP.EQ.1) GO TO 200
      CALL PLOT (X,Y,2)
200   CONTINUE
C     READ (*,*)
C     CALL PLOT (0.,0.,999)

      RETURN

      END


      SUBROUTINE PLINE (X1,Y1,X2,Y2,KSIDE)
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      CALL FACTOR (ZFACT)
      IF (XLEN.GT.0.) CALL FACTOR (ZFACTN)
      GXUNIT=11.0/NX*SCFACT
      GYUNIT=8.5/NY
      XUNIT=GXUNIT/SCALE
      YUNIT=GYUNIT/SCALE
      XS=X1*XUNIT
      YS=Y1*YUNIT
      IF (KSIDE.EQ.1) CALL NEWPEN (9)
      IF (KSIDE.EQ.2) CALL NEWPEN (12)
      IF (XLEN.GT.0.) CALL ZOOMIT (XS,YS,ISKIP)
      CALL PLOT (XS,YS,3)
      XT=X2*XUNIT
      YT=Y2*YUNIT
      IF (XLEN.GT.0.) CALL ZOOMIT (XT,YT,ISKIP)
      CALL PLOT (XT,YT,2)
      RETURN
      END


      SUBROUTINE OLDPOS
C
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/OLDMOVE/BXYOLD(200,2),RXYOLD(200,2)
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*16 BHIER,RHIER,IHIER

      DO 10 I=1,NB
      BXYOLD(I,1)=BXYA(I,1)
      BXYOLD(I,2)=BXYA(I,2)
10    CONTINUE

      DO 11 I=1,NR
      RXYOLD(I,1)=RXYA(I,1)
      RXYOLD(I,2)=RXYA(I,2)
11    CONTINUE

      RETURN
      END


      SUBROUTINE PDUST
C     PLOTS DUST PUFFS
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/EDPUFFS/IDERASE,PTDRASE(2400,3)
C
C
      ISKIP=0
C
C     DRAWS A GREEN  CIRCLE FOR EACH PUFF LOCATION.
C     AND SAVES THIS PUFF FOR LATER ERASING.
C     RADIUS IS BASED ON PUFF TIME WHEN AT MAX ZOOM.
C     AT MIN ZOOM, CIRCLE IS DRAWN LARGER SO YOU CAN SEE IT.
C
      CALL FACTOR (ZFACT)
      GXUNIT=11.0/NX*SCFACT
      GYUNIT=8.5/NY
      XUNIT=GXUNIT/SCALE
      YUNIT=GYUNIT/SCALE
      IDERASE=0

      DO 100 I=IDUST1,IDUST2
      TIME=PTDUST(I,3)
      RADIUS=TIME*EXPUFF*XUNIT*1.0
      X=PTDUST(I,1)*XUNIT
      Y=PTDUST(I,2)*YUNIT
C     WRITE (*,*) 'RADIUS  X  Y'
C     WRITE (*,*) RADIUS,X,Y
      IF (XLEN.GT.0.) THEN
          CALL FACTOR (ZFACTN)
C         RADIUS=RADIUS*ZFACT/ZFACTN
      ENDIF
          CALL NEWPEN (10)
      IF ((X-15.*RADIUS).LE.0.0) ISKIP=1
      IF ((Y-15.*RADIUS).LE.0.0) ISKIP=1
      IF (XLEN.GT.0.) THEN
          CALL ZOOMIT (X,Y,ISKIP)
          IF ((Y+RADIUS).GT.(.95*YMAX)) ISKIP=1
          IF ((X-2.*RADIUS).LE.0.0) ISKIP=1
          IF ((Y-2.*RADIUS).LE.0.0) ISKIP=1
      ENDIF
      IF (ISKIP.EQ.1) GO TO 100
C     DRAW CIRCLE
      CALL CIRCLE (X,Y,RADIUS)
        IDERASE=IDERASE+1
        PTDRASE(IDERASE,1)=PTDUST(I,1)
        PTDRASE(IDERASE,2)=PTDUST(I,2)
        PTDRASE(IDERASE,3)=PTDUST(I,3)
100   CONTINUE
      RETURN
      END


      SUBROUTINE PENG
C     PLOTS ENGINE SMOKE PUFFS
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/EENPUFFS/IEERASE,PTERASE(2400,3)
C
C
      ISKIP=0
C
C     DRAWS A GREEN  CIRCLE FOR EACH PUFF LOCATION.
C     AND SAVES THIS PUFF FOR LATER ERASING.
C     RADIUS IS BASED ON PUFF TIME WHEN AT MAX ZOOM.
C     AT MIN ZOOM, CIRCLE IS DRAWN LARGER SO YOU CAN SEE IT.
C
      CALL FACTOR (ZFACT)
      GXUNIT=11.0/NX*SCFACT
      GYUNIT=8.5/NY
      XUNIT=GXUNIT/SCALE
      YUNIT=GYUNIT/SCALE
      IEERASE=0

      DO 100 I=IENG1,IENG2
      TIME=PTENG(I,3)
      RADIUS=TIME*EXPUFF*XUNIT*1.0
      X=PTENG(I,1)*XUNIT
      Y=PTENG(I,2)*YUNIT
      IF (XLEN.GT.0.) THEN
          CALL FACTOR (ZFACTN)
C         RADIUS=RADIUS*ZFACT/ZFACTN
      ENDIF
          CALL NEWPEN (10)
      IF ((X-15.*RADIUS).LE.0.0) ISKIP=1
      IF ((Y-15.*RADIUS).LE.0.0) ISKIP=1
      IF (XLEN.GT.0.) THEN
          CALL ZOOMIT (X,Y,ISKIP)
          IF ((Y+RADIUS).GT.(.95*YMAX)) ISKIP=1
          IF ((X-2.*RADIUS).LE.0.0) ISKIP=1
          IF ((Y-2.*RADIUS).LE.0.0) ISKIP=1
```

```
      ENDIF
      IF (ISKIP.EQ.1) GO TO 100
C     DRAW CIRCLE
      CALL CIRCLE (X,Y,RADIUS)
        IERASE=IERASE+1
        PTERASE(IERASE,1)=PTENG(I,1)
        PTERASE(IERASE,2)=PTENG(I,2)
        PTERASE(IERASE,3)=PTENG(I,3)
100   CONTINUE
      RETURN
      END


      SUBROUTINE MPUFFS
C     MOVES OBSCURANT PUFFS
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
C
      PI=3.14159
      DT=DTPUFF
      V=IWSPEED*1.853
      AZIN=IWDIR
      CALL FDIR (AZIN,AZOUT)

      IF (NDUST.GT.0) THEN
C     MOVE DUST
      DO 100 I=IDUST1,IDUST2
C       MOVE IT
      PTDUST(I,1)=PTDUST(I,1)+V*DT*COS(AZOUT*PI/180.)/3600.
      PTDUST(I,2)=PTDUST(I,2)+V*DT*SIN(AZOUT*PI/180.)/3600.
       PTDUST(I,3)=PTDUST(I,3)+DTPUFF
100   CONTINUE
      ENDIF

      IF (NENG.GT.0) THEN
C     MOVE ENGINE SMOKE
      DO 200 I=IENG1,IENG2
C       MOVE IT
      PTENG(I,1)=PTENG(I,1)+V*DT*COS(AZOUT*PI/180.)/3600.
      PTENG(I,2)=PTENG(I,2)+V*DT*SIN(AZOUT*PI/180.)/3600.
       PTENG(I,3)=PTENG(I,3)+DTPUFF
200   CONTINUE
      ENDIF

      IF (NSMK.NE.0) THEN
C     MOVE SMOKE GRENADE CURTAIN
      DO 300 I=ISMK1,ISMK2
C       MOVE IT
      PSMK(I,1)=PSMK(I,1)+V*DT*COS(AZOUT*PI/180.)/3600.
      PSMK(I,2)=PSMK(I,2)+V*DT*SIN(AZOUT*PI/180.)/3600.
       TASMK(I,1)=TASMK(I,1)+DTPUFF
300   CONTINUE
      ENDIF

      IF (NART.NE.0) THEN
C     MOVE ARTILLERY SMOKE CURTAIN
      DO 400 I=IART1,IART2
C       MOVE IT
      PART(I,1)=PART(I,1)+V*DT*COS(AZOUT*PI/180.)/3600.
      PART(I,2)=PART(I,2)+V*DT*SIN(AZOUT*PI/180.)/3600.
           TAART(I,1)=TAART(I,1)-DTPUFF
400   CONTINUE
      ENDIF

      RETURN
      END


      SUBROUTINE DELPUFFS
C     DELETE DISSIPATED OBSCURANTS
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
C
      IF (NDUST.GT.0) THEN
C     DELETE DISSIPATED DUST
      DO 100 I=IDUST1,IDUST2
      J=I
        IF (PTDUST(J,3).GE.TENDPUFF) THEN
C         THIS PUFF HAS EXPIRED
          ISTART=J+1
          PTDUST(J,1)=-1
          PTDUST(J,2)=-1
          PTDUST(J,3)=1.E6
          NDUST=NDUST-1
        ELSE
C         THIS IS THE FIRST PUFF TO CARRY FORWARD
          ISTART=J
          GO TO 101
        ENDIF
100   CONTINUE
101   CONTINUE
      IDUST1=ISTART
        IF (NDUST.LE.0.OR.IDUST1.GE.IDUST2) THEN
        IDUST1=0
        IDUST2=0
        NDUST=0
        ELSE
      NDUST=IDUST2-IDUST1+1
        ENDIF
      ENDIF

      IF (NENG.GT.0) THEN
C     DELETE DISSIPATED ENGINE SMOKE
      DO 200 I=IENG1,IENG2
      J=I
        IF (PTENG(J,3).GE.TENDPUFF) THEN
C         THIS PUFF HAS EXPIRED
          ISTART=J+1
          PTENG(J,1)=-1
          PTENG(J,2)=-1
          PTENG(J,3)=1.E6
          NENG=NENG-1
        ELSE
C         THIS IS THE FIRST PUFF TO CARRY FORWARD
          ISTART=J
          GO TO 201
        ENDIF
200   CONTINUE
201   CONTINUE
      IENG1=ISTART
        IF (NENG.LE.0.OR.IENG1.GE.IENG2) THEN
        IENG1=0
        IENG2=0
        NENG=0
        ELSE
      NENG=IENG2-IENG1+1
        ENDIF
      ENDIF

      IF (NSMK.NE.0) THEN
C     DELETE DISSIPATED SMOKE GRENADE CURTAIN
      DO 300 I=ISMK1,ISMK2
      J=I
        IF (TASMK(J,1).GE.TENDPUFF) THEN
C         THIS PUFF HAS EXPIRED
          ISTART=J+1
        ELSE
C         THIS IS THE FIRST PUFF TO CARRY FORWARD
          ISTART=J
          GO TO 301
        ENDIF
300   CONTINUE
301   CONTINUE
      ISMK1=ISTART
        IF (ISMK1.GE.ISMK2) THEN
        ISMK1=0
        ISMK2=0
        NSMK=0
        ELSE
      NSMK=ISMK2-ISMK1+1
        ENDIF
      ENDIF

      IF (NART.NE.0) THEN
C     DELETE DISSIPATED ARTILLERY SMOKE PUFF
      DO 400 I=IART1,IART2
      J=I
      IF (TAART(J,1).LE.0.0) THEN
C         THIS PUFF HAS EXPIRED
          ISTART=J+1
        ELSE
C         THIS IS THE FIRST PUFF TO CARRY FORWARD
          ISTART=J
          GO TO 401
        ENDIF
400   CONTINUE
401   CONTINUE
      IART1=ISTART
        IF (IART1.GE.IART2) THEN
        IART1=0
        IART2=0
        NART=0
        ELSE
      NART=IART2-IART1+1
        ENDIF
      ENDIF

      RETURN
      END


      SUBROUTINE NEWPUFFS
C     CREATES NEW OBSCURANT PUFFS AS REQUIRED (DUST AND ENGINE)
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
```

```
      COMMON/OBSCURE/NGRENB(200,2),NGRENR(200,2),IBESMK(200),
     & IRESMK(200)
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C
      IF (IDUST.EQ.1) THEN
C     PLAYING DUST, CREATE NEW DUST PUFFS
        IF (NDUST.GT.0) THEN
C       TEST TO SEE IF DUST ARRAY NEEDS TO BE COMPRESSED BACK
C       TO COLUMN 1
        ITEST=NR+NB+IDUST2
          IF (ITEST.GT.NDUSTMX) THEN
C           COMPRESS ARRAY BEFORE STARTING
            J=0
            DO 100 I=IDUST1,IDUST2
            J=J+1
            PTDUST(J,1)=PTDUST(I,1)
            PTDUST(J,2)=PTDUST(I,2)
            PTDUST(J,3)=PTDUST(I,3)
100           CONTINUE
            IDUST1=1
            IDUST2=J
          ENDIF
        ENDIF
C     CREATE NEW PUFFS AFTER IDUST2
        J=0
        DO 200 I=1,NB
        IF (IBSYS(I,2).EQ.0) GO TO 200
C       SKIP KILLED VEHICLES
        IF (IBSA(I,1).LT.6) GO TO 200
C       SKIP VEHICLES MOVING UNDER 6 KM/HR
        J=J+1
        X=BXYA(I,1)
        Y=BXYA(I,2)
        K=IDUST2+J
        PTDUST(K,1)=X
        PTDUST(K,2)=Y
        PTDUST(K,3)=DTPUFF
C       WRITE (*,*) 'PTDUST 1  2  3'
C       WRITE (*,*) (PTDUST(K,L),L=1,3)
200     CONTINUE

        DO 300 I=1,NR
        IF (IRSYS(I,2).EQ.0) GO TO 300
C       SKIP KILLED VEHICLES
        IF (IRSA(I,1).LT.6) GO TO 300
C       SKIP VEHICLES MOVING UNDER 6 KM/HR
        J=J+1
        X=RXYA(I,1)
        Y=RXYA(I,2)
        K=IDUST2+J
        PTDUST(K,1)=X
        PTDUST(K,2)=Y
        PTDUST(K,3)=DTPUFF
300     CONTINUE

        IDUST2=IDUST2+J
              NDUST=IDUST2-IDUST1+1

      ENDIF
C
C     ENGINE SMOKE
        IF (NENG.GT.0) THEN
C       TEST TO SEE IF ENGINE SMOKE ARRAY NEEDS TO BE COMPRESSED BACK
C       TO COLUMN 1
        ITEST=NR+NB+IENG2
          IF (ITEST.GT.NENGMX) THEN
C           COMPRESS ARRAY BEFORE STARTING
            J=0
            DO 110 I=IENG1,IENG2
            J=J+1
            PTENG(J,1)=PTENG(I,1)
            PTENG(J,2)=PTENG(I,2)
            PTENG(J,3)=PTENG(I,3)
110           CONTINUE
            IENG1=1
            IENG2=J
          ENDIF
        ENDIF
C     CREATE NEW ENGINE SMOKE PUFFS AFTER IENG2
        J=0
        DO 210 I=1,NB
        IF (IBSYS(I,2).EQ.0) GO TO 210
C       SKIP KILLED VEHICLES
        IF (IBESMK(I).EQ.0) GO TO 210
C       SKIP VEHICLES WITH ENGINE SMOKE TURNED OFF
        J=J+1
        X=BXYA(I,1)
        Y=BXYA(I,2)
        K=IENG2+J
        PTENG(K,1)=X
        PTENG(K,2)=Y
        PTENG(K,3)=DTPUFF
C       WRITE (*,*) 'PTDUST 1  2  3'
C       WRITE (*,*) (PTDUST(K,L),L=1,3)
210     CONTINUE

        DO 310 I=1,NR
        IF (IRSYS(I,2).EQ.0) GO TO 310
C       SKIP KILLED VEHICLES
        IF (IRESMK(I).EQ.0) GO TO 310
C       SKIP VEHICLES WITH ENGINE SMOKE TURNED OFF
        J=J+1
        X=RXYA(I,1)
        Y=RXYA(I,2)
        K=IENG2+J
        PTENG(K,1)=X
        PTENG(K,2)=Y
        PTENG(K,3)=DTPUFF
310     CONTINUE

        IENG2=IENG2+J
      NENG=IENG2-IENG1+1

C
        RETURN
        END


        SUBROUTINE ERASEPUFFS
C       ERASES ALL PREVIOUSLY DRAWN DUST AND ENGINE SMOKE PUFFS
        COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
        COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
        COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/EDPUFFS/IDERASE,PTDRASE(2400,3)
        COMMON/EENPUFFS/IEERASE,PTERASE(2400,3)
C
C
C         ISKIP=0
C
C       DRAWS A BLACK CIRCLE FOR EACH PUFF LOCATION.
C       RADIUS IS BASED ON PUFF TIME WHEN AT MAX ZOOM.
C       AT MIN ZOOM, CIRCLE IS DRAWN LARGER SO YOU CAN SEE IT.
C
        CALL FACTOR (ZFACT)
        GXUNIT=11.0/NX*SCFACT
        GYUNIT=8.5/NY
        XUNIT=GXUNIT/SCALE
        YUNIT=GYUNIT/SCALE

        DO 200 J=1,2
          IF (J.EQ.1) IERASE=IDERASE
          IF (J.EQ.2) IERASE=IEERASE
        DO 100 I=1,IERASE
          IF (J.EQ.1) THEN
          TIME=PTDRASE(I,3)
          X=PTDRASE(I,1)*XUNIT
          Y=PTDRASE(I,2)*YUNIT
          ELSE
          TIME=PTERASE(I,3)
          X=PTERASE(I,1)*XUNIT
          Y=PTERASE(I,2)*YUNIT
          ENDIF
        RADIUS=TIME*EXPUFF*XUNIT*1.0
C         WRITE (*,*) 'RADIUS  X  Y'
C         WRITE (*,*) RADIUS,X,Y
          IF (XLEN.GT.0.) THEN
                CALL FACTOR (ZFACTN)
C               RADIUS=RADIUS*ZFACT/ZFACTN
          ENDIF
              CALL NEWPEN (0)
C         IF ((X-RADIUS).LE.0.0) ISKIP=1
C         IF ((Y-RADIUS).LE.0.0) ISKIP=1
          IF (XLEN.GT.0.) THEN
                CALL ZOOMIT (X,Y,ISKIP)
C               IF ((Y+RADIUS).GT.(.95*YMAX)) ISKIP=1
          ENDIF
C         IF (ISKIP.EQ.1) GO TO 100
C         DRAW CIRCLE
        CALL CIRCLE (X,Y,RADIUS)
100     CONTINUE
200     CONTINUE
        RETURN
        END


        SUBROUTINE OBSCURANTS
C
C       ENTER ORDERS FOR UNITS TO EMPLOY ENGINE SMOKE, SMOKE GRENADES
C       OR ARTILLERY SMOKE (NOT DONE YET)
C
        COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
C       COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
C     & NC(15),NCOL(15),JCON
        COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
        COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
        COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
        COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDS(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)
        COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
        COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
        COMMON/FIRE/IBPRIME(200,2),IBSECOND(200,2),IRPRIME(200,2),
     & IRSECOND(200,2),WRNGB(200,2),WRNGR(200,2)
```

```
      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
      COMMON/OBSCURE/NGRENB(200,2),NGRENR(200,2),IBESMK(200),
     & IRESMK(200)
C     COMMON/TYPES/NBAT,NCOMP,ICOMP(10),NPLAT(10),IPLAT(10,10),
C     & NSQD(10,10),JPLAT,JSQD
C
      CHARACTER*8 BFNAME,RFNAME
C     CHARACTER*80 LINE
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*1 C2
C     CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
      KSIDE=0
10    WRITE (*,*)
      WRITE (*,*) 'OBSCURANT ORDERS'
      WRITE (*,*) 'enter BLUE or RED, or QUIT'
      WRITE (*,*)
      READ (*,*,ERR=10) C2
      IF (C2(1:1).EQ.'B') KSIDE=1
      IF (C2(1:1).EQ.'R') KSIDE=2
      IF (C2(1:1).EQ.'Q') GO TO 1000
      IF (KSIDE.EQ.0) GO TO 10
C
5     WRITE (*,*)
      WRITE (*,*) 'CURRENT STATUS'
11    FORMAT (1X,A80)
C     IF (C2(1:1).EQ.'B') WRITE (*,11) L1BM
C     IF (C2(1:1).EQ.'R') WRITE (*,11) L1RM
      WRITE (*,*)' UNIT HIERARCHY     IDENTITY              ENGINE
     &SMOKE    SMOKE GRENADES'
      WRITE (*,*) '                                    1 = ON
     &0 = OFF  # SHOTS  FIRE=1'
      IF (C2(1:1).EQ.'B') THEN
C     BLUE
      KK=0
      DO 130 K=1,NB
121     FORMAT (1X,I3,1X,A16,A24,3(I6,4X))
            IF (IBSYS(K,2).EQ.1.AND.IBSYS(K,1).LE.10) THEN
      WRITE (*,121) K,BHIER(K),BUNIT(K),IBESMK(K),NGRENB(K,1),
     & NGRENB(K,2)
      KK=KK+1
      ENDIF
      IF (KK.EQ.20) THEN
            READ (*,*)
            KK=0
      ENDIF
130   CONTINUE
C
      ELSE
C     RED
      KK=0
      DO 140 K=1,NR
            IF (IRSYS(K,2).EQ.1.AND.IRSYS(K,1).LE.10) THEN
      WRITE (*,121) K,RHIER(K),RUNIT(K),IRESMK(K),NGRENR(K,1),
     & NGRENR(K,2)
      KK=KK+1
      ENDIF
      IF (KK.EQ.20) THEN
            READ (*,*)
            KK=0
      ENDIF
140   CONTINUE
      ENDIF

141   WRITE (*,*) 'ENTER COMMAND UNIT (0-RELISTS -1-QUIT)'
      WRITE (*,*)
      READ (*,*,ERR=141) ICOMMAND
      IF (ICOMMAND.EQ.0) GO TO 5
      IF (ICOMMAND.EQ.-1) GO TO 10
C
142   WRITE (*,*) 'ENTER NUMBER RANGE OF SUBORDINATES FOR'
      WRITE (*,*) 'SUB-UNIT ALLOCATION (#START #END) (0 0 FOR NONE)'
      WRITE (*,*)
      READ (*,*,ERR=142) NSUB1,NSUB2
      NSUB=1
      IF (NSUB1.LE.0) THEN
            NSUB1=0
            NSUB2=0
            NSUB=0
      ENDIF
      IF (NSUB2.LE.0) THEN
            NSUB1=0
            NSUB2=0
            NSUB=0
      ENDIF
      IF (NSUB.EQ.0) GO TO 145
C
C143  WRITE (*,*) 'SUB-UNIT ALLOCATIONS'
      ISUB=2
C
145   WRITE (*,*)
      WRITE (*,*) 'ENTER 1 TO TURN ENGINE SMOKE ON; 0 -- OFF'
      WRITE (*,*)
      READ (*,*,ERR=145) IENGSMK
      IF (IENGSMK.LT.0.OR.IENGSMK.GT.1) GO TO 145
146   WRITE (*,*)
      WRITE (*,*) 'ENTER 1 TO FIRE SMOKE GRENADES; 0 -- NO FIRE'
      WRITE (*,*)
      READ (*,*,ERR=146) IGSMK
      IF (IGSMK.LT.0.OR.IGSMK.GT.1) GO TO 146
C
C     BLUE
      IF (KSIDE.EQ.1) THEN
      IF (IBSYS(ICOMMAND,2).EQ.1.AND.IBSYS(ICOMMAND,1).LE.10) THEN
        IBESMK(ICOMMAND)=IENGSMK
        IF (NGRENB(ICOMMAND,1).GT.0) NGRENB(ICOMMAND,2)=IGSMK
        ENDIF
          IF (ISUB.EQ.2) THEN
C           EVERY ONE GETS THE SAME ORDERS
            DO 175 I=NSUB1,NSUB2
                IF (IBSYS(I,2).EQ.1.AND.IBSYS(I,1).LE.10) THEN
              IBESMK(I)=IENGSMK
              IF (NGRENB(I,1).GT.0) NGRENB(I,2)=IGSMK
              ENDIF
175   CONTINUE
          ENDIF
      ENDIF
C
C     RED
      IF (KSIDE.EQ.2) THEN
      IF (IRSYS(ICOMMAND,2).EQ.1.AND.IRSYS(ICOMMAND,1).LE.10) THEN
        IRESMK(ICOMMAND)=IENGSMK
        IF (NGRENR(ICOMMAND,1).GT.0) NGRENR(ICOMMAND,2)=IGSMK
        ENDIF
          IF (ISUB.EQ.2) THEN
C           EVERY ONE GETS THE SAME ORDERS
            DO 185 I=NSUB1,NSUB2
                IF (IRSYS(I,2).EQ.1.AND.IRSYS(I,1).LE.10) THEN
              IRESMK(I)=IENGSMK
              IF (NGRENR(I,1).GT.0) NGRENR(I,2)=IGSMK
              ENDIF
185   CONTINUE
          ENDIF
      ENDIF
C
      GO TO 141

1000  CONTINUE
      RETURN
      END


      SUBROUTINE GRENADES
C
C     FIRES SMOKE GRENADES AND PUFFS CURTAINS
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
      COMMON/OBSCURE/NGRENB(200,2),NGRENR(200,2),IBESMK(200),
     & IRESMK(200)
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
      COMMON/FIRE/IBPRIME(200,2),IBSECOND(200,2),IRPRIME(200,2),
     & IRSECOND(200,2),WRNGB(200,2),WRNGR(200,2)

      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

C     SMOKE GRENADES
      IF (NSMK.GT.0) THEN
C     TEST TO SEE IF GRENADE SMOKE ARRAY NEEDS TO BE COMPRESSED BACK
C     TO COLUMN 1
      ITEST=NR+NB+ISMK2
        IF (ITEST.GT.NSMKMX) THEN
C       COMPRESS ARRAY BEFORE STARTING
        J=0
        DO 110 I=ISMK1,ISMK2
        J=J+1
        PSMK(J,1)=PSMK(I,1)
        PSMK(J,2)=PSMK(I,2)
        TASMK(J,1)=TASMK(I,1)
        TASMK(J,2)=TASMK(I,2)
110     CONTINUE
        ISMK1=1
        ISMK2=J
        ENDIF
      ENDIF
C     CREATE NEW SMOKE GRENADE CURTAINS AFTER ISMK2
      J=0
      PI=3.14159
      DO 210 I=1,NB
      IF (IBSYS(I,2).EQ.0) GO TO 210
C     SKIP KILLED VEHICLES
      IF (NGRENB(I,2).EQ.0) GO TO 210
C     SKIP VEHICLES WITH GRENADE SMOKE TURNED OFF
      IF (NGRENB(I,1).EQ.0) GO TO 210
C     SKIP VEHICLES THAT ARE OUT OF GRENADES
```

```
      J=J+1
      X=BXYA(I,1)
      Y=BXYA(I,2)
      K=ISMK2+J
C     FIRE GRENADE AND REDUCE NUMBER BY 1
            NGRENB(I,1)=NGRENB(I,1)-1
            WRITE (10,*) ' BLUE ',I,' FIRES SMOKE GRENADE VOLLEY '
C     ORIENTATION OF SMOKE CURTAIN IS PERPENDICULAR TO THE CENTER
C     OF THE PRIMARY SECTOR OF FIRE
      A1=IBPRIME(I,1)
      A2=IBPRIME(I,2)
      IF (A1.GT.A2) A2=A2+360.0
      AVG=A1+A2
      AVG=AVG/2.0
      APERP=AVG+90.
      IF (APERP.GE.360.0) APERP=APERP-360.0
      TASMK(K,2)=APERP
C     CURTAIN IS THROWN A DISTANCE OF 20 METERS IN FRONT OF THE
C     VEHICLE
      CALL FDIR (AVG,AZIN)
      PSMK(K,1)=X+COS(AZIN/180.*PI)*.02
      PSMK(K,2)=Y+SIN(AZIN/180.*PI)*.02
      TASMK(K,1)=DTPUFF
210   CONTINUE

      DO 310 I=1,NR
      IF (IRSYS(I,2).EQ.0) GO TO 310
C     SKIP KILLED VEHICLES
      IF (NGRENR(I,2).EQ.0) GO TO 310
C     SKIP VEHICLES WITH GRENADE SMOKE TURNED OFF
      IF (NGRENR(I,1).EQ.0) GO TO 310
C     SKIP VEHICLES THAT ARE OUT OF GRENADES
      J=J+1
      X=RXYA(I,1)
      Y=RXYA(I,2)
      K=ISMK2+J
C     FIRE GRENADE AND REDUCE NUMBER BY 1
            NGRENR(I,1)=NGRENR(I,1)-1
            WRITE (10,*) ' RED ',I,' FIRES SMOKE GRENADE VOLLEY'
C     ORIENTATION OF SMOKE CURTAIN IS PERPENDICULAR TO THE CENTER
C     OF THE PRIMARY SECTOR OF FIRE
      A1=IRPRIME(I,1)
      A2=IRPRIME(I,2)
      IF (A1.GT.A2) A2=A2+360.0
      AVG=A1+A2
      AVG=AVG/2.0
      APERP=AVG+90.
      IF (APERP.GE.360.0) APERP=APERP-360.0
      TASMK(K,2)=APERP
C     CURTAIN IS THROWN A DISTANCE OF 20 METERS IN FRONT OF THE
C     VEHICLE
      CALL FDIR (AVG,AZIN)
      PSMK(K,1)=X+COS(AZIN/180.*PI)*.02
      PSMK(K,2)=Y+SIN(AZIN/180.*PI)*.02
      TASMK(K,2)=APERP
310   CONTINUE

      ISMK2=ISMK2+J
            NSMK=ISMK2-ISMK1+1

C
      RETURN
      END


      SUBROUTINE PSMKCURTAINS
C     PLOTS SMOKE GRENADE CURTAINS
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/EDSMKG/ISGERASE,PSGERASE(400,2),ASGERASE(400)
C
      ISKIP=0
C
C     DRAWS A GREEN  LINE FOR EACH SMOKE CURTAIN LOCATION.
C     AND SAVES THIS LINE FOR LATER ERASING.
C
      CALL FACTOR (ZFACT)
      GXUNIT=11.0/NX*SCFACT
      GYUNIT=8.5/NY
      XUNIT=GXUNIT/SCALE
      YUNIT=GYUNIT/SCALE
      ISGERASE=0
C     CURTAIN LENGTH IS 150 FEET, BUT PLOTTED AT TWICE SCALE (LIKE
C     WITH VEHICLES SO THEY CAN BE SEEN BETTER)
      WCURT=150.*.3048/1000.*2.0
      PI=3.14159
            CALL NEWPEN (10)

      DO 100 I=ISMK1,ISMK2
C     CENTER POINT OF CURTAIN
      XC=PSMK(I,1)
      YC=PSMK(I,2)
C     CONVERT ORIENTATION AZIMUTH TO AN ANGLE
      CALL FDIR (TASMK(I,2),AZIN)
C     FIND STARTING POINT OF CURTAIN
      X1=XC+COS(AZIN/180.*PI)*WCURT/2.
      Y1=YC+SIN(AZIN/180.*PI)*WCURT/2.
C     ENDIG POINT
      X2=XC-COS(AZIN/180.*PI)*WCURT/2.
      Y2=YC-SIN(AZIN/180.*PI)*WCURT/2.
C     DRAW THIS LINE
      IF (XLEN.GT.0.) CALL FACTOR (ZFACTN)
      XS=X1*XUNIT
      YS=Y1*YUNIT
      ISKIP=0
      IF (XLEN.GT.0.) CALL ZOOMIT (XS,YS,ISKIP)
      IF (ISKIP.EQ.1) GO TO 100
      CALL PLOT (XS,YS,3)
      XT=X2*XUNIT
      YT=Y2*YUNIT
      IF (XLEN.GT.0.) CALL ZOOMIT (XT,YT,ISKIP)
      IF (ISKIP.EQ.1) GO TO 100
C     DRAW LINE
      CALL PLOT (XT,YT,2)
        ISGERASE=ISGERASE+1
        PSGERASE(ISGERASE,1)=PSMK(I,1)
        PSGERASE(ISGERASE,2)=PSMK(I,2)
        ASGERASE(ISGERASE)=TASMK(I,2)
100   CONTINUE
      RETURN
      END


      SUBROUTINE ERSMKCURT
C     ERASES ALL PREVIOUSLY DRAWN SMOKE GRENADE CURTAINS
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/EDSMKG/ISGERASE,PSGERASE(400,2),ASGERASE(400)
C
C
C     ISKIP=0
C
C     DRAWS A BLACK  LINE FOR EACH SMOKE CURTAIN LOCATION.
C
      CALL FACTOR (ZFACT)
      GXUNIT=11.0/NX*SCFACT
      GYUNIT=8.5/NY
      XUNIT=GXUNIT/SCALE
      YUNIT=GYUNIT/SCALE

      DO 100 I=1,ISGERASE
C     CURTAIN LENGTH IS 150 FEET, BUT PLOTTED AT TWICE SCALE (LIKE
C     WITH VEHICLES SO THEY CAN BE SEEN BETTER)
      WCURT=150.*.3048/1000.*2.0
      PI=3.14159
      CALL NEWPEN (0)
C     CENTER POINT OF CURTAIN
      XC=PSGERASE(I,1)
      YC=PSGERASE(I,2)
C     CONVERT ORIENTATION AZIMUTH TO AN ANGLE
      CALL FDIR (ASGERASE(I),AZIN)
C     FIND STARTING POINT OF CURTAIN
      X1=XC+COS(AZIN/180.*PI)*WCURT/2.
      Y1=YC+SIN(AZIN/180.*PI)*WCURT/2.
C     ENDING POINT
      X2=XC-COS(AZIN/180.*PI)*WCURT/2.
      Y2=YC-SIN(AZIN/180.*PI)*WCURT/2.
C     DRAW THIS LINE
      IF (XLEN.GT.0.) CALL FACTOR (ZFACTN)
      XS=X1*XUNIT
      YS=Y1*YUNIT
      ISKIP=0
      IF (XLEN.GT.0.) CALL ZOOMIT (XS,YS,ISKIP)
      IF (ISKIP.EQ.1) GO TO 100
      CALL PLOT (XS,YS,3)
      XT=X2*XUNIT
      YT=Y2*YUNIT
      IF (XLEN.GT.0.) CALL ZOOMIT (XT,YT,ISKIP)
      IF (ISKIP.EQ.1) GO TO 100
C     DRAW LINE
      CALL PLOT (XT,YT,2)
100   CONTINUE
      RETURN
      END


      SUBROUTINE DMASK (XS,YS,RMAX1,RMAX2,RMAX3)
C
C     FIND ANGLE AND RANGE TO ALL DUST PUFFS WITHIN MAX RANGE OF
C     THIS SYSTEM SO THAT MASKING CAN BE EVALUATED
C
      COMMON/MASKD/DMSK(2400,3),NDMSK
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX


C     XS,YS   -  SHOOTER COORDINATES
C     RMAX1,2,3 -  MAX RANGE OF SYSTEM WEAPONS
C     DMSK(I,1) -  X COORD OF EACH DUST PUFF
C     DMSK(I,2) -  Y COORD OF EACH DUST PUFF
```

```
C     DMSK(I,3) -  RADIUS FOR EACH DUST PUFF
C     NDMSK     -  NUMBER OF DUST PUFFS WITHIN MAX RANGE
C
      PI=3.14159
      NDMSK=0
      IF (NDUST.GT.0) THEN
C     CYCLE THROUGH ALL DUST PUFFS TO FILL MASKING ARRAY
      RMAXX=0
      IF (RMAX1.GT.RMAXX) RMAXX=RMAX1
      IF (RMAX2.GT.RMAXX) RMAXX=RMAX2
      IF (RMAX3.GT.RMAXX) RMAXX=RMAX2
      RMAXX=RMAXX/1000.0
      DO 20 I=IDUST1,IDUST2
        XD=PTDUST(I,1)
        YD=PTDUST(I,2)
        RADIUS=PTDUST(I,3)*EXPUFF
      RANGE=((XD-XS)**2+(YD-YS)**2)**.5
        IF ((RANGE-RADIUS).GT.RMAXX) GO TO 20
        NDMSK=NDMSK+1
CC      FIND ANGLE TO THIS DUST PUFF
C       TOP=YD-YS
C       BOT=XD-XS
C       IF (BOT.EQ.0.0) BOT=BOT+.001
C       DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
C       IF (BOT.LT.0.0) THEN
C            IF (TOP.LT.0.0) DIR=DIR+180.
C            IF (TOP.GT.0.0) DIR=180.-DIR
C       ELSE
C            IF (TOP.LT.0.0) DIR=360.-DIR
C       ENDIF
CC      CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
C       CALL FAZ (DIR,AZIN)
        DMSK(NDMSK,1)=XD
        DMSK(NDMSK,2)=YD
        DMSK(NDMSK,3)=RADIUS
20      CONTINUE
        ENDIF
        RETURN
        END


        SUBROUTINE DSEE (XS,YS,XT,YT,AZIMUTH,JSKIP)
C       EVALUATES DUST MASKING BETWEEN SHOOTER AND TARGET
C
C       XS,YS   SHOOTER REAL WORLD COORDINATES
C       XT,YT   TARGET  REAL WORLD COORDINATES
C       AZIN    AZIMUTH TO TARGET
C       JSKIP   FLAG TO SKIP THIS TARGET IF MASKED (=0 TO SKIP)

        COMMON/ELEVA/CONT(20),IELEV(10000)
        COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
        COMMON/MASKD/DMSK(2400,3),NDMSK

        DOUBLE PRECISION ANGLE,ANGLEI

        PI=3.14159

C       WRITE (10,*) 'NDMSK ',NDMSK
        IF (NDMSK.GT.0) THEN
C
C       FIND SHOOTER AND TARGET GRID COORDINATES
C       TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
C       CENTER TO CENTER ABOUT THE SCALE.
        RRANGE=((YT-YS)**2+(XT-XS)**2)**.5
        IXS=NINT(XS/SCALE)
        IYS=NINT(YS/SCALE)
        IXT=NINT(XT/SCALE)
        IYT=NINT(YT/SCALE)
        IF (IXS.LT.1) IXS=1
        IF (IXT.LT.1) IXT=1
        IF (IYS.LT.1) IYS=1
        IF (IYT.LT.1) IYT=1
        IF (IXS.GT.NX) IXS=NX
        IF (IXT.GT.NX) IXT=NX
        IF (IYS.GT.NY) IYS=NY
        IF (IYT.GT.NY) IYT=NY
C       X1=IXS*SCALE
C       X2=IXT*SCALE
C       Y1=IYS*SCALE
C       Y2=IYT*SCALE
C       RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5
C
C       FIND SHOOTER AND TARGET ELEVATION
        IS=NY*(IXS-1)+IYS
        SELEV=CONT(IELEV(IS))
        IT=NY*(IXT-1)+IYT
        TELEV=CONT(IELEV(IT))
C
C       ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
C       FOR VEHICLE HEIGHT
        SELEV=SELEV+.002
        TELEV=TELEV+.002
C
C       FIND ELEVATION ANGLE TO THIS TARGET
        TOP=TELEV-SELEV
        BOT=RRANGE
        ANGLE=ATAN(TOP/BOT)
C
        DO 200 I=1,NDMSK
C       WRITE (10,*) 'PUFF ',I
        IF (JSKIP.EQ.0) GO TO 300
C
C       CYCLE THROUGH ALL DUST PUFFS
C       TO EVALUATE DUST MASKING, FIND THE LEFT AND RIGHT AZIMUTH
C       BRACKETING EACH DUST PUFF WITHIN RANGE.  DROP IF OUT OF ANGLE
RANGE.
C       THEN FIND THE ELEVATION ANGLE TO THE TOP OF THE DUST PUFF.
C       DROP PUFF IF NOT HIGH ENOUGH
C
C       EVALUATE RANGE TO DUST PUFF
        XP=DMSK(I,1)
        YP=DMSK(I,2)
C       WRITE (10,*) 'XP YP ', XP,YP
        RADIUS=DMSK(I,3)
C       WRITE (10,*) 'RADIUS ',RADIUS
      RP=((YP-YS)**2+(XP-XS)**2)**.5
C       WRITE (10,*) 'PUFF RANGE ',RP,' TARGET RANGE ',RRANGE
        IF ((RP-RADIUS).GT.RRANGE) GO TO 200
C
C       EVALUATE ANGLES TO DUST PUFF
        TOP=YP-YS
        BOT=XP-XS
        IF (TOP.EQ.0.0.AND.BOT.EQ.0.0) JSKIP=0
        IF (BOT.EQ.0.0) BOT=BOT+.001
        DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
        IF (BOT.LT.0.0) THEN
             IF (TOP.LT.0.0) DIR=DIR+180.
             IF (TOP.GT.0.0) DIR=180.-DIR
        ELSE
             IF (TOP.LT.0.0) DIR=360.-DIR
        ENDIF
C       CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
        CALL FAZ (DIR,AZIN)
        IF (RADIUS.LT.0.030) RADIUS=0.030
        IF (RP.LT.RADIUS) RP=RADIUS
        DELAZ=ASIN(RADIUS/RP)*180./PI
        AZLFT=AZIN-DELAZ
        AZRGT=AZIN+DELAZ
        IF (AZLFT.LT.0.0) AZLFT=AZLFT+360.0
        IF (AZRGT.LT.0.0) AZRGT=AZRGT+360.0
        IF (AZLFT.GT.360.0) AZLFT=AZLFT-360.0
        IF (AZRGT.GT.360.0) AZRGT=AZRGT-360.0
        IF (AZLFT.GT.AZRGT) AZRGT=AZRGT+360.
        IF (AZIMUTH.LT.0.0) AZIMUTH=AZIMUTH+360.0
        IF (AZIMUTH.GT.360.0) AZIMUTH=AZIMUTH-360.0
C       WRITE (10,*)'AZLEFT',AZLFT,' AZRIGHT ',AZRGT,' AZIMUTH ',AZIMUTH
        IF (AZIMUTH.LT.AZLFT.OR.AZIMUTH.GT.AZRGT) GO TO 200
C
C       EVALUATE HEIGHT OF DUST PUFF
C
C       FIND ELEVATION OF THE GROUND UNDER THIS PUFF
        IXP=NINT(XP/SCALE)
        IYP=NINT(YP/SCALE)
        IF (IXP.LT.1) IXP=1
        IF (IYP.LT.1) IYP=1
        IF (IXP.GT.NX) IXP=NX
        IF (IYP.GT.NY) IYP=NY
C       PX2=IXP*SCALE
C       PY2=IYP*SCALE
C       PRANGE=((PY2-Y1)**2+(PX2-X1)**2)**.5
        IP=NY*(IXP-1)+IYP
        PELEV=CONT(IELEV(IP))
C       ADD TO THIS ELEVATION THE HEIGHT OF THE PUFF
C       MAX PUFF HEIGHT IS 30 METERS (100 FEET)
        HEIGHT=RADIUS
        IF (RADIUS.GT.0.030) HEIGHT=0.030
        PELEV=PELEV+HEIGHT
C       FIND ELEVATION ANGLE TO THIS DUST PUFF
        TOP=PELEV-SELEV
C       BOT=RANGE
        BOT=RP
        ANGLEI=ATAN(TOP/BOT)
C       WRITE (10,*) 'TARGET ELEVATION ANGLE ',ANGLE
C       WRITE (10,*) 'PUFF ELEVATION ANGLE ',ANGLEI
        IF (ANGLEI.GT.ANGLE) JSKIP=0
200     CONTINUE

        ENDIF

300     RETURN
        END


        SUBROUTINE ESMASK (XS,YS,RMAX1,RMAX2,RMAX3)
C
C       FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS WITHIN
C       MAX RANGE OF THIS SYSTEM SO THAT MASKING CAN BE EVALUATED
C
        COMMON/MASKES/ESMSK(2400,3),NESMSK
        COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX


C       XS,YS     -  SHOOTER COORDINATES
C       RMAX1,2,3 -  MAX RANGE OF SYSTEM WEAPONS
C       ESMSK(I,1) -  X COORD OF EACH ENGINE SMOKE PUFF
C       ESMSK(I,2) -  Y COORD OF EACH ENGINE SMOKE PUFF
C       ESMSK(I,3) -  RADIUS FOR EACH ENGINE SMOKE PUFF
C       NESMSK     -  NUMBER OF ENGINE SMOKE PUFFS WITHIN MAX RANGE
C
        PI=3.14159
```

```
      NESMSK=0
      IF (NENG.GT.0) THEN
C     CYCLE THROUGH ALL SMOKE PUFFS TO FILL MASKING ARRAY
      RMAXX=0
      IF (RMAX1.GT.RMAXX) RMAXX=RMAX1
      IF (RMAX2.GT.RMAXX) RMAXX=RMAX2
      IF (RMAX3.GT.RMAXX) RMAXX=RMAX2
      RMAXX=RMAXX/1000.
      DO 20 I=IENG1,IENG2
        XD=PTENG(I,1)
        YD=PTENG(I,2)
        RADIUS=PTENG(I,3)*EXPUFF
      RANGE=((XD-XS)**2+(YD-YS)**2)**.5
      IF ((RANGE-RADIUS).GT.RMAXX) GO TO 20
      NESMSK=NESMSK+1
CC      FIND ANGLE TO THIS SMOKE PUFF
C       TOP=YD-YS
C       BOT=XD-XS
C       IF (BOT.EQ.0.0) BOT=BOT+.001
C       DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
C       IF (BOT.LT.0.0) THEN
C           IF (TOP.LT.0.0) DIR=DIR+180.
C           IF (TOP.GT.0.0) DIR=180.-DIR
C       ELSE
C           IF (TOP.LT.0.0) DIR=360.-DIR
C       ENDIF
CC      CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
C       CALL FAZ (DIR,AZIN)
      ESMSK(NESMSK,1)=XD
      ESMSK(NESMSK,2)=YD
      ESMSK(NESMSK,3)=RADIUS
20    CONTINUE
      ENDIF
      RETURN
      END


      SUBROUTINE ESSEE (XS,YS,XT,YT,AZIMUTH,,ISKIP)
C     EVALUATES ENGINE SMOKE MASKING BETWEEN SHOOTER AND TARGET
C
C     XS,YS   SHOOTER REAL WORLD COORDINATES
C     XT,YT   TARGET  REAL WORLD COORDINATES
C     AZIN    AZIMUTH TO TARGET
C     ISKIP   FLAG TO SKIP THIS TARGET IF MASKED (=0 TO SKIP)

      COMMON/ELEVA/CONT(20),IELEV(10000)
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/MASKES/ESMSK(2400,3),NESMSK

      DOUBLE PRECISION ANGLE,ANGLEI

      PI=3.14159

      IF (NESMSK.GT.0) THEN
C
C     FIND SHOOTER AND TARGET GRID COORDINATES
C     TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
C     CENTER TO CENTER ABOUT THE SCALE.
      RRANGE=((YT-YS)**2+(XT-XS)**2)**.5
      IXS=NINT(XS/SCALE)
      IYS=NINT(YS/SCALE)
      IXT=NINT(XT/SCALE)
      IYT=NINT(YT/SCALE)
      IF (IXS.LT.1) IXS=1
      IF (IXT.LT.1) IXT=1
      IF (IYS.LT.1) IYS=1
      IF (IYT.LT.1) IYT=1
      IF (IXS.GT.NX) IXS=NX
      IF (IXT.GT.NX) IXT=NX
      IF (IYS.GT.NY) IYS=NY
      IF (IYT.GT.NY) IYT=NY
C     X1=IXS*SCALE
C     X2=IXT*SCALE
C     Y1=IYS*SCALE
C     Y2=IYT*SCALE
C     RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5
C
C     FIND SHOOTER AND TARGET ELEVATION
      IS=NY*(IXS-1)+IYS
      SELEV=CONT(IELEV(IS))
      IT=NY*(IXT-1)+IYT
      TELEV=CONT(IELEV(IT))
C
C     ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
C     FOR VEHICLE HEIGHT
      SELEV=SELEV+.002
      TELEV=TELEV+.002
C
C     FIND ELEVATION ANGLE TO THIS TARGET
      TOP=TELEV-SELEV
      BOT=RRANGE
      ANGLE=ATAN(TOP/BOT)
C
      DO 200 I=1,NESMSK
      IF (JSKIP.EQ.0) GO TO 300
C
C     CYCLE THROUGH ALL ENGINE SMOKE PUFFS
C     TO EVALUATE MASKING, FIND THE LEFT AND RIGHT AZIMUTH
C     BRACKETING EACH PUFF WITHIN RANGE.  DROP IF OUT OF ANGLE RANGE.
C     THEN FIND THE ELEVATION ANGLE TO THE TOP OF THE PUFF.
C     DROP PUFF IF NOT HIGH ENOUGH
C
C     EVALUATE RANGE TO SMOKE PUFF
      XP=ESMSK(I,1)
      YP=ESMSK(I,2)
      RADIUS=ESMSK(I,3)
      RP=((YP-YS)**2+(XP-XS)**2)**.5
      IF ((RP-RADIUS).GT.RRANGE) GO TO 200
C
C     EVALUATE ANGLES TO SMOKE PUFF
      TOP=YP-YS
      BOT=XP-XS
      IF (TOP.EQ.0.0.AND.BOT.EQ.0.0) JSKIP=0
      IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
      IF (BOT.LT.0.0) THEN
          IF (TOP.LT.0.0) DIR=DIR+180.
          IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
          IF (TOP.LT.0.0) DIR=360.-DIR
      ENDIF
C     CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
      CALL FAZ (DIR,AZIN)
      IF (RADIUS.LT.0.030) RADIUS=0.030
      IF (RP.LT.RADIUS) RP=RADIUS
      DELAZ=ASIN(RADIUS/RP)*180./PI
      AZLFT=AZIN-DELAZ
      AZRGT=AZIN+DELAZ
      IF (AZLFT.LT.0.0) AZLFT=AZLFT+360.0
      IF (AZRGT.LT.0.0) AZRGT=AZRGT+360.0
      IF (AZLFT.GT.360.0) AZLFT=AZLFT-360.0
      IF (AZRGT.GT.360.0) AZRGT=AZRGT-360.0
      IF (AZLFT.GT.AZRGT) AZRGT=AZRGT+360.
      IF (AZIMUTH.LT.0.0) AZIMUTH=AZIMUTH+360.0
      IF (AZIMUTH.GT.360.0) AZIMUTH=AZIMUTH-360.0
      IF (AZIMUTH.LT.AZLFT.OR.AZIMUTH.GT.AZRGT) GO TO 200
C
CC    EVALUATE HEIGHT OF DUST PUFF
C
C     FIND ELEVATION OF THE GROUND UNDER THIS PUFF
      IXP=NINT(XP/SCALE)
      IYP=NINT(YP/SCALE)
      IF (IXP.LT.1) IXP=1
      IF (IYP.LT.1) IYP=1
      IF (IXP.GT.NX) IXP=NX
      IF (IYP.GT.NY) IYP=NY
C     PX2=IXP*SCALE
C     PY2=IYP*SCALE
C     PRANGE=((PY2-Y1)**2+(PX2-X1)**2)**.5
      IP=NY*(IXP-1)+IYP
      PELEV=CONT(IELEV(IP))
C     ADD TO THIS ELEVATION THE HEIGHT OF THE PUFF
C     MAX PUFF HEIGHT IS 30 METERS (100 FEET)
      HEIGHT=RADIUS
      IF (RADIUS.GT.0.030) HEIGHT=0.030
      PELEV=PELEV+HEIGHT
C     FIND ELEVATION ANGLE TO THIS DUST PUFF
      TOP=PELEV-SELEV
C     BOT=PRANGE
      BOT=RP
      ANGLEI=ATAN(TOP/BOT)
      IF (ANGLEI.GT.ANGLE) JSKIP=0
200   CONTINUE

      ENDIF

300   RETURN
      END


      SUBROUTINE PKLESS (KSIDE,IS,IT,RNG,BORE,TOF,PK)
C
C     REDUCES PK FOR WEAPONS WHICH CANNOT RANGE TO THE TARGET
C     DUE TO DUST OR SMOKE OBSCURANTS ON THE LINE OF SIGHT
C
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     &,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)

      DIMENSION SIG(40),AREA(40)
      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
        DATA SIG /0.,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.,1.1,1.2,1.3,1.4,1.5,
     & 1.6,1.7,1.8,1.9,2.,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3.,3.1,
     & 3.2,3.3,3.4,3.5,3.6,3.7,3.8,3.9/
        DATA AREA /.5,.5398,.5793,.6179,.6554,.6915,.7258,.758,.7881,
     & .8159,.8413,.8643,.8849,.9032,.9192,.9332,.9452,.9554,.9641,
     & .9713,.9772,.9821,.9861,.9893,.9918,.9938,.9953,.9965,.9974,
     & .9981,.9987,.9990,.9993,.9995,.9997,.9998,.9998,.9999,.9999,
     & 1.000/

      PI=3.14159

      RANGE=RNG*1000.

C     WRITE (*,*) (SIG(I),I=1,40)
C     WRITE (*,*) (AREA(I),I=1,40)

C     WRITE (*,*) 'RANGE  BORE  TOF  PKIN'
C     WRITE (*,*) RANGE,BORE,TOF,PK
C
      IF (KSIDE.EQ.1) THEN
C     BLUE SHOOTER / RED TARGET
      SS=REAL(ISPDB(IS))/3600.*1000.
```

```
            ST=REAL(ISPOR(IT))/3600.*1000.
          CALL FDIR (REAL(IBSA(IS,2)),AZS)
          CALL FDIR (REAL(IRSA(IT,2)),AZT)
          AZS=AZS/180.*PI
          AZT=AZT/180.*PI
          XS=BXYA(IS,1)
          YS=BXYA(IS,2)
          XT=RXYA(IT,1)
          YT=RXYA(IT,2)
          ELSE
            SS=REAL(ISPOR(IS))/3600.*1000.
            ST=REAL(ISPOB(IT))/3600.*1000.
          CALL FDIR (REAL(IRSA(IS,2)),AZS)
          CALL FDIR (REAL(IBSA(IT,2)),AZT)
          AZS=AZS/180.*PI
          AZT=AZT/180.*PI
          XS=RXYA(IS,1)
          YS=RXYA(IS,2)
          XT=BXYA(IT,1)
          YT=BXYA(IT,2)
          ENDIF

C         DEFINE UNIT VECTOR CONNECTING SHOOTER AND TARGET
          RMAG=((XS-XT)**2+(YS-YT)**2)**.5
          DX=XS-XT
          DY=YS-YT
          UI=DX/RMAG
          UJ=DY/RMAG
C
C         FIND RELATIVE CROSSING SPEED OF SHOOTER AND TARGET
C         BY TAKING CROSS PRODUCT OF EACH VEHICLE VELOCITY VECTOR
C         ONTO THIS UNIT VECTOR.  THEN TAKE DIFFERENCE IN CROSS
C         PRODUCTS.
C

C         WRITE (*,*) 'SS  ST  AZS  AZT'
C         WRITE (*,*) SS,ST,AZS,AZT
C         WRITE (*,*) 'UI UJ'
C         WRITE (*,*) UI,UJ

          AS=SS*COS(AZS)
          BS=SS*SIN(AZS)
          AT=ST*COS(AZT)
          BT=ST*SIN(AZT)

C         THE CROSS PRODUCT OF THESE VECTOR QUANTITIES IS THE CROSSING
C         SPEED
          SCROSS=ABS((AS*UJ-BS*UI)-(AT*UJ-BT*UI))

C         WRITE (*,*) 'SCROSS ',SCROSS

C         FIND TRUE ANGULAR RATE OF TARGET
C         OMEGAT=SCROSS/RANGE

C         FIND TIME TO TARGET RANGE
C         TB=TOF
          TT=RANGE/BORE*TOF
C
C         SINCE THERMAL SIGHTS HAVE A RETICLE, GUNNERS CAN USUALLY
C         ESTIMATE TARGET RANGE BASED ON THE SIZE OF THE TARGET WITHIN
C         THE RETICLE AND THEN BRACKET THE RANGE.  THIS WAY A BETTER RANGE
C         ESTIMATION CAN BE ENTERED INTO THE FIRE CONTROL COMPUTER OTHER
C         THAN THE BASIC BATTLESIGHT RANGE.  FOR THIS REASON, THE ENTERED
C         BATTLESIGHT RANGE IS ONLY USED TO GET TIME OF FLIGHT DATA.  AT
C         THIS POINT, THE GUNNER ESTIMATES A BETTER BATTLESIGHT RANGE
C         BASED ON THE FOLLOWING RANGE BRACKETS.
          TB=1200./BORE*TOF
          IF (RANGE.GE.3500.) BORE=4000.
          IF (RANGE.GE.2750.AND.RANGE.LT.3500.) BORE=3000.
          IF (RANGE.GE.2250.AND.RANGE.LT.2750.) BORE=2500.
          IF (RANGE.GE.1750.AND.RANGE.LT.2250.) BORE=2000.
          IF (RANGE.GE.1350.AND.RANGE.LT.1750.) BORE=1500.
          IF (RANGE.LT.1350.) BORE=1200.
          TB=BORE/1200.*TB

C         FIND DISTANCE TARGET HAS CROSSED IN THIS TIME
C         DT=TT*SCROSS

C         FIND TRUE LEAD ANGLE TO TARGET (MILS)
C         AT=DT/RANGE*1000.
          AT=SCROSS/BORE*TB

C         APPLY THE TRUE ANGULAR RATE TO THE BATTLESIGHT RANGE
C         TO FIND THE DISTANCE THE FIRE CONTROL COMPENSATES.
C         DB=RANGE*OMEGAT*TB

C         FIND LEAD ANGLE TO BATTLESIGHT TARGET (MILS)
C         AB=DB/BORE*1000.
          AB=SCROSS/RANGE*TB

C         FIND THE DIFFERENCE IN THE LEAD ANGLES (MILS)
          DELAIM=ABS(AT-AB)

C         WRITE (*,*) 'DELAIM ',DELAIM

C         FIND THE AIMPOINT OFFSET DISTANCE (METERS)
C         DIST=DELAIM*RANGE/1000.
          DIST=DELAIM*RANGE

C         WRITE (*,*) 'DIST ',DIST

C         ASSUME WEAPON PK/H=1.0 AND THEREFORE HIT PROBABILITY IS
C         EQUAL TO PK/S AS ENTERED ABOVE
C         FIND DEFLECTION AND RANGE ERROR STANDARD DEVIATIONS
          DAREA=PK**.5
C         DAREA=DAREA
          DO 10 I=1,39
          PROB1=(AREA(I)-0.5)*2.0
          PROB2=(AREA(I+1)-0.5)*2.0
          IF (DAREA.LE.PROB2.AND.DAREA.GE.PROB1) GO TO 20
10        CONTINUE
20        CONTINUE
          BOT=PROB2-PROB1
          TOP=SIG(I+1)-SIG(I)
          DSIG=SIG(I)+(DAREA-PROB1)*TOP/BOT
          RSIG=DSIG

C         FIND TARGET DEFLECTION LEFT AND RIGHT LIMITS IN TERMS
C         OF STANDARD DEVIATIONS WHEN HORIZONTAL AIMPOINT ERROR IS INCLUDED
C         ASSUME TARGET IS THREE BY THREE METERS SQUARE
C         THEREFORE FIND ADDITIONAL DEFLECTION ERROR IN TERMS OF A TARGET
C         WIDTH STANDARD DEVIATION
          DIST=ABS(DIST/1.5)
          BLFT=-DSIG-DIST
          BRGT=DSIG-DIST
          SLFT=ABS(BLFT)
          SRGT=ABS(BRGT)

          J=1
          K=1
C         FIND PROBABILITIES ASSOCIATED WITH THESE SIGMA BOUNDARIES
          DO 30 I=1,39
          IF (SLFT.LE.SIG(I+1).AND.SLFT.GE.SIG(I)) THEN
          A2=AREA(I)-0.5
          J=I
          ENDIF
          IF (SRGT.LE.SIG(I+1).AND.SRGT.GE.SIG(I)) THEN
          A1=AREA(I)-0.5
          K=I
          ENDIF
30        CONTINUE

          TOP=AREA(J+1)-AREA(J)
          BOT=SIG(J+1)-SIG(J)
          A2=A2+(SLFT-SIG(J))*TOP/BOT

          TOP=AREA(K+1)-AREA(K)
          BOT=SIG(K+1)-SIG(K)
          A1=A1+(SRGT-SIG(K))*TOP/BOT

          IF (SLFT.GE.SIG(40)) A2=0.5
        IF (SRGT.GE.SIG(40)) A1=0.5

          IF (BRGT.GE.0.0) THEN
          DPROB=A1+A2
          ELSE
          DPROB=A2-A1
          ENDIF

C         VERTICAL AIMPOINT ERROR BASED ON TIME OF FLIGHT TO TRUE
C         RANGE WHEN AIMING AT BATTLESIGHT RANGE
          G=9.807

C         FIND MUZZLE VELOCITY (M/S)
C         V0=BORE/TOF
        V0=BORE/TB

C         FIND SUPER ELEVATION ANGLE TO TARGET CENTER AT BATTLESIGHT
C         RANGE
          SE=0.5*ASIN(BORE/V0**2*G)

          SE1=SE*180./PI

C         WRITE (*,*) 'SUPER ELEVATION 1',SE1

C         FIND HEIGHT ABOVE OR BELOW TARGET CENTER AT TRUE TARGET RANGE
          HEIGHT=-G/2.*TT**2+V0*SIN(SE)*TT

C         WRITE (*,*) 'HEIGHT ',HEIGHT

C         EVALUATE NEW RANGE ERROR PROBABILTIES

C         FIND TARGET RANGE TOP AND BOTTOM LIMITS IN TERMS
C         OF STANDARD DEVIATIONS WHEN THE VERTICAL AIMPOINT ERROR IS INCLUDED
C         ASSUME TARGET IS THREE BY THREE METERS SQUARE
          VERR=ABS(HEIGHT/1.5)
          BTOP=RSIG-VERR
          BBOT=-RSIG-VERR

C         FIND THESE BOUNDARIES IN TERMS OF STANDARD DEVIATIONS
          STOP=ABS(BTOP)
          SBOT=ABS(BBOT)

          J=1
          K=1
C         FIND PROBABILITIES ASSOCIATED WITH THESE SIGMA BOUNDARIES
          DO 40 I=1,39
          IF (STOP.LE.SIG(I+1).AND.STOP.GE.SIG(I)) THEN
          A1=AREA(I)-0.5
          J=I
          ENDIF
          IF (SBOT.LE.SIG(I+1).AND.SBOT.GE.SIG(I)) THEN
          A2=AREA(I)-0.5
          K=I
```

```
      ENDIF
40    CONTINUE

      TOP=AREA(J+1)-AREA(J)
      BOT=SIG(J+1)-SIG(J)
      A1=A1+(STOP-SIG(J))*TOP/BOT

      TOP=AREA(K+1)-AREA(K)
      BOT=SIG(K+1)-SIG(K)
      A2=A2+(SBOT-SIG(K))*TOP/BOT

      IF (STOP.GE.SIG(40)) A1=0.5
      IF (SBOT.GE.SIG(40)) A2=0.5

      IF (STOP.GE.0.0) THEN
       RPROB=A1+A2
      ELSE
       RPROB=A2-A1
      ENDIF

C     CALCULATE NEW PK BASED ON BATTLESIGHT RANGE ERROR

      PK=DPROB*RPROB

C     WRITE (*,*) 'PKOUT ',PK,' DPROB ',DPROB,' RPROB ',RPROB

      RETURN
      END



      SUBROUTINE LIVING

C     FOR AIDING IN THE DECISION PROCESS WHETHER ONE WOULD
C     SHOOT OR HOLD FIRE EVEN WITH A VERY LOW PK, HAVING A
C     ROUGH IDEA OF HOW MANY THREAT TARGETS EXIST COMPARED TO
C     HOW MANY FRIENDLY VEHICLES EXIST IS A REASONABLE FACTOR
C     TO CONSIDER


      COMMON/ALIVE/NBA,NRA,BTHREAT,RTHREAT
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT

      NBA=0
      NRA=0
      DO 10 I=1,NB
10     IF (IBSYS(I,2).EQ.1.AND.IBSYS(I,1).LE.10) NBA=NBA+1
      DO 20 I=1,NR
20     IF (IRSYS(I,2).EQ.1.AND.IRSYS(I,1).LE.10) NRA=NRA+1
C
C     EVALUATE THREAT PICTURE BASED ON NUMBER OF THEM VERSUS NUMBER OF
C      YOU.  IF THREAT=1, ALWAYS SHOOT; IF THREAT=0, CONSIDER PK AND
C      AMMUNITION FACTORS.
C
      BTHREAT=1
      RTHREAT=1

      RETURN
      END


      SUBROUTINE GRMASK (XS,YS,RMAX1,RMAX2,RMAX3)
C
C     FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS WITHIN
C      MAX RANGE OF THIS SYSTEM SO THAT MASKING CAN BE EVALUATED
C
      COMMON/MASKGR/GRMSK(400,3),NGRMSK
       COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX


C      XS,YS     -  SHOOTER COORDINATES
C      RMAX1,2,3 -  MAX RANGE OF SYSTEM WEAPONS
C      GRMSK(I,1) -  X COORD OF EACH CURTAIN CENTER
C      GRMSK(I,2) -  Y COORD OF EACH CURTAIN CENTER
C      GRMSK(I,3) -  ORIENTATION ANGLE (CARTESIAN) OF CURTAIN
C      NGRMSK     -  NUMBER OF SMOKE GRENADE CURTAINS WITHIN MAX RANGE
C
      PI=3.14159
      NGRMSK=0
      IF (NSMK.GT.0) THEN
C      CYCLE THROUGH ALL SMOKE PUFFS TO FILL MASKING ARRAY
       RMAXX=0
       IF (RMAX1.GT.RMAXX) RMAXX=RMAX1
       IF (RMAX2.GT.RMAXX) RMAXX=RMAX2
       IF (RMAX3.GT.RMAXX) RMAXX=RMAX2
       RMAXX=RMAXX/1000.
       DO 20 I=ISMK1,ISMK2
        XD=PSMK(I,1)
        YD=PSMK(I,2)
       RANGE=((XD-XS)**2+(YD-YS)**2)**.5
       IF (RANGE.GT.RMAXX) GO TO 20
       NGRMSK=NGRMSK+1
       CALL FDIR (TASMK(I,2),GRMSK(I,3))
       GRMSK(NGRMSK,1)=XD
       GRMSK(NGRMSK,2)=YD
20     CONTINUE
      ENDIF
      RETURN
      END


      SUBROUTINE GRSEE (XS,YS,XT,YT,AZIMUTH,JSKIP)
C     EVALUATES ENGINE SMOKE MASKING BETWEEN SHOOTER AND TARGET
C
C     XS,YS   SHOOTER REAL WORLD COORDINATES
C     XT,YT   TARGET  REAL WORLD COORDINATES
C     AZIN    AZIMUTH TO TARGET
C     JSKIP   FLAG TO SKIP THIS TARGET IF MASKED (=0 TO SKIP)

      COMMON/ELEVA/CONT(20),IELEV(10000)
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/MASKGR/GRMSK(400,3),NGRMSK

      DOUBLE PRECISION ANGLE,ANGLEI

      PI=3.14159

      IF (NGRMSK.GT.0) THEN
      CALL FDIR (AZIMUTH,DIRT)
      WCURT=150.*.3048/1000.
C
C     FIND SHOOTER AND TARGET GRID COORDINATES
C     TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
C     CENTER TO CENTER ABOUT THE SCALE.
      RRANGE=((YT-YS)**2+(XT-XS)**2)**.5
       IXS=NINT(XS/SCALE)
       IYS=NINT(YS/SCALE)
       IXT=NINT(XT/SCALE)
       IYT=NINT(YT/SCALE)
       IF (IXS.LT.1) IXS=1
       IF (IXT.LT.1) IXT=1
       IF (IYS.LT.1) IYS=1
       IF (IYT.LT.1) IYT=1
       IF (IXS.GT.NX) IXS=NX
       IF (IXT.GT.NX) IXT=NX
       IF (IYS.GT.NY) IYS=NY
       IF (IYT.GT.NY) IYT=NY
C      X1=IXS*SCALE
C      X2=IXT*SCALE
C      Y1=IYS*SCALE
C      Y2=IYT*SCALE
C      RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5
C
C     FIND SHOOTER AND TARGET ELEVATION
       IS=NY*(IXS-1)+IYS
       SELEV=CONT(IELEV(IS))
       IT=NY*(IXT-1)+IYT
       TELEV=CONT(IELEV(IT))
C
C     ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
C     FOR VEHICLE HEIGHT
      SELEV=SELEV+.002
      TELEV=TELEV+.002

C
C     FIND ELEVATION ANGLE TO THIS TARGET
      TOP=TELEV-SELEV
      BOT=RRANGE
       ANGLE=ATAN(TOP/BOT)
C
      DO 200 I=1,NGRMSK
       IF (JSKIP.EQ.0) GO TO 300
C
C     CYCLE THROUGH ALL SMOKE GRENADE CURTAINS
C      TO EVALUATE MASKING, FIND THE LEFT AND RIGHT AZIMUTH
C      BRACKETING EACH CURTAIN WITHIN RANGE.  DROP IF OUT OF ANGLE RANGE.
C      THEN FIND THE ELEVATION ANGLE TO THE TOP OF THE CURTAIN.
C      DROP IF NOT HIGH ENOUGH
C
C     EVALUATE RANGE TO CURTAIN
       XP=GRMSK(I,1)
       YP=GRMSK(I,2)
      RP=((YP-YS)**2+(XP-XS)**2)**.5
       IF (RP.GT.RRANGE) GO TO 200
C
C     EVALUATE ANGLES TO SMOKE CURTAIN
       TOP=YP-YS
       BOT=XP-XS
       IF (TOP.EQ.0.0.AND.BOT.EQ.0.0) JSKIP=0
       IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
       IF (BOT.LT.0.0) THEN
            IF (TOP.LT.0.0) DIR=DIR+180.
            IF (TOP.GT.0.0) DIR=180.-DIR
       ELSE
            IF (TOP.LT.0.0) DIR=360.-DIR
       ENDIF
C
C     FIND LEFT AND RIGHT LIMIT ANGLES OF THIS CURTAIN.
C     FIND THE INCLUDED ANGLE BETWEEN THE CURTAIN AND THE RANGE
C      TO THE CURTAIN.
C      TO DO THIS, TAKE THE CROSS PRODUCT OF THE CURTAIN VECTOR
C      ON TO THE UNIT VECTOR DEFINING THE RANGE TO THE CURTAIN.
      RI=BOT/RP*COS(DIR/180.*PI)
      RJ=TOP/RP*SIN(DIR/180.*PI)
      CI=WCURT/2.*COS(GRMSK(I,3)/180.*PI)
```

```
      CJ=WCURT/2.*SIN(GRMSK(L3)/180.*PI)
      DELTA=ABS(RI*CJ-CI*RJ)
       IF (RP.LT.DELTA) RP=DELTA
      DELAZ=ASIN(DELTA/RP)*180./PI
       AZLFT=DIR-DELAZ
       AZRGT=DIR+DELAZ
       IF (AZLFT.LT.0.0) AZLFT=AZLFT+360.0
       IF (AZRGT.LT.0.0) AZRGT=AZRGT+360.0
       IF (AZLFT.GT.360.0) AZLFT=AZLFT-360.0
       IF (AZRGT.GT.360.0) AZRGT=AZRGT-360.0
        IF (AZLFT.GT.AZRGT) AZRGT=AZRGT+360.
       IF (DIRT.LT.0.0) DIRT=DIRT+360.0
       IF (DIRT.GT.360.0) DIRT=DIRT-360.0
        IF (DIRT.LT.AZLFT.OR.DIRT.GT.AZRGT) GO TO 200
C
C     EVALUATE HEIGHT OF SMOKE GRENADE CURTAIN
C
C     FIND ELEVATION OF THE GROUND UNDER THIS CURTAIN
       IXP=NINT(XP/SCALE)
       IYP=NINT(YP/SCALE)
        IF (IXP.LT.1) IXP=1
        IF (IYP.LT.1) IYP=1
        IF (IXP.GT.NX) IXP=NX
        IF (IYP.GT.NY) IYP=NY
C       PX2=IXP*SCALE
C       PY2=IYP*SCALE
C       PRANGE=((PY2-Y1)**2+(PX2-X1)**2)**.5
        IP=NY*(IXP-1)+IYP
       PELEV=CONT(IELEV(IP))
C       ADD TO THIS ELEVATION THE HEIGHT OF THE PUFF
C       MAX CURTAIN HEIGHT IS 30 METERS (100 FEET)
       HEIGHT=0.030
       PELEV=PELEV+HEIGHT
C       FIND ELEVATION ANGLE TO THIS SMOKE CURTAIN
       TOP=PELEV-SELEV
C         BOT=PRANGE
       BOT=RP
       ANGLEI=ATAN(TOP/BOT)
       IF (ANGLEI.GT.ANGLE) JSKIP=0
200     CONTINUE

       ENDIF

300     RETURN
       END


       SUBROUTINE PRIORITIES
C
C      ENTER ORDERS FOR UNITS TO CHANGE TARGET PRIORITIES
C
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACQB(10,5),ACQR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)
      COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/FIRE/IBPRIME(200,2),IBSECOND(200,2),IRPRIME(200,2),
     & IRSECOND(200,2),WRNGB(200,2),WRNGR(200,2)
      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
      COMMON/OBSCURE/NGRENB(200,2),NGRENR(200,2),IBESMK(200),
     & IRESMK(200)
      COMMON/PRIOTY/KPRIB(200,10),KPRIR(200,10)
C
      DIMENSION LPRI(10)
      CHARACTER*9 BFNAME,RFNAME
C     CHARACTER*80 LINE
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*1 C2
C     CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
      KSIDE=0
10      WRITE (*,*)
       WRITE (*,*) 'TARGET PRIORITY ORDERS'
       WRITE (*,*) 'enter BLUE or RED, or QUIT'
        WRITE (*,*)
       READ (*,*,ERR=10) C2
        IF (C2(1:1).EQ.'B') KSIDE=1
        IF (C2(1:1).EQ.'R') KSIDE=2
        IF (C2(1:1).EQ.'Q') GO TO 1000
        IF (KSIDE.EQ.0) GO TO 10
C
5       WRITE (*,*)
       WRITE (*,*) 'CURRENT PRIORITY FACTORS'
11      FORMAT (1X,A80)
       WRITE (*,*)' UNIT HIERARCHY     IDENTITY        INTEGER FACTORS
     &APPLIED TO PRIORITY ALGORITM'
       WRITE (*,*) '                                                ENEMY
     & SYSTEM NUMBER'
       WRITE (*,*) '                                                1 2
     & 3 4 5 6 7 8 9 10'
       WRITE (*,*)
       IF (C2(1:1).EQ.'B') THEN
C        BLUE
         KF=NRF
         KK=0
         DO 130 K=1,NB
121       FORMAT (1X,I3,1X,A16,A24,10(I3))
         IF (IBSYS(K,2).EQ.1.AND.IBSYS(K,1).LE.10) THEN
          WRITE (*,121) K,BHIER(K),BUNIT(K),(KPRIB(K,KI),KI=1,KF)
          KK=KK+1
          ENDIF
          IF (KK.EQ.20) THEN
              READ (*,*)
              KK=0
          ENDIF
130       CONTINUE
C
         ELSE
C        RED
         KF=NBF
         KK=0
         DO 140 K=1,NR
        IF (IRSYS(K,2).EQ.1.AND.IRSYS(K,1).LE.10) THEN
        WRITE (*,121) K,RHIER(K),RUNIT(K),(KPRIR(K,KI),KI=1,KF)
          KK=KK+1
          ENDIF
          IF (KK.EQ.20) THEN
              READ (*,*)
              KK=0
          ENDIF
140       CONTINUE
         ENDIF

141       WRITE (*,*) 'ENTER COMMAND UNIT (0-RELISTS -1-QUIT)'
         WRITE (*,*)
         READ (*,*,ERR=141) ICOMMAND
          IF (ICOMMAND.EQ.0) GO TO 5
          IF (ICOMMAND.EQ.-1) GO TO 10
C
142       WRITE (*,*) 'ENTER NUMBER RANGE OF SUBORDINATES FOR'
         WRITE (*,*) 'SUB-UNIT ALLOCATION (#START #END) (0 0 FOR NONE)'
          WRITE (*,*)
         READ (*,*,ERR=142) NSUB1,NSUB2
         NSUB=1
         IF (NSUB1.LE.0) THEN
              NSUB1=0
              NSUB2=0
              NSUB=0
         ENDIF
         IF (NSUB2.LE.0) THEN
              NSUB1=0
              NSUB2=0
              NSUB=0
         ENDIF
         IF (NSUB.EQ.0) GO TO 145
C
C143      WRITE (*,*) 'SUB-UNIT ALLOCATIONS'
         ISUB=2
C
145       WRITE (*,*)
         WRITE (*,*) 'ENTER THE',KF,' TARGET PRIORITY FACTORS'
         WRITE (*,*)
         READ (*,*,ERR=145) (LPRI(KI),KI=1,KF)
148       WRITE (*,*)
C
C        BLUE
         IF (KSIDE.EQ.1) THEN
          IF (IBSYS(ICOMMAND,2).EQ.1) THEN
           DO 147 KI=1,KF
147          KPRIB(ICOMMAND,KI)=LPRI(KI)
          ENDIF
            IF (ISUB.EQ.2) THEN
C             EVERY ONE GETS THE SAME ORDERS
              DO 175 I=NSUB1,NSUB2
                IF (IBSYS(I,2).EQ.1) THEN
                 DO 148 KI=1,KF
148                KPRIB(I,KI)=LPRI(KI)
                ENDIF
175       CONTINUE
            ENDIF
         ENDIF
C
C        RED
         IF (KSIDE.EQ.2) THEN
          IF (IRSYS(ICOMMAND,2).EQ.1) THEN
           DO 176 KI=1,KF
176          KPRIR(ICOMMAND,KI)=LPRI(KI)
          ENDIF
            IF (ISUB.EQ.2) THEN
C             EVERY ONE GETS THE SAME ORDERS
              DO 185 I=NSUB1,NSUB2
                IF (IRSYS(I,2).EQ.1) THEN
                 DO 177 KI=1,KF
177                KPRIR(I,KI)=LPRI(KI)
                ENDIF
185       CONTINUE
            ENDIF
         ENDIF
```

```
C
      GO TO 141

1000  CONTINUE
      RETURN
      END


      SUBROUTINE GRADE (KSIDE,K)
      COMMON/ACEL/ACELB(10,5),ACELR(10,5),GRADEB(200),GRADER(200)
      COMMON/ELEVA/CONT(20),IELEV(10000)
       COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM

      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

       PI=3.14159

       IF (KSIDE.EQ.1) THEN
         XS=BXYA(K,1)
         YS=BXYA(K,2)
         AZ=IBSA(K,2)
       ELSE
         XS=RXYA(K,1)
         YS=RXYA(K,2)
         AZ=IRSA(K,2)
       ENDIF

C      CONVERT DIRECTION AZIMUTH TO CARTESIAN ANGLE
       CALL FDIR (AZ,DIR)

C      FIND VEHICLE GRID COORDINATES
C      TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
C      CENTER TO CENTER ABOUT THE SCALE.
       IXS=NINT(XS/SCALE)
       IYS=NINT(YS/SCALE)
       IF (IXS.LT.1) IXS=1
       IF (IYS.LT.1) IYS=1
       IF (IXS.GT.NX) IXS=NX
       IF (IYS.GT.NY) IYS=NY
       X1=IXS*SCALE
       Y1=IYS*SCALE

C      FIND NEXT GRID OVER ON THE DIRECTION OF MOVEMENT

       XT=XS+SCALE*COS(DIR/180.*PI)
       YT=YS+SCALE*SIN(DIR/180.*PI)
       IXT=NINT(XT/SCALE)
       IYT=NINT(YT/SCALE)
       IF (IXT.LT.1) IXT=1
       IF (IYT.LT.1) IYT=1
       IF (IXT.GT.NX) IXT=NX
       IF (IYT.GT.NY) IYT=NY
       X2=IXT*SCALE
       Y2=IYT*SCALE

      RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5
C
C      FIND VEHICLE AND NEXT GRID ELEVATION (IT'S ALREADY IN KM)
       IS=NY*(IXS-1)+IYS
       SELEV=CONT(IELEV(IS))
       IT=NY*(IXT-1)+IYT
       TELEV=CONT(IELEV(IT))
C
       TOP=TELEV-SELEV
       BOT=RANGE
       IF (TOP.EQ.0.) GRAD=0.0
       IF (BOT.EQ.0.) GRAD=0.0
       IF (TOP.NE.0.0.AND.BOT.NE.0.0) GRAD=ATAN(TOP/BOT)*180/PI

       IF (KSIDE.EQ.1) THEN
         GRADEB(K)=GRAD
       ELSE
         GRADER(K)=GRAD
       ENDIF

       RETURN
       END


      SUBROUTINE FINDACEL (KSIDE,K,ARATE)
      COMMON/ACEL/ACELB(10,5),ACELR(10,5),GRADEB(200),GRADER(200)
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
       COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT

       IDT=IDTIME

       IF (KSIDE.EQ.1) THEN
        GRADE=GRADEB(K)
         ISWANT=IBSA(K,1)
         ISNOW=ISPDB(K)
       NSYS=IBSYS(K,1)
       IF (NSYS.GT.10) NSYS=1
         V0=ACELB(NSYS,1)
         VG=ACELB(NSYS,3)
         A0=V0/ACELB(NSYS,2)
         AG=VG/ACELB(NSYS,5)
         G=ACELB(NSYS,4)
       ELSE
         GRADE=GRADER(K)
         ISWANT=IRSA(K,1)
         ISNOW=ISPDR(K)
       NSYS=IRSYS(K,1)
       IF (NSYS.GT.10) NSYS=1
         V0=ACELR(NSYS,1)
         VG=ACELR(NSYS,3)
         A0=V0/ACELR(NSYS,2)
         AG=VG/ACELR(NSYS,5)
         G=ACELR(NSYS,4)
       ENDIF

C      WRITE (*,*) 'GRADE  ISWANT  ISNOW  V0  VG  A0  AG'
C      WRITE (*,*) GRADE,ISWANT,ISNOW,V0,VG,A0,AG

C      CHANGE IN ACCELERATION WITH RESPECT TO GRADE
       DADG=(AG-A0)/G

C      MAX ALLOWABLE ACCELERATION BASED ON CURRENT GRADE
       AMAX=DADG*GRADE+A0
       IF (AMAX.LT.0.0) AMAX=0.0
       IF (AMAX.GT.A0) AMAX=A0

C      MAX ALLOWABLE SPEED BASED ON CURRENT GRADE
       VMAX=(VG-V0)/G*GRADE+V0
       IF (VMAX.LT.0.0) VMAX=0.0
       IF (VMAX.GT.V0) VMAX=V0
       IVMAX=INT(VMAX)

C      IF DESIRED SPEED IS GREATER THAN ALLOWABLE SPEED, TRUNCATE
       IF (ISWANT.GT.IVMAX) ISWANT=IVMAX

C      FIND DIFFERENCE BETWEEN ACTUAL AND DESIRED SPEED
       IDV=ISWANT-ISNOW

C      CONVERT TO AN ACCELERATION WITHIN THIS TIME STEP
       DVDT=REAL(IDV/IDT)

       ARATE=DVDT
       IF (ARATE.GT.AMAX) ARATE=AMAX

C      WRITE (*,*) 'GRADE  ISWANT  ISNOW  IVMAX  AMAX  ARATE'
C      WRITE (*,*) GRADE,ISWANT,ISNOW,IVMAX,AMAX,ARATE

       RETURN
       END


       SUBROUTINE STATUS
C
       COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
       COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACQB(10,5),ACQR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)
      COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/FIRE/IBPRIME(200,2),IBSECOND(200,2),IRPRIME(200,2),
     & IRSECOND(200,2),WRNGB(200,2),WRNGR(200,2)
      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
       COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
      COMMON/ALIVE/NBA,NRA,BTHREAT,RTHREAT
      COMMON/ALIVE2/IBA(10),IRA(10),IBT(10),IRT(10),DELB(10),DELR(10)
       COMMON/EMENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
     & ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2)
C
       CHARACTER*9 BFNAME,RFNAME
C      CHARACTER*80 LINE
       CHARACTER*16 BHIER,RHIER,IHIER
       CHARACTER*24 BUNIT,RUNIT
       CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
       CHARACTER*1 C2
C      CHARACTER*1 C2,C3,C4,C5,C8,C7,C8,C9,C10,BLANK
C
10     WRITE (*,*)
       KSIDE=0
       WRITE (*,*) 'CURRENT STATUS OF FORCES'
       WRITE (*,*) 'enter BLUE or RED, CUMULATIVE or QUIT'
       WRITE (*,*)
       READ (*,*,ERR=10) C2
```

```
      IF (C2(1:1).EQ.'B') KSIDE=1
      IF (C2(1:1).EQ.'R') KSIDE=2
      IF (C2(1:1).EQ.'C') KSIDE=3
      IF (C2(1:1).EQ.'Q') GO TO 1000
      IF (KSIDE.EQ.0) GO TO 10
C
            IF (C2(1:1).EQ.'B'.OR.C2(1:1).EQ.'C') THEN
C     BLUE
            IF (KSIDE.EQ.1) THEN
5     WRITE (*,*)
      WRITE (*,*) 'CURRENT STATUS'
      WRITE (*,*) ' UNIT HIERARCHY    IDENTITY              TRUE
     & ENERGY     AMMUNITION'
      WRITE (*,*) '                                  SPEED
     &        WPN #1    #2    #3'
            ENDIF
            KK=0
            NB1=0
            DO 130 K=1,NB
            IF (IBSYS(K,1).GT.10) GO TO 130
            NB1=NB1+1
            IF (KSIDE.EQ.1) THEN
120   FORMAT (1X,I3,1X,A16,A24,I6,1X,F8.3,1X,3(I6,1X))
121   FORMAT (1X,I3,1X,A16,A24,I6,1X,9X,3(I6,1X))
      IF (IBSYS(K,2).EQ.1) THEN
      NSYS=IBSYS(K,1)
            NWP=NWB(NSYS)
      IF (IBSYS(K,1).NE.NBLUE) THEN
      WRITE (*,121) K,BHIER(K),BUNIT(K),ISPDB(K),
     & (NSHOTB(K,IWP),IWP=1,NWP)
      ELSE
      WRITE (*,120) K,B          BUNIT(K),ISPDB(K),ELSYS(K),
     & (NSHOTB(K,IWP),IWP=1,NWP)
      ENDIF
      KK=KK+1
      ENDIF
      IF (KK.EQ.20) THEN
            READ (*,*)
            KK=0
            ENDIF
            ENDIF
130   CONTINUE
C
      ENDIF

            IF (C2(1:1).EQ.'R'.OR.C2(1:1).EQ.'C') THEN
C     RED
122   FORMAT (1X,I3,1X,A16,A24,4(I6,1X))
      WRITE (*,*)
      WRITE (*,*) 'CURRENT STATUS'
      WRITE (*,*) ' UNIT HIERARCHY    IDENTITY              TRUE
     &     AMMUNITION'
      WRITE (*,*) '                                  SPEED
     & WPN #1    #2    #3'
      KK=0
            NR1=0
            DO 140 K=1,NR
            IF (IRSYS(K,1).GT.10) GO TO 140
            NR1=NR1+1
            IF (KSIDE.EQ.2) THEN
            IF (IRSYS(K,2).EQ.1) THEN
      NSYS=IRSYS(K,1)
      NWP=NWR(NSYS)
      WRITE (*,122) K,RHIER(K),RUNIT(K),ISPDR(K),
     & (NSHOTR(K,IWP),IWP=1,NWP)
      KK=KK+1
      ENDIF
      IF (KK.EQ.20) THEN
            READ (*,*)
            KK=0
            ENDIF
            ENDIF
140   CONTINUE
      ENDIF

      IF (C2(1:1).EQ.'C') THEN
      WRITE (*,*)
      WRITE (*,*) 'CUMULATIVE STATS'
      WRITE (*,*)
      CALL LIVING
            ILOSSB=NB1-NBA
            ILOSSR=NR1-NRA
      BLER=0.
      RLER=0.
      IF (ILOSSR.NE.0) BLER=REAL(ILOSSB)/REAL(ILOSSR)
      IF (ILOSSB.NE.0) RLER=REAL(ILOSSR)/REAL(ILOSSB)
      CALL LIVING2
200   FORMAT (1X,A4,I21,2X,I6,2X,I6,7X,F8.2)
201   FORMAT (10(1X,I2,'/',I2,'/',I2))
      WRITE (*,*) 'NUMBER OF ORIGINAL SYSTEMS / CURRENT / LOSSES
     & L.E.R.'
      WRITE (*,*)
C     WRITE (*,*) 'BLUE'
            WRITE (*,200) 'BLUE',NB1,NBA,ILOSSB,BLER
      WRITE (*,*)
C     WRITE (*,*) 'RED'
            WRITE (*,200) 'RED ',NR1,NRA,ILOSSR,RLER
      WRITE (*,*)
      WRITE (*,*) 'BY SYSTEM NUMBER'
      WRITE (*,*)
      WRITE (*,*) 'BLUE'
      WRITE (*,*) 'ORIGINAL / CURRENT / LOSSES'
      WRITE (*,*) '   #1    #2    #3    #4    #5    #6
     & #7    #8    #9    #10'
      WRITE (*,201) (IBT(K),IBA(K),DELB(K),K=1,NBF)
      WRITE (*,*)
      WRITE (*,*) 'RED'
      WRITE (*,*) 'ORIGINAL / CURRENT / LOSSES'
      WRITE (*,*) '   #1    #2    #3    #4    #5    #6
     & #7    #8    #9    #10'
      WRITE (*,201) (IRT(K),IRA(K),DELR(K),K=1,NBF)

      ENDIF
C
      GO TO 10

1000  CONTINUE
      RETURN
      END


      SUBROUTINE LIVING2

      COMMON/ALIVE2/IBA(10),IRA(10),IBT(10),IRT(10),DELB(10),DELR(10)
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACQB(10,5),ACQR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)

      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*9 BFNAME,RFNAME

      DO 5 I=1,10
       IBT(I)=0
       IRT(I)=0
       IBA(I)=0
5      IRA(I)=0

      DO 10 I=1,NB
      NTYPE=IBSYS(I,1)
      IF (NTYPE.GT.10) GO TO 10
        IBT(NTYPE)=IBT(NTYPE)+1
      IF (IBSYS(I,2).EQ.1) IBA(NTYPE)=IBA(NTYPE)+1
10     CONTINUE

      DO 20 I=1,NR
        NTYPE=IRSYS(I,1)
      IF (NTYPE.GT.10) GO TO 20
      IRT(NTYPE)=IRT(NTYPE)+1
      IF (IRSYS(I,2).EQ.1) IRA(NTYPE)=IRA(NTYPE)+1
20     CONTINUE

      DO 30 I=1,NBF
30      DELB(I)=IBT(I)-IBA(I)

        DO 40 I=1,NRF
40      DELR(I)=IRT(I)-IRA(I)

        RETURN
        END


        SUBROUTINE DIGIN
C
C       AUTOMATICALLY STOPS VEHICLES AND PLACES THEM IN
C        DEFILADE POSITION
C
        COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
        COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACQB(10,5),ACQR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMMR(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
      COMMON/DEFILADE/IDEFB(200),IDEFR(200)
C
      CHARACTER*9 BFNAME,RFNAME
C     CHARACTER*90 LINE
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*1 C2
C     CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
       KSIDE=0
10     WRITE (*,*)
      WRITE (*,*) 'enter BLUE or RED, or QUIT'
```

```
      WRITE (*,*)
      READ (*,ERR=10) C2
      IF (C2(1:1).EQ.'B') KSIDE=1
      IF (C2(1:1).EQ.'R') KSIDE=2
      IF (C2(1:1).EQ.'Q') GO TO 1000
      IF (KSIDE.EQ.0) GO TO 10
C
5     WRITE (*,*)
      WRITE (*,*) 'CURRENT DEFILADE STATUS'
      WRITE (*,*)' UNIT HIERARCHY    IDENTITY          1 = YES
     & 0 = NO'
      IF (C2(1:1).EQ.'B') THEN
C      BLUE
       KK=0
       DO 130 K=1,NB
121     FORMAT (1X,I3,1X,A16,A24,7X,I3)
        IF (IBSYS(K,2).EQ.1) THEN
        WRITE (*,121) K,BHIER(K),BUNIT(K),IDEFB(K)
        KK=KK+1
        ENDIF
        IF (KK.EQ.20) THEN
            READ (*,*)
            KK=0
        ENDIF
130    CONTINUE
C
       ELSE
C      RED
       KK=0
       DO 140 K=1,NR
        IF (IRSYS(K,2).EQ.1) THEN
        WRITE (*,121) K,RHIER(K),RUNIT(K),IDEFR(K)
        KK=KK+1
        ENDIF
        IF (KK.EQ.20) THEN
            READ (*,*)
            KK=0
        ENDIF
140    CONTINUE
       ENDIF

141    WRITE (*,*) 'ENTER COMMAND UNIT (0-RELISTS -1-QUIT)'
       WRITE (*,*)
       READ (*,*,ERR=141) ICOMMAND
        IF (ICOMMAND.EQ.0) GO TO 5
        IF (ICOMMAND.EQ.-1) GO TO 10

142    WRITE (*,*) 'ENTER NUMBER RANGE OF SUBORDINATES TO FOLLOW'
       WRITE (*,*) 'THIS ORDER (#START #END) (0 0 FOR NONE)'
       WRITE (*,*)
       READ (*,*,ERR=142) NSUB1,NSUB2
       IF (NSUB1.LE.0) THEN
            NSUB1=0
            NSUB2=0
       ENDIF
       IF (NSUB2.LE.0) THEN
            NSUB1=0
            NSUB2=0
       ENDIF
C
150    WRITE (*,*)
       WRITE (*,*) 'TO DEFILADE THESE VEHICLES ENTER 1, ELSE 0'
       WRITE (*,*)
       READ (*,*,ERR=150) JDEF
        IF (JDEF.LT.0.OR.JDEF.GT.1) GO TO 150
C      DEFILADE EACH VEHICLE
       SPEED=0.0
        IF (KSIDE.EQ.1) THEN

        IBSA(ICOMMAND,1)=INT(SPEED)
        IDEFB(ICOMMAND)=JDEF
              MODEMB(ICOMMAND)=3
        ELSE
        IRSA(ICOMMAND,1)=INT(SPEED)
        IDEFR(ICOMMAND)=JDEF
              MODEMR(ICOMMAND)=3
        ENDIF
        DO 200 I=NSUB1,NSUB2
         IF (KSIDE.EQ.1) THEN
              IF (IBSYS(I,2).EQ.1) THEN
              IBSA(I,1)=INT(SPEED)
              IDEFB(I)=JDEF
                            MODEMB(I)=3
              ENDIF
         ELSE
              IF (IRSYS(I,2).EQ.1) THEN
              IRSA(I,1)=INT(SPEED)
              IDEFR(I)=JDEF
                            MODEMR(I)=3
              ENDIF
         ENDIF
200     CONTINUE

        GO TO 141

1000   CONTINUE
       RETURN
       END


       SUBROUTINE EMCHARGE
C

C      CHARGES EM STORAGE SYSTEM
C
       COMMON/EMENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
     & ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2)
       COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
       COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)

       CHARACTER*16 BHIER,RHIER,IHIER
       CHARACTER*24 BUNIT,RUNIT
       CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

       DO 600 I=1,NB

       NSYS=IBSYS(I,1)
       IF (NSYS.EQ.NBLUE) THEN

        IF (IBSYS(I,2).NE.0) THEN
         TEM(I,MODEEM(I))=TEM(I,MODEEM(I))+IDTIME

         IF (MODEEM(I).EQ.2.AND.ITIME.GE.TE(I)) THEN
         ELSYS(I)=ELSYS(I)+EG(3)*IDTIME

         IF (ELSYS(I).GT.EG(2)) THEN
          ELSYS(I)=EG(2)
          MODEEM(I)=1
          TE(I)=0.0
         ENDIF

         ENDIF

         IF (MODEEM(I).EQ.1.AND.ELSYS(I).LE.EG(4)) THEN
          MODEEM(I)=2
          TE(I)=ITIME+EG(5)
         ENDIF

         ENDIF

         ENDIF

600    CONTINUE
       RETURN
       END



       SUBROUTINE RSIGNAT
C
C      READS SIGNATURE FILE INFORMATION
C
       COMMON/SIGTURS/RSMIN,RSMAX,BDETCT(10,10),RDETCT(10,10),
     & BWSIG(30,10),RWSIG(30,10)

       CHARACTER*80 LINE

303    FORMAT (A80)
503    FORMAT (1X,A80)
       OPEN (1,FILE='SIGNAT.IN')
       READ (1,303) LINE
C      WRITE (*,503) LINE
       READ (1,303) LINE
C      WRITE (*,503) LINE
       READ (1,303) LINE
C      WRITE (*,503) LINE
       READ (1,303) LINE
C      WRITE (*,503) LINE
       READ (1,303) LINE
C      WRITE (*,503) LINE
       READ (1,303) LINE
C      WRITE (*,503) LINE
       READ (1,303) LINE
C      WRITE (*,503) LINE
320    FORMAT (2F8.0)
520    FORMAT (1X,2F8.0)
       READ (1,320) RSMIN,RSMAX
C      WRITE (*,520) RSMIN,RSMAX
       READ (1,303) LINE
C      WRITE (*,503) LINE
       READ (1,303) LINE
C      WRITE (*,503) LINE
       READ (1,303) LINE
C      WRITE (*,503) LINE
321    FORMAT (10F8.2)
521    FORMAT (1X,10F8.2)
       DO 600 I=1,10
       READ (1,321) (BDETCT (I,J),J=1,10)
C      WRITE (*,521) (BDETCT (I,J),J=1,10)
600    CONTINUE
       READ (1,303) LINE
C      WRITE (*,503) LINE
       READ (1,303) LINE
C      WRITE (*,503) LINE
       READ (1,303) LINE
C      WRITE (*,503) LINE
       DO 700 I=1,10
       READ (1,321) (RDETCT (I,J),J=1,10)
C      WRITE (*,521) (RDETCT (I,J),J=1,10)
700    CONTINUE
       DO 1000 IC=1,2
       READ (1,303) LINE
```

```
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      DO 900 I=1,10
      READ (1,303) LINE
C     WRITE (*,503) LINE
      DO 800 J=1,3
      II=(I-1)*3+J
      IF (IC.EQ.1) READ (1,321) (BWSIG(II,JJ),JJ=1,10)
C     IF (IC.EQ.1) WRITE (*,521) (BWSIG(II,JJ),JJ=1,10)
      IF (IC.EQ.2) READ (1,321) (RWSIG(II,JJ),JJ=1,10)
C     IF (IC.EQ.2) WRITE (*,521) (RWSIG(II,JJ),JJ=1,10)
800   CONTINUE
900   CONTINUE
1000  CONTINUE
      CLOSE (1)
      RETURN
      END


      SUBROUTINE CHECKSIG (KSIDE,K,J,R,ISIG)
C
C     CHECKS TO SEE IF TARGET VEHICLE IS DETECTABLE BASED ON
C     SIGNATURES
C
C     IF ISIG IS RETURNED AS 0 THEN TARGET SIGNATURE IS NOT
C     DETECTED, ELSE ISIG = 1
C
C     KSIDE   SEARCHER SIDE
C     K       SEARCHER NUMBER
C     J       TARGET  NUMBER
C     R       TARGET  RANGE (KM)
C     ISIG    SIGNATURE FLAG

      COMMON/SIGTURS/RSMIN,RSMAX,BDETCT(10,10),RDETCT(10,10),
     & BWSIG(30,10),RWSIG(30,10)
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/OBSCURE/NGRENB(200,2),NGRENR(200,2),IBESMK(200),
     & IRESMK(200)
      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
      COMMON/BATTLE/TIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      COMMON/DEFILADE/IDEFB(200),IDEFR(200)

      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT

      ISIG=0

      RANGE=R*1000.0

C     SEE IF TARGET IS MOVING
      IF (KSIDE.EQ.1) ISPD=ISPDR(J)
      IF (KSIDE.EQ.2) ISPD=ISPDB(J)
      IF (ISPD.GE.1) ISIG=1
      IF (ISIG.EQ.1) RETURN

C     SEE IF TARGET IS BLOWING ENGINE SMOKE OR FIRING SMOKE GRENADES
      IF (KSIDE.EQ.1) THEN
       IBLOW1=IRESMK(J)
       IBLOW2=NGRENR(J,2)
      ELSE
       IBLOW1=IBESMK(J)
       IBLOW2=NGRENB(J,2)
      ENDIF
      IF (IBLOW1.EQ.1.OR.IBLOW2.EQ.1) ISIG=1
      IF (ISIG.EQ.1) RETURN

C     SEE IF TARGET IS FIRING A WEAPON AND IF SIGNATURE IS DETECTED
      IF (KSIDE.EQ.1) MODEF=MODEFR(J)
      IF (KSIDE.EQ.2) MODEF=MODEFB(J)
      IF (MODEF.EQ.2.OR.MODEF.EQ.6.OR.MODEF.EQ.8) THEN
C      TARGET HAS ACTIVE WEAPONS
C      FIND WEAPON NUMBER AND ITS SIGNATURE
       IF (KSIDE.EQ.1) THEN
        ITWP=ITGTR(J,2)
        ITSYS=IRSYS(J,1)
        MYSYS=IBSYS(K,1)
        IWANT=(ITSYS-1)*3+ITWP
        DFACT=RWSIG(IWANT,MYSYS)
       ELSE
        ITWP=ITGTB(J,2)
        ITSYS=IBSYS(J,1)
        MYSYS=IRSYS(K,1)
        IWANT=(ITSYS-1)*3+ITWP
        DFACT=BWSIG(IWANT,MYSYS)
       ENDIF
C      INTERPOLATE BASED ON RANGE
       TOP=DFACT-1.0
       BOT=RSMAX-RSMIN
       SLOPE=TOP/BOT
       B=1.0-SLOPE*RSMIN
       DFACT=B+SLOPE*RANGE
       IF (DFACT.LT.0.0) DFACT=0.0
       IF (DFACT.GT.1.0) DFACT=1.0
C      EVALUATE THIS DETECTION PROBABILITY
       DRAND=RND()
C      WRITE (10,*) 'LAUNCH SIGNATURE: DFACT   PROB'
C      WRITE (10,*) DFACT,DRAND
       IF (DFACT.GT.DRAND) ISIG=1
      ENDIF
      IF (ISIG.EQ.1) RETURN

C     SINCE TARGET IS NOT MOVING OR MAKING ITSELF OBVIOUS
C     EVALUATE BASIC TARGET DETECTION PROBABILITY
      RDEF=1.
      IF (KSIDE.EQ.1) THEN
       ITSYS=IRSYS(J,1)
       MYSYS=IBSYS(K,1)
       DFACT=BDETCT(MYSYS,ITSYS)
C      IF TARGET IS IN DEFILADE THIS DETECTION FACTOR WILL BE HALVED
       IF (IDEFR(J).EQ.1) RDEF=2.
      ELSE
       ITSYS=IBSYS(J,1)
       MYSYS=IRSYS(K,1)
       DFACT=RDETCT(MYSYS,ITSYS)
       IF (IDEFB(J).EQ.1) RDEF=2.
      ENDIF
C     INTERPOLATE BASED ON RANGE
      TOP=DFACT-1.0
      BOT=RSMAX-RSMIN
      SLOPE=TOP/BOT
      B=1.0-SLOPE*RSMIN
      DFACT=B+SLOPE*RANGE
      DFACT=DFACT/RDEF
      IF (DFACT.LT.0.0) DFACT=0.0
      IF (DFACT.GT.1.0) DFACT=1.0
C     EVALUATE THIS DETECTION PROBABILITY
      DRAND=RND()
C     WRITE (10,*) 'DETECTION SIGNATURE: DFACT   PROB'
C     WRITE (10,*) DFACT,DRAND
      IF (DFACT.GT.DRAND) ISIG=1

      RETURN
      END


      SUBROUTINE READAPS
C
C     READS ACTIVE PROTECTION SYSTEM FILE INFORMATION
C
      COMMON/APS/IBSAPS(10),IRSAPS(10),BAPSF(10,3),RAPSF(10,3),
     & BAPST,RAPST,IRAMSB(2),IRAMSR(2),IAPSONB(200,2),IAPSONR(200,2)

      CHARACTER*90 LINE

303   FORMAT (A90)
503   FORMAT (1X,A90)
      OPEN (1,FILE='APS.IN')
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
313   FORMAT (10I8)
314   FORMAT (1X,10I8)
      READ (1,313) (IBSAPS(K),K=1,10)
C     WRITE (*,314) (IBSAPS(K),K=1,10)
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,313) (IRSAPS(K),K=1,10)
C     WRITE (*,314) (IRSAPS(K),K=1,10)
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
320   FORMAT (10F8.2)
520   FORMAT (1X,10F8.2)
      READ (1,320) (BAPSF(K,1),K=1,10)
      READ (1,320) (BAPSF(K,2),K=1,10)
      READ (1,320) (BAPSF(K,3),K=1,10)
C     WRITE (*,520) (BAPSF(K,1),K=1,10)
C     WRITE (*,520) (BAPSF(K,2),K=1,10)
C     WRITE (*,520) (BAPSF(K,3),K=1,10)
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,303) LINE
C     WRITE (*,503) LINE
      READ (1,320) (RAPSF(K,1),K=1,10)
      READ (1,320) (RAPSF(K,2),K=1,10)
      READ (1,320) (RAPSF(K,3),K=1,10)
C     WRITE (*,520) (RAPSF(K,1),K=1,10)
```

```
C     WRITE (*,520) (RAPSF(K,2),K=1,10)
C     WRITE (*,520) (RAPSF(K,3),K=1,10)
      READ (1,303) LINE
C     WRITE (*,303) LINE
      READ (1,303) LINE
C     WRITE (*,303) LINE
      READ (1,303) LINE
C     WRITE (*,303) LINE
530   FORMAT (F8.2,2I8)
531   FORMAT (1X,F8.2,2I8)
      READ (1,530) BAPST,(IRAMSB(K),K=1,2)
C     WRITE (*,531) BAPST,(IRAMSB(K),K=1,2)
      READ (1,303) LINE
C     WRITE (*,303) LINE
      READ (1,303) LINE
C     WRITE (*,303) LINE
      READ (1,303) LINE
C     WRITE (*,303) LINE
      READ (1,530) RAPST,(IRAMSR(K),K=1,2)
C     WRITE (*,531) RAPST,(IRAMSR(K),K=1,2)
      CLOSE (1)
C     INITIALIZE ACTIVE PROTECTION TO OFF FOR ALL VEHICLES
      DO 600 I=1,200
       IAPSONB(I,2)=IRAMSB(2)
       IAPSONR(I,2)=IRAMSR(2)
       IAPSONB(I,1)=0
600    IAPSONR(I,1)=0
      RETURN
      END


            SUBROUTINE CHECKAPS (KSIDE,KSYS,J,KWP,PK,IFAPS,RNG)
C
CC    CHECKS TO SEE IF TARGET VEHICLE HAS ACTIVE PROTECTION
CC    TURNED ON AND CAN COUNTER THIS ATTACK
CC
CC    ATTACKER PK IS SET TO 0 IF ACTIVE PROTECTION DEFEATS THIS
CC    ATTACK
CC
CC    KSIDE   SHOOTER SIDE
CC    KSYS    SHOOTER SYSTEM NUMBER
CC    J       TARGET NUMBER
CC    KWP     SHOOTER WEAPON NUMBER
CC    PK      SHOOTER PK AGAINST THIS TARGET
CC    IFAPS   FLAG FOR ACTIVE PROTECTION (SEE OUTPUT1)
CC              ACTIVE PROTECTION FLAG (1 INTERCEPT, 0 MISS, -1
C               N/A THIS WEAPON, 2 OUT OF SHOTS, 3 NO TIME)

      COMMON/APS/IBSAPS(10),IRSAPS(10),BAPSF(10,3),RAPSF(10,3),
     & BAPST,RAPST,IRAMSB(2),IRAMSR(2),IAPSONB(200,2),IAPSONR(200,2)
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
            COMMON/SIMAPS/ISIMB(200),ISIMR(200)
            COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)

            CHARACTER*9 BFNAME,RFNAME
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT

       IF (KSIDE.EQ.1) THEN
        ITSYS=IRSYS(J,1)
        IHVAPS=IRSAPS(ITSYS)
        FAPS=RAPSF(KSYS,KWP)
        NSHOTS=IAPSONR(J,2)
               LIVE=IRSYS(J,2)
               ISIM=ISIMR(J)
               ISIMAX=IRAMSR(1)
               ELSE
        ITSYS=IBSYS(J,1)
        IHVAPS=IBSAPS(ITSYS)
        FAPS=BAPSF(KSYS,KWP)
        NSHOTS=IAPSONB(J,2)
               LIVE=IBSYS(J,2)
               ISIM=ISIMB(J)
               ISIMAX=IRAMSB(1)
              ENDIF

C     ACTIVATE ACTIVE PROTECTION BASED ON SIMULTANEITY AND TIME TO
C     IMPACT

            IAMON=0
            TRNG=RNG*1000.

      IF (ISIM.LT.ISIMAX) THEN

C     FIND TIME OF FLIGHT TO SEE IF APS COULD REACT BEFORE HIT
            IF (KSIDE.EQ.1) THEN
               IST=(KSYS-1)*3+KWP
               NRG=NRWB(KSYS,KWP)
               ISRM=(KSYS-1)*18+(KWP-1)*6+NRG
               ISR1=(KSYS-1)*18+(KWP-1)*6+1
               RMIN=RB(ISR1)
               RMAX=RB(ISRM)
                TMIN=TEMINB(IST)
                TMAX=TEMAXB(IST)
               CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,ETIME)
                THIT=ETIME-TMIN
               ELSE
                IST=(KSYS-1)*3+KWP
                NRG=NRWR(KSYS,KWP)
                ISRM=(KSYS-1)*18+(KWP-1)*6+NRG
                ISR1=(KSYS-1)*18+(KWP-1)*6+1
                RMIN=RR(ISR1)
                RMAX=RR(ISRM)
                 TMIN=TEMINR(IST)
                 TMAX=TEMAXR(IST)
                CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,ETIME)
                 THIT=ETIME-TMIN
               ENDIF

               IF (KSIDE.EQ.1) THEN
                JSYS=IRSYS(J,1)
                FAPS=RAPSF(KSYS,KWP)
                IF (IRSAPS(JSYS).EQ.1) THEN
                 TMIN=RAPST
                 IF (THIT.GE.TMIN.AND.FAPS.GT.0.0) THEN
                  ISIMR(J)=ISIMR(J)+1
                  IAMON=1
                 ENDIF
                ENDIF
               ELSE
                JSYS=IBSYS(J,1)
                FAPS=BAPSF(KSYS,KWP)
                IF (IBSAPS(JSYS).EQ.1) THEN
                 TMIN=BAPST
                 IF (THIT.GE.TMIN.AND.FAPS.GT.0.0) THEN
                  ISIMB(J)=ISIMB(J)+1
                  IAMON=1
                 ENDIF
                ENDIF
               ENDIF
                ENDIF

C     THE ORDER TO THE FOLLOWING LOGIC IS CRITICAL TO SELECTING
C     THE CORRECT FLAG

               IF (NSHOTS.EQ.0) IFAPS=2
            IF (IAMON.EQ.0)  IFAPS=3
               IF (IHVAPS.EQ.0) IFAPS=-1
               IF (FAPS.LE.0)   IFAPS=-1

C                   ACTIVE PROTECTION FLAG (1 INTERCEPT, 0 MISS, -1
C                   N/A THIS WEAPON, 2 OUT OF SHOTS, 3 NO TIME)

             IF (IHVAPS.EQ.1.AND.IAMON.EQ.1.AND.NSHOTS.GE.1.AND.LIVE.EQ.1) THEN
C     APS CAN RESPOND
              PROB=RND()
C      WRITE (10,*) 'FAPS    PROB'
C      WRITE (10,*) FAPS,PROB
              IF (FAPS.GT.PROB) THEN
               PK=0.0
               IFAPS=1
               IF (KSIDE.EQ.1) THEN
                   WRITE (10,*) ' RED ',J,' FIRES APS -- INTERCEPT !'
               ELSE
                   WRITE (10,*) ' BLUE ',J,' FIRES APS -- INTERCEPT !'
               ENDIF
              ELSE
               IFAPS=0
               IF (KSIDE.EQ.1) THEN
                   WRITE (10,*) ' RED ',J,' FIRES APS -- MISS ....'
               ELSE
                   WRITE (10,*) ' BLUE ',J,' FIRES APS -- MISS ....'
               ENDIF
              ENDIF
               IF (KSIDE.EQ.1) THEN
                IAPSONR(J,2)=IAPSONR(J,2)-1
               ELSE
                IAPSONB(J,2)=IAPSONB(J,2)-1
               ENDIF
              ENDIF
              RETURN
              END


      SUBROUTINE OUTPUT1 (KTIME,KSIDE,K,KSYS,KWP,IENERGY,RANGE,J,JSYS,
     &                    IFAPS,KILL)

C     WRITE OUTPUT DATA FOR POST PROCESSING ON SHOTS TAKEN DURING
BATTLE

C     OUTPUT 1:
C
C     KTIME         KILL TIME
C     KSIDE         SHOOTER SIDE (1 BLUE, 2 RED)
C     K           SHOOTER NUMBER
C     KSYS          SHOOTER SYSTEM
C     KWP           SHOOTER WEAPON
C     IENERGY       BLUE EM GUN ENERGY % SHOT USED
C     RANGE         TARGET RANGE
C     J           TARGET NUMBER
C     JSYS          TARGET SYSTEM
C     AZT           ATTACK AZIMUTH ON TARGET TURRET
C     AZH           ATTACK AZIMUTH ON TARGET HULL
C     IFAPS         ACTIVE PROTECTION FLAG (1 INTERCEPT, 0 MISS, -1
C                   N/A THIS WEAPON, 2 OUT OF SHOTS, 3 NO TIME)
C     KILL          TARGET KILLED FLAG (1 - KILLED, 0 - MISSED)
C     AZOUT         SHOT AZIMUTH RELATIVE TO SHOOTER HULL FRONT
C
```

```
      COMMON/FIRE/IBPRIME(200,2),IBSECOND(200,2),IRPRIME(200,2),
     & IRSECOND(200,2),WRNGB(200,2),WRNGR(200,2)
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
           COMMON/TURRET/AZTB(200),AZTR(200),TRATEB(10),TRATER(10)

      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

C     FIND ATTACK AZIMUTH ON TARGET TURRENT AND HULL
C
C     HULL IS ASSUMED TO BE POINTING IN THE DIRECTION OF TRAVEL
C     TURRENT IS ASSUMED TO BE POINTING TOWARDS AN INTENDED
C     TARGET OR IF THIS TARGET VEHICLE IS NOT ENGAGING, THEN TURRET
C     IS STILL POINTING TOWARDS LAST TARGET.
C     IF TARGET VEHICLE IS IN SURVEILLANCE MODE, TURRET IS
C     INLINE WITH THE HULL
C
      IF (KSIDE.EQ.1) THEN
           AZMT=IRSA(J,2)
           AZS=AZTB(K)+180.0
           AZTT=AZTR(J)
      ELSE
           AZMT=IBSA(J,2)
           AZS=AZTR(K)+180.0
           AZTT=AZTB(J)
          ENDIF

      AZH=AZS-AZMT
           AZT=AZS-AZTT

           IF (AZH.GE.360.0) AZH=AZH-360.0
           IF (AZH.LT.0.0) AZH=AZH+360.0
           IF (AZT.GE.360.0) AZT=AZT-360.0
           IF (AZT.LT.0.0) AZT=AZT+360.0

      IAZH=NINT(AZH)
           IAZT=NINT(AZT)
           IF (IAZH.GE.360) IAZH=IAZH-360
           IF (IAZT.GE.360) IAZT=IAZT-360
           IRNG=NINT(RANGE*1000.)

C     FIND SHOOTER AZIMUTH TO TARGET RELATIVE TO SHOOTER HULL FRONT

           IF (KSIDE.EQ.1) THEN
            AZHS=IBSA(K,2)
            AZTS=AZTB(K)
           ELSE
            AZHS=IRSA(K,2)
            AZTS=AZTR(K)
           ENDIF

           IF (AZTS.LT.AZHS) AZTS=AZTS+360.0
           AZOUT=AZTS-AZHS

           IF (AZOUT.GE.360.0) AZOUT=AZOUT-360.0
           IF (AZOUT.LT.0.0) AZOUT=AZOUT+360.0

           IAZOUT=NINT(AZOUT)
           IF (IAZOUT.GE.360) IAZOUT=IAZOUT-360

1     FORMAT (14I6)

      WRITE (11,1) KTIME,KSIDE,K,KSYS,KWP,IENERGY,IRNG,J,JSYS,
     &           IFAPS,KILL,IAZH,IAZT,IAZOUT

      RETURN
      END


           SUBROUTINE OUTPUT2 (KTIME,JSIDE,J,KILL)

C     WRITE OUTPUT DATA FOR POST PROCESSING ON SYSTEM STATUS WHEN
C     KILLED OR AT END OF SIMULATION

C     OUTPUT 2:
C
C     KTIME          KILL TIME
C     JSIDE        SYSTEM SIDE (1 BLUE, 2 RED)
C     J           VEHICLE NUMBER
C     JSYS          SYSTEM NUMBER
C     JSPD          SPEED
C     NWP           NUMBER OF WEAPONS
C     JAM1-3        REMAINING AMMUNITION EACH WEAPON
C     JENERGY         BLUE EM GUN ENERGY (MJ) REMAINING
C     MODEF         SYSTEM FIRE MODE
C     MODEM         SYSTEM MOVEMENT MODE
C     JDEF          DEFILADE (1) OR EXPOSED (0)
C     KILL         TARGET KILLED FLAG (1 - ALIVE, 0 - KILLED)
C
           COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
           COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
      COMMON/DEFILADE/IDEFB(200),IDEFR(200)
      COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
      COMMON/EMENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
     & ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2)
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)

      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*9 BFNAME,RFNAME

      JAM1=0
      JAM2=0
      JAM3=0

      IF (JSIDE.EQ.1) THEN
       JSYS=IBSYS(J,1)
       JSPD=IBSA(J,1)
       NWP=NWB(JSYS)
       JAM1=NSHOTB(J,1)
       IF (NWP.GE.2) JAM2=NSHOTB(J,2)
       IF (NWP.EQ.3) JAM3=NSHOTB(J,3)
       JENERGY=NINT(ELSYS(J))
       MODEF=MODEFB(J)
       MODEM=MODEMB(J)
       JDEF=IDEFB(J)
      ELSE
       JSYS=IRSYS(J,1)
       JSPD=IRSA(J,1)
       NWP=NWR(JSYS)
       JAM1=NSHOTR(J,1)
       IF (NWP.GE.2) JAM2=NSHOTR(J,2)
       IF (NWP.EQ.3) JAM3=NSHOTR(J,3)
       JENERGY=0
       MODEF=MODEFR(J)
       MODEM=MODEMR(J)
       JDEF=IDEFR(J)
      ENDIF

1     FORMAT (14I4)

      IF (JSYS.GT.10) GO TO 10
      WRITE (12,1) KTIME,JSIDE,J,JSYS,JSPD,NWP,JAM1,JAM2,JAM3,JENERGY,
     &           MODEF,MODEM,JDEF,KILL
10    CONTINUE
      RETURN
      END


      SUBROUTINE WEND
C
C     WRITES FINAL STATUS OF VEHICLES WHICH SURVIVED
C
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)

      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

      DO 100 J=1,NB
      IF (IBSYS(J,2).EQ.1) CALL OUTPUT2 (ITIME,1,J,1)
100   CONTINUE

      DO 200 J=1,NR
      IF (IRSYS(J,2).EQ.1) CALL OUTPUT2 (ITIME,2,J,1)
200   CONTINUE

      RETURN
      END


      SUBROUTINE TURAZ (KSIDE,K,J,AZIMUTH)

C     SETS TURRET AZIMUTH TO CURRENT TARGET AZIMUTH

      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM

      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

      PI=3.14159

      IF (KSIDE.EQ.1) THEN
       XS=BXYA(K,1)
       YS=BXYA(K,2)
```

```
            XT=RXYA(J,1)
            YT=RXYA(J,2)
           ELSE
            XS=RXYA(K,1)
            YS=RXYA(K,2)
            XT=BXYA(J,1)
            YT=BXYA(J,2)
           ENDIF

           TOP=YT-YS
           BOT=XT-XS
            IF (BOT.EQ.0.0) BOT=BOT+.001
            DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
            IF (BOT.LT.0.0) THEN
                IF (TOP.LT.0.0) DIR=DIR+180.
                IF (TOP.GT.0.0) DIR=180.-DIR
            ELSE
                IF (TOP.LT.0.0) DIR=360.-DIR
            ENDIF
           CALL FAZ (DIR,AZIMUTH)

      RETURN
      END



           SUBROUTINE WMODES
C
C     WRITES MODE TIMES FOR EACH SIXTY CYCLE PERIOD ONLY
C     FOR BLUE SYSTEM #1
C
           COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
           COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
           COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
           COMMON/EMENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
     & ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2)
           COMMON/ESMOKE/TESMK(200)
C
           CHARACTER*16 BHIER,RHIER,IHIER
           CHARACTER*24 BUNIT,RUNIT
           CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C
1     FORMAT (5I6)
           IDELT=60*IDTIME
           DO 100 K=1,NB
           IF (IBSYS(K,2).EQ.1) THEN
            KSYS=IBSYS(K,1)
            IF (KSYS.EQ.1) THEN
             DO 50 MM=1,3
50      WRITE (13,1) ITIME,IDELT,K,MM,NINT(TCMMB(K,MM))
             DO 60 MEM=1,2
60      WRITE (13,1) ITIME,IDELT,K,MEM,NINT(TEM(K,MEM))
             DO 70 MF=1,9
70      WRITE (13,1) ITIME,IDELT,K,MF,NINT(TCFMB(K,MF))
C     ENGINE SMOKE DUTY CYCLE TIME
           WRITE (13,1) ITIME,IDELT,K,0,NINT(TESMK(K))
            ENDIF
           ENDIF
100   CONTINUE
           RETURN
           END


           SUBROUTINE WOTHERS (BACEL)
C
C     WRITES OTHER OUTPUTS FOR BLUE EM VEHICLE #1 EACH TIME
C     STEP ... VEHICLE SPEED, ACCELERATION, EM ENERGY
C
C     FOR BLUE SYSTEM #1
C
           COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
           COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMR(200),
     & IAMB(200)
           COMMON/EMENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
     & ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2)
           COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
           COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
           COMMON/ACEL/ACELB(10,5),ACELR(10,5),GRADEB(200),GRADER(200)


C
           DIMENSION BACEL(200)
C
           CHARACTER*16 BHIER,RHIER,IHIER
           CHARACTER*24 BUNIT,RUNIT
           CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C
1     FORMAT (8I6)
           DO 100 K=1,NB
           IF (IBSYS(K,2).EQ.1) THEN
            KSYS=IBSYS(K,1)
            IF (KSYS.EQ.1)
     &WRITE (14,1) ITIME,IDTIME,K,ISPDB(K),NINT(BACEL(K)),NINT(ELSYS(K))
     &           ,IBSA(K,2),NINT(GRADEB(K))
            ENDIF
100   CONTINUE
      RETURN
      END



      SUBROUTINE READCOOK

C     READS VEHICLE BURNING PROBABILITY TABLE AND INITIALIZES ARRAY

      COMMON/COOKERS/NCOOK,XYCOOK(400,2),PBURNB(10),PBURNR(10),
     & KBURNB(200),KBURNR(200)

      CHARACTER*90 LINE

      NCOOK=0
      DO 10 I=1,400
      XYCOOK(I,1)=-1.
10      XYCOOK(I,2)=-1.
      DO 11 I=1,200
      KBURNB(I)=0
11      KBURNR(I)=0

303     FORMAT (A90)
503     FORMAT (1X,A90)
      OPEN (1,FILE='COOKERS.IN')
      READ (1,303) LINE
C       WRITE (*,503) LINE
      READ (1,303) LINE
C       WRITE (*,503) LINE
      READ (1,303) LINE
C       WRITE (*,503) LINE
      READ (1,303) LINE
C       WRITE (*,503) LINE
313     FORMAT (10F8.2)
314     FORMAT (1X,10F8.2)
      READ (1,313) (PBURNB(K),K=1,10)
C       WRITE (*,314) (PBURNB(K),K=1,10)
      READ (1,303) LINE
C       WRITE (*,503) LINE
      READ (1,313) (PBURNR(K),K=1,10)
C       WRITE (*,314) (PBURNR(K),K=1,10)

      CLOSE (1)
      RETURN
      END


      SUBROUTINE COOKS (KSIDE,NUM)

      COMMON/COOKERS/NCOOK,XYCOOK(400,2),PBURNB(10),PBURNR(10),
     & KBURNB(200),KBURNR(200)
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
C
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

      PB=RND()

      IF (KSIDE.EQ.1) THEN
       NSYS=IBSYS(NUM,1)
       PBURN=PBURNB(NSYS)
       X=BXYA(NUM,1)
       Y=BXYA(NUM,2)
       IF (PBURN.GE.PB) THEN
        NCOOK=NCOOK+1
        XYCOOK(NCOOK,1)=X
        XYCOOK(NCOOK,2)=Y
        KBURNB(NUM)=1
       ENDIF

      ELSE
       NSYS=IRSYS(NUM,1)
       PBURN=PBURNR(NSYS)
       X=RXYA(NUM,1)
       Y=RXYA(NUM,2)
       IF (PBURN.GE.PB) THEN
        NCOOK=NCOOK+1
        XYCOOK(NCOOK,1)=X
        XYCOOK(NCOOK,2)=Y
        KBURNR(NUM)=1
       ENDIF

      ENDIF

      RETURN
      END


      SUBROUTINE COOKMASK (XS,YS,RMAX1,RMAX2,RMAX3)
C
C     FIND ANGLE AND RANGE TO ALL COOKERS WITHIN MAX RANGE OF
C     THIS SYSTEM SO THAT MASKING CAN BE EVALUATED
```

```
C
      COMMON/MASKCOOK/COOKMSK(400,3),NCOOKMSK
      COMMON/COOKERS/NCOOK,XYCOOK(400,2),PBURNB(10),PBURNR(10),
     & KBURNB(200),KBURNR(200)

C     XS,YS      -- SHOOTER COORDINATES
C     RMAX1,2,3  -- MAX RANGE OF SYSTEM WEAPONS
C     COOKMSK(I,1) -- X COORD OF EACH COOKER
C     COOKMSK(I,2) -- Y COORD OF EACH COOKER
C     COOKMSK(I,3) -- RADIUS FOR EACH COOKER PUFF
C     NCOOKMSK   -- NUMBER OF COOKERS WITHIN MAX RANGE
C
      PI=3.14159
      NCOOKMSK=0
      IF (NCOOK.GT.0) THEN
C     CYCLE THROUGH ALL COOKERS TO FILL MASKING ARRAY
      RMAXX=0
      IF (RMAX1.GT.RMAXX) RMAXX=RMAX1
      IF (RMAX2.GT.RMAXX) RMAXX=RMAX2
      IF (RMAX3.GT.RMAXX) RMAXX=RMAX2
      RMAXX=RMAXX/1000.0
      DO 20 I=1,NCOOK
        XD=XYCOOK(I,1)
        YD=XYCOOK(I,2)
       RADIUS=0.030
      RANGE=((XD-XS)**2+(YD-YS)**2)**.5
      IF ((RANGE-RADIUS).GT.RMAXX) GO TO 20
      NCOOKMSK=NCOOKMSK+1
      COOKMSK(NCOOKMSK,1)=XD
      COOKMSK(NCOOKMSK,2)=YD
      COOKMSK(NCOOKMSK,3)=RADIUS
20    CONTINUE
      ENDIF
      RETURN
      END


      SUBROUTINE COOKSEE (XS,YS,XT,YT,AZIMUTH,JSKIP)
C     EVALUATES DUST MASKING BETWEEN SHOOTER AND TARGET
C
      COMMON/MASKCOOK/COOKMSK(400,3),NCOOKMSK
      COMMON/COOKERS/NCOOK,XYCOOK(400,2),PBURNB(10),PBURNR(10),
     & KBURNB(200),KBURNR(200)

C     COOKMSK(I,1) -- X COORD OF EACH COOKER
C     COOKMSK(I,2) -- Y COORD OF EACH COOKER
C     COOKMSK(I,3) -- RADIUS FOR EACH COOKER PUFF
C     NCOOKMSK   -- NUMBER OF COOKERS WITHIN MAX RANGE
C
C     XS,YS  SHOOTER REAL WORLD COORDINATES
C     XT,YT  TARGET  REAL WORLD COORDINATES
C     AZIN   AZIMUTH TO TARGET
C     JSKIP  FLAG TO SKIP THIS TARGET IF MASKED (=0 TO SKIP)

      COMMON/ELEVA/CONT(20),IELEV(10000)
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT

      DOUBLE PRECISION ANGLE,ANGLEI

      PI=3.14159

      IF (NCOOKMSK.GT.0) THEN
C
C     FIND SHOOTER AND TARGET GRID COORDINATES
C     TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
C     CENTER TO CENTER ABOUT THE SCALE.
      RRANGE=((YT-YS)**2+(XT-XS)**2)**.5
      IXS=NINT(XS/SCALE)
      IYS=NINT(YS/SCALE)
      IXT=NINT(XT/SCALE)
      IYT=NINT(YT/SCALE)
      IF (IXS.LT.1) IXS=1
      IF (IXT.LT.1) IXT=1
      IF (IYS.LT.1) IYS=1
      IF (IYT.LT.1) IYT=1
      IF (IXS.GT.NX) IXS=NX
      IF (IXT.GT.NX) IXT=NX
      IF (IYS.GT.NY) IYS=NY
      IF (IYT.GT.NY) IYT=NY
C     X1=IXS*SCALE
C     X2=IXT*SCALE
C     Y1=IYS*SCALE
C     Y2=IYT*SCALE
C     RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5
C
C     FIND SHOOTER AND TARGET ELEVATION
      IS=NY*(IXS-1)+IYS
      SELEV=CONT(IELEV(IS))
      IT=NY*(IXT-1)+IYT
      TELEV=CONT(IELEV(IT))
C
C     ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
C     FOR VEHICLE HEIGHT
      SELEV=SELEV+.002
      TELEV=TELEV+.002
C
C     FIND ELEVATION ANGLE TO THIS TARGET
      TOP=TELEV-SELEV
      BOT=RRANGE
      ANGLE=ATAN(TOP/BOT)
C
      DO 200 I=1,NCOOKMSK

      IF (JSKIP.EQ.0) GO TO 300
C
C     CYCLE THROUGH ALL COOKERS
C     TO EVALUATE MASKING, FIND THE LEFT AND RIGHT AZIMUTH
C     BRACKETING EACH PUFF WITHIN RANGE.  DROP IF OUT OF ANGLE RANGE.
C     THEN FIND THE ELEVATION ANGLE TO THE TOP OF THE PUFF.
C     DROP PUFF IF NOT HIGH ENOUGH
C
C     EVALUATE RANGE TO COOKER PUFF
      XP=COOKMSK(I,1)
      YP=COOKMSK(I,2)
C     WRITE (10,*) 'XP YP ', XP,YP
      RADIUS=COOKMSK(I,3)
C     WRITE (10,*) 'RADIUS ',RADIUS
      RP=((YP-YS)**2+(XP-XS)**2)**.5
C     WRITE (10,*) 'PUFF RANGE ',RP,' TARGET RANGE ',RRANGE
      IF ((RP-RADIUS).GT.RRANGE) GO TO 200
C
C     EVALUATE ANGLES TO COOKER PUFF
      TOP=YP-YS
      BOT=XP-XS
      IF (TOP.EQ.0.0.AND.BOT.EQ.0.0) JSKIP=0
      IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
      IF (BOT.LT.0.0) THEN
                IF (TOP.LT.0.0) DIR=DIR+180.
                IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
                IF (TOP.LT.0.0) DIR=360.-DIR
      ENDIF
C     CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
      CALL FAZ (DIR,AZIN)
      IF (RADIUS.LT.0.030) RADIUS=0.030
      IF (RP.LT.RADIUS) RP=RADIUS
      DELAZ=ASIN(RADIUS/RP)*180./PI
      AZLFT=AZIN-DELAZ
      AZRGT=AZIN+DELAZ
      IF (AZLFT.LT.0.0) AZLFT=AZLFT+360.0
      IF (AZRGT.LT.0.0) AZRGT=AZRGT+360.0
      IF (AZLFT.GT.360.0) AZLFT=AZLFT-360.0
      IF (AZRGT.GT.360.0) AZRGT=AZRGT-360.0
      IF (AZLFT.GT.AZRGT) AZRGT=AZRGT+360.
      IF (AZIMUTH.LT.0.0) AZIMUTH=AZIMUTH+360.0
      IF (AZIMUTH.GT.360.0) AZIMUTH=AZIMUTH-360.0
C     WRITE (10,*)'AZLEFT',AZLFT,' AZRIGHT ',AZRGT,' AZIMUTH ',AZIMUTH
      IF (AZIMUTH.LT.AZLFT.OR.AZIMUTH.GT.AZRGT) GO TO 200
C
C     EVALUATE HEIGHT OF COOKER PUFF
C
C     FIND ELEVATION OF THE GROUND UNDER THIS PUFF
      IXP=NINT(XP/SCALE)
      IYP=NINT(YP/SCALE)
      IF (IXP.LT.1) IXP=1
      IF (IYP.LT.1) IYP=1
      IF (IXP.GT.NX) IXP=NX
      IF (IYP.GT.NY) IYP=NY
C     PX2=IXP*SCALE
C     PY2=IYP*SCALE
C     PRANGE=((PY2-Y1)**2+(PX2-X1)**2)**.5
      IP=NY*(IXP-1)+IYP
      PELEV=CONT(IELEV(IP))
C     ADD TO THIS ELEVATION THE HEIGHT OF THE PUFF
C     MAX PUFF HEIGHT IS 30 METERS (100 FEET)
      HEIGHT=RADIUS
      IF (RADIUS.GT.0.030) HEIGHT=0.030
      PELEV=PELEV+HEIGHT
C     FIND ELEVATION ANGLE TO THIS DUST PUFF
      TOP=PELEV-SELEV
C     BOT=RANGE
      BOT=RP
      ANGLEI=ATAN(TOP/BOT)
C     WRITE (10,*) 'TARGET ELEVATION ANGLE ',ANGLE
C     WRITE (10,*) 'PUFF ELEVATION ANGLE ',ANGLEI
      IF (ANGLEI.GT.ANGLE) JSKIP=0
200   CONTINUE

      ENDIF

300   RETURN
      END


      SUBROUTINE READMINES

C     READS MINEFIELD DATA

      COMMON/MINES/NMINES,XYMINE(100,2),WLMINE(100,2),NAZMINE(100,3)
      COMMON/MINEKILLS/MINEK(200,2)
      COMMON/MINEBREECH/MBREECH(200,2),MINEON(100)

      CHARACTER*80 LINE

C     NMINES  -- NUMBER OF MINEFIELDS
C     XYMINE  -- X AND Y COORDINATES OF CENTER
C     WLMINE  -- WIDTH AND LENGTH OF MINEFIELD
C     NAZMINE -- NUMBER OF MINES IN FIELD, AZIMUTH OF LENGTH,
C                SURFACE (1) OR BURRIED (0)

      DO 10 I=1,100
      MINEON(I)=1
      WLMINE(I,1)=0.
      WLMINE(I,2)=0.
```

```
            NAZMINE(I,1)=0
            NAZMINE(I,2)=0
            NAZMINE(I,3)=0
            XYMINE(I,1)=-1.
10      XYMINE(I,2)=-1.

303     FORMAT (A80)
503     FORMAT (1X,A80)
            OPEN (1,FILE='MINES.IN')
            READ (1,303) LINE
C       WRITE (*,503) LINE
            READ (1,303) LINE
C       WRITE (*,503) LINE
            READ (1,303) LINE
C       WRITE (*,503) LINE
311     FORMAT (I8)
312     FORMAT (1X,I8)
            READ (1,311) NMINES
            IF (NMINES.GT.100) NMINES=100
            IF (NMINES.LT.0) NMINES=0
C       WRITE (*,312) NMINES
313     FORMAT (8I8)
314     FORMAT (1X,8I8)
            READ (1,303) LINE
C       WRITE (*,503) LINE
            READ (1,303) LINE
C       WRITE (*,503) LINE
            READ (1,303) LINE
C       WRITE (*,503) LINE
            DO 320 I=1,NMINES

            READ (1,313) N,IX,IY,L,IW,IAZ,NUM,LAID
C       WRITE (*,314) N,IX,IY,L,IW,IAZ,NUM,LAID
            XYMINE(I,1)=REAL(IX)/1000.
            XYMINE(I,2)=REAL(IY)/1000.
            WLMINE(I,1)=REAL(IW)/1000.
            WLMINE(I,2)=REAL(L)/1000.
            NAZMINE(I,1)=NUM
            NAZMINE(I,2)=IAZ
            NAZMINE(I,3)=LAID

320     CONTINUE

            CLOSE (1)

            DO 400 I=1,200
            MINEK(I,1)=0
            MINEK(I,2)=0
            MBREECH(I,1)=0
            MBREECH(I,2)=0
400     CONTINUE

            RETURN
            END


            SUBROUTINE PLOTMINES
C
C       PLOTS MINEFIELD LOCATIONS ON MAP
C       FILLS THE AREA IF THE MINEFIELD IS SURFACE LAID OR BURRIED
C       BUT "DISCOVERED" BY A VEHICLE GETTING BLASTED
C       IF UNDISCOVERED AND BURRIED, THE AREA IS NOT FILLED IN
C
            COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
            COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT

            COMMON/MINES/NMINES,XYMINE(100,2),WLMINE(100,2),NAZMINE(100,3)
            COMMON/MINEBREECH/MBREECH(200,2),MINEON(100)

            DIMENSION XARRAY(4),YARRAY(4)
C
C
            PI=3.14159
C
C       DRAWS AND FILLS A SQUARE FOR SURFACE OR KNOWN MINEFIELD
C
            CALL FACTOR (ZFACT)

            GXUNIT=11.0/NX*SCFACT
            GYUNIT=8.5/NY
            XUNIT=GXUNIT/SCALE
            YUNIT=GYUNIT/SCALE

            DO 200 I=1,NMINES
C       IF (NAZMINE(I,3).EQ.0) GO TO 200
            IF (MINEON(I).EQ.0) GO TO 200
            XC=XYMINE(I,1)
            YC=XYMINE(I,2)
            WIDTH=WLMINE(I,1)
            WLONG=WLMINE(I,2)
            AZ=REAL(NAZMINE(I,2))
            CALL FDIR (AZ,DIR)
            DIR=DIR/180.*PI
            X1=XC+WLONG/2.*COS(DIR)-WIDTH/2.*SIN(DIR)
            Y1=YC+WLONG/2.*SIN(DIR)+WIDTH/2.*COS(DIR)
            X2=X1+WIDTH*SIN(DIR)
            Y2=Y1-WIDTH*COS(DIR)
            X3=X2-WLONG*COS(DIR)
            Y3=Y2-WLONG*SIN(DIR)
            X4=X3-WIDTH*SIN(DIR)
            Y4=Y3+WIDTH*COS(DIR)

            IF (XLEN.GT.0.) CALL FACTOR (ZFACTN)

            ISKIP=0

            CALL NEWPEN (13)

            XARRAY(1)=X1*XUNIT
            XARRAY(2)=X2*XUNIT
            XARRAY(3)=X3*XUNIT
            XARRAY(4)=X4*XUNIT
            YARRAY(1)=Y1*YUNIT
            YARRAY(2)=Y2*YUNIT
            YARRAY(3)=Y3*YUNIT
            YARRAY(4)=Y4*YUNIT
            Y10=.95*YMAX
            X10=.95*XMAX
            X11=.05*XMAX
            Y11=.05*YMAX

            DO 100 J=1,4

            IF (XARRAY(J).LE.X11) XARRAY(J)=X11
            IF (YARRAY(J).LE.Y11) YARRAY(J)=Y11
            IF (XLEN.GT.0.) THEN
                    CALL ZOOMIT (XARRAY(J),YARRAY(J),ISKIP)
                    IF (ISKIP.EQ.1) THEN
                     IF (YARRAY(J).GT.Y10) YARRAY(J)=Y10
                     IF (XARRAY(J).GT.X10) XARRAY(J)=X10
                     ISKIP=0
                    ENDIF
            ENDIF
100     CONTINUE

            IF (XLEN.GT.0.) THEN

            IF (XARRAY(1).LE.X11.AND.XARRAY(2).LE.X11.AND.XARRAY(3).LE.X11.
           &  AND.XARRAY(4).LE.X11) ISKIP=1
            IF (YARRAY(1).LE.Y11.AND.YARRAY(2).LE.Y11.AND.YARRAY(3).LE.Y11.
           &  AND.YARRAY(4).LE.Y11) ISKIP=1
            IF (XARRAY(1).GE.X10.AND.XARRAY(2).GE.X10.AND.XARRAY(3).GE.X10.
           &  AND.XARRAY(4).GE.X10) ISKIP=1
            IF (YARRAY(1).GE.Y10.AND.YARRAY(2).GE.Y10.AND.YARRAY(3).GE.Y10.
           &  AND.YARRAY(4).GE.Y10) ISKIP=1

            ENDIF

            IF (ISKIP.EQ.1) GO TO 200

            IF (NAZMINE(I,3).EQ.1) THEN
C           DRAW AND FILL FOUR SIDED SQUARE
            CALL FILL (4,XARRAY,YARRAY)
            ENDIF
            IF (NAZMINE(I,3).EQ.0) THEN
C           DRAW EMPTY BOX
            CALL PLOT (XARRAY(1),YARRAY(1),3)
            CALL PLOT (XARRAY(2),YARRAY(2),2)
            CALL PLOT (XARRAY(3),YARRAY(3),2)
            CALL PLOT (XARRAY(4),YARRAY(4),2)
            CALL PLOT (XARRAY(1),YARRAY(1),2)
            ENDIF
200     CONTINUE
C       READ (*,*)
            RETURN
            END


            SUBROUTINE MINEFIELD (KSIDE,K,DELPOS,DIR,INNOW,INNEXT)
            COMMON/MINES/NMINES,XYMINE(100,2),WLMINE(100,2),NAZMINE(100,3)
            COMMON/MINEBREECH/MBREECH(200,2),MINEON(100)
            COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
           & L1BM,L2M,L3M,L1RM
            COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
           & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
C
            CHARACTER*16 BHIER,RHIER,IHIER
            CHARACTER*24 BUNIT,RUNIT
            CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

C           KSIDE    UNIT SIDE
C           K        UNIT NUMBER
C           DELPOS   DISTANCE TO BE TRAVELED IN THIS TIME STEP
C           DIR      TRAVEL DIRECTION (CARTESIAN ANGLE)

            PI=3.14159
            INNOW=0
            INNEXT=0

            IF (KSIDE.EQ.1) THEN
              XK=BXYA(K,1)
              YK=BXYA(K,2)
            ELSE
              XK=RXYA(K,1)
              YK=RXYA(K,2)
            ENDIF

            DO 100 I=1,NMINES
            IF (MINEON(I).EQ.0) GO TO 100
            IF (INNOW.NE.0.OR.INNEXT.NE.0) GO TO 200

            XC=XYMINE(I,1)
            YC=XYMINE(I,2)
            WIDTH=WLMINE(I,1)
```

```
            WLONG=WLMINE(I,2)
            AZ=REAL(NAZMINE(I,2))
            CALL FDIR (AZ,OBL)

C     SEE IF THIS UNIT IS CURRENTLY IN THIS MINEFIELD

C     FIND RANGE TO CENTER OF THE MINEFIELD
            RC=((YC-YK)**2+(XC-XK)**2)**.5
C     FIND RANGE ANGLE TO CENTER OF MINEFIELD
            TOP=YC-YK
            BOT=XC-XK
            IF (BOT.EQ.0.0) BOT=BOT+.001
            RDIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
            IF (BOT.LT.0.0) THEN
                        IF (TOP.LT.0.0) RDIR=RDIR+180.
                        IF (TOP.GT.0.0) RDIR=180.-RDIR
            ELSE
                        IF (TOP.LT.0.0) RDIR=360.-RDIR
            ENDIF

C     FIND LINE-OF-SIGHT THICKNESS OF THIS MINEFIELD TOWARDS UNIT
            ALPHA=ABS((OBL-90.)-RDIR)
            IF (ALPHA.EQ.90.0.OR.ALPHA.EQ.270.) THEN
             TLOS=WLONG/2.
            ELSE
             TLOS=(ABS(WIDTH/COS(ALPHA/180.*PI)))/2.
             IF (TLOS.GT.(WLONG/2.)) TLOS=WLONG/2.
            ENDIF

            IF (TLOS.GT.RC) INNOW=I
C     THIS UNIT IS CURRENTLY IN THIS MINEFIELD

C     FIND IF THIS UNIT WILL BE IN THE FIELD IF IT CONTINUES
C     TO MOVE IN CURRENT DIRECTION
            XK=XK+DELPOS*COS(DIR*PI/180.)
            YK=YK+DELPOS*SIN(DIR*PI/180.)

C     FIND RANGE TO CENTER OF THE MINEFIELD
            RC=((YC-YK)**2+(XC-XK)**2)**.5
C     FIND RANGE ANGLE TO CENTER OF MINEFIELD
            TOP=YC-YK
            BOT=XC-XK
            IF (BOT.EQ.0.0) BOT=BOT+.001
            RDIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
            IF (BOT.LT.0.0) THEN
                        IF (TOP.LT.0.0) RDIR=RDIR+180.
                        IF (TOP.GT.0.0) RDIR=180.-RDIR
            ELSE
                        IF (TOP.LT.0.0) RDIR=360.-RDIR
            ENDIF

C     FIND LINE-OF-SIGHT THICKNESS OF THIS MINEFIELD TOWARDS UNIT
            ALPHA=ABS((OBL-90.)-RDIR)
            IF (ALPHA.EQ.90.0.OR.ALPHA.EQ.270.) THEN
             TLOS=WLONG/2.
            ELSE
             TLOS=(ABS(WIDTH/COS(ALPHA/180.*PI)))/2.
             IF (TLOS.GT.(WLONG/2.)) TLOS=WLONG/2.
            ENDIF

            IF (TLOS.GT.RC) INNEXT=I
C     THIS UNIT WILL REMAIN IN OR ENTER THIS MINEFIELD

100   CONTINUE

200   CONTINUE

            RETURN
            END



            SUBROUTINE MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)

            COMMON/MINES/NMINES,XYMINE(100,2),WLMINE(100,2),NAZMINE(100,3)
            COMMON/MINEKILLS/MINEK(200,2)

            IF (INNOW.NE.0) THEN
             AFIELD=WLMINE(INNOW,1)*WLMINE(INNOW,2)
             DENS=REAL(NAZMINE(INNOW,1))/AFIELD
             IFOUND=INNOW
            ELSE
             IF (INNEXT.NE.0) THEN
              AFIELD=WLMINE(INNEXT,1)*WLMINE(INNEXT,2)
              DENS=REAL(NAZMINE(INNEXT,1))/AFIELD
              IFOUND=INNEXT
             ENDIF
      ENDIF

      IF (IFOUND.NE.0) THEN
      AREA=DELPOS*.005
      PMU=DENS*AREA
      P0=EXP(-PMU)
      P1=1.0-P0
      PLIVE=RND()
       IF (P1.GT.PLIVE) MINEK(K,KSIDE)=IFOUND
C       BOOM .....
      ENDIF

      RETURN
      END
```

```
      SUBROUTINE WMINE (KTIME,JSIDE,J,NFIELD)

C     MINEFIELD KILLS DATA OUTPUT
C
C      KTIME          KILL TIME
C      JSIDE        SYSTEM SIDE (1 BLUE, 2 RED)
C      J           VEHICLE NUMBER
C      JSYS          SYSTEM NUMBER
C      NFIELD        MINEFIELD NUMBER

      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT

      IF (JSIDE.EQ.1) THEN
       JSYS=IBSYS(J,1)
      ELSE
       JSYS=IRSYS(J,1)
      ENDIF

1     FORMAT (5(I4,3X))

      WRITE (15,1) KTIME,JSIDE,J,JSYS,NFIELD

      RETURN
      END



      SUBROUTINE BREECH
C
C      PLACES VEHICLES IN MINEFIELD BREECH MODE WHICH FIXES
C      THEIR SPEED TO 4 KM/HR, BUT MAKES THEM INVULNERABLE TO MINES.
C      THIS ROUTINE ALSO ALLOWS THE OPERATOR TO TURN MINEFIELDS
C      ON OR OFF TO SIMULATE BREECHED MINEFIELDS OR UNITS MOVING
C      QUICKLY THROUGH SAFE LANES IN KNOWN MINEFIELDS.
C
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
     & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
     & NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
     & TEMINR(30),ACQB(10,5),ACQR(10,5),SRCHB(10),SRCHR(10),
     & MILB(10,3),MILR(10,3)
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
     & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
     & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
     & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMMR(200),TRMFB(200),TRMMR(200),
     & TRMFR(200),TRDESB(200),TRDESR(200)
      COMMON/MINES/NMINES,XYMINE(100,2),WLMINE(100,2),NAZMINE(100,3)
      COMMON/MINEKILLS/MINEK(200,2)
      COMMON/MINEBREECH/MBREECH(200,2),MINEON(100)
C
      CHARACTER*8 BFNAME,RFNAME
C     CHARACTER*90 LINE
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*1 C2
C     CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
2       WRITE (*,*)
      WRITE (*,*) '1 -- TURN MINEFIELDS ON AND OFF'
      WRITE (*,*) '2 -- PLACE VEHICLES IN BREECH MODE'
      WRITE (*,*) '3 -- QUIT'
      WRITE (*,*)
      READ (*,*,ERR=2) IC1
      IF (IC1.EQ.3) GO TO 1000
      IF (IC1.EQ.2) GO TO 10
3       IF (IC1.EQ.1) THEN
      WRITE (*,*) 'FIELD #   ON/OFF    Xcenter   Ycenter'
4       FORMAT (1X,2(I5,4X),2(F8.3,1X))
      DO 5 I=1,NMINES
5       WRITE (*,4) I,MINEON(I),XYMINE(I,1),XYMINE(I,2)
      WRITE (*,*)
      WRITE (*,*) 'ENTER FIELD # (0-QUIT), OFF=0, ON=1'
      WRITE (*,*)
      READ (*,*,ERR=3) I,MINEON(I)
      IF (I.EQ.0) GO TO 2
      GO TO 3
      ENDIF
      GO TO 2
C
C     BREECH MODE
C
10    KSIDE=0
      WRITE (*,*)
      WRITE (*,*) 'enter BLUE or RED, or QUIT'
      WRITE (*,*)
      READ (*,*,ERR=10) C2
```

```
      IF (C2(1:1).EQ.'B') KSIDE=1
      IF (C2(1:1).EQ.'R') KSIDE=2
      IF (C2(1:1).EQ.'O') GO TO 2
      IF (KSIDE.EQ.0) GO TO 10
C
6     WRITE (*,*)
      WRITE (*,*) 'CURRENT BREECH STATUS'
      WRITE (*,*)' UNIT HIERARCHY    IDENTITY          1 = YES
     & 0 = NO  -1 = RECKLESS'
      IF (C2(1:1).EQ.'B') THEN
C     BLUE
      KK=0
      DO 130 K=1,NB
121   FORMAT (1X,I3,1X,A16,A24,7X,I3)
      IF (IBSYS(K,2).GE.1) THEN
      WRITE (*,121) K,BHIER(K),BUNIT(K),MBREECH(K,1)
      KK=KK+1
      ENDIF
      IF (KK.EQ.20) THEN
            READ (*,*)
            KK=0
      ENDIF
130   CONTINUE
C
      ELSE
C     RED
      KK=0
      DO 140 K=1,NR
      IF (IRSYS(K,2).GE.1) THEN
      WRITE (*,121) K,RHIER(K),RUNIT(K),MBREECH(K,2)
      KK=KK+1
      ENDIF
      IF (KK.EQ.20) THEN
            READ (*,*)
            KK=0
      ENDIF
140   CONTINUE
      ENDIF

141   WRITE (*,*) 'ENTER COMMAND UNIT (0-RELISTS -1-QUIT)'
      WRITE (*,*)
      READ (*,*,ERR=141) ICOMMAND
      IF (ICOMMAND.EQ.0) GO TO 6
      IF (ICOMMAND.EQ.-1) GO TO 10

142   WRITE (*,*) 'ENTER NUMBER RANGE OF SUBORDINATES TO FOLLOW'
      WRITE (*,*) 'THIS ORDER (#START #END) (0 0 FOR NONE)'
      WRITE (*,*)
      READ (*,*,ERR=142) NSUB1,NSUB2
      IF (NSUB1.LE.0) THEN
            NSUB1=0
            NSUB2=0
      ENDIF
      IF (NSUB2.LE.0) THEN
            NSUB1=0
            NSUB2=0
      ENDIF
C
150   WRITE (*,*)
      WRITE (*,*) 'TO BREECH MODE THESE VEHICLES ENTER 1, ELSE 0'
      WRITE (*,*) 'RECKLESS BREECH (NO STOPPING OR CLEARING) ENTER -1'
      WRITE (*,*) 'FOR 0 AND -1, ENTER DESIRED SPEED AT MOVEMENT ORDER
     &COMMAND LEVEL'
      WRITE (*,*)
      READ (*,*,ERR=150) JDEF
      IF (JDEF.LT.-1.OR.JDEF.GT.1) GO TO 150

      IF (KSIDE.EQ.1) THEN
      IBSA(ICOMMAND,1)=4
      ISPDB(ICOMMAND)=4
      MBREECH(ICOMMAND,1)=JDEF
      MODEMB(ICOMMAND)=1
      ELSE
      IRSA(ICOMMAND,1)=4
      ISPDR(ICOMMAND)=4
      MBREECH(ICOMMAND,2)=JDEF
      MODEMR(ICOMMAND)=1
      ENDIF
      DO 200 I=NSUB1,NSUB2
      IF (KSIDE.EQ.1) THEN
            IF (IBSYS(I,2).GE.1) THEN
            IBSA(I,1)=4
            ISPDB(I)=4
            MBREECH(I,1)=JDEF
            MODEMB(I)=1
            ENDIF
      ELSE
            IF (IRSYS(I,2).GE.1) THEN
            IRSA(I,1)=4
            ISPDR(I)=4
            MBREECH(I,2)=JDEF
            MODEMR(I)=1
            ENDIF
      ENDIF
200   CONTINUE

      GO TO 141

1000  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PKMOVE (KSIDE,IS,IT,RNG,BORE,TOF,PK)
C
C     REDUCES PK WHEN ENGAGING MOVING TARGETS AND WHEN SHOOTER IS
MOVING
C     FIRE-AND-FORGET WEAPONS ONLY.
C     TWO ERROR SOURCES DOMINATE THIS PROBLEM:
C     1)  THE ACCURACY OF TURRET TRAVERSE RATE SENSORS (ASSUMED
C         1% ERROR);
C     2)  THE MOVEMENT OF THE TARGET UP OR DOWN A GRADE.
C     THE FIRST CONDITION RESULTS IN IMPROPER LEAD ANGLE.
C     THE SECOND RESULTS IN WRONG SUPER-ELEVATION OF THE GUN.
C     ALL OTHER MOVEMENTS OF THE SHOOTER AND TARGET
C     ARE COMPENSATED WELL ENOUGH TO ASSUME NO ERROR.
C
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/ACEL/ACELB(10,5),ACELR(10,5),GRADEB(200),GRADER(200)
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)

      DIMENSION SIG(40),AREA(40)
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      DATA SIG /0.,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.,1.1,1.2,1.3,1.4,1.5,
     & 1.6,1.7,1.8,1.9,2.,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3.,3.1,
     & 3.2,3.3,3.4,3.5,3.6,3.7,3.8,3.9/
      DATA AREA /.5,.5398,.5793,.6179,.6554,.6915,.7258,.758,.7881,
     & .8159,.8413,.8643,.8849,.9032,.9192,.9332,.9452,.9554,.9641,
     & .9713,.9772,.9821,.9861,.9893,.9918,.9938,.9953,.9965,.9974,
     & .9981,.9987,.9990,.9993,.9995,.9997,.9998,.9998,.9999,.9999,
     & 1.000/

      PI=3.14159

      RANGE=RNG*1000.
C
      IF (KSIDE.EQ.1) THEN
C     BLUE SHOOTER / RED TARGET
      SS=REAL(ISPDB(IS))/3600.*1000.
      ST=REAL(ISPDR(IT))/3600.*1000.
      CALL FDIR (REAL(IBSA(IS,2)),AZS)
      CALL FDIR (REAL(IRSA(IT,2)),AZT)
      AZS=AZS/180.*PI
      AZT=AZT/180.*PI
      XS=BXYA(IS,1)
      YS=BXYA(IS,2)
      XT=RXYA(IT,1)
      YT=RXYA(IT,2)
      GRADET=GRADER(IT)
      ELSE
      SS=REAL(ISPDR(IS))/3600.*1000.
      ST=REAL(ISPDB(IT))/3600.*1000.
      CALL FDIR (REAL(IRSA(IS,2)),AZS)
      CALL FDIR (REAL(IBSA(IT,2)),AZT)
      AZS=AZS/180.*PI
      AZT=AZT/180.*PI
      XS=RXYA(IS,1)
      YS=RXYA(IS,2)
      XT=BXYA(IT,1)
      YT=BXYA(IT,2)
      GRADET=GRADEB(IT)
      ENDIF

C     DEFINE UNIT VECTOR CONNECTING SHOOTER AND TARGET
      RMAG=((XS-XT)**2+(YS-YT)**2)**.5
      DX=XS-XT
      DY=YS-YT
      UI=DX/RMAG
      UJ=DY/RMAG
C
C     FIND RELATIVE CROSSING SPEED OF SHOOTER AND TARGET
C     BY TAKING CROSS PRODUCT OF EACH VEHICLE VELOCITY VECTOR
C     ONTO THIS UNIT VECTOR.  THEN TAKE DIFFERENCE IN CROSS
C     PRODUCTS.
C
      AS=SS*COS(AZS)
      BS=SS*SIN(AZS)
      AT=ST*COS(AZT)
      BT=ST*SIN(AZT)

C     THE CROSS PRODUCT OF THESE VECTOR QUANTITIES IS THE CROSSING
C     SPEED

      SCROSS=ABS((AS*UJ-BS*UI)-(AT*UJ-BT*UI))

C     THIS CROSSING SPEED WILL HAVE A 1% ERROR TO GIVE DEFLECTION
C     AIMPOINT ERROR WHEN BULLET ARRIVES AT TARGET RANGE.

C     FIND TIME TO TARGET RANGE
      TT=RANGE/BORE*TOF
C
C     FIND HORIZONTAL DISTANCE ERROR TO TARGET (METERS)
      DHT=TT*SCROSS*.01

C     THIS BECOMES THE AIMPOINT OFFSET DISTANCE (METERS)
      DIST=DHT

C     ASSUME WEAPON PK/Hit=1.0 AND THEREFORE HIT PROBABILITY IS
C     EQUAL TO PK/S AS ENTERED ABOVE
C     FIND DEFLECTION AND RANGE ERROR STANDARD DEVIATIONS
      DAREA=PK**.5
C     RAREA=DAREA
```

```
      DO 10 I=1,39
      PROB1=(AREA(I)-0.5)**2.0
      PROB2=(AREA(I+1)-0.5)**2.0
      IF (DAREA.LE.PROB2.AND.DAREA.GE.PROB1) GO TO 20
10    CONTINUE
20    CONTINUE
      BOT=PROB2-PROB1
      TOP=SIG(I+1)-SIG(I)
      DSIG=SIG(I)+(DAREA-PROB1)*TOP/BOT
      RSIG=DSIG

C     FIND TARGET DEFLECTION LEFT AND RIGHT LIMITS IN TERMS
C     OF STANDARD DEVIATIONS WHEN HORIZONTAL AIMPOINT ERROR IS INCLUDED
C     ASSUME TARGET IS THREE BY THREE METERS SQUARE
C     THEREFORE FIND ADDITIONAL DEFLECTION ERROR IN TERMS OF A TARGET
C     WIDTH STANDARD DEVIATION
      DIST=ABS(DIST/1.5)
      BLFT=-DSIG-DIST
      BRGT=DSIG-DIST
      SLFT=ABS(BLFT)
      SRGT=ABS(BRGT)

      J=1
      K=1
C     FIND PROBABILITIES ASSOCIATED WITH THESE SIGMA BOUNDARIES
      DO 30 I=1,39
      IF (SLFT.LE.SIG(I+1).AND.SLFT.GE.SIG(I)) THEN
       A2=AREA(I)-0.5
       J=I
      ENDIF
      IF (SRGT.LE.SIG(I+1).AND.SRGT.GE.SIG(I)) THEN
       A1=AREA(I)-0.5
       K=I
      ENDIF
30    CONTINUE

      TOP=AREA(J+1)-AREA(J)
      BOT=SIG(J+1)-SIG(J)
      A2=A2+(SLFT-SIG(J))*TOP/BOT

      TOP=AREA(K+1)-AREA(K)
      BOT=SIG(K+1)-SIG(K)
      A1=A1+(SRGT-SIG(K))*TOP/BOT

      IF (SLFT.GE.SIG(40)) A2=0.5
      IF (SRGT.GE.SIG(40)) A1=0.5

      IF (BRGT.GE.0.0) THEN
       DPROB=A1+A2
      ELSE
       DPROB=A2-A1
      ENDIF

C
C     VERTICAL AIMPOINT ERROR IS BASED ON AMOUNT OF VERTICAL DISTANCE
C     THE TARGET WILL MOVE WHEN DRIVING UP OR DOWN A GRADE, BASED ON
C      THE TIME OF FLIGHT TO TARGET RANGE.
      DVT=ST*SIN(GRADET*PI/180.)*TT

C     THIS BECOMES THE HEIGHT ABOVE OR BELOW THE TARGET CENTER AT
C     TARGET RANGE
      HEIGHT=ABS(DVT)

C     EVALUATE NEW RANGE ERROR PROBABILTIES

C     FIND TARGET RANGE TOP AND BOTTOM LIMITS IN TERMS
C     OF STANDARD DEVIATIONS WHEN THE VERTICAL AIMPOINT ERROR IS INCLUDED
C     ASSUME TARGET IS THREE BY THREE METERS SQUARE
      VERR=ABS(HEIGHT/1.5)
      BTOP=RSIG-VERR
      BBOT=-RSIG-VERR

C     FIND THESE BOUNDARIES IN TERMS OF STANDARD DEVIATIONS
      STOP=ABS(BTOP)
      SBOT=ABS(BBOT)

      J=1
      K=1
C     FIND PROBABILITIES ASSOCIATED WITH THESE SIGMA BOUNDARIES
      DO 40 I=1,39
      IF (STOP.LE.SIG(I+1).AND.STOP.GE.SIG(I)) THEN
       A1=AREA(I)-0.5
       J=I
      ENDIF
      IF (SBOT.LE.SIG(I+1).AND.SBOT.GE.SIG(I)) THEN
       A2=AREA(I)-0.5
       K=I
      ENDIF
40    CONTINUE

      TOP=AREA(J+1)-AREA(J)
      BOT=SIG(J+1)-SIG(J)
      A1=A1+(STOP-SIG(J))*TOP/BOT

      TOP=AREA(K+1)-AREA(K)
      BOT=SIG(K+1)-SIG(K)
      A2=A2+(SBOT-SIG(K))*TOP/BOT

      IF (STOP.GE.SIG(40)) A1=0.5
      IF (SBOT.GE.SIG(40)) A2=0.5

      IF (BTOP.GE.0.0) THEN
       RPROB=A1+A2
      ELSE
       RPROB=A2-A1
      ENDIF

C     CALCULATE NEW PK BASED ON MOVING TARGET AIMPOINT ERROR

      PK=DPROB*RPROB

      RETURN
      END


      SUBROUTINE READARTY
C     READ ARTILLERY SYSTEMS CHARACTERISTICS
C
C     INPUT CHARACTERISTICS FROM DATA FILES
C
C     FILE STRUCTURE IS DEFINED IN FILE "TILLERY.IN"
C
C     NBFA   # OF BLUE ARTILLERY SYSTEM FILES          (10 MAX)
C     NRFA   # OF RED ARTILLERY SYSTEM FILES           (10 MAX)
C     BANAME  BLUE ARTILLERY SYSTEM FILE NAMES          (10 MAX)
C     RANAME  RED ARTILLERY SYSTEM FILE NAMES           (10 MAX)
C     NAMB   # OF BLUE MUNITIONS FOR EACH ARTILLERY SYSTEM  (5 MAX)
C     NAMR   # OF RED MUNITIONS FOR EACH ARTILLERY SYSTEM   (5 MAX)
C     NAMMOB  NUMBER OF ROUNDS FOR EACH BLUE MUNITION     (10,5)
C     NAMMOR  NUMBER OF ROUNDS FOR EACH RED MUNITION      (10,5)
C     IATYPB  BLUE AMMO TYPE (1,2,OR 3)                   (10,5)
C     IATYPR  RED AMMO TYPE (1,2, OR 3)                   (10,5)
C     IAFAFB  BLUE MUNITION M-I-L OR F-A-F                (10,5)
C     IAFAFR  RED MUNITION M-I-L OR F-A-F                 (10,5)
C     IARNGB  MAX RANGE,TOF,CEP,#SUBS,PATTERN AREA FOR SUBS  (10,5,5)
C     IARNGR  MAX RANGE,TOF,CEP,#SUBS,PATTERN AREA FOR SUBS  (10,5,5)
C     ARTPKB  BLUE ARTILLERY PK TABLE AGAINST 20 RED SYSTEMS  (20,10,5)
C     ARTPKR  RED ARTILLERY PK TABLE AGAINST 20 BLUE SYSTEMS  (20,10,5)
C     IABTUBE BLUE TUBE SEPARATION DISTANCE               (10)
C     IARTUBE RED TUBE SEPARATION DISTANCE                (10)
C      ISCOOTB SHOOT/SCOOT, #RNDS, SCOOT DIST, SCOOT TIME     (10,4)
C      ISCOOTR SHOOT/SCOOT, #RNDS, SCOOT DIST, SCOOT TIME     (10,4)

      COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(10)
     &  ,NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5)
     &  ,IAFAFB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
     &  ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
     &  ISCOOTB(10,4),ISCOOTR(10,4)

      CHARACTER*9 BANAME,RANAME
      CHARACTER*80 LINE

C     INITIALIZE ARRAYS
      DO 116 JJ=1,10
      NAMB(JJ)=0
      NAMR(JJ)=0
      BANAME(JJ)=' '
      RANAME(JJ)=' '
      IABTUBE(JJ)=0
      IARTUBE(JJ)=0
      DO 117 II=1,5
      NAMMOB(JJ,II)=0
      NAMMOR(JJ,II)=0
      IATYPB(JJ,II)=0
      IATYPR(JJ,II)=0
      IAFAFB(JJ,II)=0
      IAFAFR(JJ,II)=0
      IF (II.LT.5) THEN
       ISCOOTB(JJ,II)=0
       ISCOOTR(JJ,II)=0
      ENDIF
      DO 114 KK=1,20
      ARTPKB(KK,JJ,II)=0.0
      ARTPKR(KK,JJ,II)=0.0
114   CONTINUE
      DO 115 KK=1,5
      IARNGB(JJ,II,KK)=0
      IARNGR(JJ,II,KK)=0
115   CONTINUE
117   CONTINUE
116   CONTINUE

C
301   FORMAT (I2)
302   FORMAT (A9)
303   FORMAT (A80)
501   FORMAT (1X,I2)
502   FORMAT (1X,A9)
503   FORMAT (1X,A80)
      OPEN (1,FILE='TILLERY.IN')
      READ (1,301) NBFA
C     WRITE (*,501) NBFA
      DO 310 I=1,NBFA
      READ (1,302) BANAME(I)
C     WRITE (*,502) BANAME(I)
310   CONTINUE
      READ (1,301) NRFA
C     WRITE (*,501) NRFA
      DO 312 I=1,NRFA
      READ (1,302) RANAME(I)
C     WRITE (*,502) RANAME(I)
312   CONTINUE
      CLOSE (1)
C
```

```
C     READ BLUE FILES
C
            DO 400 I=1,NBFA
            OPEN (1,FILE=BANAME(I))
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
320   FORMAT (6I8)
520   FORMAT (1X,8I8)
            READ (1,320) NAMB(I),(NAMMOB(I,J),J=1,5)
C     WRITE (*,520) NAMB(I),(NAMMOB(I,J),J=1,5)
            DO 350 J=1,NAMB(I)
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
321   FORMAT (3I8)
521   FORMAT (1X,3I8)
            READ (1,321) IAMMO,IATYPB(I,J),IAFAFB(I,J)
C     WRITE (*,521) IAMMO,IATYPB(I,J),IAFAFB(I,J)
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
331   FORMAT (5I8)
531   FORMAT (1X,5I8)
            READ (1,331) (IARNGB(I,J,K),K=1,5)
C     WRITE (*,531) (IARNGB(I,J,K),K=1,5)
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
341   FORMAT (10F8.2)
361   FORMAT (1X,10F8.2)
            READ (1,341) (ARTPKB(K,I,J),K=1,10)
C     WRITE (*,361) (ARTPKB(K,I,J),K=1,10)
            READ (1,341) (ARTPKB(K,I,J),K=11,20)
C     WRITE (*,361) (ARTPKB(K,I,J),K=11,20)
350   CONTINUE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
322   FORMAT (I8)
522   FORMAT (1X,I8)
            READ (1,322) IABTUBE(I)
C     WRITE (*,522) IABTUBE(I)
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
381   FORMAT (4I8)
382   FORMAT (1X,4I8)
            READ (1,381) (ISCOOTB(I,K),K=1,4)
C     WRITE (*,382) (ISCOOTB(I,K),K=1,4)

            CLOSE (1)
400   CONTINUE
C
C     READ RED  FILES
C
            DO 1400 I=1,NRFA
            OPEN (1,FILE=RANAME(I))
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,320) NAMR(I),(NAMMOR(I,J),J=1,5)
C     WRITE (*,520) NAMR(I),(NAMMOR(I,J),J=1,5)
            DO 1350 J=1,NAMR(I)
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,321) IAMMO,IATYPR(I,J),IAFAFR(I,J)
C     WRITE (*,521) IAMMO,IATYPR(I,J),IAFAFR(I,J)
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,331) (IARNGR(I,J,K),K=1,5)
C     WRITE (*,531) (IARNGR(I,J,K),K=1,5)
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,341) (ARTPKR(K,I,J),K=1,10)
C     WRITE (*,361) (ARTPKR(K,I,J),K=1,10)
            READ (1,341) (ARTPKR(K,I,J),K=11,20)
C     WRITE (*,361) (ARTPKR(K,I,J),K=11,20)
1350  CONTINUE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,322) IARTUBE(I)
C     WRITE (*,522) IARTUBE(I)
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,303) LINE
C     WRITE (*,503) LINE
            READ (1,361) (ISCOOTR(I,K),K=1,4)
C     WRITE (*,382) (ISCOOTR(I,K),K=1,4)

            CLOSE (1)
1400  CONTINUE

            RETURN
            END


            SUBROUTINE ARTINIT
C           INITIALIZES ARTILLERY UNIT AVAILABILITY

            COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
           & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

            COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(10)
           & ,NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5)
           & ,IAFAFB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
           & ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
           & ISCOOTB(10,4),ISCOOTR(10,4)

            COMMON/AAVAIL/IBAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)

            COMMON/SCOOT/ITSC(200,2),ITAVL(200,2)

            COMMON/AMISH/NFMIS(2),MFUB(200,5),MFUR(200,5),XYFUB(200,2),
           & XYFUR(200,2),MILAB(200,10),MILAR(200,10)

            CHARACTER*9 BANAME,RANAME

            CHARACTER*16 BHIER,RHIER,IHIER
            CHARACTER*24 BUNIT,RUNIT
            CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

C           AVAILABILITY FLAGS
C            1 -- AVAILABLE FOR A FIRE MISSION
C            0 -- UNAVAILABLE (CURRENTLY COMMITTED -- SHOOTING)
C           -1 -- UNAVAILABLE (SCOOTING)
C            9 -- NOT AN ARTILLERY UNIT

            NFMIS(1)=0
            NFMIS(2)=0
            DO 10 I=1,200
            DO 5 J=1,5
            MFUB(I,J)=0
5           MFUR(I,J)=0
            DO 6 J=1,10
            MILAB(I,J)=0
6           MILAR(I,J)=0
            DO 7 J=1,2
            ITSC(I,J)=0
            ITAVL(I,J)=0
            XYFUB(I,J)=0.0
7           XYFUR(I,J)=0.0
            IF (IBSYS(I,1).GT.10) THEN
             IBAVL(I)=1
             KSYS=IBSYS(I,1)-10
             NRARTB(I,1)=NAMMOB(KSYS,1)
             NRARTB(I,2)=NAMMOB(KSYS,2)
             NRARTB(I,3)=NAMMOB(KSYS,3)
             NRARTB(I,4)=NAMMOB(KSYS,4)
             NRARTB(I,5)=NAMMOB(KSYS,5)
            ELSE
             IBAVL(I)=9
             NRARTB(I,1)=0
             NRARTB(I,2)=0
             NRARTB(I,3)=0
             NRARTB(I,4)=0
             NRARTB(I,5)=0
            ENDIF
            IF (IRSYS(I,1).GT.10) THEN
             IRAVL(I)=1
             KSYS=IRSYS(I,1)-10
             NRARTR(I,1)=NAMMOR(KSYS,1)
             NRARTR(I,2)=NAMMOR(KSYS,2)
             NRARTR(I,3)=NAMMOR(KSYS,3)
             NRARTR(I,4)=NAMMOR(KSYS,4)
             NRARTR(I,5)=NAMMOR(KSYS,5)
            ELSE
             IRAVL(I)=9
             NRARTR(I,1)=0
             NRARTR(I,2)=0
             NRARTR(I,3)=0
             NRARTR(I,4)=0
             NRARTR(I,5)=0
```

```
      ENDIF
10  CONTINUE

      RETURN
      END


      SUBROUTINE IDFIREORS
C
CCC   ENTER INDIRECT-FIRE ORDERS FOR ARTILLERY UNITS
C
      CHARACTER*1 C2

10    KSIDE=0
      WRITE (*,*)
      WRITE (*,*) 'ARTILLERY FIRE MISSION ORDERS'
      WRITE (*,*) 'enter BLUE or RED, or QUIT'
      WRITE (*,*)
      READ (*,*,ERR=10) C2
      IF (C2(1:1).EQ.'B') KSIDE=1
      IF (C2(1:1).EQ.'R') KSIDE=2
      IF (C2(1:1).EQ.'Q') GO TO 1000
      IF (KSIDE.EQ.0) GO TO 10
C
5     WRITE (*,*)
      WRITE (*,*) '1 -- LIST CURRENT MISSION BACKLOG'
      WRITE (*,*) '2 -- LIST AVAILABLE ARTILLERY UNITS'
      WRITE (*,*) '3 -- LIST AVAILABLE TARGETS'
      WRITE (*,*) '4 -- MATCH A TARGET WITH ARTILLERY UNIT RANGES'
      WRITE (*,*) '5 -- REQUEST A FIRE MISSION'
      WRITE (*,*) '6 -- QUIT'
      WRITE (*,*)
      READ (*,*,ERR=5) MISSION
      IF (MISSION.LT.0.OR.MISSION.GT.6) GO TO 5
      IF (MISSION.EQ.6) GO TO 10

      IF (MISSION.EQ.1) CALL BACKLOG (KSIDE)
      IF (MISSION.EQ.2) CALL ARTAVAIL (KSIDE)
      IF (MISSION.EQ.3) CALL ARTTGTS (KSIDE)
      IF (MISSION.EQ.4) CALL ARTRNG (KSIDE)
      IF (MISSION.EQ.5) CALL ARTMISH (KSIDE)
      GO TO 5

1000  CONTINUE
      RETURN
      END


      SUBROUTINE BACKLOG (KSIDE)
C     DISPLAYS CURRENT ARTILLERY FIRE MISSION BACKLOG

      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

      COMMON/AMISH/NFMIS(2),MFUB(200,5),MFUR(200,5),XYFUB(200,2),
     & XYFUR(200,2),MILAB(200,10),MILAR(200,10)

      COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(1
0)
     & ,NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5)
     & ,IAFAFB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
     & ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
     & ISCOOTB(10,4),ISCOOTR(10,4)

      CHARACTER*9 BANAME,RANAME

      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

      WRITE (*,*)
      WRITE (*,*) 'FIRE MISSION BACKLOG'
      IF (KSIDE.EQ.1) THEN
       WRITE (*,*) 'BLUE'
       NM=NFMIS(1)
      ELSE
       WRITE (*,*) 'RED'
       NM=NFMIS(2)
      ENDIF
   WRITE (*,*) 'MISS #  UNIT #  UNIT I.D.          #RNDS  RNDTYPE
     &  AIM XY      AREA   E.T.A'
      KK=0
      DO 20 K=1,NM
10    FORMAT (1X,I5,2X,I5,3X,A24,I5,1X,I5,1X,F7.3,1X,F7.3,1X,I6,1X,I5)
11    FORMAT (1X,10I5)
      IF (KSIDE.EQ.1) THEN
      IF (MFUB(K,1).EQ.0) GO TO 20
      WRITE (*,10) K,MFUB(K,1),BUNIT(MFUB(K,1)),MFUB(K,2),MFUB(K,3),
     &XYFUB(K,1),XYFUB(K,2),MFUB(K,4),MFUB(K,5)
       KASYS=IBSYS(MFUB(K,1),1)-10
       IF (IAFAFB(KASYS,MFUB(K,3)).EQ.1) THEN
        WRITE (*,*) 'GUIDED PROJECTILE TARGET LIST'
        WRITE (*,11) (MILAB(K,J),J=1,10)
       ENDIF
      ELSE
      IF (MFUR(K,1).EQ.0) GO TO 20
      WRITE (*,10) K,MFUR(K,1),RUNIT(MFUR(K,1)),MFUR(K,2),MFUR(K,3),
     &XYFUR(K,1),XYFUR(K,2),MFUR(K,4),MFUR(K,5)
       KASYS=IRSYS(MFUR(K,1),1)-10
       IF (IAFAFR(KASYS,MFUR(K,3)).EQ.1) THEN
        WRITE (*,*) 'GUIDED PROJECTILE TARGET LIST'
        WRITE (*,11) (MILAR(K,J),J=1,10)
       ENDIF
      ENDIF
      KK=KK+1
      IF (KK.EQ.20) THEN
              READ (*,*)
              KK=0
      ENDIF
20    CONTINUE

      RETURN
      END


      SUBROUTINE ARTAVAIL (KSIDE)

C     DISPLAYS CURRENTLY AVAILABLE ARTILLERY UNITS AND STATUS

      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

      COMMON/AAVAIL/IBAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)

      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*8 STATUS

C       AVAILABILITY FLAGS
C       1 -- AVAILABLE FOR A FIRE MISSION
C       0 -- UNAVAILABLE (CURRENTLY COMMITTED -- SHOOTING)
C      -1 -- UNAVAILABLE (SCOOTING)
C       9 -- NOT AN ARTILLERY UNIT

      STATUS=' '
      WRITE (*,*)
      WRITE (*,*) 'ARTILLERY UNIT AVAILABILITY'
      IF (KSIDE.EQ.1) THEN
       WRITE (*,*) 'BLUE'
      ELSE
       WRITE (*,*) 'RED'
      ENDIF
      WRITE (*,*) 'UNIT #  UNIT I.D.          #TUBES  AMMO BY #
     &STATUS'
      KK=0
      DO 20 K=1,200
10      FORMAT (1X,I5,2X,A24,I3,1X,5(I3,1X),A8)
      IF (KSIDE.EQ.1) THEN
       IF (IBSYS(K,1).GT.10) THEN
        IF (IBAVL(K).EQ.1) STATUS='********'
        IF (IBAVL(K).EQ.0) STATUS='SHOOTING'
        IF (IBAVL(K).EQ.-1) STATUS='SCOOTING'
        IF (IBSYS(K,2).LE.0) STATUS='*KILLED'
       WRITE (*,10) K,BUNIT(K),IBSYS(K,2),(NRARTB(K,J),J=1,5),STATUS
        KK=KK+1
       ENDIF
      ELSE
       IF (IRSYS(K,1).GT.10) THEN
        IF (IRAVL(K).EQ.1) STATUS='********'
        IF (IRAVL(K).EQ.0) STATUS='SHOOTING'
        IF (IRAVL(K).EQ.-1) STATUS='SCOOTING'
        IF (IRSYS(K,2).LE.0) STATUS='*KILLED'
       WRITE (*,10) K,RUNIT(K),IRSYS(K,2),(NRARTR(K,J),J=1,5),STATUS
        KK=KK+1
       ENDIF
      ENDIF
      IF (KK.EQ.20) THEN
              READ (*,*)
              KK=0
      ENDIF
20    CONTINUE

      RETURN
      END


      SUBROUTINE ARTTGTS (KSIDE)
C
CC      FINDS AND DISPLAYS ALL POSSIBLE ARTILLERY TARGETS BASED ON
C       FORWARD OBSERVER AND AIRBORNE SURVEILLANCE ASSETS
C
C       AIRBORNE ASSETS GIVE POSITION, DIRECTION, AND SPEED OF ALL
C       THREAT VEHICLES, BUT THEY ARE UNIDENTIFIED.
C
CC      THE FORWARD OBSERVER CAN IDENTIFY A VEHICLE BY SYSTEM TYPE
C
      COMMON/ARTTGT/ARTGTB(200,2),ARTGTR(200,2),ITLB,ITLR

      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/TRACKING/BOREB(10,3),BORER(10,3),IRNGB(30,4),IRNGR(30,4),
     & ITRKB(10,4),ITRKR(10,4),TOFB(10,3),TOFR(10,3)
      COMMON/FLAGS/ITER

      COMMON/AAVAIL/IBAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)
```

```
      INTEGER ARTGTB,ARTGTR

      CHARACTER*16 BHIER,RHIER,HIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

      IF (KSIDE.EQ.1) THEN
C     BLUE

      WRITE (*,*)
      WRITE (*,*) ' RED UNITS TARGETABLE FOR BLUE ARTILLERY FIRE'
      WRITE (*,*)

      IF (ITLB.EQ.0) THEN

      DO 10 I=1,NR
      ARTGTB(I,1)=0
      ARTGTB(I,2)=0
      IF (IRSYS(I,2).GE.1) ARTGTB(I,1)=1
10    CONTINUE

      DO 150 K=1,NB
C                                   LOOP 1
C     CHECK TO SEE IF THIS BLUE UNIT IS KILLED AND SKIP
      IF (IBSYS(K,2).EQ.0.OR.IBSYS(K,1).GT.10) GO TO 150
C
C     FIND ALL RED TARGETS                LOOP 3
C
C     FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
C     SO THAT MASKING CAN BE EVALUATED
C
      KSIDE=1
      MYSIDE=1
      IBLUE=K
      CALL VMASK (KSIDE,IBLUE)
C
C     FIND ALL RED TARGETS OUT TO 6000 METERS RANGE

      NSYS=IBSYS(K,1)
      RMAX1=6000.

C     GET X,Y COORDINATES FOR THIS BLUE UNIT
      XB=BXYA(K,1)
      YB=BXYA(K,2)
C
C     IF THIS BLUE SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C     FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C     EVALUATED
      ITRK1=ITRKB(NSYS,1)
      IF (ITRK1.EQ.0)
     & CALL DMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C
C     FIND ANGLE AND RANGE TO ALL BURNING VEHICLES, SINCE THEY COMPLETELY
C     MASK VISUAL AND IR SYSTEMS
      CALL COOKMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C
C     IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C     FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C     CAN BE EVALUATED
      ITRK2=ITRKB(NSYS,2)
      IF (ITRK2.EQ.0)
     & CALL ESMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C
C     IF THIS BLUE SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C     FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C     CAN BE EVALUATED
      ITRK3=ITRKB(NSYS,3)
      IF (ITRK3.EQ.0)
     & CALL GRMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C
C     IF THIS BLUE SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C     FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
MASKING
C     CAN BE EVALUATED
      ITRK4=ITRKB(NSYS,4)
      IF (ITRK4.EQ.0)
     & CALL ASMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C
      DO 149 J=1,NR
C                                   LOOP 4
C     CHECK TO SEE IF THIS RED UNIT IS KILLED OR ALREADY IDENTIFIED
      IF (IRSYS(J,2).EQ.0.OR.ARTGTB(J,2).EQ.1) GO TO 149
C     FIND THIS RED UNIT'S X,Y COORDINATES
      XR=RXYA(J,1)
      YR=RXYA(J,2)
C     FIND RANGE TO THIS TARGET
      RANGE=((XR-XB)**2+(YR-YB)**2)**.5

C     CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
C     NOT DETECTED
      CALL CHECKSIG (1,K,J,RANGE,JSKIP)
      IF (JSKIP.EQ.0) GO TO 149

C     IF THIS TARGET IS OUTSIDE THE DETECTION RANGE SKIP IT
      JSKIP=0
      IF ((RANGE*1000.).LE.RMAX1) JSKIP=1
      IF (JSKIP.EQ.0) GO TO 149
C     TARGET IS WITHIN DETECTION RANGE
C
C     EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE
C     MASKING AND OBSCURANTS AND DROP THIS
C     TARGET IF UNOBSERVABLE
C
C     TERRAIN MASKING
      IF (ITER.EQ.1) CALL TERRMASK (XB,YB,XR,YR,JSKIP)
      IF (JSKIP.EQ.0) GO TO 149
C
C     VEHICLE MASKING
      CALL VSEE (MYSIDE,K,J,RANGE,AZIN,JSKIP,NB,NR)
C
C     BURNING VEHICLE MASKING
      CALL COOKSEE (XB,YB,XR,YR,AZIN,JSKIP)

      IF (JSKIP.EQ.0) GO TO 149
C
C     DUST MASKING
C     IF THIS BLUE SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C     EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET
      JDUST=1
      IF (ITRK1.EQ.0)
     & CALL DSEE (XB,YB,XR,YR,AZIN,JDUST)
C     IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE THROUGH IT
C     SKIP THIS TARGET
      IF (JDUST.EQ.0.AND.ITRK1.EQ.0) GO TO 149
C
C     ENGINE SMOKE MASKING
C     IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C     EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
      JENG=1
      IF (ITRK2.EQ.0)
     & CALL ESSEE (XB,YB,XR,YR,AZIN,JENG)
      IF (JENG.EQ.0.AND.ITRK2.EQ.0) GO TO 149
C
C     SMOKE GRENADE CURTAIN MASKING
C     IF THIS BLUE SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C     EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
      JSMK=1
      IF (ITRK3.EQ.0)
     & CALL GRSEE (XB,YB,XR,YR,AZIN,JSMK)
      IF (JSMK.EQ.0.AND.ITRK3.EQ.0) GO TO 149
C
C     ARTILLERY SMOKE CURTAIN MASKING
C     IF THIS BLUE SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C     EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS TARGET
      JART=1
      IF (ITRK4.EQ.0)
     & CALL ASSEE (XB,YB,XR,YR,AZIN,JART)
      IF (JART.EQ.0.AND.ITRK4.EQ.0) GO TO 149

C     THIS RED VEHICLE CAN BE TARGETED BY BLUE ARTILLERY
      ARTGTB(J,2)=1

149   CONTINUE

C     END OF FINDING A TARGET FOR THIS BLUE SYSTEM

150   CONTINUE
      ITLB=1
      ENDIF
C
C     LIST AVAILABLE TARGETS
      WRITE (*,*) 'RED #  SYS #  SPEED  HEADING   X      Y'
      KK=0
      DO 20 K=1,200
11    FORMAT (1X,I5,2X,I5,2X,I5,2X,I5,4X,I5,4X,F8.4,F8.4)
C     COUNTER-BATTERY FLAG (DISPLAY UNIT ID IF ARTILLERY AND SHOOTING)
      KBFLAG=-10
      IF (IRSYS(K,2).GT.0) KBFLAG=IRAVL(K)
      IF (ARTGTB(K,2).EQ.1.OR.KBFLAG.EQ.0) THEN
      WRITE (*,11) K,IRSYS(K,1),ISPDR(K),IRSA(K,2),RXYA(K,1),RXYA(K,2)
      KK=KK+1
      ENDIF
      IF (ARTGTB(K,1).EQ.1.AND.ARTGTB(K,2).EQ.0.AND.KBFLAG.NE.0) THEN
12    FORMAT (1X,I5,2X,'..?..',2X,I5,4X,I5,4X,F8.4,F8.4)
      WRITE (*,12) K,ISPDR(K),IRSA(K,2),RXYA(K,1),RXYA(K,2)
      KK=KK+1
      ENDIF
      IF (KK.EQ.20) THEN
                  READ (*,*)
                  KK=0
      ENDIF
20    CONTINUE

      ELSE
C     RED

      WRITE (*,*)
      WRITE (*,*) ' BLUE UNITS TARGETABLE FOR RED ARTILLERY FIRE'
      WRITE (*,*)

      IF (ITLR.EQ.0) THEN

      DO 110 I=1,NB
      ARTGTR(I,1)=0
      ARTGTR(I,2)=0
      IF (IBSYS(I,2).GE.1) ARTGTR(I,1)=1
110   CONTINUE

      DO 1150 K=1,NR
C                                   LOOP 1
C     CHECK TO SEE IF THIS RED UNIT IS KILLED AND SKIP
```

```
            IF (IRSYS(K,2).EQ.0.OR.IRSYS(K,1).GT.10) GO TO 1150
C
C     FIND ALL BLUE TARGETS                 LOOP 3
C
C     FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
C     SO THAT MASKING CAN BE EVALUATED
C
            KSIDE=2
            MYSIDE=2
            IRED=K
            CALL VMASK (KSIDE,IRED)
C
C     FIND ALL BLUE TARGETS OUT TO 6000 METERS RANGE

            NSYS=IRSYS(K,1)
            RMAX1=6000.

C     GET X,Y COORDINATES FOR THIS RED  UNIT
            XB=RXYA(K,1)
            YB=RXYA(K,2)
C
C     IF THIS RED  SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C     FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C     EVALUATED
            ITRK1=ITRKR(NSYS,1)
            IF (ITRK1.EQ.0)
     & CALL DMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C
C     FIND ANGLE AND RANGE TO ALL BURNING VEHICLES, SINCE THEY COMPLETELY
C     MASK VISUAL AND IR SYSTEMS
            CALL COOKMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C
C     IF THIS RED  SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C     FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C     CAN BE EVALUATED
            ITRK2=ITRKR(NSYS,2)
            IF (ITRK2.EQ.0)
     & CALL ESMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C
C     IF THIS RED  SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C     FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C     CAN BE EVALUATED
            ITRK3=ITRKR(NSYS,3)
            IF (ITRK3.EQ.0)
     & CALL GRMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C
C     IF THIS RED  SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C     FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
MASKING
C     CAN BE EVALUATED
            ITRK4=ITRKR(NSYS,4)
            IF (ITRK4.EQ.0)
     & CALL ASMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C
            DO 1149 J=1,NB
C                                 LOOP 4
C     CHECK TO SEE IF THIS BLUE UNIT IS KILLED OR ALREADY IDENTIFIED
            IF (IBSYS(J,2).EQ.0.OR.ARTGTR(J,2).EQ.1) GO TO 1149
C     FIND THIS BLUE UNIT'S X,Y COORDINATES
            XR=BXYA(J,1)
            YR=BXYA(J,2)
C     FIND RANGE TO THIS TARGET
            RANGE=((XR-XB)**2+(YR-YB)**2)**.5

C     CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
C     NOT DETECTED
            CALL CHECKSIG (2,K,J,RANGE,JSKIP)
            IF (JSKIP.EQ.0) GO TO 1149

C     IF THIS TARGET IS OUTSIDE THE DETECTION RANGE SKIP IT
            JSKIP=0
            IF ((RANGE*1000.).LE.RMAX1) JSKIP=1
            IF (JSKIP.EQ.0) GO TO 1149
C     TARGET IS WITHIN DETECTION RANGE
C
C     EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE
C     MASKING AND OBSCURANTS AND DROP THIS
C     TARGET IF UNOBSERVABLE
C
C     TERRAIN MASKING
            IF (ITER.EQ.1) CALL TERRMASK (XB,YB,XR,YR,JSKIP)
            IF (JSKIP.EQ.0) GO TO 1149
C
C     VEHICLE MASKING
            CALL VSEE (MYSIDE,K,J,RANGE,AZIN,JSKIP,NB,NR)
C
C     BURNING VEHICLE MASKING
            CALL COOKSEE (XB,YB,XR,YR,AZIN,JSKIP)

            IF (JSKIP.EQ.0) GO TO 1149
C
C     DUST MASKING
C     IF THIS RED SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C     EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET
            JDUST=1
            IF (ITRK1.EQ.0)
     & CALL DSEE (XB,YB,XR,YR,AZIN,JDUST)
C     IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE THROUGH IT
C     SKIP THIS TARGET
            IF (JDUST.EQ.0.AND.ITRK1.EQ.0) GO TO 1149
C
C     ENGINE SMOKE MASKING
C     IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C     EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
      JENG=1
      IF (ITRK2.EQ.0)
     & CALL ESSEE (XB,YB,XR,YR,AZIN,JENG)
      IF (JENG.EQ.0.AND.ITRK2.EQ.0) GO TO 1149
C
C     SMOKE GRENADE CURTAIN MASKING
C     IF THIS RED SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C     EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
      JSMK=1
      IF (ITRK3.EQ.0)
     & CALL GRSEE (XB,YB,XR,YR,AZIN,JSMK)
      IF (JSMK.EQ.0.AND.ITRK3.EQ.0) GO TO 1149
C
C     ARTILLERY SMOKE CURTAIN MASKING
C     IF THIS RED  SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C     EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS TARGET
      JART=1
      IF (ITRK4.EQ.0)
     & CALL ASSEE (XB,YB,XR,YR,AZIN,JART)
      IF (JART.EQ.0.AND.ITRK4.EQ.0) GO TO 1149

C     THIS BLUE VEHICLE CAN BE TARGETED BY RED ARTILLERY
      ARTGTR(J,2)=1

1149  CONTINUE

C     END OF FINDING A TARGET FOR THIS RED SYSTEM

1150  CONTINUE
      ITLR=1
      ENDIF
C
C     LIST AVAILABLE TARGETS
      WRITE (*,*) 'BLUE#  SYS #  SPEED  HEADING  X     Y'
      KK=0
      DO 120 K=1,200
      KBFLAG=-10
      IF (IBSYS(K,2).GT.0) KBFLAG=IBAVL(K)
      IF (ARTGTR(K,2).EQ.1.OR.KBFLAG.EQ.0) THEN
      WRITE (*,11) K,IBSYS(K,1),ISPDB(K),IBSA(K,2),BXYA(K,1),BXYA(K,2)
      KK=KK+1
      ENDIF
      IF (ARTGTR(K,1).EQ.1.AND.ARTGTR(K,2).EQ.0.AND.KBFLAG.NE.0) THEN
      WRITE (*,12) K,ISPDB(K),IBSA(K,2),BXYA(K,1),BXYA(K,2)
      KK=KK+1
      ENDIF
      IF (KK.EQ.20) THEN
            READ (*,*)
            KK=0
      ENDIF
120   CONTINUE


      ENDIF

      RETURN
      END


      SUBROUTINE ARTRNG (KSIDE)

C     DISPLAYS CURRENTLY AVAILABLE ARTILLERY UNITS WITHIN RANGE
C     OF A SELECTED TARGET

      COMMON/AAVAIL/IBAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)
      COMMON/ARTTGT/ARTGTB(200,2),ARTGTR(200,2),ITLB,ITLR

      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/TRACKING/BOREB(10,3),BORER(10,3),IRNGB(30,4),IRNGR(30,4),
     & ITRKB(10,4),ITRKR(10,4),TOFB(10,3),TOFR(10,3)

      COMMON/AMISH/NFMIS(2),MFUB(200,5),MFUR(200,5),XYFUB(200,2),
     & XYFUR(200,2),MILAB(200,10),MILAR(200,10)

      COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(10)
     & ,NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5)
     & ,IAFAFB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
     & ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
     & ISCOOTB(10,4),ISCOOTR(10,4)

      INTEGER ARTGTB,ARTGTR
      DIMENSION RMAX(5),IRMAX(5)

      CHARACTER*9 BANAME,RANAME
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

C     AVAILABILITY FLAGS
C     1 -- AVAILABLE FOR A FIRE MISSION

      IF (KSIDE.EQ.1) THEN
```

```
          IF (ITLB.EQ.0) THEN
          DO 211 I=1,NR
          ARTGTB(I,1)=0
          ARTGTB(I,2)=0
211   CONTINUE
          ENDIF

10    WRITE (*,*)
          WRITE (*,*) 'ENTER RED TARGET NUMBER (0=QUIT)'
          WRITE (*,*)
          READ (*,*,ERR=10) J
          IF (J.LE.0) GO TO 1200
          IF (ARTGTB(J,1).EQ.0.AND.ARTGTB(J,2).EQ.0) THEN
           WRITE (*,*) 'NOT TARGETABLE ... SEE TARGETABLE LIST'
           GO TO 10
          ENDIF
          XT=RXYA(J,1)
          YT=RXYA(J,2)
          WRITE (*,*)
          WRITE (*,*) 'AVAILABLE BLUE ARTILLERY WITHIN RANGE OF TARGET ',
     &J
          WRITE (*,*) 'UNIT #  SYS #   UNIT I.D.        RANGE    T.O.F
     & AMMO BY # 1=YES'
          WRITE (*,*)
11    FORMAT (1X,I5,1X,I5,1X,A24,I5,1X,I5,1X,5(I1,1X))
          DO 100 K=1,NB
          IF (IBSYS(K,1).LE.10) GO TO 100
          IF (IBSYS(K,2).LE.0) GO TO 100
          IF (IBAVL(K).NE.1) GO TO 100
          KSYS=IBSYS(K,1)-10
          XB=BXYA(K,1)
          YB=BXYA(K,2)
          RANGE=((XT-XB)**2+(YT-YB)**2)**.5
          RANGE=RANGE*1000.
          ITOF=0
          DO 50 KAMMO=1,5
          RMAX(KAMMO)=IARNGB(KSYS,KAMMO,1)
          IF (RMAX(KAMMO).GE.RANGE) THEN
           IRMAX(KAMMO)=1
           ITOF=NINT(RANGE/RMAX(KAMMO)*REAL(IARNGB(KSYS,KAMMO,2)))
          ELSE
           IRMAX(KAMMO)=0
          ENDIF
50    CONTINUE
          IF (ITOF.NE.0) WRITE (*,11) K,IBSYS(K,1),BUNIT(K),NINT(RANGE),
     &ITOF,(IRMAX(KAMMO),KAMMO=1,5)
100   CONTINUE
          GO TO 10

          ELSE

          IF (ITLR.EQ.0) THEN
          DO 1211 I=1,NB
          ARTGTR(I,1)=0
          ARTGTR(I,2)=0
1211  CONTINUE
          ENDIF

110   WRITE (*,*)
          WRITE (*,*) 'ENTER BLUE TARGET NUMBER (0=QUIT)'
          WRITE (*,*)
          READ (*,*,ERR=110) J
          IF (J.LE.0) GO TO 1200
          IF (ARTGTR(J,1).EQ.0.AND.ARTGTR(J,2).EQ.0) THEN
           WRITE (*,*) 'NOT TARGETABLE ... SEE TARGETABLE LIST'
           GO TO 110
          ENDIF
          XT=BXYA(J,1)
          YT=BXYA(J,2)
          WRITE (*,*)
          WRITE (*,*) 'AVAILABLE RED ARTILLERY WITHIN RANGE OF TARGET ',
     &J
          WRITE (*,*) 'UNIT #  SYS #   UNIT I.D.        RANGE    T.O.F
     & AMMO BY # 1=YES'
          WRITE (*,*)
          DO 1100 K=1,NR
          IF (IRSYS(K,1).LE.10) GO TO 1100
          IF (IRSYS(K,2).LE.0) GO TO 1100
          IF (IRAVL(K).NE.1) GO TO 1100
          KSYS=IRSYS(K,1)-10
          XB=RXYA(K,1)
          YB=RXYA(K,2)
          RANGE=((XT-XB)**2+(YT-YB)**2)**.5
          RANGE=RANGE*1000.
          ITOF=0
          DO 150 KAMMO=1,5
          RMAX(KAMMO)=IARNGR(KSYS,KAMMO,1)
          IF (RMAX(KAMMO).GE.RANGE) THEN
           IRMAX(KAMMO)=1
           ITOF=NINT(RANGE/RMAX(KAMMO)*REAL(IARNGR(KSYS,KAMMO,2)))
          ELSE
           IRMAX(KAMMO)=0
          ENDIF
150   CONTINUE
          IF (ITOF.NE.0) WRITE (*,11) K,IRSYS(K,1),RUNIT(K),NINT(RANGE),
     &ITOF,(IRMAX(KAMMO),KAMMO=1,5)
1100  CONTINUE
          GO TO 110

          ENDIF

1200  RETURN
      END


      SUBROUTINE ARTMISH (KSIDE)
C     REQUESTS AN ARTILLERY FIRE MISSION

      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

      COMMON/AMISH/NFMIS(2),MFUB(200,5),MFUR(200,5),XYFUB(200,2),
     & XYFUR(200,2),MILAB(200,10),MILAR(200,10)

      COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(10)
     & ,NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5)
     & ,IAFAFB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
     & ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
     & ISCOOTB(10,4),ISCOOTR(10,4)

      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
     & ,BXYP(200,2),RXYP(200,2),ISPDB(200),ISPDR(200)
      COMMON/TRACKING/BOREB(10,3),BORER(10,3),IRNGB(30,4),IRNGR(30,4),
     & ITRKB(10,4),ITRKR(10,4),TOFB(10,3),TOFR(10,3)
      COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
     & PKTB(200,2),PKTR(200,2),IKILLB(200),IKILLR(200),IAMB(200),
     & IAMB(200)

      COMMON/AAVAIL/IBAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)
      COMMON/ARTTGT/ARTGTB(200,2),ARTGTR(200,2),ITLB,ITLR

      COMMON/SCOOT/ITSC(200,2),ITAVL(200,2)

      INTEGER ARTGTB,ARTGTR

      DIMENSION IGUIDE(10)

      CHARACTER*9 BANAME,RANAME

      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C
CCC     FIRE MISSION DATA LIST
CCC     K          -- MISSION NUMBER
CCC     NFMIS(2)   -- # OF BLUE/RED FIRE MISSIONS IN BACKLOG
CCC     MFUB/R(K,1) -- ARTILLERY UNIT NUMBER
CCC          (K,2) -- # ROUNDS TO BE FIRED FOR EACH TUBE
CCC          (K,3) -- ROUND TYPE #
CCC          (K,4) -- TOTAL BEATEN AREA FOR ALL ROUNDS (METERS**2)
CCC          (K,5) -- ESTIMATED TIME OF ARRIVAL (IMPACT) SECONDS
CCC     XYFUB/R    -- X,Y POSITION (KM) OF IMPACT POINT
CCC     MILAB/R    -- GUIDED PROJECTILE TARGET LIST (UP TO 10)
CCC
C     COMPRESS EXISTING LOG TO TAKE OUT RESOLVED MISSIONS
      IF (KSIDE.EQ.1) THEN
      IF (ITLB.EQ.0) THEN
      DO 211 I=1,NR
      ARTGTB(I,1)=0
      ARTGTB(I,2)=0
211   CONTINUE
      ENDIF
       NTOTAL=NFMIS(1)
       NUM1=0
       DO 4 I=1,NFMIS(1)
       IF (MFUB(I,1).EQ.0) THEN
        NUM1=I
        DO 3 K=NUM1,NFMIS(1)
        MFUB(K,1)=MFUB((K+1),1)
        MFUB(K,2)=MFUB((K+1),2)
        MFUB(K,3)=MFUB((K+1),3)
        MFUB(K,4)=MFUB((K+1),4)
        MFUB(K,5)=MFUB((K+1),5)
        XYFUB(K,1)=XYFUB((K+1),1)
        XYFUB(K,2)=XYFUB((K+1),2)
        DO 6 J=1,10
6         MILAB(K,J)=MILAB((K+1),J)
        NTOTAL=NTOTAL-1
3       CONTINUE
       ENDIF
4      CONTINUE
       IF (NTOTAL.LE.0) NTOTAL=0
       NFMIS(1)=NTOTAL

      ELSE

      IF (ITLR.EQ.0) THEN
      DO 1211 I=1,NB
      ARTGTR(I,1)=0
      ARTGTR(I,2)=0
1211  CONTINUE
      ENDIF
       NTOTAL=NFMIS(2)
       NUM1=0
       DO 41 I=1,NFMIS(2)
       IF (MFUR(I,1).EQ.0) THEN
        NUM1=I
        DO 31 K=NUM1,NFMIS(2)
        MFUR(K,1)=MFUR((K+1),1)
```

```
            MFUR(K,2)=MFUR((K+1),2)
            MFUR(K,3)=MFUR((K+1),3)
            MFUR(K,4)=MFUR((K+1),4)
            MFUR(K,5)=MFUR((K+1),5)
            XYFUR(K,1)=XYFUR((K+1),1)
            XYFUR(K,2)=XYFUR((K+1),2)
            DO 61 J=1,10
61        MILAR(K,J)=MILAR((K+1),J)
            NTOTAL=NTOTAL-1
31        CONTINUE
            ENDIF
41        CONTINUE
            IF (NTOTAL.LE.0) NTOTAL=0
            NFMIS(2)=NTOTAL
            ENDIF


            PI=3.14159
5       CONTINUE
            DO 7 I=1,10
7       IGUIDE(I)=0
            NUMA=0
            KSYS=0
            NUMR=0
            ITYPE=0
            MILTYPE=0
            NTARG=0
            IAREA=0
            IXT=0
            IYT=0
            IDIR=0
            IAZ=0
            ISPD=0
            IETA=0
            ITOFMAX=0
            IRANGE=0
            IRMAX=0
            IDELAY=0

10      CONTINUE
            WRITE (*,*)
            WRITE (*,*) 'REQUEST NEW FIRE MISSION'
C                                   LOOP 1
            IF (KSIDE.EQ.1) THEN
C       BLUE
            NUMNEW=NFMIS(1)+1
            WRITE (*,*) 'BLUE'
            WRITE (*,*) 'NEW MISSION NUMBER    ',NUMNEW
            WRITE (*,*) '1 -- ARTILLERY UNIT # ',NUMA
            WRITE (*,*) '2 -- TARGET UNIT #    ',NTARG
            WRITE (*,*) 'PROJECTED TARGET POSITION: XT (M) ',IXT,' YT ',IYT
            WRITE (*,*) '3 -- BEATEN AREA (M**2) OR BASKET AREA',IAREA
            WRITE (*,*) '4 -- # ROUNDS PER TUBE ',NUMR
            WRITE (*,*) '5 -- ROUND TYPE #      ',ITYPE
            WRITE (*,*) '6 -- DELAY FIRING TIME ',IDELAY
            WRITE (*,*) 'E.T.A (sec) --         ',IETA
            WRITE (*,*) 'TARGET RANGE --        ',IRANGE
            WRITE (*,*)
            IF (MILTYPE.EQ.1) THEN
              WRITE (*,*) '7 -- GUIDED PROJECTILE TARGET LIST'
              WRITE (*,*) (IGUIDE(J),J=1,10)
              WRITE (*,*)
            ENDIF
            IF (NTARG.EQ.-1) THEN
              WRITE (*,*) '8 -- SMOKE ROUNDS AIMPOINT XT,YT'
              WRITE (*,*) IXT,IYT
              WRITE (*,*)
            ENDIF
            IF (NUMA.GT.0.AND.NTARG.GT.0.AND.NUMR.GT.0.AND.ITYPE.GT.0) THEN
             KSYS=IBSYS(NUMA,1)
             IF (KSYS.GT.10) THEN
              KASYS=KSYS-10
              IFC=IATYPB(KASYS,ITYPE)
              IF (IFC.EQ.2) THEN
C       ESTIMATE AREA TARGET PATTERN EFFECTIVENESS
               CALL FCEST (1,NUMA,KASYS,ITYPE,NUMR,NTARG,IAREA)
              ENDIF
              IF (IFC.EQ.1) THEN
C       ESTIMATE POINT TARGET PATTERN EFFECTIVENESS
               CALL PTEST (1,NUMA,KASYS,ITYPE,NUMR,NTARG,IAREA)
              ENDIF
             ENDIF
            ENDIF
            WRITE (*,*)
            WRITE (*,*) 'ENTER ITEM NUMBER TO CHANGE (ONE AT A TIME)'
            WRITE (*,*) '0 = QUIT'
            WRITE (*,*)
            READ (*,*,ERR=10) ITEM
            IF (ITEM.LT.0.OR.ITEM.GT.8) GO TO 10
            IF (ITEM.EQ.0) GO TO 100
11      IF (ITEM.EQ.7) THEN
             WRITE (*,*) 'ENTER UP TO TEN TARGET NUMBERS (FILL WITH 0)'
             WRITE (*,*)
             READ (*,*,ERR=11) (IGUIDE(J),J=1,10)
            ELSE
9       IF (ITEM.EQ.8) THEN
             WRITE (*,*) 'ENTER XT, YT SMOKE AIMPOINT (METERS)'
             WRITE (*,*)
             READ (*,*,ERR=9) IXT,IYT
            ELSE
             WRITE (*,*) 'ENTER NEW VALUE OF ITEM ',ITEM
             WRITE (*,*)
             READ (*,*,ERR=11) NITEM
            ENDIF
C           IF (NITEM.LT.0) GO TO 11
            ENDIF
            IF (ITEM.EQ.1) NUMA=NITEM
            IF (ITEM.EQ.2) NTARG=NITEM
            IF (ITEM.EQ.3) IAREA=NITEM
            IF (ITEM.EQ.4) NUMR=NITEM
            IF (ITEM.EQ.5) ITYPE=NITEM
            IF (ITEM.EQ.6) IDELAY=NITEM
12          IF (NTARG.EQ.0) THEN
             IXT=0
             IYT=0
             IRANGE=0
             IETA=0
             IDIR=0
             IAZ=0
             ISPD=0
            ENDIF
            IF (NTARG.EQ.-1) ISPD=0
            IF (NTARG.GT.0) THEN
             IF (ARTGTB(NTARG,1).EQ.0.AND.ARTGTB(NTARG,2).EQ.0) THEN
              WRITE (*,*) 'TARGET ',NTARG,' IS NOT ON TARGETABLE LIST'
              NTARG=0
              WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
              READ (*,*)
              GO TO 12
             ENDIF
             IXT=NINT(RXYA(NTARG,1)*1000.)
             IYT=NINT(RXYA(NTARG,2)*1000.)
             ISPD=ISPDR(NTARG)
             IAZ=IRSA(NTARG,2)
             AZIN=REAL(IAZ)
             CALL FDIR(AZIN,AZOUT)
             IDIR=NINT(AZOUT)

C            WRITE (*,*) 'IXT, IYT, ISPD, IAZ, IDIR'
C            WRITE (*,*) IXT,IYT,ISPD,IAZ,IDIR

            ENDIF
13          IF (NUMA.EQ.0) THEN
             KSYS=0
             IXT=0
             IYT=0
             IRANGE=0
             IETA=0
             IMOVE=0
             ITSTART=15
            ELSE
             KSYS=IBSYS(NUMA,1)
             KASYS=IBSYS(NUMA,1)-10
             IF (KSYS.LE.10) THEN
              WRITE (*,*) 'BLUE ',NUMA,' IS NOT AN ARTILLERY UNIT'
              NUMA=0
              WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
              READ (*,*)
              GO TO 13
             ENDIF
             IF (IBAVL(NUMA).NE.1) THEN
              WRITE (*,*) 'BLUE ARTILLERY UNIT',NUMA,' IS NOT AVAILABLE'
              NUMA=0
              WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
              READ (*,*)
              GO TO 13
             ENDIF
             IXA=NINT(BXYA(NUMA,1)*1000.)
             IYA=NINT(BXYA(NUMA,2)*1000.)
             IMOVE=ISPDB(NUMA)

C            WRITE (*,*) 'IXA, IYA, IMOVE'
C            WRITE (*,*) IXA,IYA,IMOVE

             IF (IMOVE.GT.0) THEN
              ITSTART=60
             ELSE
              ITSTART=15
             ENDIF
            ENDIF
14          IF (ITYPE.EQ.0) THEN
             IXT=0
             IYT=0
             IRANGE=0
             IETA=0
             MILTYPE=0
             IRMAX=0
             ITOFMAX=0
            ELSE
             IF (KSYS.NE.0) THEN
              MAXTYPE=0
              IF (NRARTB(NUMA,1).GT.0) MAXTYPE=1
              IF (NRARTB(NUMA,2).GT.0) MAXTYPE=2
              IF (NRARTB(NUMA,3).GT.0) MAXTYPE=3
              IF (NRARTB(NUMA,4).GT.0) MAXTYPE=4
              IF (NRARTB(NUMA,5).GT.0) MAXTYPE=5
              IF (ITYPE.GT.MAXTYPE) THEN
               WRITE (*,*) 'NO AMMO AVAILABLE FOR TYPE ',ITYPE
               ITYPE=0
               WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
               READ (*,*)
               GO TO 14
              ENDIF
              KASYS=IBSYS(NUMA,1)-10
```

```
            MILTYPE=IAFAFB(KASYS,ITYPE)
            IF (MILTYPE.EQ.1.AND.ARTGTB(NTARG,2).EQ.0) THEN
            WRITE (*,*) 'TARGET ',NTARG,' IS NOT TARGETABLE WITH A M-I-L
     &PROJECTILE, RESELECT PROJECTILE TYPE'
             ITYPE=0
             WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
             READ (*,*)
             GO TO 14
            ENDIF
            IRMAX=IARNGB(KASYS,ITYPE,1)
            ITOFMAX=IARNGB(KASYS,ITYPE,2)

C       WRITE (*,*) 'IRMAX, ITOFMAX'
C       WRITE (*,*) IRMAX,ITOFMAX

            IF (IATYPB(KASYS,ITYPE).EQ.3) NTARG=-1

           ENDIF
          ENDIF
15      IF (NUMR.GT.0.AND.ITYPE.NE.0.AND.NUMA.NE.0) THEN
           MAXRNDS=NRARTB(NUMA,ITYPE)
           IF (NUMR.GT.MAXRNDS) THEN
            WRITE (*,*) 'INSUFFICIENT AMMO TO FILL REQUEST '
            WRITE (*,*) 'MAX AVAILABLE OF THIS TYPE ',MAXRNDS
            WRITE (*,*) 'RESELECT NUMBER OF ROUNDS TO FIRE'
            NUMR=0
            WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
            READ (*,*)
            GO TO 15
           ENDIF
           ISCOOT=ISCOOTB(KASYS,1)
           MAXSCOOT=ISCOOTB(KASYS,2)
           IF (NUMR.GT.MAXSCOOT.AND.ISCOOT.EQ.1) THEN
            WRITE (*,*) 'THIS SYSTEM SHOOTS AND SCOOTS '
            WRITE (*,*) 'MAX # ROUNDS EACH MISSION ',MAXSCOOT
            WRITE (*,*) 'RESELECT NUMBER OF ROUNDS TO FIRE'
            NUMR=0
            WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
            READ (*,*)
            GO TO 15
           ENDIF
          ELSE
           NUMR=0
          ENDIF
16      IF (IDELAY.NE.0) THEN
           IF (IDELAY.LT.ITSTART) THEN
            WRITE (*,*) 'DELAY FIRING TIME IS LESS THAN MINIMUM'
            WRITE (*,*) 'SETUP TIME ... DELAY TIME SET TO 0'
            WRITE (*,*)
            IDELAY=0
            WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
            READ (*,*)
           ENDIF
          ENDIF

C                                    LOOP 2
          IF (NTARG.NE.0.AND.NUMA.NE.0.AND.ITYPE.NE.0) THEN
C      CALCULATE ESTIMATED TIME OF ARRIVAL AND FINAL RANGE TO
C      THE TARGET
C      IF TARGET IS MOVING, SOLVE QUADRATIC FORMULA FOR ESTIMATED
C      POSITION AND ETA.
           IF (ISPD.EQ.0) THEN
           IRANGE=NINT(((REAL(IXT)-REAL(IXA))**2+(REAL(IYT)-REAL(IYA))
     &**2)**.5)
           IETA=NINT((REAL(IRANGE)/REAL(IRMAX))*REAL(ITOFMAX))
           IF (IETA.GT.ITOFMAX) THEN
          WRITE (*,*) 'TARGET ',NTARG,' IS OUT OF RANGE OF THIS ARTILLERY
     &UNIT/AMMUNITION PAIR -- SELECT AGAIN'
           IRANGE=0
           IETA=0
           GO TO 10
          ELSE
           IETA=IETA+ITSTART
           IF (IDELAY.GT.ITSTART) IETA=IETA+(IDELAY-ITSTART)
          ENDIF
         ELSE
          X0=REAL(IXT)
          Y0=REAL(IYT)
          XA=REAL(IXA)
          YA=REAL(IYA)
          V=REAL(ISPD)*1000./3600.
          CK=REAL(ITOFMAX)/REAL(IRMAX)
          COST=COS(REAL(IDIR)/180.*PI)
          SINT=SIN(REAL(IDIR)/180.*PI)
          ITGO=ITSTART
          IF (IDELAY.GT.ITSTART) ITGO=ITGO+(IDELAY-ITSTART)

C       WRITE (*,*) 'ITGO'
C       WRITE (*,*) ITGO

          X0=X0+V*COST*REAL(ITGO)
          Y0=Y0+V*SINT*REAL(ITGO)

C       WRITE (*,*) 'X0,Y0,XA,YA'
C       WRITE (*,*) X0, Y0, XA, YA

          A=V*V*COST*ABS(COST)*CK*CK+V*V*SINT*ABS(SINT)*CK*CK-1.
          B=2.*(X0-XA)*V*COST*CK+2.*(Y0-YA)*V*SINT*CK
          C=(X0-XA)**2+(Y0-YA)**2

C       WRITE (*,*) 'A,B,C'
C       WRITE (*,*) A,B,C

          R1=(-B+(B*B-4.*A*C)**.5)/(2.*A)
          R2=(-B-(B*B-4.*A*C)**.5)/(2.*A)

C       WRITE (*,*) 'R1,R2'
C       WRITE (*,*) R1,R2

          IF (R1.GE.R2) RANGE=R1
          IF (R2.GT.R1) RANGE=R2
          IRANGE=NINT(RANGE)
          IETA=NINT((REAL(IRANGE)/REAL(IRMAX))*REAL(ITOFMAX))
          IF (IETA.GT.ITOFMAX) THEN
          WRITE (*,*) 'TARGET ',NTARG,' IS MOVING OUT OF RANGE OF THIS
     *ARTILLERY UNIT/AMMUNITION PAIR -- SELECT AGAIN'
           IRANGE=0
           IETA=0
           GO TO 10
          ELSE
           IXT=NINT(X0+V*COST*REAL(IETA))
           IYT=NINT(Y0+V*SINT*REAL(IETA))
           IETA=IETA+ITGO
          ENDIF
         ENDIF
C                                    END LOOP 2
         ENDIF
         GO TO 10

100      CONTINUE

C        CHECK TO SEE THAT ALL REQUIRED MISSION DATA FIELDS ARE FILLED
C        AND THEN ENTER MISSION INTO BACKLOG

         IF (NUMA.EQ.0.OR.NTARG.EQ.0.OR.NUMR.EQ.0.OR.ITYPE.EQ.0) THEN
101         WRITE (*,*) 'MISSION FIELDS INCOMPLETE'
          WRITE (*,*) '1 -- RETURN TO INPUT MENU'
          WRITE (*,*) '0 -- CANCEL FIRE MISSION'
          WRITE (*,*)
          READ (*,*,ERR=101) ISTOP
          IF (ISTOP.EQ.1) GO TO 10
          IF (ISTOP.EQ.0) GO TO 200
          GO TO 101
         ENDIF

         IF (IAFAFB(KASYS,ITYPE).EQ.1.AND.IGUIDE(1).EQ.0) THEN
102         WRITE (*,*) 'NO TARGET IN GUIDED PROJECTILE TARGET LIST'
          WRITE (*,*) '1 -- RETURN TO INPUT MENU'
          WRITE (*,*) '0 -- CANCEL FIRE MISSION'
          WRITE (*,*)
          READ (*,*,ERR=102) ISTOP
          IF (ISTOP.EQ.1) GO TO 10
          IF (ISTOP.EQ.0) GO TO 200
          GO TO 102
         ENDIF

C        ENTER DATA INTO FIRE MISSION LOG
         K=NUMNEW
         NFMIS(1)=K
         MFUB(K,1)=NUMA
         MFUB(K,2)=NUMR
         MFUB(K,3)=ITYPE
         MFUB(K,4)=IAREA
         MFUB(K,5)=IETA
         XYFUB(K,1)=REAL(IXT)/1000.
         XYFUB(K,2)=REAL(IYT)/1000.
         DO 103 IK=1,10
103         MILAB(K,IK)=IGUIDE(IK)
         IBSA(NUMA,1)=0
         ISPDB(NUMA)=0
         IBAVL(NUMA)=0

C        SET UP SHOOT AND SCOOT INFO AND RE-AVAILABILITY TIME FOR
C        THIS BLUE ARTILLERY UNIT
         IF (ISCOOT.EQ.1) THEN
          ITSC(NUMA,1)=ITGO+60+ITIME
          ITAVL(NUMA,1)=ITGO+60+ISCOOTB(KASYS,4)+ITIME
         ELSE
          ITSC(NUMA,1)=0
          ITAVL(NUMA,1)=ITGO+60+ISCOOTB(KASYS,4)+ITIME
         ENDIF

C        EXPEND APPROPRIATE NUMBER OF ROUNDS PER TUBE
         NRARTB(NUMA,ITYPE)=NRARTB(NUMA,ITYPE)-NUMR

         GO TO 5

C                                    (ELSE) LOOP 1
         ELSE

C        RED
         NUMNEW=NFMIS(2)+1
         WRITE (*,*) 'RED '
         WRITE (*,*) 'NEW MISSION NUMBER    ',NUMNEW
         WRITE (*,*) '1 -- ARTILLERY UNIT # ',NUMA
         WRITE (*,*) '2 -- TARGET UNIT #    ',NTARG
         WRITE (*,*) 'PROJECTED TARGET POSITION: XT (M) ',IXT,' YT ',IYT
         WRITE (*,*) '3 -- BEATEN AREA (M^2) OR BASKET AREA',IAREA
         WRITE (*,*) '4 -- # ROUNDS PER TUBE ',NUMR
         WRITE (*,*) '5 -- ROUND TYPE #      ',ITYPE
         WRITE (*,*) '6 -- DELAY FIRING TIME ',IDELAY
         WRITE (*,*) 'E.T.A (sec) --         ',IETA
         WRITE (*,*) 'TARGET RANGE --        ',IRANGE
         WRITE (*,*)
```

```
            IF (MILTYPE.EQ.1) THEN
             WRITE (*,*) '7 - GUIDED PROJECTILE TARGET LIST'
             WRITE (*,*) (IGUIDE(J),J=1,10)
             WRITE (*,*)
            ENDIF
            IF (NTARG.EQ.-1) THEN
             WRITE (*,*) '8 - SMOKE ROUNDS AIMPOINT XT,YT'
             WRITE (*,*) IXT,IYT
             WRITE (*,*)
            ENDIF
            IF (NUMA.GT.0.AND.NTARG.GT.0.AND.NUMR.GT.0.AND.ITYPE.GT.0) THEN
             KSYS=IRSYS(NUMA,1)
             IF (KSYS.GT.10) THEN
             KASYS=KSYS-10
              IFC=IATYPR(KASYS,ITYPE)
              IF (IFC.EQ.2) THEN
C       ESTIMATE AREA TARGET PATTERN EFFECTIVENESS
               CALL FCEST (2,NUMA,KASYS,ITYPE,NUMR,NTARG,IAREA)
              ENDIF
             ENDIF
            ENDIF
            WRITE (*,*)
            WRITE (*,*) 'ENTER ITEM NUMBER TO CHANGE (ONE AT A TIME)'
            WRITE (*,*) '0 = QUIT'
            WRITE (*,*)
            READ (*,*,ERR=10) ITEM
            IF (ITEM.LT.0.OR.ITEM.GT.8) GO TO 10
            IF (ITEM.EQ.0) GO TO 3100
311     IF (ITEM.EQ.7) THEN
            WRITE (*,*) 'ENTER UP TO TEN TARGET NUMBERS (FILL WITH 0)'
            WRITE (*,*)
            READ (*,*,ERR=311) (IGUIDE(J),J=1,10)
           ELSE
309      IF (ITEM.EQ.8) THEN
            WRITE (*,*) 'ENTER XT, YT SMOKE AIMPOINT (METERS)'
            WRITE (*,*)
            READ (*,*,ERR=309) IXT,IYT
           ELSE
            WRITE (*,*) 'ENTER NEW VALUE OF ITEM ',ITEM
            WRITE (*,*)
            READ (*,*,ERR=311) NITEM
            ENDIF
C        IF (NITEM.LT.0) GO TO 311
            ENDIF
            IF (ITEM.EQ.1) NUMA=NITEM
            IF (ITEM.EQ.2) NTARG=NITEM
            IF (ITEM.EQ.3) IAREA=NITEM
            IF (ITEM.EQ.4) NUMR=NITEM
            IF (ITEM.EQ.5) ITYPE=NITEM
            IF (ITEM.EQ.6) IDELAY=NITEM
312     IF (NTARG.EQ.0) THEN
            IXT=0
            IYT=0
            IRANGE=0
            IETA=0
            IDIR=0
            IAZ=0
            ISPD=0
           ENDIF
           IF (NTARG.EQ.-1) ISPD=0
           IF (NTARG.GT.0) THEN
            IF (ARTGTR(NTARG,1).EQ.0.AND.ARTGTR(NTARG,2).EQ.0) THEN
             WRITE (*,*) 'TARGET ',NTARG,' IS NOT ON TARGETABLE LIST'
             NTARG=0
             WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
             READ (*,*)
             GO TO 312
            ENDIF
            IXT=NINT(8XYA(NTARG,1)*1000.)
            IYT=NINT(8XYA(NTARG,2)*1000.)
            ISPD=ISPDB(NTARG)
            IAZ=IBSA(NTARG,2)
            AZIN=REAL(IAZ)
            CALL FDIR(AZIN,AZOUT)
            IDIR=NINT(AZOUT)

C       WRITE (*,*) 'IXT, IYT, ISPD, IAZ, IDIR'
C       WRITE (*,*) IXT,IYT,ISPD,IAZ,IDIR

           ENDIF
313     IF (NUMA.EQ.0) THEN
            KSYS=0
            IXT=0
            IYT=0
            IRANGE=0
            IETA=0
            IMOVE=0
            ITSTART=15
           ELSE
            KSYS=IRSYS(NUMA,1)
            KASYS=IRSYS(NUMA,1)-10
            IF (KSYS.LE.10) THEN
             WRITE (*,*) 'RED ',NUMA,' IS NOT AN ARTILLERY UNIT'
             NUMA=0
             WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
             READ (*,*)
             GO TO 313
            ENDIF
            IF (IRAVL(NUMA).NE.1) THEN
             WRITE (*,*) 'RED ARTILLERY UNIT',NUMA,' IS NOT AVAILABLE'
             NUMA=0
             WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
             READ (*,*)
             GO TO 313
            ENDIF
            IXA=NINT(RXYA(NUMA,1)*1000.)
            IYA=NINT(RXYA(NUMA,2)*1000.)
            IMOVE=ISPDR(NUMA)

C       WRITE (*,*) 'IXA, IYA, IMOVE'
C       WRITE (*,*) IXA,IYA,IMOVE

            IF (IMOVE.GT.0) THEN
             ITSTART=60
            ELSE
             ITSTART=15
            ENDIF
           ENDIF
314        IF (ITYPE.EQ.0) THEN
            IXT=0
            IYT=0
            IRANGE=0
            IETA=0
            MILTYPE=0
            IRMAX=0
            ITOFMAX=0
           ELSE
            IF (KSYS.NE.0) THEN
             MAXTYPE=0
             IF (NRARTR(NUMA,1).GT.0) MAXTYPE=1
             IF (NRARTR(NUMA,2).GT.0) MAXTYPE=2
             IF (NRARTR(NUMA,3).GT.0) MAXTYPE=3
             IF (NRARTR(NUMA,4).GT.0) MAXTYPE=4
             IF (NRARTR(NUMA,5).GT.0) MAXTYPE=5
             IF (ITYPE.GT.MAXTYPE) THEN
              WRITE (*,*) 'NO AMMO AVAILABLE FOR TYPE ',ITYPE
              ITYPE=0
              WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
              READ (*,*)
              GO TO 314
             ENDIF
             KASYS=IRSYS(NUMA,1)-10
             MILTYPE=IAFAFR(KASYS,ITYPE)
             IF (MILTYPE.EQ.1.AND.ARTGTR(NTARG,2).EQ.0) THEN
             WRITE (*,*) 'TARGET ',NTARG,' IS NOT TARGETABLE WITH A M-I-L
          &PROJECTILE, RESELECT PROJECTILE TYPE'
              ITYPE=0
              WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
              READ (*,*)
              GO TO 314
             ENDIF
             IRMAX=IARNGR(KASYS,ITYPE,1)
             ITOFMAX=IARNGR(KASYS,ITYPE,2)
C            WRITE (*,*) 'IRMAX, ITOFMAX'
C            WRITE (*,*) IRMAX,ITOFMAX

             IF (IATYPR(KASYS,ITYPE).EQ.3) NTARG=-1

            ENDIF
           ENDIF
315        IF (NUMR.GT.0.AND.ITYPE.NE.0.AND.NUMA.NE.0) THEN
            MAXRNDS=NRARTR(NUMA,ITYPE)
            IF (NUMR.GT.MAXRNDS) THEN
             WRITE (*,*) 'INSUFFICIENT AMMO TO FILL REQUEST '
             WRITE (*,*) 'MAX AVAILABLE OF THIS TYPE ',MAXRNDS
             WRITE (*,*) 'RESELECT NUMBER OF ROUNDS TO FIRE'
             NUMR=0
             WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
             READ (*,*)
             GO TO 315
            ENDIF
            ISCOOT=ISCOOTR(KASYS,1)
            MAXSCOOT=ISCOOTR(KASYS,2)
            IF (NUMR.GT.MAXSCOOT.AND.ISCOOT.EQ.1) THEN
             WRITE (*,*) 'THIS SYSTEM SHOOTS AND SCOOTS '
             WRITE (*,*) 'MAX # ROUNDS EACH MISSION ',MAXSCOOT
             WRITE (*,*) 'RESELECT NUMBER OF ROUNDS TO FIRE'
             NUMR=0
             WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
             READ (*,*)
             GO TO 315
            ENDIF
           ELSE
            NUMR=0
           ENDIF
316        IF (IDELAY.NE.0) THEN
            IF (IDELAY.LT.ITSTART) THEN
             WRITE (*,*) 'DELAY FIRING TIME IS LESS THAN MINIMUM'
             WRITE (*,*) 'SETUP TIME ... DELAY TIME SET TO 0'
             WRITE (*,*)
             IDELAY=0
             WRITE (*,*) 'PAUSE ... ENTER TO CONTINUE'
             READ (*,*)
            ENDIF
           ENDIF

C                                        LOOP 2
       IF (NTARG.NE.0.AND.NUMA.NE.0.AND.ITYPE.NE.0) THEN
C        CALCULATE ESTIMATED TIME OF ARRIVAL AND FINAL RANGE TO
C        THE TARGET
C        IF TARGET IS MOVING, SOLVE QUADRATIC FORMULA FOR ESTIMATED
C        POSITION AND ETA.
         IF (ISPD.EQ.0) THEN
         IRANGE=NINT(((REAL(IXT)-REAL(IXA))**2+(REAL(IYT)-REAL(IYA))
```

```
&**2)**.5)
                IETA=NINT((REAL(IRANGE)/REAL(IRMAX))*REAL(ITOFMAX))
                IF (IETA.GT.ITOFMAX) THEN
               WRITE (*,*) 'TARGET ',NTARG,' IS OUT OF RANGE OF THIS ARTILLERY
  &UNIT/AMMUNITION PAIR -- SELECT AGAIN'
                IRANGE=0
                IETA=0
                GO TO 10
               ELSE
                IETA=IETA+ITSTART
                IF (IDELAY.GT.ITSTART) IETA=IETA+(IDELAY-ITSTART)
               ENDIF
              ELSE
               XO=REAL(IXT)
               YO=REAL(IYT)
               XA=REAL(IXA)
               YA=REAL(IYA)
               V=REAL(ISPD)*1000./3600.
               CK=REAL(ITOFMAX)/REAL(IRMAX)
               COST=COS(REAL(IDIR)/180.*PI)
               SINT=SIN(REAL(IDIR)/180.*PI)
               ITGO=ITSTART
               IF (IDELAY.GT.ITSTART) ITGO=ITGO+(IDELAY-ITSTART)

C       WRITE (*,*) 'ITGO'
C       WRITE (*,*) ITGO

               XO=XO+V*COST*REAL(ITGO)
               YO=YO+V*SINT*REAL(ITGO)

C       WRITE (*,*) 'XO,YO,XA,YA'
C       WRITE (*,*) XO, YO, XA, YA

               A=V*V*COST*ABS(COST)*CK*CK+V*V*SINT*ABS(SINT)*CK*CK-1.
               B=2.*(XO-XA)*V*COST*CK+2.*(YO-YA)*V*SINT*CK
               C=(XO-XA)**2+(YO-YA)**2

C       WRITE (*,*) 'A,B,C'
C       WRITE (*,*) A,B,C

               R1=(-B+(B*B-4.*A*C)**.5)/(2.*A)
               R2=(-B-(B*B-4.*A*C)**.5)/(2.*A)

C       WRITE (*,*) 'R1,R2'
C       WRITE (*,*) R1,R2

               IF (R1.GE.R2) RANGE=R1
               IF (R2.GT.R1) RANGE=R2
               IRANGE=NINT(RANGE)
               IETA=NINT((REAL(IRANGE)/REAL(IRMAX))*REAL(ITOFMAX))
               IF (IETA.GT.ITOFMAX) THEN
                WRITE (*,*) 'TARGET ',NTARG,' IS MOVING OUT OF RANGE OF THIS
  *ARTILLERY UNIT/AMMUNITION PAIR -- SELECT AGAIN'
                IRANGE=0
                IETA=0
                GO TO 10
               ELSE
                IXT=NINT(XO+V*COST*REAL(IETA))
                IYT=NINT(YO+V*SINT*REAL(IETA))
                IETA=IETA+ITGO
               ENDIF
              ENDIF
C                                       END LOOP 2
             ENDIF
             GO TO 10

3100   CONTINUE

C      CHECK TO SEE THAT ALL REQUIRED MISSION DATA FIELDS ARE FILLED
C      AND THEN ENTER MISSION INTO BACKLOG

             IF (NUMA.EQ.0.OR.NTARG.EQ.0.OR.NUMR.EQ.0.OR.ITYPE.EQ.0) THEN
3101       WRITE (*,*) 'MISSION FIELDS INCOMPLETE'
             WRITE (*,*) '1 -- RETURN TO INPUT MENU'
             WRITE (*,*) '0 -- CANCEL FIRE MISSION'
             WRITE (*,*)
             READ (*,*,ERR=3101) ISTOP
             IF (ISTOP.EQ.1) GO TO 10
             IF (ISTOP.EQ.0) GO TO 200
             GO TO 3101
            ENDIF

             IF (IAFAFR(KASYS,ITYPE),EQ.1.AND.IGUIDE(1),EQ.0) THEN
3102       WRITE (*,*) 'NO TARGET IN GUIDED PROJECTILE TARGET LIST'
             WRITE (*,*) '1 -- RETURN TO INPUT MENU'
             WRITE (*,*) '0 -- CANCEL FIRE MISSION'
             WRITE (*,*)
             READ (*,*,ERR=3102) ISTOP
             IF (ISTOP.EQ.1) GO TO 10
             IF (ISTOP.EQ.0) GO TO 200
             GO TO 3102
            ENDIF

C      ENTER DATA INTO FIRE MISSION LOG
             K=NUMNEW
             NFMIS(2)=K
             MFUR(K,1)=NUMA
             MFUR(K,2)=NUMR
             MFUR(K,3)=ITYPE
             MFUR(K,4)=IAREA
             MFUR(K,5)=IETA
             XYFUR(K,1)=REAL(IXT)/1000.
             XYFUR(K,2)=REAL(IYT)/1000.
             DO 3103 IK=1,10
3103         MILAR(K,IK)=IGUIDE(IK)
             IRSA(NUMA,1)=0
             ISPOR(NUMA)=0
             IRAVL(NUMA)=0

C       SET UP SHOOT AND SCOOT INFO AND RE-AVAILABILITY TIME FOR
C       THIS RED  ARTILLERY UNIT
             IF (ISCOOT.EQ.1) THEN
              ITSC(NUMA,2)=ITGO+60+ITIME
              ITAVL(NUMA,2)=ITGO+60+ISCOOTR(KASYS,4)+ITIME
             ELSE
              ITSC(NUMA,2)=0
              ITAVL(NUMA,2)=ITGO+60+ISCOOTR(KASYS,4)+ITIME
             ENDIF

C       EXPEND APPROPRIATE NUMBER OF ROUNDS PER TUBE
             NRARTR(NUMA,ITYPE)=NRARTR(NUMA,ITYPE)-NUMR

             GO TO 5

C                                       END LOOP 1
             ENDIF

200    CONTINUE

       RETURN
       END


       SUBROUTINE FIREMISH (ITIME,IDTIME)
C
C      RESOLVES CURRENT BACKLOG OF ARTILLERY IN-DIRECT FIRE MISSIONS
C
       COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
      &  L1BM,L2M,L3M,L1RM
       COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
      &  IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

       COMMON/ARTTGT/ARTGTB(200,2),ARTGTR(200,2),ITLB,ITLR
       COMMON/AMISH/NFMIS(2),MFUB(200,5),MFUR(200,5),XYFUB(200,2),
      &  XYFUR(200,2),MILAB(200,10),MILAR(200,10)

       COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(10)
      &  ,NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5)
      &  ,IAFAFB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
      &  ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
      &  ISCOOTB(10,4),ISCOOTR(10,4)

       COMMON/AAVAIL/IBAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)
       COMMON/SCOOT/ITSC(200,2),ITAVL(200,2)

       DIMENSION IGUIDE(10),ITARGR(200),ITARGB(200)
       CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
       CHARACTER*9 BFNAME,RFNAME,BANAME,RANAME
       CHARACTER*16 BHIER,RHIER,IHIER
       CHARACTER*24 BUNIT,RUNIT

       INTEGER ARTGTB,ARTGTR

       PI=3.14159

       DO 10 IT=1,200
       ITARGB(IT)=0
10     ITARGR(IT)=0

C      RESOLVE BLUE FIRE MISSIONS
       DO 500 N=1,NFMIS(1)
       IF (MFUB(N,1).EQ.0) GO TO 500
       IETA=MFUB(N,5)
       IF (IETA.GT.0) THEN
C        NOT ARRIVED ON TARGET YET
        MFUB(N,5)=MFUB(N,5)-IDTIME
        GO TO 500
       ELSE
C        THIS MISSION HAS COME DUE
        NUMA=MFUB(N,1)
        XA=BXYA(NUMA,1)*1000.
        YA=BXYA(NUMA,2)*1000.
        KASYS=IBSYS(NUMA,1)-10
        NTUBES=IBSYS(NUMA,2)
        NUMR=MFUB(N,2)
        ITYPE=MFUB(N,3)
        MILTYPE=0
        IF (IAFAFB(KASYS,ITYPE).EQ.1) MILTYPE=1
        IATYPE=IATYPB(KASYS,ITYPE)
        IF (MILTYPE.EQ.1) IATYPE=0
        MAXRNG=IARNGB(KASYS,ITYPE,1)
        MAXCEP=IARNGB(KASYS,ITYPE,3)
        NSUBS=IARNGB(KASYS,ITYPE,4)
        IF (NSUBS.LE.0) NSUBS=1
        PATAREA=REAL(IARNGB(KASYS,ITYPE,5))
        BEATAREA=REAL(MFUB(N,4))
        XT=XYFUB(N,1)*1000.
        YT=XYFUB(N,2)*1000.

C       ADJUST IMPACT POINT BASED ON CEP AND RANDOM NUMBER
        RANGE=((XT-XA)**2+(YT-YA)**2)**.5
        CEPM=REAL(MAXCEP)*RANGE/REAL(MAXRNG)
        RDEP=.573*CEPM
        SIGRD=RDEP/.6745
```

```
            CALL CEP (SIGMA)
            SIGRD=SIGRD*SIGMA
            XT=XT+SIGRD
            YT=YT+SIGRD
            ACEP=PI*SIGRD*SIGRD
            IF (BEATAREA.LT.ACEP) BEATAREA=ACEP
            IF (BEATAREA.LT.PATAREA) BEATAREA=PATAREA
            RADIUS=(BEATAREA/PI)**.5

C     WRITE (*,*) 'ACEP, PATAREA, BEATAREA, RADIUS'
C     WRITE (*,*) ACEP,PATAREA,BEATAREA,RADIUS

            DO 103 IK=1,10
103   IGUIDE(IK)=MILAS(N,IK)

            IF (IATYPE.EQ.3) THEN
C
C     CREATE ARTILLERY SMOKE CLOUD
C
            RADIUS=ARTPKB(1,KASYS,ITYPE)
            A1=PI*RADIUS*RADIUS
            AT=A1*NUMR*NTUBES
            IF (BEATAREA.GE.AT) THEN
             RADIUS=(AT/PI)**.5
            ELSE
             RADIUS=(BEATAREA/PI)**.5
            ENDIF
            TSMK=ARTPKB(2,KASYS,ITYPE)
            CALL ARTSMOKE (ITIME,XT,YT,RADIUS,TSMK)
            ENDIF

            IF (MILTYPE.EQ.1) THEN
C     EVALUATE MAN-IN-THE-LOOP TERMINAL GUIDANCE ON DESIGNATED
C     TARGET LIST
C     LISTED TARGETS MUST BE IN THE BASKET AREA PLUS CEP ERROR
C     IF THESE TARGETS ARE NO LONGER THERE, ANY SHORTFALL WILL
C     BE MADE UP BY FINDING ANY OTHER ENEMY TARGETS IN AREA.

C     FIND ALL RED TARGETS IN BASKET AREA
            DO 110 IT=1,NR
            IF (IRSYS(IT,2).LE.0) GO TO 110
            XR=RXYA(IT,1)*1000.
            YR=RXYA(IT,2)*1000.
            R=((XR-XT)**2+(YR-YT)**2)**.5
            IF (R.LE.RADIUS) THEN
C     THIS RED VEHICLE CAN BE TARGETED
            ITARGR(IT)=1
            ENDIF
110   CONTINUE

C     COMPARE TARGET LIST AGAINST AVAILABLE TARGETS
            NSHOTS=NUMR*NTUBES*NSUBS

115   DO 120 IT=1,10
C     IF NO SHOTS LEFT, SKIP IT
            IF (NSHOTS.LE.0) GO TO 120
            NT=IGUIDE(IT)
C     IF NO TARGET ON LIST, SKIP IT
            IF (NT.EQ.0) GO TO 120
C     IF ALREADY KILLED SKIP IT
            IF (IRSYS(NT,2).LE.0) THEN
             IGUIDE(IT)=0
             GO TO 120
            ENDIF
C     SEE IF THIS TARGET IS IN BASKET AND THEN SHOOT IT
            IF (ITARGR(NT).EQ.0) THEN
             IGUIDE(IT)=0
             GO TO 120
            ENDIF
C     SHOOT IT
            JSYS=IRSYS(NT,1)
            PKIT=ARTPKB(JSYS,KASYS,ITYPE)
            IF (JSYS.GT.10) THEN
C     COUNTER-BATTERY SITUATION
            JTUBES=IRSYS(NT,2)
            JASYS=JSYS-10
            DTUBES=REAL(IARTUBE(JASYS))
            NKILLS=0
            CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOTS,PKIT,
     &MILTYPE,IATYPE)
            DO 116 INK=1,NKILLS
            WRITE (10,*) 'RED ',NT,' KILLED BY BLUE ARTILLERY ',NUMA,
     &            ' M-I-L AMMO TYPE ',ITYPE
            CALL WARTY (ITIME,2,NT,1,NUMA,ITYPE)
            CALL OUTPUT2 (ITIME,2,NT,0)
116   CONTINUE
            IRSYS(NT,2)=IRSYS(NT,2)-NKILLS
            IGUIDE(IT)=0
            ITARGR(NT)=0
            ELSE
            PROB=RND()
             IF (PROB.LE.PKIT) THEN
C     TARGET KILLED
              WRITE (10,*) 'RED ',NT,' KILLED BY BLUE ARTILLERY ',NUMA,
     &            ' M-I-L AMMO TYPE ',ITYPE
              IRSYS(NT,2)=IRSYS(NT,2)-1
              IF (IRSYS(NT,2).LT.0) IRSYS(NT,2)=0
C     REMOVE FROM TARGET LIST, SINCE KILLED
              IGUIDE(IT)=0
              ITARGR(NT)=0
              CALL WARTY (ITIME,2,NT,1,NUMA,ITYPE)
             CALL OUTPUT2 (ITIME,2,NT,0)
            ENDIF
           NSHOTS=NSHOTS-1
           ENDIF
120     CONTINUE

        NUM=0
C       IF ANY SHOTS REMAIN, GO DOWN TARGETABLE LIST AND SHOOT AT THEM
        IF (NSHOTS.GE.1) THEN
         DO 130 IT=1,NR
         IF (NUM.EQ.10) GO TO 130
         IF (NUM.EQ.NSHOTS) GO TO 130
         IF (ITARGR(IT).GT.0) THEN
          NUM=NUM+1
          IGUIDE(NUM)=IT
         ENDIF
130      CONTINUE
        ENDIF

        IF (NUM.GT.0) GO TO 115

        ENDIF


        IF (IATYPE.EQ.1) THEN
C       EVALUATE POINT TARGET MUNITIONS/SUBMUNITIONS
C       THESE ARE FIRE-AND-FORGET TERMINALLY GUIDED
C       PROJECTILES/SUBMUNITIONS SUCH AS SADARM OR GUIDED MORTAR
C       ROUNDS
C       FIND ALL VEHICLES IN THE BASKET AREA AND EVALUATE THEIR
C       PROBABILITIES OF BEING TARGETED AND KILLED
C       REMEMBER ... UNLESS THERE IS IFF, WHICH THERE ISN'T, FRIENDLY
C       VEHICLES MAY ALSO BE TARGETED, SO FIND ALL POTENTIAL VEHICLES
C       IN BASKED AREA, INCLUDING KILLED VEHICLES. SINCE HOW THE
C       HECK WILL A MUNITION LIKE THIS KNOW THE DIFFERENCE ?
C       THE FINER POINTS OF THESE ISSUES WILL HAVE TO BE MODELED LATER.
C       FOR NOW, THESE FAF GUIDED PROJECTILES ARE BASICALLY DUMB.
C
        NTR=0
C       FIND ALL RED AND BLUE TARGETS IN BASKET AREA, EVEN DEAD ONES
        DO 210 IT=1,NR
        XR=RXYA(IT,1)*1000.
        YR=RXYA(IT,2)*1000.
        R=((XR-XT)**2+(YR-YT)**2)**.5
        IF (R.LE.RADIUS) THEN
C       THIS RED VEHICLE CAN BE TARGETED
        ITARGR(IT)=1
        NTR=NTR+1
        ENDIF
210     CONTINUE

        NTB=0
        DO 211 IT=1,NB
        XR=BXYA(IT,1)*1000.
        YR=BXYA(IT,2)*1000.
        R=((XR-XT)**2+(YR-YT)**2)**.5
        IF (R.LE.RADIUS) THEN
C       THIS BLUE VEHICLE CAN BE TARGETED
        ITARGB(IT)=1
        NTB=NTB+1
        ENDIF
211     CONTINUE
        NTOTAL=NTR+NTB
        IF (NTOTAL.LE.0) GO TO 300

C       FIND EXPECTED NUMBER OF SHOTS PER AVAILABLE TARGET
        NSHOTS=NUMR*NTUBES*NSUBS
        SEACH=REAL(NSHOTS)/REAL(NTOTAL)
C       BASED ON POISSON'S DISTRIBUTION, FIND THE PROBABILITY
C       THAT A VEHICLE WILL BE WITHIN RANGE OF AT LEAST ONE SUBMUNITION
        P1=1-EXP(-SEACH)

C       LIMIT SHOOT OUT TO NSUBS CYCLES, SINCE THE UNIFORMITY OF
C       THE PATTERN AND TARGET DENSITIES ARE UNKNOWN.
C       IT IS REASONABLE TO ASSUME THAT ANY ONE TARGET WILL BE
C       WITHIN THE ATTACK RANGE OF ANY ONE MISSILE.

        NCYC=0
215     IF (NCYC.GT.NSUBS) GO TO 300

        DO 220 IR=1,NR
        IF (NSHOTS.LE.0) GO TO 300
        IF (ITARGR(IR).NE.1) GO TO 220
        PROB=RND()
        IF (PROB.GT.P1) GO TO 220
C       SHOOT IT
        JSYS=IRSYS(IR,1)
        PKIT=ARTPKB(JSYS,KASYS,ITYPE)
        IF (JSYS.GT.10) THEN
C       COUNTER-BATTERY SITUATION
        JTUBES=IRSYS(IR,2)
        JASYS=JSYS-10
        DTUBES=REAL(IARTUBE(JASYS))
        NKILLS=0
        NSHOT=NSUBS

C       WRITE (*,*) 'CALL COUNTERB, NKILLS,JTUBES,DTUBES,RADIUS,
C      &NSHOT,PKIT,MILTYPE,IATYPE'
C       WRITE (*,*) NKILLS,JTUBES,DTUBES,RADIUS,
C      &NSHOT,PKIT,MILTYPE,IATYPE

        CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,PKIT,
```

```
     &MILTYPE,IATYPE)

C       WRITE (*,*) 'NKILLS'
C       WRITE (*,*) NKILLS

              NSHOTS=NSHOTS-(NSUBS-NSHOT)
              DO 216 INK=1,NKILLS
              WRITE (10,*) 'RED ',IR,' KILLED BY BLUE ARTILLERY ',NUMA,
     &             ' F-A-F GUIDED AMMO TYPE ',ITYPE
              CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
              CALL OUTPUT2 (ITIME,2,IR,0)
216     CONTINUE
              IRSYS(IR,2)=IRSYS(IR,2)-NKILLS
              IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
              ELSE
              PROB=RND()
               IF (PROB.LE.PKIT.AND.IRSYS(IR,2).GT.0) THEN
C       TARGET KILLED
               WRITE (10,*) 'RED ',IR,' KILLED BY BLUE ARTILLERY ',NUMA,
     &             ' F-A-F GUIDED AMMO TYPE ',ITYPE
               IRSYS(IR,2)=IRSYS(IR,2)-1
               IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
               CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
               CALL OUTPUT2 (ITIME,2,IR,0)
              ENDIF
              NSHOTS=NSHOTS-1
              ENDIF
220     CONTINUE

              DO 221 IB=1,NB
              IF (NSHOTS.LE.0) GO TO 300
              IF (ITARGB(IB).NE.1) GO TO 221
              PROB=RND()
              IF (PROB.GT.P1) GO TO 221
C       SHOOT IT
              JSYS=IBSYS(IB,1)
              PKIT=ARTPKB(JSYS,KASYS,ITYPE)
              IF (JSYS.GT.10) THEN
C       COUNTER-BATTERY SITUATION
              JTUBES=IBSYS(IB,2)
              JASYS=JSYS-10
              DTUBES=REAL(IABTUBE(JASYS))
              NKILLS=0
              NSHOT=NSUBS
              CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,PKIT,
     &MILTYPE,IATYPE)
              NSHOTS=NSHOTS-(NSUBS-NSHOT)
              DO 217 INK=1,NKILLS
              WRITE (10,*) 'BLUE ',IB,' KILLED BY BLUE ARTILLERY ',NUMA,
     &             ' F-A-F GUIDED AMMO TYPE ',ITYPE
              CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
              CALL OUTPUT2 (ITIME,1,IB,0)
217     CONTINUE
              IBSYS(IB,2)=IBSYS(IB,2)-NKILLS
              IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
              ELSE
              PROB=RND()
               IF (PROB.LE.PKIT.AND.IBSYS(IB,2).GT.0) THEN
C       TARGET KILLED
               WRITE (10,*) 'BLUE ',IB,' KILLED BY BLUE ARTILLERY ',NUMA,
     &             ' F-A-F GUIDED AMMO TYPE ',ITYPE
               IBSYS(IB,2)=IBSYS(IB,2)-1
               IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
               CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
               CALL OUTPUT2 (ITIME,1,IB,0)
              ENDIF
              NSHOTS=NSHOTS-1
              ENDIF
221     CONTINUE

              NCYC=NCYC+1
              IF (NSHOTS.GT.0) GO TO 215

300     CONTINUE

              ENDIF


              IF (IATYPE.EQ.2) THEN
C     EVALUATE UNGUIDED AREA MUNITIONS SUCH AS EXPLOSIVE ROUNDS (he)
C     AND BOMBLET DISPENSING CARGO PROJECTILES (dpicm)
C     AGAIN, THESE MUNITIONS HAVE NO FRIENDS
C     FIND ALL RED AND BLUE TARGETS IN THE BASKET AREA PLUS CEP ERROR
C     AND EVALUATE FRACTIONAL COVERAGE.
              NTR=0
C     FIND ALL RED AND BLUE TARGETS IN BASKET AREA
              DO 310 IT=1,NR
              IF (IRSYS(IT,2).LE.0) GO TO 310
              XR=RXYA(IT,1)*1000.
              YR=RXYA(IT,2)*1000.
              R=((XR-XT)**2+(YR-YT)**2)**.5
              IF (R.LE.RADIUS) THEN
C     THIS RED VEHICLE CAN BE TARGETED
              ITARGR(IT)=1
              NTR=NTR+1
              ENDIF
310     CONTINUE

              NTB=0
              DO 311 IT=1,NB
              IF (IBSYS(IT,2).LE.0) GO TO 311
              XR=BXYA(IT,1)*1000.
              YR=BXYA(IT,2)*1000.
              R=((XR-XT)**2+(YR-YT)**2)**.5
              IF (R.LE.RADIUS) THEN
C       THIS BLUE VEHICLE CAN BE TARGETED
              ITARGB(IT)=1
              NTB=NTB+1
              ENDIF
311     CONTINUE
              NTOTAL=NTR+NTB
              IF (NTOTAL.LE.0) GO TO 400

C       FIND SHOT DENSITY IN BEATEN AREA

              NSHOTS=NUMR*NTUBES*NSUBS

              IF (NSHOTS.EQ.0) GO TO 400

              SDENS=REAL(NSHOTS)/BEATAREA

C       WRITE (*,*) 'NTR,NTB,NSHOTS,SDENS'
C       WRITE (*,*) NTR,NTB,NSHOTS,SDENS

              DO 320 IR=1,NR
              IF (ITARGR(IR).NE.1) GO TO 320
C       SHOOT IT
              JSYS=IRSYS(IR,1)
              PKRAD=ARTPKB(JSYS,KASYS,ITYPE)
              PKAREA=PI*PKRAD*PKRAD
              EXVAL=PKAREA*SDENS
              PKIT=1-EXP(-EXVAL)

C       WRITE (*,*) 'PKRAD,PKAREA,EXVAL,PKIT'
C       WRITE (*,*) PKRAD,PKAREA,EXVAL,PKIT

              IF (JSYS.GT.10) THEN
C       COUNTER-BATTERY SITUATION
              JTUBES=IRSYS(IR,2)
              JASYS=JSYS-10
              DTUBES=REAL(IARTUBE(JASYS))
              NKILLS=0
              CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOTS,PKIT,
              &MILTYPE,IATYPE)
              DO 315 INK=1,NKILLS
              WRITE (10,*) 'RED ',IR,' KILLED BY BLUE ARTILLERY ',NUMA,
     &             ' SUBMUNITION AMMO TYPE ',ITYPE
              CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
              CALL OUTPUT2 (ITIME,2,IR,0)
315     CONTINUE
              IRSYS(IR,2)=IRSYS(IR,2)-NKILLS
              IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
              ELSE
              PROB=RND()
               IF (PROB.LE.PKIT.AND.IRSYS(IR,2).GT.0) THEN
C       TARGET KILLED
               WRITE (10,*) 'RED ',IR,' KILLED BY BLUE ARTILLERY ',NUMA,
     &             ' SUBMUNITION AMMO TYPE ',ITYPE
               IRSYS(IR,2)=IRSYS(IR,2)-1
               IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
               CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
               CALL OUTPUT2 (ITIME,2,IR,0)
              ENDIF
              ENDIF
320     CONTINUE

              DO 321 IB=1,NB
              IF (ITARGB(IB).NE.1) GO TO 321
C       SHOOT IT
              JSYS=IBSYS(IB,1)
              PKRAD=ARTPKB(JSYS,KASYS,ITYPE)
              PKAREA=PI*PKRAD*PKRAD
              EXVAL=PKAREA*SDENS
              PKIT=1-EXP(-EXVAL)
              IF (JSYS.GT.10) THEN
C       COUNTER-BATTERY SITUATION
              JTUBES=IBSYS(IB,2)
              JASYS=JSYS-10
              DTUBES=REAL(IABTUBE(JASYS))
              NKILLS=0
              CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOTS,PKIT,
              &MILTYPE,IATYPE)
              DO 316 INK=1,NKILLS
              WRITE (10,*) 'BLUE ',IB,' KILLED BY BLUE ARTILLERY ',NUMA,
     &             ' SUBMUNITION AMMO TYPE ',ITYPE
              CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
              CALL OUTPUT2 (ITIME,1,IB,0)
316     CONTINUE
              IBSYS(IB,2)=IBSYS(IB,2)-NKILLS
              IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
              ELSE
              PROB=RND()
               IF (PROB.LE.PKIT.AND.IBSYS(IB,2).GT.0) THEN
C       TARGET KILLED
               WRITE (10,*) 'BLUE ',IB,' KILLED BY BLUE ARTILLERY ',NUMA,
     &             ' SUBMUNITION AMMO TYPE ',ITYPE
               IBSYS(IB,2)=IBSYS(IB,2)-1
               IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
               CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
               CALL OUTPUT2 (ITIME,1,IB,0)
              ENDIF
              ENDIF
321     CONTINUE
```

```
400   CONTINUE
            ENDIF

            MFUB(N,1)=0
            MFUB(N,2)=0
            MFUB(N,3)=0
            MFUB(N,4)=0
            MFUB(N,5)=0
            XYFUB(N,1)=0.
            XYFUB(N,2)=0.

C     DRAW FIRE MISSION IMPACT ZONE ON ACTIVE GRID
            CALL DRAWART (1,XT,YT,RADIUS,IATYPE)

            ENDIF
500   CONTINUE


C     NOW DO RED FIRE MISSIONS
            DO 1500 N=1,NFMIS(2)
            IF (MFUR(N,1).EQ.0) GO TO 1500
            IETA=MFUR(N,5)
            IF (IETA.GT.0) THEN
C     NOT ARRIVED ON TARGET YET
            MFUR(N,5)=MFUR(N,5)-IDTIME
            GO TO 1500
            ELSE
C     THIS MISSION HAS COME DUE
            NUMA=MFUR(N,1)
            XA=RXYA(NUMA,1)*1000.
            YA=RXYA(NUMA,2)*1000.
            KASYS=IRSYS(NUMA,1)+10
            NTUBES=IRSYS(NUMA,2)
            NUMR=MFUR(N,2)
            ITYPE=MFUR(N,3)
            MILTYPE=0
            IF (IAFAFR(KASYS,ITYPE).EQ.1) MILTYPE=1
            IATYPE=IATYPR(KASYS,ITYPE)
            IF (MILTYPE.EQ.1) IATYPE=0
            MAXRNG=IARNGR(KASYS,ITYPE,1)
            MAXCEP=IARNGR(KASYS,ITYPE,3)
            NSUBS=IARNGR(KASYS,ITYPE,4)
            IF (NSUBS.LE.0) NSUBS=1
            PATAREA=REAL(IARNGR(KASYS,ITYPE,5))
            BEATAREA=REAL(MFUR(N,4))
            XT=XYFUR(N,1)*1000.
            YT=XYFUR(N,2)*1000.

C     ADJUST IMPACT POINT BASED ON CEP AND RANDOM NUMBER
            RANGE=((XT-XA)**2+(YT-YA)**2)**.5
            CEPM=REAL(MAXCEP)*RANGE/REAL(MAXRNG)
            RDEP=.573*CEPM
            SIGRD=RDEP/.6745
            CALL CEP (SIGMA)
            SIGRD=SIGRD*SIGMA
            XT=XT+SIGRD
            YT=YT+SIGRD
            ACEP=PI*SIGRD*SIGRD
            IF (BEATAREA.LT.ACEP) BEATAREA=ACEP
            IF (BEATAREA.LT.PATAREA) BEATAREA=PATAREA
            RADIUS=(BEATAREA/PI)**.5

C     WRITE (*,*) 'ACEP, PATAREA, BEATAREA, RADIUS'
C     WRITE (*,*) ACEP,PATAREA,BEATAREA,RADIUS

            DO 1103 IK=1,10
1103  IGUIDE(IK)=MILAR(N,IK)


            IF (IATYPE.EQ.3) THEN
C
C     CREATE ARTILLERY SMOKE CLOUD
C
            RADIUS=ARTPKR(1,KASYS,ITYPE)
            A1=PI*RADIUS*RADIUS
            AT=A1*NUMR*NTUBES
            IF (BEATAREA.GE.AT) THEN
              RADIUS=(AT/PI)**.5
            ELSE
              RADIUS=(BEATAREA/PI)**.5
            ENDIF
            TSMK=ARTPKR(2,KASYS,ITYPE)
            CALL ARTSMOKE (ITIME,XT,YT,RADIUS,TSMK)
            ENDIF

            IF (MILTYPE.EQ.1) THEN
C     EVALUATE MAN-IN-THE-LOOP TERMINAL GUIDANCE ON DESIGNATED
C     TARGET LIST
C     LISTED TARGETS MUST BE IN THE BASKET AREA PLUS CEP ERROR
C     IF THESE TARGETS ARE NO LONGER THERE, ANY SHORTFALL WILL
C     BE MADE UP BY FINDING ANY OTHER ENEMY TARGETS IN AREA.

C     FIND ALL BLUE TARGETS IN BASKET AREA
            DO 1110 IT=1,NB
            IF (IBSYS(IT,2).LE.0) GO TO 1110
            XR=BXYA(IT,1)*1000.
            YR=BXYA(IT,2)*1000.
            R=((XR-XT)**2+(YR-YT)**2)**.5
            IF (R.LE.RADIUS) THEN
C     THIS RED VEHICLE CAN BE TARGETED
            ITARGB(IT)=1
            ENDIF
1110  CONTINUE

C     COMPARE TARGET LIST AGAINST AVAILABLE TARGETS
      NSHOTS=NUMR*NTUBES*NSUBS

1115  DO 1120 IT=1,10
C     IF NO SHOTS LEFT, SKIP IT
      IF (NSHOTS.LE.0) GO TO 1120
      NT=IGUIDE(IT)
C     IF NO TARGET ON LIST, SKIP IT
      IF (NT.EQ.0) GO TO 1120
C     IF ALREADY KILLED SKIP IT
      IF (IBSYS(NT,2).LE.0) THEN
       IGUIDE(IT)=0
       GO TO 1120
      ENDIF
C     SEE IF THIS TARGET IS IN BASKET AND THEN SHOOT IT
      IF (ITARGB(NT).EQ.0) THEN
       IGUIDE(IT)=0
       GO TO 1120
      ENDIF
C     SHOOT IT
      JSYS=IBSYS(NT,1)
      PKIT=ARTPKR(JSYS,KASYS,ITYPE)
      IF (JSYS.GT.10) THEN
C     COUNTER-BATTERY SITUATION
      JTUBES=IBSYS(NT,2)
      JASYS=JSYS-10
      DTUBES=REAL(IABTUBE(JASYS))
      NKILLS=0
      CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOTS,PKIT,
     &MILTYPE,IATYPE)
      DO 1116 INK=1,NKILLS
      WRITE (10,*) 'BLUE ',NT,' KILLED BY RED ARTILLERY ',NUMA,
     &             ' M-I-L AMMO TYPE ',ITYPE
      CALL WARTY (ITIME,1,NT,2,NUMA,ITYPE)
      CALL OUTPUT2 (ITIME,1,NT,0)
1116  CONTINUE
      IBSYS(NT,2)=IBSYS(NT,2)-NKILLS
      IGUIDE(IT)=0
      ITARGB(NT)=0
      ELSE
      PROB=RND()
       IF (PROB.LE.PKIT) THEN
C     TARGET KILLED
       WRITE (10,*) 'BLUE ',NT,' KILLED BY RED ARTILLERY ',NUMA,
     &             ' M-I-L AMMO TYPE ',ITYPE
       IBSYS(NT,2)=IBSYS(NT,2)-1
       IF (IBSYS(NT,2).LT.0) IBSYS(NT,2)=0
C     REMOVE FROM TARGET LIST, SINCE KILLED
       IGUIDE(IT)=0
       ITARGB(NT)=0
       CALL WARTY (ITIME,1,NT,2,NUMA,ITYPE)
       CALL OUTPUT2 (ITIME,1,NT,0)
      ENDIF
      NSHOTS=NSHOTS-1
      ENDIF
1120  CONTINUE

      NUM=0
C     IF ANY SHOTS REMAIN, GO DOWN TARGETABLE LIST AND SHOOT AT THEM
      IF (NSHOTS.GE.1) THEN
       DO 1130 IT=1,NB
       IF (NUM.EQ.10) GO TO 1130
       IF (NUM.EQ.NSHOTS) GO TO 1130
       IF (ITARGB(IT).GT.0) THEN
       NUM=NUM+1
       IGUIDE(NUM)=IT
      ENDIF
1130  CONTINUE
      ENDIF

      IF (NUM.GT.0) GO TO 1115

      ENDIF


      IF (IATYPE.EQ.1) THEN
C     EVALUATE POINT TARGET MUNITIONS/SUBMUNITIONS
C     THESE ARE FIRE-AND-FORGET TERMINALLY GUIDED
C     PROJECTILES/SUBMUNITIONS SUCH AS SADARM OR GUIDED MORTAR
C     ROUNDS
C     FIND ALL VEHICLES IN THE BASKET AREA AND EVALUATE THEIR
C     PROBABILITIES OF BEING TARGETED AND KILLED
C     REMEMBER ... UNLESS THERE IS IFF, WHICH THERE ISN'T, FRIENDLY
C     VEHICLES MAY ALSO BE TARGETED, SO FIND ALL POTENTIAL VEHICLES
C     IN BASKED AREA, INCLUDING KILLED VEHICLES, SINCE HOW THE
C     HECK WILL A MUNITION LIKE THIS KNOW THE DIFFERENCE ?
C     THE FINER POINTS OF THESE ISSUES WILL HAVE TO BE MODELED LATER.
C     FOR NOW, THESE FAF GUIDED PROJECTILES ARE BASICALLY DUMB.
C
      NTR=0
C     FIND ALL RED AND BLUE TARGETS IN BASKET AREA, EVEN DEAD ONES
      DO 1210 IT=1,NR
      XR=RXYA(IT,1)*1000.
      YR=RXYA(IT,2)*1000.
      R=((XR-XT)**2+(YR-YT)**2)**.5
      IF (R.LE.RADIUS) THEN
C     THIS RED VEHICLE CAN BE TARGETED
      ITARGR(IT)=1
      NTR=NTR+1
```

```
            ENDIF
1210    CONTINUE

            NTB=0
            DO 1211 IT=1,NB
            XR=BXYA(IT,1)*1000.
            YR=BXYA(IT,2)*1000.
            R=((XR-XT)**2+(YR-YT)**2)**.5
            IF (R.LE.RADIUS) THEN
C       THIS BLUE VEHICLE CAN BE TARGETED
            ITARGB(IT)=1
            NTB=NTB+1
            ENDIF
1211    CONTINUE
            NTOTAL=NTR+NTB
            IF (NTOTAL.LE.0) GO TO 1300

C       FIND EXPECTED NUMBER OF SHOTS PER AVAILABLE TARGET
            NSHOTS=NUMR*NTUBES*NSUBS
            SEACH=REAL(NSHOTS)/REAL(NTOTAL)
C       BASED ON POISSON'S DISTRIBUTION, FIND THE PROBABILITY
C       THAT A VEHICLE WILL BE WITHIN RANGE OF AT LEAST ONE SUBMUNITION
            P1=1-EXP(-SEACH)

C       LIMIT SHOOT OUT TO NSUBS CYCLES, SINCE THE UNIFORMITY OF
C       THE PATTERN AND TARGET DENSITIES ARE UNKNOWN.
C       IT IS REASONABLE TO ASSUME THAT ANY ONE TARGET WILL BE
C       WITHIN THE ATTACK RANGE OF ANY ONE MISSILE.

            NCYC=0
1215    IF (NCYC.GT.NSUBS) GO TO 1300

            DO 1220 IR=1,NR
            IF (NSHOTS.LE.0) GO TO 1300
            IF (ITARGR(IR).NE.1) GO TO 1220
            PROB=RND()
            IF (PROB.GT.P1) GO TO 1220
C       SHOOT IT
            JSYS=IRSYS(IR,1)
            PKIT=ARTPKR(JSYS,KASYS,ITYPE)
            IF (JSYS.GT.10) THEN
C       COUNTER-BATTERY SITUATION
            JTUBES=IRSYS(IR,2)
            JASYS=JSYS-10
            DTUBES=REAL(IARTUBE(JASYS))
            NKILLS=0
            NSHOT=NSUBS

C       WRITE (*,*) 'CALL COUNTERB, NKILLS,JTUBES,DTUBES,RADIUS,
C      &NSHOT,PKIT,MILTYPE,IATYPE'
C       WRITE (*,*) NKILLS,JTUBES,DTUBES,RADIUS,
C      &NSHOT,PKIT,MILTYPE,IATYPE

            CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,PKIT,
     &MILTYPE,IATYPE)

C       WRITE (*,*) 'NKILLS'
C       WRITE (*,*) NKILLS

            NSHOTS=NSHOTS-(NSUBS-NSHOT)
            DO 1216 INK=1,NKILLS
            WRITE (10,*) 'RED ',IR,' KILLED BY RED ARTILLERY ',NUMA,
     &           ' F-A-F GUIDED AMMO TYPE ',ITYPE
            CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
            CALL OUTPUT2 (ITIME,2,IR,0)
1216    CONTINUE
            IRSYS(IR,2)=IRSYS(IR,2)-NKILLS
            IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
            ELSE
            PROB=RND()
            IF (PROB.LE.PKIT.AND.IRSYS(IR,2).GT.0) THEN
C       TARGET KILLED
            WRITE (10,*) 'RED ',IR,' KILLED BY RED ARTILLERY ',NUMA,
     &           ' F-A-F GUIDED AMMO TYPE ',ITYPE
            IRSYS(IR,2)=IRSYS(IR,2)-1
            IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
            CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
            CALL OUTPUT2 (ITIME,2,IR,0)
            ENDIF
            NSHOTS=NSHOTS-1
            ENDIF
1220    CONTINUE

            DO 1221 IB=1,NB
            IF (NSHOTS.LE.0) GO TO 1300
            IF (ITARGB(IB).NE.1) GO TO 1221
            PROB=RND()
            IF (PROB.GT.P1) GO TO 1221
C       SHOOT IT
            JSYS=IBSYS(IB,1)
            PKIT=ARTPKR(JSYS,KASYS,ITYPE)
            IF (JSYS.GT.10) THEN
C       COUNTER-BATTERY SITUATION
            JTUBES=IBSYS(IB,2)
            JASYS=JSYS-10
            DTUBES=REAL(IABTUBE(JASYS))
            NKILLS=0
            NSHOT=NSUBS
            CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,PKIT,
     &MILTYPE,IATYPE)
            NSHOTS=NSHOTS-(NSUBS-NSHOT)
            DO 1217 INK=1,NKILLS
            WRITE (10,*) 'BLUE ',IB,' KILLED BY RED ARTILLERY ',NUMA,
     &           ' F-A-F GUIDED AMMO TYPE ',ITYPE
            CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
            CALL OUTPUT2 (ITIME,1,IB,0)
1217    CONTINUE
            IBSYS(IB,2)=IBSYS(IB,2)-NKILLS
            IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
            ELSE
            PROB=RND()
            IF (PROB.LE.PKIT.AND.IBSYS(IB,2).GT.0) THEN
C       TARGET KILLED
            WRITE (10,*) 'BLUE ',IB,' KILLED BY RED ARTILLERY ',NUMA,
     &           ' F-A-F GUIDED AMMO TYPE ',ITYPE
            IBSYS(IB,2)=IBSYS(IB,2)-1
            IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
            CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
            CALL OUTPUT2 (ITIME,1,IB,0)
            ENDIF
            NSHOTS=NSHOTS-1
            ENDIF
1221    CONTINUE

            NCYC=NCYC+1
            IF (NSHOTS.GT.0) GO TO 1215

1300    CONTINUE

            ENDIF


            IF (IATYPE.EQ.2) THEN
C       EVALUATE UNGUIDED AREA MUNITIONS SUCH AS EXPLOSIVE ROUNDS (he)
C       AND BOMBLET DISPENSING CARGO PROJECTILES (dpicm)
C       AGAIN, THESE MUNITIONS HAVE NO FRIENDS
C       FIND ALL RED AND BLUE TARGETS IN THE BASKET AREA PLUS CEP ERROR
C       AND EVALUATE FRACTIONAL COVERAGE.
            NTR=0
C       FIND ALL RED AND BLUE TARGETS IN BASKET AREA
            DO 1310 IT=1,NR
            IF (IRSYS(IT,2).LE.0) GO TO 1310
            XR=RXYA(IT,1)*1000.
            YR=RXYA(IT,2)*1000.
            R=((XR-XT)**2+(YR-YT)**2)**.5
            IF (R.LE.RADIUS) THEN
C       THIS RED VEHICLE CAN BE TARGETED
            ITARGR(IT)=1
            NTR=NTR+1
            ENDIF
1310    CONTINUE

            NTB=0
            DO 1311 IT=1,NB
            IF (IBSYS(IT,2).LE.0) GO TO 1311
            XR=BXYA(IT,1)*1000.
            YR=BXYA(IT,2)*1000.
            R=((XR-XT)**2+(YR-YT)**2)**.5

C       WRITE (*,*) 'BLUE, X  Y  R  RADIUS'
C       WRITE (*,*) IT,XR,YR,R,RADIUS

            IF (R.LE.RADIUS) THEN

C       WRITE (*,*) 'YES'

C       THIS BLUE VEHICLE CAN BE TARGETED
            ITARGB(IT)=1
            NTB=NTB+1
            ENDIF
1311    CONTINUE
            NTOTAL=NTR+NTB

C       WRITE (*,*) 'NTOTAL,NR,NB'
C       WRITE (*,*) NTOTAL,NTR,NTB

            IF (NTOTAL.LE.0) GO TO 1400

C       FIND SHOT DENSITY IN BEATEN AREA

            NSHOTS=NUMR*NTUBES*NSUBS

            IF (NSHOTS.EQ.0) GO TO 1400

            SDENS=REAL(NSHOTS)/BEATAREA

C       WRITE (*,*) 'NTR,NTB,NSHOTS,SDENS'
C       WRITE (*,*) NTR,NTB,NSHOTS,SDENS

            DO 1320 IR=1,NR
            IF (ITARGR(IR).NE.1) GO TO 1320
C       SHOOT IT
            JSYS=IRSYS(IR,1)
            PKRAD=ARTPKR(JSYS,KASYS,ITYPE)
            PKAREA=PI*PKRAD*PKRAD
            EXVAL=PKAREA*SDENS
            PKIT=1-EXP(-EXVAL)

C       WRITE (*,*) 'PKRAD,PKAREA,EXVAL,PKIT'
C       WRITE (*,*) PKRAD,PKAREA,EXVAL,PKIT

            IF (JSYS.GT.10) THEN
C       COUNTER-BATTERY SITUATION
            JTUBES=IRSYS(IR,2)
```

```
            JASYS=JSYS-10
            DTUBES=REAL(IARTUBE(JASYS))
            NKILLS=0
            CALL COUNTER8 (NKILLS,JTUBES,DTUBES,RADIUS,NSHOTS,PKIT,
     &MILTYPE,IATYPE)
            DO 1315 NK=1,NKILLS
            WRITE (10,*) 'RED ',IR,' KILLED BY RED ARTILLERY ',NUMA,
     &           ' SUBMUNITION AMMO TYPE ',ITYPE
            CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
            CALL OUTPUT2 (ITIME,2,IR,0)
1315    CONTINUE
            IRSYS(IR,2)=IRSYS(IR,2)-NKILLS
            IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
            ELSE
            PROB=RND()
            IF (PROB.LE.PKIT.AND.IRSYS(IR,2).GT.0) THEN
C       TARGET KILLED
            WRITE (10,*) 'RED ',IR,' KILLED BY RED ARTILLERY ',NUMA,
     &           ' SUBMUNITION AMMO TYPE ',ITYPE
            IRSYS(IR,2)=IRSYS(IR,2)-1
            IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
            CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
            CALL OUTPUT2 (ITIME,2,IR,0)
            ENDIF
            ENDIF
1320    CONTINUE

            DO 1321 IB=1,NB
            IF (ITARGB(IB).NE.1) GO TO 1321
C       SHOOT IT
            JSYS=IBSYS(IB,1)
            PKRAD=ARTPKR(JSYS,KASYS,ITYPE)
            PKAREA=PI*PKRAD*PKRAD
            EXVAL=PKAREA*SDENS
            PKIT=1-EXP(-EXVAL)

C       WRITE (*,*) 'PKRAD,PKAREA,EXVAL,PKIT'
C       WRITE (*,*) PKRAD,PKAREA,EXVAL,PKIT

            IF (JSYS.GT.10) THEN
C       COUNTER-BATTERY SITUATION
            JTUBES=IBSYS(IB,2)
            JASYS=JSYS-10
            DTUBES=REAL(IABTUBE(JASYS))
            NKILLS=0
            CALL COUNTER8 (NKILLS,JTUBES,DTUBES,RADIUS,NSHOTS,PKIT,
     &MILTYPE,IATYPE)
            DO 1316 NK=1,NKILLS
            WRITE (10,*) 'BLUE ',IB,' KILLED BY RED ARTILLERY ',NUMA,
     &           ' SUBMUNITION AMMO TYPE ',ITYPE
            CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
            CALL OUTPUT2 (ITIME,1,IB,0)
1316    CONTINUE
            IBSYS(IB,2)=IBSYS(IB,2)-NKILLS
            IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
            ELSE
            PROB=RND()
            IF (PROB.LE.PKIT.AND.IBSYS(IB,2).GT.0) THEN
C       TARGET KILLED
            WRITE (10,*) 'BLUE ',IB,' KILLED BY RED ARTILLERY ',NUMA,
     &           ' SUBMUNITION AMMO TYPE ',ITYPE
            IBSYS(IB,2)=IBSYS(IB,2)-1
            IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
            CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
            CALL OUTPUT2 (ITIME,1,IB,0)
            ENDIF
            ENDIF
1321    CONTINUE

1400    CONTINUE

            ENDIF

            MFUR(N,1)=0
            MFUR(N,2)=0
            MFUR(N,3)=0
            MFUR(N,4)=0
            MFUR(N,5)=0
            XYFUR(N,1)=0.
            XYFUR(N,2)=0.

C       DRAW FIRE MISSION IMPACT ZONE ON ACTIVE GRID
            CALL DRAWART (2,XT,YT,RADIUS,IATYPE)

            ENDIF
1500    CONTINUE


            RETURN
            END



            SUBROUTINE CEP (SIGMA)
C
C       THIS SUBROUTINE DETERMINES FOR THIS ONE EVENT THE RANGE AND
C       DEFLECTION IMPACT POINT ERROR FOR THE ARTILLERY BARRAGE BASED
C       ON THE CEP INPUT.
C       THIS IMPACT ERROR IS DETERMINED BY TAKING SUCCESSIVE RANDOM
C       NUMBERS (FROM A FLAT PROBABILITY CURVE), (UP TO 7 CYCLES).
C       THE RANDOM NUMBER IS EVALUATED FOR BEING ABOVE OR BELOW 0.5
C       (THE 50% CHANCE POINT), AND THEN ASSIGNED TO A SIGMA BASKET.
C       ALL SIGMA BASKETS ARE OF EQUAL AREA (THEREFORE, EQUAL PROBABILITY).
C       SINCE THE PROBABILITY DENSITY IS A BELL CURVE, THE SIGMA SPACING
C       IS UNEVEN, ALTHOUGH THE PROBABILITIES ARE THE SAME.  THIS RESULTS
C       IN THE IMPACT ERROR BEING ASSIGNED TO A SIGMA BASKET WHICH
C       FOLLOWS THE NORMAL PROBABILITY CURVE.
C       SIGMA BASKETS ARE DEFINED TO WITHIN 3% ERROR AT SMALL SIGMAS TO
C       .8% ERROR AT LARGE SIGMAS (MORE BASKETS ARE NEEDED AS THE CURVE
C       THINS OUT AT THE TAILS, SO THAT ULTIMATE IMPACT POINT ERROR ERRORS
C       ARE REDUCED.
C

            DIMENSION SB(20),RL(20)

            CHARACTER*7 RL,LTEST

            SB(1)=0.04
            SB(2)=0.12
            SB(3)=0.196
            SB(4)=0.2775
            SB(5)=0.3825
            SB(6)=0.4475
            SB(7)=0.535
            SB(8)=0.6275
            SB(9)=0.725
            SB(10)=0.8275
            SB(11)=0.99
            SB(12)=1.125
            SB(13)=1.235
            SB(14)=1.425
            SB(15)=1.605
            SB(16)=1.77
            SB(17)=1.925
            SB(18)=2.07
            SB(19)=2.285
            SB(20)=2.71

            RL(1)='1000000'
            RL(2)='1000100'
            RL(3)='1001000'
            RL(4)='1001100'
            RL(5)='1010000'
            RL(6)='1010100'
            RL(7)='1011000'
            RL(8)='1011100'
            RL(9)='1100000'
            RL(10)='1100100'
            RL(11)='1101000'
            RL(12)='1101100'
            RL(13)='1110000'
            RL(14)='1110100'
            RL(15)='1111000'
            RL(16)='1111010'
            RL(17)='1111100'
            RL(18)='1111101'
            RL(19)='1111110'
            RL(20)='1111111'

            DO 100 I=1,7

            P=RND()
            IF (P.LE.0.50) THEN
C           ERROR IS NEGATIVE, ASSIGN VALUE OF 0
            L(I:I)='0'
            ELSE
C           ERROR IS POSITIVE, ASSIGN VALUE OF 1
            L(I:I)='1'
            ENDIF

100     CONTINUE

            NFLAG=1

            IF (L(1:1).EQ.'0') THEN
             NFLAG=-1
             L(1:1)='1'
            ENDIF

            SIGMA=0

            DO 200 I=1,20
             LTEST=RL(I)
            IF (I.LE.14) THEN
             IF (L(1:5).EQ.LTEST(1:5)) SIGMA=SB(I)
            ENDIF
            IF (I.EQ.15.OR.I.EQ.16) THEN
             IF (L(1:6).EQ.LTEST(1:6)) SIGMA=SB(I)
            ENDIF
            IF (I.GE.17) THEN
             IF (L(1:7).EQ.LTEST(1:7)) SIGMA=SB(I)
            ENDIF
200     CONTINUE

            SIGMA=SIGMA*NFLAG

            RETURN
            END



            SUBROUTINE ARTSMOKE (ITIME,XT,YT,RADIUS,TSMK)
C
C       CREATES ARTILLERY SMOKE PUFFS
```

```
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX

C     ARTILLERY SMOKE CLOUDS
      IF (NART.GT.0) THEN
C      TEST TO SEE IF SMOKE ARRAY NEEDS TO BE COMPRESSED BACK
C      TO COLUMN 1
       ITEST=1+IART2
       IF (ITEST.GT.NARTMX) THEN
C       COMPRESS ARRAY BEFORE STARTING
        J=0
        DO 110 I=IART1,IART2
        J=J+1
        PART(J,1)=PART(I,1)
        PART(J,2)=PART(I,2)
        TAART(J,1)=TAART(I,1)
        TAART(J,2)=TAART(I,2)
110     CONTINUE
        IART1=1
        IART2=J
       ENDIF
      ENDIF
C     CREATE NEW ARTILLERY SMOKE PUFF AFTER IART2
       IART2=IART2+1
       K=IART2
       PART(K,1)=XT/1000.
       PART(K,2)=YT/1000.
       TAART(K,1)=TSMK
       TAART(K,2)=RADIUS
       NART=IART2-IART1+1
C
       RETURN
       END


       SUBROUTINE PARTSMK
C     BASED ON SUBROUTINE PENG
C     PLOTS ARTILLERY SMOKE PUFFS
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX
       COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
       COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
       COMMON/EARPUFFS/IAERASE,PAERASE(99,3)
C
C
       ISKIP=0
C
C     DRAWS A GREEN  CIRCLE FOR EACH PUFF LOCATION.
C     AND SAVES THIS PUFF FOR LATER ERASING.
C     RADIUS IS BASED ON PUFF TIME WHEN AT MAX ZOOM.
C     AT MIN ZOOM, CIRCLE IS DRAWN LARGER SO YOU CAN SEE IT.
C
       CALL FACTOR (ZFACT)
       GXUNIT=11.0/NX*SCFACT
       GYUNIT=8.5/NY
       XUNIT=GXUNIT/SCALE
       YUNIT=GYUNIT/SCALE
       IAERASE=0

       DO 100 I=IART1,IART2
       RADIUS=TAART(I,2)*XUNIT*1.0/1000.
       X=PART(I,1)*XUNIT
       Y=PART(I,2)*YUNIT
       IF (XLEN.GT.0.) THEN
            CALL FACTOR (ZFACTN)
C      RADIUS=RADIUS*ZFACT/ZFACTN
       ENDIF
            CALL NEWPEN (10)
       IF ((X-15.*RADIUS).LE.0.0) ISKIP=1
       IF ((Y-15.*RADIUS).LE.0.0) ISKIP=1
       IF (XLEN.GT.0.) THEN
            CALL ZOOMIT (X,Y,ISKIP)
            IF ((Y+RADIUS).GT.(.95*YMAX)) ISKIP=1
            IF ((X-2.*RADIUS).LE.0.0) ISKIP=1
            IF ((Y-2.*RADIUS).LE.0.0) ISKIP=1
       ENDIF
       IF (ISKIP.EQ.1) GO TO 100
C     DRAW CIRCLE
       CALL CIRCLE (X,Y,RADIUS)
        IAERASE=IAERASE+1
        PAERASE(IAERASE,1)=PART(I,1)
        PAERASE(IAERASE,2)=PART(I,2)
        PAERASE(IAERASE,3)=TAART(I,2)
100   CONTINUE
       RETURN
       END


       SUBROUTINE ASMASK (XS,YS,RMAX1,RMAX2,RMAX3)
C
C     FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE PUFFS WITHIN
C     MAX RANGE OF THIS SYSTEM SO THAT MASKING CAN BE EVALUATED
C
      COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
     & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
     & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
     & TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
     & NARTMX

      COMMON/MASKAR/NARMSK,ARMSK(99,3)

C      XS,YS     -- SHOOTER COORDINATES
C      RMAX1,2,3 -- MAX RANGE OF SYSTEM WEAPONS
C      ARMSK(I,1) -- X COORD OF EACH ARTY SMOKE PUFF
C      ARMSK(I,2) -- Y COORD OF EACH ARTY SMOKE PUFF
C      ARMSK(I,3) -- RADIUS FOR EACH ARTY SMOKE PUFF
C      NARMSK    -- NUMBER OF ARTILLERY SMOKE PUFFS WITHIN MAX RANGE
C
      PI=3.14159
      NARMSK=0
      IF (NART.GT.0) THEN
C       CYCLE THROUGH ALL SMOKE PUFFS TO FILL MASKING ARRAY
      RMAXX=0
      IF (RMAX1.GT.RMAXX) RMAXX=RMAX1
      IF (RMAX2.GT.RMAXX) RMAXX=RMAX2
      IF (RMAX3.GT.RMAXX) RMAXX=RMAX2
      RMAXX=RMAXX/1000.
      DO 20 I=IART1,IART2
       XD=PART(I,1)
       YD=PART(I,2)
       RADIUS=PART(I,3)/1000.
      RANGE=((XD-XS)**2+(YD-YS)**2)**.5
      IF ((RANGE-RADIUS).GT.RMAXX) GO TO 20
      NARMSK=NARMSK+1
      ARMSK(NARMSK,1)=XD
      ARMSK(NARMSK,2)=YD
      ARMSK(NARMSK,3)=RADIUS
20      CONTINUE
      ENDIF
      RETURN
      END


      SUBROUTINE ASSEE (XS,YS,XT,YT,AZIMUTH,JSKIP)
C       EVALUATES ARTILLERY SMOKE MASKING BETWEEN SHOOTER AND TARGET
C
C       XS,YS   SHOOTER REAL WORLD COORDINATES
C       XT,YT   TARGET  REAL WORLD COORDINATES
C       AZIN    AZIMUTH TO TARGET
C       JSKIP   FLAG TO SKIP THIS TARGET IF MASKED (=0 TO SKIP)

      COMMON/ELEVA/CONT(20),IELEV(10000)
      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT

      COMMON/MASKAR/NARMSK,ARMSK(99,3)

      DOUBLE PRECISION ANGLE,ANGLE1

      PI=3.14159

      IF (NARMSK.GT.0) THEN
C
C       FIND SHOOTER AND TARGET GRID COORDINATES
C       TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
C       CENTER TO CENTER ABOUT THE SCALE.
      RRANGE=((YT-YS)**2+(XT-XS)**2)**.5
      IXS=NINT(XS/SCALE)
      IYS=NINT(YS/SCALE)
      IXT=NINT(XT/SCALE)
      IYT=NINT(YT/SCALE)
      IF (IXS.LT.1) IXS=1
      IF (IXT.LT.1) IXT=1
      IF (IYS.LT.1) IYS=1
      IF (IYT.LT.1) IYT=1
      IF (IXS.GT.NX) IXS=NX
      IF (IXT.GT.NX) IXT=NX
      IF (IYS.GT.NY) IYS=NY
      IF (IYT.GT.NY) IYT=NY
C
C       FIND SHOOTER AND TARGET ELEVATION
      IS=NY*(IXS-1)+IYS
      SELEV=CONT(IELEV(IS))
      IT=NY*(IXT-1)+IYT
      TELEV=CONT(IELEV(IT))
C
C       ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
C       FOR VEHICLE HEIGHT
      SELEV=SELEV+.002
      TELEV=TELEV+.002
C
C       FIND ELEVATION ANGLE TO THIS TARGET
      TOP=TELEV-SELEV
      BOT=RRANGE
      ANGLE=ATAN(TOP/BOT)
C
      DO 200 I=1,NARMSK
      IF (JSKIP.EQ.0) GO TO 300
C
C       CYCLE THROUGH ALL ARTY SMOKE PUFFS
C       TO EVALUATE MASKING, FIND THE LEFT AND RIGHT AZIMUTH
C       BRACKETING EACH PUFF WITHIN RANGE.  DROP IF OUT OF ANGLE RANGE.
C       THEN FIND THE ELEVATION ANGLE TO THE TOP OF THE PUFF.
C       DROP PUFF IF NOT HIGH ENOUGH
C
C       EVALUATE RANGE TO SMOKE PUFF
      XP=ARMSK(I,1)
      YP=ARMSK(I,2)
```

```
      RADIUS=ARMSK(L3)
      RP=((YP-YS)**2+(XP-XS)**2)**.5
      IF ((RP-RADIUS).GT.RRANGE) GO TO 200
C
C     EVALUATE ANGLES TO SMOKE PUFF
      TOP=YP-YS
      BOT=XP-XS
      IF (TOP.EQ.0.0.AND.BOT.EQ.0.0) JSKIP=0
      IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
      IF (BOT.LT.0.0) THEN
            IF (TOP.LT.0.0) DIR=DIR+180.
            IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
            IF (TOP.LT.0.0) DIR=360.-DIR
      ENDIF
C     CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
      CALL FAZ (DIR,AZIN)
      IF (RADIUS.LT.0.030) RADIUS=0.030
      IF (RP.LT.RADIUS) RP=RADIUS
      DELAZ=ASIN(RADIUS/RP)*180./PI
      AZLFT=AZIN-DELAZ
      AZRGT=AZIN+DELAZ
      IF (AZLFT.LT.0.0) AZLFT=AZLFT+360.0
      IF (AZRGT.LT.0.0) AZRGT=AZRGT+360.0
      IF (AZLFT.GT.360.0) AZLFT=AZLFT-360.0
      IF (AZRGT.GT.360.0) AZRGT=AZRGT-360.0
      IF (AZLFT.GT.AZRGT) AZRGT=AZRGT+360.
      IF (AZIMUTH.LT.0.0) AZIMUTH=AZIMUTH+360.0
      IF (AZIMUTH.GT.360.0) AZIMUTH=AZIMUTH-360.0
      IF (AZIMUTH.LT.AZLFT.OR.AZIMUTH.GT.AZRGT) GO TO 200
C
CCC   EVALUATE HEIGHT OF SMOKE PUFF
C
C     FIND ELEVATION OF THE GROUND UNDER THIS PUFF
      IXP=NINT(XP/SCALE)
      IYP=NINT(YP/SCALE)
      IF (IXP.LT.1) IXP=1
      IF (IYP.LT.1) IYP=1
      IF (IXP.GT.NX) IXP=NX
      IF (IYP.GT.NY) IYP=NY
      IP=NY*(IXP-1)+IYP
      PELEV=CONT(IELEV(IP))
C     ADD TO THIS ELEVATION THE HEIGHT OF THE PUFF
C     MAX PUFF HEIGHT IS 30 METERS (100 FEET)
      HEIGHT=RADIUS
      IF (RADIUS.GT.0.030) HEIGHT=0.030
      PELEV=PELEV+HEIGHT
C     FIND ELEVATION ANGLE TO THIS DUST PUFF
      TOP=PELEV-SELEV
      BOT=RP
      ANGLEI=ATAN(TOP/BOT)
      IF (ANGLEI.GT.ANGLE) JSKIP=0
200   CONTINUE

      ENDIF

300   RETURN
      END


      SUBROUTINE DRAWART (KSIDE,XT,YT,RADIUS,IATYPE)

C     BASED ON SUBROUTINE PENG
C     PLOTS ARTILLERY FIRE MISSION ON ACTIVE GRID

      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
      COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
C
      IF (IATYPE.EQ.3) RETURN
      ISKIP=0
C
CCC   DRAWS A COLORED CIRCLE (RED OR BLUE) FOR EACH MISSION LOCATION.
C
      CALL FACTOR (ZFACT)
      GXUNIT=11.0/NX*SCFACT
      GYUNIT=8.5/NY
      XUNIT=GXUNIT/SCALE
      YUNIT=GYUNIT/SCALE
      RADIUS=RADIUS*XUNIT*1.0/1000.
      X=XT*XUNIT/1000.
      Y=YT*YUNIT/1000.
      IF (XLEN.GT.0.) THEN
            CALL FACTOR (ZFACTN)
C     RADIUS=RADIUS*ZFACT/ZFACTN
      ENDIF
            IF (KSIDE.EQ.1) CALL NEWPEN (9)
            IF (KSIDE.EQ.2) CALL NEWPEN (12)
      IF ((X-15.*RADIUS).LE.0.0) ISKIP=1
      IF ((Y-15.*RADIUS).LE.0.0) ISKIP=1
      IF (XLEN.GT.0.) THEN
            CALL ZOOMIT (X,Y,ISKIP)
            IF ((Y+RADIUS).GT.(.95*YMAX)) ISKIP=1
            IF ((X-2.*RADIUS).LE.0.0) ISKIP=1
            IF ((Y-2.*RADIUS).LE.0.0) ISKIP=1
      ENDIF
      IF (ISKIP.EQ.1) GO TO 100
C     DRAW CIRCLE
      CALL CIRCLE (X,Y,RADIUS)
100   CONTINUE
      RETURN
      END


      SUBROUTINE WARTY (KTIME,JSIDE,J,KSIDE,K,ITYPE)

C     ARTILLERY KILLS DATA OUTPUT
C
C     KTIME       KILL TIME
C     JSIDE       TARGET SIDE (1 BLUE, 2 RED)
C     J           VEHICLE NUMBER
C     JSYS        SYSTEM NUMBER
C     KSIDE       SHOOTER SIDE
C     K           SHOOTER NUMBER
C     KSYS        SHOOTER ARTILLERY SYSTEM NUMBER
C     ITYPE       ARTILLERY AMMUNITION NUMBER

      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT

      IF (JSIDE.EQ.1) THEN
       JSYS=IBSYS(J,1)
      ELSE
       JSYS=IRSYS(J,1)
      ENDIF
      IF (KSIDE.EQ.1) THEN
       KSYS=IBSYS(K,1)
      ELSE
       KSYS=IRSYS(K,1)
      ENDIF

1     FORMAT (8(I4,3X))

      WRITE (16,1) KTIME,JSIDE,J,JSYS,KSIDE,K,KSYS,ITYPE

      RETURN
      END


      SUBROUTINE COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOTS,PKIT,
     &MILTYPE,IATYPE)

C     EVALUATES ARTILLERY FIRE ON ARTILLERY UNITS WHERE EACH
C     UNIT NUMBER REPRESENTS MULTIPLE TUBES OR VEHICLES

      PI=3.14159

      NTARG=1
      NKILLS=0

      IF (JTUBES.EQ.0) RETURN

      IF (JTUBES.EQ.2) THEN
       IF (DTUBES.LE.RADIUS) NTARG=2
      ENDIF

      IF (JTUBES.GE.3) THEN

C     DEFINE A POLYGON WHERE EACH TUBE IS LOCATED THE SAME
C     DISTANCE DTUBES FROM EACH OTHER TUBE
C     FIND THE OUTSIDE RADIUS OF THIS POLYGON
      A=DTUBES
      ALPHA=2.*PI/REAL(JTUBES)
      R=A/2/SIN(ALPHA/2.)

C     IF THE MUNITION PATTERN IS AIMED AT ONE OF THE VERTICES,
C     THEN TO GET ALL OF THE TUBES IN THE PATTERN, THE PATTERN
C     RADIUS MUST BE TWICE THE POLYGON RADIUS.

C     IF THE PATTERN IS AIMED AT THE CENTER OF THE POLYGON, THEN
C     ALL TUBES CAN BE HIT IF THE PATTERN RADIUS IS EQUAL TO THE
C     POLYGON RADIUS

C     SINCE, WE DO NOT KNOW, ASSUME IT IS SOMEWHERE IN BETWEEN.
C     THE NUMBER OF TUBES WHICH CAN BE TARGETED EQUALS THE RATIO
C     OF THE TWO DISTANCES.

      RATIO=RADIUS/R

      IF (RATIO.GE.2.) THEN
       NTARG=JTUBES
      ELSE
       NTARG=NINT(RATIO*REAL(JTUBES-1))
      ENDIF

      IF (NTARG.LT.1) NTARG=1
      IF (NTARG.GT.JTUBES) NTARG=JTUBES

      ENDIF

      IF (MILTYPE.EQ.1.OR.IATYPE.EQ.1) THEN
C     M-I-L OR F-A-F GUIDED MUNITION OR          ION
      KSHOT=NSHOTS
      DO 10 I=1,KSHOT
      IF (NTARG.LT.1) RETURN
      PROB=RND()
      IF (PKIT.GE.PROB) THEN
```

```
            NKILLS=NKILLS+1
            NTARG=NTARG-1
          ENDIF
          NSHOTS=NSHOTS-1
10    CONTINUE
          ENDIF

          IF (IATYPE.EQ.2) THEN
C     UNGUIDED MUNITION OR SUBMUNITIONS
C     APPLY FRACTIONAL COVERAGE TO NUMBER OF TARGETS IN PATTERN AREA
          NKILLS=NINT(PKIT*REAL(NTARG))
          ENDIF

C     WRITE (*,*) 'NKILLS'
C     WRITE (*,*) NKILLS

          RETURN
          END




          SUBROUTINE FCEST (KSIDE,NUMA,KASYS,ITYPE,NUMR,JTARG,IAREA)
C
C     ESTIMATES FRACTIONAL COVERAGE OF AREA TARGETS WHEN SELECTING
C     AN ARTILLERY MISSION WHICH USES he OR dpicm MUNITIONS
C
          COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
          COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
          COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(1
0)
     & ,NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5)
     & ,IAFAFB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
     & ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
     & ISCOOTB(10,4),ISCOOTR(10,4)

          CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
          CHARACTER*9 BFNAME,RFNAME,BANAME,RANAME
          CHARACTER*16 BHIER,RHIER,IHIER
          CHARACTER*24 BUNIT,RUNIT

          PI=3.14159

          IF (KSIDE.EQ.1) THEN
C     BLUE
          NTUBES=IBSYS(NUMA,2)
          NSUBS=IARNGB(KASYS,ITYPE,4)
          MCEP=IARNGB(KASYS,ITYPE,3)
          MRNG=IARNGB(KASYS,ITYPE,1)
          IF (NSUBS.LE.0) NSUBS=1
          PATAREA=REAL(IARNGB(KASYS,ITYPE,5))
          XA=BXYA(NUMA,1)*1000.
          YA=BXYA(NUMA,1)*1000.
          XT=RXYA(JTARG,1)*1000.
          YT=RXYA(JTARG,2)*1000.
          JSYS=IRSYS(JTARG,1)
          PKRAD=ARTPKB(JSYS,KASYS,ITYPE)
          ELSE
C     RED
          NTUBES=IRSYS(NUMA,2)
          NSUBS=IARNGR(KASYS,ITYPE,4)
          MCEP=IARNGR(KASYS,ITYPE,3)
          MRNG=IARNGR(KASYS,ITYPE,1)
          IF (NSUBS.LE.0) NSUBS=1
          PATAREA=REAL(IARNGR(KASYS,ITYPE,5))
          XA=RXYA(NUMA,1)*1000.
          YA=RXYA(NUMA,1)*1000.
          XT=BXYA(JTARG,1)*1000.
          YT=BXYA(JTARG,2)*1000.
          JSYS=IBSYS(JTARG,1)
          PKRAD=ARTPKR(JSYS,KASYS,ITYPE)
          ENDIF

          RANGE=((XT-XA)**2+(YT-YA)**2)**.5
          CEP=RANGE/REAL(MRNG)*REAL(MCEP)
          BEATAREA=REAL(IAREA)

          IF (BEATAREA.LT.PATAREA) BEATAREA=PATAREA
          RADIUS=(BEATAREA/PI)**.5
          IF (RADIUS.LT.CEP) THEN
          RADIUS=CEP
          BEATAREA=PI*RADIUS*RADIUS
          ENDIF

          NTR=0
C     FIND ALL RED AND BLUE TARGETS IN BASKET AREA
          DO 310 IT=1,NR
          IF (IRSYS(IT,2).LE.0) GO TO 310
          NT=IRSYS(IT,2)
          IF (NT.GT.1) THEN
          JASYS=IRSYS(IT,1)-10
          DTUBES=REAL(IARTUBE(JASYS))
          NSHOTS=NUMR*NTUBES*NSUBS
          NKILLS=0
          CALL COUNTERB (NKILLS,NT,DTUBES,RADIUS,NSHOTS,1.0,0,2)
          NT=NKILLS
          ENDIF
          XR=RXYA(IT,1)*1000.
          YR=RXYA(IT,2)*1000.
          R=((XR-XT)**2+(YR-YT)**2)**.5
          IF (R.LE.RADIUS) THEN
C     THIS RED VEHICLE CAN BE TARGETED
          NTR=NTR+1*NT
          ENDIF
310     CONTINUE

          NTB=0
          DO 311 IT=1,NB
          IF (IBSYS(IT,2).LE.0) GO TO 311
          NT=IBSYS(IT,2)

C     WRITE (*,*) 'BLUE'
C     WRITE (*,*) IT

          IF (NT.GT.1) THEN
          JASYS=IBSYS(IT,1)-10
          DTUBES=REAL(IABTUBE(JASYS))
          NSHOTS=NUMR*NTUBES*NSUBS

C     WRITE (*,*) 'NSHOTS, DTUBES, JASYS'
C     WRITE (*,*) NSHOTS,DTUBES,JASYS

          NKILLS=0
          CALL COUNTERB (NKILLS,NT,DTUBES,RADIUS,NSHOTS,1.0,0,2)
          NT=NKILLS

C     WRITE (*,*) 'NKILLS'
C     WRITE (*,*) NT

          ENDIF
          XR=BXYA(IT,1)*1000.
          YR=BXYA(IT,2)*1000.
          R=((XR-XT)**2+(YR-YT)**2)**.5

C     WRITE (*,*) 'X  Y  R  RADIUS'
C     WRITE (*,*) XR,YR,R,RADIUS

          IF (R.LE.RADIUS) THEN

C     WRITE (*,*) 'YES'

C     THIS BLUE VEHICLE CAN BE TARGETED
          NTB=NTB+1*NT
          ENDIF
311     CONTINUE

C     FIND SHOT DENSITY IN BEATEN AREA
          NSHOTS=NUMR*NTUBES*NSUBS

          IF (BEATAREA.LE.0.0) GO TO 400
          SDENS=REAL(NSHOTS)/BEATAREA

          PKAREA=PI*PKRAD*PKRAD
          EXVAL=PKAREA*SDENS
          PKIT=1-EXP(-EXVAL)
          FC=PKIT*100.


          WRITE (*,*) 'ESTIMATED FRACTIONAL COVERAGE'
          WRITE (*,*) 'BASED ON BEATEN AREA'
          IF (KSIDE.EQ.1) THEN
          WRITE (*,*) '# BLUE TARGETS, # RED TARGETS, F.C % ON RED'
          ELSE
          WRITE (*,*) '# BLUE TARGETS, # RED TARGETS, F.C % ON BLUE'
          ENDIF
          WRITE (*,*) NTB,NTR,FC
          WRITE (*,*)


400     CONTINUE
          RETURN
          END



          SUBROUTINE PTEST (KSIDE,NUMA,KASYS,ITYPE,NUMR,JTARG,IAREA)
C
C     ESTIMATES POINT TARGET MUNITION EFFECTIVENESS WHEN SELECTING
C     AN ARTILLERY MISSION WHICH USES m-l-l OR l-a-l MUNITIONS
C
          COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
     & L1BM,L2M,L3M,L1RM
          COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
     & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

          COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(10)
     & ,NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5)
     & ,IAFAFB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
     & ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
     & ISCOOTB(10,4),ISCOOTR(10,4)

          CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
          CHARACTER*9 BFNAME,RFNAME,BANAME,RANAME
          CHARACTER*16 BHIER,RHIER,IHIER
          CHARACTER*24 BUNIT,RUNIT

          PI=3.14159

          IF (KSIDE.EQ.1) THEN
C     BLUE
```

```
            NTUBES=IBSYS(NUMA,2)
            NSUBS=IARNGB(KASYS,ITYPE,4)
            MCEP=IARNGB(KASYS,ITYPE,3)
            MRNG=IARNGB(KASYS,ITYPE,1)
            IF (NSUBS.LE.0) NSUBS=1
            PATAREA=REAL(IARNGB(KASYS,ITYPE,5))
            XA=BXYA(NUMA,1)*1000.
            YA=BXYA(NUMA,1)*1000.
            XT=RXYA(JTARG,1)*1000.
            YT=RXYA(JTARG,2)*1000.
            JSYS=IRSYS(JTARG,1)
            MILTYPE=IAFAB(KASYS,ITYPE)
            IATYPE=IATYPB(KASYS,ITYPE)
           ELSE
C     RED
            NTUBES=IRSYS(NUMA,2)
            NSUBS=IARNGR(KASYS,ITYPE,4)
            MCEP=IARNGR(KASYS,ITYPE,3)
            MRNG=IARNGR(KASYS,ITYPE,1)
            IF (NSUBS.LE.0) NSUBS=1
            PATAREA=REAL(IARNGR(KASYS,ITYPE,5))
            XA=RXYA(NUMA,1)*1000.
            YA=RXYA(NUMA,1)*1000.
            XT=BXYA(JTARG,1)*1000.
            YT=BXYA(JTARG,2)*1000.
            JSYS=IBSYS(JTARG,1)
            MILTYPE=IAFAR(KASYS,ITYPE)
            IATYPE=IATYPR(KASYS,ITYPE)
           ENDIF

            RANGE=((XT-XA)**2+(YT-YA)**2)**.5
            CEP=RANGE/REAL(MRNG)*REAL(MCEP)
            BEATAREA=REAL(IAREA)

            IF (BEATAREA.LT.PATAREA) BEATAREA=PATAREA
            RADIUS=(BEATAREA/PI)**.5
            IF (RADIUS.LT.CEP) THEN
             RADIUS=CEP
             BEATAREA=PI*RADIUS*RADIUS
            ENDIF

            IF (KSIDE.EQ.1) THEN

C     BLUE

            IF (MILTYPE.EQ.1) THEN
            IATYPE=0
C     EVALUATE MAN-IN-THE-LOOP TERMINAL GUIDANCE ON DESIGNATED
C      TARGET LIST
C     LISTED TARGETS MUST BE IN THE BASKET AREA PLUS CEP ERROR
C     IF THESE TARGETS ARE NO LONGER THERE, ANY SHORTFALL WILL
C     BE MADE UP BY FINDING ANY OTHER ENEMY TARGETS IN AREA.

            NSHOTS=NUMR*NTUBES*NSUBS
            NTR=0
C     FIND ALL RED TARGETS IN BASKET AREA
            DO 110 IT=1,NR
            IF (IRSYS(IT,2).LE.0) GO TO 110
            XR=RXYA(IT,1)*1000.
            YR=RXYA(IT,2)*1000.
            R=((XR-XT)**2+(YR-YT)**2)**.5
            IF (R.LE.RADIUS) THEN
C     THIS RED VEHICLE CAN BE TARGETED
            NA=1
            JSYS=IRSYS(IT,1)
            PKIT=ARTPKB(JSYS,KASYS,ITYPE)
            IF (JSYS.GT.10) THEN
C     COUNTER-BATTERY SITUATION
            JTUBES=IRSYS(IT,2)
            JASYS=JSYS-10
            DTUBES=REAL(IARTUBE(JASYS))
            NSHOT=NSHOTS
            CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
     &MILTYPE,IATYPE)
            NA=NKILLS
            ENDIF
            NTR=NTR+1*NA
            ENDIF
110    CONTINUE
            FC=PKIT*100.

            WRITE (*,*) 'ESTIMATED NUMBER OF TARGETS IN PATTERN AREA'
            WRITE (*,*) '# RED TGT, % KILLS'
            WRITE (*,*) NTR,FC

            ENDIF


            IF (IATYPE.EQ.1) THEN
C     EVALUATE POINT TARGET MUNITIONS/SUBMUNITIONS
C     THESE ARE FIRE-AND-FORGET TERMINALLY GUIDED
C     PROJECTILES/SUBMUNITIONS SUCH AS SADARM OR GUIDED MORTAR
C     ROUNDS
C     FIND ALL VEHICLES IN THE BASKET AREA AND EVALUATE THEIR
C      PROBABILITIES OF BEING TARGETED AND KILLED
C     REMEMBER ... UNLESS THERE IS IFF, WHICH THERE ISN'T, FRIENDLY
C     VEHICLES MAY ALSO BE TARGETED, SO FIND ALL POTENTIAL VEHICLES
C     IN BASKED AREA, INCLUDING KILLED VEHICLES, SINCE HOW THE
C     HECK WILL A MUNITION LIKE THIS KNOW THE DIFFERENCE ?
C      THE FINER POINTS OF THESE ISSUES WILL HAVE TO BE MODELED LATER.
C     FOR NOW, THESE FAF GUIDED PROJECTILES ARE BASICALLY DUMB.
C
            NTR=0
C     FIND ALL RED AND BLUE TARGETS IN BASKET AREA, EVEN DEAD ONES
            DO 210 IT=1,NR
            XR=RXYA(IT,1)*1000.
            YR=RXYA(IT,2)*1000.
            R=((XR-XT)**2+(YR-YT)**2)**.5
            IF (R.LE.RADIUS) THEN
C     THIS RED VEHICLE CAN BE TARGETED
            NA=1
            JSYS=IRSYS(IT,1)
            PKIT=ARTPKB(JSYS,KASYS,ITYPE)
            IF (JSYS.GT.10) THEN
C     COUNTER-BATTERY SITUATION
            JTUBES=IRSYS(IT,2)
            JASYS=JSYS-10
            DTUBES=REAL(IARTUBE(JASYS))
            NKILLS=0
            NSHOT=NSUBS
            CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
            &MILTYPE,IATYPE)
            NA=NKILLS
            ENDIF
            NTR=NTR+1*NA
            ENDIF
210    CONTINUE

            NTB=0
            DO 211 IT=1,NB
            XR=BXYA(IT,1)*1000.
            YR=BXYA(IT,2)*1000.
            R=((XR-XT)**2+(YR-YT)**2)**.5
            IF (R.LE.RADIUS) THEN
C     THIS BLUE VEHICLE CAN BE TARGETED
            NA=1
            JSYS=IBSYS(IT,1)
            IF (JSYS.GT.10) THEN
C     COUNTER-BATTERY SITUATION
            JTUBES=IBSYS(IT,2)
            JASYS=JSYS-10
            DTUBES=REAL(IABTUBE(JASYS))
            NKILLS=0
            NSHOT=NSUBS
            CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
            &MILTYPE,IATYPE)
            NA=NKILLS
            ENDIF
            NTB=NTB+1*NA
            ENDIF
211    CONTINUE
            NTOTAL=NTR+NTB

            IF (NTOTAL.LE.0) GO TO 300

C     FIND EXPECTED NUMBER OF SHOTS PER AVAILABLE TARGET
            NSHOTS=NUMR*NTUBES*NSUBS
            SEACH=REAL(NSHOTS)/REAL(NTOTAL)
C     BASED ON POISSON'S DISTRIBUTION, FIND THE PROBABILITY
C     THAT A VEHICLE WILL BE WITHIN RANGE OF ONE SUBMUNITION
            P1=1.-EXP(-SEACH)

            P2=1.-(1.-PKIT)**NSUBS

300    CONTINUE

            FC=P1*P2*100.

            WRITE (*,*) 'ESTIMATED NUMBER OF TARGETS IN PATTERN AREA'
            WRITE (*,*) '# BLUE TGT, # RED TGT, % KILLS RED'
            WRITE (*,*) NTB,NTR,FC


            ENDIF

            ELSE

C     RED

            IF (MILTYPE.EQ.1) THEN
            IATYPE=0
C     EVALUATE MAN-IN-THE-LOOP TERMINAL GUIDANCE ON DESIGNATED
C      TARGET LIST
C     LISTED TARGETS MUST BE IN THE BASKET AREA PLUS CEP ERROR
C     IF THESE TARGETS ARE NO LONGER THERE, ANY SHORTFALL WILL
C     BE MADE UP BY FINDING ANY OTHER ENEMY TARGETS IN AREA.

            NSHOTS=NUMR*NTUBES*NSUBS
            NTB=0
C     FIND ALL BLUE TARGETS IN BASKET AREA
            DO 410 IT=1,NB
            IF (IBSYS(IT,2).LE.0) GO TO 410
            XB=BXYA(IT,1)*1000.
            YB=BXYA(IT,2)*1000.
            R=((XB-XT)**2+(YB-YT)**2)**.5
            IF (R.LE.RADIUS) THEN
C     THIS BLUE VEHICLE CAN BE TARGETED
            NA=1
            JSYS=IBSYS(IT,1)
            PKIT=ARTPKR(JSYS,KASYS,ITYPE)
            IF (JSYS.GT.10) THEN
C     COUNTER-BATTERY SITUATION
            JTUBES=IBSYS(IT,2)
            JASYS=JSYS-10
```

```
            DTUBES=REAL(IABTUBE(JASYS))
            NSHOT=NSHOTS
            CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
     &MILTYPE,IATYPE)
            NA=NKILLS
            ENDIF
            NTB=NTB+1*NA
            ENDIF
410   CONTINUE
            FC=PKIT*100.

            WRITE (*,*) 'ESTIMATED NUMBER OF TARGETS IN PATTERN AREA'
            WRITE (*,*) '# BLUE TGT, % KILLS'
            WRITE (*,*) NTB,FC

            ENDIF


            IF (IATYPE.EQ.1) THEN
C     EVALUATE POINT TARGET MUNITIONS/SUBMUNITIONS
C     THESE ARE FIRE-AND-FORGET TERMINALLY GUIDED
C     PROJECTILES/SUBMUNITIONS SUCH AS SADARM OR GUIDED MORTAR
C     ROUNDS
C     FIND ALL VEHICLES IN THE BASKET AREA AND EVALUATE THEIR
C     PROBABILITIES OF BEING TARGETED AND KILLED
C     REMEMBER ... UNLESS THERE IS IFF, WHICH THERE ISN'T, FRIENDLY
C     VEHICLES MAY ALSO BE TARGETED, SO FIND ALL POTENTIAL VEHICLES
C     IN BASKED AREA, INCLUDING KILLED VEHICLES, SINCE HOW THE
C     HECK WILL A MUNITION LIKE THIS KNOW THE DIFFERENCE ?
C     THE FINER POINTS OF THESE ISSUES WILL HAVE TO BE MODELED LATER.
C     FOR NOW, THESE FAF GUIDED PROJECTILES ARE BASICALLY DUMB.
C
            NTR=0
C     FIND ALL RED AND BLUE TARGETS IN BASKET AREA, EVEN DEAD ONES
            DO 510 IT=1,NR
            XR=RXYA(IT,1)*1000.
            YR=RXYA(IT,2)*1000.
            R=((XR-XT)**2+(YR-YT)**2)**.5
            IF (R.LE.RADIUS) THEN
C     THIS RED VEHICLE CAN BE TARGETED
            NA=1
            JSYS=IRSYS(IT,1)
            IF (JSYS.GT.10) THEN
C     COUNTER-BATTERY SITUATION
            JTUBES=IRSYS(IT,2)
            JASYS=JSYS-10
            DTUBES=REAL(IARTUBE(JASYS))
            NKILLS=0
            NSHOT=NSUBS*NUMR*NTUBES
            CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
     &MILTYPE,IATYPE)
            NA=NKILLS
            ENDIF
            NTR=NTR+1*NA
            ENDIF
510   CONTINUE

            NTB=0
            DO 511 IT=1,NB
            XR=BXYA(IT,1)*1000.
            YR=BXYA(IT,2)*1000.
            R=((XR-XT)**2+(YR-YT)**2)**.5
            IF (R.LE.RADIUS) THEN
C     THIS BLUE VEHICLE CAN BE TARGETED
            NA=1
            JSYS=IBSYS(IT,1)
            PKIT=ARTPKR(JSYS,KASYS,ITYPE)
            IF (JSYS.GT.10) THEN
C     COUNTER-BATTERY SITUATION
            JTUBES=IBSYS(IT,2)
            JASYS=JSYS-10
            DTUBES=REAL(IABTUBE(JASYS))
            NKILLS=0
            NSHOT=NSUBS*NUMR*NTUBES
            CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
     &MILTYPE,IATYPE)
            NA=NKILLS
            ENDIF
            NTB=NTB+1*NA
            ENDIF
511   CONTINUE
            NTOTAL=NTR+NTB

            IF (NTOTAL.LE.0) GO TO 600

C     FIND EXPECTED NUMBER OF SHOTS PER AVAILABLE TARGET
            NSHOTS=NUMR*NTUBES*NSUBS
            SEACH=REAL(NSHOTS)/REAL(NTOTAL)
C     BASED ON POISSON'S DISTRIBUTION, FIND THE PROBABILITY
C     THAT A VEHICLE WILL BE WITHIN RANGE OF AT LEAST ONE SUBMUNITION
            P1=1.-EXP(-SEACH)
C     PROBABILITY OF BEING KILLED IF IN RANGE OF ALL SUBMUNITIONS
C     IN ONE MISSILE
            P2=1.-(1.-PKIT)**NSUBS

600   CONTINUE

            FC=P1*P2*100.

            WRITE (*,*) 'ESTIMATED NUMBER OF TARGETS IN PATTERN AREA'
            WRITE (*,*) '# RED TGT, # BLUE TGT, % KILLS BLUE'
            WRITE (*,*) NTR,NTB,FC
```

^Z

```
            ENDIF
            ENDIF
            RETURN
            END
```