

US EPA ARCHIVE DOCUMENT

APPENDIX B

PREPAR (PREPAR2.FOR) PROGRAM FILE

PREPAR (PREPAR2.FOR) Program File

0 1 2 3 4 5 6 7
12345678901234567890123456789012345678901234567890123456789012

C PROGRAM PREPAR

```

C-----C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C-----C

```

```

C**** NOTE NRU IS NOW BEING SET TO THE NUMBER OF DISTANCES STORED IN
C**** THE POPULATION FILE ( IE NRADS ) , IF ITS A POPULATION RUN
C**** THIS WAS DONE TO ELIMINATE USER ENTERED INPUT IN THE INSTREAM JCL
C**** IF NRU ( THE MAX DISTANCE) IS SPECIFIED IT IS OVERWRITTEN EVEN IF
C**** IT WAS SPECIFIED BY THE USER, IN THE SUBROUTINE GET_MIDPTS_LAT_LON
C**** THE VALUE HAS NEVER BEEN CHANGED FROM THAT OF THE MAX DISTANCE
C**** BEFORE (NRADS IN THE POP FILE) , SO DOING IT THIS WAY SHOULD
C**** ENHANCE RATHER THAN HINDER THE PROCESS OF SUPPLYING THE INFO FOR
C**** A RUN. IF, FOR SOME REASON, THE VALUE OF NRU NEEDS TO BE OTHER
C**** THAN THE MAX DISTANCES SUPPLIED IN THE POP FILE THAN THE SUB-
C**** ROUTINE GET_MIDPTS_LAT_LON SHOULD BE CHANGED TO NO LONGER
C**** SET NRU TO THE NUMBER SPECIFIED IN THE POP FILE. THIS IS THE
C**** ONLY PLACE THE CODE NEEDS TO BE CHANGED. EVERTHING ELSE HAS BEEN
C**** LEFT AS IS. ( J. MCGUE 10/1988).

```

```

CHARACTER*80 DATA
CHARACTER*8 ANMNUC(36),ANMORG(11,36),ORG(11),
- ORGNAM(11),RNAME(280)
CHARACTER*4 DATTYP,FMT,SFMT,TYPES(17)
CHARACTER*2 ENAME(78), STATE
CHARACTER*1 AISOL(36),CLCS(280),ISOL,RESP(280,20)

REAL*4 APH(6),ADIA(6),AVEL(6),AQH(6),AAREA(6),ADIM(6),
- BOUND(20),FB(3),FM(3),FV(3),J0,LID,
- MSUM,PR(7),TG(3),VDTAB(3)

REAL ABSBV1(36),ABSBV2(36),ACFNGA(11,36),ACFNHA(11,36),ACFSBA(36),

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

- ACFSBW(36),ACFSUR(36),AFROG(11,36),AFSBFI(36),AFSBMI(36),
- ALAMRR(36),ALMH20(36),ALMSUR(36),ATDCF(36),ATDCW(36)

REAL DISTANCES(21), LATITUDE, LONGITUDE
INTEGER*4 IDIST(20),INTPA(20,20),LIST(20),OPTION(9)
INTEGER*2 I,J,MAXORG

LOGICAL PTEST,SCAN, RN_RUN

LOGICAL FOODARRAY_GEN_AUTO, USERARRAY

COMMON /OPTIDATA/ OPTION,LIST,LIPO,NNTB,NRTB,NSTB,NTTB,NUTB,NVTB,
- T,TSUBB,GSFAC
COMMON /GRIDDATA/ BOUND,IDIST,NOL,NOU,NRL,NRU,SQSD
COMMON /PLUMDATA/ PR
COMMON /METEDATA/ LIDAI,LID,RR,TA,TAC,TG,Z,ZO,JO,DF
COMMON /PHYSDATA/ APH,ADIA,AVEL,AQH,AAREA,ADIM
COMMON /RADICHAR/ ANMNUC,ANMORG,ORGNAM,AISOL,ISOL
COMMON /AGDTDATA/ FSUBG,FSUBL,FSUBP,FSUBS,F1V,F2V,F3V,
- F1B,F2B,F3B,F3BEFM,F3MLKM,F3VEGM,F1M,F2M,F3M,IMPFIX,LAMW,MSUBB,
- P,QSUBF,RBEF,RMLK,RVEG,R1,R2,TAUBEF,TSUBE1,TSUBE2,TSUBF,TSUBH1,
- TSUBH2,TSUBH3,TSUBH4,TSUBS,VSUBM,YSUBV1,YSUBV2

COMMON /USAGDATA/ DD1,BRTHRT,UV,UM,UF,UL

COMMON /POOLDATA/ DILFAC,USEFAC

COMMON /MODIDATA/ ABSBV1,ABSBV2,ACFNGA,ACFNHA,ACFSBA,ACFSBW,
- ACFSUR,AFROG,AFSBMI,AFSBFI,ALAMRR,ALMH20,ALMSUR,
- ATDCF,ATDCW

COMMON /FILEC/ CLCS,ENAME,ORG,RESP,RNAME

REAL DB, DM, FC

COMMON / FOOD_DENSITIES / DB, DM, FC

EQUIVALENCE (FV(1),F1V),(FV(2),F2V),(FV(3),F3V),(FB(1),F1B),
- (FB(2),F2B),(FB(3),F3B),(FM(1),F1M),(FM(2),F2M),
- (FM(3),F3M)

NAMelist /OPTI/ OPTION,LIST,LIPO,NNTB,NRTB,NSTB,NTTB,NUTB,IMPFIX,
- TSUBB,GSFAC

NAMelist /GRID/ NOL,NOU,NRL,NRU,IDIST,BOUND

NAMelist /PLUM/ PR

NAMelist /METE/ LIDAI,LID,RR,TA,TG,Z,ZO,JO,DF

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
12345678901234567890123456789012345678901234567890123456789012

NAMelist /AGDT/ IMPFIX, F1V, F2V, F3V, F1B, F2B, F3B,
- F1M, F2M, F3M, LAMW, TSubH1, TSubH2, TSubH3, TSubH4,
- TSubE1, TSubE2, YSubV1, YSubV2, FSubP, FSubS, QSubF, TSubF,
- TSubS, FSubG, FSubL, P, TAUBEF, MSUBB, VSubM, R1, R2, FV, FB, FM,
+ FOODARRAY_GEN_AUTO

NAMelist /USAG/ DD1, BRTHRT, UV, UM, UF, UL
NAMelist /POOL/ DILFAC, USEFAC
NAMelist /INIT/ IUNIT, VDTAB, PH, VEL, QH, DIA,
- KFLAG, IFLAG, RD1, RD2, RW1, RW2

DATA TYPES/'OPTI', 'GRID', 'PLUM', 'METE', 'PHYS', 'RADI', 'AG D',
- 'USAG', 'POOL', 'MODI', 'INIT', 'WIND', 'AG A', 'POPU',
- 'DEPO', 'CONC', 'COMM', PTEST/.FALSE./, SCAN/.FALSE./,
- IUSER/5/, INTPA/400*1/, IUNIT/21/, IWIND/34/, NNUCS/0/, NUMST/1/

DATA DISTANCES/ 21 * 0.0 /, RN_RUN / .FALSE. /

DATA STATE / ' ' /

CHARACTER*36 DATE_AND_TIME

DATA DATE_AND_TIME/' //

C*** THE FOLLOWING VAR IS SET TO TRUE WHEN THE ARRAYS SUPPLIED BY THE
C*** USER IS READ IN.

DATA USERARRAY /.FALSE./

C USERARRAY TRUE IF YOU WANT TO SPECIFY YOUR OWN FOOD
C ARRAYS IE TO RUN THE FOOD JOB SEPERATELY
C AND HAVE THE CODE READ IN THE FILE
C SPECIFIED (THIS IS THE OLD WAY PRIOR TO
C 10 1988). FALSE IF YOU DON'T WANT THE CODE
C TO READ A USER SPECIFIED ARRAY.
C DEFAULT - FALSE
C * PLEASE NOTE IF A USERARRAY IS TO BE SUPPLIED (TRUE), SET
C FOODARRAY_GEN_AUTO TO FALSE IN THE NAMelist AGDT. IT
C WILL BE OVERWRITTEN OTHERWISE

C*****
C
C**** NEW VARS ADDED 10 1988, THE CODE HAS BEEN CHANGED TO
C AUTOMATICALLY CALCULATE THE FOOD ARRAYS FOR ALL NON RADON RUNS

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

C HOWEVER IF SOMETIME IN THE FUTURE THIS SHOULD CHANGE THE
C FOLLOWING LOGICAL VARS CAN BE SPECIFIED IN THE AG DATA NAMELIST
C TO OVER RIDE THE DEFAULTS

C
C FOODARRAY_GEN_AUTO TRUE IF YOU WANT THE FOOD ARRAYS TO BE
C AUTOMATICALLY GENERATED WITHIN THE CODE SET
C FALSE IF YOU DO WANT THE ARRAYS GENERATED..
C JUST USE THE DEFAULT VALUES OR READ IN THE
C DATA SPECIFIED IN THE JCL.
C DEFAULT = TRUE

C*****

C
C DATA FOODARRAY_GEN_AUTO /.TRUE. /

C***

C VARIABLE NAMES ARE TAKEN FROM AIRDOS-EPA. WHERE ADDITIONAL SUBSCRIPT
C ARE NEEDED IN THIS ROUTINE, AN 'A' WAS ADDED TO THE BEGINNING OF THE
C VARIABLE NAME.

C

LID-LIDAI

CALL ERRSET(208,256,-1,1)

C*** GET SYSTEM DATE AND TIME FOR REPORTS

C***

CALL MAKE_HEADING(DATE_AND_TIME)

C*** OUTPUT DATE AND TIME TO FILE 26 FOR DARTAB AND TO FILE 27

C*** FOR AIRDOS

WRITE (26,1000) DATE_AND_TIME

WRITE (27,1000) DATE_AND_TIME

1000 FORMAT (A)

C***** READ RADIONUCLIDE FILE *****
CALL FILRED(DUNIT,MAXORG)

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

DO 3 I=1,MAXORG
3 ORGNAM(I)=ORG(I)

C***** WRITE USER INPUT TO A TEMPORARY FILE *****

5 READ(IUSER,1500,END=7)DATA
1500 FORMAT(A80)
 WRITE(21,1500)DATA
 GO TO 5

7 REWIND IUSER
 REWIND 21

C***** FIRST GET INSTREAM DATA FOR NEW REPORT *****

CALL GET_NEW_INPUT (STATE)

C***** READ USER DATA UNTIL VALID DATA TYPE IS FOUND *****

10 READ(21,1100,END=220)DATTYP
1100 FORMAT(A4)
 DO 15 I=1,17
 IF(DATTYP.EQ.TYPES(I)) THEN
 SCAN = .FALSE.
 GO TO 30
 ENDIF
15 CONTINUE
 IF(.NOT.SCAN) WRITE(6,1110) DATTYP
1110 FORMAT(' ...WARNING... INVALID DATA TYPE= ',A4,' READ. ',
- 'SCANNING FOR NEXT VALID DATA TYPE.')

SCAN=.TRUE.
GO TO 10

C***** PROCESS THE DATA THAT FOLLOWS VALID DATA TYPE *****

30 WRITE(6,1120)DATTYP
1120 FORMAT(' READING DATA TYPE= ',A4)
 IF (DATTYP.EQ.'OPTI') THEN
 IMPFIX=999
 READ(21,OPTI)
 IF(IMPFIX.EQ.999.AND.LIPO.EQ.0) IMPFIX=1
 IF(IMPFIX.EQ.999.AND.LIPO.EQ.1) IMPFIX=0
 ELSE IF (DATTYP.EQ.'PLUM') THEN
 READ(21,PLUM)
 PTEST=.TRUE.
 ELSE IF (DATTYP.EQ.'METE') THEN
 LID=9E9
 READ(21,METE)
 IF(LID.NE.9E9) LIDAI=LID+0.5
 LID=LIDAI

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```
ELSE IF (DATTYP.EQ.'PHYS') THEN
  CALL PHYSICAL(NUMST)
ELSE IF (DATTYP.EQ.'RADI') THEN
  CALL NUCLIDES(NUMST,NNUCS,MAXORG, RN_RUN )
ELSE IF (DATTYP.EQ.'WIND') THEN
  CALL RD21(FMT,IUNIT,IWIND)
  SFMT=FMT
  CALL GETWND(SFMT,IUNIT)
ELSE IF (DATTYP.EQ.'POPU') THEN
  CALL POPU(IUNIT,INTPA,LIPO, DISTANCES, LATITUDE, LONGITUDE)
ELSE IF (DATTYP.EQ.'MODI') THEN
  CALL MODS(NNUCS)
ELSE IF (DATTYP.EQ.'AG D') THEN
  READ(21,AGDT)
ELSE IF (DATTYP.EQ.'AG A') THEN
  CALL AGARRAYS( IUNIT, FOODARRAY_GEN_AUTO, USERARRAY )
ELSE IF (DATTYP.EQ.'GRID') THEN
  READ(21,GRID)
ELSE IF (DATTYP.EQ.'USAG') THEN
  READ(21,USAG)
ELSE IF (DATTYP.EQ.'POOL') THEN
  READ(21,POOL)
ELSE IF (DATTYP.EQ.'INIT') THEN
  READ(21,INIT)
ELSE IF (DATTYP.EQ.'DEPO') THEN
  CALL DEPO(IUNIT)
ELSE IF (DATTYP.EQ.'CONC') THEN
  CALL CONC(NNUCS,IUNIT)
ELSE IF (DATTYP.EQ.'COMM') THEN
  CALL COMM
ENDIF
GO TO 10
```

C***** PERFORM MISCELLANEOUS COMPUTATIONS *****

```
220 REWIND 21
    WRITE(6,1600)
1600 FORMAT(' *** END OF USER DATA; BEGIN PROCESSING ***/'1')
    NRU=MINO(NRU,20)
    IF(NRU.EQ.20) GO TO 222
    J1=NRU+1
    DO 221 J=J1,20
221   IDIST(J)=0
222   SQSD=IDIST(NRU)/10.
```

C*** IF NOT A RN-222 RUN AND FOODARRAYS HAVE BEEN CHOSEN TO BE
C*** GENERATED BY THE PROGRAM CALL SUBROUTINE TO GENERATE THE ARRAYS

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

C*** DO NOT GENERATE IF INDIVIDUAL RUN

IF (LIPO .EQ. 1) THEN

IF (.NOT. RN_RUN) THEN

IF (FOODARRAY_GEN_AUTO) THEN

CALL GENERATE_FOODARRAYS (STATE, OPTION(2), DISTANCES)

C ELSE

C

C

C

C

USE DEFAULT VALUES IF FOODARRAYS ARE NOT TO BE AUTOMATICALLY
GENERATED

ENDIF

ENDIF

ENDIF

C*** SET NUTB TO PRINT WORKING LEVELS IF RADON RUN

IF (RN_RUN) THEN

NUTB = 1

ENDIF

T=TSUBB*365.24

TAC=TA

IF(TA.LT.100.) TA=TA+273.16

IF(TAC.GE.100.) TAC=TAC-273.16

BRTHRT=BRTHRT*1.E6/(365.24*24.)

IF(F1V.EQ.0..AND.F2V.EQ.0..AND.F3V.EQ.0.) F3V=1.

VSUM=F1V+F2V+F3V

RVEG=0.

IF(F1V.NE.0.) RVEG=F1V/(F1V+F2V)

F1V=F1V/VSUM

F2V=F2V/VSUM

F3V=F3V/VSUM

F3VEGM=F3V

IF(F1B.EQ.0..AND.F2B.EQ.0..AND.F3B.EQ.0.) F3B=1.

BSUM=F1B+F2B+F3B

RBEF=0.

IF(F1B.NE.0.) RBEF=F1B/(F1B+F2B)

F1B=F1B/BSUM

F2B=F2B/BSUM

F3B=F3B/BSUM

F3BEFM=F3B

IF(F1M.EQ.0..AND.F2M.EQ.0..AND.F3M.EQ.0.) F3M=1.

MSUM=F1M+F2M+F3M

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

RMLK=0.
IF(F1M.NE.0.) RMLK=F1M/(F1M+F2M)
F1M=F1M/MSUM
F2M=F2M/MSUM
F3M=F3M/MSUM
F3MLKM=F3M

C***** CHECK CONDITIONS WHICH MAY CAUSE ERRORS AND WRITE WARNINGS *****
NNUCS=MAX0(NNUCS,1)
OPTION(6)=MIN0(OPTION(6),NNUCS)
NNTB=MIN0(NNTB,NNUCS)

IF(OPTION(5).EQ.1.AND.OPTION(2).EQ.0) WRITE(6,1400)
1400 FORMAT(' OPTION(5)=1 AND OPTION(2)=0. THIS IS',
- ' NOT ALLOWED IN AIRDOS.')

IF(OPTION(5).EQ.1) GO TO 240
DO 230 IN=1,20
IF(LIST(IN).NE.0) GO TO 235
230 CONTINUE
GO TO 240
235 WRITE(6,1410)
1410 FORMAT(' ... WARNING ... OPTION(5) IS NOT SET TO 1. ',
- ' INPUT VALUES FOR LIST WILL BE IGNORED.')

240 IF(OPTION(5).EQ.1) GO TO 255
DO 245 IN=1,20
IF(BOUND(IN).NE.0) GO TO 250
245 CONTINUE
GO TO 255
250 WRITE(6,1140)
1140 FORMAT(' ... WARNING ... OPTION(5) IS NOT SET TO 1. ',
- ' INPUT VALUES FOR BOUND WILL BE IGNORED.')

255 IF(PTEST.AND.OPTION(4).NE.2) WRITE(6,1430)
1430 FORMAT(' ... WARNING ... INPUT PLUME RISE IGNORED IN AIRDOS',
- ' BECAUSE OPTION(4) IS NOT SET TO 2.')

DO 265 IST=1,NUMST
IF(OPTION(7).EQ.0.AND.AAREA(IST).NE.0) WRITE(6,1440)IST
1440 FORMAT(' ... WARNING ... POINT SOURCE SELECTED (OPT(7)=1',
- ' AND SOURCE AREA NOT ZERO FOR SOURCE #',I2)
265 CONTINUE
IF(LIPO.EQ.0.AND.IMPFIX.EQ.0) WRITE(6,1450)
1450 FORMAT(' ... WARNING ... IMPFIX =0 FOR INDIVIDUAL DOSE ',
- ' CALCULATION (LIPO=0)')

IF(LIPO.EQ.1.AND.IMPFIX.EQ.1) WRITE(6,1460)
1460 FORMAT(' ... WARNING ... IMPFIX=1 FOR POPULATION DOSE ',

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

- 'CALCULATION (LIPO-1)')
  IF(IMPFIX.EQ.1) WRITE(6,1470)
1470 FORMAT(' ... NOTE ... IMPFIX-1, FRACTION IMPORTED FOOD FIXED BY ',
- 'FRACTIONS INPUT, USAGE COMPUTATION IN AIRDOS WILL BE IGNORED')
```

```

C***** WRITE AIRDOS-EPA INPUT FILE *****
      CALL FILOUT(NUMST, NNUCS, MAXORG, INTPA)
```

```

C***** WRITE PREPAR REPORT:
C*****      'REPORT' WRITES THE ORIGINAL REPORT.
C*****      'OUTPUT_TO_DARTAB' APPENDS ADDITIONAL DATA TO THE FILE
C*****      WRITTEN TO IN 'GET_NEW_INPUT'. THIS FILE WILL BE
C*****      USED IN DARTAB TO CREATE THE NEW SYNOPSIS REPORT.
C
```

```

      CALL REPORT(SFMT, NUMST, NNUCS, INTPA, DATE_AND_TIME)
      CALL OUTPUT_TO_DARTAB(NNUCS, NUMST, FOODARRAY_GEN_AUTO, USERARRAY,
+      STATE, RN_RUN, ALMSUR, LATITUDE, LONGITUDE)
```

STOP

END

```

C*****
C
C
C           INITIALIZATION
C
C
C
C*****
```

```

C-----C
C
C           BLOCK DATA
C           -----
C  INITIALIZES VARIABLES WITH DEFAULT VALUES TO BE USED WHEN THE
C  DATA ENTERED BY THE USER IS NOT COMPLETE.
C-----C
```

BLOCK DATA

CHARACTER*8 ANMNUC(36), ANMORG(11,36), ORGNAM(11)

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
12345678901234567890123456789012345678901234567890123456789012

CHARACTER*1 AISOL(36), ISOL

REAL AAREA(6), AAMAD(36), AANLAM(36), ABSBV1(36), ABSBV2(36),
- ACFNGA(11, 36), ACFNHA(11, 36), ACFSBA(36),
- ACFSBW(36), ACFSUR(36),
- ACON(36, 20, 20), ADIA(6), ADIM(6),
- AF(36, 5), AFLING(36), AFLINH(36), AFROG(11, 36), AFSBFI(36),
- AFSBMI(36), ALAMRR(36), ALMH20(36), ALMSUR(36),
- APDIA(36), APH(6), AQH(6), ARD1(36), ARD2(36),
- AREL(36, 6), ARHO(36), ARW1(36), ARW2(36),
- ASC(36), ATDCF(36), ATDCW(36), AVD(36), AVEL(6), AVG(36),
- BOUND(20), GCON(36, 20, 20),
- INTFC(20, 20), JO, LAMW, MSUBB, PR(7), TG(3), VDCOEF(20, 20),
- VDTAB(3)

C

INTEGER AI(36, 5), AIFLAG(36), AKFLAG(36), IDIST(20),
- LIST(20), NOBCT(20, 20), NOMCT(20, 20),
- NORG(36), OPTION(9)

INTEGER DECAY_CHAIN_FLAG (36), DAUGHTERS
REAL DB, DM, FC
COMMON / FOOD_DENSITIES / DB, DM, FC

COMMON /OPTIDATA/ OPTION, LIST, LIPO, NNTB, NRTB, NSTB, NTTB, NUTB, NVTB,
- T, TSUBB, GSFAC
COMMON /GRIDDATA/ BOUND, IDIST, NOL, NOU, NRL, NRU, SQSD
COMMON /PLUMDATA/ PR
COMMON /METEDATA/ LIDAI, LID, RR, TA, TAC, TG, Z, ZO, JO, DF
COMMON /PHYSDATA/ APH, ADIA, AVEL, AQH, AAREA, ADIM
COMMON /RADICHAR/ ANMNUC, ANMORG, ORGNAM, AISOL, ISOL
COMMON /RADIDATA/ AAMAD, AANLAM, AF, AFLING, AFLINH, AI, AIFLAG,
- AKFLAG, ANLAMO, APDIA, ARD1, ARD2, AREL, ARHO, ARW1,
- ARW2, ASC, AVD, AVG, CUTOFF, F1, F2, F3, F4, F5,
- I1, I2, I3, I4, I5, IAN, NORG, DAUGHTERS,
- NUMORG, SEQWL, VDTAB, DECAY_CHAIN_FLAG
COMMON /AGTDATA/ FSUBG, FSUBL, FSUBP, FSUBS, F1V, F2V, F3V,
- F1B, F2B, F3B, F3BEFM, F3MLKM, F3VEGM, F1M, F2M, F3M, IMPFIX, LAMW, MSUBB,
- P, QSUBF, RBEF, RMLK, RVEG, R1, R2, TAUBEF, TSUBE1, TSUBE2, TSUBF, TSUBH1,
- TSUBH2, TSUBH3, TSUBH4, TSUBS, VSUBM, YSUBV1, YSUBV2
COMMON /USAGDATA/ DD1, BRTHRT, UV, UM, UF, UL
COMMON /POOLDATA/ DILFAC, USEFAC
COMMON /MODIDATA/ ABSBV1, ABSBV2, ACFNGA, ACFNHA, ACFSBA, ACFSBW,
- ACFSUR, AFROG, AFSBMI, AFSBFI, ALAMRR, ALMH20, ALMSUR,
- ATDCF, ATDCW

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

COMMON /AGRIDATA/ INTFC,NOBCT,NOMCT
COMMON /DEPODATA/ VDCEOF
COMMON /CONCDATA/ ACON,GCON

COMMON/STAR/ PERD(16),UDCAT(7,16),UDAV(7,16),FRAW(7,16),
- WDCS(16,8,7),NS,IS(8),NW,WS(7),UBAR

DATA ANLAMO/1E-2/,PERD/16*.0625/,UDCAT/112*4./,CUTOFF/1E-2/,
- UDAV/112*5./,FRAW/112*.14286/,NS/7/,IS/1,2,3,4,5,6,7,0/,NW/6/,
- WS/0.7716667,2.572222,4.372778,6.945000,9.774444,11.83222,0.0/,
- ANMNUC/36*'NO-NUC'/, NORG/36*11/,JO/5.0/,DF/1E-5/,
- VDTAB/0.,0.018,3.5/,
- ORGNAM/'TOT.BODY','R MAR ','LUNGS','ENDOST','S WALL',
- 'LLI WALL','THYROID','LIVER',
- 'KIDNEYS','TESTES','OVARIES'/,TSUBB/1./,
- OPTION/0,1,0,0,0,0,0,0,1/,LIST /20*0/,
- LIPO /0/,NNTB /0/,NRTB /0/,
- NSTB /2/,NTTB /0/,NUTB /0/,NVTB /0/, NUMORG/11/,
- NOL /1/,NOU /16/,NRL /1/,NRU /20/,
- IDIST /1000,2000,3000,4000,5000,6000,7000,8000,9000,
- 10000,11000,12000,13000,14000,15000,16000,17000,18000,
- 19000,20000/, SQSD/2000./,BOUND /20*0./,PR /7*0./,
- LIDAI /1000/,RR /100./,TA /20./,
- TG /.0728,.1090,.1455/,
- SEQWL /.7/
DATA GSFAC/1./,ISOL /' '/,
- I1/0/,I2/0/,I3/0/,I4/0/,I5/0/,AI/180*0/,
- F1/0./,F2/0./,F3/0./,F4/0./,F5/0./,AF/180*0./,
- IMPFIX/1/, LAMW /.0029/,TSUBH1/0./,
- TSUBH2/2160./,TSUBH3/336./,TSUBH4/336./,TSUBE1/720./,
- TSUBE2/1440./,YSUBV1/.28/,YSUBV2/.716/,FSUBP /.4/,FSUBS /.43/,
- QSUBF /15.6/,TSUBF /2./,TSUBS /20./,FSUBG /1./,
- FSUBL /1./,P /215./,TAUBF/.00381/,MSUBB /200./,VSUBM /11./,
- R1 /.57/,R2 /.2/,BRTHRT/8035:28/,DD1 /.5/,
- UV /176./,UM /112./,UF /85./,UL /18./,DILFAC/1./,
- USEFAC/0./,NOBCT /400*4/,NOMCT /400*2/,INTFC /400*1E+4/,
- AAREA/6*0./,Z/10./,Z0/.01/,UBAR/5./,
- ACON/14400*0/,GCON/14400*0/,VDCEOF/400*0./

DATA DAUGHTERS / 0 /
DATA DECAY_CHAIN_FLAG / 36*0 /
DATA DB / 0.0 /, DM /0.0/, FC /0.0/

C*****
C NOTE THE DEFAULT FOR THE FOOD FRACTIONS F1V,F2V...F3B ETC WERE CHANGED
C TO URBAN 7/88 INSTEAD OF 100%. THIS CAN BE CHANGED AGAIN AT ANY TIME

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

      READ(22,150)TITL,ITYPE
150   FORMAT(A80/A80,T7,I1)
      WRITE(6,155)TITL
155   FORMAT(' ',A80)
      IF(ITYPE.LT.2) UNIT=RAD
      READ(22,160)NORG,IW1
      MAXORG=NORG
160   FORMAT(I5/5X,I1)
      WRITE(6,165)NORG
165   FORMAT(' ',I2,' AIRDOS ORGANS PAIRED WITH DARTAB-RAD RISK ORGANS:')
      DO 4 I=1,NORG
          READ(22,167)ORG(I),DUM
          4   WRITE(6,168)ORG(I),DUM
167   FORMAT(1X,4A8)
168   FORMAT(' ',4A8)
170   FORMAT(3A8)
      IF(IW1.EQ.0) GO TO 7
      READ(22,350)NORGW
      WRITE(6,180)NORGW
180   FORMAT(' ',I2,' ORGANS AND WEIGHTS USED TO PRODUCE WEIGHTED-SUM')
      DO 6 I=1,NORGW
          READ(22,170)DUM
          6   WRITE(6,170)DUM
      GO TO 9
          7   WRITE(6,185)
185   FORMAT(' WEIGHTED-SUM ORGAN NOT INCLUDED IN THIS FILE')
C
          9   I=0
          10  I=I+1
              READ(22,100)ENAME(I),CLCS(I),FBSV1(I),FBSV2(I),
              -   FFSMI(I),FFSFI(I),FLAMSR(I),FLAMH2(I)
100   FORMAT(A2,1X,A1,6E9.2)

```

C***** LAM SUR (FLAMSR) IS IMMEDIATELY SET TO DEFAULT 5.48E-5 *****
 C***** THE ALLRAD FILE CONTENTS MAY BE CHANGED TO THIS LATER *****
 FLAMSR(I) = 5.48E-5

```

      IF(ENAME(I).NE.ZZ) GO TO 10
C
      I=0
15   I=I+1
      READ(22,200,END=35)RNAME(I),INDEX(I),(INDEF(I,J),J=1,2),
      -   FCFSBA(I),FCFSBW(I),FCFSUR(I),FLAMRR(I),FANLAM(I)
200  FORMAT(A8,I2,2I5,5E10.3)
      READ(22,300)(FFROG(J,I),J=1,NORG)
300  FORMAT(7E10.3)

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

350  FORMAT(I5)
      DO 30 J=1,2
      READ(22,350)KIND(I,J)
      KI=KIND(I,J)
      IF(KI.NE.0) GO TO 20
      KI=1
C 1 - INGESTION, 2 - INHALATION
20   DO 30 K=1,KI
      IF(J.EQ.1) READ(22,300)GIABS(I,K,1),
      (FCFNKA(IO,I,K),IO=1,NORG)
      IF(J.EQ.2) READ(22,500)RESP(I,K),SIZE(I,K),GIABS(I,K,2),
      (FCFNHA(IO,I,K),IO=1,NORG)
30   CONTINUE
500  FORMAT(A1,9X,6E10.3/7E10.3)
      GO TO 15
35  MAXNUC=I
      REWIND 22
      RETURN
      END

```

```

C-----C
C          GET_NEW_INPUT          C
C          -----                C
C  READS INSTREAM DATA AND WRITES TO A FILE TO BE USED BY      C
C  DARTAB TO PRODUCE THE SYNOPSIS REPORT.  ADDITIONAL DATA     C
C  IS WRITTEN TO THIS FILE IN THE SUBROUTINE OUTPUT_TO_DARTAB.  C
C-----C

```

SUBROUTINE GET_NEW_INPUT (STATE)

CHARACTER*2 STATE

CHARACTER*80 LINE

C*** READ SYNOPSIS DATA UNTIL '*' THEN READ 3 LINES THEN GET STATE
C*** IT SHOULD BE THE FORTH LINE AFTER *

```

10  CONTINUE
    READ(21,1000) LINE
1000 FORMAT(A80)

    IF ( LINE(1:1) .EQ. '*' ) THEN

```


PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

WRITE(26,1000) LINE

DO 5 I = 1,3
  READ(21,1000) LINE
  IF ( LINE(1:6) .EQ. 'OPTION' ) THEN
    BACKSPACE 21
    WRITE(6,*) ' A BIG BOO ...ERROR IN FACILITY DATA '
    RETURN
  ELSE
    WRITE(26,1000) LINE
  ENDIF
5 CONTINUE
  READ(21,1000) LINE
  STATE = LINE (1:2)
  WRITE(26,1000) LINE
  ELSE
C***** DO NOT OUTPUT FOOD FILE REFERENCE ANYMORE ..CODE DOES IT
C***** AUTOMATICALLY NOW..2/89
    J = INDEX (LINE, 'FOODLIB' )
    IF ( J .EQ. 0 )
+     WRITE(26,1000) LINE
    GO TO 10
  ENDIF

C***  READ SYNOPSIS DATA UNTIL 'OPTION' READ ....THIS SIGNALS THE END
20 CONTINUE
  READ(21,1000) LINE
  IF ( LINE(1:6) .EQ. 'OPTION' ) THEN
    BACKSPACE 21
    RETURN
  ELSE
    WRITE(26,1000) LINE
    GO TO 20
  ENDIF
END

```

```

C-----C
C                                           C
C           PHYSICAL                       C
C           -----                       C
C  READS PHYSICAL SOURCE DATA (DATTYP = 'PHYS').   CALLED BY   C
C  MAIN PROGRAM.                                   C
C-----C

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

SUBROUTINE PHYSICAL (NUMST)

REAL*4 APH(6),ADIA(6),AVEL(6),AQH(6),AAREA(6),ADIM(6),
- PH,DIA,VEL,QH,AREA
COMMON /PHYSDATA/ APH,ADIA,AVEL,AQH,AAREA,ADIM
NAMELIST /PHYS/ PH,DIA,VEL,QH,AREA

DATA PH/1./,DIA/0./,VEL/0./,QH/0./

C***** FIRST READ NUMBER OF SOURCES (PHYS) *****

75 READ(21,*)NUMST
IF(NUMST.GT.6) WRITE(6,1160)NUMST
1160 FORMAT(' ... WARNING ... ONLY 6 SOURCES ARE ALLOWED. ',
- I2,' -NUMBER OF STACKS INPUT.')

IF(NUMST.LE.0) NUMST=1

C***** READ DATA FOR EACH SOURCE *****

DO 80 IN=1,NUMST
SVAREA=AREA
SVPH=PH
SVVEL=VEL
SVQH=QH
SVDIA=DIA
READ(21,PHYS)

C***** SKIP ARRAY ASSIGNMENT IF SOURCE # IS > 6 *****

IF(IN.GT.6) GO TO 77
AAREA(IN)=AREA
APH(IN)=PH
AVEL(IN)=VEL
AQH(IN)=QH
ADIA(IN)=DIA
77 AREA=SVAREA
PH=SVPH
VEL=SVVEL
QH=SVQH
DIA=SVDIA
80 CONTINUE
IF(NUMST.GT.6) NUMST=6
DO 225 I=1,NUMST
225 ADIM(I)=2.*SQRT(AAREA(I)/3.1415927)
RETURN
END

C-----C
C C

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

C                               NUCLIDES                               C
C                               -----                               C
C   READS RADIONUCLIDE DATA (DATTYP = 'RADI').  CALLED BY MAIN   C
C   PROGRAM.                                                            C
C_____C

```

SUBROUTINE NUCLIDES(NUMST, NNUCS, MAXORG, RN_RUN)

C**** THE DEFAULT FOR REL RATES WAS CHANGED FROM 1.0 TO 0.0 8/88 ****

```

CHARACTER*8 ANMNUC(36), ANMORG(11, 36), BLANK, NUC,
- NAMNUC, ORG(11), ORGAN(11), ORGNAM(11), RNAME(280)
CHARACTER*4 UNIT
CHARACTER*2 ENAME(78)
CHARACTER*1 AISOL(36), CLCS(280), ISOL, RESP(280, 20)

LOGICAL RN_RUN, RNFLAG
REAL*8 G, MU, P
REAL*4 AAMAD(36), AANLAM(36), AF(36, 5),
- AF1ING(36), AF1INH(36), APDIA(36),
- ARD1(36), ARD2(36), ARHO(36), ARW1(36), ARW2(36),
- AREL(36, 6), ASC(36), AVD(36), AVG(36), F(5), JO, LID, REL(6),
- SVRD1, SVRD2, SVREL(6), SVRW1, SVRW2, TG(3), VDTAB(3),
- AMAD, RR, VD, VG, SC, PDIA, RHO, A, KS, D, CUTOFF

INTEGER*4 AI(36, 5), AIFLAG(36), AKFLAG(36),
- IFLAG, II(5), KFLAG, NORG(36), ORGID(11), SVI(5), SVIFLG, SVKFLG
INTEGER*2 MAXORG
INTEGER DECAY_CHAIN_FLAG (36), DAUGHTERS

COMMON /FILEC/ CLCS, ENAME, ORG, RESP, RNAME
COMMON /METEDATA/ LIDAI, LID, RR, TA, TAC, TG, Z, ZO, JO, DF
COMMON /RADICHR/ ANMNUC, ANMORG, ORGNAM, AISOL, ISOL
COMMON /RADIDATA/ AAMAD, AANLAM, AF, AF1ING, AF1INH, AI, AIFLAG,
- AKFLAG, ANLAM0, APDIA, ARD1, ARD2, AREL, ARHO, ARW1,
- ARW2, ASC, AVD, AVG, CUTOFF, F1, F2, F3, F4, F5,
- I1, I2, I3, I4, I5, IAN, NORG, DAUGHTERS,
- NUMORG, SEQWL, VDTAB, DECAY_CHAIN_FLAG
COMMON /STAR/ PERD(16), UDCAT(7, 16), UDAV(7, 16), FRAW(7, 16),
- WDCS(16, 8, 7), NS, IS(8), NW, WS(7), UBAR

NAMELIST /RADI/ NAMNUC, RHO, REL, NUMORG, KFLAG, IFLAG, RD1, RD2,
- RW1, RW2, ISOL, AMAD, F1INH, I1, I2, I3, I4, I5, F1, F2, F3, F4, F5, F1ING,
- VG, VD, SC, PDIA, ORGAN, SEQWL, UNIT, CUTOFF, VGO, ANLAM, ANLAM0, IAN,
- NUC, F, II

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```
DATA AMAD/1./,BLANK/'      '/,
- FLING/.2/,FLINH/.2/,IFLAG/0/,KFLAG/0/,NAMNUC/'TYPO      '/,
- RD1/0./,RD2/0./,REL/6*0./,RHO/1./,RW1/0./,RW2/0./,SVI/5*0/,
- SVPDIA/9E9/,SVRHO/9E9/,SVAMAD/9E9/
- P/0.76/,MU/1.7894285D-5/,G/9.80665/
```

```
EQUIVALENCE (NUC,NAMNUC),(F(1),F1),(F(2),F2),(F(3),F3),(F(4),F4),
- (F(5),F5),(II(1),I1),(II(2),I2),(II(3),I3),(II(4),I4),
- (II(5),I5),(CUTOFF,VG0)
```

C***** FIRST READ NUMBER OF NUCLIDES (RADI) *****

```
85 READ(21,*) NNUCS
   IF(NNUCS.LE.0) NNUCS=1
   IF(NNUCS.GT.36) WRITE(6,1170) NNUCS
1170 FORMAT(' ... WARNING ... NNUCS=',I3,'. ONLY THE',
- ' FIRST 36 WILL BE PASSED TO AIRDOS.')
```

C***** READ DATA FOR EACH NUCLIDE *****

```
DO 105 IN=1,NNUCS
  SVKFLG=KFLAG
  SVIFLG=IFLAG
  SVRD1=RD1
  SVRD2=RD2
  SVRW1=RW1
  SVRW2=RW2
  DO 851 J=1,5
    II(J)=0
851   F(J)=9E9
  DO 86 IST=1,NUMST
86   SVREL(IST)=REL(IST)
  ANLAM=9E9
  IAN=0
  VG=9E9
  VD=9E9
  SC=9E9
  PDIA=9E9
  RHO=9E9
  AMAD=9E9
  ISOL=' '
  FLING=9E9
  FLINH=9E9
  ORGAN(1)=BLANK
  READ(21,RADI)
  IF ( NAMNUC .EQ. 'RN-222' ) THEN
    RNFLAG = .TRUE.
  ENDIF
```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

C***** SKIP ARRAY ASSIGNMENT IF NUCLIDE # IS > THAN 36 *****
      IF(IN.GT.36) GO TO 103
      ANMNUC(IN)=NAMNUC
      FAC=1.
      IF(UNIT.EQ.'BQ') FAC=3.7E10
      DO 87 J=1,NUMST
87      AREL(IN,J)=REL(J)/FAC
      NORG(IN)=NUMORG
      IF(ORGAN(1).EQ.BLANK) GO TO 92
      DO 90 IO=1,NUMORG
      DO 88 IOL=1,MAXORG
      IF(ORGAN(IO).EQ.ORGNAM(IOL)) GO TO 89
88      CONTINUE
      WRITE(6,1180)ORGAN(IO),NAMNUC
1180     FORMAT(' ORGAN ',A8,' FOR ',A8,' HAS NO MATCH IN DOSE FILE')
      IOL=1
89      ANMORG(IO,IN)=ORGAN(IO)
90      ORGID(IO)=IOL
      GO TO 95
92      DO 94 J=1,NUMORG
      ORGID(J)=J
94      ANMORG(J,IN)=ORGNAM(J)
95      AKFLAG(IN)=KFLAG
      AIFLAG(IN)=IFLAG
      ARD1(IN)=RD1
      ARD2(IN)=RD2
      ARW1(IN)=RW1
      ARW2(IN)=RW2
      IF(IFLAG.NE.1) ARD1(IN)=0.
      IF(IFLAG.NE.1) ARD2(IN)=0.
      IF(IFLAG.NE.1) ARW1(IN)=0.
      IF(IFLAG.NE.1) ARW2(IN)=0.
      DO 97 J=1,5
      IF(F(J).EQ.9E9) GO TO 97
C***** SET DAUGHTERS FLAG TO TRUE (1) *****
      DAUGHTERS, = 1
      I=SVI(J)
      IF(II(J).EQ.0) GO TO 96
      I=IN+II(J)
96      IF(I.EQ.0) I=IN-1
      IF(I.LT.1.OR.I.EQ.IN.OR.I.GT.NNUCS) WRITE(6,1176) J,I,IN,NAMNUC
1176     FORMAT(' ... WARNING ... I',I1,'-',I2,' FOR IN=',I2,', ',A8)
      SVI(J)=I
      AI(IN,J)=I
      AF(IN,J)=F(J)
97      CONTINUE

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

RTAMAD=AMAD
IF(PDIA.NE.9E9) SVPDIA=PDIA
IF(RHO.NE.9E9) SVRHO=RHO
IF(ISOL.EQ.'*'.AND.AMAD.EQ.9E9) AMAD=0.
IF(ISOL.EQ.'*'.AND.PDIA.EQ.9E9) PDIA=0.
IF(ISOL.EQ.'*'.AND.RHO.EQ.9E9) RHO=0.
IF(AMAD.EQ.9E9) AMAD=SVAMAD
IF(AMAD.LT.0.) PDIA=AMAD
AMAD=AMAX1(0.,AMAD)
IF(PDIA.EQ.9E9) PDIA=SVPDIA
IF(RHO.EQ.9E9) RHO=SVRHO
IF(RHO.EQ.9E9.OR.PDIA.LE.0.) RHO=1.0
IF(ISOL.NE.'*'.AND.RHO.NE.1.0.AND.AMAD.NE.9E9.AND.AMAD.EQ.PDIA)
-   WRITE(6,1175)NAMNUC,RHO
1175 FORMAT(' ... WARNING ... FOR ',A8,' AMAD=PDIA BUT RHO=',
-   F8.4)
CALL GETTAB(IN,NAMNUC,AMAD,ISOL,FLING,FLINH,ORGID,MAXORG)
AISOL(IN)=ISOL
AAMAD(IN)=AMAD
AF1INH(IN)=F1INH
AF1ING(IN)=F1ING
IF(ISOL.NE.'*'.AND.RTAMAD.NE.9E9) SVAMAD=RTAMAD
IF(ISOL.EQ.'*') PDIA=0.
IF(ISOL.EQ.'*') RHO=0.
IF(PDIA.EQ.9E9) PDIA=AMAD/SQRT(RHO)
APDIA(IN)=AMAX1(PDIA,0.)
ARHO(IN)=RHO
C*** SET VD AND VG ACCORDING TO WHETHER THIS NUC IS GAS OR PARTICLE ****

C*** THE FOLLOWING 8 LINES (COMMENTED) OF THE ORIGINAL CODE HAVE BEEN
C*** REPLACED. THE SUBROUTINES SETSC AND SETVD AND NO LONGER USED.
C*** INSTEAD, SC AND VD ARE SET TO THE FOLLOWING DEFAULTS:
C***     VD IS 3.5E-2 IF IODINE, 0.0 IF GASEOUS, ELSE 1.8E-3.
C***     SC IS 0.0 IF GASEOUS, ELSE 1E-7 TIMES RAINFALL RATE.      8/88

C***** ORIGINAL CODE *****
C     IF(PDIA.GT.0.) GO TO 100
C     AVG(IN)=0.0
C     AVD(IN)=VDTAB(1.-PDIA)
C     PDIA=0.
C     GO TO 102
C100  UB=UBAR
C     CALL SETVD(AVD(IN),AVG(IN),PDIA,RHO,Z,ZO,UB)
C102  CALL SETSC(ASC(IN),RR,PDIA,RHO,LID,JO,DF)
C*****

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```
C***** NEW CODE *****
  IF(ISOL.EQ.'*') THEN
    AVD(IN) = 0.0
    AVG(IN) = 0.0
    ASC(IN) = 0.0
  ELSE
    IF(NAMNUC(1:2).EQ.'I-') THEN
      AVD(IN) = 0.035
    ELSE
      AVD(IN) = 0.0018
    ENDIF
    IF(PDIA.GT.0.0) THEN
      D = PDIA * 1D-6
      A = D / 2.0
      KS = 1.0 + 1.D-8 / (P*A) * (6.32 + 2.01 * DEXP(-P*A*2.19D7))
      AVG(IN) = (1.D3 * RHO) * D**2 * G / (18.0*MU) * KS
    ELSE
      AVG(IN) = 0.0
    ENDIF
    ASC(IN) = 1E-7 * RR
  ENDIF
C***** END OF NEW CODE ADDED 8/88 *****
```

```
  IF(VD.NE.9E9) AVD(IN)=VD
  IF(VG.NE.9E9) AVG(IN)=VG
  IF(SC.NE.9E9) ASC(IN)=SC
  IF(ANLAM.NE.9E9) AANLAM(IN)=ANLAM
  IF(IAN.LT.0) AANLAM(IN)=AANLAM(IN+IAN)
  IF(IAN.GT.0) WRITE(6,1185) NAMNUC
1185  FORMAT('0','... WARNING ... FOR ',A8,', ATTEMPT TO ',
-      'FORWARD REFERENCE FOR ANLAM IS NOT ALLOWED')
103  KFLAG=SVKFLG
     IFLAG=SVIFLG
     RD1=SVRD1
     RD2=SVRD2
     RW1=SVRW1
     RW2=SVRW2
     DO 104 IST=1,NUMST
104   REL(IST)=SVREL(IST)
105   CONTINUE
     IF(NNUCS.GT.36) NNUCS=36

  IF (RNFLAG .AND. (NNUCS .EQ. 1) ) THEN
    RN_RUN = .TRUE.
  ELSE
    RN_RUN = .FALSE.
```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

ENDIF

RETURN
 END

```

C-----C
C                                           C
C                               GETWND      C
C                               -----      C
C   READS AIRDOS-EPA FORMATTED WIND DATA, OR READS STAR DATA AND      C
C   CONVERTS IT TO AIRDOS-EPA FORMAT.  THE AVERAGE WIND SPEED (UBAR)      C
C   IS CALCULATED.  CALLED BY THE MAIN PROGRAM.                          C
C-----C
  
```

SUBROUTINE GETWND(SFMT, IWIND)

CHARACTER*4 AIRD, ASTAR, DEFA, FMT, FREE, SFMT, STR, USER
 CHARACTER*80 FORM

REAL FREQ(7,16,7)
 INTEGER*2 I, J, K

COMMON/STAR/ PERD(16),UDCAT(7,16),UDAV(7,16),FRAW(7,16),
 - WDCS(16,8,7),NS,IS(8),NW,WS(7),UBAR
 NAMELIST/STAR/ NS,IS,NW,WS
 DATA AIRD/'AIRD'/,FREE/'FREE'/,USER/'USER'/,DEFA/'DEFA'/,
 - STR/' &ST'/,ASTAR/'STAR'/

IF(SFMT.EQ.AIRD) GO TO 110
 IF(SFMT.EQ.ASTAR) GO TO 3
 BACKSPACE 21
 GO TO 110 ,

```

C READ STAR DATA
3 READ(21,1220)FMT
  BACKSPACE 21
  IF(FMT.EQ.STR) READ(21,STAR)
  READ(21,1220)FMT
  IF(FMT.EQ.USER) READ(21,1220)FORM
  IFMT=0
  IF(FMT.EQ.FREE) IFMT=1
  IF(FMT.EQ.USER) IFMT=2
  IF(FMT.EQ.DEFA) IFMT=3
  
```


PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```
      IF(IFMT.EQ.0) BACKSPACE 21
C  CLEAR ARRAY WDCS.
      DO 5 I=1,16
        DO 5 J=1,8
          DO 5 K=1,7
            5 WDCS(I,J,K)=0.
C  READ JOINT WIND FREQUENCY DATA. WIND IS FROM DIRECTION I.
      DO 10 J=1,NS
        DO 10 I=1,16
          GO TO (7,8,6),IFMT
        6  READ(IWIND,1000,END=15)(WDCS(I,J,K),K=1,NW)
          GO TO 10
        7  READ(IWIND,*,END=15)(WDCS(I,J,K),K=1,NW)
          GO TO 10
        8  READ(IWIND,FORM,END=15)(WDCS(I,J,K),K=1,NW)
    10  CONTINUE
1000  FORMAT(T8,6F7.5)
      GO TO 16
    15  NS=J-1
      WRITE(6,1100) NS
1100  FORMAT(1X,'... WARNING ... STAR DATA CONTAIN ONLY ',I1,
-      ' CLASSES. ')
C  CONVERT STAR DATA TO AIRDOS-EPA ARRAYS
C  DEFAULT VALUES OF WS CORRESPOND TO 1.5, 5.0, 8.5, 13.5, 19.0,
C  AND 23.0 KNOTS RESPECTIVELY. NOTE THAT 1 NAUT. MILE = 1852 M,
C  WHILE 1 STAT. MILE = 1609.344 M.
C  DEFAULT VALUES OF IS CORRESPOND TO STABILITY CATEGORIES A-G.
    16  SUM=0.
      DO 18 I=1,16
        DO 18 J=1,NS
          DO 18 K=1,NW
    18  SUM=SUM+WDCS(I,J,K)
      DO 20 ID=1,16
        PERD(ID)=0.
          DO 20 IC=1,7
            FRAW(IC,ID)=0.
            UDAV(IC,ID)=0.
            UDCAT(IC,ID)=0.
          DO 20 IW=1,NW
    20  FREQ(IC,ID,IW)=0.
C  COMPUTE AIRDOS-EPA WIND FREQUENCIES, FREQ. WIND IS TOWARD
C  DIRECTION ID. DIRECTIONS ARE COUNTER CLOCKWISE FROM NORTH.
      DO 30 I=1,16
        ID=MOD(25-I,16)+1
        DO 30 J=1,NS
          IC=IS(J)
```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

      DO 30 K=1,NW
      IW=K
30  FREQ(IC, ID, IW)=FREQ(IC, ID, IW)+WDCS(I, J, K)/SUM
      DO 50 ID=1,16
      DO 50 IC=1,7
      DO 40 IW=1,NW
40  FRAW(IC, ID)=FRAW(IC, ID)+FREQ(IC, ID, IW)
50  PERD(ID)=PERD(ID)+FRAW(IC, ID)
      DO 60 ID=1,16
      DO 60 IC=1,7
      SUM=FRAW(IC, ID)
      IF(SUM.EQ.0.) GO TO 57
      DO 55 IW=1,NW
      UDAV(IC, ID)=UDAV(IC, ID)+FREQ(IC, ID, IW)*WS(IW)/SUM
      UDCAT(IC, ID)=UDCAT(IC, ID)+FREQ(IC, ID, IW)/(WS(IW)*SUM)
55  CONTINUE
57  CONTINUE
60  CONTINUE
C  MAKE UDCAT THE HARMONIC WIND SPEED.
      DO 70 ID=1,16
      DO 70 IC=1,7
      IF(UDCAT(IC, ID).EQ.0.) GO TO 70
      UDCAT(IC, ID)=1.0/UDCAT(IC, ID)
70  CONTINUE
C  CONDITION FRAW WITH PERD.
      DO 80 ID=1,16
      SUM=PERD(ID)
      IF(SUM.EQ. 0.0 ) GO TO 80
      DO 75 IC=1,7
      FRAW(IC, ID)=FRAW(IC, ID)/SUM
75  CONTINUE
80  CONTINUE
      GO TO 140

C
C  READ AIRDOS-EPA ARRAYS
110 READ(21,1220)FMT
      IF(FMT.EQ.FREE) GO TO 120
      IF(FMT.EQ.USER) GO TO 130
      IF(FMT.NE.DEFA) BACKSPACE 21
      READ(IWIND,1200)PERD
      READ(IWIND,1200)((UDCAT(IC, ID), ID=1,16), IC=1,7)
      READ(IWIND,1200)((UDAV(IC, ID), ID=1,16), IC=1,7)
1200  FORMAT(16F5.0)
      READ(IWIND,1210)FRAW
1210  FORMAT(7F10.0)
      GO TO 140

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

120  READ(IWIND,*)PERD
      READ(IWIND,*)((UDCAT(IC, ID), ID-1, 16), IC-1, 7)
      READ(IWIND,*)((UDAV(IC, ID), ID-1, 16), IC-1, 7)
      READ(IWIND,*)FRAW
      GO TO 140
130  READ(21, 1220)FORM
1220  FORMAT(20A4)
      READ(IWIND, FORM)PERD
      READ(IWIND, FORM)((UDCAT(IC, ID), ID-1, 16), IC-1, 7)
      READ(IWIND, FORM)((UDAV(IC, ID), ID-1, 16), IC-1, 7)
      READ(21, 1220)FORM
      READ(IWIND, FORM)FRAW

C
C  COMPUTE UBAR, THE AVERAGE WIND SPEED FOR COMPUTATION OF VD
140  UBAR=0.
      DO 160 ID=1, 16
          UB=0.
          DO 150 IC=1, 7
150      UB=UB+UDAV(IC, ID)*FRAW(IC, ID)
160      UBAR=UBAR+UB*PERD(ID)
      RETURN
      END

```

```

C-----C
C                                           C
C                               POPU          C
C                               ----          C
C  READS POPULATION DATA (DATTYP = 'POPU').  CALLED BY MAIN  C
C  PROGRAM.                                           C
C-----C

```

```

SUBROUTINE POPU(IUNIT, INTPA, LIPO, DISTANCES, LATITUDE, LONGITUDE)

CHARACTER*80 FORM
CHARACTER*4  DEFA, FMT, FREE, USER

REAL DISTANCES (21), LATITUDE, LONGITUDE

INTEGER*4  INTPA(20, 20), IFILE

DATA DEFA/'DEFA'/, FREE/'FREE'/, USER/'USER'/

IFILE=21

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

155 CALL RD21(FMT,IUNIT,IFILE)
159 IF(FMT.EQ.FREE) GO TO 160
    IF(FMT.EQ.USER) GO TO 165
    IF(FMT.NE.DEFA) BACKSPACE 21
    READ(IUNIT,1200)INTPA
1200  FORMAT(10I6)
    GO TO 166
160  READ(IUNIT,*)INTPA
    GO TO 166
165  READ(21,1220)FORM
1220  FORMAT(A80)
    READ(IUNIT,FORM)INTPA
166  CONTINUE
    IF (LIPO.EQ.1) CALL GET_MIDPTS_LAT_LON (IUNIT,DISTANCES, LATITUDE,
+      LONGITUDE)
    RETURN
    END

```

```

C-----C
C                                           C
C                               MODS          C
C                               ----          C
C  READS MODIFICATIONS OF NUCLIDE DATA (DATTYP = 'MODI').  C
C  CALLED BY MAIN PROGRAM.                               C
C-----C

```

SUBROUTINE MODS(NNUCS)

```

CHARACTER*8 ANMNUC(36),ANMORG(11,36),BLANK,NUC,
- NAMNUC,ORGNAM(11)
CHARACTER*1 AISOL(36),ISOL

REAL*4 AAMAD(36),AANLAM(36),
- ABSBV1(36),ABSBV2(36),ACFNGA(11,36),ACFNHA(11,36),
- ACFSBA(36),ACFSBW(36),ACFSUR(36),AF(36,5),AFROG(11,36),
- AFSBFI(36),AFSBI(36),AFLING(36),
- AF1INH(36),ALAMRR(36),ALMH2O(36),ALMSUR(36),APDIA(36),
- ARD1(36),ARD2(36),ARHO(36),ARW1(36),ARW2(36),AREL(36,6),
- ASC(36),ATDCF(36),ATDCW(36),AVD(36),AVG(36),
- CFINGA(11),CFINHA(11),FROG(11),LAMH2O,LAMRR,LAMSUR,
- VDTAB(3)

INTEGER*4 AI(36,5),AIFLAG(36),AKFLAG(36),NORG(36)
INTEGER  DECAY_CHAIN_FLAG (36), DAUGHTERS

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

LOGICAL MTEST

```
COMMON /RADICHAR/ ANMNUC,ANMORG,ORGNAM,AISOL,ISOL
COMMON /RADIDATA/ AAMAD,AANLAM,AF,AFLING,AFLINH,AI,AIFLAG,
- AKFLAG,ANLAMO,APDIA,ARD1,ARD2,AREL,ARHO,ARW1,
- ARW2,ASC,AVD,AVG,CUTOFF,F1,F2,F3,F4,F5,
- I1,I2,I3,I4,I5,IAN,NORG,DAUGHTERS,
- NUMORG,SEQWL,VDTAB, DECAY_CHAIN_FLAG
COMMON /MODIDATA/ ABSBV1,ABSBV2,ACFNGA,ACFNHA,ACFSBA,ACFSBW,
- ACFSUR,AFROG,AFSBMI,AFSBFI,ALAMRR,ALMH2O,ALMSUR,
- ATDCF,ATDCW
```

```
NAMELIST /MODI/ NAMNUC,ANLAM,LAMRR,CFSBA,CFSBW,CFSUR,TDCF,
- TDCW,FROG,FSUBMI,FSUBFI,BSUBV1,BSUBV2,LAMSUR,LAMH2O,CFINHA,
- CFINGA,FLINH,FLING,ISOL,AMAD,PDIA,RHO,IAN,NUC,VD,VG,SC
```

DATA BLANK/' '/

EQUIVALENCE (NUC,NAMNUC)

```
125 READ(21,*)NMOD
IF(NMOD.EQ.0) RETURN
DO 130 IN=1,NMOD
  NAMNUC=BLANK
  FLING=9E9
  ISOL=BLANK
  AMAD=9E9
  PDIA=9E9
  RHO=9E9
  FLINH=9E9
  VD=9E9
  VG=9E9
  SC=9E9
  ANLAM=9E9
  IAN=0
  LAMRR=9E9
  CFSBA=9E9
  CFSBW=9E9
  CFSUR=9E9
  TDCF=9E9
  TDCW=9E9
  FSUBMI=9E9
  FSUBFI=9E9
  BSUBV1=9E9
  BSUBV2=9E9
  LAMSUR=9E9
```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

LAMH20=9E9
DO 126 I=1,11
  CFINHA(I)=9E9
  CFINGA(I)=9E9
126   FROG(I)=9E9
      READ(21,MODI)
      MTEST=.FALSE.
      DO 128 J=1,NNUCS
        IF(NAMNUC.NE.BLANK.AND.NAMNUC.NE.ANMNUC(J)) GO TO 128
        IF(FLING.NE.9E9.AND.FLING.NE.AFLING(J)) GO TO 128
        IF(ISOL.NE.BLANK.AND.ISOL.NE.AISOL(J)) GO TO 128
        IF(AMAD.NE.9E9.AND.AMAD.NE.AAMAD(J)) GO TO 128
        IF(PDIA.NE.9E9.AND.PDIA.NE.APDIA(J)) GO TO 128
        IF(RHO.NE.9E9.AND.RHO.NE.ARHO(J)) GO TO 128
        MTEST=.TRUE.
        IF(VD.NE.9E9) AVD(J)=VD
        IF(FLINH.NE.9E9) AFLINH(J)=FLINH
        IF(VG.NE.9E9) AVG(J)=VG
        IF(SC.NE.9E9) ASC(J)=SC
        IF(ANLAM.NE.9E9) AANLAM(J)=ANLAM
        IF(IAN.NE.0) AANLAM(J)=AANLAM(J+IAN)
        IF(LAMRR.NE.9E9) ALAMRR(J)=LAMRR
        IF(CFSBA.NE.9E9) ACFSBA(J)=CFSBA
        IF(CFSBW.NE.9E9) ACFSBW(J)=CFSBW
        IF(CFSUR.NE.9E9) ACFSUR(J)=CFSUR
        IF(TDCF.NE.9E9) ATDCF(J)=TDCF
        IF(TDCW.NE.9E9) ATDCW(J)=TDCW
        IF(FSUBMI.NE.9E9) AFSBMI(J)=FSUBMI
        IF(FSUBFI.NE.9E9) AFSBFI(J)=FSUBFI
        IF(BSUBV1.NE.9E9) ABSBV1(J)=BSUBV1
        IF(BSUBV2.NE.9E9) ABSBV2(J)=BSUBV2
        IF(LAMSUR.NE.9E9) ALMSUR(J)=LAMSUR
        IF(LAMH20.NE.9E9) ALMH20(J)=LAMH20
      DO 127 I=1,11
        IF(CFINHA(I).NE.9E9) ACFNHA(I,J)=CFINHA(I)
        IF(CFINGA(I).NE.9E9) ACFNGA(I,J)=CFINGA(I)
        IF(FROG(I).NE.9E9) AFROG(I,J)=FROG(I)
127   CONTINUE
128   CONTINUE
      IF(.NOT.MTEST) WRITE(6,1196) IN,NAMNUC
1196  FORMAT(' ... WARNING ... ,MODIFICATION',I3,' FOR NUCLIDE:',A8,
           ' NOT DONE')
130   CONTINUE
      RETURN
      END

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
12345678901234567890123456789012345678901234567890123456789012

```
C-----C
C                                           C
C           AGARRAYS                       C
C           -----                       C
C READS AGRICULTURAL ARRAYS (DATTYP = 'AG A'). CALLED BY MAIN C
C PROGRAM.                                C
C-----C
```

SUBROUTINE AGARRAYS(IUNIT, FOODARRAY_GEN_AUTO, USERARRAY)

C***** ALL ARRAYS CAN BE ENTERED VIA FREE FORMAT, *****
C***** DEFAULT FORMAT, OR USER-SUPPLIED FORMAT.*****

LOGICAL FOODARRAY_GEN_AUTO, USERARRAY

CHARACTER*80 FORM
CHARACTER*4 DEFA,FMT,FREE,USER

REAL*4 INTFC(20,20)

INTEGER*4 NOBCT(20,20),NOMCT(20,20), IFILE

COMMON /AGRIDATA/ INTFC,NOBCT,NOMCT

DATA DEFA/'DEFA'//,FREE/'FREE'//,USER/'USER'//

IFILE=21

```
140 CALL RD21(FMT,IUNIT,IFILE)
C*** CHECK IF DATA IS REDUDANT IE IF CODE HAS BEEN CHANGED SINCE THE
C*** JCL WAS MADE TO AUTOMATICALLY CALCULATE FOOD ARRAYS AND THE USER
C*** DOES NOT WANT TO SPECIFY THEIR OWN ARRAY
```

```
IF ( .NOT. FOODARRAY_GEN_AUTO ) THEN
C*** READ THE INFO
    USERARRAY = .TRUE.
144 IF(FMT.EQ.FREE) GO TO 145
    IF(FMT.EQ.USER) GO TO 150
    IF(FMT.NE.DEFA) BACKSPACE 21
    READ(IUNIT,1200)NOBCT
```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

1200      READ(IUNIT,1200)NOMCT
          FORMAT(10I6)
1210      READ(IUNIT,1210)INTFC
          FORMAT(8E10.4)
          RETURN
145       READ(IUNIT,*)NOBCT
          READ(IUNIT,*)NOMCT
          READ(IUNIT,*)INTFC
          RETURN
150       READ(21,1220)FORM
1220      FORMAT(A80)
          READ(IUNIT,FORM)NOBCT
          READ(IUNIT,FORM)NOMCT
          READ(21,1220)FORM
          READ(IUNIT,FORM)INTFC
ELSE
C***     DON'T READ THE ARRAY .. JUST READ THE REST OF THE JCL INTSTURCS
          USERARRAY = .FALSE.
          IF ( FMT. EQ. USER ) THEN
              READ(21,1220)FORM
              READ(21,1220)FORM
          ENDIF
          ENDIF

          RETURN
          END

```

```

C-----C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C                                     C
C-----C

```

```

SUBROUTINE DEPO(IUNIT)

CHARACTER*80 FORM
CHARACTER*4  DEFA, FMT, FREE, USER

REAL*4  VDCOEF(20,20)

INTEGER  IFILE
COMMON /DEPODATA/ VDCOEF

```


PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

DATA DEFA/'DEFA'/,FREE/'FREE'/,USER/'USER'/

IFILE-21

185 CALL RD21(FMT,IUNIT,IFILE)
189 IF(FMT.EQ.FREE) GO TO 190
    IF(FMT.EQ.USER) GO TO 195
    IF(FMT.NE.DEFA) BACKSPACE 21
    READ(IUNIT,1210)VDCOEF
1210 FORMAT(8E10.4)
    RETURN
190 READ(IUNIT,*)VDCOEF
    RETURN
195 READ(21,1220)FORM
1220 FORMAT(A80)
    READ(IUNIT,FORM)VDCOEF
    RETURN
END

```

```

C-----C
C                                           C
C                                           C
C                CONC                       C
C                ----                       C
C  READS CONCENTRATIONS USED IN AIRDOS-EPA ROUTINE DIRECT   C
C  (DATYYP = 'CONC'). CALLED BY MAIN PROGRAM.               C
C-----C

```

```

SUBROUTINE CONC(NNUCS,IUNIT)

CHARACTER*80 FORM
CHARACTER*4 DEFA,FMT,FREE,USER

REAL*4 ACON(36,20,20),GCON(36,20,20)
INTEGER IFILE

COMMON /CONCDATA/ ACON,GCON

DATA DEFA/'DEFA'/,FREE/'FREE'/,USER/'USER'/

IFILE-21

200 CALL RD21(FMT,IUNIT,IFILE)
203 NN=NNUCS
    IF(NN.EQ.0) NN=36

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

      IF(FMT.EQ.FREE) GO TO 206
      IF(FMT.EQ.USER) GO TO 210
      IF(FMT.NE.DEFA) BACKSPACE 21
      DO 204 IN=1,NN
        READ(IUNIT,1210,END=214)((ACON(IN,I,J),J=1,20),I=1,20)
204    READ(IUNIT,1210)((GCON(IN,I,J),J=1,20),I=1,20)
1210  FORMAT(8E10.4)
      RETURN
206    DO 208 IN=1,NN
        READ(IUNIT,*,END=214)((ACON(IN,I,J),J=1,20),I=1,20)
208    READ(IUNIT,*)((GCON(IN,I,J),J=1,20),I=1,20)
      RETURN
210    READ(21,1220)FORM
1220  FORMAT(A80)
      DO 212 IN=1,NN
        READ(IUNIT,FORM,END=214)((ACON(IN,I,J),J=1,20),I=1,20)
212    READ(IUNIT,FORM)((GCON(IN,I,J),J=1,20),I=1,20)
      RETURN
214    IN=IN-1
      WRITE(6,1230) IN
1230  FORMAT(' ... WARNING ... ACON, GCON INPUT FOR ONLY',
-      I3,' NUCLIDES')
      RETURN
      END

```

```

C-----C
C
C          COMM
C          ----
C  CALLED BY MAIN PROGRAM (DATYP = 'COMM').
C  READS COMMENTS TO BE PRINTED IN SYNOPSIS REPORT.  ALWAYS
C  WRITES 2 COMMENT LINES TO FILE.  WRITES BLANK LINE(S) IF THERE
C  ARE LESS THAN 2 LINES IN INSTREAM DATA.  IGNORES EXTRA LINES
C  IF THERE ARE MORE THAN 2.
C
C-----C

```

SUBROUTINE COMM

```

CHARACTER*80 COMMENT,BLANKS
INTEGER*2    COUNT

```

```

DATA BLANKS/'
+          '/

```

```

COUNT = 0

```

```

100 READ(21,1000)COMMENT

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```
1000 FORMAT(A80)
      IF(COMMENT.EQ.'*') THEN
110   IF (COUNT .LT. 2) THEN
        WRITE(26,1000)BLANKS
        COUNT = COUNT + 1
      ELSE
        RETURN
      ENDIF
      GO TO 110
    ENDIF
    IF (COUNT .LT. 2) WRITE(26,1000)COMMENT
    COUNT = COUNT + 1
    GO TO 100
  END
```

```
C-----
C
C           GENERATE FOOD ARRAYS
C
C THE FOLLOWING SUBROUTINE WAS ADDED TO GENERATE FOOD ARRAYS
C AUTOMATICALLY DEPENDING ON THE STATE SPECIFIED IN THE JCL
C J. MCGUE 10 1988
C-----
```

```
      SUBROUTINE GENERATE_FOODARRAYS ( STATE, CIRCLE, DISTANCES )
C
C
C   VARIABLES PASSED
C
C   INTEGER CIRCLE
C   CHARACTER*2 STATE
C   REAL DISTANCES (21)
C
C   GLOBAL VARIABLES
C
C   REAL DB, DM, FC
C   COMMON / FOOD_DENSITIES / DB, DM, FC
C
C   LOCAL VARIABLES
C
```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

C *** STATE_INDEX WILL POINT TO THE LOCATION OF THE CORRECT STATE GIVEN
 C *** AND THEREFORE TO THE VALUES OF THE CORRECT DB, DM, FC

INTEGER NUM_STATES, STATE_INDEX

CHARACTER*2 STATE_ABBREVIATIONS(48)

C INITIALIZATION
 C

DATA NUM_STATES / 48 /

DATA STATE_ABBREVIATIONS / 'AL', 'AR', 'AZ', 'CA', 'CO', 'CT',
 + 'DE', 'FL', 'GA', 'IA', 'ID', 'IL',
 + 'IN', 'KS', 'KY', 'LA', 'MA', 'MD',
 + 'ME', 'MI', 'MN', 'MO', 'MS', 'MT',
 + 'NC', 'ND', 'NE', 'NH', 'NJ', 'NM',
 + 'NV', 'NY', 'OH', 'OK', 'OR', 'PA',
 + 'RI', 'SC', 'SD', 'TN', 'TX', 'UT',
 + 'VA', 'VT', 'WA', 'WI', 'WV', 'WY' /

C*** FIND STATE IN STATE TABLE AND GET VALUES FOR DB,DM, AND FC
 C*** ACCORDING TO STATE SPECIFIED. THEN CALCULATE THE FOOD
 C*** DISTRIBUTIONS FOR BEEF, MILK, AND VEGETABLES

CALL BIN_SEARCH (STATE_ABBREVIATIONS, STATE_INDEX, NUM_STATES,
 + 1, STATE)

IF (STATE_INDEX .NE. 0) THEN

CALL GET_DB_DM_FC (DB, DM, FC, STATE_INDEX)
 CALL CALC_FOOD_DISTRIBUTIONS (DB, DM, FC, CIRCLE , DISTANCES)

ELSE

C*** SET THE STATE TO AN ERROR 'ZZ'

STATE = 'ZZ'

ENDIF

RETURN

END

C _____ C
 C _____ C
 C GETTAB C

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```
C
C           -----
C   CALLED BY THE SUBROUTINE 'NUCLIDES'.  GETTAB SELECTS THE
C   APPROPRIATE DATA FOR THE GIVEN NUCLIDE NAME FROM THE
C   NUCLIDE FILE (ALLRAD).
C
C
C-----C
```

SUBROUTINE GETTAB(I,NAMNUC,AMAD,ISOL,F1ING,F1INH,ORGID,MAXORG)

CHARACTER*8 NAMNUC,ORG(11),RNAME(280)

CHARACTER*3 H3

CHARACTER*2 ENAME(78)

CHARACTER*1 BLANK,CLCS(280),ISOL,RESP(280,20),STAR

REAL*4 AAMAD(36),AANLAM(36),ABSBV1(36),ABSBV2(36),
- ACFNGA(11,36),ACFNHA(11,36),ACFSBA(36),ACFSBW(36),
- ACFSUR(36),AF(36,5),AFROG(11,36),AFSBFI(36),AFSBMI(36),
- AF1ING(36),AF1INH(36),
- ALAMRR(36),ALMH2O(36),ALMSUR(36),APDIA(36),ARD1(36),ARD2(36),
- AREL(36,6),ARHO(36),ARW1(36),ARW2(36),ASC(36),AVD(36),AVG(36),
- ATDCF(36),ATDCW(36),VDCOEF(20,20),VDTAB(3)

INTEGER*4 AI(36,5),AIFLAG(36),AKFLAG(36),EIND,NORG(36),ORGID(11),
- RIND

INTEGER*2 MAXNUC,MAXORG

INTEGER DECAY_CHAIN_FLAG(36), DAUGHTERS

COMMON /RADIDATA/ AAMAD,AANLAM,AF,AF1ING,AF1INH,AI,AIFLAG,
- AKFLAG,ANLAMO,APDIA,ARD1,ARD2,AREL,ARHO,ARW1,
- ARW2,ASC,AVD,AVG,CUTOFF,F1,F2,F3,F4,F5,
- I1,I2,I3,I4,I5,IAN,NORG,DAUGHTERS,
- NUMORG,SEQWL,VDTAB, DECAY_CHAIN_FLAG

COMMON /MODIDATA/ ABSBV1,ABSBV2,ACFNGA,ACFNHA,ACFSBA,ACFSBW,
- ACFSUR,AFROG,AFSBMI,AFSBFI,ALAMRR,ALMH2O,ALMSUR,
- ATDCF,ATDCW

COMMON /DEPODATA/ VDCOEF

COMMON /FILEC/ CLCS,ENAME,ORG,RESP,RNAME

COMMON /FILE/ FANLAM(280),FLAMRR(280),FCFSBA(280),FCFSBW(280),
- FCFSUR(280),FFROG(11,280),INDEX(280),
- FFSMI(78),FFSFI(78),FBSV1(78),FBSV2(78),FLAMSR(78),
- FLAMH2(78),FCFNHA(11,280,20),FCFNGA(11,280,20),
- GIABS(280,20,2),SIZE(280,20),KIND(280,2),
- INDEF(280,2),MAXNUC

DATA H3/'H-3'/,BLANK/' '/,STAR/'*'/

C

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

C THE LIST OF NUCLIDE NAMES IS SEARCHED TO OBTAIN THE INDEX OF
C THE RADIONUCLIDE. IF THE NAME IS NOT FOUND, DUMMY VALUES
C ARE SELECTED AND A WARNING MESSAGE IS PRINTED

C

 LASTR=0
 MIN=1
 MAX=MAXNUC

C

C THE INDEX NEEDED FOR THE ELEMENT-SPECIFIC DATA IS KEPT IN THE
C RADIONUCLIDE FILE IN ARRAY INDEX.

C

10 RIND=(MIN+MAX)/2
 IF(RIND.EQ.LASTR) GO TO 30
 LASTR=RIND
 IF(RNAME(RIND).EQ.NAMNUC) GO TO 40
 IF(RNAME(RIND).GT.NAMNUC) GO TO 20
 MIN=RIND
 GO TO 10
20 MAX=RIND
 GOTO 10
30 WRITE(6,100)NAMNUC
100 FORMAT(' ... WARNING ... NUCLIDE NAME ',A8,' NOT FOUND IN',
- ' DATA BASE, DUMMY VALUES USED FOR THE NUCLIDE')
 RIND=MAXNUC

C

C SET VALUES FOR THIS NUCLIDE, EXCEPT INTERNAL DF'S

40 EIND=INDEX(RIND)
 AANLAM(I)=FANLAM(RIND)
 ALAMRR(I)=FLAMRR(RIND)
 ACFSBA(I)=FCFSBA(RIND)
 ACFSBW(I)=FCFSBW(RIND)
 ACFSUR(I)=FCFSUR(RIND)
 AFSBMI(I)=FFSMI(EIND)
 AFSBFI(I)=FFSFI(EIND)
 ABSBV1(I)=FBSV1(EIND)
 ABSBV2(I)=FBSV2(EIND)
 ALMSUR(I)=FLAMSR(EIND)
 ALMH2O(I)=FLAMH2(EIND)
 DO 42 J=1,MAXORG
42 AFROG(J,I)=FFROG(ORGID(J),RIND)
 IF(NAMNUC.EQ.H3) GO TO 44
 ATDCF(I)=0.
 ATDCW(I)=0.
 GO TO 46
44 ATDCF(I)=6.18
 ATDCW(I)=0.057

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```
C
C SELECT INTERNAL DOSE FACTORS BY SIZE, GIABS, AND RESP. CLASS
C USE DEFAULT IF THESE VALUES ARE NOT FOUND
C
C DO INHALATION FACTORS FIRST
46 KMAX=KIND(RIND,2)
   KINH=INDEF(RIND,2)
   IF(ISOL.NE.BLANK.OR.RIND.EQ.MAXNUC) GO TO 47
   ISOL=RESP(RIND,KINH)
   IF(ISOL.NE.CLCS(EIND)) WRITE(6,110) NAMNUC,ISOL,CLCS(EIND)
110  FORMAT(' ... WARNING ... DEFAULT CLASS SET FOR ',A8,
-      ' IS ',A1,' NOT ',A1)
   IF(ISOL.EQ.STAR) AMAD=0.
47  IF(AMAD.NE.9E9) GO TO 471
   DSIZE=1.0
   IF(ISOL.EQ.STAR) DSIZE=0.0
   AMAD=SIZE(RIND,KINH)
   IF(AMAD.NE.DSIZE) WRITE(6,120) NAMNUC,AMAD,DSIZE
120  FORMAT(' ... WARNING ... DEFAULT AMAD SET FOR ',A8,
-      ' IS ',F5.2,' NOT ',F4.1)
471 IF(KMAX.EQ.1) GO TO 59
   IF(Resp(RIND,KINH).EQ.ISOL.AND.AMAD.EQ.SIZE(RIND,KINH).AND.
-   FLINH.EQ.GIABS(RIND,KINH,2)) GO TO 60
   MINH=KMAX
   MAXH=0
   DO 48 J=1,KMAX
     IF(Resp(RIND,J).NE.ISOL) GO TO 48
     MINH=MINO(J,MINH)
     MAXH=MAXO(J,MAXH)
48   CONTINUE
C IF RESP NOT FOUND, USE THE DEFAULT
C IF ONE RESP MATCH, USE THAT VALUE
C IF MANY RESP MATCHES, CHECK SIZE AND FLINH
   IF(MAXH-MINH) 59, 49, 50
49  KINH=MINH
   GO TO 59
50  J1=0
   J2=0
   DO 54 J=MINH,MAXH
     IF(AMAD.NE.SIZE(RIND,J).OR.
-     FLINH.NE.9E9.AND.FLINH.NE.GIABS(RIND,J,2)) GO TO 53
     J1=J
     KINH=J1
     GO TO 59
53  IF(J2.NE.0.AND.SIZE(RIND,J).EQ.SIZE(RIND,INDEF(RIND,2)).AND.
-     (FLINH.EQ.9E9.OR.FLINH.EQ.GIABS(RIND,J,2))) J2=J
```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

54 CONTINUE
   IF(J2.NE.0) KINH=J2
59 IF(F1INH.EQ.9E9) F1INH=GIABS(RIND,KINH,2)
   IF(RESP(RIND,KINH).NE.ISOL.OR.AMAD.NE.SIZE(RIND,KINH).OR.
- F1INH.NE.GIABS(RIND,KINH,2))
- WRITE(6,200)ISOL,F1INH,AMAD
- ,NAMNUC,RESP(RIND,KINH),GIABS(RIND,KINH,2),SIZE(RIND,KINH)
200 FORMAT(' ... WARNING ... CLASS=',A1,
- ' GI ABS (INH)=' ,F6.4, ' AMAD=' ,F5.2, ' NOT FOUND'
- ', ' , THESE VALUES SELECTED' /
- ' .. FOR- ' ,A8,7X,A1,14X,F6.4,6X,F5.2)
   IF(ISOL.NE.RESP(RIND,KINH)) ISOL=RESP(RIND,KINH)
   IF(AMAD.NE.SIZE(RIND,KINH)) AMAD=SIZE(RIND,KINH)
   IF(F1INH.NE.GIABS(RIND,KINH,2)) F1INH=GIABS(RIND,KINH,2)
C
C INGESTION FACTORS, CHECK FLING
60 KMAX=KIND(RIND,1)
   KING=INDEF(RIND,1)
   IF(FLING.EQ.9E9) FLING=GIABS(RIND,KING,1)
   IF(KMAX.EQ.1) GO TO 69
   DO 62 J=1,KMAX
     IF(FLING.NE.GIABS(RIND,J,1)) GO TO 62
     KING=J
     GO TO 70
62 CONTINUE
69 IF(FLING.NE.GIABS(RIND,KING,1)) WRITE(6,400)FLING,
- NAMNUC,GIABS(RIND,KING,1)
400 FORMAT(' ... WARNING ... ',
- ' GI ABS (ING)=' ,F6.4, ' NOT FOUND'
- ', ' , VALUE SELECTED' /
- ' .. FOR- ' ,A8,14X,F6.4)
   FLING=GIABS(RIND,KING,1)
C
70 DO 80 J=1,MAXORG
   ACFNHA(J,I)=FCFNHA(ORGID(J),RIND,KINH)
   ACFNGA(J,I)=FCFNGA(ORGID(J),RIND,KING)
80 CONTINUE
   RETURN
   END

```

```

C*****
C
C
C
C          OUTPUT ROUTINES CALLED BY THE MAIN PROGRAM
C
C

```


PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
12345678901234567890123456789012345678901234567890123456789012

C *
C *
C*****

C-----C
C C
C FILEOUT C
C ----- C
C WRITES DATA TO A FILE IN THE FORMAT USED BY AIRDOS-EPA. C
C CALLED BY THE MAIN PROGRAM. C
C C
C-----C

SUBROUTINE FILEOUT(NUMST, NNUCS, MAXORG, INTPA)

CHARACTER*8 ANMNUC(36), ANMORG(11, 36), BLANK,
- ONAMES(11), ORG(11), ORGNAM(11), RNAME(280)
CHARACTER*2 ENAME(78)
CHARACTER*1 AISOL(36), CLCS(280), ISOL, RESP(280, 20)

REAL*4 AAMAD(36), AANLAM(36), AAREA(6), ACFSBA(36), ACFSBW(36),
- ACFSUR(36), ACON(36, 20, 20), ADIA(6), ADIM(6), AF(36, 5),
- AFROG(11, 36), AFLING(36), AFLINH(36), ALAMRR(36), APDIA(36), APH(6),
- AQH(6), ARD1(36), ARD2(36), AREL(36, 6), ARHO(36), ARW1(36),
- ARW2(36), ASC(36), AVD(36), AVEL(6), AVG(36),
- ABSBV1(36), ABSBV2(36), ACFNGA(11, 36), ACFNHA(11, 36),
- AFSBFI(36), AFSBMI(36), ALMH20(36), ALMSUR(36), BOUND(20),
- GCON(36, 20, 20), INTFC(20, 20), JO, LAMW,
- MSUBB, PR(7), ATDCF(36), ATDCW(36), TG(3), VDcoef(20, 20), VDTAB(3)

INTEGER*4 IDIST(20), INTPA(20, 20), LIST(20), NOBCT(20, 20),
- NOMCT(20, 20), NORG(36), OPTION(9), AI(36, 5), AIFLAG(36), AKFLAG(36)
INTEGER*2 I, J, K, MAXNUC, MAXORG
INTEGER DECAY_CHAIN_FLAG(36), DAUGHTERS

COMMON /OPTIDATA/ OPTION, LIST, LIPO, NNTB, NRTB, NSTB, NTTB, NUTB, NVTB,
- T, TSUBB, GSFAC
COMMON /GRIDDATA/ BOUND, IDIST, NOL, NOU, NRL, NRU, SQSD
COMMON /PLUMDATA/ PR
COMMON /METEDATA/ LIDAI, LID, RR, TA, TAC, TG, Z, ZO, JO, DF
COMMON /PHYSDATA/ APH, ADIA, AVEL, AQH, AAREA, ADIM
COMMON /RADICHAR/ ANMNUC, ANMORG, ORGNAM, AISOL, ISOL
COMMON /RADIDATA/ AAMAD, AANLAM, AF, AFLING, AFLINH, AI, AIFLAG,
- AKFLAG, ANLAMO, APDIA, ARD1, ARD2, AREL, ARHO, ARW1,
- ARW2, ASC, AVD, AVG, CUTOFF, F1, F2, F3, F4, F5,

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

- I1, I2, I3, I4, I5, IAN, NORG, DAUGHTERS,
- NUMORG, SEQWL, VDTAB, DECAY_CHAIN_FLAG
COMMON /AGDTDATA/ FSUBG, FSUBL, FSUBP, FSUBS, F1V, F2V, F3V,
- F1B, F2B, F3B, F3BEFM, F3MLKM, F3VEGM, F1M, F2M, F3M, IMPFIX, LAMW, MSUBB,
- P, QSUBF, RBEF, RMLK, RVEG, R1, R2, TAUBEF, TSUBE1, TSUBE2, TSUBF, TSUBH1,
- TSUBH2, TSUBH3, TSUBH4, TSUBS, VSUBM, YSUBV1, YSUBV2
COMMON /USAGDATA/ DD1, BRTHRT, UV, UM, UF, UL
COMMON /POOLDATA/ DILFAC, USEFAC
COMMON /MODIDATA/ ABSBV1, ABSBV2, ACFNGA, ACFNHA, ACFNSA, ACFNSB,
- ACFNSR, AFROG, AFSBMI, AFSBFI, ALAMRR, ALMH2O, ALMSUR,
- ATDCF, ATDCW
COMMON /AGRIDATA/ INTFC, NOBCT, NOMCT
COMMON /DEPODATA/ VDCEOF
COMMON /CONCDATA/ ACON, GCON

COMMON/STAR/ PERD(16),UDCAT(7,16),UDAV(7,16),FRAW(7,16),
- WDCS(16,8,7),NS,IS(8),NW,WS(7),UBAR

COMMON/FILEC/ CLCS,ENAME,ORG,RESP,RNAME
COMMON/FILE/ FANLAM(280),FLAMRR(280),FCFSBA(280),FCFSBW(280),
- FCFSUR(280),FFROG(11,280),INDEX(280),
- FFSMI(78),FFSFI(78),FBSV1(78),FBSV2(78),FLAMSR(78),
- FLAMH2(78),FCFNHA(11,280,20),FCFNGA(11,280,20),
- GIABS(280,20,2),SIZE(280,20),KIND(280,2),
- INDEF(280,2),MAXNUC

DATA ONAMES/'TOT.BODY','R MAR','LUNGS','ENDOST','S WALL',
- 'LLI WALL','THYROID','LIVER','KIDNEYS','TESTES',
- 'OVARIES'/, BLANK/' '/

```

```

C
C WRITE INPUT TO ROUTINE MAIN
WRITE(27,1000) OPTION
1000 FORMAT('OPTIONS ',5I1,I2,3I1)
C
C WRITE INPUT TO ROUTINE CONCEN
IF(OPTION(1).EQ.2) GO TO 20
WRITE(27,1010)NOL,NOU,NRL,NRU
1010 FORMAT(8I10)
WRITE(27,1020)PR
1020 FORMAT(8(1PE10.3))
WRITE(27,1030)
1030 FORMAT('AREA')
WRITE(27,1020)SQSD
WRITE(27,1020)SEQWL
WRITE(27,1010)IDIST
WRITE(27,1040)

```

PREPAR (PREPAR2.FOR) Program File
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012
1040  FORMAT('AIR')
      WRITE(27,1010)LIDAI
      WRITE(27,1020)RR,TA,TG
      WRITE(27,1050)PERD
1050  FORMAT(16F5.4)
      WRITE(27,1052)((UDCAT(I,J),J=1,16),I=1,7)
      WRITE(27,1052)((UDAV(I,J),J=1,16),I=1,7)
1052  FORMAT(16F5.2)
      WRITE(27,1055)((FRAW(I,J),I=1,7),J=1,16)
1055  FORMAT(7F10.4)
      WRITE(27,1060)
1060  FORMAT('STACKS')
      WRITE(27,1010)NUMST
      DO 5 I=1,NUMST
5     WRITE(27,1020)APH(I),ADIA(I),AVEL(I),AQH(I),ADIM(I)
      WRITE(27,1070)
1070  FORMAT('NUCLIDES')
      WRITE(27,1010)NNUCS
      DO 10 I=1,NNUCS
      WRITE(27,1080)ANMNUC(I)
1080  FORMAT(A8)
      FVG=AVG(I)
      IF(FVG.LT.CUTOFF) FVG=0.
      FNLAM=AANLAM(I)
      IF(FNLAM.LT.ANLAMO) FNLAM=0.
      WRITE(27,1020)FNLAM,ASC(I),AVD(I),FVG
10    WRITE(27,1020)(AREL(I,J),J=1,NUMST)
C
      IF(OPTION(5).NE.1) GO TO 20
      WRITE(27,1020)BOUND
      WRITE(27,1090)((VDCOEF(I,J),J=NRL,NRU),I=NOL,NOU)
1090  FORMAT(10F8.4)
      WRITE(27,1100)LIST
1100  FORMAT(20I1)
C
C  WRITE INPUT TO ROUTINE DIRECT
20   IF(OPTION(1).NE.2) GO TO 40
      WRITE(27,1010)NOL,NOU,NRL,NRU
      WRITE(27,1020)SQSD
      WRITE(27,1010)IDIST
      WRITE(27,1020)SEQWL
      WRITE(27,1010)NNUCS
      DO 30 I=1,NNUCS
      WRITE(27,1110)ANMNUC(I),AVD(I),ASC(I)
1110  FORMAT(A8,1PE12.3,1PE10.3)
      WRITE(27,1020)((ACON(I,J,K),K=1,20),J=1,20)

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```
30    WRITE(27,1020)((GCON(I,J,K),K-1,20),J-1,20)
C
C    WRITE INPUT TO ROUTINE DOSEN
40    IF(OPTION(1).EQ.1) GO TO 80
      NVTB=0
      DO 44 I=1,NNUCS
      NUMORG=NORG(I)
      DO 44 I=1,NUMORG
         DO 42 J=1,11
          IF(ANMORG(1,I).EQ.ONAMES(J)) GO TO 44
42    CONTINUE
      NVTB=1
44    CONTINUE
      IF(NVTB.EQ.0.OR.MAXORG.EQ.11) GO TO 48
      J=NUMORG+1
      DO 46 I=J,11
46    ORG(I)=BLANK
48    WRITE(27,1010)LIPO
      WRITE(27,1010)NNTB,NRTB,NSTB,NTTB,NUTB,NVTB
      WRITE(27,1130)NOBCT
1130    FORMAT(16I5)
      WRITE(27,1130)NOMCT
      WRITE(27,1020)INTFC
      WRITE(27,1140)INTPA
1140    FORMAT((8(I9,'.')))
      WRITE(27,1010)IMPFIX
      WRITE(27,1020)RVEG,F3VEGM,RBEF,F3BEFM,RMLK,F3MLKM
      WRITE(27,1020)BRTHRT,DILFAC,USEFAC,T,DD1
      WRITE(27,1020)TSUBH1,TSUBH2,TSUBH3,TSUBH4
      WRITE(27,1020)LAMW
      WRITE(27,1020)TSUBE1,TSUBE2
      WRITE(27,1020)YSUBV1,YSUBV2
      WRITE(27,1020)FSUBP
      WRITE(27,1020)FSUBS
      WRITE(27,1020)QSUBF
      WRITE(27,1020)TSUBF
      WRITE(27,1020)UV,UM,UF,UL
      WRITE(27,1020)TSUBS
      WRITE(27,1020)FSUBG,FSUBL
      WRITE(27,1020)TSUBB
      WRITE(27,1020)P
      WRITE(27,1090)TAUBEF,MSUBB,VSUBM
      WRITE(27,1090)R1,R2
      IF(NVTB.EQ.1) WRITE(27,1160)ORG
1160    FORMAT(10A8)
      DO 70 I=1,NNUCS
```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

NUMORG=NORG(I)
CFSUR=GSFAC*ACFSUR(I)
WRITE(27,1190)ANMNUC(I),NUMORG,ALAMRR(I),ACFSBA(I),ACFSBW(I),
-   CFSUR,AKFLAG(I),ATDCF(I),ATDCW(I)
1190  FORMAT(A8,I2,4(1PE10.3),I10,2(1PE10.3))
WRITE(27,1200)(AFROG(J,I),J=1,11)
1200  FORMAT(11F5.3)
WRITE(27,1020)AFSBMI(I),AFSBFI(I),ABSBV1(I),ABSBV2(I)
WRITE(27,1210)ALMSUR(I),ALMH2O(I),AIFLAG(I),ARD1(I),ARD2(I),
-   ARW1(I),ARW2(I)
1210  FORMAT(2E10.3,I1,6F8.4)
WRITE(27,1220)(AI(I,J),J=1,5),(AF(I,J),J=1,5)
1220  FORMAT(5I4,5(1PE10.3))
WRITE(27,1230)AISOL(I),AAMAD(I),AF1INH(I)
1230  FORMAT(T11,A1,T15,1PE10.3,T27,1PE10.3)
DO 50 J=1,NUMORG
50    WRITE(27,1240)ANMORG(J,I),ACFNHA(J,I)
1240  FORMAT(A8,T10,1PE10.3)
WRITE(27,1250)AF1ING(I)
1250  FORMAT(T11,1PE10.3)
DO 60 J=1,NUMORG
60    WRITE(27,1250)ACFNHA(J,I)
70    CONTINUE
80    RETURN
END

```

```

C-----C
C                                           C
C                                           C
C                REPORT                      C
C                -----                     C
C  PRODUCES A FORMATTED LISTING OF THE AIRDOS-EPA INPUT DATA.  C
C  CALLED BY THE MAIN PROGRAM.                C
C-----C

```

SUBROUTINE REPORT(FMT,NUMST,NNUGS,INTPA,DATE_AND_TIME)

```

CHARACTER*36 DATE_AND_TIME
CHARACTER*8 ANMNUC(36), ANMORG(11,36), ORGNAM(11)
CHARACTER*5 DASHES(20)
CHARACTER*4 FMT, STAR, YLAB(20)
CHARACTER*3 DIR(16), OUTNUM(20)
CHARACTER*1 AISOL(36), ISOL,STAB(7)

```

REAL AAREA(6),AAMAD(36),AANLAM(36),ABSBV1(36),ABSBV2(36),

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
12345678901234567890123456789012345678901234567890123456789012

- ACON(36,20,20),ADIA(6),ADIM(6),
 - AF(36,5),AF1ING(36),AF1INH(36),ALAMRR(36),
 - APDIA(36),APH(6),AQH(6),
 - ARD1(36),ARD2(36),AREL(36,6),ARHO(36),
 - ARW1(36),ARW2(36),ASC(36),AVD(36),AVG(36),AVEL(6),
 - ACFNGA(11,36),ACFNHA(11,36),ACFSBA(36),ACFSBW(36),ACFSUR(36),
 - AFROG(11,36),AFSBFI(36),
 - AFSBMI(36),FTOT(7),GCON(36,20,20),INTFC(20,20),
 - ALMH2O(36),ALMSUR(36),LAMW,MSUBB,PR(7),
 - ATDCF(36),ATDCW(36),TG(3),VDCOEF(20,20),VDTAB(3)

INTEGER*4 AI(36,5),AIFLAG(36),
 - AKFLAG(36),BOUND(20),DD(16),
 - IDIST(20),INTPA(20,20),IXLAB(20),LIST(20),
 - NOBCT(20,20),NOMCT(20,20),NORG(36),OPTION(9)
 INTEGER DAUGHTERS, DECAY_CHAIN_FLAG (36)

COMMON /OPTIDATA/ OPTION,LIST,LIPO,NNTB,NRTB,NSTB,NTTB,NUTB,NVTB,
 - T,TSUBB,GSFAC

COMMON /GRIDDATA/ BOUND,IDIST,NOL,NOU,NRL,NRU,SQSD

COMMON /PLUMDATA/ PR

COMMON /METEDATA/ LIDAI,LID,RR,TA,TAC,TG,Z,ZO,JO,DF

COMMON /PHYSDATA/ APH,ADIA,AVEL,AQH,AAREA,ADIM

COMMON /RADICHA/ ANMNUC,ANMORG,ORGNAM,AISOL,ISOL

COMMON /RADIDATA/ AAMAD,AANLAM,AF,AF1ING,AF1INH,AI,AIFLAG,

- AKFLAG,ANLAM0,APDIA,ARD1,ARD2,AREL,ARHO,ARW1,

- ARW2,ASC,AVD,AVG,CUTOFF,F1,F2,F3,F4,F5,

- I1,I2,I3,I4,I5,IAN,NORG,DAUGHTERS,

- NUMORG,SEQWL,VDTAB, DECAY_CHAIN_FLAG

COMMON /AGDTDATA/ FSUBG,FSUBL,FSUBP,FSUBS,F1V,F2V,F3V,

- F1B,F2B,F3B,F3BEFM,F3MLKM,F3VEGM,F1M,F2M,F3M,IMPFIX,LAMW,MSUBB,

- P,QSUBF,RBEF,RMLK,RVEG,R1,R2,TAUBEF,TSUBE1,TSUBE2,TSUBF,TSUBH1,

- TSUBH2,TSUBH3,TSUBH4,TSUBS,VSUBM,YSUBV1,YSUBV2

COMMON /USAGDATA/ DD1,BRTHRT,UV,UM,UF,UL

COMMON /POOLDATA/ DILFAC,USEFAC

COMMON /MODIDATA/ ABSBV1,ABSBV2,ACFNGA,ACFNHA,ACFSBA,ACFSBW,

- ACFSUR,AFROG,AFSBMI,AFSBFI,ALAMRR,ALMH2O,ALMSUR,

- ATDCF,ATDCW

COMMON /AGRIDATA/ INTFC,NOBCT,NOMCT

COMMON /DEPODATA/ VDCOEF

COMMON /CONCDATA/ ACON,GCON

COMMON/STAR/ PERD(16),UDCAT(7,16),UDAV(7,16),FRAW(7,16),

- WDCS(16,8,7),NS,IS(8),NW,WS(7),UBAR

DATA DASHES/20*'-----'/

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

DATA STAR/'STAR'/
DATA DIR/' N','NNW',' NW','WNW',' W','WSW',' SW','SSW',
- ' S','SSE',' SE','ESE',' E','ENE',' NE','NNE'/
DATA OUTNUM/' 1',' 2',' 3',' 4',' 5',' 6',' 7',
- ' 8',' 9',' 10',' 11',' 12',' 13',' 14',' 15',
- ' 16',' 17',' 18',' 19',' 20'/
DATA STAB/'A','B','C','D','E','F','G'/
DATA DD/1,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2/

```

```

IF(OPTION(2).EQ.1) GO TO 2
C SQUARE GRID
DO 1 I=1,20
  IXLAB(I)=I
  1 YLAB(I)=OUTNUM(I)
  GO TO 6
C CIRCULAR GRID
2 CONTINUE
DO 3 I=1,16
  3 YLAB(I)=DIR(I)
DO 4 I=1,20
  4 IXLAB(I)=IDIST(I)
C
C WRITE OPTION DATA
6 WRITE(28,100) DATE_AND_TIME
100 FORMAT('1',T10,'DATE',2X,A)
WRITE(28,105)
105 FORMAT('0',T3,'MAIN OPTIONS:')
IF(OPTION(1).EQ.0) WRITE(28,110)
110 FORMAT('0',T10,'CONCEN AND DOSEN',T40,'OPTION(1)=0')
IF(OPTION(1).EQ.1) WRITE(28,120)
120 FORMAT('0',T10,'CONCEN ONLY',T40,'OPTION(1)=1')
IF(OPTION(1).EQ.2) WRITE(28,130)
130 FORMAT('0',T10,'DIRECT AND DOSEN',T40,'OPTION(1)=2')
IF(OPTION(2).EQ.0) WRITE(28,140)
140 FORMAT('0',T10,'SQUARE GRID',T40,'OPTION(2)=0')
IF(OPTION(2).EQ.1) WRITE(28,150)
150 FORMAT('0',T10,'CIRCULAR GRID',T40,'OPTION(2)=1')
IF(OPTION(1).EQ.2) GO TO 5
WRITE(28,155)
155 FORMAT('0',T3,'CONCEN OPTIONS:')
IF(OPTION(3).EQ.0) WRITE(28,160)
160 FORMAT('0',T10,'SECTOR-AVERAGED',T40,'OPTION(3)=0')
IF(OPTION(3).EQ.1) WRITE(28,170)
170 FORMAT('0',T10,'PLUME CENTER-LINE',T40,'OPTION(3)=0')
IF(OPTION(4).EQ.0) WRITE(28,180)

```

PREPAR (PREPAR2.FOR) Program File
(continued)

```
0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012

180  FORMAT('0',T10,'BUOYANT PLUME',T40,'OPTION(4)=0')
      IF(OPTION(4).EQ.1) WRITE(28,190)
190  FORMAT('0',T10,'MOMENTUM-TYPE PLUME',T40,'OPTION(4)=1')
      IF(OPTION(4).EQ.2) WRITE(28,200)
200  FORMAT('0',T10,'PLUME RISE INPUT',T40,'OPTION(4)=2')
      IF(OPTION(5).EQ.0) WRITE(28,210)
210  FORMAT('0',T10,'FIXED DEPOSITION VELOCITY',T40,'OPTION(5)=0')
      IF(OPTION(5).EQ.1) THEN
          WRITE(28,220)
          WRITE(28,221)
      ENDIF
220  FORMAT('0',T10,'DEPOSTION VELOCITY VARIES WITH')
221  FORMAT(T15,'DIRECTION AND DISTANCE',T40,'OPTION(5)=1')
      IF(OPTION(6).EQ.0) WRITE(28,230)
230  FORMAT('0',T10,'NO PUNCH, CONCEN',T40,'OPTION(6)=0')
      IF(OPTION(6).NE.0) WRITE(28,235)OPTION(6)
235  FORMAT('0',T10,'PUNCH CONCEN',T40,'OPTION(6)=' ,I2)
      IF(OPTION(7).EQ.0) WRITE(28,240)
240  FORMAT('0',T10,'POINT SOURCE',T40,'OPTION(7)=0')
      IF(OPTION(7).EQ.1) WRITE(28,250)
250  FORMAT('0',T10,'AREA SOURCE',T40,'OPTION(7)=1')
      IF(OPTION(8).EQ.0) WRITE(28,260)
260  FORMAT('0',T10,'PRINT CONCEN MAIN TABLE',T40,'OPTION(8)=0')
      IF(OPTION(8).EQ.1) WRITE(28,270)
270  FORMAT('0',T10,'NO PRINT CONCEN MAIN TABLE',T40,'OPTION(8)=1')
      IF(OPTION(8).EQ.0.AND.OPTION(5).EQ.1) THEN
          WRITE(28,280)
          WRITE(28,281)(I,I=1,20),(LIST(I),I=1,20)
      ENDIF
280  FORMAT('0',T10,'CONCENTRATIONS PRINTED ONLY FOR')
281  FORMAT('0',T15,'DISTANCES FOR WHICH',
-    ' LIST(I)=0'/' INDEX',20I3/' LIST',20I3)
      IF(OPTION(9).EQ.0) WRITE(28,290)
290  FORMAT('0',T10,'PRINT CONCEN CHI/Q TABLES',T40,'OPTION(9)=0')
      IF(OPTION(9).EQ.1) WRITE(28,300)
300  FORMAT('0',T10,'NO PRINT CONCEN CHI/Q TABLES',T40,'OPTION(9)=1')
      5  IF(OPTION(1).EQ.1) GO TO 10
          WRITE(28,305)
305  FORMAT('0',T3,'DOSEN OPTIONS:')
      IF(LIPO.EQ.0) WRITE(28,310)
310  FORMAT('0',T10,'INDIVIDUAL ASSESSMENT',T40,'LIPO=0')
      IF(LIPO.EQ.1) WRITE(28,320)
320  FORMAT('0',T10,'POPULATION ASSESSMENT',T40,'LIPO=1')
      IF(NNTB.NE.0) WRITE(28,330)NNTB
330  FORMAT('0',T10,'PRINT DOSEN TABLES',T40,'NNTB=' ,I2)
      IF(NNTB.EQ.0) WRITE(28,335)
```


PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

335  FORMAT('0',T10,'NO PRINT DOSEN TABLES',T40,'NNTB=0')
      IF(NRTB.EQ.0) WRITE(28,340)
340  FORMAT('0',T10,'NO PUNCH DOSES',T40,'NRTB=0')
      IF(NRTB.EQ.1) WRITE(28,350)
350  FORMAT('0',T10,'PUNCH DOSES',T40,'NRTB=1')
      IF(NSTB.EQ.0) WRITE(28,360)
360  FORMAT('0',T10,'NO DARTAB OUTPUT',T40,'NSTB=0')
      IF(NSTB.EQ.1) WRITE(28,370)
370  FORMAT('0',T10,'DARTAB FILE AND PRINT',T40,'NSTB=1')
      IF(NTTB.EQ.0) WRITE(28,380)
      IF(NSTB.EQ.2) WRITE(28,385)
385  FORMAT('0',T10,'DARTAB FILE ONLY',T40,'NSTB=2')
380  FORMAT('0',T10,'NO PRINT DOSE SUMMARY',T40,'NTTB=0')
      IF(NTTB.EQ.1) WRITE(28,390)
390  FORMAT('0',T10,'PRINT DOSE SUMMARY',T40,'NTTB=1')
      IF(NUTB.EQ.0) WRITE(28,400)
400  FORMAT('0',T10,'NO RN-222 WORKING LEVELS',T40,'NUTB=0')
      IF(NUTB.EQ.1) WRITE(28,410)
410  FORMAT('0',T10,'RN-222 WORKING LEVELS',T40,'NUTB=1')
      IF(NVTB.EQ.0) WRITE(28,411)
411  FORMAT('0',T10,'NO READ ORGAN NAMES',T40,'NVTB=0')
      IF(NVTB.EQ.1) WRITE(28,412)
412  FORMAT('0',T10,'READ ORGAN NAMES',T40,'NVTB=1')
      WRITE(28,420) TSUBB
      WRITE(28,421) T
420  FORMAT('0',T10,'BUILDUP TIME IN SOIL',T40,'TSUBB=',1PG10.4,
- ' YEARS')
421  FORMAT(T40,'T=',1PG10.4,' DAYS')
      WRITE(28,100) DATE_AND_TIME
      WRITE(28,425)
425  FORMAT('0',T3,'GRID DATA:')
      IF(OPTION(2).EQ.0) WRITE(28,430)NOL,NOU,NRL,NRU
430  FORMAT('0',T10,'BOUNDS OF X-INDICES',T40,'NOL=',I2,' NOU=',I2,
- '/' ' ,T10,'BOUNDS OF Y-INDICES',T40,'NRL=',I2,' NRU=',I2)
      IF(OPTION(2).EQ.1) WRITE(28,440)NOL,NOU,NRL,NRU
440  FORMAT('0',T10,'BOUNDS OF DIRECTION-INDICES',T40,'NOL=',I2,
- ' NOU=',I2/' ' ,T10,'BOUNDS OF DISTANCE-INDICES',T40,'NRL=',I2,
- ' NRU=',I2)
      WRITE(28,450) SQSD,NRU,IDIST(NRU)
450  FORMAT('0',T3,'SQSD=',F6.1,' (M), COMPUTED FROM IDIST(' ,I2,
- ')=' ,I5,' (M)')
      IF(OPTION(2).EQ.1) THEN
          WRITE(28,460)
          WRITE(28,461)(IDIST(J),J=1,NRU)
      ENDIF
460  FORMAT('0',T3,'IDIST, THE ARRAY OF RADIAL DISTANCES (M)')

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

461  FORMAT(T3,I6)
      IF(OPTION(5).EQ.1) WRITE(28,470)BOUND
470  FORMAT('0',T3,'BOUND (M) ',10F8.0/ ' ',11X,10F8.0)
      IF(OPTION(1).EQ.1) GO TO 10
C
C  WRITE DEPOSITION VELOCITY COEFFICIENTS
10  IF(OPTION(5).NE.1) GO TO 15
      WRITE(28,100) DATE_AND_TIME
      WRITE(28,530)(IDIST(I),I=NRL,NRU)
530  FORMAT('0',T3,'VDCOEF, FACTOR',
- ' ' TO ALLOW FOR VARIATION IN DEPOSITION VELOCITY',
- ' ' WITH DIRECTION AND DISTANCE'//' ',20I6)
      DO 12 I=NOL,NOU
12  WRITE(28,535)DIR(I),(VDCOEF(I,J),J=1,20)
535  FORMAT(' ',T3,A3,1X,20F6.3)
C
C  WRITE SOURCE DATA
15  IF(OPTION(1).EQ.2) GO TO 20
      WRITE(28,100) DATE_AND_TIME
      WRITE(28,540)NUMST
540  FORMAT('0',T10,'NUMBER OF SOURCES',T40,'NUMST=',I1)
      WRITE(28,520)NNUCS
520  FORMAT('0',T10,'NUMBER OF NUCLIDES',T40,'NNUCS=',I2)
      WRITE(28,550)(I,I=1,NUMST)
550  FORMAT('0',10X,'SOURCE #:',6I10)
      WRITE(28,551)
551  FORMAT(' ')
      WRITE(28,555)(APH(I),I=1,NUMST)
555  FORMAT(T3,'HEIGHT',T40,'PH= ',6F10.1,' (M)')
      WRITE(28,560)(ADIA(I),I=1,NUMST)
560  FORMAT(T3,'DIAMETER',T40,'DIA= ',6F10.2,' (M)')
      WRITE(28,570)(AVEL(I),I=1,NUMST)
570  FORMAT(T3,'EXIT VELOCITY',T40,'VEL= ',6F10.0,' (M/S)')
      WRITE(28,580)(AQH(I),I=1,NUMST)
580  FORMAT(T3,'HEAT RELEASE RATE',T40,'QH= ',6F10.0,' (CAL/S)')
      IF(OPTION(7).EQ.1) WRITE(28,590)(ADIM(I),I=1,NUMST)
590  FORMAT(T3,'DIAMETER',T40,'DIM SOURCE= ',6F10.1,' (M)')
      IF(OPTION(7).EQ.1) WRITE(28,600)(AAREA(I),I=1,NUMST)
600  FORMAT(T3,' AREA (M**2) ',6F10.1)
      IF(OPTION(4).EQ.0) WRITE(28,180)
C  IF(OPTION(4).EQ.1) WRITE(28,190)
      IF(OPTION(4).EQ.2) THEN
          WRITE(28,601)
          WRITE(28,602)
          WRITE(28,603)(PR(J),J=1,7)
      ENDIF

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

601  FORMAT('0',T3,'PLUME RISE')
602  FORMAT(' ',T6,'CLASS A-G:',T24,'(M)')
603  FORMAT(' ',T8, F10.2)
      WRITE(28,610)
      WRITE(28,481)
481  FORMAT('          ')
610  FORMAT(T3,'NUCLIDE RELEASE RATE, REL (CI/YR)')
      DO 17 I=1,NNUCS
      17  WRITE(28,615)I,ANMNUC(I),(AREL(I,J),J=1,NUMST)
615  FORMAT(' ',4X,I2,1X,A8,3X,6(1PG10.4,1X))
C
C  WRITE CONCENTRATIONS USED IN AIRDOS-EPA ROUTINE DIRECT
20  IF(OPTION(1).NE.2) GO TO 25
      KK=(NRU-NRL)/10+1
      DO 24 IN=1,NNUCS
      WRITE(28,100) DATE_AND_TIME
      WRITE(28,620)ANMNUC(IN)
620  FORMAT('0',A8,'ACON, AIR CONCENTRATION (PICOCURIES/CM**3)')
      DO 22 K=1,KK
      J1=NRL+10*(K-1)
      JJ=MINO(J1+9,NRU)
      WRITE(28,931)(IXLAB(J),J=J1,JJ)
      WRITE(28,932)
      DO 22 I=NOL,NOU
      22  WRITE(28,933) YLAB(I),(ACON(IN,I,J),J=J1,JJ)
      IF(KK.NE.1) THEN
          WRITE(28,100) DATE_AND_TIME
      ENDIF
      WRITE(28,640)ANMNUC(IN)
640  FORMAT('0',A8,'GCON, SURFACE DEPOSITION RATE (PICOCURIES/
-   ',CM**2-S)')
      DO 24 K=1,KK
      J1=NRL+10*(K-1)
      JJ=MINO(J1+9,NRU)
      IF(KK.NE.1) WRITE(28,931)(IXLAB(J),J=J1,JJ)
      WRITE(28,932)
      DO 24 I=NOL,NOU
      24  WRITE(28,933) YLAB(I),(GCON(IN,I,J),J=J1,JJ)
C
25  IF(OPTION(1).EQ.2) GO TO 35
      WRITE(28,100) DATE_AND_TIME
      WRITE(28,684)
684  FORMAT('0INDEX',T16,'NAME',T29,'ISOL',T42,'LAMSUR',T57,'UPTAKE',
-   T73,'AMAD')
      WRITE(28,685)
685  FORMAT(T29,'CLASS',T44,'1/D',T57,'FLING',T72,'MICRONS')

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

        WRITE(28,6899)
        DO 1035 I=1,NNUCS
1035     WRITE(28,686)I,ANMNUC(I),AISOL(I),ALMSUR(I),AF1ING(I),
        -AAMAD(I)
        686  FORMAT(T4,I3,T14,A8,T31,A,T41,1PE8.2,T57,OPF4.2,T73,F5.2)
        WRITE(28,6899)
        WRITE(28,6899)
        WRITE(28,6899)
6899     FORMAT('
        WRITE(28,689)
        689  FORMAT(T2,'INDEX',T16,'NAME',T30,'SC',T43,'VD',T60,'VG',
        - T73,'ANLAM')
        WRITE(28,6891)
6891     FORMAT(T30,'1/S',T43,'M/S',T60,'M/S',T74,'1/D')
        DO 1036 I=1,NNUCS
1036     WRITE(28,6892)I,ANMNUC(I),ASC(I),AVD(I),AVG(I),AANLAM(I)
6892     FORMAT(T4,I3,T14,A8,T27,1PE8.2,T40,1PE8.2,T57,1PE8.2,T71,
        - 1PE8.2)
        IF(CUTOFF.GT.0.) WRITE(28,687)CUTOFF
        687  FORMAT('0 ***NOTE: VG SET TO ZERO FOR AIRDOS UNLESS GREATER',
        - ' THAN',1PE10.3)
        IF(ANLAMO.GT.0.) WRITE(28,688)ANLAMO
        688  FORMAT('0 ***NOTE: ANLAM SET TO ZERO FOR AIRDOS UNLESS GREATER',
        - ' THAN',1PE10.3)
C
C  WRITE WIND DATA
        WRITE(28,100) DATE_AND_TIME
        WRITE(28,790)
        790  FORMAT('0',' FOR EACH STABILITY CLASS')
        WRITE(28,791)STAB
        791  FORMAT(T8,A1,T18,A1,T27,A1,T36,A1,T45,A1,T54,A1,T63,A1,T71,
        - 'PERD')
        WRITE(28,800)
        800  FORMAT(' OUDCAT, HARMONIC AVERAGE WIND SPEEDS ( WIND TOWARDS )'
        +      ,T69,'WIND FREQ.')
        WRITE(28,801)(DIR(J),(UDCAT(I,J),I=1,7),PERD(J),J=1,16)
        801  FORMAT(' ',A3,T6,F6.3,T16,F6.3,T25,F6.3,T34,F6.3,T43,
        -F6.3,T52,F6.3,T61,F6.3,T70,F6.3)
        WRITE(28,932)
        805  WRITE(28,806)
        806  FORMAT(' OUDAV, ARITHMETIC AVERAGE WIND SPEEDS ( WIND TOWARDS )')
        WRITE(28,810)(DIR(J),(UDAV(I,J),I=1,7),J=1,16)
        810  FORMAT(' ',A3,T6,F6.3,T16,F6.3,T25,F6.3,T34,F6.3,T43,
        - F6.3,T52,F6.3,T61,F6.3)
C
        WRITE(28,100) DATE_AND_TIME

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

      WRITE(28,790)
      WRITE(28,812)STAB
812  FORMAT(T8,A1,T17,A1,T26,A1,T36,A1,T45,A1,T54,A1,T63,A1)
      WRITE(28,815)
815  FORMAT('0',
+      'FRAW, FREQUENCIES OF STABILITY CLASSES ( WIND TOWARDS )')
      WRITE(28,820)(DIR(J),(FRAW(I,J),I=1,7),J=1,16)
820  FORMAT(' ',A3,T7,1PE8.2,T16,1PE8.2,T25,1PE8.2,T34,1PE8.2,T43,
- 1PE8.2,T52,1PE8.2,T61,1PE8.2)
      DO 1820 I=1,7
          FTOT(I)=0.
          DO 1820 J=1,16
1820      FTOT(I)=FTOT(I)+FRAW(I,J)*PERD(J)
      WRITE(28,822)FTOT
822  FORMAT(' TOT',T7,1PE8.2,T16,1PE8.2,T25,1PE8.2,T34,1PE8.2,T43,
- 1PE8.2,T52,1PE8.2,T61,1PE8.2)
      WRITE(28,825)
      WRITE(28,826)LIDAI
825  FORMAT('0',T3,'HEIGHT OF LID')
826  FORMAT(' ',T6,'LIDAI= ',I7,' (M)')
      WRITE(28,830)
      WRITE(28,831)RR
830  FORMAT('0',T3,'RAINFALL RATE')
831  FORMAT(' ',T6,'RR= ',F10.1,' (CM/Y)')
      WRITE(28,832)
      WRITE(28,833)TAC,TA
832  FORMAT('0',T3,'AVERAGE AIR TEMPERATURE')
833  FORMAT('0',T6,'TA= ',F10.1,' (DEG C) ',F10.1,' (K)')
      WRITE(28,834)
      WRITE(28,835)ZO
834  FORMAT('0',T3,'SURFACE ROUGHNESS LENGTH')
835  FORMAT(' ',T6,'ZO= ',F10.3,' (M)')
      WRITE(28,836)
      WRITE(28,837)Z
836  FORMAT('0',T3,'HEIGHT OF WIND MEASUREMENTS')
837  FORMAT(' ',T6,'Z= ',F10.1,' (M)')
      WRITE(28,838)
      WRITE(28,839)UBAR
838  FORMAT('0',T3,'AVERAGE WIND SPEED')
839  FORMAT(' ',T6,'UBAR= ',F8.2,' (M/S)')
      WRITE(28,8835)
      WRITE(28,8836)TG(1)
      WRITE(28,8837)TG(2)
      WRITE(28,8838)TG(3)
8835  FORMAT('0',T3,'VERTICAL TEMPERATURE GRADIENTS: (TG) (K/M)')
8836  FORMAT(T6,'STABILITY E ',T20,F10.3)

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

8837 FORMAT(T6,'STABILITY F ',T20,F10.3)
8838 FORMAT(T6,'STABILITY G ',T20,F10.3)
C
  IF(FMT.NE.STAR) GO TO 35
  DO 33 J=1,NS
    IF(MOD(J+2,3).NE.0) GO TO 33
    WRITE(28,100) DATE_AND_TIME
    WRITE(28,840)
840   FORMAT('0','STAR INPUT, WIND FREQUENCIES ( WIND FROM )')
    33   WRITE(28,850)STAB(IS(J)),(DIR(DD(I))),(WDGS(I,J,K),
      K=1,NW),I=1,16)
850   FORMAT('0','CLASS: ',A1/(16(' ',A3,6(1PE10.3)/)))
C
C WRITE NUCLIDE DATA
  35   IF(OPTION(1).EQ.1) RETURN
CC
  IF ( DAUGHTERS .EQ. 1 ) THEN
    DO 1040 I=1,NNUCS
      DO 1040 J=1,5
        IF(AI(I,J).NE.0) GO TO 1050
1040   CONTINUE
        GO TO 1060
1050   WRITE(28,100)DATE_AND_TIME
        WRITE(28,678)
678   FORMAT('0 I,F INGROWTH FACTORS'//7X,'NUCLIDE      PARENT'/
      ' INDEX NAME      I NAME      F'/)
    DO 1034 I=1,NNUCS
      IF(AI(I,1).EQ.0) GO TO 1034
      WRITE(28,682)I,ANMNUC(I),AI(I,1),ANMNUC(AI(I,1)),AF(I,1)
682   FORMAT(' ',I3,3X,A8,I2,2X,A8,1PG10.4)

C***** DETERMINE WHICH DECAY CHAIN SERIES IT IS AND SET
C***** MAIN PARENT TO THE CORRECT VALUE FOR THE DATABASE
C***** 1 ----> TH-232 *****
C***** 2 ----> U-238 *****
C***** 3 ----> CS-137 *****
C***** 4 ----> BA-140 *****
C***** 5 ----> MO-99 *****
C***** 0 ----> NOT MAIN PARENT *****

  IF ( ANMNUC ( AI(I,1) ) .EQ. 'TH-232') THEN
    DECAY_CHAIN_FLAG ( AI(I,1) ) = 1
  ELSE IF ( ANMNUC ( AI(I,1)-1 ) .EQ. 'U-238' ) THEN
    DECAY_CHAIN_FLAG ( AI(I,1)-1 ) = 2
  ELSE IF ( ANMNUC ( AI(I,1) ) .EQ. 'CS-137' ) THEN
    DECAY_CHAIN_FLAG ( AI(I,1) ) = 3

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

ELSE IF ( ANMNUC ( AI(I,1) ) .EQ. 'BA-140' ) THEN
  DECAY_CHAIN_FLAG ( AI(I,1) ) = 4
ELSE IF ( ANMNUC ( AI(I,1) ) .EQ. 'MO-99' ) THEN
  DECAY_CHAIN_FLAG ( AI(I,1) ) = 5
ELSE
C***** DO NOTHING
ENDIF

DO 1033 J=2,5
  IF(AI(I,J).EQ.0) GO TO 1034
1033  WRITE(28,770)AI(I,J),ANMNUC(AI(I,J)),AF(I,J)
  770  FORMAT(15X,I2,2X,A8,1PG10.4)
1034  CONTINUE
ENDIF

1060 WRITE(28,100) DATE_AND_TIME
WRITE(28,910)
910  FORMAT('0',' ' NOBCT, NUMBER OF BEEF CATTLE')
  KK=(NRU-NRL)/7 + 1
  DO 131 K=1, KK
  J1=NRL+7*(K-1)
  JJ=MINO(J1+6, NRU)
  WRITE(28,911)(IXLAB(J), J=J1, JJ)
911  FORMAT('0',4X,7I6)
  WRITE(28,912)(DASHES(J), J=J1, JJ)
912  FORMAT(' ',4X,7(1X,A5))
  DO 131 I=NOL, NOU
131  WRITE(28,915) YLAB(I), (NOBCT(J, I), J=J1, JJ)
915  FORMAT(' ',A3,7I6)
  WRITE(28,100) DATE_AND_TIME
  WRITE(28,920)
920  FORMAT('0',' ' NOMCT, NUMBER OF MILK CATTLE')
  KK=(NRU-NRL)/7 + 1
  DO 132 K=1, KK
  J1=NRL+7*(K-1)
  JJ=MINO(J1+6, NRU)
  WRITE(28,911)(IXLAB(J), J=J1, JJ)
  WRITE(28,912)(DASHES(J), J=J1, JJ)
  DO 132 I=NOL, NOU
132  WRITE(28,915) YLAB(I), (NOMCT(J, I), J=J1, JJ)
C
  WRITE(28,100) DATE_AND_TIME
  WRITE(28,930)
930  FORMAT('0 INTFC, AREA OF VEGETABLE CROP PRODUCTION (M**2)')
  KK=(NRU-NRL)/7 + 1
  DO 133 K=1, KK

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
12345678901234567890123456789012345678901234567890123456789012

```

J1=NRL+7*(K-1)
JJ=MINO(J1+6,NRU)
WRITE(28,931)(IXLAB(J),J=J1,JJ)
931 FORMAT('0',4X,7I10)
WRITE(28,913)(DASHES(J),J=J1,JJ)
913 FORMAT(' ',4X,7(5X,A5))
932 FORMAT(1X)
DO 133 I=NOL,NOU
133 WRITE(28,933) YLAB(I),(INTFC(J,I),J=J1,JJ)
933 FORMAT(' ',A3,1X,7(1PE10.2))
C
WRITE(28,100) DATE_AND_TIME
WRITE(28,932)
WRITE(28,941)
941 FORMAT('0 INTPA, POPULATION')
KK=(NRU-NRL)/7 + 1
DO 135 K=1,KK
J1=NRL+7*(K-1)
JJ=MINO(J1+6,NRU)
WRITE(28,940)(IXLAB(J),J=J1,JJ)
940 FORMAT('0',4X,7I7)
WRITE(28,914)(DASHES(J),J=J1,JJ)
914 FORMAT(' ',4X,7(2X,A5))
DO 135 I=NOL,NOU
135 WRITE(28,136) YLAB(I),(INTPA(J,I),J=J1,JJ)
136 FORMAT(' ',A3,7I7)
C
C THESE FORMATS WERE MODIFIED FROM AIRDOS-EPA
9025 FORMAT(' ',T15,'(KG/Y), INGESTION OF PRODUCE',
> T2,'UV-',1PG9.4)
9026 FORMAT(' ',T15,'(L/Y), INGESTION OF MILK',
> T2,'UM-',1PG9.4)
9027 FORMAT(' ',T15,'(KG/Y), INGESTION OF MEAT',
> T2,'UF-',1PG9.4)
9028 FORMAT('0',T15,'(KG/Y), INGESTION OF LEAFY VEGETABLES'
> ,T2,'UL-',1PG9.4)
40 RETURN
END

```

```

C-----C
C
C          OUTPUT_TO_DARTAB
C          -----
C APPENDS SELECTED INFORMATION TO THE FILE WRITTEN TO IN
C GET_NEW_INPUT. THIS FILE WILL BE USED IN DARTAB TO CREATE
C THE SYNOPSIS REPORT. CALLED FROM THE MAIN PROGRAM.
C-----C

```


PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

EQUIVALENCE (FV(1),F1V), (FV(2),F2V), (FV(3),F3V), (FB(1),F1B),
-           (FB(2),F2B), (FB(3),F3B), (FM(1),F1M), (FM(2),F2M),
-           (FM(3),F3M)

```

```

WRITE(26,1020)TA
1020 FORMAT(8(1PE10.3))
WRITE(26,1020)RR
WRITE(26,1010)LIDAI
1010 FORMAT(8I10)
WRITE (26,1015) LATITUDE
WRITE (26,1015) LONGITUDE
1015 FORMAT (F8.4)

```

```

C*** OUTPUT THE FOOD FRACTIONS FOR FOOD ARRAYS GENERATED OR AN
C*** APPROPRIATE MESSAGE IF NO FOOD ARRAYS ,ERROR ETC.

```

```

IF ( ( LIPO .EQ. 1 ) .AND.
+   ( ( .NOT. RN_RUN ) .OR. USERARRAY ) ) THEN

  IF ( STATE .EQ. 'ZZ' ) THEN
    WRITE(26,*)' ERROR IN STATE GIVEN..DEFAULT FOOD ARRAYS USED'

  ELSE IF ( FOODARRAY_GEN_AUTO ) THEN
    WRITE(26,1025) DB, DM, FC
1025   format ( ' FOOD FRACTIONS USED: DB = ',1PE10.2,' DM = ',
+           1PE10.2,' FC = ',1PE10.2 )

  ELSE IF ( USERARRAY ) THEN
    WRITE(26,*)'FOOD ARRAYS WERE PROVIDED BY THE USER '

  ELSE
    WRITE(26,*)
+   'FOOD ARRAYS WERE NOT GENERATED OR SUPPLIED FOR THIS RUN',
+   ' . DEFAULT VALUES USED.'

  ENDIF

```

```

ELSE
C*** INDIVIDUAL RUN, OR RADON RUN DEFAULT ARRAYS ARE USED

```

```

WRITE(26,*)
+   'FOOD ARRAYS WERE NOT GENERATED OR SUPPLIED FOR THIS RUN',
+   ' . DEFAULT VALUES USED.'

```

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```

ENDIF

WRITE(26,1020)F1V,F2V,F3V,F1B,F2B,F3B,F1M,F2M,F3M
WRITE (26,1030) RN_RUN
1030 FORMAT(I1)
IF ( .NOT. RN_RUN )
+   WRITE (26,*) DAUGHTERS
WRITE(26,1010)NNUCS,NUMST
DO 10 I=1,NNUCS
    WRITE(26,1080)ANMNUC(I)
1080  FORMAT(A8)
IF ( .NOT. RN_RUN )
+   WRITE (26, *) DECAY_CHAIN_FLAG (I)
WRITE(26,1090)AISOL(I),AAMAD(I), ALMSUR(I)
1090  FORMAT(A1,1X,1PE10.3,1X,1PE10.3)
WRITE(26,1020)(AREL(I,J),J=1,NUMST)
10  CONTINUE
WRITE(26,2000)OPTION(7),OPTION(4)
2000  FORMAT(I1,1X,I1)
WRITE(26,1020)(APH(I),I=1,NUMST)
IF ( OPTION(7) .EQ. 1 ) THEN
    WRITE(26,1020)(AAREA(I),I=1,NUMST)
    WRITE(26,1020)(ADIM(I),I=1,NUMST)
ELSE
    WRITE(26,1020)(ADIA(I),I=1,NUMST)
ENDIF
IF ( OPTION(4) .EQ. 0 ) THEN
    WRITE(26,1020)(AQH(I),I=1,NUMST)
ELSE IF ( OPTION(4) .EQ. 1 ) THEN
    WRITE(26,1020)(AVEL(I),I=1,NUMST)
ELSE
    WRITE(26,1020)PR
ENDIF
RETURN
END

```

```

C*****
C
C
C
C
C
C
C
C
C*****

```

UTILITY PROGRAMS

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
12345678901234567890123456789012345678901234567890123456789012

```
C-----
C
C           MAKE HEADING
C
C THIS ROUTINE WAS ADDED 8/88. IT CALLS THE SYSTEM DATE AND TIME
C ALL INFORMATION IS RETURNED IN TODAY ( SEE FORTRAN MANUAL )
C THIS INFORMATION IS PRINTED ON THE TOP OF EVERY PAGE.THEREFORE
C THIS ROUTINE EXTRACTS THE VALUES FROM TODAY AND PLACES IT INTO
C A CHARACTER STRING TO PRINT AT THE TOP OF THE REPORTS. THE
C ROUTINE INT TO CHAR IS CALLED WHEN NECESSARY.
C-----
```

```
C
C   SUBROUTINE MAKE_HEADING( DATE_AND_TIME )
C
C   CHARACTER*3 DAYWK(7)
C   CHARACTER*2 AMPM(2)
C   CHARACTER*9 MONTH(12)
C
C   CHARACTER*36 DATE_AND_TIME
C
C   INTEGER*4 TODAY(14)
C   INTEGER*2 LENGTH
C
C   DATA DAYWK/'SUN','MON','TUE','WED','THR','FRI','SAT'/
C   DATA AMPM/'AM','PM'/
C   DATA MONTH /'January ','Febuary ','March ','April ','
+               'May ','June ','July ','August ','
+               'September','October ','November ','December '/
```

C***** DETERMINE SYSTEM DATE AND TIME *****

CALL DATIMX(TODAY)

C*** PUT INTO CHARACTER STRING FOR HEADING AT TOP OF ALL REPORTS *****

C** GET DAY OF WEEK

DATE_AND_TIME(1:3) = DAYWK(TODAY(12))

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

C** GET DAY OF MONTH .. NEED TO CONVERT INTEGER TO CHARACTER FOR STRING

CALL INT_TO_CHAR(6 ,DATE_AND_TIME, TODAY(6))

C** GET MONTH OF THE YEAR...GET CHARACTER NAME FOR THE MONTH AND
C** PLACE IN STRING

LENGTH = LENSTR(MONTH(TODAY(7)))
DATE_AND_TIME(8:LENGTH+7) = MONTH(TODAY(7))

C** GET THE YEAR...CONVERT INTEGER TO CHARACTER AND PLACE IN STRING

DATE_AND_TIME(LENGTH+8:LENGTH+8) = ','
CALL INT_TO_CHAR(LENGTH+14,DATE_AND_TIME,TODAY(8))

C** GET THE HOUR .. NEED TO CONVERT INTEGER TO CHARACTER FOR STRING

CALL INT_TO_CHAR(LENGTH+18,DATE_AND_TIME,TODAY(10))
DATE_AND_TIME(LENGTH+19:LENGTH+19) = ':'

C** GET MINUTES .. NEED TO CONVERT INTEGER TO CHARACTER FOR STRING

CALL INT_TO_CHAR(LENGTH+21,DATE_AND_TIME,TODAY(4))
IF (DATE_AND_TIME(LENGTH+20:LENGTH+20) .EQ. ' ') THEN
DATE_AND_TIME(LENGTH+20:LENGTH+20) = '0'
ENDIF
DATE_AND_TIME(LENGTH+22:LENGTH+22) = ':'

C** GET SECONDS .. NEED TO CONVERT INTEGER TO CHARACTER FOR STRING

CALL INT_TO_CHAR(LENGTH+24,DATE_AND_TIME,TODAY(3))
IF (DATE_AND_TIME(LENGTH+23:LENGTH+23) .EQ. ' ') THEN
DATE_AND_TIME(LENGTH+23:LENGTH+23) = '0'
ENDIF

C** GET AM PM INDICATOR

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

ENDIF

C ***** GET THE LONGITUDE *****

J = INDEX (FIRST_LINE, 'LON-')

IF (J .EQ. 0) THEN

 LONGITUDE = 0.0

ELSE

 READ (FIRST_LINE (J+4:J+12), 1010) LONGITUDE

ENDIF

C ***** GET THE NUMBER OF DISTANCES *****

J = INDEX (FIRST_LINE, 'NRADS-')

IF (J .EQ. 0) THEN

 J = INDEX (FIRST_LINE, 'NRADS')

 IF (J .NE. 0) THEN

 K = INDEX (FIRST_LINE (J:80), '-')

 IF (K .EQ. 0) THEN

 READ (FIRST_LINE(J+5:80), *) NUMBER_OF_DISTANCES

 ELSE

 READ (FIRST_LINE(J+K+1:80), *) NUMBER_OF_DISTANCES

 ENDIF

 ELSE

 NUMBER_OF_DISTANCES = 1

 ENDIF

ELSE

 READ (FIRST_LINE (J+6:J+8), *) NUMBER_OF_DISTANCES

ENDIF

C*****
C**** PLEASE SEE COMMENTS AT BEGINNING OF PROGRAM CONCERNING NRU *
C**** IF THE USER NEEDS TO SPECIFY A DIFFERENT VALUE THAN THAT FOUND*
C**** IN THE POP FILE (NUMBER_OF_DISTANCES ABOVE), THE FOLLOWING *
C**** LINE OF CODE IS WHAT NEEDS TO BE DELETED. THAT'S IT... *

NRU = NUMBER_OF_DISTANCES

C*****
1001 FORMAT(I1)
1002 FORMAT(I2)

C**** READ DISTANCES

 READ(IUNIT,*) (DISTANCES(I), I = 1,NUMBER_OF_DISTANCES)

C**** CONVERT KILOMETERS TO METERS AND THEN

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```
C**** CONVERT TO INTEGER (ADD .5 FOR ROUNDING)
DO 25 I = 1,NUMBER_OF_DISTANCES
    DISTANCES (I) = DISTANCES (I) * 1000.0
    INT_DISTANCES(I+1) = DISTANCES(I) + 0.5
25 CONTINUE
```

```
C**** CALCULATE MIDPOINTS
DO 30 I = 1, NRU
    IDIST(I) = ( INT_DISTANCES(I) + INT_DISTANCES(I+1) ) / 2
    IF ( I .GT. 20 ) GO TO 40
30 CONTINUE
40 RETURN
END
```

```
C-----C
C                                           C
C                                           C
C                               RD21          C
C                               ----          C
C   READS THE ARRAY INPUT OPTIONS (FILE, SKIP, ETC.).  ALLOWS FREE   C
C   FORMAT OF THE NUMBER FOLLOWING THE OPTION.  CALLED BY SEVERAL   C
C   SUBROUTINES AS WELL AS MAIN PROGRAM.                            C
C-----C
```

SUBROUTINE RD21(FMT,IUNIT,IFILE)

```
C   READ(21,FORMAT)A WHERE THE INPUT FORMAT IS A4.
C   IF A.EQ.'FILE'.OR.'SKIP' THEN BACKSPACE UNIT 21 AND
C   READ(21,FORMAT)A,I WHERE THE INPUT FORMAT IS (A4,I8) WITH ANY
C   TRAILING BLANKS IN THE I8 FIELD TREATED AS DELIMITERS
C   C.B.NELSON 30SEP1983
```

CHARACTER*4 B(8),BLANK,FILE,FMT,SKIP

DATA BLANK/' ','/','FILE','FILE','/','SKIP','SKIP'/

```
50 IDATA=0
   READ(21,100) FMT
100 FORMAT(A4)
   IF(FMT.NE.FILE.AND.FMT.NE.SKIP) GO TO 200
   BACKSPACE 21
   READ(21,110) B,IDATA
110 FORMAT(T5,8A1,T5,I8)
   IF(IDATA.EQ.0) GO TO 200
```


PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```
      DO 10 N=1,8
      IF(B(9-N).NE.BLANK) GO TO 20
10    IDATA=IDATA/10
20    WRITE(6,120) FMT, IDATA
120   FORMAT(3X,A4, '-', I4)
200   IF(FMT.NE.FILE) GO TO 201
      IFILE=IDATA
      IF(IFILE.EQ.0) IFILE=IUNIT
      GO TO 50
201   IF(IFILE.NE.IUNIT.AND.IUNIT.NE.21) REWIND IUNIT
      IUNIT=IFILE
      IF(FMT.NE.SKIP) RETURN
      IF(IDATA.EQ.0) GO TO 50
      DO 202 I=1, IDATA
202     READ(IUNIT,1100)
1100    FORMAT(A4)
      GO TO 50
      END
```

```
C-----
C
C          CALC_FOOD_DISTRIBUTIONS
C-----
C
```

```
      SUBROUTINE CALC_FOOD_DISTRIBUTIONS ( DB, DM, FC, CIRCLE,
+          DISTANCES )
C
```

```
C      FOOD.  A UTILITY TO CALCULATE MEAT, MILK, AND VEGETABLE
C      DISTRIBUTION DATA FOR AIRDOS-EPA FROM DENSITIES.
C      C.B.NELSON 13MAR81
```

```
C
C*** THIS IS A MODIFIED VERSION OF THE FOOD UTILITY.  IT IS NO LONGER
C      A STAND ALONE PROGRAM, BUT HAS BEEN ADDED TO PREPAR TO
C      AUTOMATICALLY GENERATE FOOD ARRAYS WITHIN THE PREPAR CODE.
C      J. MCGUE 26OCT88
C***
```

```
C      VARIABLES PASSED
C
```

```
      INTEGER CIRCLE
      REAL DB, DM, FC
      REAL DISTANCES(21)
```


PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

 NOMCT(I,J) = 0.0
 INTFC(I,J) = 0.0
34 CONTINUE

 ENDIF
35 CONTINUE

37 NEWNRL = I

 DO 45 J = NOL,NOU
 DO 40 I = NEWNRL, NRU

 NOBCT(I,J) = DB * A(I) * 1E-4+0.5
 NOMCT(I,J) = DM * A(I) * 1E-4+0.5
 INTFC(I,J) = FC * A(I)

40 CONTINUE
45 CONTINUE

 GO TO 70

C
C** SQUARE GRID CALCULATIONS

50 S=SQSD**2
 DO 60 J = NOL,NOU
 DO 60 I = NEWNRL,NRU
 NOBCT(I,J) = DB * S * 1E-4+0.5
 NOMCT(I,J) = DM * S * 1E-4+0.5
60 INTFC(I,J) = FC * S

C
70 CONTINUE

C
 RETURN
 END

C
C
C GET DB DM FC (FOOD FRACTIONS)
C
C

SUBROUTINE GET_DB_DM_FC (DB, DM, FC, STATE_INDEX)

PREPAR (PREPAR2.FOR) Program File
(continued)

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123456789012

```
      NUM = NUM / 10
      IF ( NUM .GT. 0 ) THEN
        GO TO 10
      ENDIF
C**  ELSE

      RETURN
      END
```

```
C
C                               LENSTR
C
```

```
      INTEGER FUNCTION LENSTR (STRING)
C
C  DETERMINE THE LENGTH OF THE STRING PASSED
C
C  VARIABLES PASSED
C
      CHARACTER STRING *(*)
C
C  LOCAL VARIABLES
C
      INTEGER*2 N, I

      N = LEN(STRING)
      DO 10 I = N, 1, -1
        IF ( STRING(I:I) .NE. ' ' ) THEN
          LENSTR = I
          RETURN
        ENDIF
10  CONTINUE
      LENSTR = 0
      RETURN
      END
```