

THE CLEAN AIR ACT ASSESSMENT PACKAGE-1988  
(CAP-88)  
A DOSE AND RISK ASSESSMENT METHODOLOGY  
FOR RADIONUCLIDE EMISSIONS TO AIR

VOLUME 2

APPENDICES A - H

Prepared By:

Deborah A. Beres  
SC&A, Inc.  
1311 Dolley Madison Blvd.  
McLean, VA 22101

Contract No. 68-D9-0170  
Work Assignment 1-28

Prepared For:

U.S. Environmental Protection Agency  
Office of Radiation Programs  
401 M Street, S.W.  
Washington, D.C. 20460

Larry Gray  
Work Assignment Manager

October 1990

## CONTENTS

LIST OF TABLES . . . . .	ii
LIST OF FIGURES . . . . .	iii
1. INTRODUCTION . . . . .	1-1
1.1 ENVIRONMENTAL TRANSPORT . . . . .	1-1
1.2 ESTIMATION OF DOSE AND RISK . . . . .	1-1
1.3 LIMITATIONS OF THE CAP-88 METHODOLOGY . . . . .	1-2
1.4 VERIFICATION OF THE CAP-88 METHODOLOGY . . . . .	1-2
1.5 REFERENCES FOR PROGRAMS INCLUDED IN CAP-88 . . . . .	1-3
2. HARDWARE/SOFTWARE REQUIREMENTS . . . . .	2-1
3. CAP-88 PACKAGE CONTENTS . . . . .	3-1
4. RUNNING THE CAP-88 PACKAGE . . . . .	4-1
4.1 PREPAR/AIRDOS-EPA INPUT DATA . . . . .	4-1
4.2 PREDA/DARTAB INPUT DATA . . . . .	4-30
4.3 IMPORTANT DIFFERENCES BETWEEN CAP-88 AND EARLIER VERSIONS OF AIRDOS-EPA . . . . .	4-39
5. CAP-88 OUTPUT . . . . .	5-1
6. BACKGROUND INFORMATION AND AIDS TO THE USER . . . . .	6-1
6.1 EPA ENVIRONMENTAL PATHWAY MODELING ASSUMPTIONS . . . . .	6-1
6.2 CALCULATION OF $Q_H$ FOR PLUME RISE . . . . .	6-11
6.3 POPULATION CENSUS FILES . . . . .	6-11
6.4 STABILITY ARRAYS . . . . .	6-12
6.5 CALCULATION OF DAUGHTER INGROWTH FACTORS . . . . .	6-16
7. REFERENCES . . . . .	7-1
APPENDIX A: AIRDOS-EPA (AIRDOS2.FOR) PROGRAM FILE	
APPENDIX B: PREPAR (PREPAR2.FOR) PROGRAM FILE	
APPENDIX C: DARTAB (DARTAB2.FOR) PROGRAM FILE	
APPENDIX D: PREDA (PREDA.FOR) PROGRAM FILE	
APPENDIX E: SAMPLE.JCL FILE	
APPENDIX F: ALLRAD88 DATA FILE	
APPENDIX G: SAMPLE PREPAR INPUT ARRAY	
APPENDIX H: PRDPOP PROGRAM FILE	
APPENDIX I: SAMPLE CAP-88 OUTPUT FILES	

CONTENTS  
(continued)

APPENDIX J: SAMPLE SYNOPSIS FILES

APPENDIX K: SECPOP PROGRAM FILES

APPENDIX L: AVAILABLE STAR DATA SETS

APPENDIX M: CHAIN PROGRAM FILES

LIST OF TABLES

Table 3-1. Elements of CAP-88 . . . . . 3-2

Table 4-1. Run identification data . . . . . 4-2

Table 4-2. Data type names and associated NAMELIST names used by  
PREPAR for scalar and vector variables . . . . . 4-4

Table 4-3. Scalar and vector variable names along with default  
values and associated NAMELIST names . . . . . 4-5

Table 4-4. Definitions of the PREPAR/AIRDOS-EPA options . . . . . 4-12

Table 4-5. Input variables used in PREPAR/AIRDOS-EPA . . . . . 4-14

Table 4-6. Input variables used in PREPAR. . . . . 4-23

Table 4-7. Data type names used by PREPAR for array input . . . . . 4-27

Table 4-8. Definition of arrays input using PREPAR . . . . . 4-28

Table 4-9. NAMELIST STAR variables . . . . . 4-29

Table 4-10. Cattle densities and vegetable crop distributions  
for use with AIRDOS-EPA . . . . . 4-31

Table 4-11. Valid PREDA NAMELIST and variable names . . . . . 4-33

Table 4-12. Possible table types output by DARTAB . . . . . 4-38

Table 6-1. Presumed sources of food for urban and rural sites . . . . . 6-1

Table 6-2. AIRDOS-EPA parameters used for generic site assessments . . . . . 6-3

Table 6-3. Default values used for element dependent factors . . . . . 6-5

Table 6-4. Fatal cancer risk factors for selected radionuclides . . . . . 6-8

Table 6-5. Sample STAR data file . . . . . 6-13

Table 6-6. JCL for creating STAR file from National Climatic  
Data Center data tapes . . . . . 6-15

LIST OF FIGURES

Figure 3-1. CAP-88 program flow chart . . . . . 3-4

APPENDIX C

DARTAB (DARTAB2.FOR) PROGRAM FILE

DARTAB (DARTAB2.FOR) Program File

0            1            2            3            4            5            6            7  
1234567890123456789012345678901234567890123456789012345678901234567890123

```
C
C -----
C DARTAB... PROGRAM TO CALCULATE AND PRINT DOSE AND RISK TABLES -
C FROM AIRDOS-EPA INTERMEDIATE OUTPUT USING DOSE AND RISK FACTORS -
C FROM RADRISK. C.L.BEGOVICH ORNL/CSD JUNE 1980 -
C -
C CHLOC, LOCTAB, AND RDSTOR CORRECTED 06/22/82. C.B.NELSON -
C LOCTAB, PREPDR, PREPHR, AND PREPRF CORRECTED 11/05/82. C.B.NELSON-
C CHLOC CORRECTED 09/19/83. C.B.NELSON -
C RDSTOR MODIFIED FOR SPECIAL VALUES OF INHAL F1 11/28/83 C.B.NELSON-
C MAIN AND RDSTOR MODIFIED FOR EXTENDED HEADER 6/12/84 -
C 6/88 MODIFIED TO DECLARE CHARACTERS AS CHARACTERS NOT REAL * 8 -
C -
C -----
```

CHARACTER\*4 RESP, ANG

```
CHARACTER*8 NUCLID, ORGN, CANC, TOTBOD, RNLOC, OGLOC, GEN, OREP, CREP,
+ RREP, PO218, PB214, BI214, PO214, PULMO, LUNGS, NDP, TDP,
+ PUL, RREPS
```

CHARACTER\*80 TITLE, FOOD\_ARRAY\_INFO

```
CHARACTER*36 DATE_AND_TIME
COMMON / HEADERINFO / DATE_AND_TIME
INTEGER*2 NUMBER_FILES
```

```
REAL*4 CONVRESP(40)
LOGICAL GENEFF, OUTPUT
REAL LLET
INTEGER RTABLE, DTABLE, TABLE(7), FTABLE, PTLOC, HLLOC, FALOC
INTEGER ICRP, IHEAD
LOGICAL SEP_DOSE_LET_TABLES, COMB_DOSE_LET_TABLES,
+ ALL_DOSE_LET_TABLES
LOGICAL SEP_RISK_LET_TABLES, COMB_RISK_LET_TABLES,
+ ALL_RISK_LET_TABLES
LOGICAL RNFLAG
```

C

DIMENSION CONC(4), ILET(2), RREPS(4)

```
DIMENSION RNLOC(10), OGLOC(10), OREP(20), RREP(20), CREP(20),
+ NUCLID(40), CANC(20), ORGN(20), GEN(3), RESP(40)
```

```
COMMON /LOCTBL_CHARS/ RNLOC, OGLOC
COMMON /WORK_LEVEL_CHARS/ OREP, RREP, CREP
COMMON /NAMES_CHARS/ NUCLID, CANC, ORGN, GEN
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

NAMELIST /INPUT/ILOC,JLOC,PLOC,AGEX,ILET,DTABLE,RTABLE,FTABLE,
A OUTPUT,GSCFAC,ICRP,IHEAD
NAMELIST /ORGAN/ORGN,NORGN,TIME
NAMELIST /QFACTR/HLET,LLET
NAMELIST /CANCER/CANC,NCANC,RELABS
NAMELIST /RNUCLD/NUCLID,NONCLD,PSIZE,RESP,GIABS
NAMELIST /LOCTBL/NTLOC,RNLOC,OGLOC,PTLOC,FALOC,HLLOC,LTABLE
NAMELIST/GENTIC/GENEFF,GEN,NGEN,GRFAC,REPPER,GLLET,GHLET
COMMON/DOSE_LET_TABLES/SEP_DOSE_LET_TABLES, COMB_DOSE_LET_TABLES,
+ ALL_DOSE_LET_TABLES
COMMON/RISK_LET_TABLES/SEP_RISK_LET_TABLES, COMB_RISK_LET_TABLES,
+ ALL_RISK_LET_TABLES

```

```

COMMON/COMEX/EXPP(20,20,40,4),POP(20,20),POPFAC,TOTFAC,NOL,NOU,
> NRL,NRU,IDIST(20),ILOC,JLOC
COMMON/COMOR/NORGN,TIME(20),DOSE(20,40,4,2),DTABLE(7)
COMMON/LETFAC/HLET(20),LLET(20)
COMMON/COMGA/NCANC,RELABS(20),RISK(20,40,4,2),RTABLE(7),
> AGEX,YRLL(20,40,4,2)
COMMON/COMRF/REF(20,40,4),FTABLE(7)
COMMON/COMNU/NONCLD,PSIZE(40),GIABS(4,40),
> INDPOP
COMMON/COMLOC/PTLOC(10),FALOC(10),
> HLLOC(10),LTABLE(10),NTLOC
COMMON/COMGEN/NGEN,GDOSE(3,40,4,2),GRISK(3,40,4,2),
> GENEFF,GRFAC(2),REPPER,GLLET(3),GHLET(3),GREF(3,40,4)
COMMON/COMRN/WLRN(20,20),
A RRISK,RREF(2),RYRLL,NOREP,NRREP,NCREP
COMMON/COMUS/ARRAY(2010)

```

```

DATA PO218/'PO-218 '/,PB214/'PB-214 '/,BI214/'BI-214 '/,
A PO214/'PO-214 '/,PULMO/'PULMINARY'/,LUNGS/'LUNGS '/,
B NDP/'*N-P* '/,TDP/'*T-B* '/,PUL/'*PUL* '/
DATA TOTBOD/'TOT BODY'/
DATA RNFLAG/.FALSE./
DATA ANG /' '/

```

EQUIVALENCE (RESP,CONVRESP)

```

DATE_AND_TIME = '
C*** READ DATA AND TIME PASSED BY PREPAR
READ (11,1000) DATE_AND_TIME

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

1000    FORMAT (A)

C\*\*\*    SUPPRESS UNDERFLOW MESSAGES (ERROR CODE 208).

        CALL ERRSET(208,256,-1,1)

C\*\*\*    SET DEFAULT VALUES.

        OUTPUT=.TRUE.

        ILOC=0

        JLOC=0

        PLOC=100.

        AGEX=70.7565

            REPPER=1.41330E-2

            GSCFAC=.5

        ICRP=1

        IHEAD=1

        NORGN=0

        NCANC=1

        CANC(1)=TOTBOD

        DO 10 J=1,7

            DTABLE(J)=0

            RTABLE(J)=0

            FTABLE(J)=0

10    CONTINUE

        RTABLE(6)=4

        SEP\_DOSE\_LET\_TABLES = .FALSE.

        COMB\_DOSE\_LET\_TABLES = .FALSE.

        ALL\_DOSE\_LET\_TABLES = .FALSE.

        SEP\_RISK\_LET\_TABLES = .FALSE.

        COMB\_RISK\_LET\_TABLES = .FALSE.

        ALL\_RISK\_LET\_TABLES = .FALSE.

        ILET(1)=1

        ILET(2)=1

        DO 20 J=1,20

            TIME(J)=70.

            HLET(J)=20.

            LLET(J)=1.

            RELABS(J)=1.

20    CONTINUE

        NGEN=0

        NOREP=4

        NCREP=1

        NRREP=0

        RREPS(1)=PO218

        RREPS(2)=PB214

        RREPS(3)=BI214

        RREPS(4)=PO214

        CREP(1)=PULMO

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

    OREP(1)=LUNGS
    OREP(2)=NDP
    OREP(3)=TDP
    OREP(4)=PUL
    READ(5,11700) TITLE
    READ(5,INPUT)
    IF (IHEAD.EQ.0) ICRP=0
    WRITE(6,11950) IHEAD,ICRP
    WRITE(6,10600) DTABLE,RTABLE,FTABLE
    IF (PLOC.NE.0) WRITE(6,10100)PLOC
    IF (ILET(1).EQ.0.OR.ILET(1).EQ.2) WRITE(6,10200)
    IF (ILET(1).EQ.1.OR.ILET(1).EQ.2) WRITE(6,10300)
    IF (ILET(2).EQ.0.OR.ILET(2).EQ.2) WRITE(6,10400)
    IF (ILET(2).EQ.1.OR.ILET(2).EQ.2) WRITE(6,10500)
C***   ILET - 0 MEANS ONLY TABLES FOR LOW AND HIGH LET SEPARATELY
C***   ILET - 1 MEANS ONLY A TABLE FOR LOW AND HIGH LET COMBINED
C***   ILET - 2 MEANS BOTH SETS OF TABLES
    IF ( ILET(1) .EQ. 0 ) THEN
        SEP_DOSE_LET_TABLES - .TRUE.
    ELSE IF ( ILET(1) .EQ. 2 ) THEN
        ALL_DOSE_LET_TABLES - .TRUE.
    ELSE
        COMB_DOSE_LET_TABLES - .TRUE.
    ENDIF
    IF ( ILET(2) .EQ. 0 ) THEN
        SEP_RISK_LET_TABLES - .TRUE.
    ELSE IF ( ILET(2) .EQ. 2 ) THEN
        ALL_RISK_LET_TABLES - .TRUE.
    ELSE
        COMB_RISK_LET_TABLES - .TRUE.
    ENDIF
    WRITE(6,11900) GSCFAC
C***   READ IN ORGAN PARAMETERS
    40 READ(5,ORGAN)
    WRITE(6,10700) NORGN
    WRITE(6,10800) (ORGN(I),TIME(I),I=1,NORGN)
    IF (ILET(1).GT.0) READ(5,QFACTR)
    IF (ILET(1).GT.0) WRITE(6,10900) (ORGN(I),LLET(I),HLET(I), I=1,
    > NORGN)
C***   READ IN CANCER PARAMETERS
    50 READ(5,CANCER)
    WRITE(6,11000) NCANC
    WRITE(6,11100) (CANC(I),RELABS(I),I=1,NCANC)
C***   READ IN GENETIC PARAMETERS
    READ(5,GENTIC)
    IF(GENEFF) WRITE(6,11800) (GEN(I),I=1,NGEN)

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      IF(GENEFF) WRITE(6,11850) GRFAC,REPPER
11800  FORMAT('OGENETIC DOSES ARE PRINTED FOR: '/
      < 1X,3(1X,A8))
11850  FORMAT(' THE RISK FACTOR (PER RAD/MILLION BIRTHS)',
      < ' FOR GENETIC DOSE ARE :'/
      < 1X,F8.1,' FOR LOW LET, AND'/
      < 1X,F8.1,' FOR HIGH LET, '/
      < ' AND THE REPLACEMENT RATE FOR THE POPULATION IS :'/
      < 1X,E10.4,' YEAR-1')
C***  CONVERT TO /MRAD/BIRTHS
      GRFAC(1)=GRFAC(1)*1.E-9
      GRFAC(2)=GRFAC(2)*1.E-9
C***  READ IN RADIONUCLIDE PARAMETERS
      READ(5,RNUCLD)
      WRITE(6,11200) NONCLD
      WRITE(6,11300)(NUCLID(I),PSIZE(I),RESP(I), (GIABS(J,I),J=1,4),
      > I=1,NONCLD)
      DO 30 I=1,NONCLD
      DO 25 K=1,4
      IF(NUCLID(I).EQ.RREPS(K)) GO TO 27
25     CONTINUE
      GO TO 30
27     NRREP=NRREP+1
      RREP(NRREP)=RREPS(K)
30     CONTINUE
      NTLOC=0
      READ(5,LOCTBL)
      IF(NTLOC.EQ.0) GO TO 55
      WRITE(6,11305) NTLOC
      WRITE(6,11310) (RNLOC(I),OGLOC(I),PTLOC(I),FALOC(I),HLLOC(I),
      > I=1,NTLOC)
55     CONTINUE
11305  FORMAT('0',I2,' LOCATION TABLES ARE TO BE OUTPUT FOR: '/
      > ' NUCLIDE  ORGAN  PATHWAY  QUANTITY  LET'/
      > '          OR CANCER'/)
11310  FORMAT(1X,A8,1X,A8,4X,I2,7X,I2,7X,I2)
C***  READ IN DOSE RATES AND HEALTH RISKS
      CALL RDSTOR(OUTPUT,CONVRESP,IHEAD,ICRP)

C***  CHOOSE LOCATION AND FIND EXPOSURES
      CALL CHLOC(PLOC,CONC,GSCFAC,NUCLID, RNFLAG, ANG)

      IF(NORGN.NE.0) CALL RDORGF(ORGN)

      CALL SUMMRY(GSCFAC )

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      IF (NORGN.LE.0) GO TO 70
C***   DECIDE IF LOW AND HIGH LET ARE TO BE SEPARATE TABLES
C***   AND THEN OUTPUT TABLES
      IDO=0
      DO 60 J=1,7
      IF (DTABLE(J).NE.0) IDO=1
      TABLE(J)=DTABLE(J)
60     CONTINUE
      IF(NTLOC.EQ.0 .OR. IDO.EQ.1) GO TO 67
      DO 65 J=1,NTLOC
      IF(FALOC(J).EQ.1)IDO=1
65     CONTINUE
67     CONTINUE
      IF (IDO.EQ.1) CALL PREPDR(TABLE,TITLE,GSCFAC )

      70 IDO=0
      DO 80 J=1,7
      IF (FTABLE(J).NE.0) IDO=1
80     TABLE(J)=FTABLE(J)
      IF(NTLOC.EQ.0 .OR. IDO.EQ.1) GO TO 87
      DO 85 J=1,NTLOC
      IF(FALOC(J).EQ.3) IDO=1
85     CONTINUE
87     CONTINUE
      IF (IDO.EQ.1) CALL PREPRF(TABLE,TITLE,GSCFAC )

C***   OUTPUT RISK TABLES
      IDO=0
      DO 90 J=1,7
      IF (RTABLE(J).NE.0) IDO=1
90     TABLE(J)=RTABLE(J)
      IF(NTLOC.EQ.0 .OR. IDO.EQ.1) GO TO 97
      DO 95 J=1,NTLOC
      IF(FALOC(J).EQ.2)IDO=1
95     CONTINUE
97     CONTINUE
      IF (IDO.EQ.1) CALL PREPHR(TABLE,TITLE,GSCFAC)

C*****
C*   OUTPUT THE NEW SYNOPSIS REPORT      ADDED 8/88 BY J.MCGUE      *
C*****

      CALL READ_INFO_FOR_SYNOPSIS_RPT( FOOD_ARRAY_INFO, NUMBER_FILES)

      CALL WRITE_SYNOPSIS_REPORT( INDDPOP, ORGN, NORGN,
+                                FOOD_ARRAY_INFO,

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0           1           2           3           4           5           6           7
123456789012345678901234567890123456789012345678901234567890123456789012

```

```

+                               NUMBER_FILES, ANG)

```

STOP

```

10100 FORMAT(' TABLES FOR THE SELECTED INDIVIDUAL WILL BE DONE FOR',
> ' THE LOCATION HAVING', F8.2, ' % OF THE HIGHEST TOTAL RISK.'/)
10200 FORMAT(' DOSE RATE TABLES FOR LOW AND HIGH LET WILL BE ',
> ' PRINTED SEPARATELY.')
10300 FORMAT(' DOSE RATE TABLES COMBINING LOW AND HIGH LET ',
> ' WILL BE PRINTED.')
10400 FORMAT(' HEALTH RISK TABLES FOR LOW AND HIGH LET WILL BE ',
> ' PRINTED SEPARATELY.')
10500 FORMAT(' HEALTH RISK TABLES COMBINING LOW AND HIGH LET ',
> ' WILL BE PRINTED.')
10600 FORMAT('00 INDICATES THE TABLE WILL NOT BE PRINTED'/
A' 1 INDICATES INDIVIDUAL VALUES WILL BE PRINTED'/
B' 2 INDICATES MEAN INDIVIDUAL VALUES WILL BE PRINTED'/
C' 3 INDICATES COLLECTIVE VALUES WILL BE PRINTED'/
D' 4 INDICATES ALL OF THE ABOVE WILL BE PRINTED'//
> ' QUANTITY TABLE NO. 1 2 3 4 5 6 7'//
> '+-----+'//
> ' 1.DOSE RATES ',7(I2,1X)//
> ' 2.HEALTH RISKS ',7(I2,1X)//
> ' 3.RISK EQUIVALENT FACTOR ',7(I2,1X))
10700 FORMAT('OTHER ARE ',I4,' ORGANS TO BE OUTPUT. THEY ARE:')/
10800 FORMAT(1X,'ORGAN',4X,'TIME',4X,'ORGAN',4X,'TIME',4X,
1'ORGAN',4X,'TIME'/
> (3(1X, A8,2X,F4.0,2X)))
10900 FORMAT('0 ORGAN DOSE EQUIVALENT FACTORS '/
A ' LOW LET HIGH LET'/
A (2X,A8,F15.5,1X,F15.4))
11000 FORMAT('OTHER ARE ',I4,' CANCERS TO BE OUTPUT.'/
> ' A 1 INDICATES ABSOLUTE RISK; A 2 IS RELATIVE RISK.')
11100 FORMAT(' CANCER CANCER CANCER CANCER'/ (1X,4(A8,
> 1X,F2.0,1X)))
11200 FORMAT('OTHER ARE ',I4,' RADIONUCLIDES TO BE OUTPUT.')
11300 FORMAT(' NUCLIDE PARTICLE SIZE CLEARANCE CLASS ',
> 20X,'G.I. ABSORPTION FRACTION'/49X,'STOMACH',8X,'SI',13X,'ULI',
> 12X,'LLI'/ (1X,A8,1X,F10.5,6X,10X,A1,4X,4F15.5))
11700 FORMAT(A80)
11900 FORMAT('0THE GROUND SURFACE CORRECTION FACTOR IS ',F5.2/)
11950 FORMAT('0IHEAD=',I2/' ICRP=',I3/)
END

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
12345678901234567890123456789012345678901234567890123456789012

```
C*****
C
C
C
C           INPUT AND DATA PREPARATION  ROUTINES
C
C
C
C*****
```

```
C-----
C
C           SUBROUTINE RDSTOR
C
C-----
```

SUBROUTINE RDSTOR(OUTPUT, CONVRESP, IHEAD, ICRP)

```
C***   THIS SUBROUTINE READS AND STORES DOSE RATES
C***   AND HEALTH RISKS FOR ORGANS AND CANCERS
C***   THIS SUBROUTINE WAS CORRECTED 6/88 BY JOAN MCGUE..GO TO 40
C***   WAS JUMPING INTO THE LOOP DO 50
C***   THE CHARACTER ARRAYS O,C,OG WERE PULLED OUT OF THE COMMON COMUS
C***   AND MADE LOCAL 6/88
```

```
CHARACTER*8 NUC,NUCLID,ORGN,CANC,GEN,OREP,CREP,RREP
LOGICAL GENEFF,OUTPUT
CHARACTER*36 DATE_AND_TIME
COMMON / HEADERINFO / DATE_AND_TIME
REAL*8 O,C,OG
REAL*8 CONVORGN(40),CONVCANC(40),CONVGEN(3),CONVCREP(20)
INTEGER IHEAD,ICRP
REAL*4 RESPIN,CONVRESP(40)
DIMENSION OREP(20), RREP(20), CREP(20),
+          NUCLID(40), CANC(20), ORGN(20), GEN(3)

COMMON /WORK_LEVEL_CHARS/ OREP, RREP, CREP
COMMON /NAMES_CHARS/ NUCLID, CANC, ORGN, GEN
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

INTEGER SAVED\_INDEX

```
COMMON/COMOR/NORGN, TIME(20), DOSE(20,40,4,2), DTABLE(7)
COMMON/COMRN/WLRN(20,20), RRISK, RREF(2), YRLL, NOREP, NRREP, NCREP
COMMON/COMCA/NCANC, RELABS(20), RISK(20,40,4,2), RTABLE(7),
> AGEX, YRLL(20,40,4,2)
COMMON/COMRF/REF(20,40,4), FTABLE(7)
COMMON/COMNU/NONCLD, PSIZE(40), GIABS (4,40),
> INDPOP
COMMON/COMUS/D(2,40), R(2,40), RF(40), YLL(2,40),
> G(2,3), DCHK(20,40), RCHK(20,40), GCHK(3,40)
COMMON/COMGEN/NGEN, GDOSE(3,40,4,2), GRISK(3,40,4,2),
> GENEFF, GRFAC(2), REPPER, GLLET(3), GHLET(3), GREF(3,40,4)
```

DIMENSION GIIN(4), O(40), C(40), OG(3)

```
LOGICAL*1 FAL, TRU, DCHK, RCHK, GCHK, IW
DATA TRU/.TRUE./, FAL/.FALSE./, ICRPIN/0/
DATA SAVED_INDEX / 1/
```

```
EQUIVALENCE (ORGN, CONVORGN), (CANC, CONVCANC), (GEN, CONVGEN),
+ (CREP, CONVCREP)
```

```
C*** ZERO OUT ALL ARRAYS
NDO=NORGN+1
DO 35 N=1,2
DO 30 J=1,4
DO 30 K=1,NONCLD
DO 10 L=1,NDO
DOSE(L,K,J,N)=0.0
IF(L.GT.NGEN) GO TO 10
GDOSE(L,K,J,N)=0.0
10 CONTINUE
NDO=NCANC+1
DO 20 L=1,NDO
RISK(L,K,J,N)=0.0
YRLL(L,K,J,N)=0.0
REF(L,K,J)=0.0
IF(L.GT.NGEN) GO TO 20
GRISK(L,K,J,N)=0.0
GREF(L,K,J)=0.0
20 CONTINUE
30 CONTINUE
35 CONTINUE
DO 38 L=1,40
DO 37 K=1,20
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

          DCHK(K,L)=TRU
          RCHK(K,L)=TRU
37      CONTINUE
          DO 39 K=1,3
          GCHK(K,L)=TRU
39      CONTINUE
38      CONTINUE
C***    READ FIRST RECORD
41     IF (IHEAD.EQ.0) READ(25,END=180) NUC,SIZEIN,RESPIN,GIIN,TIMIN,
        >  IND
          IF (IHEAD.EQ.1) READ(25,END=180) NUC,SIZEIN,RESPIN,GIIN,TIMIN,
        >  IND,ICRPIN

          IF (ICRPIN.EQ.1.AND.ICRP.EQ.0) GO TO 125
          IFIND=0
C***    CHECK TO SEE IF THE RADIONUCLIDE IS IN OUTPUT LIST
          SAVED_INDEX = 1
40      CONTINUE
          DO 50 K= SAVED_INDEX, NONCLD
          IF (NUC.EQ.NUCLID(K)) GO TO 60
50      CONTINUE
          IF(IFIND.EQ.0) GO TO 125
          GO TO 41
C***    FIND OUT WHAT TYPE OF RECORD FOLLOWS
60      SAVED_INDEX = K+1

          IRA=IND/10
          IF(IRA.GT.2) GO TO 401
          ICHOS=IND-IRA*10
          IF (ICHOS.NE.2) GO TO 75
          IF (ABS(GIIN(2)-GIABS(2,K)).GT.1.E-2*GIABS(2,K)) GO TO 40
75     IF (ICHOS.NE.3) GO TO 80
          IF (ABS(SIZEIN-PSIZE(K)).GT.1.E-2*PSIZE(K)) GO TO 40

          IF (RESPIN.NE.CONVRESP(K)) GO TO 40
          IF(GIABS(1,K).NE.0..AND.ABS(GIABS(1,K)-GIIN(2)).GT.
        < 1E-2*GIABS(1,K)) GO TO 40
80     IF (IND.LE.5) GO TO 130
C***    THE NEXT TWO RECORDS CONTAIN CANCERS AND RISKS
          IF(IFIND.NE.0) GO TO 81
          READ(25) NC, ILET, (C(I), I=1, NC)
          READ(25) ((R(L,I), L=1, ILET), I=1, NC)
          READ(25) ((YLL(L,I), L=1, ILET), I=1, NC)
          READ(25) (RF(I), I=1, NC), TRF

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

81        IFIND=1
          REF(NCANG+1,K,ICHOS-1)=TRF+REF(NCANG+1,K,ICHOS-1)
C***      CHECK TO SEE IF THE CANCER IS IN OUTPUT LIST
          DO 110 I=1,NC
          DO 90 J=1,NCANC

          IF (C(I).EQ.CONVCANC(J)) GO TO 100
90        CONTINUE
          GO TO 110
C***      THE CANCER NAMES MATCH, NOW DO WE HAVE RELATIVE OR ABS RISK
100       IF (RELABS(J).NE.IRA) GO TO 110
C***      A MATCH, STORE THE RISK
          RCHK(J,K)=FAL
          RISK(J,K,ICHOS-1,1)=R(1,I)
          YRLL(J,K,ICHOS-1,1)=YLL(1,I)
          IF (ILET.LE.1) GO TO 105
          RISK(J,K,ICHOS-1,2)=R(2,I)
          YRLL(J,K,ICHOS-1,2)=YLL(2,I)
105       REF(J,K,ICHOS-1)=RF(I)
          IF(IND.NE.3) GO TO 110
          DO 106 L=1,NRREP
          IF(NUCLID(K).EQ.RREP(L)) GO TO 107
106       CONTINUE
          GO TO 110
107       DO 108 L=1,NCREP

          IF(C(I).EQ.CONVCREP(L)) GO TO 109
108       CONTINUE
          GO TO 110
109       RISK(J,K,2,2)=0.0
          YRLL(J,K,2,2)=0.0
110       CONTINUE
C***      THE RISKS HAVE BEEN STORED OR SKIPPED, GO TO NEXT RECORD
          GO TO 40
C***      SKIP THE RECORDS
125       READ(25)
          READ(25)
          IF(IND.LE.5 .OR. IND.GT.90) GO TO 41
          READ(25)
          READ(25)
          GO TO 41
C***      THE NEXT TWO RECORDS CONTAIN ORGANS AND DOSE RATES
130       IF(IFIND.NE.0) GO TO 131
          READ(25) NO,ILET,(O(I),I=1,NO)

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      READ(25) ((D(L,I),L=1,ILET),I=1,NO)
131   IFIND=1
C***   FOR INTERNAL DOSES, ALSO CHECK OTHER NUCLIDE PARAMETERS
C***   CHECK TO SEE IF THE ORGANS ARE ON OUTPUT LIST
      DO 170 I=1,NO
      DO 150 J=1,NORGN

      IF (O(I).EQ.CONVORGN(J)) GO TO 160
150   CONTINUE
      GO TO 170
C***   A MATCH SO STORE THE DOSE
160   IF(IND.GT.3) GO TO 159
      IF(ABS(TIMIN-TIME(J)).GT.1.E-2*TIME(J)) GO TO 40
159   DOSE(J,K,IND-1,1)=D(1,I)
      IF (ILET.GT.1) DOSE(J,K,IND-1,2)=D(2,I)
      DCHK(J,K)=FAL
      IF(IND.NE.3) GO TO 170
      DO 162 L=1,NRREP
      IF(NUCLID(K).EQ.RREP(L)) GO TO 163
162   CONTINUE
      GO TO 170
163   DO 165 L=1,NOREP
      IF(ORGN(J).EQ.OREP(L)) GO TO 167
165   CONTINUE
      GO TO 170
167   DOSE(J,K,2,2)=0.0
170   CONTINUE
      IF(IND.LT.4) GO TO 40
      DO 172 I=1,NO
      DO 152 J=1,NGEN

      IF(O(I).EQ.CONVGEN(J)) GO TO 161
152   CONTINUE
      GO TO 172
161   GDOSE(J,K,IND-1,1)=D(1,I)*30.
      IF(ILET.GT.1)GDOSE(J,K,IND-1,2)=D(2,I)*30.
172   CONTINUE
      GO TO 40
C***   DOSES HAVE BEEN SKIPPED OR STORED, GO TO NEXT RECORD
180   CONTINUE
      IW=TRU
      DO 200 K=1,NONCLD
      DO 200 J=1,NORGN
      IF(.NOT.DCHK(J,K)) GO TO 200

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      IF(IW) WRITE(27,900)
      WRITE(27,901) ORGN(J),NUCLID(K)
      IW=FAL
200    CONTINUE
900    FORMAT('0 THE FOLLOWING NUCLIDES AND ',
      < 'ORGAN DOSE FACTORS WERE NOT FOUND',
      > ' IN THE INPUT DATA SETS: '/
      > ' ORGAN NUCLIDE' /)
901    FORMAT(1X,A8,1X,A8)
      IW=TRU
      DO 300 K=1, NONCLD
      DO 300 J=1, NCANC
      IF(.NOT.RCHK(J,K)) GO TO 300
      IF(IW) WRITE(27,902)
      IW=FAL
902    FORMAT('0THE FOLLOWING NUCLIDES AND ',
      < 'CANCER RISK FACTORS WERE NOT FOUND',
      > ' IN THE INPUT DATA SETS: '/
      > ' CANCER NUCLIDE' /)
      WRITE(27,901) CANC(J),NUCLID(K)
300    CONTINUE
      IF(.NOT.GENEFF) GO TO 706
      IW=TRU
      DO 400 K=1, NONCLD
      DO 400 J=1, NGEN
      IF(.NOT.GCHK(J,K)) GO TO 400
      IF(IW) WRITE(27,903)
      IW=FAL
903    FORMAT('0THE FOLLOWING NUCLIDES AND ',
      < 'GENETIC DOSE FACTORS WERE NOT',
      > ' FOUND IN THE INPUT DATA SETS: '/
      > ' GEN.DOSE NUCLIDE')
      WRITE(27,901) GEN(J),NUCLID(K)
400    CONTINUE
      DO 625 J=1, NONCLD
      DO 625 K=3,4
      NGN=NGEN-1
      AVG=0.0
      DO 600 I=1, NGN
      AVG=AVG+GDOSE(I,J,K,1)
600    CONTINUE
      GDOSE(NGEN,J,K,1)=AVG/FLOAT(NGN)
625    CONTINUE
      DO 700 L=1,2
      DO 700 I=1,4
      DO 700 K=1, NONCLD

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

GRISK(1,K,I,L)=GDOSE(3,K,I,L)*GRFAC(L)
700 CONTINUE
DO 705 I=1,4
DO 705 K=1, NONCLD
GREF(1,K,I)=(GRISK(1,K,I,1)+GRISK(1,K,I,2))/(30.*GRFAC(1))
705 CONTINUE
706 IF(OUTPUT) CALL FACOUT( DATE_AND_TIME)

RETURN
401 IF(IRA.NE.9) GO TO 800
ICHOS=0
IF(IND.EQ.98) ICHOS=3
IF(IND.EQ.99) ICHOS=2
IF(ICHOS.EQ.0) GO TO 125
IF(.NOT.GENEFF) GO TO 125
IF(ICHOS.NE.2) GO TO 507
IF(ABS(GIIN(2)-GIABS(2,K)).GT.1.E-2*GIABS(2,K)) GO TO 40
507 IF(ICHOS.NE.3) GO TO 510
IF(ABS(SIZEIN-PSIZE(K)).GT.1.E-2*PSIZE(K)) GO TO 40

IF(RESPIN.NE.CONVRESP(K)) GO TO 40
IF(GIABS(1,K).EQ.0..AND.GIIN(2).GT.1..OR.
& GIABS(1,K).NE.0..AND.ABS(GIIN(2)-GIABS(1,K)).GT.
& 1E-3*GIABS(1,K)) GO TO 40
510 IF(IFIND.NE.0) GO TO 511
READ(25) NG,LET,(OG(I),I=1,NG)
READ(25) ((G(L,I),L=1,LET),I=1,NG)
511 IFIND=1
520 DO 530 I=1,NG
DO 525 J=1,NGEN

IF(OG(I).EQ.CONVGEN(J)) GO TO 540
525 CONTINUE
GO TO 530
540 GDOSE(J,K,ICHOS-1,1)=G(1,I)
IF(LET.GT.1) GDOSE(J,K,ICHOS-1,2)=G(2,I)
GCHK(J,K)=FAL
530 CONTINUE
GO TO 40
800 IF(IND.NE.33) GO TO 125
READ(25) NC,ILET,(C(I),I=1,NC)
READ(25) ((R(L,I),L=1,ILET),I=1,NC)
READ(25) ((YLL(L,I),L=1,ILET),I=1,NC)
READ(25) (RF(I),I=1,NC),TRF
IFIND=1
CONVCREP(1)=C(1)

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0 1 2 3 4 5 6 7  
12345678901234567890123456789012345678901234567890123456789012

```
NCREP=NC
RRISK=R(1,1)
RYRLL=YLL(1,1)
RREF(1)=RF(1)
RREF(2)=TRF
GO TO 40
END
```

```
C-----
C
C          SUBROUTINE CHLOC
C
C-----
```

SUBROUTINE CHLOC(PLOC, CONC, GSCFAC, NUCLID, RNFLAG, ANG)

```
CHARACTER*8 NUC, NUCLID, RADON
CHARACTER*4 ANGLE, ANG
CHARACTER*36 DATE_AND_TIME
COMMON / HEADERINFO / DATE_AND_TIME
```

DIMENSION NUCLID(40)

```
LOGICAL NFLAG(40), RNFLAG
DIMENSION CONC(4), FAC(4), IOR(4), GI(4)
COMMON/COMEX/EXPP(20,20,40,4), POP(20,20), POPFAC, TOTFAC, NOL, NOU,
> NRL, NRU, IDIST(20), ILOC, JLOC
COMMON/COMCA/NCANC, RELABS(20), RISK(20,40,4,2), RTABLE(7),
> AGEX, YRLL(20,40,4,2)
COMMON/COMRF/REF(20,40,4), FTABLE(7)
COMMON/COMNU/NONCLD, PSIZE(40), GIABS(4,40), IND
COMMON/COMRN/WLRN(20,20), RRISK, RREF(2), RYRLL, NOREP, NRREP, NCREP

COMMON/COMUS/TRISK(20,20)
```

```
C RSKLIN AND POPLIN WERE ADDED FOR THE NEW REPORT 2/8/88. THEY
C ARE SEQUENTIAL ARRAYS WHICH WILL STORE THE VALUES OF THE POPULATION
C AND RISKS IN LINEAR ORDER AS THE SORTED RISK ARRAY. THE VALUES ARE
C STORED WITH THE DIRECTIONS DISTANCES TOGETHER IE. N 1, N 2, N 3 ETC.
```

```
DIMENSION ANGLE(16)
REAL SRISK(400), RSKLIN(400)
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

REAL      OUTPUT(20,20)
INTEGER   POPLIN(400)
LOGICAL   USER_SUPPLIED_IJLOC
  
```

C\*\*\* FOLLOWING VARS AND COMMON ADDED FOR NEW SYNOPSIS REPORT 9/1988

```

REAL PCI_PER_LITER_CONC (20,20)
REAL WLI, MAX_PCI_LITERS, FATAL_CANCER_RISK, ORGAN_DOSES(20),
+   PATHWAY_DOSES(4), NUC_DOSES(36)
INTEGER LOC_DIST
COMMON / IND_RESULTS / WLI, LOC_DIST, MAX_PCI_LITERS,
+   FATAL_CANCER_RISK, ORGAN_DOSES,
+   PATHWAY_DOSES, NUC_DOSES

REAL EFFECT_PERSON_REM, PERSON_WORKING_LEVEL, POP_ORGAN_DOSES(20)
INTEGER IO(400), NLOC, NOP

COMMON / POPU_RESULTS / POPLIN, RSKLIN, SRISK, IO, NLOC, NOP,
+   EFFECT_PERSON_REM, PERSON_WORKING_LEVEL,
+   POP_ORGAN_DOSES
  
```

C\*\*\*

```

DATA USER_SUPPLIED_IJLOC /.FALSE./
DATA FAC/2*1.E-5,10.,1000./
DATA BRTHRT/.83E6/
DATA RADON/'RN-222 '/
DATA IOR/3,4,1,2/
DATA NFLAG/40*.FALSE./
DATA OUTPUT/400*0.0/, PCI_PER_LITER_CONC / 400 * 0.0 /

DATA ANGLE/'N ', 'NNE ', 'NE ', 'ENE ', 'E ',
A 'ESE ', 'SE ', 'SSE ', 'S ', 'SSW ', 'SW ', 'WSW ',
B 'W ', 'WNW ', 'NW ', 'NNW '/
  
```

C\*\*\* THE FOLLOWING EQUIVALENCE WAS DELETED ... COULD SEE NO REASON FOR IT  
C\*\*\* EXCEPT TO SAVE SPACE, HOWEVER THE VALUES IN SRISK NEED TO BE SAVED FOR  
C\*\*\* THE NEW SYNOPSIS REPORT. 9/1988

CC EQUIVALENCE(TRISK(1,1),SRISK(1))

```

FAC(4)=FAC(4)*GSCFAC
NOP=JLOC+(ILOC-1)*20
DO 2 K=1,20
  
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      DO 2 L=1,20
      POP(L,K)=1.0
      WLRN(L,K)=0.0
      TRISK(L,K)=0.0
2     CONTINUE
      DO 10 K=1,40
      DO 10 J=1,20
      DO 5 I=1,4
      DO 5 L=1,20
5     EXPP(L,J,K,I)=0.0
10    CONTINUE
20   READ(26,END=110) NUC,SIZE,RSP,GI,TIM,IND
      READ(26)NOL,NOU,NRL,NRU,(IDIST(I),I=NRL,NRU)
      DO 30 J=1,NONCLD
      IF(NFLAG(J)) GO TO 30
      IF(NUC.EQ.NUCLID(J)) GO TO 40
30   CONTINUE
      GO TO 90
40   NFLAG(J)=.TRUE.
      IF (NUC.EQ.RADON) GO TO 160
      DO 70 ILO=NOL,NOU
      DO 70 JLO=NRL,NRU
      READ(26) CONC
      DO 50 NC=1,NCANC
      DO 50 L=1,4
      DO 50 N=1,2
      TRISK(JLO,ILO)=TRISK(JLO,ILO)+RISK(NC,J,IOR(L),N)*
A    CONC(L)*FAC(IOR(L))
50   CONTINUE
      DO 60 I=1,4
      EXPP(JLO,ILO,J,IOR(I))=CONC(I)
60   CONTINUE
70   CONTINUE
80   IF (IND.EQ.1) READ(26) POP
      GO TO 20
90   DO 100 ILO=NRL,NRU
      DO 100 JLO=NOL,NOU
      READ(26)
100  CONTINUE
      IF (IND.EQ.1) READ(26)
      GO TO 20

110  IF (ILOC.NE.0.AND.JLOC.NE.0) THEN
      USER_SUPPLIED_IJLOC = .TRUE.
      GO TO 130
      ENDIF

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

LENO=NOU-NOL+1
LENR=NRU-NRL+1
NLOC=LENO*LENR
J=0
IF (IND.EQ.1) THEN
C  CALCULATE NUMBER DEATHS PER YEAR FOR NEW REPORT 2/8/88
C*** IT SHOULD BE NOTED HERE THAT WHEN MORE TIME PERMITS THIS SHOULD BE
C*** MOVED TO SUBROUTINE POP_RESULTS*****

      CALL CALC_DEATHS_PER_YEAR(OUTPUT,NOL,NOU,NRL,NRU,NONCLD,
+                               EXPP,AGEX,RISK,NCANC,RRISK,
+                               RNFLAG,WLRN)
      ENDIF
      DO 120 ILO=NOL,NOU
      DO 120 JLO=NRL,NRU
        J = J + 1
        IF(IND .EQ. 1) THEN
          IF ( POP(ILO,JLO) .NE. 0.0 ) TRISK(JLO,ILO) =
+          TRISK(JLO,ILO)/POP(ILO,JLO)
C  THE POPULATIONS ARRAY STORES THE DIRECTIONS AND DISTANCES IN
C  THE OPPOSITE ORDER OF THE TRISK ARRAY. IE. N 1, NW 1, NNE 3 ETC.
C  SO ILO AND JLO ARE REVERSED IN ORDER FOR POPLIN TO BE CORRECT.
C*** THESE SHOULD BE MOVED ALSO TO SUB. POP_RESULTS.

          POPLIN(J) = POP(ILO,JLO)
          RSKLIN(J) = OUTPUT(JLO,ILO)
        ENDIF

        SRISK(J) = TRISK(JLO,ILO)
        IO(J) = J
120 CONTINUE
      CALL VSORTP(SRISK,NLOC,IO)
      NOP=NLOC*PLOC*.01+.5
      IF (NOP.GE.NLOC) NOP=NLOC
      IF (NOP.LE.1) NOP=1
CC  OUTPUT THE NEW REPORT ***LEAVE IN FOR NOW FOR DEBUG
CC  IF ( IND .EQ. 1 ) THEN
CC      CALL RISK_FREQ_REPORT(SRISK,POPLIN,RSKLIN,IO,NLOC,NOP)
CC  ENDIF

      LOC=IO(NOP)
      ILOC=(LOC-1)/LENR+1
      JLOC=LOC-(ILOC-1)*LENR
      IF (JLOC.EQ.0) JLOC=NRU
      ILOC=ILOC+NOL-1

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      JLOC=JLOC+NRL-1
130 POPFAC=1.
      TOTFAC=1.
      IF (IND.NE.1) GO TO 150
      IF (POP(ILOC,JLOC).NE.0.0) POPFAC=1./POP(ILOC,JLOC)
      TOTPOP=0.0
      DO 140 K1=NRL,NRU
      DO 140 K2=NOL,NOU
140 TOTPOP=TOTPOP+POP(K2,K1)
      TOTFAC=1./TOTPOP
150   ANG=ANGLE(MOD(17-ILOC,16)+1)
      WRITE(27,101)DATE_AND_TIME
101  FORMAT('1',T10,'DATE',2X,A)

C***   SAVE VALUES FOR SYNOPSIS REPORT

      IF ( USER_SUPPLIED_IJLOC ) THEN
      WRITE(27,10000) IDIST(JLOC),ANG,TRISK(JLOC,ILOC)
      FATAL_CANCER_RISK = TRISK(JLOC,ILOC)
      ELSE
      WRITE(27,10000) IDIST(JLOC),ANG,SRISK(NOP)
      FATAL_CANCER_RISK = SRISK(NOP)
      ENDIF

      LOC_DIST = IDIST(JLOC)
      MAX_PCI_LITERS = PCI_PER_LITER_CONC(ILOC,JLOC)
C***

      RETURN
160 CONTINUE
C***   SPECIAL SECTION FOR RADON-222
      RNFLAG = .TRUE.
      DO 190 ILO=NOL,NOU
      DO 190 JLO=NRL,NRU
      READ(26) WFRAC,WLEVEL, PCI_PER_LITER_CONC(ILO,JLO)
170  EXPP(JLO,ILO,J,3)=WLEVEL*1.E-6/(10.*WFRAC)
      EXPP(JLO,ILO,J,2)=WLEVEL*BRTHRT*8760./(10.*WFRAC)
      EXPP(JLO,ILO,J,1)=0.0
      EXPP(JLO,ILO,J,4)=0.0
      DO 180 N=1,2
      DO 180 K=1,4
      DO 180 NC=1,NCANC
      TRISK(JLO,ILO)=TRISK(JLO,ILO)+RISK(NC,J,K,N)*EXPP(JLO,ILO,J,K)*
A   FAC(K)
180  CONTINUE
      WLRN(JLO,ILO)=WLEVEL

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0 1 2 3 4 5 6 7  
12345678901234567890123456789012345678901234567890123456789012

```
TRISK(JLO,ILO)-TRISK(JLO,ILO)+WLEVEL*RRISK
190 CONTINUE
WRITE(27,20100)
20100 FORMAT(' THERE ARE NO GROUND SURFACE CONCENTRATION',
> ' OR INGESTION RATE EXPOSURES FOR RN-222.')
```

GO TO 80

```
10000 FORMAT(' THE LOCATION USED FOR THE SELECTED INDIVIDUAL',
> ' EXPOSURE IS ',/' --->',I7,' METERS ',A4,' FROM THE SOURCE. '/
> ' THE LIFETIME FATAL CANCER RISK IS ',1PE10.2,')
END
```

---

C  
C  
C SUBROUTINE PREPDR  
C  
C

---

SUBROUTINE PREPDR(TABLE,TITLE,GSCFAC)

C CORRECTED FOR LOCTAB ORGAN/CANCER "SUM" OPTION. CBN 11/05/82  
C\*\*\* THIS ROUTINE PREPARES DOSE RATES TO BE OUTPUT.

CHARACTER\*8 ORGN,NUCLID,RNLOC,OGLOC,ORC,ORG, LAST,GEN,OREP,RREP,  
+ CREP,GON, CANC

CHARACTER\*80 TITLE, NOTE, NOT2  
CHARACTER\*8 TITL2, TLET  
CHARACTER\*32 NUN  
CHARACTER\*36 DATE\_AND\_TIME  
COMMON / HEADERINFO / DATE\_AND\_TIME

CHARACTER\*40 TITLA, TITLB, TITLGA, TITLGB

DIMENSION RNLOC(10), OGLOC(10), OREP(20), RREP(20), CREP(20),  
+ NUCLID(40), CANC(20), ORGN(20), GEN(3)

COMMON /LOCTBL\_CHARS/ RNLOC, OGLOC  
COMMON /WORK\_LEVEL\_CHARS/ OREP, RREP, CREP  
COMMON /NAMES\_CHARS/ NUCLID, CANC, ORGN, GEN

LOGICAL SEP\_DOSE\_LET\_TABLES, COMB\_DOSE\_LET\_TABLES,

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

+            ALL_DOSE_LET_TABLES
COMMON/DOSE_LET_TABLES/SEP_DOSE_LET_TABLES, COMB_DOSE_LET_TABLES,
+            ALL_DOSE_LET_TABLES
COMMON/HEAD/ORC
COMMON/COMRN/WLRN(20,20),RRISK,RREF(2),RYRLL,NOREP,NRREP,NCREP
COMMON/COMWOR/FACO(20,4)
COMMON/COMGEN/NGEN,GDOSE(3,40,4,2),GRISK(3,40,4,2),
>            GENEFF,GRFAC(2),REPPER,GLLET(3),GHLET(3),GREF(3,40,4)
COMMON/COMLOC/PTLOC(10),FALOC(10),
>            HLLOC(10),LTABLE(10),NTLOC
REAL LLET
INTEGER TABLE,FALOC,HLLOC,PTLOC
DIMENSION TABLE(7),TITLA(3),FACD(4,3),TLET(3),
>            ITAB(7),TITLB(3),TITLGA(3),TITLGB(3),
>            FACG(4,3),OFAC(20,20),NUN(3)
LOGICAL IDO,GENEFF
COMMON/COMOR/NORGN,TIME(20),DOSE(20,40,4,2)
COMMON/LETFAC/HLET(20),LLET(20)
COMMON/COMNU/NONCLD,PSIZE(40),GIABS(4,40),
>            INDDOP
DATA ORG/' ORGAN ',LAST/'WT. SUM ',GON/' GONAD '/
DATA NUN/'(WORKING LEVEL) ',
+            '(WORKING LEVEL) ',
+            '(PERSON WORKING LEVEL) '/

DATA NOTE/'RADON DAUGHTER EXPOSURE:
+            '/
DATA NOT2/'
+            '/

DATA TITLA/'INDIVIDUAL DOSE RATE (MRAD/YEAR) ',
+            'MEAN INDIVIDUAL DOSE RATE (MRAD/YEAR) ',
+            'COLLECTIVE DOSE RATE (PERSON RAD /YEAR) '/

DATA TITLB/'INDIVIDUAL DOSE EQ. RATE(MREM/YEAR) ',
+            'MEAN INDIVIDUAL DOSE EQ. RATE (MREM/YR) ',
+            'COLLECTIVE DOSE EQ. (PERSON REM /YEAR) '/

DATA TITLGA/'INDIVIDUAL GENETIC DOSE (MRAD) ',
+            'MEAN INDIVIDUAL GENETIC DOSE (MRAD) ',
+            'COLLECTIVE GENETIC DOSE (PERSON RAD) '/

DATA TITLGB/'INDIVIDUAL GENETIC DOSE EQ. (MREM) ',

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

+           'MEAN INDIVIDUAL GENETIC DOSE EQ. (MREM) ',
+           'COLLECTIVE GENETIC DOSE EQ. (PERSON REM)'/

DATA TLET/'LOW LET ', 'HIGH LET', '          '/

DATA FACD/1.,1.,1.,100.,1.,1.,1.,100.,.001,.001,.001,.1/
DATA FACG/1.,1.,1.,100.,1.,1.,1.,100.,.001,.001,.001,.1/
C*****?????????
C***      MULTIPLY DOSES BY EXPOSURES
C***      PREPARE TABLE FOR HIGH AND LOW LET SEPARATELY
          DO 5 J=1,3
          FACD(4,J)=FACD(4,J)*GSCFAC
          FACG(4,J)=FACG(4,J)*GSCFAC
5         CONTINUE
          DO 50 L=1,2
          TITL2 = TLET(L)
          DO 40 IT=1,3
          DO 42 J=1,NOREP
          DO 42 K=1,1
          OFAC(J,K)=1.
42        CONTINUE
          IDO=.FALSE.
          IF(COMB_DOSE_LET_TABLES) GO TO 35
          DO 30 K=1,7
          ITAB(K)=0
          IF (TABLE(K).NE.IT.AND.TABLE(K).NE.4) GO TO 30
          ITAB(K)=1
          IDO=.TRUE.
30       CONTINUE
35       CONTINUE
          IF (ITAB(6).NE.0) ITAB(6)=0
          ORC=ORG
          IF (IDO) CALL MULT(IT,FACD(1,IT),DOSE(1,1,1,L),NORGN,ORGN,TITLE,
> TITLA(IT),ITAB,TITL2,NOTE,NUN(IT),OFAC,OREP,NOREP,20,NUCLID,
+ NONCLD )
          ORC=GON
          IF(IDO.AND.GENEFF) CALL MULT(IT,FACG(1,IT),GDOSE(1,1,1,L),NGEN,
> GEN,TITLE,TITLGA(IT),ITAB,TITL2,NOT2,NOT2,0.,0.,0,3,NUCLID,
+ NONCLD )
C        IF(IT.EQ.2) GO TO 38
          IF(NTLOC.EQ.0) GO TO 38
          DO 45 LL=1,NTLOC
          IF(FALOC(LL).NE.1) GO TO 45
          IF(HLLOC(LL).EQ.1) GO TO 45
          IF(LTABLE(LL).NE.IT .AND. LTABLE(LL).NE.4) GO TO 45
          CALL LOCTAB(IT,RNLOC(LL),PTLOC(LL),OGLOC(LL),FACD(1,IT),

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

A DOSE(1,1,1,L),TITLA(IT),TITL2,ORGN,NORGN+1,20,1,NUCLID,
+ DATE_AND_TIME)

      IF(GENEFF) CALL LOCTAB(IT,RNLOC(LL),PTLOC(LL),OGLOC(LL),
<  FACG(1,IT),GDOSE(1,1,1,L),TITLGA(IT),TITL2,GEN,NGEN,
<  3,1,NUCLID, DATE_AND_TIME)

45      CONTINUE
38      CONTINUE
      IF (INDPOP.NE.1) GO TO 50
40      CONTINUE
50      CONTINUE
C***      COMBINE HIGH AND LOW LET USING INPUT QUALITY FACTORS
          TITL2 = TLET(3)

          DO 80 I=1,4
          DO 80 J=1,NONCLD
          DO 80 K=1,NORGN
          DOSE(K,J,I,1)=(DOSE(K,J,I,1)*LLET(K)+ DOSE(K,J,I,2)*HLET(K))
          DOSE(NORGN+1,J,I,1)=DOSE(NORGN+1,J,I,1)+DOSE(K,J,I,1)*FACO(K,I)
          IF (K.GT.NGEN) GO TO 80
          GDOSE(K,J,I,1)=(GDOSE(K,J,I,1)*GLLET(K)+GDOSE(K,J,I,2)*GHLET(K))
80      CONTINUE
          DO 100 IT=1,3
          IDO=.FALSE.
          IF(SEP_DOSE_LET_TABLES) GO TO 95
          DO 90 K=1,6
          ITAB(K)=0
          IF (TABLE(K).NE.IT.AND.TABLE(K).NE.4) GO TO 90
          ITAB(K)=1
          IDO=.TRUE.
90      CONTINUE
95      CONTINUE
          ORGN(NORGN+1)=LAST
          ORC=ORG
          IF (IDO) CALL MULT(IT,FACD(1,IT),DOSE,NORGN+1,ORGN,TITLE,
<  TITLB(IT),ITAB,TITL2,NOTE,NUN(IT),OFAC,OREP,NOREP,20,NUCLID,
+  NONCLD )
          ORC=GON
          IF(IDO.AND.GENEFF) CALL MULT(IT,FACG(1,IT),GDOSE,NGEN,GEN,
<  TITLE,TITLGB(IT),ITAB,TITL2,NOT2,NOT2,0.,0.,0,3,NUCLID,NONCLD)
C      IF(IT.EQ.2) GO TO 138
          IF(NTLOC.EQ.0) GO TO 138
          DO 135 LL=1,NTLOC
          IF(FALOC(LL).NE.1) GO TO 135
          IF(HLLOC(LL).EQ.0) GO TO 135

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
12345678901234567890123456789012345678901234567890123456789012

```

      IF(LTABLE(LL).NE.IT .AND. LTABLE(LL).NE.4) GO TO 135
      CALL LOCTAB(IT,RNLOC(LL),PTLOC(LL),OGLOC(LL),FACD(1,IT),
> DOSE,TITLB(IT),TITL2,ORGN,NORGN+1,20,1,NUCLID,
+ DATE_AND_TIME)

```

```

      IF(GENEFF) CALL LOCTAB(IT,RNLOC(LL),PTLOC(LL),OGLOC(LL),
> FACG(1,IT),GDOSE,TITLGB(IT),TITL2,GEN,NGEN,3,1,NUCLID,
+ DATE_AND_TIME)

```

```

135     CONTINUE
138     CONTINUE
      IF (INDPOP.NE.1) RETURN
100    CONTINUE
      RETURN
      END

```

```

C-----
C
C           SUBROUTINE PREPHR
C
C-----

```

SUBROUTINE PREPHR(TABLE,TITLE,GSCFAC)

```

C     CORRECTED FOR LOCTAB ORGAN/CANCER "SUM" OPTION.  CBN 11/05/82
C***   PREPARE HEALTH RISKS TO BE OUTPUT
      CHARACTER*8 CANC,NUCLID,RNLOC,OGLOC,ORG,CAN,LAST,GEN,RREP,OREP,
+      CREP, GON, ORGN

```

```

      CHARACTER*80 NOTE,NOT2, TITLE
      CHARACTER*32 NUN
      CHARACTER*36 DATE_AND_TIME
      COMMON / HEADERINFO / DATE_AND_TIME

```

```

      CHARACTER*8 TLET, TITL2
      CHARACTER*40 TITLA, TITLG

```

```

      DIMENSION RNLOC(10), OGLOC(10), OREP(20), RREP(20), CREP(20),
+      NUCLID(40), CANC(20), ORGN(20), GEN(3)

```

```

      COMMON /LOCTBL_CHARS/ RNLOC, OGLOC
      COMMON /WORK_LEVEL_CHARS/ OREP, RREP, CREP
      COMMON /NAMES_CHARS/ NUCLID, CANC, ORGN, GEN

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

LOGICAL SEP_RISK_LET_TABLES, COMB_RISK_LET_TABLES,
+   ALL_RISK_LET_TABLES
COMMON/RISK_LET_TABLES/SEP_RISK_LET_TABLES, COMB_RISK_LET_TABLES,
+   ALL_RISK_LET_TABLES
COMMON/HEAD/ORG
COMMON/COMRN/WLRN(20,20),
A  RRISK,RREF(2),RYRLL,NOREP,NRREP,NCREP
INTEGER TABLE,FALOC,HLLOC,PTLOC
LOGICAL IDO,GENEFF
COMMON/COMLOC/PTLOC(10),FALOC(10),
> HLLOC(10),LTABLE(10),NTLOC
COMMON/COMGEN/NGEN,GDOSE(3,40,4,2),GRISK(3,40,4,2),
> GENEFF,GRFAC(2),REPPER,GLLET(3),GHLET(3),GREF(3,40,4)
DIMENSION TABLE(7),TITLA(3),TLET(3),
> FACD(4,3),ITAB(7),TITLG(3),FACG(4,3),NUN(3),DRISK(2)

```

```

COMMON/COMCA/NCANC,RELABS(20),RISK(20,40,4,2),RTABLE(7),
> AGEY,YRLL(20,40,4,2)
COMMON/COMNU/NONCLD,PSIZE(40),GIABS(4,40),
> INDPOP
DATA CAN/' CANCER '/,LAST/' TOTAL '/,GON/' GONAD '/
DATA NOTE/' RADON DAUGHTER EXPOSURE RISK:
+           '/
DATA NOT2/'
+           '/
DATA NUN/'
+           ',
+           ',
+           '(DEATH/YR)
DATA TITLA/' INDIVIDUAL LIFETIME RISK (DEATHS) ',
+           'MEAN INDIVIDUAL LIFETIME RISK (DEATHS) ',
+           'FATAL CANCER RATE (DEATH/YR) '/
DATA TITLG/' INDIVIDUAL GENETIC EFFECTS PER BIRTH ',
+           'MEAN INDIVIDUAL GENETIC EFFECTS /BIRTH ',
+           'COLLECTIVE GENETIC EFFECT(EFFECTIVE/YR) '/

```

```

DATA FACD/2*1.E-5,10.,1000.,2*1.E-5,10.,1000.,4*0./
DATA FACG/1.,1.,1.,100.,1.,1.,1.,100.,.001,.001,.001,.1/

```

C\*\*\*\*\*???????

```

DATA TLET/' LOW LET ', 'HIGH LET', 'COMB.LET'/
C*** MULTIPLY RISKS BY EXPOSURES
C*** PREPARE HIGH AND LOW LET SEPARATELY
DO 2 J=1,3
FACD(4,J)=GSCFAC*FACD(4,J)
FACG(4,J)=GSCFAC*FACG(4,J)
2 CONTINUE

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

DO 5 J=1,4
  FACG(J,3)=FACG(J,2)*REPPER
5  FACD(J,3)=FACD(J,2)/AGEX
  DO 50 L=1,2
    TITL2 = TLET(L)
    DO 25 I=1,4
      DO 25 J=1, NONCLD
        DO 25 K=1, NCANC
          RISK(NCANC+1, J, I, L)=RISK(K, J, I, L)+RISK(NCANC+1, J, I, L)
25  CONTINUE
    CANC(NCANC+1)=LAST
  DO 40 IT=1,3
    IDO=.FALSE.
    IF(COMB_RISK_LET_TABLES) GO TO 35
  DO 30 K=1,7
    ITAB(K)=0
    IF (TABLE(K).NE.IT.AND.TABLE(K).NE.4) GO TO 30
    IDO=.TRUE.
    ITAB(K)=1
30 CONTINUE
35  CONTINUE
    ORG=CAN
    DRISK(1)=RRISK
    IF(IT.EQ.3) DRISK(1)=RRISK/AGEX
    DRISK(2)=DRISK(1)
    IF (IDO) CALL MULT(IT, FACD(1, IT), RISK(1, 1, 1, L), NCANC+1, CANC, TITLE,
>  TITLA(IT), ITAB, TITL2, NOTE, NUN(IT), DRISK, CREP, NCREP, 20, NUCLID,
+  NONCLD )
    ORG=GON
    IF(IDO.AND. GENEFF) CALL MULT(IT, FACG(1, IT), GRISK(1, 1, 1, L),
>  1, GEN(3), TITLE, TITLG(IT), ITAB, TITL2, NOT2, NOT2, 0., 0., 0, 3, NUCLID,
+  NONCLD )
C  IF(IT.EQ.2) GO TO 38
    IF(NTLOC.EQ.0) GO TO 38
    DO 45 LL=1, NTLOC
      IF(FALOC(LL).NE.2) GO TO 45
      IF(HLLOC(LL).EQ.1) GO TO 45
      IF(LTABLE(LL).NE.IT .AND. LTABLE(LL).NE.4) GO TO 45
      CALL LOCTAB(IT, RNLOC(LL), PTLOC(LL), OGLOC(LL), FACD(1, IT),
>  RISK(1, 1, 1, L), TITLA(IT), TITL2, CANC, NCANC+1, 20, 3, NUCLID,
+  DATE_AND_TIME)

    IF(GENEFF) CALL LOCTAB(IT, RNLOC(LL), PTLOC(LL), OGLOC(LL),
<  FACG(1, IT), GRISK(1, 1, 1, L), TITLG(IT), TITL2,
<  GEN(3), 1, 3, 3, NUCLID, DATE_AND_TIME)

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012
45      CONTINUE
38      CONTINUE
      IF (INDPOP.NE.1) GO TO 50
40      CONTINUE
50      CONTINUE
C***    COMBINE HIGH AND LOW LET USING INPUT QUALITY FACTORS
      DO 70 I=1,4
      DO 70 J=1, NONCLD
          RISK(NCANC+1, J, I, 1) = 0.0
      DO 70 K=1, NCANC
          RISK(K, J, I, 1) = RISK(K, J, I, 1) + RISK(K, J, I, 2)
          RISK(NCANC+1, J, I, 1) = RISK(NCANC+1, J, I, 1) + RISK(K, J, I, 1)
          IF (K.GT.1) GO TO 70
          GRISK(K, J, I, 1) = GRISK(K, J, I, 1) + GRISK(K, J, I, 2)
70      CONTINUE
      TITL2 = TLET(3)
      DO 100 IT=1, 3
      IDO = .FALSE.
          IF (SEP_RISK_LET_TABLES) GO TO 95
      DO 90 K=1, 6
      ITAB(K) = 0
      IF (TABLE(K).NE.IT.AND.TABLE(K).NE.4) GO TO 90
      IDO = .TRUE.
      ITAB(K) = 1
90      CONTINUE
95      CONTINUE
      DRISK(1) = RRISK
      IF (IT.EQ.3) DRISK(1) = RRISK/AGEX
      DRISK(2) = DRISK(1)
      ORG = CAN
      IF (IDO) CALL MULT(IT, FACD(1, IT), RISK, NCANC+1, CANG, TITLE,
+          TITLA(IT), ITAB, TITL2, NOTE, NUN(IT), DRISK,
+          CREP, NCREP, 20, NUCLID, NONCLD )
      ORG = GON
      IF (IDO.AND.GENEFF) CALL MULT(IT, FAGG(1, IT), GRISK, 1, GEN(3),
>      TITLE, TITLG(IT), ITAB, TITL2, NOT2, NOT2, 0., 0., 0, 3, NUCLID, NONCLD)
C      IF (IT.EQ.2) GO TO 138
      IF (NTLOC.EQ.0) GO TO 138
      DO 135 LL=1, NTLOC
      IF (FALOC(LL).NE.2) GO TO 135
      IF (HLLOC(LL).EQ.0) GO TO 135
      IF (LTABLE(LL).NE.IT.AND.LTABLE(LL).NE.4) GO TO 135
      CALL LOCTAB(IT, RNLOC(LL), PTLOC(LL), OGLOC(LL), FACD(1, IT),
>      RISK, TITLA(IT), TITL2, CANG, NCANC+1, 20, 3, NUCLID,
+      DATE_AND_TIME)

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      IF(GENEFF) CALL LOCTAB(IT,RNLOC(LL),PTLOC(LL),OGLOC(LL),
>  FACG(1,IT),GRISK,TITLG(IT),TITL2,GEN(3),1,3,3,NUCLID,
+  DATE_AND_TIME)

```

```

135     CONTINUE
138     CONTINUE
      IF (INDPOP.NE.1) RETURN
100    CONTINUE
      RETURN
      END

```

```

C-----
C
C          SUBROUTINE PREPRF
C
C-----

```

SUBROUTINE PREPRF(TABLE,TITLE,GSCFAC)

```

C      CORRECTED FOR LOCTAB ORGAN/CANCER "SUM" OPTION.  CBN 11/05/82
C***    PREPARE HEALTH EQUIVALENT FACTORS TO BE OUTPUT.

```

```

      CHARACTER*8  CANC, NUCLID, LAST, RNLOC, OGLOC, ORC, CAN, GEN, OREP, RREP,
+      CREP, GON, ORGN, TITL2
      CHARACTER*80 TITLE, NOTE, NOTA
      CHARACTER*40 TITLA, TITLG
      CHARACTER*32 NUN
      CHARACTER*36 DATE_AND_TIME
      COMMON / HEADERINFO / DATE_AND_TIME

```

```

      DIMENSION RNLOC(10), OGLOC(10), OREP(20), RREP(20), CREP(20),
+      NUCLID(40), CANC(20), ORGN(20), GEN(3)

```

```

      COMMON /LOCTBL_CHARS/ RNLOC, OGLOC
      COMMON /WORK_LEVEL_CHARS/ OREP, RREP, CREP
      COMMON /NAMES_CHARS/ NUCLID, CANC, ORGN, GEN

```

```

      COMMON/COMRN/WLRN(20,20),RRISK,RREF(2),RYRLL,NOREP,NRREP,NCREP
      LOGICAL GENEFF
      COMMON/HEAD/ORC
      COMMON/COMCA/NCANC,RELABS(20),RISK(20,40,4,2)
      COMMON/COMRF/REF(20,40,4),FTABLE(7)
      COMMON/COMNU/NONCLD,PSIZE(40),GIABS(4,40),INDPOP
      COMMON/COMLOC/PTLOC(10),FALOC(10),

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```
> HLLOC(10),LTABLE(10),NTLOC
COMMON/COMGEN/NGEN,GDOSE(3,40,4,2),GRISK(3,40,4,2),
> GENEFF,GRFAC(2),REPPER,GLLET(3),GHLET(3),GREF(3,40,4)
INTEGER TABLE,FALOC,HLLOC,PTLOC
LOGICAL IDO
DIMENSION TABLE(7),TITLA(3),FACD(4,3),
> ITAB(7),TITLG(3),FACG(4,3),DREF(2),NUN(3)
```

```
DATA NUN/'(MREM/YR)
+ '(MREM/YR)
+ '(PERSON REM/YR)
DATA CAN/' CANCER '/,GON/' GONAD '/
DATA LAST/'W BODY '/
DATA NOTE/'RADON DAUGHTER EXPOSURE RISK EQUIVALENT:
```

```
+ '/
DATA NOTA/'
+ '/
```

```
DATA FACD/1.,1.,1.E6,1.E8,1.,1.,1.E6,1.E8,.001,
A .001,1.E3,1.E5/
DATA FACG/1.,1.,1.,100.,1.,1.,1.,100.,.001,.001,.001,.1/
```

```
DATA TITLA/'INDIVIDUAL RISK EQ. RATE (MREM/YEAR) ',
+ 'MEAN INDIVIDUAL RISK EQ. RATE(MREM/YEAR)',
+ 'COLLECTIVE RISK EQ RATE(PERSON REM/YEAR)'/
```

```
DATA TITLG/'INDIVIDUAL GENETIC RISK EQ. (MREM/YEAR) ',
+ 'MEAN INDIVIDUAL GENETIC RISK EQ.(MREM/Y)',
+ 'COLL. GENETIC RISK EQ. (PERSON REM/YEAR)'/
```

```
DATA TITL2/'
NCANR=NCANC+1
CANC(NCANR)=LAST
CREP(NCREP+1)=LAST
DO 5 J=1,3
FACD(4,J)=FACD(4,J)*GSCFAC
FACG(4,J)=FACG(4,J)*GSCFAC
```

```
5 CONTINUE
DO 20 IT=1,3
IDO=.FALSE.
DO 10 K=1,7
ITAB(K)=0
IF (TABLE(K).NE.IT.AND.TABLE(K).NE.4) GO TO 10
ITAB(K)=1
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      IDO=.TRUE.
10  CONTINUE
      NCRR=NCREP+1
      IF(NCREP.EQ.0) NCRR=0
      ORC=CAN
      FAC=1.
      IF(IT.EQ.3)FAC=.001
      DO 12 J=1,NCRR
12  DREF(J)=RREF(J)*FAC
      IF (IDO) CALL MULT(IT,FACD(1,IT),REF,NCANR,CANC,TITLE, TITLA(IT)
> ,ITAB,TITL2,NOTE,NUN(IT),DREF,CREP,NCRR,20,NUCLID, NONCLD)
      ORC=GON
      IF(IDO.AND.GENEFF) CALL MULT(IT,FACG(1,IT),GREF,1,GEN(3),
> TITLE,TITLG(IT),ITAB,TITL2,NOTA,NOTA,0.,0.,0,3,NUCLID,NONCLD)
C  IF(IT.EQ.2) GO TO 38
      IF(NTLOC.EQ.0) GO TO 38
      DO 35 LL=1,NTLOC
      IF(FALOC(LL).NE.3) GO TO 35
      IF(LTABLE(LL).NE.IT .AND. LTABLE(LL).NE.4) GO TO 35
      CALL LOCTAB(IT,RNLOC(LL),PTLOC(LL),OGLOC(LL),FACD(1,IT),
A  REF,TITLA(IT),TITL2,CANC,NCANR,20,2,NUCLID,
+  DATE_AND_TIME)

      IF(GENEFF) CALL LOCTAB(IT,RNLOC(LL),PTLOC(LL),OGLOC(LL),
A  FACG(1,IT),GREF,TITLG(IT),TITL2,GEN(3),1,3,2,NUCLID,
+  DATE_AND_TIME)

35  CONTINUE
38  CONTINUE
      IF (INDPOP.NE.1) RETURN
20  CONTINUE
      RETURN
      END

```

---

```

C
C
C          SUBROUTINE READ INFO FOR SYNOPSIS RPT
C
C
C

```

---

```

      SUBROUTINE READ_INFO_FOR_SYNOPSIS_RPT( FOOD_ARRAY_INFO,
+
C  _____
          NUMBER_FILES )

```

---

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

C    VARIABLES PASSED  
C

CHARACTER\*80 FOOD\_ARRAY\_INFO

INTEGER\*2    NUMBER\_FILES

C    GLOBAL VARIABLES  
C

CHARACTER\*80 FILES\_USED(10), NAME\_OF\_PERSON, PHONE\_NUMBER

COMMON / GENERIC\_INFO / FILES\_USED, NAME\_OF\_PERSON, PHONE\_NUMBER

CHARACTER\*80 COMMENTS(2)

CHARACTER\*72 FACILITY, ADDRESS

CHARACTER\*38 SOURCE\_CATEGORY

CHARACTER\*25 CITY

CHARACTER\*10 ZIPCODE

CHARACTER\*4    YEAR

CHARACTER\*2    STATE

COMMON / FACIL\_INFO / COMMENTS, FACILITY, ADDRESS,  
+                                SOURCE\_CATEGORY, CITY, ZIPCODE, YEAR, STATE

REAL            VEG\_LOCAL, VEG\_REGIONAL, VEG\_IMPORTED, MEAT\_LOCAL,  
+                MEAT\_REGIONAL, MEAT\_IMPORTED, MILK\_LOCAL,  
+                MILK\_REGIONAL, MILK\_IMPORTED

COMMON / FOOD\_INFO / VEG\_LOCAL, VEG\_REGIONAL, VEG\_IMPORTED,  
+                                MEAT\_LOCAL, MEAT\_REGIONAL, MEAT\_IMPORTED,  
+                                MILK\_LOCAL, MILK\_REGIONAL, MILK\_IMPORTED

INTEGER\*4    TEMPERATURE, RAINFALL\_RATE, LID\_HEIGHT  
REAL            LATITUDE, LONGITUDE

COMMON / SITE\_INFO / TEMPERATURE, RAINFALL\_RATE, LID\_HEIGHT,  
+                                LATITUDE, LONGITUDE

INTEGER\*2    NUMBER\_STACKS, SOURCE\_TYPE, PLUME\_RISE\_TYPE  
REAL            HEIGHT(6), AREA(6), AREA\_DIAMETER(6),  
+                STACK\_DIAMETER(6), BOUYANCY(6), MOMENTUM(6), ENTERED(7)

DARTAB (DARTAB2.FOR) Program File  
(continued)

0 1 2 3 4 5 6 7  
12345678901234567890123456789012345678901234567890123456789012

```
COMMON / EMMIS_INFO / HEIGHT, AREA, AREA_DIAMETER,  
+ STACK_DIAMETER, BOUYANCY, MOMENTUM,  
+ ENTERED,  
+ NUMBER_STACKS, SOURCE_TYPE, PLUME_RISE_TYPE
```

```
CHARACTER*8 NAME_NUC(36)  
CHARACTER*1 ISOL(36)  
COMMON / NUCCHARINFO / NAME_NUC, ISOL
```

```
INTEGER*2 NUMBER_NUCS, DECAY_CHAIN_FLAG (36), DAUGHTERS  
LOGICAL RN_RUN  
REAL AMAD(36), RELEASE_RATE(36,6), ALAMSUR(36)  
COMMON / NUC_INFO / AMAD, RELEASE_RATE, NUMBER_NUCS,  
+ DAUGHTERS, DECAY_CHAIN_FLAG, ALAMSUR, RN_RUN
```

```
CHARACTER*90 SAVED_TITLE_LINES (20)  
CHARACTER*26 ID_CODE  
COMMON / TITLES / SAVED_TITLE_LINES, ID_CODE
```

C LOCAL VARIABLES  
C LOCAL VARIABLES

```
INTEGER*2 I, J, RFLAG  
REAL TEMP, RAIN_RATE
```

C\*\*\*\*\* READ IN THE TITLE PAGE INFO \*\*\*\*\*

```
READ(11,1000) NAME_OF_PERSON  
READ(11,1000) PHONE_NUMBER
```

C\*\*\*\*\* READ IN THE NAMES OF THE INPUT FILES USED FOR THE RUN \*\*\*\*\*

```
I = 0  
10 CONTINUE  
I = I + 1  
READ(11,1000) FILES_USED(I)
```

C\*\*\*\* READ FILES UNTIL DELIMITER \* ENCOUNTERED

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```
IF ( FILES_USED(I)(1:1) .EQ. '*' ) THEN
  NUMBER_FILES = I - 1
  GO TO 20
ELSE
  GO TO 10
ENDIF
```

20 CONTINUE

C\*\*\*\*\* READ IN THE INFO ON THE FACILITY \*\*\*\*\*

```
C*** READ(11,1000) ID_CODE
READ(11,1000) FACILITY
READ(11,1000) ADDRESS
READ(11,1000) CITY
READ(11,1000) STATE
READ(11,1000) ZIPCODE
READ(11,1000) SOURCE_CATEGORY
READ(11,1000) YEAR
IF ( YEAR .EQ. 'YEAR' ) THEN
  YEAR = '1986'
ENDIF
```

```
READ(11,1000) COMMENTS(1)
READ(11,1000) COMMENTS(2)
```

C\*\*\*\*\* READ IN THE SITE SPECIFIC INFO \*\*\*\*\*

```
READ(11,*) TEMP
READ(11,*) RAIN_RATE
READ(11,*) LID_HEIGHT
READ(11,*) LATITUDE
READ(11,*) LONGITUDE
READ(11,1000) FOOD_ARRAY_INFO
```

```
READ(11,*) VEG_LOCAL, VEG_REGIONAL, VEG_IMPORTED,
+ MEAT_LOCAL, MEAT_REGIONAL, MEAT_IMPORTED,
+ MILK_LOCAL, MILK_REGIONAL, MILK_IMPORTED
```

C\*\*\*\*\* READ IN THE INFO ON THE SOURCE TERM \*\*\*\*\*

```
READ(11,*) RFLAG
IF ( RFLAG .EQ. 0 ) THEN
  RN_RUN = .FALSE.
ELSE
  RN_RUN = .TRUE.
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```
ENDIF
IF ( .NOT. RN_RUN ) THEN
  READ(11,*) DAUGHTERS
  READ(11,*) NUMBER_NUCS, NUMBER_STACKS
  DO 30 I = 1, NUMBER_NUCS
    READ(11,1000) NAME_NUC(I)
    READ(11,*) DECAY_CHAIN_FLAG (I)
    READ(11,1030) ISOL(I), AMAD(I), ALAMSUR(I)
    READ(11,*) ( RELEASE_RATE(I,J), J = 1,NUMBER_STACKS )
30  CONTINUE
  ELSE
    READ(11,*) NUMBER_NUCS, NUMBER_STACKS
    DO 35 I = 1, NUMBER_NUCS
      READ(11,1000) NAME_NUC(I)
      READ(11,1030) ISOL(I), AMAD(I), ALAMSUR(I)
      READ(11,*) ( RELEASE_RATE(I,J), J = 1,NUMBER_STACKS )
35  CONTINUE
  ENDIF
```

C\*\*\*\*\* READ IN THE EMISSION INFO \*\*\*\*\*

```
  READ(11,1040) SOURCE_TYPE, PLUME_RISE_TYPE
```

C\*\*\* SOURCE TYPE 1 = AREA, 0 = STACK \*\*\*\*\*  
C\*\*\* PLUME\_RISE 0 = BOUYANT, 1 = MOMENTUM, 2 = ENTERED \*\*\*\*\*

```
  READ(11,*) ( HEIGHT(J), J = 1,NUMBER_STACKS )
```

```
IF ( SOURCE_TYPE .EQ. 1 ) THEN
  READ(11,*) ( AREA (J), J = 1, NUMBER_STACKS )
  READ(11,*) ( AREA_DIAMETER (J), J = 1, NUMBER_STACKS )
ELSE
  READ(11,*) ( STACK_DIAMETER (J) , J = 1, NUMBER_STACKS )
ENDIF
```

```
IF ( PLUME_RISE_TYPE . EQ. 0 ) THEN
  READ(11,*) ( BOUYANCY(J) , J = 1, NUMBER_STACKS )
ELSE IF ( PLUME_RISE_TYPE . EQ. 1 ) THEN
  READ(11,*) ( MOMENTUM (J) , J = 1, NUMBER_STACKS )
ELSE
  READ(11,*) ( ENTERED(J) , J = 1, 7)
ENDIF
```

C\*\*\*\*\* CONVERT TEMPERATURE TO CELSIUS  
TEMP = TEMP - 273.0



DARTAB (DARTAB2.FOR) Program File  
(continued)

0           1           2           3           4           5           6           7  
123456789012345678901234567890123456789012345678901234567890123456789012

C\*\*\*\*\* CONVERT RAINFALL RATE AND TEMPERATURE TO NEAREST INTEGER  
  RAINFALL\_RATE = NINT(RAIN\_RATE)  
  TEMPERATURE   = NINT(TEMP        )

1000 FORMAT(A)  
1030 FORMAT( A, 1X, E10.3, 1X, E10.3)  
1040 FORMAT( I1, 1X, I1)

RETURN

END

```
C*****
C                                                                    *
C                                                                    *
C                                                                    *
C              OUTPUT ROUTINES AND REPORTS                            *
C                                                                    *
C                                                                    *
C                                                                    *
C*****
```

```
C=====
C                                                                    =
C              SUBROUTINE FACOUT                                       =
C                                                                    =
C=====
```

SUBROUTINE FACOUT( DATE\_AND\_TIME )

CHARACTER\*8 NUCLID, ORGN, CANC, GEN, RN222, WBODY  
CHARACTER\*36 DATE\_AND\_TIME

DIMENSION NUCLID(40), CANC(20), ORGN(20), GEN(3)

COMMON /NAMES\_CHARS/ NUCLID, CANC, ORGN, GEN

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
12345678901234567890123456789012345678901234567890123456789012

```

DIMENSION ARRAY(20,40,4),ORGN(1),PATH(7),DTABLE(7),WLSUM(4),
+            RFAC(20), NUCLID(40), CREP(20)

```

```

COMMON/COMNU/NONCLD,PSIZE(40),GIABS (4,40),INDPOP

```

```

COMMON/COMUS/SUMX(40,7),SUMY(40,3),PERX(40),PERY(40),TVAL(40),
> FAGO(20,4)

```

```

DATA RADON/'RN-222  '/
DATA PATH/'INGESTION            ','INHALATION            ','AIR IMMERSION    '
+            ','GROUND SURFACE    ','INTERNAL            ','EXTERNAL        '
+            ','                    ','                    '/'

```

```

DATA SUM/'TOTAL  '/

```

```

C***        SUM OVER ALL NUCLIDES FOR EACH ORGAN AND PATHWAY

```

```

C***        PATHWAY 5 IS INTERNAL, 6 IS EXTERNAL, AND 7 IS ALL PATHWAYS

```

```

DO 10 I=1,7
DO 10 K=1,NORGN
10 SUMX(K,I)=0.0
DO 30 I=1,4
DO 20 K=1,NORGN
DO 20 J=1,NONCLD
SUMX(K,I)=SUMX(K,I)+ARRAY(K,J,I)
IF (I.LE.2) SUMX(K,5)=SUMX(K,5)+ARRAY(K,J,I)
IF (I.GE.3) SUMX(K,6)=SUMX(K,6)+ARRAY(K,J,I)
SUMX(K,7)=SUMX(K,7)+ARRAY(K,J,I)

```

```

20 CONTINUE

```

```

30 CONTINUE

```

```

C***        TABLE 1

```

```

IF (DTABLE(1).EQ.0) GO TO 80

```

```

DO 70 I=1,4

```

```

WRITE(27,10000) TITLE,TITL1,TITLA

```

```

WRITE(27,10200) PATH(I)

```

```

WRITE(27,10300) ORC,(ORGN(K),K=1,NORGN)

```

```

WRITE(27,10400)

```

```

DO 60 J=1,NONCLD

```

```

DO 40 K=1,NORGN

```

```

PERX(K)=0.0

```

```

40 IF (SUMX(K,I).NE.0.0) PERX(K)=ARRAY(K,J,I)/SUMX(K,I)*100.

```

```

WRITE(27,10500) NUCLID(J),(ARRAY(K,J,I),K=1,NORGN)

```

```

WRITE(27,10600) PATH(I),(PERX(K),K=1,NORGN)

```

```

DO 50 K=1,NORGN

```

```

II=5

```

```

IF (I.GT.2) II=6

```

```

PERX(K)=0.0

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012
WRITE(27,10200) GEN(3),((GRISK(1,K,J,N),N=1,2),J=1,2),
A (GRISK(1,K,J,1),J=3,4)
200 CONTINUE
WRITE(27,10176)
10176 FORMAT('0',28X,'YEARS OF LIFE LOST FACTORS',35X,/
A 1X,'CANCER',10X,'INGESTION',10X,'INHALATION',
A 9X,'AIR',5X,'GROUND'/
B 12X,'LOW LET',3X,'HIGH LET',2X,'LOW LET',
B 3X,'HIGH LET',2X,'IMMERSION',1X,'SURFACE')
WRITE(27,10250) (CANC(I),((YRLL(I,K,J,N),N=1,2),J=1,2),
A (YRLL(I,K,J,1),J=3,4),I = 1, NCANC)

10250 FORMAT((1X,A8,1X,6(1PE10.1)))
WRITE(27,10190)
10190 FORMAT('1 RISK EQUIVALENT CONVERSION FACTORS'/
A 1X,'CANCER',3X,'INGESTION',1X,'INHALATION',
B 4X,'AIR',4X,'GROUND'/31X,'IMMERSION',2X,'SURFACE')
NALL=NCANC+1
CANC(NALL)=WBODY
WRITE(27,10275) (CANC(I),(REF(I,K,J),J=1,4),I=1,NALL)
10275 FORMAT((1X,A8,1X,4(1PE10.2)))
IF(.NOT.GENEFF) GO TO 300
WRITE(27,10600)
10600 FORMAT(' GENETIC EFFECT RISK EQ. CONVERSION FACTOR')
WRITE(27,10275) GEN(3),(GREF(1,K,I),I=1,4)
300 CONTINUE
1000 CONTINUE
DO 400 I=1,NONCLD
IF(NUCLID(I).EQ.RN222) GO TO 450
400 CONTINUE
RETURN
450 WRITE(27,10700) RRISK,RYRLL,RREF
10700 FORMAT('1FOR RN-222 WORKING LEVEL CALCULATIONS: '/
A ' RISK CONVERSION FACTOR = ',1PE10.2/
B ' YEARS OF LIFE LOST FACTOR = ',1PE10.2/
C ' RISK EQ. CONVERSION FACTOR (PULMINARY) = ',1PE10.2/
D ' RISK EQ. CONVERSION FACTOR (W BODY ) = ',1PE10.2)
RETURN
END

```

---

```

C
C
C          SUBROUTINE RDORGF
C
C

```

---

DARTAB (DARTAB2.FOR) Program File  
(continued)

0 1 2 3 4 5 6 7  
12345678901234567890123456789012345678901234567890123456789012

```
SUBROUTINE RDORGF(ORGN)
CHARACTER*8 ORGN,ORGB
COMMON/COMWOR/FACO(20,4)
DIMENSION ORGN(20),ORGB(20),ORGDAT(20),IPATH(20)
COMMON/COMOR/NORGN,TIME(20),DOSE(20,40,4,2)
NAMELIST/ ORGANF/NORGB,ORGB,ORGDAT,IPATH
READ(5,ORGANF)
WRITE(27,10000)

DO 10 J=1,NORGB
IF (IPATH(J).NE.5) THEN
WRITE(27,10200) ORGB(J),ORGDAT(J),IPATH(J)
ELSE IF (IPATH(J).EQ.5) THEN
WRITE(27,10300) ORGB(J),ORGDAT(J)
ENDIF
10 CONTINUE

DO 80 K=1,NORGN
DO 30 I=1,4
30 FACO(K,I)=0.0
DO 40 J=1,NORGB
IF (ORGN(K).EQ.ORGB(J)) GO TO 50
40 CONTINUE
GO TO 80
50 IF (IPATH(J).EQ.5) GO TO 60
FACO(K,IPATH(J))=ORGDAT(J)
GO TO 80
60 DO 70 I=1,4
FACO(K,I)=ORGDAT(J)
70 CONTINUE
80 CONTINUE
RETURN
10000 FORMAT('OORGAN DOSE WEIGHTING FACTORS'//
A ' ORGAN FACTORS PATHWAYS'//)
10200 FORMAT(1X,A8,1X,F8.5,2X,I2)
10300 FORMAT(1X,A8,1X,F8.5,2X,' 1 2 3 4')

END
```

---

C  
C  
C  
C

SUBROUTINE SUMMARY

---

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
12345678901234567890123456789012345678901234567890123456789012

C

SUBROUTINE SUMMRY(GSCFAC)

CC\*\* THE OUTPUT OF A TITLE HAS BEEN REMOVED 8/88

CHARACTER\*8 NUCLID, ORGN, GEN, CANC, TOTAL, WTSUM, RN222

CHARACTER\*36 DATE\_AND\_TIME

COMMON / HEADERINFO / DATE\_AND\_TIME

DIMENSION NUCLID(40), CANC(20), ORGN(20), GEN(3)

COMMON /NAMES\_CHARS/ NUCLID, CANC, ORGN, GEN

LOGICAL GENEFF, RNWR

REAL LLET

LOGICAL SEP\_DOSE\_LET\_TABLES, COMB\_DOSE\_LET\_TABLES,

+ ALL\_DOSE\_LET\_TABLES

COMMON/DOSE\_LET\_TABLES/SEP\_DOSE\_LET\_TABLES, COMB\_DOSE\_LET\_TABLES,

+ ALL\_DOSE\_LET\_TABLES

C

C\*\*\* FOLLOWING VARS AND COMMON ADDED FOR NEW SYNOPSIS REPORT 9/1988

REAL WLI, MAX\_PCI\_LITERS, FATAL\_CANCER\_RISK, ORGAN\_DOSES(20),

+ PATHWAY\_DOSES(4), NUC\_DOSES(36)

INTEGER LOC\_DIST

COMMON / IND\_RESULTS / WLI, LOC\_DIST, MAX\_PCI\_LITERS,

+ FATAL\_CANCER\_RISK, ORGAN\_DOSES,

+ PATHWAY\_DOSES, NUC\_DOSES

REAL SRISK(400), RSKLIN(400), EFFECT\_PERSON\_REM,

+ PERSON\_WORKING\_LEVEL, POP\_ORGAN\_DOSES(20)

INTEGER POPLIN(400)

INTEGER IO(400), NLOC, NOP

COMMON / POPU\_RESULTS / POPLIN, RSKLIN, SRISK, IO, NLOC, NOP,

+ EFFECT\_PERSON\_REM, PERSON\_WORKING\_LEVEL,

+ POP\_ORGAN\_DOSES

C\*\*\*

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
12345678901234567890123456789012345678901234567890123456789012
COMMON/COMEX/EXPP(20,20,40,4),POP(20,20),POPFAC,TOTFAC,NOL,NOU,
> NRL,NRU,IDIST(20),ILOC,JLOC
COMMON/COMOR/NORGN,TIME(20),DOSE(20,40,4,2),DTABLE(7)
COMMON/COMRN/WLRN(20,20),
A  RRISK,RREF(2),RYRLL,NOREP,NRREP,NCREP
COMMON/LETFAC/HLET(20),LLET(20)
COMMON/COMNU/NONCLD,PSIZE(40),GIABS(4,40),
>INDPOP
COMMON/COMGEN/NGEN,GDOSE(3,40,4,2),GRISK(3,40,4,2),
> GENEFF,GRFAC(2),REPPER,GLLET(3),GHLET(3),GREF(3,40,4)
COMMON/COMUS/STLOW(40),STHIG(40),STCOM(40),STGLO(40),
A  STGHI(40),STGCO(40)
DIMENSION DOSFAC(4),GENFAC(4)
DATA RN222/'RN-222' /
DATA DOSFAC/1.,1.,1.,100./,
+ GENFAC/1.,1.,1.,100./,TOTAL/'TOTAL' /,WTSUM/'WT.SUM' /
COMMON/COMWOR/DFAC(20,4)
DOSFAC(4)=GSCFAC*DOSFAC(4)
GENFAC(4)=GSCFAC*GENFAC(4)
RNWR=.FALSE.
DO 5 J=1,NONCLD
IF(NUCLID(J).EQ.RN222) RNWR=.TRUE.
5  CONTINUE
   WLC=0.0
   DO 10 II=NOL,NOU
   DO 10 JJ=NRL,NRU
   WLC=WLC+WLRN(JJ,II)
10 CONTINUE
   WLE=WLC*TOTFAC
C***** DOSE RATES BY ORGAN
   DO 50 J=1,40
   STHIG(J)=0.0
   STLOW(J)=0.0
   STCOM(J)=0.0
   STGLO(J)=0.0
   STGHI(J)=0.0
   STGCO(J)=0.0
50 CONTINUE
   SUML=0.0
   SUMH=0.0
   SUMC=0.0
   DO 100 L=1,NORGN
   DO 100 K=1,4
   FACNEW=POPFAC*DOSFAC(K)
   FACNE2=POPFAC*GENFAC(K)
   DO 100 J=1,NONCLD

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012

```

STHIG(L)=STHIG(L)+DOSE(L,J,K,2)\*FACNEW\*EXPP(JLOC,ILOC,J,K)  
 SUMH=SUMH+DOSE(L,J,K,2)\*FACNEW\*EXPP(JLOC,ILOC,J,K)\*DFAC(L,K)  
 STLOW(L)=STLOW(L)+DOSE(L,J,K,1)\*FACNEW\*EXPP(JLOC,ILOC,J,K)  
 SUML=SUML+DOSE(L,J,K,1)\*FACNEW\*EXPP(JLOC,ILOC,J,K)\*DFAC(L,K)  
 STCOM(L)=STCOM(L)+DOSE(L,J,K,1)\*FACNEW\*EXPP(JLOC,ILOC,J,K)  
 < \*LLET(L)+DOSE(L,J,K,2)\*FACNEW\*EXPP(JLOC,ILOC,J,K)\*HLET(L)  
 SUMC=SUMC+DOSE(L,J,K,1)\*FACNEW\*EXPP(JLOC,ILOC,J,K)\*  
 < LLET(L)\*DFAC(L,K)+DOSE(L,J,K,2)\*FACNEW\*EXPP(JLOC,ILOC,J,K)\*  
 < HLET(L)\*DFAC(L,K)  
 IF(L.GT.NGEN) GO TO 100  
 STGHI(L)=STGHI(L)+GDOSE(L,J,K,2)\*FACNE2\*EXPP(JLOC,ILOC,J,K)  
 STGLO(L)=STGLO(L)+GDOSE(L,J,K,1)\*FACNE2\*EXPP(JLOC,ILOC,J,K)  
 STGCO(L)=STGCO(L)+GDOSE(L,J,K,1)\*FACNE2\*EXPP(JLOC,ILOC,J,K)  
 < \*GLLET(L)+GDOSE(L,J,K,2)\*FACNE2\*EXPP(JLOC,ILOC,J,K)\*GHLET(L)  
100 CONTINUE

C\*\*\*\*\* SAVE COMBINED DOSES FOR SYNOPSIS REPORT  
 DO 15 L = 1, NORGN  
     ORGAN\_DOSES(L) = STCOM(L)  
15 CONTINUE

WRITE(27,101)DATE\_AND\_TIME  
101 FORMAT('1',T10,'DATE',2X,A)  
 WRITE(27,10100) (ORGN(L),L=1,NORGN),WTSUM  
 IF ( (SEP\_DOSE\_LET\_TABLES) .OR. (ALL\_DOSE\_LET\_TABLES) ) THEN  
     WRITE(27,10205) (STLOW(L),L=1,NORGN),SUML  
     WRITE(27,10210) (STHIG(L),L=1,NORGN),SUMH  
 ENDIF  
 IF ( ( COMB\_DOSE\_LET\_TABLES ) .OR. ( ALL\_DOSE\_LET\_TABLES ))  
 + WRITE(27,10215) (STCOM(L),L=1,NORGN),SUMC  
 WLI=WLRN(JLOC,ILOC)\*POPFAC  
 IF(RNWR)WRITE(27,20216) WLI  
20216 FORMAT('ORADON DAUGHTER EXPOSURE: '/  
 A ' (WORKING LEVEL)',9X,1PE10.2)

10100 FORMAT('0',/30X,'ORGAN DOSE/EXPOSURE SUMMARY'//  
 A '0\*\*\* SELECTED INDIVIDUAL \*\*\*'/  
 B 'ODOSE RATES: '/  
 > 17X,' ORGANS:',10(2X,A8)/(26X,10(2X,A8))  
10205 FORMAT(' LOW LET (MRAD/Y)',8X,10(1PE10.2)/  
 A (26X,10(1PE10.2)))  
10206 FORMAT(' LOW LET (PERSON RAD/Y)',1X,10(1PE10.2)/  
 A (26X,10(1PE10.2)))  
10210 FORMAT(' HIGH LET (MRAD/Y)',7X,10(1PE10.2)/  
 A (26X,10(1PE10.2)))  
10211 FORMAT(' HIGH LET (PERSON RAD/Y)',10(1PE10.2)/

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
12345678901234567890123456789012345678901234567890123456789012
A (26X,10(1PE10.2)))
10215  FORMAT(' DOSE EQUIVALENT (MREM/Y)',10(1PE10.2)/
A (26X,10(1PE10.2)))
10216  FORMAT(' DOSE EQ. (PERSON REM/Y)',10(1PE10.2)/
A (26X,10(1PE10.2)))
      IF(INDPOP.NE.1) GO TO 1000
      DO 150 J=1,40
      STHIG(J)=0.0
      STLOW(J)=0.0
      STCOM(J)=0.0
      STGHI(J)=0.0
      STGLO(J)=0.0
      STGCO(J)=0.0
150    CONTINUE
      SUML=0.0
      SUMH=0.0
      SUMC=0.0
      DO 200 L=1,NORGN
      DO 200 K=1,4
      FACNEW=TOTFAC*DOSFAC(K)
      FACNE2=TOTFAC*GENFAC(K)
      DO 200 J=1,NONCLD
      DO 200 II=NOL,NOU
      DO 200 JJ=NRL,NRU
      STHIG(L)=STHIG(L)+DOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW
      SUMH=SUMH+DOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW*DFAC(L,K)
      STLOW(L)=STLOW(L)+DOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
      SUML=SUML+DOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW*DFAC(L,K)
      STCOM(L)=STCOM(L)+DOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
      > *LLET(L)+DOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW*HLET(L)
      SUMC=SUMC+DOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW*
      > LLET(L)*DFAC(L,K)+DOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW*
      > HLET(L)*DFAC(L,K)
      IF(L.GT.NGEN) GO TO 200
      STGHI(L)=STGHI(L)+GDOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2
      STGLO(L)=STGLO(L)+GDOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2
      STGCO(L)=STGCO(L)+GDOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2
      > *GLLET(L)+GDOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2*GHLET(L)
200    CONTINUE
      WRITE(27,10400) (ORGN(L),L=1,NORGN),WTSUM
10400  FORMAT(/'0*** MEAN INDIVIDUAL ***'/
A 'ODOSE RATE: '/
+ 17X,' ORGANS:',10(2X,A8)/(26X,10(2X,A8)))
      IF ( (SEP_DOSE_LET_TABLES) .OR. (ALL_DOSE_LET_TABLES) ) THEN
      WRITE(27,10205) (STLOW(L),L=1,NORGN),SUML
      WRITE(27,10210) (STHIG(L),L=1,NORGN),SUMH

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      ENDIF
      IF ( ( COMB_DOSE_LET_TABLES ) .OR. ( ALL_DOSE_LET_TABLES ) )
+     WRITE(27,10215) (STCOM(L),L=1,NORGN),SUMC
      IF(RNWR)WRITE(27,20216) WLE
      DO 350 K=1,40
      STLOW(K)=0.0
      STHIG(K)=0.0
      STCOM(K)=0.0
      IF(K.GT.NGEN) GO TO 350
      STGLO(K)=0.0
      STGHI(K)=0.0
      STGCO(K)=0.0
350     CONTINUE
      SUML=0.0
      SUMH=0.0
      SUMC=0.0
      DO 400 K=1,4
      DO 400 J=1,NONCLD
      TEXPP=0.0
      DO 403 II=NOL,NOU
      DO 403 JJ=NRL,NRU
      TEXPP=TEXPP+EXPP(JJ,II,J,K)
403     CONTINUE
      TEXPP=TEXPP*.001
      DO 400 L=1,NORGN
      STLOW(L)=STLOW(L)+DOSE(L,J,K,1)*TEXPP*DOSFAC(K)
      SUML=SUML+DOSE(L,J,K,1)*TEXPP*DOSFAC(K)*DFAC(L,K)
      STHIG(L)=STHIG(L)+DOSE(L,J,K,2)*TEXPP*DOSFAC(K)
      SUMH=SUMH+DOSE(L,J,K,2)*TEXPP*DOSFAC(K)*DFAC(L,K)
      STCOM(L)=STCOM(L)+DOSE(L,J,K,1)*TEXPP*DOSFAC(K)*LLET(L)+
      > DOSE(L,J,K,2)*TEXPP*DOSFAC(K)*HLET(L)
      SUMC=SUMC+DOSE(L,J,K,1)*TEXPP*DOSFAC(K)*LLET(L)*DFAC(L,K)+
      > DOSE(L,J,K,2)*TEXPP*DOSFAC(K)*HLET(L)*DFAC(L,K)
      IF(L.GT.NGEN) GO TO 400
      STGLO(L)=STGLO(L)+GDOSE(L,J,K,1)*TEXPP*GENFAC(K)
      STGHI(L)=STGHI(L)+GDOSE(L,J,K,2)*TEXPP*GENFAC(K)
      STGCO(L)=STGCO(L)+GDOSE(L,J,K,1)*TEXPP*GENFAC(K)*GLLET(L)+
      > GDOSE(L,J,K,2)*TEXPP*GENFAC(K)*GHLET(L)
400     CONTINUE

C***   SAVE EFFECTIVE PERSON-REM/ YEAR FOR SYNOPSIS REPORT

      EFFECT_PERSON_REM = STCOM (NORGN)

C****  SAVE COMBINED DOSES FOR SYNOPSIS REPORT
      DO 16 L = 1, NORGN - 1

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      POP_ORGAN_DOSES(L) = STCOM(L)
16      CONTINUE

      WRITE(27,10700) (ORGN(L),L-1,NORGN),WTSUM
10700  FORMAT(/'0*** COLLECTIVE POPULATION ***'/
      A  'ODOSE RATE: '/
      > 17X,' ORGANS:',10(2X,A8)/(26X,10(2X,A8)))
      IF ( (SEP_DOSE_LET_TABLES ) .OR. (ALL_DOSE_LET_TABLES) ) THEN
          WRITE(27,10206) (STLOW(L),L-1,NORGN),SUML
          WRITE(27,10211) (STHIG(L),L-1,NORGN),SUMH
      ENDIF
      IF ( (COMB_DOSE_LET_TABLES ) .OR. (ALL_DOSE_LET_TABLES) )
      +   WRITE(27,10216) (STCOM(L),L-1,NORGN),SUMC

      IF (RNWR) THEN

C***      SAVE PERSON WORKING LEVELS FOR SYNOPSIS REPORT

          PERSON_WORKING_LEVEL = WLC

          WRITE(27,20416) WLC
20416  FORMAT('ORADON DAUGHTER EXPOSURE: '/
      A      ' (PERSON WORKING LEVEL)',2X,1PE10.2)

      ENDIF

C****  DOSE RATES BY PATHWAY
1000  DO 1050 J=1,40
      STHIG(J)=0.0
      STLOW(J)=0.0
      STCOM(J)=0.0
      STGHI(J)=0.0
      STGLO(J)=0.0
      STGCO(J)=0.0
1050  CONTINUE
      DO 1199 K=1,4
      FACNE2=POPFAC*GENFAC(K)
      DO 1100 L=1,NORGN
      FACNEW=POPFAC*DOSFAC(K)*DFAC(L,K)
      DO 1100 J=1,NONCLD
      STHIG(K)=STHIG(K)+DOSE(L,J,K,2)*FACNEW*EXPP(JLOC,ILOC,J,K)
      STLOW(K)=STLOW(K)+DOSE(L,J,K,1)*FACNEW*EXPP(JLOC,ILOC,J,K)
      STCOM(K)=STCOM(K)+DOSE(L,J,K,1)*FACNEW*EXPP(JLOC,ILOC,J,K)
      < *LLET(L)+DOSE(L,J,K,2)*FACNEW*EXPP(JLOC,ILOC,J,K)*HLET(L)
      IF(L.NE.NGEN) GO TO 1100
      STGHI(K)=STGHI(K)+GDOSE(L,J,K,2)*FACNE2*EXPP(JLOC,ILOC,J,K)

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012
STGLO(K)=STGLO(K)+GDOSE(L,J,K,1)*FACNE2*EXPP(JLOC,ILOC,J,K)
STGCO(K)=STGCO(K)+GDOSE(L,J,K,1)*FACNE2*EXPP(JLOC,ILOC,J,K)
< *GLLET(L)+GDOSE(L,J,K,2)*FACNE2*EXPP(JLOC,ILOC,J,K)*GHLET(L)
1100 CONTINUE
    IF(K.GT.2) GO TO 1101
    STHIG(5)=STHIG(5)+STHIG(K)
    STLOW(5)=STLOW(5)+STLOW(K)
    STCOM(5)=STCOM(5)+STCOM(K)
    STGHI(5)=STGHI(5)+STGHI(K)
    STGLO(5)=STGLO(5)+STGLO(K)
    STGCO(5)=STGCO(5)+STGCO(K)
1101 IF(K.LT.3) GO TO 1102
    STHIG(6)=STHIG(6)+STHIG(K)
    STLOW(6)=STLOW(6)+STLOW(K)
    STCOM(6)=STCOM(6)+STCOM(K)
    STGHI(6)=STGHI(6)+STGHI(K)
    STGLO(6)=STGLO(6)+STGLO(K)
    STGCO(6)=STGCO(6)+STGCO(K)
1102 STHIG(7)=STHIG(7)+STHIG(K)
    STLOW(7)=STLOW(7)+STLOW(K)
    STCOM(7)=STCOM(7)+STCOM(K)
    STGHI(7)=STGHI(7)+STGHI(K)
    STGLO(7)=STGLO(7)+STGLO(K)
    STGCO(7)=STGCO(7)+STGCO(K)
1199 CONTINUE
    WRITE(27,101)DATE_AND_TIME
    WRITE(27,11100)
11100 FORMAT('0'/30X,' PATHWAY DOSE/EXPOSURE SUMMARY'/
A '0*** SELECTED INDIVIDUAL ***'/
B 'ODOSE RATES: '/
< ' WEIGHTED SUMS OF ORGAN DOSE RATES'/
< '           PATHWAYS:  INGESTION ',
< ' INHALATION AIR      GROUND',
< '   INTERNAL EXTERNAL TOTAL'/
< 45X,' IMMERSION SURFACE')
    IF ( ( SEP_DOSE_LET_TABLES ) .OR. ( ALL_DOSE_LET_TABLES ) ) THEN
        WRITE(27,10205) (STLOW(L),L=1,7)
        WRITE(27,10210) (STHIG(L),L=1,7)
    ENDIF
    IF ( ( COMB_DOSE_LET_TABLES ) .OR. ( ALL_DOSE_LET_TABLES ) )
+   WRITE(27,10215) (STCOM(L),L=1,7)

    IF(RNWR) THEN
        WRITE(27,20216) WLI
    ELSE
C**** SAVE DOSES BY PATHWAY FOR DATABASE

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
12345678901234567890123456789012345678901234567890123456789012

```

      DO 20 L = 1, 4
      PATHWAY_DOSES(L) = STCOM(L)
20    CONTINUE
      ENDIF

      IF(INDPOP.NE.1) GO TO 11000
      DO 1150 J=1,40
      STHIG(J)=0.0
      STLOW(J)=0.0
      STCOM(J)=0.0
      STGHI(J)=0.0
      STGLO(J)=0.0
      STGCO(J)=0.0
1150  CONTINUE
      DO 1299 K=1,4
      FACNE2=TOTFAC*GENFAC(K)
      DO 1200 L=1,NORGN
      FACNEW=TOTFAC*DOSFAC(K)*DFAC(L,K)
      DO 1200 J=1,NONCLD
      DO 1200 II=NOL,NOU
      DO 1200 JJ=NRL,NRU
      STHIG(K)=STHIG(K)+DOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW
      STLOW(K)=STLOW(K)+DOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
      STCOM(K)=STCOM(K)+DOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
      > *LLET(L)+DOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW*HLET(L)
      IF(L.NE.NGEN) GO TO 1200
      STGHI(K)=STGHI(K)+GDOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2
      STGLO(K)=STGLO(K)+GDOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2
      STGCO(K)=STGCO(K)+GDOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2*
      A GLLET(L)+GDOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2*GHLET(L)
1200  CONTINUE
1203  IF(K.GT.2) GO TO 1201
      STHIG(5)=STHIG(5)+STHIG(K)
      STLOW(5)=STLOW(5)+STLOW(K)
      STCOM(5)=STCOM(5)+STCOM(K)
      STGHI(5)=STGHI(5)+STGHI(K)
      STGLO(5)=STGLO(5)+STGLO(K)
      STGCO(5)=STGCO(5)+STGCO(K)
1201  IF(K.LT.3) GO TO 1202
      STHIG(6)=STHIG(6)+STHIG(K)
      STLOW(6)=STLOW(6)+STLOW(K)
      STCOM(6)=STCOM(6)+STCOM(K)
      STGHI(6)=STGHI(6)+STGHI(K)
      STGLO(6)=STGLO(6)+STGLO(K)
      STGCO(6)=STGCO(6)+STGCO(K)
1202  STHIG(7)=STHIG(7)+STHIG(K)

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      STLOW(7)=STLOW(7)+STLOW(K)
      STCOM(7)=STCOM(7)+STCOM(K)
      STGHI(7)=STGHI(7)+STGHI(K)
      STGLO(7)=STGLO(7)+STGLO(K)
      STGCO(7)=STGCO(7)+STGCO(K)
1299  CONTINUE
      WRITE(27,11400)
11400  FORMAT(/'0*** MEAN INDIVIDUAL ***'/
      A  'ODOSE RATES: '/
      <  ' WEIGHTED SUMS OF ORGAN DOSE RATES' /
      <  '           PATHWAYS:  INGESTION ',
      <  ' INHALATION AIR   GROUND',
      <  '   INTERNAL EXTERNAL TOTAL' /
      <  45X, ' IMMERSION SURFACE' )
      IF ( (SEP_DOSE_LET_TABLES ) .OR. (ALL_DOSE_LET_TABLES) ) THEN
          WRITE(27,10205) (STLOW(L),L=1,7)
          WRITE(27,10210) (STHIG(L),L=1,7)
      ENDIF
      IF ( ( COMB_DOSE_LET_TABLES ) .OR. ( ALL_DOSE_LET_TABLES ) )
+      WRITE(27,10215) (STCOM(L),L=1,7)
      IF(RNWR)WRITE(27,20216) WLE
      DO 1350 K=1,40
          STLOW(K)=0.0
          STHIG(K)=0.0
          STCOM(K)=0.0
          STGLO(K)=0.0
          STGHI(K)=0.0
          STGCO(K)=0.0
1350  CONTINUE
      DO 1499 K=1,4
      DO 1400 J=1, NONCLD
          TEXPP=0.0
      DO 1404 II=NOL, NOU
      DO 1404 JJ=NRL, NRU
          TEXPP=TEXPP+EXPP(JJ, II, J, K)
1404  CONTINUE
          TEXPP=TEXPP*.001
      DO 1400 L=1, NORG
          STLOW(K)=STLOW(K)+DOSE(L, J, K, 1)*TEXPP*DOSFAC(K)*DFAC(L, K)
          STHIG(K)=STHIG(K)+DOSE(L, J, K, 2)*TEXPP*DOSFAC(K)*DFAC(L, K)
          STCOM(K)=STCOM(K)+DOSE(L, J, K, 1)*TEXPP*DOSFAC(K)*LLET(L)*
      >  DFAC(L, K)+DOSE(L, J, K, 2)*TEXPP*DOSFAC(K)*HLET(L)*DFAC(L, K)
          IF(L.NE.NGEN) GO TO 1400
          STGLO(K)=STGLO(K)+GDOSE(L, J, K, 1)*TEXPP*GENFAC(K)
          STGHI(K)=STGHI(K)+GDOSE(L, J, K, 2)*TEXPP*GENFAC(K)
          STGCO(K)=STGCO(K)+GDOSE(L, J, K, 2)*TEXPP*GENFAC(K)*GHLET(L)+

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
12345678901234567890123456789012345678901234567890123456789012

```

      A  GDOSE(L,J,K,1)*TEXPP*GENFAC(K)*GLLET(L)
1400  CONTINUE
1403  IF(K.GT.2) GO TO 1401
      STLOW(5)=STLOW(5)+STLOW(K)
      STHIG(5)=STHIG(5)+STHIG(K)
      STCOM(5)=STCOM(5)+STCOM(K)
      STGLO(5)=STGLO(5)+STGLO(K)
      STGHI(5)=STGHI(5)+STGHI(K)
      STGCO(5)=STGCO(5)+STGCO(K)
      GO TO 1402
1401  STLOW(6)=STLOW(6)+STLOW(K)
      STHIG(6)=STHIG(6)+STHIG(K)
      STCOM(6)=STCOM(6)+STCOM(K)
      STGLO(6)=STGLO(6)+STGLO(K)
      STGHI(6)=STGHI(6)+STGHI(K)
      STGCO(6)=STGCO(6)+STGCO(K)
1402  STLOW(7)=STLOW(7)+STLOW(K)
      STHIG(7)=STHIG(7)+STHIG(K)
      STCOM(7)=STCOM(7)+STCOM(K)
      STGLO(7)=STGLO(7)+STGLO(K)
      STGHI(7)=STGHI(7)+STGHI(K)
      STGCO(7)=STGCO(7)+STGCO(K)
1499  CONTINUE
11700 FORMAT(/'0*** COLLECTIVE POPULATION ***'/
      A  'ODOSE RATES: '/
      <  ' WEIGHTED SUMS OF ORGAN DOSE RATES' /
      <  ' PATHWAYS: INGESTION ',
      <  ' INHALATION AIR GROUND',
      <  ' INTERNAL EXTERNAL TOTAL' /
      <  45X,' IMMERSION SURFACE')
      WRITE(27,11700)
      IF ( (SEP_DOSE_LET_TABLES) .OR. (ALL_DOSE_LET_TABLES) ) THEN
          WRITE(27,10206) (STLOW(L),L=1,7)
          WRITE(27,10211) (STHIG(L),L=1,7)
      ENDIF
      IF ( ( COMB_DOSE_LET_TABLES ) .OR. ( ALL_DOSE_LET_TABLES ) )
+      WRITE(27,10216) (STCOM(L),L=1,7)
      IF(RNWR)WRITE(27,20416) WLC
C**** DOSE RATES BY NUCLIDE
11000 DO 2050 J=1,40
      STHIG(J)=0.0
      STLOW(J)=0.0
      STCOM(J)=0.0
      STGLO(J)=0.0
      STGHI(J)=0.0
      STGCO(J)=0.0

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012
2050  CONTINUE
      SUML=0.0
      SUMH=0.0
      SUMC=0.0
      SUMGL=0.0
      SUMGH=0.0
      SUMGC=0.0
      DO 2109 J=1, NONCLD
      DO 2100 K=1, 4
      FACNE2=POPFAC*GENFAC(K)
      DO 2100 L=1, NORGN
      FACNEW=POPFAC*DOSFAC(K)*DFAC(L, K)
      STHIG(J)=STHIG(J)+DOSE(L, J, K, 2)*FACNEW*EXPP(JLOC, ILOC, J, K)
      STLOW(J)=STLOW(J)+DOSE(L, J, K, 1)*FACNEW*EXPP(JLOC, ILOC, J, K)
      STCOM(J)=STCOM(J)+DOSE(L, J, K, 1)*FACNEW*EXPP(JLOC, ILOC, J, K)
      < *LLET(L)+DOSE(L, J, K, 2)*FACNEW*EXPP(JLOC, ILOC, J, K)*HLET(L)
      IF(L.NE.NGEN) GO TO 2100
      STGHI(J)=STGHI(J)+GDOSE(L, J, K, 2)*FACNE2*EXPP(JLOC, ILOC, J, K)
      STGLO(J)=STGLO(J)+GDOSE(L, J, K, 1)*FACNE2*EXPP(JLOC, ILOC, J, K)
      STGCO(J)=STGCO(J)+GDOSE(L, J, K, 1)*FACNE2*EXPP(JLOC, ILOC, J, K)*
      A GLLET(L)+GDOSE(L, J, K, 2)*FACNE2*EXPP(JLOC, ILOC, J, K)*GHLET(L)
2100  CONTINUE
      SUML=SUML+STLOW(J)
      SUMH=SUMH+STHIG(J)
      SUMC=SUMC+STCOM(J)
      SUMGL=SUMGL+STGLO(J)
      SUMGH=SUMGH+STGHI(J)
      SUMGC=SUMGC+STGCO(J)
2109  CONTINUE
      WRITE(27, 101) DATE AND TIME
      WRITE(27, 12100) (NUCLID(L), L=1, NONCLD), TOTAL
12100 FORMAT('0', /30X, 'NUCLIDE DOSE/EXPOSURE SUMMARY'//
      A '0*** SELECTED INDIVIDUAL ***'/
      A 'ODOSE RATES: '/
      A ' WEIGHTED SUMS OF ORGAN DOSE RATES' /
      A 15X, ' NUCLIDES: ', 10(2X, A8)/(26X, 10(2X, A8)))
      IF ( (SEP_DOSE_LET_TABLES) .OR. (ALL_DOSE_LET_TABLES) ) THEN
          WRITE(27, 10205) (STLOW(L), L=1, NONCLD), SUML
          WRITE(27, 10210) (STHIG(L), L=1, NONCLD), SUMH
      ENDIF
      IF ( ( COMB_DOSE_LET_TABLES ) .OR. ( ALL_DOSE_LET_TABLES ) )
      +   WRITE(27, 10215) (STCOM(L), L=1, NONCLD), SUMC

      IF(RNWR) THEN
          WRITE(27, 20216) WLI
      ELSE

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
12345678901234567890123456789012345678901234567890123456789012

```

C****      SAVE DOSES BY PATHWAY FOR DATABASE
           DO 30 L = 1, NONCLD
             NUC_DOSES(L) = STCOM(L)
30          CONTINUE
           ENDIF

           IF(INDPOP.NE.1) GO TO 21000
           DO 2150 J=1,40
             STHIG(J)=0.0
             STLOW(J)=0.0
             STCOM(J)=0.0
             STGLO(J)=0.0
             STGHI(J)=0.0
             STGCO(J)=0.0
2150        CONTINUE
             SUML=0.0
             SUMH=0.0
             SUMC=0.0
             SUMGL=0.0
             SUMGH=0.0
             SUMGC=0.0
             DO 2209 J=1,NONCLD
               DO 2200 K=1,4
                 FACNE2=TOTFAC*GENFAC(K)
                 DO 2200 L=1,NORGN
                   FACNEW=TOTFAC*DOSFAC(K)*DFAC(L,K)
                   DO 2200 II=NOL,NOU
                     DO 2200 JJ=NRL,NRU
                       STHIG(J)=STHIG(J)+DOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW
                       STLOW(J)=STLOW(J)+DOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
                       STCOM(J)=STCOM(J)+DOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
                       > *LLET(L)+DOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW*HLET(L)
                       IF(L.NE.NGEN) GO TO 2200
                       STGLO(J)=STGLO(J)+GDOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2
                       STGHI(J)=STGHI(J)+GDOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2
                       STGCO(J)=STGCO(J)+GDOSE(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2*
2200          A  GLLET(L)+GDOSE(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2*GHLET(L)
                   CONTINUE
                   SUML=SUML+STLOW(J)
                   SUMH=SUMH+STHIG(J)
                   SUMC=SUMC+STCOM(J)
                   SUMGL=SUMGL+STGLO(J)
                   SUMGH=SUMGH+STGHI(J)
                   SUMGC=SUMGC+STGCO(J)
2209        CONTINUE
             WRITE(27,12400) (NUCLID(L),L=1,NONCLD),TOTAL

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012
12400  FORMAT(/'0*** MEAN INDIVIDUAL ***'/
A      'ODOSE RATES: '/
A      ' WEIGHTED SUMS OF ORGAN DOSE RATES' /
>      15X, ' NUCLIDES: ', 10(2X, A8) / (26X, 10(2X, A8)))
      IF ( (SEP_DOSE_LET_TABLES) .OR. (ALL_DOSE_LET_TABLES) ) THEN
          WRITE(27, 10205) (STLOW(L), L-1, NONCLD), SUML
          WRITE(27, 10210) (STHIG(L), L-1, NONCLD), SUMH
      ENDIF
      IF ( (COMB_DOSE_LET_TABLES) .OR. (ALL_DOSE_LET_TABLES) )
+      WRITE(27, 10215) (STCOM(L), L-1, NONCLD), SUMC
      IF(RNWR)WRITE(27, 20216) WLE
      DO 2350 K=1, 40
          STLOW(K)=0.0
          STHIG(K)=0.0
          STCOM(K)=0.0
          STGLO(K)=0.0
          STGHI(K)=0.0
          STGCO(K)=0.0
2350  CONTINUE
          SUML=0.0
          SUMH=0.0
          SUMC=0.0
          SUMGL=0.0
          SUMGH=0.0
          SUMGC=0.0
          DO 2409 J=1, NONCLD
              DO 2400 K=1, 4
                  TEXPP=0.0
                  DO 2403 II=NOL, NOU
                      DO 2403 JJ=NRL, NRU
                          TEXPP=TEXPP+EXPP(JJ, II, J, K)
2403  CONTINUE
                          TEXPP=TEXPP*.001
                          DO 2400 L=1, NORGN
                              STLOW(J)=STLOW(J)+DOSE(L, J, K, 1)*TEXPP*DOSFAC(K)*DFAC(L, K)
                              STHIG(J)=STHIG(J)+DOSE(L, J, K, 2)*TEXPP*DOSFAC(K)*DFAC(L, K)
                              STCOM(J)=STCOM(J)+DOSE(L, J, K, 1)*TEXPP*DOSFAC(K)*LLET(L)*
                              > DFAC(L, K)+DOSE(L, J, K, 2)*TEXPP*DOSFAC(K)*HLET(L)*DFAC(L, K)
                              IF(L.NE.NGEN) GO TO 2400
                              STGLO(J)=STGLO(J)+GDOSE(L, J, K, 1)*TEXPP*GENFAC(K)
                              STGHI(J)=STGHI(J)+GDOSE(L, J, K, 2)*TEXPP*GENFAC(K)
                              STGCO(J)=STGCO(J)+GDOSE(L, J, K, 1)*TEXPP*GENFAC(K)*GLLET(L)+
                              A GDOSE(L, J, K, 2)*TEXPP*GENFAC(K)*GHLET(L)
2400  CONTINUE
          SUML=SUML+STLOW(J)
          SUMH=SUMH+STHIG(J)

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

SUMC=SUMC+STCOM(J)
SUMGL=SUMGL+STGLO(J)
SUMGH=SUMGH+STGHI(J)
SUMGC=SUMGC+STGCO(J)
2409 CONTINUE
WRITE(27,12700) (NUCLID(L),L=1,NONCLD),TOTAL
12700 FORMAT(/'0*** COLLECTIVE POPULATION ***'/
A 'ODOSE RATES: '/
A ' WEIGHTED SUMS OF ORGAN DOSE RATES' /
> 15X, ' NUCLIDES: ',10(2X,A8)/(26X,10(2X,A8))
IF ( (SEP_DOSE_LET_TABLES) .OR. (ALL_DOSE_LET_TABLES) ) THEN
WRITE(27,10206) (STLOW(L),L=1,NONCLD),SUML
WRITE(27,10211) (STHIG(L),L=1,NONCLD),SUMH
ENDIF
IF ( ( COMB_DOSE_LET_TABLES ) .OR. ( ALL_DOSE_LET_TABLES ) )
+ WRITE(27,10216) (STCOM(L),L=1,NONCLD),SUMC
IF(RNWR)WRITE(27,20416) WLC
21000 CALL SUMMR2(RNWR,GSCFAC,DATE_AND_TIME)

RETURN
END

```

```

C-----
C
C          SUBROUTINE SUMMR2
C
C-----

```

SUBROUTINE SUMMR2(RNWR,GSCFAC,DATE\_AND\_TIME)

CC\*\* THE OUTPUT OF A TITLE WAS REMOVED 8/88

```

CHARACTER*8 NUCLID, CANC, GEN, ORGN, TOTAL, PUL
CHARACTER*36 DATE_AND_TIME

```

```

LOGICAL GENEFF,RNWR
LOGICAL SEP_RISK_LET_TABLES, COMB_RISK_LET_TABLES,
+ ALL_RISK_LET_TABLES
+ COMMON/RISK_LET_TABLES/SEP_RISK_LET_TABLES, COMB_RISK_LET_TABLES,
+ ALL_RISK_LET_TABLES

```

C

```

DIMENSION NUCLID(40), CANC(20), ORGN(20), GEN(3)

```

```

COMMON /NAMES_CHARS/ NUCLID, CANC, ORGN, GEN

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

COMMON/COMEX/EXPP(20,20,40,4),POP(20,20),POPFAC,TOTFAC,NOL,NOU,
> NRL,NRU,IDIST(20),ILOC,JLOC
COMMON/COMRN/WLRN(20,20),RRISK,RREF(2),RYRLL,NOREP,NRREP,NCREP
COMMON/COMCA/NCANC,RELABS(20),RISK(20,40,4,2),RTABLE(7),
> AGEX,YRLL(20,40,4,2)
COMMON/COMRF/REF(20,40,4),FTABLE(7)
COMMON/COMNU/NONCLD,PSIZE(40),GIABS(4,40),
> INDDPOP
COMMON/COMGEN/NGEN,GDOSE(3,40,4,2),GRISK(3,40,4,2),
> GENEFF,GRFAC(2),REPPER,GLLET(3),GHLET(3),GREF(3,40,4)
COMMON/COMUS/STLOW(40),STHIG(40),STCOM(40),STREQ(40),
A STGLO(40),STGHI(40),STGCO(40),STLLL(40),STHLL(40),STCLL(40),
B STGRQ(40)
DIMENSION RISFAC(4),REQFAC(4),GENFAC(4)
DATA PUL/'PULMINARY'/
DATA RISFAC/2*1.E-5,10.,1000./,
< REQFAC/1.,1.,1.E6,1.E8/,
< GENFAC/1.,1.,1.,100./,TOTAL/'TOTAL'/'/
C***** RISK RATES BY CANCER
RISFAC(4)=RISFAC(4)*GSCFAC
REQFAC(4)=REQFAC(4)*GSCFAC
GENFAC(4)=GENFAC(4)*GSCFAC
LP=1
DO 25 K=1,NCANC
IF(CANC(K).EQ.PUL)LP=K
25 CONTINUE
DO 50 J=1,40
STHIG(J)=0.0
STLOW(J)=0.0
STCOM(J)=0.0
STGLO(J)=0.0
STGHI(J)=0.0
STGCO(J)=0.0
STREQ(J)=0.0
STGRQ(J)=0.0
STLLL(J)=0.0
STHLL(J)=0.0
STCLL(J)=0.0
50 CONTINUE
SUML=0.0
SUMH=0.0
SUMC=0.0
SUMLLL=0.0
SUMHLL=0.0
SUMCLL=0.0
TBEQ=0.0

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

RLL=0.0
YRP=RYRLL*WLRN(JLOC, ILOC)*POPFAC
RR=RRISK*WLRN(JLOC, ILOC)*POPFAC
IF(WLRN(JLOC, ILOC).NE.0.0)RLL=RYRLL/RRISK
RP=RR
DO 109 L=1, NCANC
DO 100 K=1, 4
FACNEW=POPFAC*RISFAC(K)
FACNE2=POPFAC*GENFAC(K)
DO 100 J=1, NONCLD
STHIG(L)=STHIG(L)+RISK(L, J, K, 2)*FACNEW*EXPP(JLOC, ILOC, J, K)
STHLL(L)=STHLL(L)+YRLL(L, J, K, 2)*FACNEW*EXPP(JLOC, ILOC, J, K)
STLOW(L)=STLOW(L)+RISK(L, J, K, 1)*FACNEW*EXPP(JLOC, ILOC, J, K)
STLLL(L)=STLLL(L)+YRLL(L, J, K, 1)*FACNEW*EXPP(JLOC, ILOC, J, K)
STCOM(L)=STCOM(L)+RISK(L, J, K, 1)*FACNEW*EXPP(JLOC, ILOC, J, K)
< +RISK(L, J, K, 2)*FACNEW*EXPP(JLOC, ILOC, J, K)
STCLL(L)=STCLL(L)+YRLL(L, J, K, 1)*FACNEW*EXPP(JLOC, ILOC, J, K)
< +YRLL(L, J, K, 2)*FACNEW*EXPP(JLOC, ILOC, J, K)
STREQ(L)=STREQ(L)+REF(L, J, K)*POPFAC*REQFAC(K)*EXPP(JLOC,
< ILOC, J, K)
IF(L.GT.1) GO TO 100
TBEQ=TBEQ+REF(NCANC+1, J, K)*POPFAC*REQFAC(K)*EXPP(JLOC,
< ILOC, J, K)
STGRQ(L)=STGRQ(L)+GREF(L, J, K)*FACNE2*EXPP(JLOC, ILOC, J, K)
STGLO(L)=STGLO(L)+GRISK(L, J, K, 1)*FACNE2*EXPP(JLOC, ILOC, J, K)
STGHI(L)=STGHI(L)+GRISK(L, J, K, 2)*FACNE2*EXPP(JLOC, ILOC, J, K)
STGCO(L)=STGCO(L)+GRISK(L, J, K, 1)*FACNE2*EXPP(JLOC, ILOC, J, K)+
A GRISK(L, J, K, 2)*FACNE2*EXPP(JLOC, ILOC, J, K)
100 CONTINUE
SUML=SUML+STLOW(L)
SUMH=SUMH+STHIG(L)
SUMLLL=SUMLLL+STLLL(L)
SUMHLL=SUMHLL+STHLL(L)
SUMCLL=SUMCLL+STCLL(L)
SUMC=SUMC+STCOM(L)
IF(L.EQ.LP) STCPH=STCLL(L)
IF(L.EQ.LP) STCCH=STCOM(L)
IF(STLOW(L).NE.0.0)STLLL(L)=STLLL(L)/STLOW(L)
IF(STHIG(L).NE.0.0)STHLL(L)=STHLL(L)/STHIG(L)
IF(STCOM(L).NE.0.0)STCLL(L)=STCLL(L)/STCOM(L)
109 CONTINUE
IF(SUML.NE.0.0)SUMLLL=SUMLLL/SUML
IF(SUMH.NE.0.0)SUMHLL=SUMHLL/SUMH
IF(SUMC.NE.0.0) YRT=(RYRLL*WLRN(JLOC, ILOC)*POPFAC+SUMCLL)/
A (SUMC+RRISK*POPFAC*WLRN(JLOC, ILOC))
IF(SUMC.NE.0.0)SUMCLL=SUMCLL/SUMC

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

YRP=STCPH+YRP
RP=RP+STCCH
IF(RP.NE.0.0) YRP=YRP/RP
WRITE(27,101)DATE_AND_TIME
101  FORMAT('1',T10,'DATE',2X,A)
WRITE(27,10100)
WRITE(27,10102)(CANC(L),L=1,NCANC),TOTAL
10100  FORMAT('0',/30X,'RISK/RISK EQUIVALENT SUMMARY'//
A  '0*** SELECTED INDIVIDUAL ***'/
B  'OLIFETIME FATAL CANCER RISK:'/)
10102  FORMAT(16X,' CANCERS:',10(2X,A8)/(26X,10(2X,A8)))
IF ( (SEP_RISK_LET_TABLES) .OR. (ALL_RISK_LET_TABLES) ) THEN
WRITE(27,10205) (STLOW(L),L=1,NCANC),SUML
WRITE(27,10210) (STHIG(L),L=1,NCANC),SUMH
ENDIF
IF ( ( COMB_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) )
+  WRITE(27,10215) (STCOM(L),L=1,NCANC),SUMC
IF(RNWR)WRITE(27,20216) RR
TT=SUMC+RR
20216  FORMAT('OLUNG CANCER RISK FROM RADON DAUGHTER EXPOSURE ',
A  8X,1PE10.2)
IF(RNWR)WRITE(27,20217) TT
20217  FORMAT(' TOTAL FATAL CANCER RISK FROM ALL EXPOSURES',
A  12X,1PE10.2)
10205  FORMAT(' LOW LET ',16X,10(1PE10.2)/(26X,10(1PE10.2)))
10210  FORMAT(' HIGH LET',16X,10(1PE10.2)/(26X,10(1PE10.2)))
10215  FORMAT(' TOTAL',18X,10(1PE10.2)/(26X,10(1PE10.2)))
WRITE(27,30100)
30100  FORMAT('O AVERAGE LIFE LOSS PER PREMATURE DEATH:')
WRITE(27,10102) (CANC(L),L=1,NCANC),TOTAL
IF ( (SEP_RISK_LET_TABLES) .OR. (ALL_RISK_LET_TABLES) ) THEN
WRITE(27,10250) (STLLL(L),L=1,NCANC),SUMLLL
WRITE(27,10255) (STHLL(L),L=1,NCANC),SUMHLL
ENDIF
IF ( ( COMB_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) )
+  WRITE(27,10260) (STCLL(L),L=1,NCANC),SUMCLL
IF(RNWR)WRITE(27,20218) RLL
20218  FORMAT('O AVG LIFE LOSS FROM RADON DAUGHTER EXPOSURE ',
A  'FOR LUNG',3X,1PE10.2)
IF(RNWR)WRITE(27,20219)YRP
20219  FORMAT(' AVG LIFE LOSS FROM ALL EXPOSURES FOR LUNG',
A  13X,1PE10.2)
IF(RNWR)WRITE(27,20220) YRT
20220  FORMAT(' AVG LIFE LOSS FROM ALL EXPOSURES (TOTAL) ',
A  12X,1PE10.2)
10250  FORMAT(' LOW LET (YR)',12X,10(1PE10.2)/(26X,10(1PE10.2)))

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012
10255  FORMAT(' HIGH LET (YR)',11X,10(1PE10.2)/(26X,10(1PE10.2)))
10260  FORMAT(' COMBINED (YR)',11X,10(1PE10.2)/(26X,10(1PE10.2)))
      RQ=RREF(1)*WLRN(JLOC,ILOC)*POPFAC
      RT=RREF(2)*WLRN(JLOC,ILOC)*POPFAC
      TTP=RQ+STREQ(LP)
      TTQ=RT+TBEQ
      IF(RNWR)WRITE(27,20221) RQ
      IF(RNWR)WRITE(27,20222) RT
      IF(RNWR)WRITE(27,20224) TTP
      IF(RNWR)WRITE(27,20223) TTQ
20221  FORMAT(' LUNG RISK EQUIVALENT(MREM/YR) FROM RADON ',
A      ' DAUGHTER EXPOSURE ',4X,1PE10.2)
20321  FORMAT(' LUNG RISK EQ. (PERSON REM/YR) FROM RADON ',
A      ' DAUGHTER EXPOSURE',6X,1PE10.2)
20222  FORMAT(' WHOLE BODY RISK EQ (MREM/YR) FROM RADON ',
A      ' DAUGHTER EXPOSURE ',6X,1PE10.2)
20322  FORMAT(' WHOLE BODY RISK EQ (PERSON REM/YR) FROM RADON ',
A      ' DAUGHTER EXPOSURE',1X,1PE10.2)
20223  FORMAT(' WHOLE BODY RISK EQ (MREM/YR) FROM ALL EXPOSURES',
A17X,1PE10.2)
20224  FORMAT(' PULMINARY RISK EQ (MREM/YR) FOR ALL EXPOSURES',
A      20X,1PE10.2)
20323  FORMAT(' WHOLE BODY RISK EQ (PERSON REM/YR) FROM ALL EXPOSURES',
A11X,1PE10.2)
20324  FORMAT(' PULMINARY RISK EQ (PERSON REM/YR) FOR ALL EXPOSURES',
A      14X,1PE10.2)

10225  FORMAT(' LOW LET (EFFECTS/BIRTH)',1X,10(1PE10.2)/
>      (26X,10(1PE10.2)))
10230  FORMAT(' HIGH LET (EFFECTS/BIRTH)',10(1PE10.2)/
>      (26X,10(1PE10.2)))
10235  FORMAT(' COMBINED (EFFECTS/BIRTH)',10(1PE10.2)/
>      (26X,10(1PE10.2)))
30300  FORMAT(' OGENETIC RISK EQUIVALENT: '/
A      '(MREM/YR)',15X,10(1PE10.2)/(26X,10(1PE10.2)))
      IF(INDPOP.NE.1) GO TO 1000
      DO 150 J=1,40
      STHIG(J)=0.0
      STLOW(J)=0.0
      STCOM(J)=0.0
      STGLO(J)=0.0
      STGHI(J)=0.0
      STGCO(J)=0.0
      STREQ(J)=0.0
      STLLL(J)=0.0
      STHLL(J)=0.0

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

150      STCLL(J)=0.0
        STGRQ(J)=0.0
        CONTINUE
        TBEQ=0.0
        SUML=0.0
        SUMH=0.0
        SUMC=0.0
        SUMLLL=0.0
        SUMHLL=0.0
        SUMCLL=0.0
        RRM=0.0
        RLL=0.0
        RQM=0.0
        RTM=0.0
        YRP=0.0
        DO 206 II=NOL,NOU
        DO 206 JJ=NRL,NRU
        RQM=RQM+RREF(1)*WLRN(JJ,II)*TOTFAC
        RTM=RTM+RREF(2)*WLRN(JJ,II)*TOTFAC
        RRM=RRISK*WLRN(JJ,II)*TOTFAC+RRM
        YRP=RYRLL*WLRN(JJ,II)*TOTFAC+YRP
206     CONTINUE
        IF(RRM.NE.0.0)RLL=YRP/RRM
        RP=YRP
        DO 209 L=1,NCANC
        DO 200 K=1,4
        FACNEW=TOTFAC*RISFAC(K)
        FACNE2=TOTFAC*GENFAC(K)
        DO 200 J=1,NONCLD
        DO 200 II=NOL,NOU
        DO 200 JJ=NRL,NRU
        STHIG(L)=STHIG(L)+RISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW
        STHLL(L)=STHLL(L)+YRLL(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW
        STLOW(L)=STLOW(L)+RISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
        STLLL(L)=STLLL(L)+YRLL(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
        STCOM(L)=STCOM(L)+RISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
        > +RISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW
        STCLL(L)=STCLL(L)+YRLL(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
        > +YRLL(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW
        STREQ(L)=STREQ(L)+REF(L,J,K)*EXPP(JJ,II,J,K)*TOTFAC
        > *REQFAC(K)
        IF(L.NE.1) GO TO 200
        TBEQ=TBEQ+REF(NCANC+1,J,K)*EXPP(JJ,II,J,K)*TOTFAC*REQFAC(K)
        STGRQ(L)=STGRQ(L)+GREF(L,J,K)*EXPP(JJ,II,J,K)*TOTFAC*GENFAC(K)
        STGLO(L)=STGLO(L)+GRISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2
        STGHI(L)=STGHI(L)+GRISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012
200      STGCO(L)=STGCO(L)+GRISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2+
A      GRISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2
CONTINUE
SUML=SUML+STLOW(L)
SUMH=SUMH+STHIG(L)
SUMC=SUMC+STCOM(L)
SUMLLL=SUMLLL+STLLL(L)
SUMHLL=SUMHLL+STHLL(L)
SUMCLL=SUMCLL+STCLL(L)
IF(L.EQ.LP) STCPH=STCLL(LP)
IF(L.EQ.LP) STCCH=STCOM(LP)
IF(STLOW(L).NE.0.0)STLLL(L)=STLLL(L)/STLOW(L)
IF(STHIG(L).NE.0.0)STHLL(L)=STHLL(L)/STHIG(L)
IF(STCOM(L).NE.0.0)STCLL(L)=STCLL(L)/STCOM(L)
209      CONTINUE
IF(SUML.NE.0.0)SUMLLL=SUMLLL/SUML
IF(SUMH.NE.0.0)SUMHLL=SUMHLL/SUMH
IF(SUMC.NE.0.0) YRT=(YRP+SUMCLL)/(RRM+SUMC)
YRP=STCPH+YRP
RP=RRM+STCCH
IF(RP.NE.0.0)YRP=YRP/RP
IF(SUMC.NE.0.0)SUMCLL=SUMCLL/SUMC
WRITE(27,101)DATE_AND_TIME
WRITE(27,10400)
WRITE(27,10402)(CANC(L),L=1,NCANC),TOTAL
10400      FORMAT('0'/30X,'RISK/RISK EQUIVALENT SUMMARY'//
A      '0*** MEAN INDIVIDUAL ***'/
B      'OLIFETIME FATAL CANCER RISK:'/)
10402      FORMAT(16X,' CANCERS:',10(2X,A8)/(26X,10(2X,A8)))
IF ( ( SEP_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) ) THEN
WRITE(27,10205) (STLOW(L),L=1,NCANC),SUML
WRITE(27,10210) (STHIG(L),L=1,NCANC),SUMH
ENDIF
IF ( ( COMB_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) )
+      WRITE(27,10215) (STCOM(L),L=1,NCANC),SUMC

IF(RNWR)WRITE(27,20216) RRM
TTM=SUMC+RRM
IF(RNWR)WRITE(27,20217)TTM
WRITE(27,30100)
WRITE(27,10102) (CANC(L),L=1,NCANC),TOTAL
IF ( ( SEP_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) ) THEN
WRITE(27,10250) (STLLL(L),L=1,NCANC),SUMLLL
WRITE(27,10255) (STHLL(L),L=1,NCANC),SUMHLL
ENDIF
IF ( ( COMB_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) )

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```
+     WRITE(27,10260) (STCLL(L),L=1,NCANC),SUMCLL

      IF(RNWR)WRITE(27,20218) RLL
      IF(RNWR)WRITE(27,20219) YRP
      IF(RNWR)WRITE(27,20220) YRT
      IF(RNWR)WRITE(27,20221) RQM
      IF(RNWR)WRITE(27,20222) RTM
      TTP=RQM+STREQ(LP)
      IF(RNWR)WRITE(27,20224) TTP
      TTQM=RTM+TBEQ
      IF(RNWR)WRITE(27,20223) TTQM
      DO 350 K=1,40
      STLOW(K)=0.0
      STHIG(K)=0.0
      STCOM(K)=0.0
      STGLO(K)=0.0
      STGHI(K)=0.0
      STGCO(K)=0.0
      STREQ(K)=0.0
      STLLL(K)=0.0
      STHLL(K)=0.0
      STCLL(K)=0.0
      STGRQ(K)=0.0
350    CONTINUE
      SUML=0.0
      TBEQ=0.0
      SUMH=0.0
      SUMC=0.0
      SUMLLL=0.0
      SUMHLL=0.0
      SUMCLL=0.0
      WLT=0.0
      DO 401 II=NOL,NOU
      DO 401 JJ=NRL,NRU
      WLT=WLT+WLRN(JJ,II)
401    CONTINUE
      RRC=RRISK*WLT/AGEX
      RQC=RREF(1)*WLT*.001
      RTC=RREF(2)*WLT*.001
      DO 400 K=1,4
      DO 400 J=1,NONCLD
      TEXPP=0.0
      DO 403 II=NOL,NOU
      DO 403 JJ=NRL,NRU
      TEXPP=TEXPP+EXPP(JJ,II,J,K)
403    CONTINUE
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

DO 400 L=1,NCANC
STLOW(L)=STLOW(L)+RISK(L,J,K,1)*TEXPP*RISFAC(K)/AGEX
STLLL(L)=STLLL(L)+YRLL(L,J,K,1)*TEXPP*RISFAC(K)/AGEX
SUML=SUML+RISK(L,J,K,1)*TEXPP*RISFAC(K)/AGEX
SUMLLL=SUMLLL+YRLL(L,J,K,1)*TEXPP*RISFAC(K)/AGEX
STHIG(L)=STHIG(L)+RISK(L,J,K,2)*TEXPP*RISFAC(K)/AGEX
STHLL(L)=STHLL(L)+YRLL(L,J,K,2)*TEXPP*RISFAC(K)/AGEX
SUMH=SUMH+RISK(L,J,K,2)*TEXPP*RISFAC(K)/AGEX
SUMHLL=SUMHLL+YRLL(L,J,K,2)*TEXPP*RISFAC(K)/AGEX
STCOM(L)=STCOM(L)+RISK(L,J,K,1)*TEXPP*RISFAC(K)/AGEX+
> RISK(L,J,K,2)*TEXPP*RISFAC(K)/AGEX
STCLL(L)=STCLL(L)+YRLL(L,J,K,1)*TEXPP*
> RISFAC(K)/AGEX+
> YRLL(L,J,K,2)*TEXPP*RISFAC(K)/AGEX
SUMC=SUMC+RISK(L,J,K,1)*TEXPP*RISFAC(K)/AGEX+
A RISK(L,J,K,2)*TEXPP*RISFAC(K)/AGEX
SUMCLL=SUMCLL+YRLL(L,J,K,1)*TEXPP*RISFAC(K)/AGEX+
A YRLL(L,J,K,2)*TEXPP*RISFAC(K)/AGEX
STREQ(L)=STREQ(L)+REF(L,J,K)*TEXPP*REQFAC(K)*.001
IF(L.NE.1) GO TO 400
TBEQ=TBEQ+REF(NCANC+1,J,K)*TEXPP*REQFAC(K)*.001
STGRQ(L)=STGRQ(L)+GREF(L,J,K)*TEXPP*GENFAC(K)*.001
STGLO(L)=STGLO(L)+GRISK(L,J,K,1)*TEXPP*GENFAC(K)*REPPER
STGHI(L)=STGHI(L)+GRISK(L,J,K,2)*TEXPP*GENFAC(K)*REPPER
STGCO(L)=STGCO(L)+GRISK(L,J,K,1)*TEXPP*GENFAC(K)*REPPER+
A GRISK(L,J,K,2)*TEXPP*GENFAC(K)*REPPER
400 CONTINUE
WRITE(27,101)DATE_AND_TIME

WRITE(27,10700) (CANC(L),L=1,NCANC),TOTAL
10700 FORMAT('0',/29X,'RISK/RISK EQUIVALENT SUMMARY'/
1 /'0*** COLLECTIVE POPULATION ***'/
A 'COLLECTIVE FATAL CANCER RISK: '/
A 16X,' CANCERS:',10(2X,A8)/(26X,10(2X,A8)))
10705 FORMAT(' LOW LET(DEATHS/YR)',6X,10(1PE10.2)/
A (26X,10(1PE10.2)))
10710 FORMAT(' HIGH LET(DEATHS/YR)',5X,10(1PE10.2)/
A (26X,10(1PE10.2)))
IF ( (SEP_RISK_LET_TABLES ) .OR. (ALL_RISK_LET_TABLES) ) THEN
WRITE(27,10705) (STLOW(L),L=1,NCANC),SUML
WRITE(27,10710) (STHIG(L),L=1,NCANC),SUMH
ENDIF
IF ( ( COMB_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) )
+ WRITE(27,10715) (STCOM(L),L=1,NCANC),SUMC
IF(RNWR)WRITE(27,20316) RRC
TTC=SUMC+RRC

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012
IF(RNWR)WRITE(27,20317) TTC
20316  FORMAT('OLUNG CANCER RISK(DEATHS/YR) FROM RADON DAUGHTER',
> ' EXPOSURE',8X,1PE10.2)
20317  FORMAT(' TOTAL FATAL CANCER RISK(DEATHS/YR) FROM ALL',
> ' EXPOSURE',12X,1PE10.2)
10750  FORMAT(' LOST LOW LET',12X,10(1PE10.2)/(26X,10(1PE10.2)))
10755  FORMAT(' LIFE LOST HIGH LET',6X,10(1PE10.2)/(26X,10(1PE10.2)))
10760  FORMAT(' LIFE LOST COMBINED',6X,10(1PE10.2)/(26X,10(1PE10.2)))
10715  FORMAT(' TOTAL (DEATHS/YR)',6X,10(1PE10.2)/
A (26X,10(1PE10.2)))
IF(RNWR)WRITE(27,20321) RQC
IF(RNWR)WRITE(27,20322) RTC
TTPC=RQC+STREQ(LP)
IF(RNWR)WRITE(27,20324) TTPC
TTQC=RTC+TBEQ
IF(RNWR)WRITE(27,20323) TTQC
10720  FORMAT('OGENETIC RISK EQUIVALENT:'/
1 ' (PERSON REM/YR)',9X,10(1PE10.2)/
A (26X,10(1PE10.2)))

10725  FORMAT(' LOW LET(EFFECTS/YR)',5X,10(1PE10.2)/
> (26X,10(1PE10.2)))
10730  FORMAT(' HIGH LET(EFFECTS/YR)',4X,10(1PE10.2)/
> (26X,10(1PE10.2)))
10735  FORMAT(' COMBINED(EFFECTS/YR)',4X,10(1PE10.2)/
> (26X,10(1PE10.2)))
WRITE(27,10720) STGRQ(1)
C****  RISK RATES BY PATHWAY
1000   DO 1050 J=1,40
        STHIG(J)=0.0
        STLOW(J)=0.0
        STCOM(J)=0.0
        STREQ(J)=0.0
        STGRQ(J)=0.0
        STGLO(J)=0.0
        STGHI(J)=0.0
        STGCO(J)=0.0
1050   CONTINUE
        DO 1199 K=1,4
        DO 1100 L=1,NCANC
        FACNEW=POPFAC*RISFAC(K)
        FACNE2=POPFAC*GENFAC(K)
        DO 1100 J=1,NONCLD
        STHIG(K)=STHIG(K)+RISK(L,J,K,2)*FACNEW*EXPP(JLOC,ILOC,J,K)
        STLOW(K)=STLOW(K)+RISK(L,J,K,1)*FACNEW*EXPP(JLOC,ILOC,J,K)
        STCOM(K)=STCOM(K)+RISK(L,J,K,1)*FACNEW*EXPP(JLOC,ILOC,J,K)

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012
< +RISK(L,J,K,2)*FACNEW*EXPP(JLOC,ILOC,J,K)
  IF(L.NE.1) GO TO 1100
  STREQ(K)=STREQ(K)+REF(NCANC+1,J,K)*POPFAC*REQFAC(K)*
< EXPP(JLOC,ILOC,J,K)
  STGRQ(K)=STGRQ(K)+GREF(L,J,K)*POPFAC*GENFAC(K)*
< EXPP(JLOC,ILOC,J,K)
  STGHI(K)=STGHI(K)+GRISK(L,J,K,2)*FACNE2*EXPP(JLOC,ILOC,J,K)
  STGLO(K)=STGLO(K)+GRISK(L,J,K,1)*FACNE2*EXPP(JLOC,ILOC,J,K)
  STGCO(K)=STGCO(K)+GRISK(L,J,K,1)*FACNE2*EXPP(JLOC,ILOC,J,K)+
A GRISK(L,J,K,2)*FACNE2*EXPP(JLOC,ILOC,J,K)
1100 CONTINUE
1103 IF(K.GT.2) GO TO 1101
  STHIG(5)=STHIG(5)+STHIG(K)
  STLOW(5)=STLOW(5)+STLOW(K)
  STCOM(5)=STCOM(5)+STCOM(K)
  STGHI(5)=STGHI(5)+STGHI(K)
  STGLO(5)=STGLO(5)+STGLO(K)
  STGCO(5)=STGCO(5)+STGCO(K)
  STGRQ(5)=STGRQ(5)+STGRQ(K)
  STREQ(5)=STREQ(5)+STREQ(K)
1101 IF(K.LT.3) GO TO 1102
  STHIG(6)=STHIG(6)+STHIG(K)
  STLOW(6)=STLOW(6)+STLOW(K)
  STCOM(6)=STCOM(6)+STCOM(K)
  STGRQ(6)=STGRQ(6)+STGRQ(K)
  STREQ(6)=STREQ(6)+STREQ(K)
  STGHI(6)=STGHI(6)+STGHI(K)
  STGLO(6)=STGLO(6)+STGLO(K)
  STGCO(6)=STGCO(6)+STGCO(K)
1102 STHIG(7)=STHIG(7)+STHIG(K)
  STLOW(7)=STLOW(7)+STLOW(K)
  STCOM(7)=STCOM(7)+STCOM(K)
  STGRQ(7)=STGRQ(7)+STGRQ(K)
  STREQ(7)=STREQ(7)+STREQ(K)
  STGHI(7)=STGHI(7)+STGHI(K)
  STGLO(7)=STGLO(7)+STGLO(K)
  STGCO(7)=STGCO(7)+STGCO(K)
1199 CONTINUE
  WRITE(27,101)DATE_AND_TIME
  WRITE(27,11100)
  WRITE(27,11102)
11100 FORMAT('0',/30X,'PATHWAY RISK/RISK EQUIVALENT SUMMARY'
A // '0*** SELECTED INDIVIDUAL ***' /
A 'OLIFETIME FATAL CANCER RISK: ' /)
11102 FORMAT('          PATHWAYS:  INGESTION ',
< ' INHALATION   AIR      GROUND',

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

< ' INTERNAL EXTERNAL TOTAL' /
< 45X, ' IMMERSION SURFACE' )
IF ( (SEP_RISK_LET_TABLES) .OR. (ALL_RISK_LET_TABLES) ) THEN
  WRITE(27,10205) (STLOW(L),L-1,7)
  WRITE(27,10210) (STHIG(L),L-1,7)
ENDIF
IF ( ( COMB_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) )
+  WRITE(27,10215) (STCOM(L),L-1,7)
IF(RNWR)WRITE(27,20216) RR
IF(RNWR)WRITE(27,20217) TT
IF(RNWR)WRITE(27,20422) RT
20422 FORMAT('OWHOLE BODY RISK EQ(MREM/YR) FROM RADON' ,
> ' DAUGHTER EXPOSURE' ,7X,1PE10.2)
IF(RNWR)WRITE(27,20223) TTQ

IF(INDPOP.NE.1) GO TO 11000
DO 1150 J=1,40
  STHIG(J)=0.0
  STLOW(J)=0.0
  STCOM(J)=0.0
  STREQ(J)=0.0
  STGRQ(J)=0.0
  STGLO(J)=0.0
  STGHI(J)=0.0
  STGCO(J)=0.0
1150 CONTINUE
  TBEQ=0.0
  DO 1299 K=1,4
    FACNEW=TOTFAC*RISFAC(K)
    FACNE2=TOTFAC*GENFAC(K)
    DO 1200 L=1,NCANG
      DO 1200 J=1,NONCLD
        DO 1200 II=NOL,NOU
          DO 1200 JJ=NRL,NRU
            STHIG(K)=STHIG(K)+RISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW
            STLOW(K)=STLOW(K)+RISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
            STCOM(K)=STCOM(K)+RISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
            > +RISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW
            IF(L.GT.1) GO TO 1200
            STREQ(K)=STREQ(K)+REF(NCANG+1,J,K)*EXPP(JJ,II,J,K)*
            > TOTFAC*REQFAC(K)
            STGRQ(K)=STGRQ(K)+GREF(L,J,K)*EXPP(JJ,II,J,K)*
            > TOTFAC*GENFAC(K)
            STGHI(K)=STGHI(K)+GRISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2
            STGLO(K)=STGLO(K)+GRISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2
            STGCO(K)=STGCO(K)+GRISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2+

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
12345678901234567890123456789012345678901234567890123456789012

```

      A GRISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2
1200 CONTINUE
1203 IF(K.GT.2) GO TO 1201
      STHIG(5)=STHIG(5)+STHIG(K)
      STLOW(5)=STLOW(5)+STLOW(K)
      STCOM(5)=STCOM(5)+STCOM(K)
      STGRQ(5)=STGRQ(5)+STGRQ(K)
      STREQ(5)=STREQ(5)+STREQ(K)
      STGLO(5)=STGLO(5)+STGLO(K)
      STGHI(5)=STGHI(5)+STGHI(K)
      STGCO(5)=STGCO(5)+STGCO(K)
1201 IF(K.LT.3) GO TO 1202
      STHIG(6)=STHIG(6)+STHIG(K)
      STLOW(6)=STLOW(6)+STLOW(K)
      STCOM(6)=STCOM(6)+STCOM(K)
      STGRQ(6)=STGRQ(6)+STGRQ(K)
      STREQ(6)=STREQ(6)+STREQ(K)
      STGLO(6)=STGLO(6)+STGLO(K)
      STGHI(6)=STGHI(6)+STGHI(K)
      STGCO(6)=STGCO(6)+STGCO(K)
1202 STHIG(7)=STHIG(7)+STHIG(K)
      STLOW(7)=STLOW(7)+STLOW(K)
      STCOM(7)=STCOM(7)+STCOM(K)
      STGRQ(7)=STGRQ(7)+STGRQ(K)
      STREQ(7)=STREQ(7)+STREQ(K)
      STGLO(7)=STGLO(7)+STGLO(K)
      STGHI(7)=STGHI(7)+STGHI(K)
      STGCO(7)=STGCO(7)+STGCO(K)
1299 CONTINUE
      WRITE(27,101)DATE_AND_TIME
      WRITE(27,11400)
      WRITE(27,11102)
11400 FORMAT('/0',/
> 29X,'PATHWAY RISK/RISK EQUIVALENT SUMMARY'//
A '0***MEAN INDIVIDUAL ***'/
B 'OLIFETIME FATAL CANCER RISK:'/)

      IF ( (SEP_RISK_LET_TABLES ) .OR. (ALL_RISK_LET_TABLES) ) THEN
          WRITE(27,10205) (STLOW(L),L=1,7)
          WRITE(27,10210) (STHIG(L),L=1,7)
      ENDIF
      IF ( ( COMB_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) )
+      WRITE(27,10215) (STCOM(L),L=1,7)

      IF(RNWR)WRITE(27,20216) RRM

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0             1             2             3             4             5             6             7  
123456789012345678901234567890123456789012345678901234567890123456789012

```
IF(RNWR)WRITE(27,20217) TTM
IF(RNWR)WRITE(27,20422) RTM
IF(RNWR)WRITE(27,20223) TTQM
DO 1350 K=1,40
STLOW(K)=0.0
STHIG(K)=0.0
STCOM(K)=0.0
STGLO(K)=0.0
STGHI(K)=0.0
STGCO(K)=0.0
STREQ(K)=0.0
STGRQ(K)=0.0
1350 CONTINUE
DO 1499 K=1,4
DO 1400 J=1,NONCLD
TEXPP=0.0
DO 1404 II=NOL,NOU
DO 1404 JJ=NRL,NRU
TEXPP=TEXPP+EXPP(JJ,II,J,K)
1404 CONTINUE
DO 1400 L=1,NCANC
STLOW(K)=STLOW(K)+RISK(L,J,K,1)*TEXPP*RISFAC(K)/AGEX
STHIG(K)=STHIG(K)+RISK(L,J,K,2)*TEXPP*RISFAC(K)/AGEX
STCOM(K)=STCOM(K)+RISK(L,J,K,1)*TEXPP*RISFAC(K)/AGEX+
> RISK(L,J,K,2)*TEXPP*RISFAC(K)/AGEX
IF(L.GT.1) GO TO 1400
STREQ(K)=STREQ(K)+REF(NCANC+1,J,K)*TEXPP*REQFAC(K)*.001
STGRQ(K)=STGRQ(K)+GREF(L,J,K)*TEXPP*GENFAC(K)*.001
STGLO(K)=STGLO(K)+GRISK(L,J,K,1)*TEXPP*GENFAC(K)*REPPER
STGHI(K)=STGHI(K)+GRISK(L,J,K,2)*TEXPP*GENFAC(K)*REPPER
STGCO(K)=STGCO(K)+GRISK(L,J,K,1)*TEXPP*GENFAC(K)*REPPER+
A GRISK(L,J,K,2)*TEXPP*GENFAC(K)*REPPER
1400 CONTINUE
1403 IF(K.GT.2) GO TO 1401
STLOW(5)=STLOW(5)+STLOW(K)
STHIG(5)=STHIG(5)+STHIG(K)
STCOM(5)=STCOM(5)+STCOM(K)
STGLO(5)=STGLO(5)+STGLO(K)
STGHI(5)=STGHI(5)+STGHI(K)
STGCO(5)=STGCO(5)+STGCO(K)
STGRQ(5)=STGRQ(5)+STGRQ(K)
STREQ(5)=STREQ(5)+STREQ(K)
GO TO 1402
1401 STLOW(6)=STLOW(6)+STLOW(K)
STHIG(6)=STHIG(6)+STHIG(K)
STCOM(6)=STCOM(6)+STCOM(K)
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

          STGLO(6)-STGLO(6)+STGLO(K)
          STGHI(6)-STGHI(6)+STGHI(K)
          STGCO(6)-STGCO(6)+STGCO(K)
          STGRQ(6)-STGRQ(6)+STGRQ(K)
          STREQ(6)-STREQ(6)+STREQ(K)
1402      STLOW(7)-STLOW(7)+STLOW(K)
          STHIG(7)-STHIG(7)+STHIG(K)
          STCOM(7)-STCOM(7)+STCOM(K)
          STGRQ(7)-STGRQ(7)+STGRQ(K)
          STREQ(7)-STREQ(7)+STREQ(K)
          STGLO(7)-STGLO(7)+STGLO(K)
          STGHI(7)-STGHI(7)+STGHI(K)
          STGCO(7)-STGCO(7)+STGCO(K)
1499      CONTINUE
          WRITE(27,101)DATE_AND_TIME
          WRITE(27,11700)
          WRITE(27,11102)
11700     FORMAT(/'0',/
> 29X,'PATHWAY RISK/RISK EQUIVALENT SUMMARY'//
A '0*** COLLECTIVE POPULATION ***'/
B '0COLLECTIVE FATAL CANCER RISK:'//

          IF ( (SEP_RISK_LET_TABLES) .OR. (ALL_RISK_LET_TABLES) ) THEN
              WRITE(27,10705) (STLOW(L),L=1,7)
              WRITE(27,10710) (STHIG(L),L=1,7)
          ENDIF
          IF ( ( COMB_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) )
+           WRITE(27,10715) (STCOM(L),L=1,7)

          IF(RNWR)WRITE(27,20316) RRC
          IF(RNWR)WRITE(27,20317) TTC
          IF(RNWR)WRITE(27,20522) RTC
20522     FORMAT('OWHOLE BODY RISK EQ(PERSON REM/YR) FROM RADON ',
> 'DAUGHTER EXPOSURE',2X,1PE10.2)
          WRITE(27,20323) TTQC
          WRITE(27,10720) (STGRQ(L),L=1,7)
C****    RISK RATES BY NUCLIDE
11000     DO 2050 J=1,40
          STHIG(J)=0.0
          STLOW(J)=0.0
          STCOM(J)=0.0
          STGLO(J)=0.0
          STGHI(J)=0.0
          STGCO(J)=0.0
          STREQ(J)=0.0
          STGRQ(J)=0.0

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

2050  CONTINUE
      SUML=0.0
      SUMH=0.0
      SUMC=0.0
      SUMGL=0.0
      SUMGH=0.0
      SUMGC=0.0
      SUMRQ=0.0
      SUMGQ=0.0
      DO 2109 J=1, NONCLD
      DO 2100 K=1, 4
      FACNEW=POPFAC*RISFAC(K)
      FACNE2=POPFAC*GENFAC(K)
      DO 2100 L=1, NCANC
      STHIG(J)=STHIG(J)+RISK(L, J, K, 2)*FACNEW*EXPP(JLOC, ILOC, J, K)
      STLOW(J)=STLOW(J)+RISK(L, J, K, 1)*FACNEW*EXPP(JLOC, ILOC, J, K)
      STCOM(J)=STCOM(J)+RISK(L, J, K, 1)*FACNEW*EXPP(JLOC, ILOC, J, K)
      < +RISK(L, J, K, 2)*FACNEW*EXPP(JLOC, ILOC, J, K)
      IF(L.GT.1) GO TO 2100
      STREQ(J)=STREQ(J)+REF(NCANC+1, J, K)*POPFAC*REQFAC(K)*
      < EXPP(JLOC, ILOC, J, K)
      SUMRQ=SUMRQ+REF(NCANC+1, J, K)*POPFAC*REQFAC(K)*
      < EXPP(JLOC, ILOC, J, K)
      STGRQ(J)=STGRQ(J)+GREF(L, J, K)*POPFAC*GENFAC(K)*
      < EXPP(JLOC, ILOC, J, K)
      SUMGQ=SUMGQ+GREF(L, J, K)*POPFAC*GENFAC(K)*
      < EXPP(JLOC, ILOC, J, K)
      STGHI(J)=STGHI(J)+GRISK(L, J, K, 2)*EXPP(JLOC, ILOC, J, K)*FACNE2
      STGLO(J)=STGLO(J)+GRISK(L, J, K, 1)*EXPP(JLOC, ILOC, J, K)*FACNE2
      STGCO(J)=STGCO(J)+GRISK(L, J, K, 1)*EXPP(JLOC, ILOC, J, K)*FACNE2
      A GRISK(L, J, K, 2)*EXPP(JLOC, ILOC, J, K)*FACNE2
2100  CONTINUE
      SUML=SUML+STLOW(J)
      SUMH=SUMH+STHIG(J)
      SUMC=SUMC+STCOM(J)
      SUMGL=SUMGL+STGLO(J)
      SUMGH=SUMGH+STGHI(J)
      SUMGC=SUMGC+STGCO(J)
2109  CONTINUE
      WRITE(27, 101) DATE_AND_TIME
      WRITE(27, 12100)
      WRITE(27, 12102) (NUCLID(L), L=1, NONCLD), TOTAL
12100  FORMAT('0', /30X, 'NUCLIDE RISK/RISK EQUIVALENT SUMMARY' /
      1  /'0*** SELECTED INDIVIDUAL ***' /
      A  'OLIFETIME FATAL CANCER RISK:')
12102  FORMAT(15X, ' NUCLIDES:', 10(2X, A8)/(26X, 10(2X, A8)))

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

IF ( (SEP_RISK_LET_TABLES ) .OR. (ALL_RISK_LET_TABLES) ) THEN
  WRITE(27,10205) (STLOW(L),L-1,NONCLD),SUML
  WRITE(27,10210) (STHIG(L),L-1,NONCLD),SUMH
ENDIF
IF ( ( COMB_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) )
+   WRITE(27,10215) (STCOM(L),L-1,NONCLD),SUMC

```

```

IF(RNWR)WRITE(27,20216) RR
IF(RNWR)WRITE(27,20217) TT
IF(RNWR)WRITE(27,20422) RT
IF(RNWR)WRITE(27,20223) TTQ

```

```

IF(INDPOP.NE.1) GO TO 21000
DO 2150 J-1,40
  STHIG(J)=0.0
  STLOW(J)=0.0
  STCOM(J)=0.0
  STREQ(J)=0.0
  STGRQ(J)=0.0
  STGLO(J)=0.0
  STGHI(J)=0.0
  STGCO(J)=0.0
2150 CONTINUE
  SUML=0.0
  SUMH=0.0
  SUMC=0.0
  SUMGL=0.0
  SUMGH=0.0
  SUMGC=0.0
  SUMRQ=0.0
  SUMGQ=0.0
  DO 2209 J-1,NONCLD
  DO 2200 L-1,NCANC
  DO 2200 K-1,4
    FACNEW=TOTFAC*RISFAC(K)
    FACNE2=TOTFAC*GENFAC(K)
    DO 2200 II=NOL,NOU
    DO 2200 JJ=NRL,NRU
      STHIG(J)=STHIG(J)+RISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW
      STLOW(J)=STLOW(J)+RISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
      STCOM(J)=STCOM(J)+RISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNEW
      > +RISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNEW
      IF(L.GT.1) GO TO 2200
      STREQ(J)=STREQ(J)+REF(NCANC+1,J,K)*EXPP(JJ,II,J,K)*TOTFAC*

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

>  REQFAC(K)
   SUMRQ=SUMRQ+REF(NCANC+1,J,K)*EXPP(JJ,II,J,K)*TOTFAC*
>  REQFAC(K)
   STGRQ(J)=STGRQ(J)+GREF(L,J,K)*EXPP(JJ,II,J,K)*TOTFAC*
>  GENFAC(K)
   SUMGQ=SUMGQ+GREF(L,J,K)*EXPP(JJ,II,J,K)*TOTFAC*
>  GENFAC(K)
   STGHI(J)=STGHI(J)+GRISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2
   STGLO(J)=STGLO(J)+GRISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2
   STGCO(J)=STGCO(J)+GRISK(L,J,K,2)*EXPP(JJ,II,J,K)*FACNE2+
+  GRISK(L,J,K,1)*EXPP(JJ,II,J,K)*FACNE2
2200  CONTINUE
      SUML=SUML+STLOW(J)
      SUMH=SUMH+STHIG(J)
      SUMC=SUMC+STCOM(J)
      SUMGL=SUMGL+STGLO(J)
      SUMGH=SUMGH+STGHI(J)
      SUMGC=SUMGC+STGCO(J)
2209  CONTINUE
      WRITE(27,101)DATE_AND_TIME
      WRITE(27,12400) (NUCLID(L),L=1,NONCLD),TOTAL
12400  FORMAT('0',/
> 30X,'NUCLIDE RISK/RISK EQUIVALENT SUMMARY'//
A '0*** MEAN INDIVIDUAL ***'/
B 'OLIFETIME FATAL CANCER RISK: '/
A 15X,' NUCLIDES:',10(2X,A8)/(26X,10(2X,A8)))

      IF ( ( SEP_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) ) THEN
         WRITE(27,10205) (STLOW(L),L=1,NONCLD),SUML
         WRITE(27,10210) (STHIG(L),L=1,NONCLD),SUMH
      ENDIF
      IF ( ( COMB_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) )
+     WRITE(27,10215) (STCOM(L),L=1,NONCLD),SUMC

      IF(RNWR)WRITE(27,20216) RRM
      IF(RNWR)WRITE(27,20217) TTM
      IF(RNWR)WRITE(27,20422) RTM
      IF(RNWR)WRITE(27,20223) TTQM
      DO 2350 K=1,40
      STLOW(K)=0.0
      STHIG(K)=0.0
      STCOM(K)=0.0
      STGLO(K)=0.0
      STGHI(K)=0.0
      STGCO(K)=0.0

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
12345678901234567890123456789012345678901234567890123456789012
2350      STREQ(K)=0.0
          STGRQ(K)=0.0
          CONTINUE
          SUML=0.0
          SUMH=0.0
          SUMC=0.0
          SUMGL=0.0
          SUMGH=0.0
          SUMGC=0.0
          SUMRQ=0.0
          SUMGQ=0.0
          DO 2409 J=1, NONCLD
          DO 2400 K=1, 4
          TEXPP=0.0
          DO 2403 II=NOL, NOU
          DO 2403 JJ=NRL, NRU
          TEXPP=TEXPP+EXPP(JJ, II, J, K)
2403      CONTINUE
          DO 2400 L=1, NCANC
          STLOW(J)=STLOW(J)+RISK(L, J, K, 1)*TEXPP*RISFAC(K)/AGEX
          STHIG(J)=STHIG(J)+RISK(L, J, K, 2)*TEXPP*RISFAC(K)/AGEX
          STCOM(J)=STCOM(J)+RISK(L, J, K, 1)*TEXPP*RISFAC(K)/AGEX+
> RISK(L, J, K, 2)*TEXPP*RISFAC(K)/AGEX
          IF(L.GT.1) GO TO 2400
          STREQ(J)=STREQ(J)+REF(NCANC+1, J, K)*TEXPP*REQFAC(K)*.001
          SUMRQ=SUMRQ+REF(NCANC+1, J, K)*TEXPP*REQFAC(K)*.001
          STGRQ(J)=STGRQ(J)+GREF(L, J, K)*TEXPP*GENFAC(K)*.001
          SUMGQ=SUMGQ+GREF(L, J, K)*TEXPP*GENFAC(K)*.001
          STGLO(J)=STGLO(J)+GRISK(L, J, K, 1)*TEXPP*GENFAC(K)*REPPER
          STGHI(J)=STGHI(J)+GRISK(L, J, K, 2)*TEXPP*GENFAC(K)*REPPER
          STGCO(J)=STGCO(J)+GRISK(L, J, K, 1)*TEXPP*GENFAC(K)*REPPER
          A +GRISK(L, J, K, 2)*TEXPP*GENFAC(K)*REPPER
2400      CONTINUE
          SUML=SUML+STLOW(J)
          SUMH=SUMH+STHIG(J)
          SUMC=SUMC+STCOM(J)
          SUMGL=SUMGL+STGLO(J)
          SUMGH=SUMGH+STGHI(J)
          SUMGC=SUMGC+STGCO(J)
2409      CONTINUE
          WRITE(27, 101) DATE AND TIME
          WRITE(27, 12700) (NUCLID(L), L=1, NONCLD), TOTAL
12700     FORMAT('0', /
> 29X, 'NUCLIDE RISK/RISK EQUIVALENT SUMMARY' //
          A '0*** COLLECTIVE POPULATION ***' /
          A '0COLLECTIVE FATAL CANCER RISK:' /

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

A 15X, ' NUCLIDES:',10(2X,A8)/(26X,10(2X,A8))

      IF ( (SEP_RISK_LET_TABLES ) .OR. (ALL_RISK_LET_TABLES) ) THEN
        WRITE(27,10705) (STLOW(L),L-1,NONCLD),SUML
        WRITE(27,10710) (STHIG(L),L-1,NONCLD),SUMH
      ENDIF
      IF ( ( COMB_RISK_LET_TABLES ) .OR. ( ALL_RISK_LET_TABLES ) )
+     WRITE(27,10715) (STCOM(L),L-1,NONCLD),SUMC

      IF(RNWR)WRITE(27,20216) RRC
      WRITE(27,20217) TTC
      IF(RNWR)WRITE(27,20522) RTC
      IF(RNWR)WRITE(27,20323) TTQC
21000 RETURN
      END

```

---

```

C
C
C           SUBROUTINE DRTAB
C
C

```

---

```

      SUBROUTINE DRTAB(ARRAY,NORGN,ORGN,TITLE,TITL1,DTABLE,TITLA,NOTE,
A  NUN,RFAC,CREP,NCREP, NUCLID)
C***  THIS ROUTINE OUTPUT THE APPROPRIATE TABLES.

      CHARACTER*8  ORGN,NUCLID,ORC,SUM,CREP,RADON

      CHARACTER*80 TITLE,NOTE
      CHARACTER*40 TITL1
      CHARACTER*8  TITLA
      CHARACTER*32 NUN
      CHARACTER*16 PATH

      COMMON/COMRN/WLRN(20,20),RRISK,RREF(2),RYRLL,NOREP,NRREP,NFREP

      COMMON/HEAD/ORC

      INTEGER DTABLE

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

3 4 5 6 7  
0123456789012345678901234567890123456789012

0,40,4), ORGN(1), PATH(7), DTABLE(7), WLSUM(4),  
 0), NUCLID(40), CREP(20)

CLD, PSIZE(40), GIABS (4,40), INDPOP

K(40,7), SUMY(40,3), PERX(40), PERY(40), TVAL(40),

```

22  '/
TION      ', 'INHALATION      ', 'AIR IMMERSION  '
D SURFACE ', 'INTERNAL        ', 'EXTERNAL      '
  '/
  
```

NUCLIDES FOR EACH ORGAN AND PATHWAY  
 INTERNAL, 6 IS EXTERNAL, AND 7 IS ALL PATHWAYS

, I)+ARRAY(K, J, I)  
 (K, 5)-SUMX(K, 5)+ARRAY(K, J, I)  
 (K, 6)-SUMX(K, 6)+ARRAY(K, J, I)  
 , 7)+ARRAY(K, J, I)

.0) GO TO 80

TITLE, TITL1, TITLA  
 PATH(I)

ORC, (ORGN(K), K-1, NORGN)

.0.0) PERX(K)=ARRAY(K, J, I)/SUMX(K, I)\*100.  
 NUCLID(J), (ARRAY(K, J, I), K-1, NORGN)  
 PATH(I), (PERX(K), K-1, NORGN)

0 1 2  
12345678901234567890123

DIMENSION ARRAY(2  
 + RFAC(20)

COMMON/COMNU/NONC

COMMON/COMUS/SUM  
 > FACO(20,4)

```

DATA RADON/'RN-22
DATA PATH/'INGEST
+          ', 'GROUND
+          ', '
  
```

DATA SUM/'TOTAL  
 C\*\*\* SUM OVER ALL NU  
 C\*\*\* PATHWAY 5 IS II

```

DO 10 I=1,7
DO 10 K=1,NORGN
10 SUMX(K,I)=0.0
DO 30 I=1,4
DO 20 K=1,NORGN
DO 20 J=1,NONCLD
SUMX(K,I)=SUMX(K
IF (I.LE.2) SUMX
IF (I.GE.3) SUMX
SUMX(K,7)=SUMX(K
  
```

20 CONTINUE  
 30 CONTINUE

C\*\*\* TABLE 1  
 IF (DTABLE(1).EQ  
 DO 70 I=1,4  
 WRITE(27,10000)  
 WRITE(27,10200)

```

WRITE(27,10300)
WRITE(27,10400)
DO 60 J=1,NONCLD
DO 40 K=1,NORGN
PERX(K)=0.0
  
```

```

40 IF (SUMX(K,I).NE
WRITE(27,10500)
WRITE(27,10600)
DO 50 K=1,NORGN
II=5
IF (I.GT.2) II=6
PERX(K)=0.0
  
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      IF (SUMX(K,II).NE.0.0) PERX(K)=ARRAY(K,J,I)/SUMX(K,II)*100.
      PERY(K)=0.0
50  IF (SUMX(K,7).NE.0.0) PERY(K)=ARRAY(K,J,I)/SUMX(K,7)*100.
      WRITE(27,10600) PATH(II), (PERX(K),K=1,NORGN)
      WRITE(27,10700) (PERY(K),K=1,NORGN)
      WRITE(27,10100)
60  CONTINUE
      WRITE(27,11000) (SUMX(K,I),K=1,NORGN)
      IF(I.NE.2) GO TO 70
      DO 75 J=1, NONCLD
      IF(NUCLID(J).NE.RADON) GO TO 75
      WRITE(27,20100) NOTE
      IF(RFAC(1).EQ.0.0) GO TO 75
      WRITE(27,10300) ORC, (CREP(JJ),JJ=1,NCREP)
      WRITE(27,20400) NUN, (RFAC(JJ),JJ=1,NCREP)
75  CONTINUE
70  CONTINUE
C***  TABLE 2
80  IF (DTABLE(2).EQ.0) GO TO 120
      DO 110 I=1,2
      WRITE(27,10000) TITLE,TITL1,TITLA
      II=I+4
      WRITE(27,10200) PATH(II)
      WRITE(27,10300) ORC, (ORGN(K),K=1,NORGN)
      WRITE(27,10400)
      DO 100 J=1, NONCLD
      DO 90 K=1, NORGN
      L=1
      IF (I.EQ.2) L=3
      TVAL(K)=ARRAY(K,J,L)+ARRAY(K,J,L+1)
      PERY(K)=0.0
      IF (SUMX(K,II).NE.0.0) PERY(K)=TVAL(K)/SUMX(K,II)*100.
      PERX(K)=0.0
90  IF (SUMX(K,7).NE.0.0) PERX(K)=TVAL(K)/SUMX(K,7)*100.
      WRITE(27,10500) NUCLID(J), (TVAL(K),K=1,NORGN)
      WRITE(27,10600) PATH(II), (PERY(K),K=1,NORGN)
      WRITE(27,10700) (PERX(K),K=1,NORGN)
100 CONTINUE
      WRITE(27,11000) (SUMX(K,II),K=1,NORGN)
      WRITE(27,10100)
      IF(I.NE.1) GO TO 110
      DO 115 J=1, NONCLD
      IF(NUCLID(J).NE.RADON) GO TO 115
      WRITE(27,20100) NOTE
      IF(RFAC(1).EQ.0.0) GO TO 115
      WRITE(27,10300) ORC, (CREP(JJ),JJ=1,NCREP)

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

        WRITE(27,20400)NUN,(RFAC(JJ),JJ-1,NCREP)
115     CONTINUE
110     CONTINUE
C***   TABLE 3
120    IF (DTABLE(3).EQ.0) GO TO 170
        WRITE(27,10000) TITLE,TITL1,TITLA
        WRITE(27,10900)
        WRITE(27,10300) ORC,(ORGN(K),K-1,NORGN)
        WRITE(27,10400)
        DO 160 J-1,NONCLD
        DO 130 K-1,NORGN
130    TVAL(K)=0.0
        DO 150 K-1,NORGN
        DO 140 I-1,4
        TVAL(K)=ARRAY(K,J,I)+TVAL(K)
140    CONTINUE
        PERX(K)=0.0
        IF (SUMX(K,7).NE.0.0) PERX(K)=TVAL(K)/SUMX(K,7)*100.
150    CONTINUE
        WRITE(27,10500) NUCLID(J),(TVAL(K),K-1,NORGN)
        WRITE(27,10700) (PERX(K),K-1,NORGN)
        WRITE(27,10100)
160    CONTINUE
        WRITE(27,11000) (SUMX(K,7),K-1,NORGN)
        DO 165 J-1,NONCLD
        IF(NUCLID(J).NE.RADON) GO TO 165
        WRITE(27,20100) NOTE
        IF(RFAC(1).EQ.0.0) GO TO 165
        WRITE(27,10300)ORC,(CREP(K),K-1,NCREP)
        WRITE(27,20400)NUN,(RFAC(JJ),JJ-1,NCREP)
165    CONTINUE
C***   TABLE 4
170    IF (DTABLE(4).EQ.0) GO TO 260
        DO 250 K-1,NORGN
        WRITE(27,10000) TITLE,TITL1,TITLA
        WRITE(27,11100) ORC,ORGN(K)
        WRITE(27,11200) (NUCLID(J),J-1,NONCLD),SUM
        WRITE(27,11300)
        DO 180 I-1,3
        DO 180 J-1,NONCLD
180    SUMY(J,I)=0.0
        DO 200 I-1,4
        DO 190 J-1,NONCLD
        IF (I.LE.2) SUMY(J,1)=SUMY(J,1)+ARRAY(K,J,I)
        IF (I.GE.3) SUMY(J,2)=SUMY(J,2)+ARRAY(K,J,I)
        SUMY(J,3)=SUMY(J,3)+ARRAY(K,J,I)

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

190 CONTINUE
200 CONTINUE
    DO 220 I=1,4
    WRITE(27,11400) PATH(I), (ARRAY(K,J,I), J=1, NONCLD),
    > SUMX(K,I)
    DO 210 J=1, NONCLD
    II=1
    IF (I.GT.2) II=2
    PERX(J)=0.0
    IF (SUMY(J,II).NE.0.0) PERX(J)=ARRAY(K,J,I)/SUMY(J,II)*100.
    PERY(J)=0.0
    IF (SUMY(J,3).NE.0.0) PERY(J)=ARRAY(K,J,I)/SUMY(J,3)*100.
210 CONTINUE
    WPP=0.0
    IF (SUMX(K,4+II).NE.0.0) WPP=SUMX(K,I)/SUMX(K,4+II)*100.
    WP=0.0
    IF (SUMX(K,7).NE.0.0) WP=SUMX(K,I)/SUMX(K,7)*100.
    WRITE(27,11500) PATH(4+II), (PERX(J), J=1, NONCLD), WPP
    WRITE(27,11600) (PERY(J), J=1, NONCLD), WP
    WRITE(27,10100)
220 CONTINUE
    DO 240 I=1,2
    WRITE(27,11400) PATH(I+4), (SUMY(J,I), J=1, NONCLD), SUMX(K,4+I)
    DO 230 J=1, NONCLD
    PERX(J)=0.0
230 IF (SUMY(J,3).NE.0.0) PERX(J)=SUMY(J,I)/SUMY(J,3)*100.
    WP=0.0
    IF (SUMX(K,7).NE.0.0) WP=SUMX(K,4+I)/SUMX(K,7)*100.
    WRITE(27,11600) (PERX(J), J=1, NONCLD), WP
    WRITE(27,10100)
240 CONTINUE
    WRITE(27,11700) (SUMY(J,3), J=1, NONCLD), SUMX(K,7)
C***** THIS SECTION OF CODE WAS FIXED TO CONFORM WITH FORTRAN 77
C***** STANDARDS. IE TO STOP IT FROM JUMPING INTO THE DO LOOP 6/88
C***** J. MCGUE

    DO 245 J = 1, NONCLD

    IF ( NUCLID(J) .EQ. RADON ) THEN
        JJ=NCREP
        IF ( ORGN(K) .EQ. SUM) THEN
            WRITE(27,20100) NOTE
            IF ( RFAC(1) .EQ. 0.0) GO TO 250
            WRITE(27,20400)NUN, RFAC(JJ)
        ELSE
            DO 255 JJ=1, NCREP

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
12345678901234567890123456789012345678901234567890123456789012

```

                IF ( ORGN(K) .EQ. CREP(JJ) ) THEN
                    WRITE(27,20100) NOTE
                    IF(RFAC(1).EQ.0.0) GO TO 250
                    WRITE(27,20400)NUN, RFAC(JJ)
                ENDIF
255             CONTINUE
                ENDIF
            ENDIF

245     CONTINUE

250 CONTINUE
C***     TABLE 5
260 IF (DTABLE(5).EQ.0) GO TO 350
        DO 340 J=1,NONCLD
            WRITE(27,10000) TITLE,TITL1,TITLA
            WRITE(27,11800) NUCLID(J)
            DO 270 I=1,3
                DO 270 K=1,NORGN
270     SUMY(K,I)=0.0
                DO 280 I=1,4
                    DO 280 K=1,NORGN
                        IF (I.LE.2) SUMY(K,1)=SUMY(K,1)+ARRAY(K,J,I)
                        IF (I.GE.3) SUMY(K,2)=SUMY(K,2)+ARRAY(K,J,I)
                        SUMY(K,3)=SUMY(K,3)+ARRAY(K,J,I)
280     CONTINUE
                    WRITE(27,11900) ORC,(ORGN(K),K=1,NORGN)
                    WRITE(27,11300)
                    DO 300 I=1,4
                        WRITE(27,11400) PATH(I),(ARRAY(K,J,I),K=1,NORGN)
                        II=1
                        IF (I.GE.3) II=2
                        DO 290 K=1,NORGN
                            PERX(K)=0.0
                            IF (SUMY(K,II).NE.0.0) PERX(K)=ARRAY(K,J,I)/SUMY(K,II)*100.
                            PERY(K)=0.0
                            IF (SUMY(K,3).NE.0.0) PERY(K)=ARRAY(K,J,I)/SUMY(K,3)*100.
290     CONTINUE
                        WRITE(27,11500) PATH(II+4),(PERX(K),K=1,NORGN)
                        WRITE(27,11600) (PERY(K),K=1,NORGN)
                        WRITE(27,10100)
300     CONTINUE
                    DO 320 I=1,2
                        WRITE(27,11400) PATH(I+4),(SUMY(K,I),K=1,NORGN)
                        DO 310 K=1,NORGN
                            PERX(K)=0.0

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      IF (SUMY(K,3).NE.0.0) PERX(K)=SUMY(K,I)/SUMY(K,3)*100.
310 CONTINUE
      WRITE(27,11600) (PERX(K),K-1,NORGN)
      WRITE(27,10100)
320 CONTINUE
      WRITE(27,11700) (SUMY(K,3),K-1,NORGN)
      DO 330 K-1,NORGN
      PERX(K)=0.0
330 IF (SUMX(K,7).NE.0.0) PERX(K)=SUMY(K,3)/SUMX(K,7)*100.
      WRITE(27,10700) (PERX(K),K-1,NORGN)
      L=1
      WRITE(27,20100) NOTE
      IF(RFAC(1).EQ.0.0) GO TO 340
      WRITE(27,10300)ORC, (CREP(K),K-1,NCREP)
      WRITE(27,20400)NUN, (RFAC(K),K-1,NCREP)
335 CONTINUE
340 CONTINUE
C*** TABLE 6
350 IF (DTABLE(6).EQ.0) GO TO 500
      DO 360 I=1,7
      TVAL(I)=0.0
      DO 360 J=1,NONCLD
      SUMX(J,I)=0.0
360 CONTINUE
390 DO 400 I=1,4
      DO 400 J=1,NONCLD
      SUMX(J,I)=ARRAY(NORGN,J,I)
      IF(I.GE.3) GO TO 401
      SUMX(J,5)=SUMX(J,5)+ARRAY(NORGN,J,I)
      TVAL(5)=TVAL(5)+ARRAY(NORGN,J,I)
      GO TO 402
401 SUMX(J,6)=SUMX(J,6)+ARRAY(NORGN,J,I)
      TVAL(6)=TVAL(6)+ARRAY(NORGN,J,I)
402 SUMX(J,7)=SUMX(J,7)+ARRAY(NORGN,J,I)
      TVAL(I)=TVAL(I)+ARRAY(NORGN,J,I)
      TVAL(7)=TVAL(7)+ARRAY(NORGN,J,I)
400 CONTINUE
410 CONTINUE
      WRITE(27,10000) TITLE,TITL1,TITLA
      WRITE(27,12100) ORC
      WRITE(27,11200) (NUCLID(J),J=1,NONCLD),SUM
      DO 430 I=1,4
      WRITE(27,11400) PATH(I),(SUMX(J,I),J=1,NONCLD),TVAL(I)
      II=5
      IF (I.GE.3) II=6
      DO 420 J=1,NONCLD

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

PERX(J)=0.0
IF (SUMX(J,II).NE.0.0) PERX(J)=SUMX(J,I)/SUMX(J,II)*100.
PERY(J)=0.0
IF (SUMX(J,7).NE.0.0) PERY(J)=SUMX(J,I)/SUMX(J,7)*100.
420 CONTINUE
WP=0.0
IF (TVAL(II).NE.0.0) WP=TVAL(I)/TVAL(II)*100.
WPP=0.0
IF (TVAL(7).NE.0.0) WPP=TVAL(I)/TVAL(7)*100.
WRITE(27,11500) PATH(II), (PERX(J),J-1,NONCLD),WP
WRITE(27,11600) (PERY(J),J-1,NONCLD),WPP
WRITE(27,10100)
430 CONTINUE
DO 450 I=5,6
WRITE(27,11400) PATH(I), (SUMX(J,I),J-1,NONCLD),TVAL(I)
DO 440 J=1,NONCLD
PERX(J)=0.0
IF (SUMX(J,7).NE.0.0) PERX(J)=SUMX(J,I)/SUMX(J,7)*100.
440 CONTINUE
WRITE(27,11600) (PERX(J),J-1,NONCLD)
WRITE(27,10100)
450 CONTINUE
WRITE(27,11700) (SUMX(J,7),J-1,NONCLD),TVAL(7)
DO 460 J=1,NONCLD
PERX(J)=0.0
IF (TVAL(7).NE.0.0) PERX(J)=SUMX(J,7)/TVAL(7)*100.
460 CONTINUE
WRITE(27,12000) (PERX(J),J-1,NONCLD)
DO 465 J=1,NONCLD
IF(NUCLID(J).NE.RADON) GO TO 465
WRITE(27,20100) NOTE
IF(RFAC(1).EQ.0.0) GO TO 470
WRITE(27,20400)NUN, RFAC
470 CONTINUE
465 CONTINUE
C*** TABLE 7
500 IF(DTABLE(7).EQ.0) RETURN
DO 510 I=1,7
DO 510 J=1,NORGN
SUMX(J,I)=0.0
510 CONTINUE
DO 550 I=1,4
DO 550 J=1,NORGN
DO 550 K=1,NONCLD
SUMX(J,I)=SUMX(J,I)+ARRAY(J,K,I)
IF(I.GE.3) GO TO 501

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0	1	2	3	4	5	6	7

```

SUMX(J, 5)=SUMX(J, 5)+ARRAY(J, K, I)
GO TO 502
501 SUMX(J, 6)=SUMX(J, 6)+ARRAY(J, K, I)
502 SUMX(J, 7)=SUMX(J, 7)+ARRAY(J, K, I)
550 CONTINUE
WRITE(27, 10000) TITLE, TITL1, TITLA
WRITE(27, 12200)
WRITE(27, 10300) ORC, (ORGN(K), K=1, NORGN)
WRITE(27, 11300)
DO 575 I=1, 4
WRITE(27, 11400) PATH(I), (SUMX(J, I), J=1, NORGN)
II=5
IF(I.GE.3) II=6
DO 560 J=1, NORGN
PERX(J)=0.0
IF(SUMX(J, II).NE.0.0) PERX(J)=SUMX(J, I)/SUMX(J, II)*100.
PERY(J)=0.0
IF(SUMX(J, 7).NE.0.0) PERY(J)=SUMX(J, I)/SUMX(J, 7)*100.
560 CONTINUE
WRITE(27, 11500) PATH(II), (PERX(J), J=1, NORGN)
WRITE(27, 11600) (PERY(J), J=1, NORGN)
WRITE(27, 10100)
575 CONTINUE
DO 590 I=5, 6
WRITE(27, 11400) PATH(I), (SUMX(J, I), J=1, NORGN)
DO 580 J=1, NORGN
PERX(J)=0.0
IF(SUMX(J, 7).NE.0.0) PERX(J)=SUMX(J, I)/SUMX(J, 7)*100.
580 CONTINUE
WRITE(27, 11600) (PERX(J), J=1, NORGN)
WRITE(27, 10100)
590 CONTINUE
WRITE(27, 11700) (SUMX(J, 7), J=1, NORGN)
DO 600 J=1, NONCLD
IF(NUCLID(J).NE.RADON) GO TO 600
WRITE(27, 20100) NOTE
IF(RFAC(1).EQ.0.0) GO TO 600
DO 610 L=1, NCREP
610 WLSUM(L)=0.0
DO 612 JJ=1, 1
DO 612 L=1, NCREP
WLSUM(L)=WLSUM(L)+RFAC(L)
612 CONTINUE
WRITE(27, 10300) ORC, (CREP(L), L=1, NCREP)
WRITE(27, 20400) NUN, (WLSUM(L), L=1, NCREP)
600 CONTINUE

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
12345678901234567890123456789012345678901234567890123456789012

```

RETURN
10000 FORMAT(1H1,20X,A80/21X,A40/21X,A8)
10100 FORMAT(1X)
10200 FORMAT('0*** FOR PATHWAY:',A16//)
10300 FORMAT('0',A8,':',22X,10(2X,A8)/(33X,10(2X,A8)))
10400 FORMAT('ONUCLIDES'//)
10500 FORMAT(1X,A8,23X,10(1PE10.2)/(33X,10(1PE10.2) ) )
10600 FORMAT(' % OF TOTAL',1X,A16,4X,10(1PE10.2)/(33X,10(1PE10.2) ) )
10700 FORMAT(' % OF TOTAL',21X,10(1PE10.2)/ (33X,10(1PE10.2) ) )
10800 FORMAT('OTOTAL',21X,10(1PE10.2)/(33X,10(1PE10.2) ) )
10900 FORMAT('0*** FOR ALL PATHWAYS:'//)
11000 FORMAT(' TOTAL',26X,10(1PE10.2)/(33X,10(1PE10.2)))
11100 FORMAT('0***FOR',A8,':',A8//)
11200 FORMAT(' NUCLIDES',23X,10(2X,A8)/(33X,10(2X,A8)))
11300 FORMAT(' PATHWAYS'//)
11400 FORMAT(1X,A16,15X,10(1PE10.2)/(33X,10(1PE10.2) ) )
11500 FORMAT(' % OF ',1X,A16,9X,10(1PE10.2)/ (33X,10(1PE10.2)))
11600 FORMAT(' % OF ALL PATHWAYS',14X,10(1PE10.2)/ (33X,10(1PE10.2)))
11700 FORMAT(' TOTAL OVER ALL PATHWAYS',8X,10(1PE10.2)/(33X,
+ 10(1PE10.2)))
11800 FORMAT('OFOR NUCLIDE:',A8)
11900 FORMAT(1X,A8,23X,10(2X,A8)/(33X,10(2X,A8)))
12000 FORMAT(11H % OF TOTAL,21X,10(1PE10.2)/(33X,10(1PE10.2) ) )
12100 FORMAT('0***SUMMED OVER ALL',A8)
12200 FORMAT('0***SUMMED OVER ALL NUCLIDES')
20100 FORMAT('0',/////10X,A80)
20400 FORMAT(1X,A32,10(1PE10.2) )
END

```

```

C-----
C
C               SUBROUTINE LOCTAB
C-----
C

```

```

SUBROUTINE LOCTAB(IT,RN,PT,OG,FACD,ARRAY,TITL1,TITLA,
< ORGN,NORGN,NDIM,NO, NUCLID,DATE_AND_TIME)

```

```

C CORRECTED FOR ORGAN/CANCER "SUM" OPTION.  CBN 11/05/82

CHARACTER*8 RN,OG,SUM,NUCLID,ORGN,WLOPT(2),PUL,
+ TBEQ,OGI

CHARACTER*80 TITLE
CHARACTER*8 TITLA

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

CHARACTER\*40 TITL1  
CHARACTER\*16 TPATH  
CHARACTER\*36 DATE\_AND\_TIME

COMMON/COMRN/WLRN(20,20),RRISK,RREF(2),RYRLL,NOREP,NRREP,NCREP  
COMMON/COMCA/DUM(6428),AGEX  
INTEGER PT

DIMENSION ARRAY(NDIM,40,4), NUCLID(40),  
+ ORGN(1),FACD(1),TPATH(7),IDIR(16),NL(40),SAVG(20)

COMMON/COMUS/OUTPUT(40,40),HOLDC(40),HOLDR(40)

DATA TPATH/'INGESTION            ','INHALATION            ',  
+            'AIR IMMERSION       ','GROUND SURFACE       ',  
+            'INTERNAL               ','EXTERNAL               ',  
+            'ALL                        '/

DATA SUM/'SUM            '/,WLOPT/'WORKLEVL','WLSUM       '/,  
+        PUL/'PULMINARY'/,TBEQ/'BODY EQ.'/'

DATA IDIR/1,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2/

COMMON/COMNU/NONCLD,PSIZE(40),GIABS (4,40),INDPOP

COMMON/COMEX/EXPP(20,20,40,4),POP(20,20),POPFAC,TOTFAC, NOL,NOU,  
> NRL,NRU, IDIST(20), ILOC, JLOC

```
10000  FORMAT('0',21X,A40/21X,A8)
        DO 10 K=1,40
          HOLDC(K)=0.0
          HOLDR(K)=0.0
          DO 10 L=1,40
10      OUTPUT(L,K)=0.0
        INB=1
        INE = NONCLD
        DO 11 J=1,NONCLD
11     NL(J)=J
          TSUM=0.0
          DO 12 J=1,2
            IF(RN.EQ.WLOPT(J)) GO TO 1000
12     CONTINUE
14     CONTINUE
          IF(RN.EQ.SUM.OR.RN.EQ.WLOPT(2)) GO TO 80
          INE=0
          DO 50 J=1,NONCLD
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

        IF(RN.NE.NUCLID(J)) GO TO 50
        INE=INE+1
        NL(INE)=J
50      CONTINUE
        IF(INE.NE.0) GO TO 80
        WRITE(27,10500) RN
10500   FORMAT(' RADIONUCLIDE ',A8,
A      ' IS NOT IN LIST. TABLE WILL BE SKIPPED.')
        RETURN
80      CONTINUE
        IF(OG.EQ.SUM) GO TO 175
        DO 150 J=1,NORGN
        IF(OG.EQ.ORGJ(J)) GO TO 170
150     CONTINUE
        WRITE(27,10600) OG
10600   FORMAT(' ORGAN ',A8,' IS NOT IN LIST.',
A      ' TABLE WILL BE SKIPPED.')
        RETURN
170     IO=J
        GO TO 180
175     IO=NORGN
180     OGI=ORGJ(IO)
        IPT=PT
        IPB=IPT
        IPE=IPT
        IF(IPT.LE.4) GO TO 184
177     CONTINUE
        IPT=IPT-4
        GO TO (181,182,183),IPT
181     IPB=1
        IPE=2
        GO TO 184
182     IPB=3
        IPE=4
        GO TO 184
183     IPB=1
        IPE=4
184     CONTINUE
        DO 300 IPT=IPB,IPE
        DO 300 INL=INB,INE
        IN=NL(INL)
        DO 300 II=NOL,NOU
        DO 300 JJ=NRL,NRU
        OUTPUT(JJ,II)=OUTPUT(JJ,II)+ARRAY(IO,IN,IPT)*FACD(IPT)*
>      EXPP(JJ,II,IN,IPT)
300     CONTINUE

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0	1	2	3	4	5	6	7
<u>123456789012345678901234567890123456789012345678901234567890123456789012</u>							
305	IF(IT.EQ.3) GO TO 500	DO 400 JJ=NRL,NRU	SOUT=0.	SPOP=0.	SAVG(JJ)=0.	DO 390 II=NOL,NOU	IF(POP(II,JJ).NE.0.) GO TO 360
	OUTPUT(JJ,II)=0.	GO TO 390	360	IF(IT.NE.1) GO TO 370	C	IT=1 (INDIVIDUAL)	SOUT=SOUT+OUTPUT(JJ,II)/POP(II,JJ)
	SPOP=SPOP+1.	GO TO 380	C	IT=2 (MEAN INDIVIDUAL)	370	SOUT=SOUT+OUTPUT(JJ,II)	SPOP=SPOP+POP(II,JJ)
	380	OUTPUT(JJ,II)=OUTPUT(JJ,II)/POP(II,JJ)	390	CONTINUE	IF(SPOP.NE.0.) SAVG(JJ)=SOUT/SPOP	400	CONTINUE
	100	WRITE(27,100)DATE_AND_TIME	FORMAT('1',T10,'DATE',2X,A)	WRITE(27,10000) TITL1,TITLA	WRITE(27,10100) RN,OGI,TPATH(PT)	DO 600 JJ=NRL,NRU	WRITE(27,10200) IDIST(JJ),(OUTPUT(JJ,DIR(II)),II=1,8)
600	CONTINUE	WRITE(27,10125)	DO 605 JJ=NRL,NRU	WRITE(27,10205) IDIST(JJ),(OUTPUT(JJ,DIR(II)),II=9,16),SAVG(JJ)	605	CONTINUE	RETURN
500	WRITE(27,100)DATE_AND_TIME	WRITE(27,10000) TITL1,TITLA	DO 680 II=1,16	DO 680 JJ=NRL,NRU	HOLDC(II)=OUTPUT(JJ,II)+HOLDC(II)	HOLDR(JJ)=OUTPUT(JJ,II)+HOLDR(JJ)	680
	TSUM=OUTPUT(JJ,II)+TSUM	WRITE(27,10100) RN,OGI,TPATH(PT)	DO 700 JJ=NRL,NRU	WRITE(27,10200) IDIST(JJ),(OUTPUT(JJ,DIR(II)),II=1,8)	WRITE(27,10250) (HOLDC(DIR(II)),II=1,8)	WRITE(27,10175)	DO 710 JJ=NRL,NRU

DARTAB (DARTAB2.FOR) Program File  
(continued)

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

710 WRITE(27,10205) IDIST(JJ), (OUTPUT(JJ, IDIR(II)), II-9, 16),  
+ HOLDR(JJ)  
WRITE(27,10250) (HOLDC(IDIR(II)), II-9, 16), TSUM  
RETURN  
1000 IF(NDIM.NE.20) RETURN  
IF(OG.EQ.SUM) GO TO 1001  
IF(OG.EQ.PUL) GO TO 1001  
IF(OG.EQ.TBEQ) GO TO 1001  
IF(J.EQ.1) GO TO 305  
GO TO 14  
1001 CONTINUE  
IF(NO.NE.1) GO TO 1200  
1100 DO 1150 II=NOL, NOU  
DO 1150 JJ=NRL, NRU  
OUTPUT(JJ, II)=WLRN(JJ, II)  
1150 CONTINUE  
OG=SUM  
PT=7  
GO TO 305  
1200 IF(NO.NE.3) GO TO 1300  
FAC=1.  
IF(IT.EQ.3) FAC=1./AGEX  
DO 1250 II=NOL, NOU  
DO 1250 JJ=NRL, NRU  
OUTPUT(JJ, II)=RRISK\*WLRN(JJ, II)\*FAC  
1250 CONTINUE  
IF(J.EQ.1) GO TO 305  
GO TO 14  
1300 IF(OG.NE.PUL) GO TO 1400  
FAC=1.  
IF(IT.EQ.3) FAC=.001  
DO 1350 II=NOL, NOU  
DO 1350 JJ=NRL, NRU  
OUTPUT(JJ, II)=RREF(1)\*WLRN(JJ, II)\*FAC  
1350 CONTINUE  
IF(J.EQ.1) GO TO 500  
GO TO 14  
1400 IF(OG.NE.TBEQ) RETURN  
FAC=1.  
IF(IT.EQ.3) FAC=.001  
DO 1450 II=NOL, NOU  
DO 1450 JJ=NRL, NRU  
OUTPUT(JJ, II)=RREF(2)\*WLRN(JJ, II)\*FAC  
1450 CONTINUE  
IF(J.EQ.1) GO TO 305  
GO TO 14

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012
10100  FORMAT(' FOR RADIONUCLIDE : ',A8/
> ' AND ORGAN/CANCER : ',A8/
> ' AND PATHWAY : ',A16/
> ' DIRECTIONS:',3X,'N',6X,'NNE',5X,'NE',6X,
> 'ENE',6X,'E',6X,'ESE',5X,'SE',6X,'SSE'/
> ' DISTANCE'/' (METERS):')
10125  FORMAT(////15X,'S',6X,'SSW',5X,'SW',6X,'WSW',6X,'W',
> 6X,'WNW',5X,'NW',6X,'NNW',5X,'AVG'/
> 1H,' DISTANCE'/' (METERS):')
10175  FORMAT(////,15X,'S',6X,'SSW',5X,'SW',6X,'WSW',6X,'W'
> ,6X,'WNW',5X,'NW',6X,'NNW',5X,'SUM'/' DISTANCE'/'
> ' (METERS):')
10200  FORMAT(I7,3X,8(1PE7.1,1X) )
10205  FORMAT(I7,3X,9(1PE7.1,1X) )
10250  FORMAT('0 SUM',5X,9(1PE7.1,1X) )
      END

```

---

```

C
C          SUBROUTINE WRITE SYNOPSIS REPORT
C
C NEW REPORT ADDED 8/88 BY J. MCGUE.  THIS REPORT IS A SUMMARY OF
C THE INPUT USED AND THE RESULTS OF THE RUN.
C
C

```

---

```

C
C THE FOLLOWING ROUTINES USE THE SUBROUTINE OUTPUT_STRING TO WRITE
C ALL THE INFORMATION ON THE SYNOPSIS REPORT.  THIS WAS DONE TO AID
C IN CHANGES MADE TO THE REPORT.  ALL OUTPUT IS READ INTO THE CHARAC
C STRING LINES BY INTERNAL WRITES, LINES IS THEN PASSED TO
C OUTPUT_STRING.  OUTPUT_STRING KEEPS TRACK OF THE
C LINE NUMBERS FOR PAGING ETC.  SEE THE SUBROUTINE FOR DOCUMENTATION.
C EVERYTHING ELSE IS SELF EXPLANATORY.
C ALL WRITES TO FILE 13 ARE FOR THE SAS DATA BASE FILE 13 HOLDS ALL
C EXTRANEIOUS INFORMATION.  THE SO CALLED 'USELESS' STUFF.
C ALL WRITES TO FILE 14 ARE FOR THE SAS DATA BASE FILE 14, HOLDS ALL
C INFORMATION THAT WILL BE USED IN REPORTS.  THE 'GOODSTUF'.
C

```

---

```

      SUBROUTINE WRITE_SYNOPSIS_REPORT( INDPOP, ORGN, NORGN,
+                                     FOOD_ARRAY_INFO,
+                                     NUMBER_FILES, ANG)
C

```

---

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

C     VARIABLES PASSED  
C

INTEGER        INDPOP, NORGN  
INTEGER\*2      NUMBER\_FILES  
INTEGER\*4      ANG  
CHARACTER\*8    ORGN(20)  
CHARACTER\*36   DATE\_AND\_TIME  
COMMON / HEADERINFO / DATE\_AND\_TIME

CHARACTER\*80   FOOD\_ARRAY\_INFO

C     GLOBAL VARIABLES  
C

CHARACTER\*80   FILES\_USED(10), NAME\_OF\_PERSON, PHONE\_NUMBER

COMMON / GENERIC\_INFO / FILES\_USED, NAME\_OF\_PERSON, PHONE\_NUMBER

INTEGER\*2      NUMBER\_STACKS, SOURCE\_TYPE, PLUME\_RISE\_TYPE  
REAL            HEIGHT(6), AREA(6), AREA\_DIAMETER(6),  
+                STACK\_DIAMETER(6), BOUYANCY(6), MOMENTUM(6), ENTERED(7)

COMMON / EMMIS\_INFO / HEIGHT, AREA, AREA\_DIAMETER,  
+                STACK\_DIAMETER, BOUYANCY, MOMENTUM,  
+                ENTERED,  
+                NUMBER\_STACKS, SOURCE\_TYPE, PLUME\_RISE\_TYPE

INTEGER\*2      NUMBER\_NUCS, DECAY\_CHAIN\_FLAG (36), DAUGHTERS  
LOGICAL        RN\_RUN  
REAL            AMAD(36), RELEASE\_RATE(36,6), ALAMSUR(36)  
COMMON / NUC\_INFO / AMAD, RELEASE\_RATE, NUMBER\_NUCS,  
+                DAUGHTERS, DECAY\_CHAIN\_FLAG, ALAMSUR, RN\_RUN

COMMON/COMEX/EXPP(20,20,40,4), POP(20,20), POPFAC, TOTFAC, NOL, NOU,  
+ NRL, NRU, IDIST(20), ILOC, JLOC

INTEGER\*2      PAGE\_NO, LINE\_COUNT, NUM\_TITLE\_LINES  
COMMON / COUNTERS / PAGE\_NO, LINE\_COUNT, NUM\_TITLE\_LINES

DARTAB (DARTAB2.FOR) Program File  
(continued)

```

0          1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890123456789012

```

```

CHARACTER*90 SAVED_TITLE_LINES (20)
CHARACTER*26 ID_CODE
COMMON / TITLES / SAVED_TITLE_LINES, ID_CODE

```

```

C   LOCAL VARIABLES
C                     

```

```

INTEGER*2    I, REPORT_NUMBER
CHARACTER*1  STACK_NUMBERS (6)
CHARACTER*8  STARFILE, POPFILE
CHARACTER*90 LINES (20)

```

```

C   INITIALIZATION
C                     

```

```

DATA STACK_NUMBERS / '1', '2', '3', '4', '5', '6' /
DATA STARFILE / ' ', POPFILE / ' '

```

```

C**  BEGIN

```

```

1000 FORMAT ( '1' )
1010 FORMAT ( 1X, A )

```

```

C***** GET THE NAMES OF THE FILES USED ...ID CODE, WIND FILE, *****
C***** POPULATION FILE, AND FOOD FILE NAMES *****

```

```

      ID_CODE = '
      CALL GET_FILE_NAMES (ID_CODE, STARFILE, POPFILE,
+                           FILES_USED, NUMBER_FILES )

```

```

      WRITE(13,1005) RN_RUN
      WRITE(14,1005) RN_RUN
1005 FORMAT (I1,10X,'RN_FLAG')

```

```

C***** WRITE FILE NAMES TO SAS FILES *****

```

```

      WRITE (14,1006) STARFILE
      WRITE (13,1006) STARFILE
      WRITE (14,1007) POPFILE

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

WRITE (13,1007) POPFILE

1006 FORMAT (A,10X,'Star')

1007 FORMAT (A,10X,'Pop')

C\*\*\*\*\* WRITE TITLE PAGE & PERSON TO GIVE TO AND PHONE NUMBER \*\*\*\*\*

WRITE (12,1000)

WRITE (12,1010) NAME\_OF\_PERSON

WRITE (12,1010) PHONE\_NUMBER

C\*\*\*\*\* WRITE HEADING \*\*\*\*\*

WRITE (12,1000)

WRITE (12,1020)

1020 FORMAT (17X,'SYNOPSIS REPORT - CAP-88 (1.00)'/)

WRITE (12,1030) ID\_CODE, DATE\_AND\_TIME

WRITE (13,1040) ID\_CODE

1030 FORMAT (1X,'ID Code: ',A,2X,'Date/Time: ',A/)

1040 FORMAT (A,10X,'ID\_Code')

C\*\*\*\*\* WRITE FACILITY INFO \*\*\*\*\*

CALL W\_FACILITY\_INFO

C\*\*\*\*\*  
C\*\*\* BEFORE THIS DID NOT NEED WORRY ABOUT PAGING ETC. FROM ON YOU DO  
C\*\*\* THEREFORE, FOR EASE IN CHANGING AND MODIFYING THE ORDER OF THE  
C\*\*\* REPORT, A CALL TO THE OUTPUT\_STRING ROUTINE WILL ALWAYS BE MADE  
C\*\*\* THIS ROUTINE WILL CHECK FOR PAGING AND REPRINTING OF TITLES  
C\*\*\*\*\*

PAGE\_NO = 1  
LINE\_COUNT = 15

C\*\*\*\*\*  
C\*\*\* CHECK TO SEE IF POPULATION OR INDIVIDUAL RUN AND WRITE RESULTS  
C\*\*\* ACCORDINGLY ( INDPOP = 1 --> POPULATION, 0 --> INDIV RUN )  
C\*\*\*\*\*

IF ( INDPOP .EQ. 0 ) THEN

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```
C***** WRITE INDIVIDUAL RESULTS *****
      CALL INDIV_RESULTS ( RN_RUN, NUMBER_NUCS, NORGN, ORGN, ILOC,
+      NUMBER_STACKS, ANG)
      ELSE
      REPORT_NUMBER - 1
```

```
C***** WRITE POPULATION RESULTS *****
      CALL POP_RESULTS ( RN_RUN, REPORT_NUMBER, ORGN, NORGN )
```

```
C***** WRITE INDIVIDUAL RESULTS *****
      CALL INDIV_RESULTS ( RN_RUN, NUMBER_NUCS, NORGN, ORGN, ILOC,
+      NUMBER_STACKS, ANG)
      ENDIF
```

```
C***** WRITE RADIONUCLIDE INFORMATION *****
      CALL WRITE_NUC_INFO( NUMBER_STACKS, STACK_NUMBERS)
```

```
C***** WRITE SITE INFORMATION *****
      CALL W_SITE_INFO
```

```
C***** WRITE EMISSION INFORMATION *****
      CALL W_EMISSION_INFO (STACK_NUMBERS)
```

```
C***** WRITE FOOD SUPPLY INFO *****
      CALL W_FOOD_INFO (FOOD_ARRAY_INFO)
```

```
C*****
C*** DETERMINE IF POPULATION OR INDIVIDUAL AND WRITE EITHER THE
C*** DISTANCES USED IF INDIV, OR THE POPOULATION ARRAY IF POP RUN
C*****
```

```
      IF ( INDPOP .EQ. 0 ) THEN
```

```
C***** WRITE INDIV DISTANCES ON SAME PAGE *****
C*** MAKE THE NEXT OUTPUT ALL A .TRUE., .TRUE. IE. NEVER SPLIT IT UP
```

```
      WRITE (LINES,1050) ( IDIST(I), I - NRL,NRU )
1050      FORMAT (/ 17X,
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

+          'DISTANCES USED FOR MAXIMUM INDIVIDUAL ASSESSMENT',
+          /17X,'-----',
+          / 1X, 10(I6,1X) )
IF ( NRU .LE. 10 ) THEN
  CALL OUTPUT_STRING ( LINES, 4 , .TRUE., .FALSE. )
ELSE
  CALL OUTPUT_STRING ( LINES, 5 , .TRUE., .TRUE. )
ENDIF

ELSE
C***** IF POP RUN OUTPUT POP ARRAY NOW *****
  CALL WRITE_POP_ARRAY( POP, IDIST, NOL, NOU, NRL, NRU )
ENDIF

C***** WRITE THE NAMES OF FILES REFERENCED FOR THE RUN *****
  WRITE (LINES,1060)
1060  FORMAT (// 23X,'REFERENCE FILE NAMES FOR ASSESSMENT',
+        / 23X,'-----' /)
  CALL OUTPUT_STRING ( LINES, 5, .TRUE., .FALSE. )

DO 10 I = 1, NUMBER_FILES
  WRITE (LINES,1010) FILES_USED(I)
  CALL OUTPUT_STRING ( LINES, 1, .FALSE., .FALSE. )

10 CONTINUE

C*****
C***   IF POP RUN OUTPUT RISK FREQUENCY TABLE
C***   IN FACTORS OF 10 FOR DOE
C*****

IF ( INDPOP .EQ. 1 ) THEN
  REPORT_NUMBER = 2
C***   FORCE A NEW PAGE TO BE DONE FOR THIS REPORT
  LINE_COUNT = 66
  CALL NEW_RISK_FREQ_REPORT ( REPORT_NUMBER )

ENDIF

RETURN

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

END

```
C*****
C
C           OUTPUT ROUTINES FOR THE SYNOPSIS REPORT           *
C           CALLED BY WRITE SYNOPSIS REPORT                   *
C                                                                 *
C*****
```

```
C-----
C
C           SUBROUTINE W SITE INFO                             -
C                                                                 -
C-----
```

C     SUBROUTINE W\_SITE\_INFO

C     VARIABLES PASSED

C     GLOBAL VARIABLES

```
INTEGER*4  TEMPERATURE, RAINFALL_RATE, LID_HEIGHT
REAL       LATITUDE, LONGITUDE
```

```
COMMON / SITE_INFO / TEMPERATURE, RAINFALL_RATE, LID_HEIGHT,
+      LATITUDE, LONGITUDE
```

C     LOCAL VARIABLES

C     CHARACTER\*90 LINES (20)

C\*\* BEGIN

```
WRITE (LINES,1000)
1000 FORMAT (/ 32X, 'SITE INFORMATION' / 32X, '-----' / )
```

```
CALL OUTPUT_STRING ( LINES, 4, .TRUE., .FALSE. )
```

```
WRITE (LINES,1010) TEMPERATURE
WRITE (13,1020) TEMPERATURE
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0           1           2           3           4           5           6           7  
123456789012345678901234567890123456789012345678901234567890123456789012

1010 FORMAT ( 7X, 'Temperature:', 4X, I3, 2X, 'C' )  
1020 FORMAT (I3,10X,'TEMP')

CALL OUTPUT\_STRING ( LINES, 1, .FALSE., .FALSE. )

WRITE (LINES,1030) RAINFALL\_RATE  
WRITE (13,1040) RAINFALL\_RATE  
1030 FORMAT ( 10X, 'Rainfall:', 4X, I3, 2X, 'cm/yr' )  
1040 FORMAT (I3,10X,'RAINFALL')

CALL OUTPUT\_STRING ( LINES, 1, .FALSE., .FALSE. )

WRITE (LINES,1050) LID\_HEIGHT  
WRITE (13,1060) LID\_HEIGHT  
1050 FORMAT ( 5X, 'Mixing Height:', 3X, I4, 1X, 'meters' )  
1060 FORMAT (I4,10X,'LID\_HITE')

CALL OUTPUT\_STRING ( LINES, 1, .FALSE., .FALSE. )

WRITE (13,1070) LATITUDE  
1070 FORMAT (F8.4,10X,'LATITUDE')  
WRITE (13,1080) LONGITUDE  
1080 FORMAT (F8.4,10X,'LONGITUDE')  
RETURN  
END

---

C  
C  
C                                   SUBROUTINE W FACILITY INFO  
C  
C

---

C    SUBROUTINE W\_FACILITY\_INFO

C    GLOBAL VARIABLES  
C

CHARACTER\*80 COMMENTS(2)  
CHARACTER\*72 FACILITY, ADDRESS  
CHARACTER\*38 SOURCE\_CATEGORY  
CHARACTER\*25 CITY  
CHARACTER\*10 ZIPCODE  
CHARACTER\*4 YEAR



DARTAB (DARTAB2.FOR) Program File  
(continued)

0           1           2           3           4           5           6           7  
123456789012345678901234567890123456789012345678901234567890123456789012

1060 FORMAT (A,10X,'State')  
1070 FORMAT (A,10X,'Zipcode')

WRITE (12,1080) SOURCE\_CATEGORY, YEAR  
1080 FORMAT (/1X, 'Source Category: ',A, 2X, 'Source Term: ',A )  
WRITE (13,1090) YEAR  
WRITE (14,1090) YEAR  
1090 FORMAT (A,10X,'YEAR')

WRITE (12,2000)  
2000 FORMAT (/1X, 'Comments: ' )

DO 10 I = 1,2  
  IF ( COMMENTS (I) .EQ. BLANK\_LINE ) THEN  
    GO TO 20  
  ELSE  
    NUM\_COMMENTS = NUM\_COMMENTS + 1  
  ENDIF

10 CONTINUE

20 WRITE (13,2005) NUM\_COMMENTS  
2005 FORMAT (I2,10X,'NUM\_CMTS')

WRITE (12,2010) COMMENTS(1)  
WRITE (12,2010) COMMENTS(2)

C\*\*\* OUTPUT TO SAS FILES ONLY THE EXACT NUMBER OF COMMENT LINES

DO 30 I = 1, NUM\_COMMENTS  
  WRITE (13,2020) COMMENTS(I)  
30 CONTINUE

2010 FORMAT (1X,A )  
2020 FORMAT (A,10X,'COMMENTS')

RETURN  
END

```
C-----  
C                                     =  
C                SUBROUTINE WRITE NUC INFO                =  
C                                     =  
C-----
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

SUBROUTINE WRITE\_NUC\_INFO( NUMBER\_STACKS, STACK\_NUMBERS)

C    VARIABLES PASSED  
C

INTEGER\*2    NUMBER\_STACKS  
CHARACTER\*1 STACK\_NUMBERS (6)

C    GLOBAL VARIABLES  
C

CHARACTER\*8    NAME\_NUC(36)  
CHARACTER\*1    ISOL(36)  
COMMON / NUCCHARINFO / NAME\_NUC, ISOL

INTEGER\*2    NUMBER\_NUCS, DECAY\_CHAIN\_FLAG (36), DAUGHTERS  
LOGICAL        RN\_RUN  
REAL            AMAD(36), RELEASE\_RATE(36,6), ALAMSUR(36)  
COMMON / NUC\_INFO / AMAD, RELEASE\_RATE, NUMBER\_NUCS,  
+                                    DAUGHTERS, DECAY\_CHAIN\_FLAG, ALAMSUR, RN\_RUN

CHARACTER\*80 COMMENTS(2)  
CHARACTER\*72 FACILITY, ADDRESS  
CHARACTER\*38 SOURCE\_CATEGORY  
CHARACTER\*25 CITY  
CHARACTER\*10 ZIPCODE  
CHARACTER\*4    YEAR  
CHARACTER\*2    STATE  
COMMON / FACIL\_INFO / COMMENTS, FACILITY, ADDRESS,  
+                                    SOURCE\_CATEGORY, CITY, ZIPCODE, YEAR, STATE

C    LOCAL VARIABLES  
C

INTEGER\*2    I, J  
CHARACTER\*8 DASHES(6)  
CHARACTER\*90 LINES (20)  
CHARACTER\*5    UNITS (6)  
CHARACTER\*7    STKWORD (6)  
CHARACTER\*61 FORMAT1, FORMAT2, FORMAT3  
REAL TOTREL

DATA UNITS / 'Ci/yr', 'Ci/yr', 'Ci/yr', 'Ci/yr', 'Ci/yr', 'Ci/yr' /  
DATA STKWORD / 'Stack #', 'Stack #', 'Stack #', 'Stack #',

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

+                    'Stack #', 'Stack #'/
DATA DASHES /6*'-----'/
DATA FORMAT1 /
+ '( ' Nuclide    Class   Amad   ',1X, (A5, 4X), 3X, ''TOTAL'') '/
DATA FORMAT2 /
+ '( ' -----        -----        ',2X, ( A8,1X ),4X, ''-----' )'/
DATA FORMAT3 /
+ '(1X, A8, 4X, A1, 4X, F4.2, 2X, (1PE8.2, 1X), 1X, 1PE10.2 )'/'

WRITE (LINES,1010) YEAR
1010 FORMAT ( /28X, 'SOURCE TERM ( ' ,A4, ' )'
+        /28X, '-----' )
CALL OUTPUT_STRING ( LINES, 3, .TRUE., .FALSE. )
WRITE (LINES, 1015) ( STKWORD (I), STACK_NUMBERS (I),
+                    I - 1,NUMBER_STACKS )
1015 FORMAT ( 24X, 6( A7, A1, 1X ) )

FORMAT1 (34:34) = STACK_NUMBERS ( NUMBER_STACKS )
WRITE (LINES(2), FORMAT1) ( UNITS (I), I - 1, NUMBER_STACKS )
C 1020 FORMAT ( ' Nuclide    Class   Amad   ',1X, 6(A, 4X), 1X, A6)

FORMAT2 (34:34) = STACK_NUMBERS ( NUMBER_STACKS )
WRITE (LINES(3), FORMAT2) ( DASHES (I) , I - 1,NUMBER_STACKS )
C1030 FORMAt ( ' -----        -----        ', 2X, ( A8,1X ) A5 )

CALL OUTPUT_STRING ( LINES, 3, .TRUE., .TRUE. )

IF ( .NOT. RN_RUN ) THEN
WRITE (14,1086) DAUGHTERS
ENDIF
DO 10 I = 1, NUMBER_NUCS
TOTREL = 0.0
DO 5 J = 1,NUMBER_STACKS
TOTREL = RELEASE_RATE (I,J) + TOTREL
5 CONTINUE

FORMAT3 (32:32) = STACK_NUMBERS ( NUMBER_STACKS )
WRITE (LINES, FORMAT3) NAME_NUC (I), ISOL (I), AMAD (I),
+        ( RELEASE_RATE (I,J), J = 1, NUMBER_STACKS ), TOTREL
CALL OUTPUT_STRING ( LINES, 1, .FALSE., .TRUE. )

C*** OUTPUT THE NUC INFO FOR THE SAS CODE

WRITE (14,1060) NAME_NUC (I)
WRITE (14,1070) ISOL (I)

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0 1 2 3 4 5 6 7  
12345678901234567890123456789012345678901234567890123456789012

```
WRITE (14,1080) AMAD (I)
WRITE (14,1085) ALAMSUR(I)
IF ( .NOT. RN_RUN ) THEN
  WRITE (14,1087) DECAY_CHAIN_FLAG (I)
ENDIF
WRITE (14,1090) ( RELEASE_RATE (I,J),
+               J - 1, NUMBER_STACKS )
C 1050  FORMAT (1X, A8, 4X, A1, 4X, F4.2, 2X, 6(1PE8.2, 1X), 1PE10.2)
1060  FORMAT (A8,10X,'NAME_N')
1070  FORMAT (A1,10X,'ISOL')
1080  FORMAT (F4.2,10X,'AMAD')
1085  FORMAT (E10.3,10X,'LAMSUR')
1086  FORMAT (I1,10X,'DAUGHTERS FLAG')
1087  FORMAT (I1,10X,'DECAY CHAIN FLAG')
1090  FORMAT (1(1PE8.2,5X,'RRATE'))

10  CONTINUE

      RETURN
      END
```

```
C-----
C
C              SUBROUTINE W EMISSION INFO
C
C-----
```

```
C  SUBROUTINE W_EMISSION_INFO (STACK_NUMBERS)
```

```
C  VARIABLES PASSED
```

```
CHARACTER*1  STACK_NUMBERS (6)
```

```
C  GLOBAL VARIABLES
```

```
INTEGER*2  NUMBER_STACKS, SOURCE_TYPE, PLUME_RISE_TYPE
REAL        HEIGHT(6), AREA(6), AREA_DIAMETER(6),
+           STACK_DIAMETER(6),BOUYANCY(6), MOMENTUM(6), ENTERED(7)
```

```
COMMON / EMMIS_INFO / HEIGHT, AREA, AREA_DIAMETER,
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

+           STACK_DIAMETER, BOUYANCY, MOMENTUM,
+           ENTERED,
+           NUMBER_STACKS, SOURCE_TYPE, PLUME_RISE_TYPE

```

C    LOCAL VARIABLES  
C                      

```

INTEGER*2    I
CHARACTER*8  DASHES8 (6)
CHARACTER*5  PASQUIL_CATS (7)

CHARACTER*90 LINES (20)

```

C    INITIALIZATION  
C                      

```

DATA DASHES8 / 6*'-----'/
DATA PASQUIL_CATS / '--A--', '--B--', '--C--', '--D--', '--E--',
+                '--F--', '--G--'/

```

C\*\*\* BEGIN

```

WRITE (LINES,1000)
1000 FORMAT (/ 29X, 'EMISSION INFORMATION' / 29X,
+         '-----'/ )
CALL OUTPUT_STRING ( LINES, 4, .TRUE., .FALSE. )

```

```

WRITE (LINES(1),1010) (STACK_NUMBERS(I), I = 1, NUMBER_STACKS )
1010 FORMAT ( 10X, ' Stack Number:',4X, 6( A , 8X ) )

```

```

WRITE ( LINES(2),1020) (DASHES8 (I), I = 1, NUMBER_STACKS )
1020 FORMAT ( 25X, 6( A, 1X ) )

```

```

CALL OUTPUT_STRING ( LINES, 2, .TRUE., .TRUE. )

```

```

C*** DETERMINE IF STACK OR AREA SOURCE AND WRITE ACCORDINGLY
C*** SOURCE TYPE = 0 --> STACK, 1 --> AREA
WRITE(13,1030)SOURCE_TYPE
1030 FORMAT (11,10X,'SRC_TYPE')

```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

IF ( SOURCE\_TYPE .EQ. 0 ) THEN

WRITE (LINES,1040) ( HEIGHT(I), I = 1,NUMBER\_STACKS )  
WRITE (13,1050) ( HEIGHT(I), I = 1,NUMBER\_STACKS )

1040        FORMAT ( 2X, 'Stack Height (meters) :', 2X, 6( F6.2, 3X ) )  
1050        FORMAT (1(F6.2,10X,'HEIGHT'))  
          CALL OUTPUT\_STRING ( LINES, 1, .FALSE., .TRUE. )

WRITE (LINES,1060) (STACK\_DIAMETER (I), I = 1,NUMBER\_STACKS)  
WRITE (13,1070) (STACK\_DIAMETER (I), I = 1,NUMBER\_STACKS)  
1060        FORMAT ( 1X, 'Stack Diameter (meters):', 2X, 6( F6.2, 3X ) )  
1070        FORMAT (1(F6.2,10X,'DIAM'))

CALL OUTPUT\_STRING ( LINES, 1, .FALSE., .TRUE. )

C\*\*\*        DETERMINE PLUME RISE TYPE AND WRITE ACCORDINGLY FOR STACK SOURCE  
C\*\*\*        0 → BOUYANT, 1 → MOMENTUM, 2 → ENTERED

WRITE (LINES,1080)  
1080        FORMAT ( 2X, 'Plume Rise' )  
          CALL OUTPUT\_STRING ( LINES, 1, .FALSE., .TRUE. )

WRITE (13,1090)PLUME\_RISE\_TYPE  
1090        FORMAT (I1,10X,'PR\_TYPE')

IF ( PLUME\_RISE\_TYPE .EQ. 0 ) THEN

WRITE (LINES,2000) ( BOUYANCY (I), I = 1, NUMBER\_STACKS )  
WRITE (13,2010) ( BOUYANCY (I), I = 1, NUMBER\_STACKS )  
2000        FORMAT ( 6X, 'Bouyant (cal/sec) :', 6( 1PE8.2, 1X ) )  
2010        FORMAT (1(1PE8.2,5X,'PL\_RISE'))  
          CALL OUTPUT\_STRING ( LINES, 1, .FALSE., .TRUE. )

ELSE IF ( PLUME\_RISE\_TYPE .EQ. 1 ) THEN

WRITE (LINES,2020) ( MOMENTUM (I), I = 1, NUMBER\_STACKS )  
WRITE (13,2010) ( MOMENTUM (I), I = 1, NUMBER\_STACKS )  
2020        FORMAT ( 7X, 'Momentum (m/sec) :', 6( 1PE8.2, 1X ) )  
          CALL OUTPUT\_STRING ( LINES, 1, .FALSE., .TRUE. )

ELSE

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

+          WRITE (LINES,2030) ( PASQUIL_CATS(I), I = 1,7 ),
          ( ENTERED (I), I = 1,7 )
2030      WRITE (13,2040) ( ENTERED (I), I = 1,7 )
          FORMAT ( 26x,7(A5,3X) / 5x, 'Entered (meters) :',
+          7(1X,F7.2) )
2040      FORMAT (1(1PE8.2,5X,'PL_RISE'))

          CALL OUTPUT_STRING ( LINES, 2, .TRUE., .FALSE. )

      ENDIF

      ELSE
C***      AREA SOURCE

          WRITE (LINES,2050) ( HEIGHT(I), I = 1,NUMBER_STACKS )
2050      FORMAT ( 2X, 'Area Height (meters) :', 2X, 6( F6.2, 3X ) )
          CALL OUTPUT_STRING ( LINES, 1, .FALSE., .TRUE. )

          WRITE (13,2060) (HEIGHT(I), I = 1,NUMBER_STACKS)
2060      FORMAT (1(F6.2,5X,'HEIGHT'))

          WRITE (13,2070) (AREA_DIAMETER (I), I=1,NUMBER_STACKS )
2070      FORMAT (1(1PE8.2,5X,'DIAM'))

          WRITE (LINES,2080) ( AREA (I), I = 1,NUMBER_STACKS )
          WRITE (13,2090) ( AREA (I), I = 1,NUMBER_STACKS )
2080      FORMAT (9X, 'Area ( sq. m) :', 6( 1X, 1PE8.2 ) )
2090      FORMAT (1(1PE8.2,5X,'AREA'))
          CALL OUTPUT_STRING ( LINES, 1, .FALSE., .TRUE. )

          WRITE (LINES,3000) ( AREA_DIAMETER (I), I = 1,NUMBER_STACKS )
3000      FORMAT (1X, 'Area Diameter (meters):', 6( 1X, 1PE8.2 ) )
          CALL OUTPUT_STRING ( LINES, 1, .FALSE., .TRUE. )

      ENDIF

      RETURN
      END
```

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



DARTAB (DARTAB2.FOR) Program File  
(continued)

0           1           2           3           4           5           6           7  
123456789012345678901234567890123456789012345678901234567890123456789012

```
WRITE (LINES,1020) VEG_LOCAL, VEG_REGIONAL, VEG_IMPORTED
WRITE (13,1030) VEG_LOCAL
WRITE (13,1040) VEG_REGIONAL
WRITE (13,1050) VEG_IMPORTED
1020 FORMAT ( 10X, 'Vegetable: ', 2X, F5.3, 6X, F5.3, 8X, F5.3 )
1030 FORMAT ( F5.3,10X,'VEG_LOC')
1040 FORMAT ( F5.3,10X,'VEG_REG')
1050 FORMAT ( F5.3,10X,'VEG_IMP')
CALL OUTPUT_STRING ( LINES, 1, .FALSE., .TRUE. )

WRITE (LINES,1060) MEAT_LOCAL, MEAT_REGIONAL, MEAT_IMPORTED
WRITE (13,1070) MEAT_LOCAL
WRITE (13,1080) MEAT_REGIONAL
WRITE (13,1090) MEAT_IMPORTED
1060 FORMAT ( 15X, 'Meat: ', 2X, F5.3, 6X, F5.3, 8X, F5.3 )
1070 FORMAT ( F5.3,10X,'BEEF_LOC')
1080 FORMAT ( F5.3,10X,'BEEF_REG')
1090 FORMAT ( F5.3,10X,'BEEF_IMP')
CALL OUTPUT_STRING ( LINES, 1, .FALSE., .TRUE. )

WRITE (LINES,2000) MILK_LOCAL, MILK_REGIONAL, MILK_IMPORTED
WRITE (13,2010) MILK_LOCAL
WRITE (13,2020) MILK_REGIONAL
WRITE (13,2030) MILK_IMPORTED
2000 FORMAT ( 15X, 'Milk: ', 2X, F5.3, 6X, F5.3, 8X, F5.3 )
2010 FORMAT (F5.3,10X,'MILK_LOC')
2020 FORMAT (F5.3,10X,'MILK_REG')
2030 FORMAT (F5.3,10X,'MILK_IMP')
CALL OUTPUT_STRING ( LINES, 1, .FALSE., .TRUE. )

WRITE (LINES,2040) FOOD_ARRAY_INFO
2040 FORMAT ( / 1X, A )
CALL OUTPUT_STRING ( LINES, 2, .FALSE., .FALSE. )

RETURN
END
```

```
C-----
C
C                               SUBROUTINE INDIV RESULTS                               -
C
C-----
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

SUBROUTINE INDIV_RESULTS( RN_RUN, NUMBER_NUCS, NORGN, ORGN, ILOC,
+
          NUMBER_STACKS, ANG)
C

```

```

C
VARIABLES PASSED
C
LOGICAL   RN_RUN
INTEGER*2 NUMBER_NUCS,NUMBER_STACKS
INTEGER*4 ANG
CHARACTER*8 ORGN(20)
INTEGER  NORGN

```

```

C
GLOBAL VARIABLES
C
REAL WLI, MAX_PCI_LITERS, FATAL_CANCER_RISK, ORGAN_DOSES(20),
+
  PATHWAY_DOSES(4), NUC_DOSES(36)
INTEGER LOC_DIST

```

```

COMMON / IND_RESULTS / WLI, LOC_DIST, MAX_PCI_LITERS,
+
          FATAL_CANCER_RISK ,ORGAN_DOSES,
+
          PATHWAY_DOSES ,NUC_DOSES

```

```

C
LOCAL VARIABLES
C
CHARACTER*15 DIRECTIONS(16)
CHARACTER*90 LINES(20)

```

```

C
INITIALIZATION
C
DATA DIRECTIONS /'NORTH          ','NORTH NORTHWEST',
+
                'NORTHWEST      ','WEST NORTHWEST ',
+
                'WEST           ','WEST SOUTHWEST ',
+
                'SOUTHWEST      ','SOUTH SOUTHWEST',
+
                'SOUTH          ','SOUTH SOUTHEAST',
+
                'SOUTHEAST     ','EAST SOUTHEAST',
+
                'EAST           ','EAST NORTHEAST',
+
                'NORTHEAST      ','NORTH NORTHEAST'/'

```

C\*\*\*\* CHECK FOR RADON RUN, IF MORE THAN 1 NUCLIDE AND RADON FLAG IS SET  
C\*\*\*\* OUTPUT FOR NON RADON RUN ELSE OUTPUT FOR RADON RUN

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

WRITE(14,1006) LOC_DIST
1006 FORMAT (I6,10X,'MAX_DIST')

WRITE(14,1007) ANG
1007 FORMAT(A4,10X,'MAX_DIR')

IF ( .NOT. RN_RUN ) THEN
  WRITE(14,1008) (NORGN-1)
1008  FORMAT(I2,10x,'NUM_ORG')
ENDIF

WRITE (13,1023) NUMBER_STACKS
WRITE (14,1023) NUMBER_STACKS
1023  FORMAT (I1,10X,'NUM_STCK')

IF (.NOT. RN_RUN ) THEN
  WRITE (14,1033) NUMBER_NUCS
  WRITE (13,1033) NUMBER_NUCS
1033  FORMAT (I2,10X,'NUM_NUCS')
ENDIF

IF ( RN_RUN) THEN

  WRITE (LINES, 1000 )
1000  FORMAT ( / 10X, 'RN-222 EXPOSURE AND RISK FOR THE INDIVIDUAL',
+      ' AT MAXIMUM RISK',
+      / 10X, '-----',
+      '-----')
  CALL OUTPUT_STRING ( LINES, 3, .TRUE., .FALSE. )

  WRITE (LINES,1010) LOC_DIST, DIRECTIONS(ILOC)
1010  FORMAT ( / 1X, 'Location to the individual:',I6,
+      ' METERS',1X, A15 )
  CALL OUTPUT_STRING ( LINES, 2, .FALSE., .FALSE. )

  WRITE (LINES,1020) WLI
  WRITE (14,1021) WLI
1020  FORMAT ( / 6X, 'Exposure in Working Levels:', 1PE10.2 )
1021  FORMAT (1PE10.2,10X,'IND_WL')
  CALL OUTPUT_STRING ( LINES, 2, .FALSE., .FALSE. )

  WRITE (LINES,1030) MAX_PCI_LITERS
  WRITE (14,1031) MAX_PCI_LITERS
1030  FORMAT ( 6X, 'pCi/liter at that location:', 1PE10.2 )
1031  FORMAT (1PE10.2,10X,'MAX_CONC')
  CALL OUTPUT_STRING ( LINES, 1, .FALSE., .FALSE. )

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

WRITE (LINES, 1040) FATAL_CANCER_RISK
WRITE (14,1041) FATAL_CANCER_RISK
1040  FORMAT ( 6X, 'Lifetime Fatal Cancer Risk:', 1PE10.2 )
1041  FORMAT (1PE10.2,10X,'LIFERISK')
      CALL OUTPUT_STRING ( LINES, 1, .FALSE., .FALSE. )

ELSE

WRITE (LINES, 1050)
1050  FORMAT ( / 25X, 'INDIVIDUAL AT MAXIMUM RISK ASSESSEMENT',
+      / 31X,      '(RN-222 RISKS EXCLUDED)',
+      / 31X,      '-----')
      CALL OUTPUT_STRING ( LINES, 4, .TRUE., .FALSE. )

WRITE (LINES,1010) LOC_DIST, DIRECTIONS(ILOC)
      CALL OUTPUT_STRING ( LINES, 2, .FALSE., .FALSE. )

WRITE (LINES,1060) ( ORGN(J), J = 1, NORGN - 1)
1060  FORMAT ( / 16X, 7( A8,2X ) / 2X, 'Organ dose')
      CALL OUTPUT_STRING ( LINES, 3, .TRUE., .TRUE. )

WRITE (LINES,1070) ( ORGAN_DOSES (J), J = 1, NORGN - 1)
WRITE (14,1071) ( ORGAN_DOSES (J), J = 1, NORGN - 1)
WRITE (14,1072) ( PATHWAY_DOSES (J), J = 1, 4)
WRITE (14,1073) ( NUC_DOSES (J), J = 1, NUMBER_NUCS)
1070  FORMAT (3X, '(mrem/yr) : ', 8(1PE8.1,2X) )
1071  FORMAT (1(1PE10.2,5X,'O_DOSE'))
1072  FORMAT (1(1PE10.2,5X,'P_DOSE'))
1073  FORMAT (1(1PE10.2,5X,'N_DOSE'))
      CALL OUTPUT_STRING ( LINES, 1, .TRUE., .FALSE. )

WRITE (LINES,1080) ORGAN_DOSES (NORGN)
WRITE (14,1081) ORGAN_DOSES (NORGN)
1080  FORMAT ( / 2X, 'ICRP Effective Dose Equivalent (mrem/yr):',
+      1PE10.2)
1081  FORMAT (1PE10.2, 10X,'I_EFECTIV')
      CALL OUTPUT_STRING ( LINES, 2, .FALSE., .FALSE. )

WRITE (LINES, 1090) FATAL_CANCER_RISK
WRITE (14,1041) FATAL_CANCER_RISK
1090  FORMAT ( 2X, 'Lifetime Fatal Cancer Risk           :',
+      1PE10.2 )
      CALL OUTPUT_STRING ( LINES, 1, .FALSE., .FALSE. )

```

ENDIF

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

RETURN  
END

```
C-----
C
C                SUBROUTINE POP RESULTS
C
C-----
```

SUBROUTINE POP\_RESULTS( RN\_RUN, REPORT\_NUMBER, ORGN, NORGN )

C

VARIABLES PASSED

C

C

LOGICAL RN\_RUN  
 INTEGER\*2 REPORT\_NUMBER  
 CHARACTER\*8 ORGN (20)  
 INTEGER NORGN

C

GLOBAL VARIABLES

C

REAL SRISK(400), RSKLIN(400), EFFECT\_PERSON\_REM,  
 + PERSON\_WORKING\_LEVEL, POP\_ORGAN\_DOSES(20)

INTEGER POPLIN(400)

INTEGER IO(400), NLOC, NOP

COMMON / POPU\_RESULTS / POPLIN, RSKLIN, SRISK, IO, NLOC, NOP,  
 + EFFECT\_PERSON\_REM, PERSON\_WORKING\_LEVEL,  
 + POP\_ORGAN\_DOSES

C

LOCAL VARIABLES

C

CHARACTER\*90 LINES(20)

C\*\*\*\* CHECK FOR RADON RUN, IF MORE THAN 1 NUCLIDE AND RADON FLAG IS SET  
 C\*\*\*\* OUTPUT FOR NON RADON RUN ELSE OUTPUT FOR RADON RUN

```
IF ( RN_RUN ) THEN
  WRITE (LINES, 1000 )
1000  FORMAT ( / 25X, 'RN-222 POPULATION ASSESSMENT',
  +       / 25X, '-----' /)
  CALL OUTPUT_STRING ( LINES, 4, .TRUE., .FALSE. )
```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

      WRITE (LINES, 1020) PERSON_WORKING_LEVEL
      WRITE (14,1021) PERSON_WORKING_LEVEL
1020  FORMAT (1X, 'Collective Exposure (Person Working Levels):',
+      1PE10.2 )
1021  FORMAT (1PE10.2,10X,'POP_WL')
      CALL OUTPUT_STRING ( LINES, 1, .FALSE., .FALSE. )

```

ELSE

```

      WRITE (LINES, 1010)
1010  FORMAT ( / 29X, 'POPULATION ASSESSMENT',
+      / 26X, '(RN-222 DOSE/RISK EXCLUDED)',
+      / 26X, '-----' /)
      CALL OUTPUT_STRING ( LINES, 5, .TRUE., .FALSE. )

```

```

      WRITE (LINES, 1030) EFFECT_PERSON_REM
      WRITE (14,1031) EFFECT_PERSON_REM
1030  FORMAT (1X, 'ICRP Collective Effective Dose Equivalent',
+      ' (Person-Rem/Year):', 1PE10.2)
1031  FORMAT (1PE10.2,10X,'P_EFECTIV')
      CALL OUTPUT_STRING ( LINES, 1, .FALSE., .FALSE. )

```

```

      WRITE (LINES,1060) ( ORGN(J), J = 1, NORGN - 1)
1060  FORMAT (/ 1X,'Collective Population'/,
+      / 16X, 7( A8,2X ) / 2X, 'Organ dose')
      CALL OUTPUT_STRING ( LINES, 5, .TRUE., .TRUE. )

```

```

      WRITE (LINES,1070) ( POP_ORGAN_DOSES (J), J = 1, NORGN - 1)
1070  FORMAT (2X, '(P-REM/YR) :', 8(1PE8.1,2X) )
      CALL OUTPUT_STRING ( LINES, 1, .TRUE., .FALSE. )

```

ENDIF

CALL NEW\_RISK\_FREQ\_REPORT ( REPORT\_NUMBER )

RETURN  
END

---

```

C
C
C           SUBROUTINE WRITE POP ARRAY
C
C

```

---

SUBROUTINE WRITE\_POP\_ARRAY( POP, IDIST, NOL, NOU, NRL, NRU)

DARTAB (DARTAB2.FOR) Program File  
(continued)

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

```
C      _____
```

```
C      VARIABLES PASSED
```

```
C      _____
```

```
INTEGER  NOL, NOU, NRL, NRU, IDIST(20)
REAL     POP(20,20)
```

```
C      LOCAL VARIABLES
```

```
C      _____
```

```
INTEGER*2 REPEAT_NUMBER, BEG_DIST_PTR, END_DIST_PTR
INTEGER*2 I, J, K
```

```
CHARACTER*5 DASHES(20)
CHARACTER*3 DIRECTION(16)
CHARACTER*90 LINES(20)
```

```
DATA DIRECTION / ' N','NNW',' NW','WNW',' W','WSW',' SW',
+                'SSW',' S','SSE',' SE','ESE',' E','ENE',
+                ' NE','NNE' /
```

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

```
WRITE (LINES, 1010)
1010 FORMAT ( / 12X, 'POPULATION ARRAY (1980 Census)',
+           / 12X, '-----' / )
CALL OUTPUT_STRING ( LINES, 4, .TRUE., .FALSE. )
```

```
C***  DETERMINE NUMBER OF TIMES DIRECTIONS PRINTED
```

```
REPEAT_NUMBER = (( NRU-NRL ) / 7) + 1
DO 20 I = 1, REPEAT_NUMBER
```

```
BEG_DIST_PTR = NRL + ( 7 * ( I - 1 ) )
END_DIST_PTR = MINO (BEG_DIST_PTR+6, NRU )
```

```
WRITE (LINES, 1020 ) (IDIST(J), J = BEG_DIST_PTR, END_DIST_PTR)
1020 FORMAT ( /, 5X, 7(I7) )
```

```
WRITE (LINES(3), 1030 )
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

+          (DASHES(J),J = BEG_DIST_PTR, END_DIST_PTR)
1030  FORMAT (   5X, 7( 2X,A5 ) )

      CALL OUTPUT_STRING ( LINES, 3, .TRUE., .TRUE. )

      DO 10 K = NOL, NOU

        WRITE (LINES, 1040 ) DIRECTION(K),
+          (POP(K,J), J = BEG_DIST_PTR, END_DIST_PTR)
1040  FORMAT (1X, A3, 1X, 7F7.0)
      CALL OUTPUT_STRING ( LINES, 1, .FALSE., .TRUE. )

10    CONTINUE

20    CONTINUE

      RETURN
      END

```

```

C ****
C   NEW REPORT ADDED 1/88 BY JOAN MCGUE
C ****
C _____

```

```

C _____
C                                     =
C   SUBROUTINE NEW RISK FREQ REPORT   =
C                                     =
C _____

```

SUBROUTINE NEW\_RISK\_FREQ\_REPORT ( REPORT\_NUMBER )

```

REAL   SRISK(400), RSKLIN(400), EFFECT_PERSON_REM,
+     PERSON_WORKING_LEVEL, POP_ORGAN_DOSES(20)

```

INTEGER POPLIN(400), IO(400), NLOC, NOP

```

COMMON / POPU_RESULTS / POPLIN, RSKLIN, SRISK, IO, NLOC, NOP,
+     EFFECT_PERSON_REM, PERSON_WORKING_LEVEL,
+     POP_ORGAN_DOSES

```

INTEGER\*2 NUM\_INTERVALS, REPORT\_NUMBER  
 INTEGER NUMBER\_OF\_PEOPLE, INDEX, TOTAL\_NUMBER\_OF\_PEOPLE

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

REAL NUMBER\_OF\_DEATHS, TOTAL\_NUMBER\_OF\_DEATHS  
 REAL RISK\_INTERVALS(14,2)

CHARACTER\*18 RISK\_FORMATS(15,2)  
 CHARACTER\*90 LINES (20)

```
DATA RISK_INTERVALS /.1,.01,.001,.0001,.00001,.000001,
+                   .0,.0,.0,.0,.0,.0,.0,.0,
+                   .3,.1,.03,.01,.003,.001,.0003,.0001,
+                   .00003,.00001,.000003,.000001,.0000003,
+                   .0000001 /
```

```
DATA RISK_FORMATS /'1.0E+00 TO 1.0E-01','1.0E-01 TO 1.0E-02',
+                 '1.0E-02 TO 1.0E-03','1.0E-03 TO 1.0E-04',
+                 '1.0E-04 TO 1.0E-05','1.0E-05 TO 1.0E-06',
+                 ' LESS THAN 1.0E-06',',',
+                 ',','',
+                 ',','',
+                 ',','',
+                 ',','1.0E+00 TO 3.0E-01',
+                 '3.0E-01 TO 1.0E-01','1.0E-01 TO 3.0E-02',
+                 '3.0E-02 TO 1.0E-02','1.0E-02 TO 3.0E-03',
+                 '3.0E-03 TO 1.0E-03','1.0E-03 TO 3.0E-04',
+                 '3.0E-04 TO 1.0E-04','1.0E-04 TO 3.0E-05',
+                 '3.0E-05 TO 1.0E-05','1.0E-05 TO 3.0E-06',
+                 '3.0E-06 TO 1.0E-06','1.0E-06 TO 3.0E-07',
+                 '3.0E-07 TO 1.0E-07',' LESS THAN 1.0E-07'/
```

C\*\*\* SET INITIAL VALUES

```
INDEX = 1
NUMBER_OF_PEOPLE = 0
TOTAL_NUMBER_OF_PEOPLE = 0
NUMBER_OF_DEATHS = 0.0
TOTAL_NUMBER_OF_DEATHS = 0.0
```

C\*\*\* DETERMINE NUMBER OF INTERVALS ACCORDING TO FIRST OR SECOND REPORT  
 C\*\*\* BEING PRINTED .. THE FIRST REPORT IS IN FACTORS OF 10 NUM  
 C\*\*\* INTERVALS = 6, THE SECOND REPORT IS IN FACTORS OF 3,  
 C\*\*\* NUM INTERVALS = 14

```
IF ( REPORT_NUMBER .EQ. 1 ) THEN
  NUM_INTERVALS = 6
ELSE
  NUM_INTERVALS = 14
ENDIF
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

WRITE (LINES,90)
90  FORMAT(/17X,'FREQUENCY DISTRIBUTION OF LIFETIME ',
+        'FATAL CANCER RISKS')
CALL OUTPUT_STRING ( LINES, 2, .TRUE., .FALSE. )

```

```

WRITE (LINES,101)
101 FORMAT(/ 34X,'NUMBER OF PEOPLE',2X,'DEATHS/YEAR',2X,'DEATHS/YEAR'
+        / 22X,'NUMBER OF',3X,'AT THIS RISK OR',5X,'AT THIS',
+        4X,'AT THIS RISK'
+        / 8X,'RISK',12X,'PEOPLE',8X,'HIGHER',11X,'RISK',
+        7X,'OR HIGHER'
+        / 8X,'-----',12X,'-----',8X,'-----',11X,'-----',
+        7X,'-----')
CALL OUTPUT_STRING ( LINES, 5, .TRUE., .TRUE. )

```

C THE ARRAY IO IS USED AS A POINTER TO THE UNSORTED ARRAYS.. IT IS  
C USED TO ACCESS THE UNSORTED ARRAYS VALUES IN THE PROPER ORDER.  
C REMEMBER THEY ORGINALLY PARRALLED TRISK (SRISK).

```

I = NLOC
10  CONTINUE
    IF ( ( I .GT. 0 ) .AND. ( INDEX .LE. NUM_INTERVALS ) ) THEN
        IF (SRISK(I) .GE. RISK_INTERVALS(INDEX,REPORT_NUMBER)) THEN
            NUMBER_OF_PEOPLE = NUMBER_OF_PEOPLE + POPLIN( IO(I) )
            NUMBER_OF_DEATHS = NUMBER_OF_DEATHS + RSKLIN( IO(I) )
            I = I - 1
        ELSE
            TOTAL_NUMBER_OF_PEOPLE = TOTAL_NUMBER_OF_PEOPLE +
+            NUMBER_OF_PEOPLE
            TOTAL_NUMBER_OF_DEATHS = TOTAL_NUMBER_OF_DEATHS +
+            NUMBER_OF_DEATHS
            IF (REPORT_NUMBER .EQ. 1) THEN
                WRITE(14,102) NUMBER_OF_PEOPLE
                WRITE(14,105) NUMBER_OF_DEATHS
            ENDIF
            WRITE (LINES, '(1X,A18,1X,I10,4X,I10,7X,1PE10.2,3X,
+            1PE10.2)')
            RISK_FORMATS(INDEX,REPORT_NUMBER),NUMBER_OF_PEOPLE,
+            TOTAL_NUMBER_OF_PEOPLE,NUMBER_OF_DEATHS,
+            TOTAL_NUMBER_OF_DEATHS
            CALL OUTPUT_STRING ( LINES, 1, .FALSE., .TRUE. )

            NUMBER_OF_PEOPLE = 0
            NUMBER_OF_DEATHS = 0.0

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

                INDEX = INDEX + 1
            ENDIF
        GOTO 10
    ENDIF

C**** ELSE
    20    CONTINUE

        IF ( I .GT. 0 ) THEN
            NUMBER_OF_PEOPLE = NUMBER_OF_PEOPLE + POPLIN( IO(I) )
            NUMBER_OF_DEATHS = NUMBER_OF_DEATHS + RSKLIN( IO(I) )
            I = I - 1
            GOTO 20
        ENDIF

        TOTAL_NUMBER_OF_PEOPLE = TOTAL_NUMBER_OF_PEOPLE +
+           NUMBER_OF_PEOPLE
        TOTAL_NUMBER_OF_DEATHS = TOTAL_NUMBER_OF_DEATHS +
+           NUMBER_OF_DEATHS

        WRITE (LINES, '(1X,A18,1X,I10,4X,I10,7X,1PE10.2,3X,1PE10.2)')
+           RISK_FORMATS(INDEX,REPORT_NUMBER),NUMBER_OF_PEOPLE,
+           TOTAL_NUMBER_OF_PEOPLE,NUMBER_OF_DEATHS,
+           TOTAL_NUMBER_OF_DEATHS
        IF(REPORT_NUMBER .EQ.1 )THEN
            WRITE(14,102) NUMBER_OF_PEOPLE
            WRITE(14,105) NUMBER_OF_DEATHS
            WRITE(14,100) TOTAL_NUMBER_OF_DEATHS
            WRITE(14,103) TOTAL_NUMBER_OF_PEOPLE
        ENDIF
    100    FORMAT(1PE10.2,10X,'TOTDEATH')
    102    FORMAT(18 ,10X,'PEOPLE')
    103    FORMAT(18 ,10X,'TOTPEOPL')
    105    FORMAT(1PE10.2,10X,'DEATHS')

        CALL OUTPUT_STRING ( LINES, 1, .FALSE., .TRUE. )

C**      WRITE (LINES,104) SRISK(NOP)
C** 104    FORMAT(/ 3X,'RISK TO THE MAXIMUM INDIVIDUAL --> ',
C**      +           1PE10.2)
C**      CALL OUTPUT_STRING ( LINES, 2, .FALSE., .FALSE. )

C**      WRITE (LINES,105) TOTAL_NUMBER_OF_DEATHS
C** 105    FORMAT(3X,'THE TOTAL NUMBER OF DEATHS/YEAR --> ',
C**      +           1PE10.2)

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

RETURN

END

```
C*****
C
C
C           MATHAMATICAL ROUTINES
C
C
C
C*****
```

```
C-----
C
C           SUBROUTINE MULT
C
C-----
```

```
SUBROUTINE MULT(IM, CONFAC, ARRAYI, NOC, NAMNOC, TITLE, TITL1, TABLE,
> TITL2, NOTE, NUN, RFACI, CREP, NCREP, NDIM, NUCLID, nonclD)
INTEGER TABLE
```

```
CHARACTER*8 NAMNOC, CREP, PUL, NUCLID
```

```
DIMENSION NUCLID(40)
```

```
COMMON/COMRN/WLRN(20, 20), RRISK, RREF(2), RYRLL, NOREP, NRREP, NFREP
```

```
CHARACTER*80 TITLE, NOTE
```

```
CHARACTER*8 TITL2
```

```
CHARACTER*32 NUN
```

```
CHARACTER*40 TITL1
```

```
COMMON/COMEX/EXPP(20, 20, 40, 4), POP(20, 20), POPFAC, TOTFAC, NOL, NOU,
> NRL, NRU, IDIST(20), ILOC, JLOC
```

```
DIMENSION ARRAYI(NDIM, 40, 4), ARRAYO(20, 40, 4), CONFAC(4),
```

```
> NAMNOC(1), TABLE(7), CREP(20), RFACI(20),
```

```
> RFACO(20)
```

```
GO TO (10, 30, 50), IM
```

```
10 DO 20 I=1, 4
```

```
DO 20 J=1, NONCLD
```

```
DO 20 K=1, NOC
```

```
ARRAYO(K, J, I)=ARRAYI(K, J, I)*EXPP(JLOC, ILOC, J, I)*CONFAC(I)*POPFAC
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```
20 CONTINUE
   RFACO(1)=0.0
   IF(NCREP.EQ.0) GO TO 80
   DO 25 I=1, NCREP
25   RFACO(I)=RFACI(I)*WLRN(JLOC, ILOC)*POPFAC
   GO TO 80
30 DO 40 I=1, 4
   DO 40 J=1, NONCLD
   DO 40 K=1, NOC
   ARRAYO(K, J, I)=0.0DO
   DO 40 II=NOL, NOU
   DO 40 JJ=NRL, NRU
   ARRAYO(K, J, I)=ARRAYO(K, J, I)+ARRAYI(K, J, I) *EXPP(JJ, II, J, I)*
   > CONFAC(I)*TOTFAC
40 CONTINUE
   RFACO(1)=0.0
   RFACO(2)=0.0
   IF(NCREP.EQ.0) GO TO 80
   DO 45 I=1, NCREP
   DO 45 II=NOL, NOU
   DO 45 JJ=NRL, NRU
   RFACO(I)=RFACI(I)*WLRN(JJ, II)*TOTFAC+RFACO(I)
45 CONTINUE
   GO TO 80
50 DO 70 I=1, 4
   DO 70 J=1, NONCLD
   TEXPP=0.0
   DO 60 II=NOL, NOU
   DO 60 JJ=NRL, NRU
   TEXPP=TEXPP+EXPP(JJ, II, J, I)
60 CONTINUE
   DO 70 K=1, NOC
   ARRAYO(K, J, I)=ARRAYI(K, J, I)*TEXPP*CONFAC(I)
70 CONTINUE
   RFACO(1)=0.0
   IF(NCREP.EQ.0) GO TO 80
   TWLRN=0.0
   DO 65 II=NOL, NOU
   DO 65 JJ=NRL, NRU
   TWLRN=TWLRN+WLRN(JJ, II)
65 CONTINUE
   DO 75 I=1, NCREP
   RFACO(I)=RFACI(I)*TWLRN
75 CONTINUE
80 CALL DRTAB(ARRAYO, NOC, NAMNOC, TITLE, TITL1, TABLE, TITL2, NOTE,
   A NUN, RFACO, CREP, NCREP, NUCLID)
```



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

RETURN  
END

C \*\*\*\*  
C CALCS FOR NEW REPORT ADDED 1/88 BY JOAN MCGUE  
C \*\*\*\*

---

C  
C  
C SUBROUTINE CALC DEATHS PER YEAR  
C  
C

---

SUBROUTINE CALC\_DEATHS\_PER\_YEAR(OUTPUT,NOL,NOU,NRL,NRU,NONCLD,  
+ EXPP,AGEX,RISK,NCANC,RRISK,RNFLAG,WLRN)

C THIS SUBROUTINE CALCULATES THE NUMBER OF DEATHS PER YEAR FOR THE  
C NEW REPORT. THE CALUCLATIONS ARE THE SAME AS THOSE USED IN THE  
C SUBROUTINE LOCTAB FOR A POPULATION RUN. THEY MAY BE REDUNDENT  
C BUT THESE CALCULATIONS AREN'T DONE IN LOCTAB UNLESS THAT REPORT  
C IS REQUESTED FOR OUTPUT. THEREFORE, IT'S EASIER TO DO THEM HERE  
C FOR THE NEW REPORT.

INTEGER NOL,NOU,NRL,NRU,NONCLD,NCANC  
REAL RISK(20,40,4,2),FACD(4),EXPP(20,20,40,4),AGEX  
REAL OUTPUT(20,20),CMBRSK(40,4),RRISK,WLRN(20,20)  
INTEGER I,J,DIST,DIRECT  
LOGICAL RNFLAG

C\*\*\*\* NOTE FACD(4) = GSCFAC \* FACD(4,2) = 500 SEE PREPHR FOR NUMBERS  
DATA FACD/1.E-5,1.E-5,10.,500./,CMBRSK/160\*0.0/

DO 5 I = 1,4  
FACD(I) = FACD(I) / AGEX  
5 CONTINUE  
DO 10 I = 1,4  
DO 10 J = 1, NONCLD  
DO 10 K = 1,NCANC

C COMBINE THE RISKS HIGH LET AND LOW LETS.

CMBRSK(J,I) = CMBRSK(J,I) + RISK(K,J,I,1) +  
RISK(K,J,I,2)

10 CONTINUE

C SEE IF RADON NUC FLAG SET..CALCULATE WLSUM \*\*\*\*  
C

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

IF (RNFLAG) THEN
  FAC = 1./AGEX
  DO 15 DIRECT = NOL,NOU
    DO 15 DIST = NRL,NRU
      OUTPUT(DIST,DIRECT) = RRISK*WLRN(DIST,DIRECT)*FAC
15  CONTINUE
ENDIF

DO 20 I = 1,4
  DO 20 J = 1,NONCLD
    DO 20 DIRECT = NOL,NOU
      DO 20 DIST = NRL,NRU
        OUTPUT(DIST,DIRECT) = OUTPUT(DIST,DIRECT) +
          CMBRSK(J,I) * FACD(I)* EXPP(DIST,DIRECT,J,I)
20  CONTINUE

RETURN
END

```

```

C*****
C
C
C
C          UTILITY ROUTINES
C
C
C
C*****

```

```

C-----
C
C          SUBROUTINE VSORTP
C
C-----

```

```

SUBROUTINE VSORTP(A,LA,IR)
C
C  DIMENSION A(1),IU(21),IL(21),IR(1)
C
C  M=1
C  I=1
C  J=LA
C  R=.375
10 IF (I.EQ.J) GO TO 100

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

20 IF (R.GT..5898437) GO TO 30
   R=R+3.90625E-2
   GO TO 40
30 R=R-.21875
40 K=I

C                               SELECT A CENTRAL ELEMENT OF THE
C                               ARRAY AND SAVE IT IN LOCATION T

   IJ=I+(J-I)*R
   T=A(IJ)
   IT=IR(IJ)

C                               IF FIRST ELEMENT OF ARRAY IS GREATER
C                               THAN T, INTERCHANGE WITH T

   IF (A(I).LE.T) GO TO 50
   A(IJ)=A(I)
   A(I)=T
   T=A(IJ)
   IR(IJ)=IR(I)
   IR(I)=IT
   IT=IR(IJ)

50 L=J

C                               IF LAST ELEMENT OF ARRAY IS LESS THAN
C                               T, INTERCHANGE WITH T

   IF (A(J).GE.T) GO TO 70
   A(IJ)=A(J)
   A(J)=T
   T=A(IJ)
   IR(IJ)=IR(J)
   IR(J)=IT
   IT=IR(IJ)

C                               IF FIRST ELEMENT OF ARRAY IS GREATER
C                               THAN T, INTERCHANGE WITH T

   IF (A(I).LE.T) GO TO 70
   A(IJ)=A(I)
   A(I)=T
   T=A(IJ)
   IR(IJ)=IR(I)
   IR(I)=IT
   IT=IR(IJ)
   GO TO 70
60 TT=A(L)
   A(L)=A(K)
   A(K)=TT
   ITT=IR(L)
   IR(L)=IR(K)
   IR(K)=ITT

C                               FIND AN ELEMENT IN THE SECOND HALF OF

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0 1 2 3 4 5 6 7  
12345678901234567890123456789012345678901234567890123456789012

```

C                               THE ARRAY WHICH IS SMALLER THAN T
70 L=L-1
   IF (A(L).GT.T) GO TO 70
C                               FIND AN ELEMENT IN THE FIRST HALF OF
C                               THE ARRAY WHICH IS GREATER THAN T
80 K=K+1
   IF (A(K).LT.T) GO TO 80
C                               INTERCHANGE THESE ELEMENTS
   IF (K.LE.L) GO TO 60
C                               SAVE UPPER AND LOWER SUBSCRIPTS OF
C                               THE ARRAY YET TO BE SORTED
   IF (L-I.LE.J-K) GO TO 90
   IL(M)=I
   IU(M)=L
   I=K
   M=M+1
   GO TO 110
90 IL(M)=K
   IU(M)=J
   J=L
   M=M+1
   GO TO 110
C                               BEGIN AGAIN ON ANOTHER PORTION OF
C                               THE UNSORTED ARRAY
100 M=M-1
   IF (M.EQ.0) RETURN
   I=IL(M)
   J=IU(M)
110 IF (M.GT.21) WRITE(27,10000)M
   IF (J-I.GE.1) GO TO 40
   IF (I.EQ.1) GO TO 10
   I=I-1
120 I=I+1
   IF (I.EQ.J) GO TO 100
   T=A(I+1)
   IT=IR(I+1)
   IF (A(I).LE.T) GO TO 120
   K=I
130 A(K+1)=A(K)
   IR(K+1)=IR(K)
   K=K-1
   IF (T.LT.A(K)) GO TO 130
   A(K+1)=T
   IR(K+1)=IT
   GO TO 120
C

```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

10000 FORMAT(' IN VSORTP, M-',I3)  
 END

```
C-----
C
C          SUBROUTINE GET FILE NAMES
C
C-----
```

```
+ SUBROUTINE GET_FILE_NAMES (ID_CODE, STARFILE, POPFILE,
+          FILES_USED, NUMBER_FILES )
```

```
CHARACTER*8 STARFILE, POPFILE
CHARACTER*26 ID_CODE
CHARACTER*80 FILES_USED(10)
INTEGER*2 NUMBER_FILES
```

```
INTEGER*2 I, K
LOGICAL JCL_FILE
```

```
C*****          LOOK FOR TYPE OF FILE BEING REFERENCE          *****
C***** LOOK FOR THE KEY WORDS 'POP', 'JCL', 'STARFILE'          *****
C*****          THEN PARSE THRU THE FILES USED TO GET THE NAMES *****
```

```
DO 10 I = 1, NUMBER_FILES
```

```
  K = INDEX ( FILES_USED (I) , 'JCL' )
  IF ( K .NE. 0 ) THEN
    JCL_FILE = .TRUE.
    CALL EXTRACT_NAME ( JCL_FILE, ID_CODE, FILES_USED (I) )
  ENDIF
```

```
10 CONTINUE
```

```
DO 20 I = 1, NUMBER_FILES
```

```
  K = INDEX ( FILES_USED (I) , 'POP' )
  IF ( K .NE. 0 ) THEN
    JCL_FILE = .FALSE.
    CALL EXTRACT_NAME ( JCL_FILE, POPFILE, FILES_USED (I) )
    IF ( POPFILE .NE. '*****' )
```

```
+ GO TO 25
```

```
  ENDIF
```

```
20 CONTINUE
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0           1           2           3           4           5           6           7  
123456789012345678901234567890123456789012345678901234567890123456789012

```
25 DO 30 I = 1, NUMBER_FILES

    K = INDEX ( FILES_USED (I) , 'STAR' )
    IF ( K .NE. 0 ) THEN
        JCL_FILE = .FALSE.
        CALL EXTRACT_NAME ( JCL_FILE, STARFILE, FILES_USED (I) )
        IF ( STARFILE .NE. '*****' )
            + GO TO 35
    ENDIF
30 CONTINUE

35 RETURN
END
```

---

```
C
C
C                                      FUNCTION LENSTR
C
C
```

---

```
                    INTEGER FUNCTION LENSTR (STRING)
C
C   DETERMINE THE LENGTH OF THE STRING PASSED
C
C           VARIABLES PASSED
C
C                      CHARACTER STRING *(*)
C
C           LOCAL VARIABLES
C
C                      INTEGER*2 N, I
C
C                      N = LEN(STRING)
C                      DO 10 I = N, 1, -1
C                          IF ( STRING(I:I) .NE. ' ' ) THEN
C                              LENSTR = I
C                              RETURN
C                          ENDIF
10           CONTINUE
C           LENSTR = 0
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

RETURN  
 END

```
C*****
C
C
C          LOW LEVEL ROUTINTES
C
C
C
C*****
```

```
C-----
C
C          SUBROUTINE OUTPUT STRING
C
C-----
```

```
C-----
C  OUTPUT STRING WRITES THE INFO FOR THE SYNOPSIS REPORT. THE INFO      =
C  CONTAINED IN LINES IS FORMED BY INTERNAL WRITES. THIS ROUTINE WAS    =
C  MADE TO AID IN LINE COUNTING AND PAGING                               =
C  LINES --> THE CHARACTER STRINGS TO BE WRITTEN FOR THE REPORT          =
C  NUM  --> NUMBER OF LINES TO BE OUTPUT AT A TIME                      =
C  TITLE --> LOGICAL VAR, IF TRUE THE LINES TO BE OUTPUT ARE A TITLE    =
C           THE LINE NUMBER IS CHECKED TO MAKE SURE THE TITLE WILL      =
C           NOT BE SPLIT UP ( IE BOTTOM AND TOP OF A PAGE )              =
C  REPRINT_TITLE --> LOGICAL VAR, IF TRUE THE LAST TITLE OUTPUT         =
C                   SAVED IN THE STRING SAVED_TITLE_LINES, IS OUTPUT-   =
C                   AT THE TOP OF A NEW PAGE.                             =
C
C  BY USING THESE VARS, TITLES CAN BE REPRINTED OR NOT REPRINTED. IT'S- =
C  UP TO YOU.                                                             =
C-----
```

```
C  SUBROUTINE OUTPUT_STRING ( LINES, NUM, TITLE, REPRINT_TITLE )
```

```
C
C  LOGICAL TITLE, REPRINT_TITLE
C
C  CHARACTER*90 LINES (20)
C
C  INTEGER*2 I
C  INTEGER  NUM
```

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

INTEGER\*2 PAGE\_NO, LINE\_COUNT, NUM\_TITLE\_LINES  
COMMON / COUNTERS / PAGE\_NO, LINE\_COUNT, NUM\_TITLE\_LINES

CHARACTER\*90 SAVED\_TITLE\_LINES (20)  
CHARACTER\*26 ID\_CODE  
COMMON / TITLES / SAVED\_TITLE\_LINES, ID\_CODE

CHARACTER\*36 DATE\_AND\_TIME  
COMMON / HEADERINFO / DATE\_AND\_TIME

IF ( TITLE ) THEN

C SAVE TITLE FOR FUTURE IF PAGE EJECT  
DO 10 I = 1, NUM  
    SAVED\_TITLE\_LINES (I) = LINES (I)  
10 CONTINUE  
    NUM\_TITLE\_LINES = NUM

I = LINE\_COUNT + NUM  
C CHECK FOR TITLE BEING SPLIT UP

IF ( I .GE. 60 ) THEN  
C NEW PAGE  
    PAGE\_NO = PAGE\_NO + 1  
    WRITE (12, 1000) ID\_CODE, DATE\_AND\_TIME, PAGE\_NO  
1000 FORMAT ('1', / 1X, 'ID CODE: ' ,A, 1X, 'DATE/TIME:', A, 1X,  
    + 'PAGE ', I1 /)  
    LINE\_COUNT = 3  
    ENDIF  
ENDIF

DO 20 I = 1, NUM

IF ( LINE\_COUNT .GT. 60 ) THEN  
C NEW PAGE  
    PAGE\_NO = PAGE\_NO + 1  
    WRITE (12, 1000) ID\_CODE, DATE\_AND\_TIME, PAGE\_NO  
    LINE\_COUNT = 3

IF ( REPRINT\_TITLE ) THEN  
C OUTPUT TITLE ON NEW PAGE



DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```

DO 30 J = 1, NUM_TITLE_LINES
  WRITE (12, 1010 ) SAVED_TITLE_LINES (J)
  LINE_COUNT = LINE_COUNT + 1
30  CONTINUE

  ENDIF

  ENDIF

  WRITE (12, 1010 ) LINES (I)
1010 FORMAT ( A )
  LINE_COUNT = LINE_COUNT + 1

20 CONTINUE

RETURN
END

```

---

```

C
C
C           SUBROUTINE EXTRACT NAME
C
C THE FOLLOWING SUBROUTINE PARSSES THRU THE GIVEN LINE TO FIND THE
C MEMBER NAME OF THE POPULATION FILE, THE WIND FILE OR THE JCL FILE
C NAME. ***** IF ITS THE JCL FILE REFERENCE LINE
C THE ID CODE IS EXTRACTED. THE ID CODE CONSISTS OF THE UNIQUE
C PARTITIONED DATA SET NAME FOR THE SOURCE CATEGORY FOLLOWED BY AN
C UNDERSCORE ' _ ' FOLLOWED BY THE MEMBER NAME OF THE JCL
C AN ERROR IS INDICATED BY BOTH THE SOURCE AND MEMBER NAME
C '*****'. FOR INSTANCE IF CAA88 IS NOT FOUND THIS IS AN ERROR.
C ***THIS CODE IS BUILT ON THE ASSUMPTION THAT THE FILE NAMES MUST BE
C IN THE FOLLOWING FORMAT 'CAA88.SOURCENAME(MEMBER)'.
C

```

---

SUBROUTINE EXTRACT\_NAME ( JCL\_FILE, FNAME, LINE )

CHARACTER FNAME \*(\*)  
 CHARACTER\*80 LINE  
 LOGICAL JCL\_FILE

INTEGER\*2 J, K  
 INTEGER\*2 LEN\_SRCE

CHARACTER\* 17 SOURCE

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

CHARACTER\* 8 MEMBER

DATA SOURCE '/'                                    '/'

C\*\* BEGIN

MEMBER = '            '

K = INDEX ( LINE, 'CAA88.' )

C\*\* IF 'CAA88' IS NOT FOUND IT IS AN ERROR

IF ( K .EQ. 0 ) THEN

IF ( JCL\_FILE ) THEN

SOURCE = '\*\*\*\*\*'

ENDIF

MEMBER = '\*\*\*\*\*'

ELSE

C\*\* FIND END OF SOURCE CATEGORY NAME

J = INDEX ( LINE , '(' )

C\*\* NO LEFT PAREN INDICATES ERROR IN MEMBER NAME

IF ( J .EQ. 0 ) THEN

IF ( JCL\_FILE ) THEN

SOURCE = '\*\*\*\*\*'

ENDIF

MEMBER = '\*\*\*\*\*'

ELSE

C\*\*\*\* EXTRACT SOURCE CAT FOR ID CODE IF LINE IS FOR JCL FILE

IF ( JCL\_FILE ) THEN

K = K + 6

LENGTH = J - K

IF ( LENGTH .GT. 17 ) THEN

J = K + 17

LENGTH = 17

ENDIF

SOURCE (1: LENGTH) = LINE ( K : J-1 )

ENDIF

K = INDEX ( LINE, ')' )

C\*\* NO RIGHT PAREN INDICATES ERROR IN MEMBER NAME

IF ( K .EQ. 0 ) THEN

MEMBER = '\*\*\*\*\*'

ELSE

DARTAB (DARTAB2.FOR) Program File  
(continued)

0            1            2            3            4            5            6            7  
123456789012345678901234567890123456789012345678901234567890123456789012

```
      LENGTH = K - J - 1
      IF ( LENGTH .GT. 8 ) THEN
        K = J + 9
        LENGTH = 8
      ENDIF
      MEMBER (1:LENGTH) = LINE ( J+1:K-1 )
    ENDIF
```

```
  ENDIF
```

```
ENDIF
```

```
C**  IF JCL FILE MAKE THE ID CODE
      IF ( JCL_FILE ) THEN
        LEN_SRCE = LENSTR ( SOURCE )
        FNAME (1:LEN_SRCE) = SOURCE (1:LEN_SRCE)
        FNAME (LEN_SRCE+1:LEN_SRCE+1) = '_'
        FNAME (LEN_SRCE+2:LEN_SRCE+9) = MEMBER
      ELSE
        FNAME ( 1:8 ) = MEMBER
      ENDIF

      RETURN
    END
```