Distribution
Category UC-722

SAND89-2370
TTC-0943
Unlimited Release
July 1992

SAND--89-2370-Vol.4

DE92 018959

RADTRAN 4
VOLUME IV: PROGRAMMER'S MANUAL

F. L. Kanipe
GRAM, Inc.
Albuquerque, New Mexico 87112

and

K. S. Neuhauser
Risk Assessment and Transportation Department 6641
Sandia National Laboratories
Albuquerque, New Mexico 87185

ABSTRACT

The RADTRAN 4 computer code is designed to analyze radiological consequences and accident risks of transporting radioactive material. This manual provides information useful for interpreting, troubleshooting, or debugging components of the code during development or revision of the program.

FOREWORD

RADTRAN 4 has been developed by and is maintained in the custody of Sandia National Laboratories (SNL) for the U.S. Department of Energy (DOE). The name RADTRAN 4 is used in this document to refer to the latest release of the computer code. The name RADTRAN is used here to refer to all releases of the code.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BDF	Building dose factor
DEC	Digital Equipment Corporation
LCF	Latent cancer fatalities
MIS	Manager of Information Systems
NRC	U.S. Nuclear Regulatory Commission
PPS	Packages per shipment
SNL	Sandia National Laboratories
TI	Transport Index

LIST OF ABBREVIATIONS

Ci	Curie
hr	Hour
km	Kilometer
m	Meter
MeV	Mega electron volt
min	Minutes
rem	Roentgen-equivalent-man
sec	Second
yr	Year

1.0 INTRODUCTION

RADTRAN 4 is a computer code that analyzes the radiological impact of transporting radioactive material. The first version of RADTRAN was developed by Sandia National Laboratories (SNL) under contract with the U.S. Nuclear Regulatory Commission (NRC) to prepare the "Final Environmental Statement on the Transportation of Radiological Material by Air and Other Modes" (NRC, 1977). RADTRAN II (Taylor and Daniel, 1982), which followed, incorporated a detailed urban model (Finley et al., 1980). RADTRAN III (Madsen et al. 1983) included model improvements and two corrections. RADTRAN 4 (Neuhauser and Kanipe, 1992a) is the most recent version of the code; it includes model improvements, as well as greatly expanded analytical capabilities.

RADTRAN 4 combines user-defined meteorological, demographic, transportation, packaging, and material factors with health physics data to calculate the expected radiological consequences and risks resulting from the shipment of radioactive material. The code examines all modes of commercial transport (i.e., by air, truck, rail, and water). The code estimates the radiological carriage effect of transport of radioactive material under normal (incident-free) conditions and under accident conditions (Figure 1-1). Regulatory compliance is also assessed in terms of maximum permissible dose rates to the population along the transport route, crew and worker dose rates, and exclusive shipment criteria. Figure 1-2 shows the nested computational structure of RADTRAN 4.

Since its introduction, RADTRAN has been used worldwide in the evaluation of proposed shipments of radioactive material. The radiological risk calculations performed by RADTRAN 4 have a variety of applications. RADTRAN 4 can be used as a tool in the preparation of environmental impact statements and in the assessment of the relative risks of different transportation and packaging scenarios. RADTRAN has been available to the public for approximately 13 years, and RADTRAN 4 is accessible via TRANSNET (Cashwell, 1989).

When a more general analysis of a shipment is desired or when no additional data are available, RADTRAN 4 provides data assistance with default settings, including an isotope library that provides 11 variables which partially characterize each constituent isotope of the materials to be shipped. Data regarding the number of curies of each isotope and other factors must be input by the user regardless of whether the isotope library is used. Any shipment parameter, however, can be altered by the user.

RADTRAN 4 is written in FORTRAN 77 and is currently run on a DEC VAX computer at SNL. The code can be made available to the user in VAX-executable form on tape or DEC RX-50 diskette, and a hard-copy printout of the code can be made available.

This manual provides useful information for interpreting, troubleshooting, and debugging components of the code for development or revision. Chapter 2.0 describes the code/system interface including computer system resources, security, data files, math libraries, and compiler and hardware requirements. A printout of the RADTRAN 4 computer

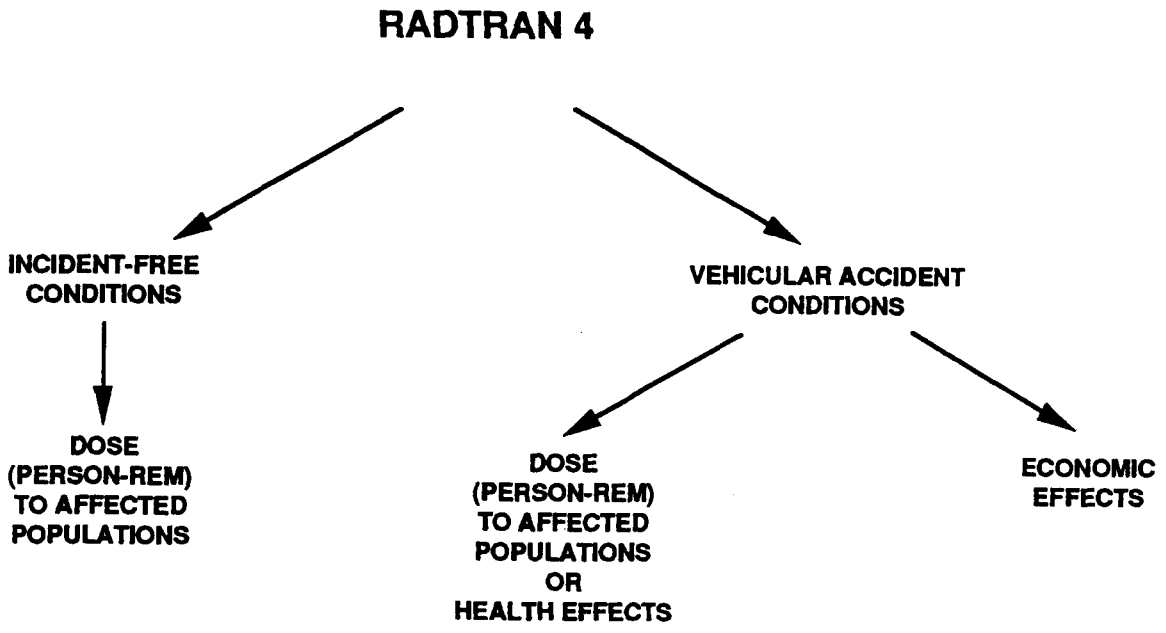
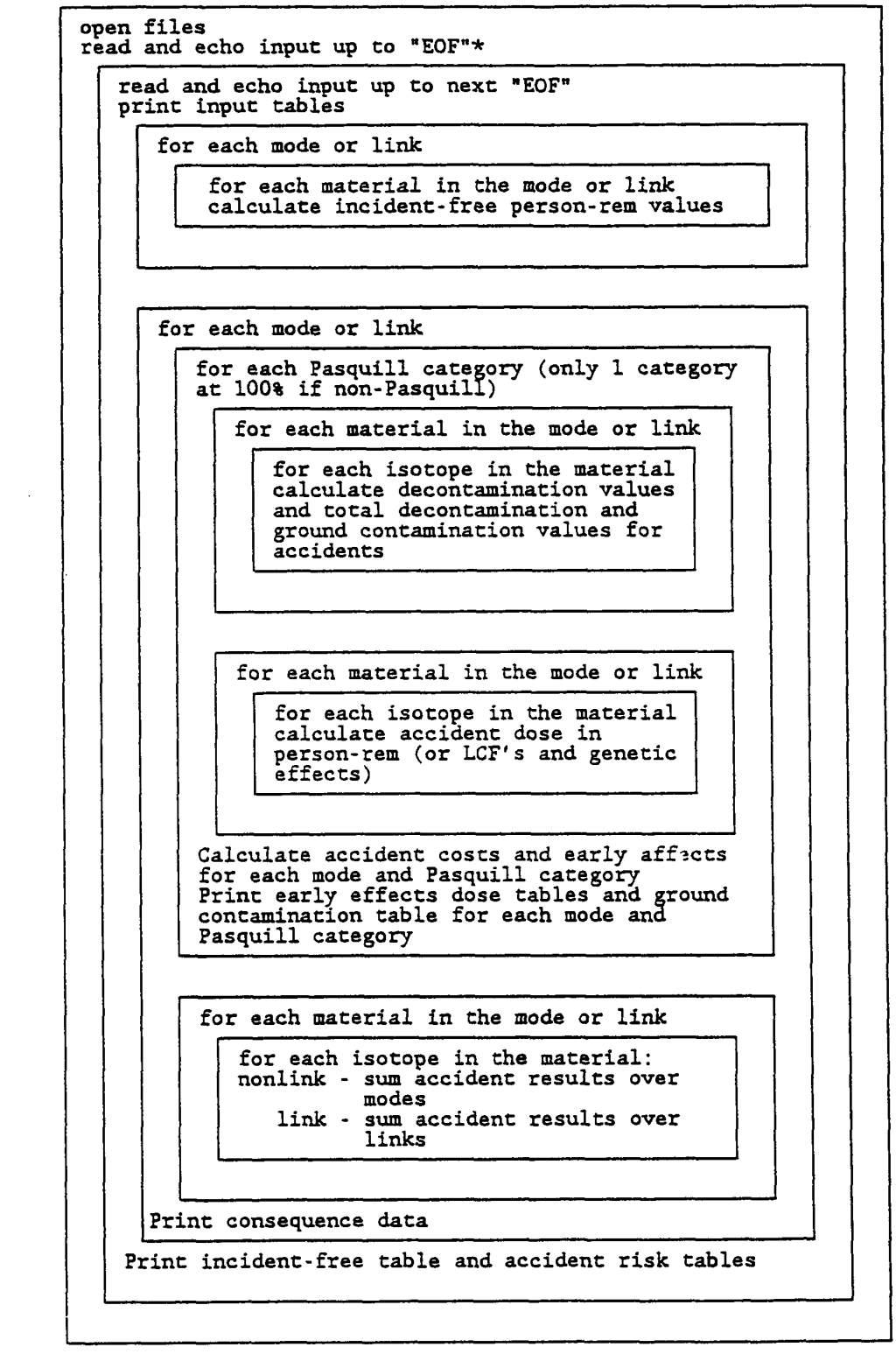


Figure 1-1. RADTRAN 4 Calculations of Incident-Free and Accident Conditions in the Transport of Radioactive Material

code is provided in Chapter 3.0. Chapter 4.0 contains variable and subroutine glossaries. Chapter 5.0 provides samples of data input files and RADTRAN 4 output and verification documentation procedures. Error reporting and processing limitations are described in Chapter 6.0, including reporting procedures for errors or newly discovered limitations of the code, updates and new releases, documentation updates, and guidelines for formal inquiry response. Chapter 7.0 provides RADTRAN 4 documentation and other sources to acquire additional information not provided in this manual.



*End-of-file input code

Figure 1-2. Computational Sequence

2.0 CODE AND SYSTEM INTERFACE

The RADTRAN 4 computer code runs on a Digital Equipment Corporation (DEC) MicroVAX 3800 located at SNL in Albuquerque, New Mexico. The following sections describe details of this system and its operation.

2.1 RADTRAN 4 System and Operations

RADTRAN 4 is compiled and run on a DEC MicroVAX 3800. This VAX is also known as TRANSNET (Cashwell, 1989). TRANSNET users have access to the most recent developmental version of RADTRAN at all times, but have restricted accounts which limit their possible tasks to a set of menu choices. The operating system is DEC VMS, Version 5.4.

For the purposes of quality control, direct access to the RADTRAN source code is restricted to the RADTRAN technical programmer and the VAX system manager. The RADTRAN technical programmer implements models developed by SNL staff and makes all corrections and other required changes. All code revisions are introduced with new revision numbers (e.g., RADTRAN 4.0.1 and RADTRAN 4.0.2 were two developmental versions of RADTRAN 4); each code revision is fully documented in the RADTRAN technical programmer's logbook.

After a revision has been completed and documented, the technical programmer makes that version available to TRANSNET users. A list of TRANSNET users can be obtained from the TRANSNET system manager. The changes are also documented separately under "Revision history/quality record," which is a menu selection for the TRANSNET user. This procedure ensures that the most recent developmental revisions are promptly available to all users. The operating system controls user identification and passwords. See the VMS operating system manuals, Version 5.4, for additional information on user identifications, passwords, file locking, control programs, and transaction log (DEC, 1989a; DEC, 1989b; DEC, 1989c).

2.2 Files Required for Execution

There are three external files that must be available for RADTRAN 4 to run correctly. The file containing the isotope library data is named RT4ISO.DAT, and the file containing other default data is named RT4INT.DAT. These two files are read in upon execution of RADTRAN 4. They have a separate varification and change procedure than that of RADTRAN 4 so that the RADTRAN version need not be changed in order to change a default value. The third file read in at execution time is FILE2.DAT, which contains data that is assembled by the user (see Neuhauser and Kanipe, 1992).

2.3 System-Dependent Features

RADTRAN 4 incorporates some of the SNL SLATEC mathematical library routines (Fong et al., 1982; Fong et al., in preparation). The SLATEC routines used in RADTRAN 4 are listed and defined in Table 2-1. The SLATEC routines I1MACH and R1MACH must have the system-dependent machine constants set before compilation. This is usually a matter of altering the constants set by removing comment markers from only those constants that apply to the particular machine being used.

Table 2-1

The SLATEC Subroutines and Functions in RADTRAN 4

Function or Subroutine	Definition
AVINT	Integrates a function tabulated at arbitrarily spaced abscissas.
BDIFF	Computes a binomial sum.
BESIO	Computes a hyperbolic Bessel function of the first kind for order zero.
BESIOE	Computes an exponentially scaled hyperbolic Bessel function of the first kind of order zero.
BESI1	Computes a modified hyperbolic Bessel function of the first kind of order one.
BESI1E	Computes an exponentially scaled hyperbolic Bessel function of the first kind of order one.
BESKO	Computes a hyperbolic Bessel function of the third kind of order zero.
BESKOE	Computes an exponentially scaled hyperbolic Bessel function of third kind of order zero.
BESK1	Computes a hyperbolic Bessel function of the third kind of order one.
BESK1E	Computes an exponentially scaled hyperbolic Bessel function of the third kind of order one.
BKIAS	Computes the repeated integrals of the K-phi Bessel function of the asymptotic expansion.
BKISR	Computes the repeated integrals of the K-phi Bessel function by the series for N = 0 to 1 and 2.
BSKIN	Computes the Bickley functions.
CSEVL	Evaluates the N-term Chebyshev series.
E1	Calculates the single-precision exponential integral.
EXINT	Computes the member sequences of exponential integrals.
GAMRN	Computes the gamma function ratio gamma (x)/gamma (x+.5).

Table 2-1

The SLATEC Subroutines and Functions in RADTRAN 4 (Concluded)

Function or Subroutine	Definition
GAUS8	Integrates real functions of one variable over finite intervals.
HKSEQ	Computes the derivatives of the PSI function.
I1MACH	Returns the integer machine-dependent parameters.
INITS	Initializes an orthogonal series so that it defines the number of terms to carry in the series to meet a specified error.
PSIXN	Returns the values of PSI(x) equals derivative of log gamma (x).
R1MACH	Returns the real machine-dependent parameters.
SSORT	Sorts an array.
XERR	Prints an error message and stops the run. This routine is a replacement for the longer XERROR routine.

To run RADTRAN on any another system, the directory location that RADTRAN 4 searches for input files RT4INT.DAT and RT4ISO.DAT must be changed. Since the system may not be familar to the programmer who is compiling RADTRAN, for a tape version it is best to assume that the current default directory will be the location of these files. This assumption must be explained to users who receive an executable version so that they keep these files in the same directory as the RADTRAN executable code.

RADTRAN 4 uses the VAX VMS FORTRAN system called DATE and TIME. These also would need to be changed on another operating system.

2.4 Compiler Requirements

The VAX/VMS FORTRAN compiler, Version 5.3, is employed to compile RADTRAN 4 at SNL. The code is written in ANSI FORTRAN 77. The VMS system subroutines DATE and TIME are called within the code; these calls would have to be changed if RADTRAN 4 is compiled on another operating system.

2.5 Hardware Requirements

RADTRAN 4 can be run without alterations on any VAX/VMS system capable of running FORTRAN, Version 5.3.

2.6 Files and Directories

To maintain the integrity of the RADTRAN 4 source code, access is restricted. Therefore, file and directory names are only available to the technical programmer and VAX system manager.

2.7 Compiling and Running

The RADTRAN code is compiled with the VAX/VMS Fortran option CHECK=BOUND. This option will stop execution if an array subscript is out of bounds and a system error message will be printed stating such.

Upon execution, RADTRAN searches the current directory for an input file named FILE2.DAT. All other files are output files that are created by RADTRAN.

3.0 ANNOTATED CODE LISTINGS

This chapter contains a printout of the RADTRAN 4 source code. Numerous comments are provided throughout the code to facilitate working with the code, including reminders about assumptions, explanations of subtleties and interdependencies, and useful cross-references.

Table 3-1 gives the order of appearance and the number of lines of each subroutine in the printout of the code.

Table 3-1

Subroutines In Order of Occurrence

Subroutine	Number of Lines
ACC	288
ARRAY	29
BANNER	57
BESSL	89
CENTER	30
CONVRT	134
COST	208
CPBYTE	17
EARLY	175
ECON	118
EFFIT	16
FINDI	31
GOTOER	6
INITLZ	125
IARRAY	27
MDT	69
NEWPAG	24
NEXT	39
NORLNK	125
NORMAL	318
PAIRS	172
PASQ	46
PRCHRD	93
PRCONS	133
PRISTP	183
PRNORD	32
PROUTA	122
PROUTB	168
PROUTC	78
PROUTD	52
PRRELD	177
PRSEGD	87
PRSEND	37
QUAD	21
RADTN4	386
RDINFR	726
RDDAT	284
SCAN	48

Table 3-1

Subroutines In Order of Occurrence
(Concluded)

Subroutine	Number of Lines
SCRIPT	67
SENSTV	417
SENSUM	208
SETISO	56
SETMAT	57
SETMOD	177
SETVAL	97
SHIELD	19
SHIFTC	10
SHIFTL	10
TRANS	14
WTRNK	105
AVINT (Beginning of the SLATEC Math Subroutines)	162
BDIFF	26
BESIO	57
BESIOE	119
BESI1	61
BESI1E	123
BESKO	60
BESKOE	104
BESK1	62
BESKIE	107
BKISR	72
BKLAS	243
BSKIN	340
CSEVL	45
E1	284
EXINT	292
GAMRN	91
HKSEQ	138
I1MACH	558
INITS	40
PSIXN	104
R1MACH	290
SSORT	252
XERR	6

```
*DECK,ACC

      SUBROUTINE ACC
C
C     THIS ROUTINE CALCULATES THE EXPECTED NUMBER OF ACCIDENTS
C     AND THEIR CONSEQUENCES.
C     THE CALCULATIONS INCLUDE THE EXPECTED NUMBER
C     OF PEOPLE EXPOSED TO DIFFERENT DOSE LEVELS, AND THE
C     RESULTING PERSON-REM DOSE OR EXPECTED NUMBERS OF LATENT
C     CANCER FATALITIES AND GENETIC EFFECTS
C
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      COMMON /PROB/ PROBEF(30,4),REMLEV(30),NE,NREMLV
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
      COMMON /CON/ TOL,NDERV,NA,NC,TWOPI,PI,Q6,WS,GAM,FMU,DZ,VV
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
```

```
C
      COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
      REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
      LOGICAL RSPEC
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
      REAL GRDPRM(20),EXPOSR(3)
      GO TO (10,150), IACC
      CALL GOTOER ('ACC-2')
   10 CONTINUE
C     - COMPUTE DOSE VS. DISTANCE FOR SPECIFIED RADIAL DISTANCES
C ... EQUATION 53 IN RADTRAN III GUIDE
      CONVAL = Q6*CI*PPS*PHTE
      DO 20 I = 1, NPOP
        DELT = DELTIM(I)
        DO 20 J = 1, NSEV
          RF = RFRAC(IGRP,J)
          ANUM = CONVAL*DELT*RF
          DO 20 K = 1, NRAD
            RD = RADIST(K,I)
            DOSDST(K,J,I) = ANUM/(RD*RD)
            TDSDST(K,J,I) = TDSDST(K,J,I)+DOSDST(K,J,I)
   20 CONTINUE
C     COMPUTE INTEGRATED EXPOSURE FOR EACH POPULATION ZONE AND SEVERITY
      CON = CI*PPS*PHTE
      RAD1 = RADIST(1,1)
      RADN = RADIST(NRAD,1)
      RATU1 = ALOG((RAD1+3.)/RAD1)
      RATU2 = RPD*RATU1+RU*ALOG(RADN/(RAD1+3.))
      RATRS = ALOG(RADN/RAD1)
C ... EQ 64 IN RADTRNA III
      CONR = 6.28E-6*CON*POPDR*RATRS*RR
C ... EQ 65 IN RADTRNA III
      CONS = 6.28E-6*CON*POPDS*RATRS*RS
      GO TO (60,70,80), IUOPT
      CALL GOTOER ('ACC-3')
C ... EQ 66 IN RADTRAN III
   60 CONU = 3.14E-6*CON*POPDU*RPD*RATU1
      GO TO 90
   70 CONU = 3.14E-6*CON*POPDU*RATU2
      GO TO 90
   80 CONU = 3.14E-6*CON*POPDU*RATRS
   90 CONTINUE
C     COMPUTE NUMBER OF LCF S AND GE S PER ACCIDENT FOR EACH
C     POPULATION ZONE AND SEVERITY
      DO 120 J = 1, NSEV
C ..... EXPOSR VALUES ARE FINAL RESULTS OF EQUATIONS 64,65,66
```

```
          EXPOSR(1) = CONR*RFRAC(IGRP,J)
          EXPOSR(2) = CONS*RFRAC(IGRP,J)
          EXPOSR(3) = CONU*RFRAC(IGRP,J)
          INGLCF(J) = 0.0
          INGGE(J) = 0.0
          DO 120 I = 1, NPOP
            DNHLCF(J,I) = WBLCF*EXPOSR(I)
            ACCLCF(J,I) = WBLCF*EXPOSR(I)
            DNHGE(J,I) = GNDSGE*EXPOSR(I)
            ACCGE(J,I) = GNDSGE*EXPOSR(I)
            GRDLCF(J,I) = 0.0
            RDFLCF(J,I) = 0.0
            CLDLCF(J,I) = 0.0
            GRDGE(J,I) = 0.0
            RDFGE(J,I) = 0.0
            CLDGE(J,I) = 0.0
  120   CONTINUE
        GO TO 310
  150   CONTINUE
C
C       DISPERSAL ACCIDENT CALCULATIONS --
C       COMPUTE DOSE VS. AREA FOR SPECIFIED ANNULAR ISODOSE AREAS FOR
C       EACH ORGAN OF INTEREST FOR EACH POPULATION ZONE AND SEVERITY
C
        XLOW = 0.
        CALL AVINT (AREADA,CHIVAL(1,LIB),NAVINT(LIB),XLOW,
       *            AREADA(NAVINT(LIB)),ANS,IERR)
        ENHFAC = ANS*BRATE*1.E-6
C       COMPUTE INHALATION DOSE IN REM FOR EARLY EFFECTS CALCULATION
C . . EQ 76 IN RADTRAN III GUIDE
        CON = CI*PPS*BRATE
        DO 170 I = 1, NAREAS
          DO 170 J = 1, NSEV
            TERM = CHIVAL(I,LIB)*RFRAC(IGRP,J)*RESP(IDSP,J)*AERSOL(IDSP,J)
C . . . . SUMMED OVER LUNG TYPES EARLYL IS EQ 76 FOR 1 YR LUNG (ORGAN 1)
            EARLYL(I,J,LNGTYP) = EARLYL(I,J,LNGTYP)+CON*TERM*RPCVAL(LIB,1)
            N = 2
            IF (.NOT.URSKF) N = NTHRS
            DO 170 K = 1, N
C . . . . . EARLYD IS EQ 76 FOR ORGANS 1 - NTHRS (OR JUST 1 AND 2 IF UNIT)
              EARLYD(I,J,K) = EARLYD(I,J,K) + CON*TERM*RPCVAL(LIB,K)
  170   CONTINUE
C       COMPUTE THE NUMBER OF LCF S DUE TO DIRECT INHALATION FOR
C       POPULATION ZONE AND SEVERITY
        CON = CI*PPS*ENHFAC
        DO 200 K = 1, NPOP1
          POPVAL = POPDEN(K)
          DO 200 J = 1, NSEV
            POPRF = POPVAL*RFRAC(IGRP,J)*AERSOL(IDSP,J)*RESP(IDSP,J)
            SUM = 0.
C           SUM OVER ORGANS 4-10
```

```
          DO 180 L = 4, NORG
            SUM = SUM+CON*POPRF*RPCVAL(LIB,L)*EXPLCF(L-3)
  180     CONTINUE
          IF (K.EQ.3) SUM = SUM*BDF*0.52
          IF (K.EQ.NPOP1) GO TO 190
          DNHLCF(J,K) = SUM
C         COMPUTE LCF S DUE TO RESUSPENSION
          RDFLCF(J,K) = SUM*(RDF-1.)
          GO TO 200
  190     CONTINUE
          DNHLCF(J,NPOP) = DNHLCF(J,NPOP)+SUM
          RDFLCF(J,NPOP) = RDFLCF(J,NPOP)+SUM*(RDF-1.)
  200 CONTINUE
C     COMPUTE GE DUE TO DIRECT INHALATION BY POPULATION ZONE
C     AND SEVERITY
      CON = CI*PPS*ENHFAC*RPCVAL(LIB,3)*GNDSGE
      DO 220 J = 1, NSEV
        RF = RFRAC(IGRP,J)*AERSOL(IDSP,J)*RESP(IDSP,J)
        DO 220 K = 1, NPOP1
          AA = CON*RF*POPDEN(K)
          IF (K.EQ.NPOP1) GO TO 210
          IF (K.EQ.3) AA = AA*BDF*0.52
          DNHGE(J,K) = AA
C         COMPUTE GE DUE TO RESUSPENSION BY POPULATION ZONE AND SEVERITY
          RDFGE(J,K) = AA*(RDF-1.)
          GO TO 220
  210     DNHGE(J,NPOP) = DNHGE(J,NPOP)+AA
          RDFGE(J,NPOP) = RDFGE(J,NPOP)+AA*(RDF-1.)
  220 CONTINUE
C
C     COMPUTE INTEGRATED GROUNDSHINE DOSE
C ... THIS COMPUTES EQ 101 FROM RADTRAN III GUIDE **EXCEPT LAMDA5 AND
C     LAMDA6 OF RADTRAN III GUIDE ARE WRONG** SEE RADTRAN II GUIDE
C . . . THETA  = -LAMDA2
C . . . PHI    = -LAMDA4
C . . . ATHETP = LAMDA1
C . . . TRMZ   = e**(-LAMDA7)
      RHLIFE = 1./HLIFE
      THETAP = .0031*HLIFE+.693
      THETA = -RHLIFE*THETAP
      PHIP = 2.1E-5*HLIFE+.693
      PHI = -RHLIFE*PHIP
      ATHETP = .63/THETAP
      TRMZ = 0.0
      IF (PHI.GT.(-.004)) TRMZ = EXP(1.83E4*PHI)
      TRM1 = ATHETP*(1.-EXP(THETA))+(.37/PHIP)*(1.-EXP(PHI))
      TRM2 = ATHETP*EXP(10.*THETA)+(.37/PHIP)*(EXP(10.*PHI)-TRMZ)
      DO 240 J = 1, NSEV
        GRDPRM(J) = 0.0
        IF (DEPVEL.NE.0.0) THEN
          DO 230 K = 1, NAREAS
```

```
              CON = 3.04E-4*PHTE*HLIFE*CULVL*1.E-6*ASIZE(K)*DECON(K,J,ISO)
              IF (TDCONF(K,J).LE.1.0) THEN
                GRDPRM(J) = GRDPRM(J) + CON*(TRM1+TRM2)
              ELSEIF (TDCONF(K,J).LE.40.0) THEN
                GRDPRM(J) = GRDPRM(J) + CON*(TRM1+1/TDCONF(K,J)*TRM2)
              ELSE
                GRDPRM(J) = GRDPRM(J) + CON*TRM1
              ENDIF
  230       CONTINUE
          ENDIF
  240   CONTINUE
C       COMPUTE GROUNDSHINE LCF S AND GE S BY SEVERITY AND
C       POPULATION ZONE
        DO 250 I = 1, NPOP
          PD = POPDEN(I)
          IF (I.EQ.NPOP) PD = PD*(.52*BDF+.1*RPD)
          DO 250 J = 1, NSEV
            GRDLCF(J,I) = GRDPRM(J)*PD*WBLCF
            GRDGE(J,I) = GRDPRM(J)*PD*GNDSGE
  250   CONTINUE
C
C       COMPUTE FOOD INGESTION
        DO 280 J = 1, NSEV
          INGLCF(J) = 0.0
          INGGE(J) = 0.0
          DO 280 K = 1, NAREAS
            IF (TDCONF(K,J).GT.40.0) GO TO 280
            CIDF = 0.0
            DO 260 L = 4, NORG
              CIDF = CIDF+INGVAL(LIB,L)*EXPLCF(L-3)
  260       CONTINUE
            GIDF = INGVAL(LIB,3)*GNDSGE
            CON = 1.E-6*ASIZE(K)*(FDTRAN(LIB)+SLTRAN(LIB))*
     *            CULVL*DECON(K,J,ISO)
            IF (TDCONF(K,J).LE.1.0) THEN
              INGLCF(J) = INGLCF(J)+CON*CIDF
              INGGE(J) = INGGE(J)+CON*GIDF
            ELSEIF (TDCONF(K,J).LE.40.0) THEN
              INGLCF(J) = INGLCF(J)+CON*CIDF/TDCONF(K,J)
              INGGE(J) = INGGE(J)+CON*GIDF/TDCONF(K,J)
            ENDIF
  280   CONTINUE
C
C       COMPUTE INTEGRATED CLOUDSHINE DOSE
        CLDCON = CI*PPS*CDF*(ENHFAC/BRATE)
        RCON = POPDR*CLDCON
        SCON = POPDS*CLDCON
        UCON = POPDU*CLDCON*(.52*BDF+.1*RPD)
        DO 290 J = 1, NSEV
          RA = RFRAC(IGRP,J)*AERSOL(IDSP,J)
          CR = RA*RCON
```

```
        CS = RA*SCON
        CU = RA*UCON
        CLDLCF(J,1) = CR*WBLCF
        CLDLCF(J,2) = CS*WBLCF
        CLDLCF(J,3) = CU*WBLCF
        CLDGE(J,1) = CR*GNDSGE
        CLDGE(J,2) = CS*GNDSGE
        CLDGE(J,3) = CU*GNDSGE
  290 CONTINUE
  310 CONTINUE
      DO 300 I = 1, NPOP
        DO 300 J = 1, NSEV
C . . . . TOTAL LCFS SUMMED OVER ISOTOPES
          DTOT(J,I) = DTOT(J,I)+DNHLCF(J,I)+RDFLCF(J,I)+GRDLCF(J,I)+
     *                CLDLCF(J,I)
          IF (I.EQ.1) DTOT(J,I) = DTOT(J,I)+INGLCF(J)
  300 CONTINUE
C     COMPUTE TOTAL LCF S AND GE S BY SEVERITY AND POPULATION ZONE
C     ALSO -- EXPECTED NUMBER OF ACCIDENTS PER YEAR
      DO 320 I = 1, NPOP
        DO 320 J = 1, NSEV
          ACCLCF(J,I) = GRDLCF(J,I)+DNHLCF(J,I)+RDFLCF(J,I)+CLDLCF(J,I)
          ACCGE(J,I) = GRDGE(J,I)+DNHGE(J,I)+RDFGE(J,I)+CLDGE(J,I)
          LCFPYR(J,I) = ACCLCF(J,I)*ACCPYR(J,I)
          GEPYR(J,I) = ACCGE(J,I)*ACCPYR(J,I)
  320 CONTINUE
      RETURN
  340 FORMAT (////35H AVINT INTEGRATION RETURNED ERROR =,I2,21H FOR FOLL
     *OWING DATA--/(5X,2E10.3))
      END
*DECK,ARRAY
      SUBROUTINE ARRAY (A,N)
C
C     THIS ROUTINE READS "N" VALUES INTO FLOATING POINT ARRAY, "A"
C
      COMMON /CARD/ IPOINT,KBYTE,CONTUF,NRFLAG,NCS
      LOGICAL CONTUF,NRFLAG
      COMMON /CCARD/ ICARD,TEST,ICARDI,NVY(10)
      CHARACTER*10 ICARD*80, TEST,ICARDI,NVY*1
C
C
      DIMENSION A(N)
C
      NRS = 0
      NR = 0
      DO 20 I = 1, N
        NR = NR+1
        IF (NR.LE.NRS) GO TO 10
        NR = 1
        NRS = 0
        VAL = VALUE(DMY)
```

```
          IF (.NOT.NRFLAG) GO TO 10
          NRS = VAL+.01
          VAL = VALUE(DMY)
   10     A(I) = VAL
   20   CONTINUE
C
        RETURN
        END
*DECK,BANNER
        SUBROUTINE BANNER (MESSAG)
        CHARACTER WORD*12, MESSAG*45, BLANK*12
        DIMENSION WORD(10)
        DATA BLANK / ' '/
        II = 0
        JJ = 0
        KK = INDEX(MESSAG(1:45),'$')
        DO 30 K = 1, 10
          WORD(K) = BLANK
          JJ = JJ+II
          II = INDEX(MESSAG(JJ+1:KK),' ')
          IF (II.NE.0) THEN
            NL = II-1
          ELSE
            NL = KK-JJ-1
          ENDIF
          NB = (11-NL)/2
          DO 10 J = 1, NL
            WORD(K)(NB+J:NB+J) = MESSAG(JJ+J:JJ+J)
   10     CONTINUE
          WORD(K)(NB+NL+1:NB+NL+1) = '$'
          IF (II.EQ.0) THEN
            DO 20 J = 1, K
              CALL SCRIPT (WORD(J))
              WRITE (4,'(///)')
   20       CONTINUE
            WRITE (4,'(16X,A46,///)')
     *           'RADTRAN 4.0.12  VERSION DATE: NOVEMBER 19,1991'
            WRITE (4,'(15X,A30,/)') '            MODE DESCRIPTIONS'
            WRITE (4,'(15X,A10,A10,A25)') 'NUMBER ','NAME     ',
     *        'CHARACTERIZATION  '
            WRITE (4,'(15X,A10,A10,A25)') '1      ','TRUCK    ',
     *        'LONG HAUL VEHICLE '
            WRITE (4,'(15X,A10,A10,A25)') '2      ','RAIL     ',
     *        'COMMERCIAL TRAIN  '
            WRITE (4,'(15X,A10,A10,A25)') '3      ','BARGE    ',
     *        'INLAND VESSEL     '
            WRITE (4,'(15X,A10,A10,A25)') '4      ','SHIP     ',
     *        'OPEN SEA VESSEL   '
            WRITE (4,'(15X,A10,A10,A25)') '5      ','CARGO AIR',
     *        'CARGO AIRCRAFT    '
            WRITE (4,'(15X,A10,A10,A25)') '6      ','PASS AIR ',
```

```
     *       'PASSENGER AIRCRAFT'
             WRITE (4,'(15X,A10,A10,A25)') '7   ','P-VAN    ',
     *       'PASSENGER VAN     '
             WRITE (4,'(15X,A10,A10,A25)') '8   ','CVAN-T   ',
     *       'COMMERCIAL VAN    '
             WRITE (4,'(15X,A10,A10,A25)') '9   ','CVAN-R   ',
     *       'COMMERCIAL VAN    '
             WRITE (4,'(15X,A10,A10,A25)') '10  ','CVAN-CA  ',
     *       'COMMERCIAL VAN    '
             RETURN
           ENDIF
   30 CONTINUE
      RETURN
      END
*DECK,BESSL
      SUBROUTINE BESSL (X,BK0,FK0)
C
C     MODIFIED BESSEL FUNCTION OF ORDER ZERO FOR FUNCTION #45 OF THE
C     RADTRAN III GUIDE.
C     SOURCE: HANDBOOK OF MATHEMATICAL FUNCTIONS. ABRAMOWITZ, M., AND
C             STEGUN, I. A., NBS APPLIED MATHEMATICS SERIES, NUMBER 55.
C
      COMMON /CON/ TOL,NDERV,NA,NC,TWOPI,PI,Q6,WS,GAM,FMU,DZ,VV
      REAL AS(7), BS(7), A(7), B(7)
      DATA AS / -0.57721566,0.422278420,0.23069756,0.03488590,0.00262698
     *  ,0.00010750,0.00000740 /
      DATA BS / 1.0000000,3.5156229,3.0899424,1.2067492,0.2659732,
     *  0.0360768,0.0045813 /
      DATA A / 1.25331414,0.07832358,0.02189568,0.01062446,0.00587872,
     *  0.00251540,0.00053208 /
      DATA B / 1.25331414,0.11190289,0.02576646,0.00933994,0.00417454,
     *  0.00163271,0.00033934 /
C
C ... EQN 9.8.1   FOR -3.75<=X<=3.75
      W = X/3.75
      BI0 = 0.0
      DO 10 J = 1, 7
        BI0 = BI0+BS(J)*W**(2*J-2)
   10 CONTINUE
C
C ... EQN 9.8.5  FOR 0<X<=2
      W = X/2
      BK0 = 0.0
      DO 20 J = 1, 7
        BK0 = BK0+AS(J)*W**(2*J-2)
   20 CONTINUE
      BK0 = BK0-ALOG(W)*BI0
C
C ... EQN 11.1.9  (INTEGRAL 0 TO X)
      W = 0.5*X
      S1 = 1.0
      S3 = 0.0
```

```
      S4 = 0.0
      T1 = 1.0
      T2 = 1.0
      T4 = 0.0
      DO 30 J = 1, 100
        S1 = S1*(W/J)**2*(2*J-1)/(2*J+1)
        S2 = S1/(2*J+1)
        S3 = S3+1./J
        S4 = S1*S3
        T1 = T1+S1
        T2 = T2+S2
        T4 = T4+S4
        IF (S1/T1.GT.1.E-10) GO TO 30
        IF (S2/T2.GT.1.E-10) GO TO 30
        IF (S4/T4.GT.1.E-10) GO TO 30
        GO TO 40
   30 CONTINUE
      WRITE (6,80)
      STOP
   40 CONTINUE
      FK0 = -(GAM+ALOG(W))*X*T1+X*(T2+T4)
C
C ... EQN 11.4.23 (INTEGRAL 0 TO INFINITY)
      FK01 = PI/2.0
C
C ... INTEGRAL X TO INFINITY
      FK0 = FK01 - FK0

      IF (X.LE.2) RETURN
C
C ... EQN 9.8.6  FOR 2<X<INFINITY
      W = -2.0/X
      BK0 = 0.0
      DO 60 J = 1, 7
        BK0 = BK0+A(J)*W**(J-1)
   60 CONTINUE
      BK0 = BK0*EXP(-X)/SQRT(X)

      IF (X.LT.7.) RETURN
C
C ... EQN 11.1.18  FOR 7<=X<=INFINITY
      W = -7.0/X
      FK0 = 0.0
      DO 70 J = 1, 7
        FK0 = FK0+B(J)*W**(J-1)
   70 CONTINUE
      FK0 = FK0*EXP(-X)/SQRT(X)
      RETURN
C
   80 FORMAT (///,5X,38HCONVERGENCE FAILED IN SUBROUTINE BESSL)
      END
```

```
*DECK,CENTER
      SUBROUTINE CENTER (STRING,I1,I2)
C
C     CENTERS ANY CHARACTERS IN 'STRING' THAT ARE BETWEEN I1 AND I2.
C     ANY OTHER CHARACTERS NOT BETWEEN I1 AND I2 ARE ELIMINATED.
C
      CHARACTER*80 STRING, TMPSTR

      IF (I1.GT.I2) RETURN
      DO 10 I = 1,80
          TMPSTR(I:I) = ' '
  10  CONTINUE
      I = I1
      J = I2
C ... FIND BEGINING OF STRING WITHIN 'STRING'.
  20  IF (STRING(I:I).EQ.' ') THEN
          I = I + 1
          GOTO 20
      ENDIF
C ... FIND END OF STRING WITHIN 'STRING'.
  30  IF (STRING(J:J).EQ.' ') THEN
          J = J - 1
          GOTO 30
      ENDIF
      IF (I.GT.J) RETURN
      ISTART = (80-J+I-1)/2
      TMPSTR (ISTART:(ISTART+J-I)) = STRING (I:J)
      STRING = TMPSTR
      RETURN
      END
*DECK,CONVRT
      SUBROUTINE CONVRT (WS)
C
C     THIS ROUTINE CONVERTS THE CONTENTS OF "TEST" INTO A REAL FLOATING
C     POINT NUMBER IF "TEST" IS OF THE APPROPRIATE FORM OR SETS THE
C     RESULT TO "4HWORD"
C     THE RESULT IS PLACED INTO THE ARGUMENT "WS"
C
      COMMON /CARD/ IPOINT,KBYTE,CONTUF,NRFLAG,NCS
      LOGICAL CONTUF,NRFLAG
      COMMON /CCARD/ ICARD,TEST,ICARDI,NVY(10)
      CHARACTER*10 ICARD*80, TEST,ICARDI,NVY*1
      DIMENSION JVY(10)
      CHARACTER*1 JVY
C
C
C     SUBROUTINE CONVRT  CONVERTS THE CONTENTS OF TEST TO REAL OR
C     SETS ITS ARG TO 4RWORD.
C
      LOGICAL FRACT
C
```

```
C
      ASSIGN 20 TO IGOTO
C
   10 CALL CPBYTE (JVY)
C
      I = 0
      GO TO IGOTO ,(20,50,130)
C
   20 ASSIGN 50 TO IGOTO
      IEX = 0
      IPX = 0
      WSA = 0
      SIGN = 1.0
      ISIGN = 0
      FRACT = .FALSE.
C
C     START SEARCH
C
      I = 1
   30 ICARDI = JVY(I)
      IF (ICARDI.NE.' '.AND.ICARDI.NE.','.AND.ICARDI.NE.'=') GO TO 40
      I = I+1
      IF (I.LE.KBYTE) GO TO 30
      GO TO 190
C
C     TEST FOR PERIOD OR LEADING SIGN
C
   40 IF (ICARDI.EQ.'.') GO TO 100
      IF (ICARDI.NE.'+'.AND.ICARDI.NE.'-') GO TO 70
      IF (ICARDI.EQ.'-') SIGN = -1.0
C
C     SEARCH NEXT CHARACTER
C
   50 I = I+1
      IF (I.LE.KBYTE) GO TO 60
      IF (.NOT.CONTUF) GO TO 180
      CALL NEXT
      GO TO 10
C
   60 ICARDI = JVY(I)
      IF (ICARDI.EQ.' '.OR.ICARDI.EQ.','.OR.ICARDI.EQ.'=') GO TO 180
C
C     TEST FOR DECIMAL POINT OR EXPONENTIAL FIELD.
C
      IF (ICARDI.EQ.'.') GO TO 100
      IF (ICARDI.EQ.'E'.OR.ICARDI.EQ.'+'.OR.ICARDI.EQ.'-') GO TO 110
C
C     TEST FOR NUMBER
C
   70 DO 80 J = 1, 10
        IF (ICARDI.EQ.NVY(J)) GO TO 90
```

```
   80 CONTINUE
      GO TO 190
C
C     ASSEMBLE NUMBER.
C
   90 WSA = 10.0*WSA+(J-1)
      IF (FRACT) IPX = IPX+1
      GO TO 50
C
C     SET UP TO CALCULATE FRACTIONAL PART
C
  100 IF (FRACT) GO TO 190
      FRACT = .TRUE.
      GO TO 50
C
C     FOUND E,+,OR-, SETUP EXPONENTIAL FIELD.
C
  110 ASSIGN 130 TO IGOTO
      IF (ICARDI.EQ.'E') GO TO 130
  120 IF (ISIGN.NE.0) GO TO 190
      ISIGN = +1
      IF (ICARDI.EQ.'-') ISIGN = -1
C
C     ADVANCE CHARACTER COUNTER AND EXPONENTIAL FIELD COUNTER.
C
  130 I = I+1
      IF (I.LE.KBYTE) GO TO 140
      IF (.NOT.CONTUF) GO TO 170
      CALL NEXT
      GO TO 10
C
  140 ICARDI = JVY(I)
      IF (ICARDI.EQ.' '.OR.ICARDI.EQ.','.OR.ICARDI.EQ.'=') GO TO 170
C
C     TEST FOR SIGN
C
      IF (ICARDI.EQ.'+'.OR.ICARDI.EQ.'-') GO TO 120
C
C     TEST FOR NUMBER
C
      DO 150 J = 1, 10
        IF (ICARDI.EQ.NVY(J)) GO TO 160
  150 CONTINUE
      GO TO 190
C
C     CALCULATE EXPONENTIAL
C
  160 IEX = 10*IEX+(J-1)
      GO TO 130
C
C     CALCULATE FINAL NUMBER
```

```
C
  170 IF (ISIGN.EQ.-1) IEX = -IEX
  180 WS = SIGN*WSA*10.0**(IEX-IPX)
      RETURN
C
C
  190 CONTINUE
      WRITE (4,200) ' EXPECTED A NUMERIC VALUE, FOUND: ',TEST
      STOP
  200 FORMAT(A34,A10)
      END
*DECK,COST
      SUBROUTINE COST
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      COMMON /MAXSZ/ MAXISO,MAXSEV,MAXGRP,MAXRAD,MAXDSP,MAXARS,
     *  MAXMOD,MAXNDL,MAXMAT,NPRAM,RSPHLF
      LOGICAL GROUP(15)
      REAL SUMCON(20), CONTAM(4,20)
```

```
C     COMPUTE ECONOMIC IMPACT BY POPULATION ZONE AND SEVERITY

      DO 15 I = 1, NGROUP
        GROUP(I) = .FALSE.
   15 CONTINUE
      HLIFE = 0

C ... FIND WHICH PACKAGES AND DISPERSABILITY CATEGORIES ARE BEING USED
      DO 28 M = 1, NMAT(MODE)
        MAT = MATSAV(MODE,M)
        DO 28 I = 1, NIS(MAT)
          ISO = ISOSAV(MAT,I)
          LIB = LIBSAV(ISO)
          DO 22 J = 1, NGROUP
            IF (IPCGRP(ISO).EQ.J) GROUP(J) = .TRUE.
   22     CONTINUE
C ......  FIND THE LONGEST HALF-LIFE
          IF (TABHLF(LIB).GT.HLIFE)  HLIFE = TABHLF(LIB)
   28 CONTINUE

      DO 30 J = 1, NSEV
C ..... FIND HIGHEST RELEASE FRACTION AND STORE IN RF
        RF = 0.0
        DO 25 I = 1, NGROUP
          IF (GROUP(I).AND.RFRAC(I,J).GT.RF)  RF = RFRAC(I,J)
   25   CONTINUE

        OCOST = 0.0
        IF ( (IEXCLU(MODE).EQ.1).OR.
     *       ((IEXCLU(MODE).EQ.0).AND.RF.GT.0.0)) THEN
          IF (RF.EQ.0.0) THEN
            OCOST = ONSCST(1)
          ELSEIF (RF.LE.0.01) THEN
            OCOST = ONSCST(2)
          ELSEIF (RF.LE.0.1) THEN
            OCOST = ONSCST(3)
          ELSE
            OCOST = ONSCST(4)
          ENDIF
        ENDIF

        DO 30 I = 1, NPOP
          ECNMPT(J,I) = EMRCST(J) + OCOST
   30 CONTINUE

      IF (IACC.EQ.1.OR.ECONF.EQ.0.0) GOTO 135

      DO 50 J = 1, NSEV
C . . . FIND TOTAL CONTAMINATION FOR EACH SEVERITY (SUMCON)
C . . . FIND AREAS OF EACH CONTAMINATION LEVEL AND SEVERITY
C . . . FIND HIGHEST CONTAMINATION FOR EACH CONTAM. LEVEL AND SEVERITY
```

```
(CONTAM)
        SUMCON(J) = 0.0
        AP1T1(J) = 0.0
        A1T20(J) = 0.0
        A20T40(J) = 0.0
        AGT40(J) = 0.0
        CONTAM(1,J) = 0.0
        CONTAM(2,J) = 0.0
        CONTAM(3,J) = 0.0
        CONTAM(4,J) = 0.0
        DO 50 K = 1, NAREAS
          SUMCON(J) = SUMCON(J) + TDCONF(K,J)
          IF (TDCONF(K,J).GT.40.) THEN
            AGT40(J) = AGT40(J)+ASIZE(K)
            IF (TDCONF(K,J) .GT. CONTAM(4,J)) CONTAM(4,J) = TDCONF(K,J)
          ELSEIF (TDCONF(K,J).GT.20.) THEN
            A20T40(J) = A20T40(J)+ASIZE(K)
            IF (TDCONF(K,J) .GT. CONTAM(3,J)) CONTAM(3,J) = TDCONF(K,J)
          ELSEIF (TDCONF(K,J).GT.1.0) THEN
            A1T20(J) = A1T20(J)+ASIZE(K)
            IF (TDCONF(K,J) .GT. CONTAM(2,J)) CONTAM(2,J) = TDCONF(K,J)
          ELSEIF (TDCONF(K,J).GT.0.1) THEN
            AP1T1(J) = AP1T1(J)+ASIZE(K)
            IF (TDCONF(K,J) .GT. CONTAM(1,J)) CONTAM(1,J) = TDCONF(K,J)
          ENDIF
   50 CONTINUE

      DO 130 J = 1, NSEV
C . . . IF NO CONTAMINATION IN THIS SEVERITY SKIP COMPUTATIONS
        IF (SUMCON(J) .EQ. 0.0)  GOTO 129
        DO 175 K = 1, 4
C . . . . FOR EACH CONTAMINATION AREA COMPUTE TBAR AND T1, T2, & T3
          IF (CONTAM(K,J) .LE. 1.0) THEN
            TBAR(K) = 0.0
          ELSE
            TBAR(K) = ALOG(CONTAM(K,J))*HLIFE/.693
          ENDIF
          TS1(K) = AMIN1(18250.,TBAR(K))
          TS2(K) = AMIN1(10.,TBAR(K))
          IF (K.EQ.3) THEN
C . . . . . T3 IS ONLY USED FOR AREA 3
            DO 145 I = 1, NPOP
              PD = POPDEN(I)
C . . . . . CORRECTION FOR NON-CONTINUOUS FUNCTION
              IF (PD .LE. 1.0)  PD = 1.001
              T3 = 10 + 8.1E-5*A20T40(J)/ALOG(PD)
              TS3(I) = AMIN1(T3,TBAR(K))
  145       CONTINUE
          ENDIF
  175   CONTINUE
```

```
      IF (HLIFE .GT. 365.25) THEN
C . . . . IS THE AREA WITH TDCONF GREATER THAN 40 < 500 METERS SQ
        IF (AGT40(J).LT.500.0) THEN
          CALL ECON (J,4,EVAL1)
          DO 70 I = 1, NPOP
            ECNMPT(J,I) = ECNMPT(J,I)+EVAL1(I)*ECONF
 70       CONTINUE
        ELSE
          CALL ECON (J,1,EVAL1)
          CALL ECON (J,2,EVAL2)
          DO 80 I = 1, NPOP
            IF (EVAL1(I).GT.EVAL2(I)) THEN
              ECNMPT(J,I) = ECNMPT(J,I)+EVAL2(I)*ECONF
            ELSE
              ECNMPT(J,I) = ECNMPT(J,I)+EVAL1(I)*ECONF
            ENDIF
 80       CONTINUE
        ENDIF
      ELSEIF (HLIFE .GT. 2) THEN
C . . . . IS THE AREA WITH TDCONF GREATER THAN 40 < 500 METERS SQ
        IF (AGT40(J).LT.500.0) THEN
          CALL ECON (J,4,EVAL1)
          CALL ECON (J,3,EVAL2)
          DO 90 I = 1, NPOP
            IF (EVAL1(I).GT.EVAL2(I)) THEN
              ECNMPT(J,I) = ECNMPT(J,I)+EVAL2(I)*ECONF
            ELSE
              ECNMPT(J,I) = ECNMPT(J,I)+EVAL1(I)*ECONF
            ENDIF
 90       CONTINUE
        ELSE
          CALL ECON (J,1,EVAL1)
          CALL ECON (J,2,EVAL2)
          CALL ECON (J,3,EVAL3)
          DO 100 I = 1, NPOP
            VAL = AMIN1(EVAL1(I),EVAL2(I),EVAL3(I))
            ECNMPT(J,I) = ECNMPT(J,I)+VAL*ECONF
100       CONTINUE
        ENDIF
      ELSE
        CALL ECON (J,3,EVAL1)
        DO 110 I = 1, NPOP
          ECNMPT(J,I) = ECNMPT(J,I)+EVAL1(I)*ECONF
110     CONTINUE
      ENDIF
C ... GOTO 129 NO CONTAMINATION IN SEVERITY J -- NO CALCULATIONS ARE NEEDED
129 CONTINUE
C ... END J = 1, NSEV
130 CONTINUE
C
135 CONTINUE
```

```
      DO 140 J = 1, NSEV
        DO 140 I = 1, NPOP
          ECNPYR(J,I) = ECNMPT(J,I)*ACCPYR(J,I)
  140 CONTINUE
      RETURN
      END
*DECK,CPBYTE
      SUBROUTINE CPBYTE (JVY)
C
C     THIS ROUTINE CONVERTS THE CONTENTS OF "TEST" INTO INDIVIDUAL
C     CHARACTERS AND PLACES THE RESULTING CHARACTERS INTO ARRAY "JVY"
C
      DIMENSION JVY(10)
      CHARACTER*1 JVY
      COMMON /CARD/ IPOINT,KBYTE,CONTUF,NRFLAG,NCS
      LOGICAL CONTUF,NRFLAG
      COMMON /CCARD/ ICARD,TEST,ICARDI,NVY(10)
      CHARACTER*10 ICARD*80, TEST,ICARDI,NVY*1
      DO 10 J = 1, KBYTE
        JVY(J) = TEST(J:J)
   10 CONTINUE
      RETURN
      END
*DECK,EARLY
      SUBROUTINE EARLY
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
```

```
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /CON/ TOL,NDERV,NA,NC,TWOPI,PI,Q6,WS,GAM,FMU,DZ,VV
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
      IF (IACC.EQ.1) THEN
C       COMPUTE EXPECTED NUMBER OF PEOPLE IN EACH ANNULAR ZONE
        IF (NRAD.LT.2) GO TO 30
C . . . RURAL AND SUBURBAN -- EQUATION 55 OF RADTRAN III GUIDE
        CONR = PI*POPDR*1.E-6
        CONS = PI*POPDS*1.E-6
        DO 10 I = 2, NRAD
          IM1 = I-1
          R2R = RADIST(I,1)
          R1R = RADIST(IM1,1)
          R2S = RADIST(I,2)
          R1S = RADIST(IM1,2)
          DRR2 = R2R*R2R-R1R*R1R
          DRS2 = R2S*R2S-R1S*R1S
          EXPEO(IM1,1) = DRR2*CONR
          EXPEO(IM1,2) = DRS2*CONS
   10   CONTINUE
C . . . URBAN -- EQUATION 54 OF RADTRAN III GUIDE
C       RPD ACCOUNTS FOR PEDESTRIANS IN URBAN AREAS FOR EXPOSURE SOURCE
        CONU = 8.*WS*RPD*POPDU*1.E-6
        DO 20 I = 2, NRAD
          IM1 = I-1
          DRAD = RADIST(I,3)-RADIST(IM1,3)
          EXPEO(IM1,3) = CONU*DRAD
   20   CONTINUE
   30   CONTINUE
C       COMPUTE EXPECTED NUMBER OF EARLY FATALITIES AND EARLY MORBIDITIES
C       FOR EACH POPULATION ZONE AND SEVERITY
C       USE 1 YR/MARROW THRESHOLD FOR EARLY MORBIDITY TEST
        THRES = EMTRSH(2)
        DO 50 I = 1, NPOP
          DO 50 J = 1, NSEV
            SUM1 = 0.0
            SUM2 = 0.0
            DO 40 K = 2, NRAD
              DOSE = TDSDST(K-1,J,I)
              EPEO = EXPEO(K-1,I)
              CALL EFFIT (4,DOSE,PREF)
              SUM1 = SUM1+PREF*EPEO
              IF (DOSE.LE.THRES) GO TO 40
              SUM2 = SUM2+EPEO
   40       CONTINUE
            SUMEF(J,I) = SUM1
```

```
            SUMEM(J,I) = SUM2
   50     CONTINUE
C
C . . DISPERSAL ACCIDENT
       ELSEIF (IACC.EQ.2) THEN
C        COMPUTE EXPECTED NUMBER OF PEOPLE IN EACH ANNULAR ZONE
         CONR = POPDR*1.E-6
         CONS = POPDS*1.E-6
         CONU = POPDU*1.E-6*0.52
         CONUP = POPDUP*1.E-6
         DO 60 I = 1, NAREAS
           EXPEO(I,2) = ASIZE(I)*CONS
           EXPEO(I,3) = ASIZE(I)*CONU
           EXPEO(I,4) = ASIZE(I)*CONUP
           EXPEO(I,1) = ASIZE(I)*CONR
   60    CONTINUE
C        COMPUTE EXPECTED NUMBER OF EARLY FATALITIES FOR EACH
C        POPULATION ZONE AND SEVERITY
         DO 130 J = 1, NSEV
           DO 80 I = 1, NPOP
             SUMEF(J,I) = 0.0
   80      CONTINUE
           DO 120 K = 1, NAREAS
             PRLUNG = 0.0
C . . . . . FOR EACH LUNG TYPE WITH NO BUILDING DOSE FACTOR
             DO 125 L = 1, 3
               CALL EFFIT (L, EARLYL(K,J,L), PROB)
               PRLUNG = PRLUNG + PROB
  125        CONTINUE
C . . . . . PROBABILITY SHOULD BE <= 1
             IF (PRLUNG .GT. 1.0)  PRLUNG = 1.0

C . . . . . MARROW
             CALL EFFIT (4, EARLYD(K,J,2), PRMAR)

C . . . . . RURAL, SUBURBAN, AND PEDESTRIAN
             SUMEF(J,1) = SUMEF(J,1) + PRLUNG*EXPEO(K,1)
     *                               + PRMAR*EXPEO(K,1)
             SUMEF(J,2) = SUMEF(J,2) + PRLUNG*EXPEO(K,2)
     *                               + PRMAR*EXPEO(K,2)
             SUMEF(J,3) = SUMEF(J,3) + PRLUNG*EXPEO(K,4)
     *                               + PRMAR*EXPEO(K,4)

             PRLUNG = 0.0
C . . . . . FOR EACH LUNG TYPE WITH A BUILDING DOSE FACTOR (REDUCES THE
DOSE)
             DO 128 L = 1, 3
               CALL EFFIT (L, EARLYL(K,J,L)*BDF, PROB)
               PRLUNG = PRLUNG + PROB
  128        CONTINUE
C . . . . . PROBABILITY SHOULD BE <= 1
```

```
            IF (PRLUNG .GT. 1.0)  PRLUNG = 1.0

C . . . . . MARROW WITH BUILDING DOSE FACTOR
            CALL EFFIT (4, EARLYD(K,J,2)*BDF, PRMAR)

C . . . . . URBAN
            SUMEF(J,3) = SUMEF(J,3) + PRLUNG*EXPEO(K,3)
     *                              + PRMAR*EXPEO(K,3)
  120     CONTINUE
  130   CONTINUE
C       COMPUTE EXPECTED NUMBER OF EARLY MORBIDITIES FOR EACH
C       POPULATION ZONE AND SEVERITY
        DO 180 J = 1, NSEV
          DO 100 I = 1, NPOP
            SUMEM(J,I) = 0.0
  100     CONTINUE
          DO 180 K = 1, NAREAS
            DO 180 L = 1, NTHRS
C . . . . . . RURAL, SUBURBAN, AND PEDESTRIAN
              IF (EARLYD(K,J,L) .GT. EMTRSH(L)) THEN
                SUMEM(J,1) = SUMEM(J,1) + EXPEO(K,1)
                SUMEM(J,2) = SUMEM(J,2) + EXPEO(K,2)
                SUMEM(J,3) = SUMEM(J,3) + EXPEO(K,4)
              ENDIF
C . . . . . . URBAN (INCLUDE BUILDING DOSE FACTOR)
              IF (EARLYD(K,J,L)*BDF .GT. EMTRSH(L)) THEN
                SUMEM(J,3) = SUMEM(J,3) + EXPEO(K,3)
              ENDIF
  180   CONTINUE

      ELSE
        WRITE (4,*) 'INVALID IACC VALUE IN EARLY'
        STOP
      ENDIF
C
      DO 220 I = 1, NPOP
        DO 220 J = 1, NSEV
C . . . . SUBTRACT EARLY FATALITIES FROM EARLY MORBIDITIES
          SUMEM(J,I) = SUMEM(J,I)-SUMEF(J,I)
C . . . . EXPECTED NUMBER OF EARLY MORBIDITIES
          EMPYR(J,I) = SUMEM(J,I)*ACCPYR(J,I)
C . . . . EXPECTED NUMBER OF EARLY FATALITIES
          EFPYR(J,I) = SUMEF(J,I)*ACCPYR(J,I)
  220 CONTINUE
      RETURN
      END
*DECK,ECON
      SUBROUTINE ECON (J,JOPT,ECNVAL)
C
C     THIS ROUTINE CALCULATES THE ECONOMIC IMPACT ARISING FROM
C     DISPERSAL ACCIDENT CLEANUP.
```

```
C
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
C
      DIMENSION ECNVAL(3)
C     FRACTION OF RURAL AREA ASSUMED TO BE UNDER CULTIVATION
      TL = XFARM
C     COMPUTE ALPHA AND 1-ALPHA (ASSOCIATED WITH CATEGORY 4)
      EARG = EXP(-.12*TS1(4)/365.)
      ALPHA = EARG*(.3+.7*EARG)
      OMALP = 1.-ALPHA
      A1 = AP1T1(J)*1.E-6
      A2 = A1T20(J)*1.E-6
      A3 = A20T40(J)*1.E-6
      A4 = AGT40(J)*1.E-6
      CON1 = 3.5E4*TL
      GO TO (10,20,30,40), JOPT
      CALL GOTOER ('ECON-1')
   10 CONTINUE
C     OPTION 1 - RURAL
```

```
      ECNVAL(1) = (1.E5+CON1)*A1+(3.1E5+CON1+(1800+34.*TS2(2))*POPDR+
     *  140.*TS2(2))*A2+(4.2E5+3.2E5*TL+(2300.+34.*TS3(1))*POPDR+
     *  140.*TS3(1))*A3+(1.E5+CON1+1.7E5*OMALP+9900.*POPDR+140.*TS1(
     *  4))*A4
C     OPTION 1 - SUBURBAN
      ECNVAL(2) = 1.2E5*A1+(2.3E5+(300.+41.*TS2(2))*POPDS+140.*TS2(2))*
     *  A2+(2.4E5+(1000.+41.*TS3(2))*POPDS+140.*TS3(2))*A3+
     *  (1.2E5+(1.2E4+2.3E4*OMALP)*POPDS+140.*TS1(4))*A4
C     OPTION 1 - URBAN
      ECNVAL(3) = 2.7E5*A1+(5.6E5+983.*POPDU+248.*POPDU*TS2(2)+140.*TS2(
     *  2))*A2+(5.6E5+(8649.+248.*TS3(3))*POPDU+140.*TS3(3))*A3+
     *  (2.7E5+(2.3E5*OMALP+1.23E5)*POPDU+140.*TS1(4))*A4
      RETURN
C
   20 CONTINUE
C     OPTION 2 - RURAL
      TS4 = 180.*A4
      ECNVAL(1) = (1.E5+CON1)*A1+(3.1E5+CON1+(1800.+34.*TS2(2))*POPDR+
     *  140.*TS2(2))*A2+(4.2E5+3.2E5*TL+(2300.+34.*TS3(1))*POPDR+
     *  140.*TS3(1))*A3+(3.0E7+3.0E5*TL+34.*TS4*POPDR+47.*TS4)*A4
C     OPTION 2 - SUBURBAN
      TS4 = 360.*A4
      ECNVAL(2) = 1.2E5*A1+(2.3E5+(300.+41.*TS2(2))*POPDS+140.*TS2(2))*
     *  A2+(2.4E5+(1000.+41.*TS3(2))*POPDS+140.*TS3(2))*A3+
     *  (2.3E5+7.1E7+41.*TS4*POPDS+47.*TS4)*A4
C     OPTION 2 - URBAN
      TS4 = 540.*A4
      ECNVAL(3) = 2.7E5*A1+(5.6E5+983.*POPDU+248.*POPDU*TS2(2)+140.*TS2(
     *  2))*A2+(5.6E5+(8649.+248.*TS3(3))*POPDU+140.*TS3(3))*A3+
     *  (5.4E5+3.6E9+41.*TS4*POPDU+47.*TS4)*A4
      RETURN
C
C     OPTION 3
   30 CONTINUE
C     OPTION 3 - RURAL
      ECNVAL(1) = (1.E5+CON1)*A1+(2.E5+CON1+34.*TBAR(2)*POPDR+140.*TBAR(
     *  2))*A2+(2.5E5+CON1+34.*TBAR(3)*POPDR+140.*TBAR(3))*A3+(2.5E5+
     *  CON1+34.*TBAR(4)*POPDR+140.*TBAR(4))*A4
C     OPTION 3 - SUBURBAN
      ECNVAL(2) = 1.2E5*A1+(2.3E5+41.*TBAR(2)*POPDS+140.*TBAR(2))*A2+
     *  (2.3E5+41.*TBAR(3)*POPDS+140.*TBAR(3))*A3+(2.3E5+41.*TBAR(4)*
     *  POPDS+140.*TBAR(4))*A4
C     OPTION 3 - URBAN
      ECNVAL(3) = 2.7E5*A1+(5.6E5+248.*TBAR(2)*POPDU+140.*TBAR(2))*A2+
     *  (5.6E5+248.*TBAR(3)*POPDU+140.*TBAR(3))*A3+(5.6E5+248.*TBAR(4)*
     *  POPDU+140.*TBAR(4))*A4
      RETURN
C     OPTION 1A
   40 CONTINUE
C     OPTION 1A - RURAL
      ECNVAL(1) = (1.E5+CON1)*A1+(3.1E5+CON1+(1800.+34.*TS2(2))*POPDR+
```

```
     *  140.*TS2(2))*A2+(4.2E5+3.2E5*TL+(2300.+34.*TS3(1))*POPDR+
     *  140.*TS3(1))*(A3+A4)
C     OPTION 1A - SUBURBAN
      ECNVAL(2) = 1.2E5*A1+(2.3E5+(300.+41.*TS2(2))*POPDS+140.*TS2(2))*
     *  A2+(2.4E5+(1000.+41.*TS3(2))*POPDS+140.*TS3(2))*(A3+A4)
C     OPTION 1A - URBAN
      ECNVAL(3) = 2.7E5*A1+(5.6E5+983.*POPDU+248.*POPDU*TS2(2)+140.*TS2(
     *  2))*A2+(5.6E5+(8649.+248.*TS3(3))*POPDU+140.*TS3(3))*(A3
     *  +A4)
      RETURN
      END
*DECK,EFFIT
      SUBROUTINE EFFIT (ISUB,DOSE,PR)
C
C     THIS ROUTINE DETERMINES EARLY FATALITY PROBABILITIES BY
C     TABLE LOOK-UP
C
      COMMON /PROB/ PROBEF(30,4),REMLEV(30),NE,NREMLV
C
      DO 10 K = 2, NREMLV
        IF (DOSE.GT.REMLEV(K)) GO TO 20
   10 CONTINUE
      PR = PROBEF(NREMLV,ISUB)
      RETURN
   20 PR = PROBEF(K-1,ISUB)
      RETURN
      END
*DECK,FINDI
      FUNCTION FINDI (X)
C
C     PERFORMS INTEGRATION OF EQUATION #11 OF THE RADTRAN III GUIDE WITH
C     EXP(-UR)*B(R) INCLUDED INSIDE THE INTEGRAL.
C
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)

      XU = X*U
      IF (XU.EQ.0.0) THEN
        ANS = 1.570796
        FINDI = ANS/X
      ELSE
        CALL BSKIN (XU,1,2,1,ANS,NZ,IERR)
        IF (IERR.NE.0.0) THEN
          WRITE (4,*) 'FINDI FAILED ON BSKIN CALL'
          STOP
        ENDIF
        ANS = ANS/X+(A(1)-A(3)*X**2)*BESK0(XU)
     *             +(A(2)*X+A(3)*X/U)*BESK1(XU)
```

```
C ....  "UNMODIFY" THE BESSEL FUNCTIONS (MULTIPLY BY EXP(-XU))
        FINDI = ANS*EXP(-XU)
      ENDIF
      RETURN
      END
*DECK,GOTOER
      SUBROUTINE GOTOER (STRING)
      CHARACTER*10 STRING
      WRITE (4,'(2X,A30)') 'GOTO ERROR AT ',STRING
      STOP
      END
*DECK,IARRAY
      SUBROUTINE IARRAY (IA,N)
C
C     THIS ROUTINE READS "N" VALUES INTO THE INTEGER ARRAY "IA"
C
      COMMON /CARD/ IPOINT,KBYTE,CONTUF,NRFLAG,NCS
      LOGICAL CONTUF,NRFLAG
      COMMON /CCARD/ ICARD,TEST,ICARDI,NVY(10)
      CHARACTER*10 ICARD*80, TEST,ICARDI,NVY*1
C
C
      DIMENSION IA(N)
      NRS = 0
      NR = 0
      DO 20 I = 1, N
        NR = NR+1
        IF (NR.LE.NRS) GO TO 10
        NR = 1
        NRS = 0
        VAL = VALUE(DMY)
        IF (.NOT.NRFLAG) GO TO 10
        NRS = VAL+.01
        VAL = VALUE(DMY)
   10   IA(I) = IFIX(VAL)
   20 CONTINUE
      RETURN
      END
*DECK,INITLZ
      SUBROUTINE INITLZ (N)
C
C     INITLZ (INITIALIZE) SETS ARRAYS TO ZERO.  DEPENDING ON THE PARAMETER
C     PASSED DIFFERENT ARRAYS ARE INITIALIZED.
C       PARAMETER = 1 -> ARRAYS THAT ARE SUMMED OVER ALL LINKS FOR EACH
C                        ISOTOPE OR MATERIAL AND ARRAYS SUMMED OVER
C                        ISOTOPES FOR EACH LINK.
C       PARAMETER = 2 -> ARRAYS THAT ARE SUMMED OVER PASQUILL CATEGORIES
C                        FOR EACH LINK.
C       PARAMETER = 3 -> ARRAYS USED TO CALCULATE VALUES FOR EACH LINK AND
C                        FOR EACH PASQUILL CATEGORY.  EITHER SUMMED OVER
C                        ISOTOPES OR FOR EACH ISOTOPES.
```

```
C
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /MAXSZ/ MAXISO,MAXSEV,MAXGRP,MAXRAD,MAXDSP,MAXARS,
     *  MAXMOD,MAXNDL,MAXMAT,NPRAM,RSPHLF
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
      REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
      LOGICAL RSPEC
      COMMON /TALLY/ TSUMRS(65,12),VSUMRS(65,12),VALINK(40,3)

      GO TO (100,200,300) N

  100 CONTINUE
C ... ARRAYS SUMMED OVER ALL LINKS
C ... FOR EACH MATERIAL
C ... SUMMED OVER PASQUILL CAT
      DO 30 I = 1, MAX0(NMODE,NLNK,2)
        DO 10 J = 1, NPRAM
          SENPRM(J,I) = 0.0
```

```
   10   CONTINUE
        DO 30 J = 1, 3
          VALINK(I,J) = 0.0
   30 CONTINUE
      DO 35 I = 1, MAX0(NMODE,NLNK,2)
        DOSMAX(I) = 0.0
   35 CONTINUE
C ... IF ROUTE-SPECIFIC SUMMED OVER ISO FOR EACH LINK
C ... ELSE SUMMED OVER LINKS FOR EACH ISOTOPE
      DO 40 I = 1, MAX0(ICOUNT,NLNK)
        DO 40 J = 1, LISTLH
          VSUMRS(I,J) = 0.0
   40 CONTINUE
      RETURN

  200 CONTINUE
C ... ARRAYS SUMMED OVER PASQUILL CATEGORIES FOR EACH LINK
      DO 50 I = 1, ICOUNT
        DO 50 J = 1, LISTLH
          TSUMRS(I,J) = 0.0
   50 CONTINUE
      DO 60 I = 1, NSEV
        DO 60 J = 1, NPOP
          PECMPT(I,J) = 0.0
          PSUMEF(I,J) = 0.0
          PSUMEM(I,J) = 0.0
          PDTOT(I,J)  = 0.0
   60 CONTINUE
C ... FOR EACH LINK (NOT SUMMED -- NEED TO ZERO EACH POP ZONE FOR LINKS)
      IF (RSPEC) THEN
        DO 65 I = 1, NSEV
          DO 65 J = 1, NPOP
            ACCRAT(I,J) = 0.0
            ACCPYR(I,J) = 0.0
   65   CONTINUE
      ENDIF
      RETURN

  300 CONTINUE
C ... ARRAYS CALCULATED FOR EACH LINK FOR EACH PASQUILL CATEGORY
C ... SUMMED OVER ISOTOPES
      DO 70 I = 1, NSEV
        DO 80 J = 1, NAREAS
          TDCONF(J,I) = 0.0
          TCLEVL(J,I) = 0.0
C . . . . FOR EACH LUNG TYPE (1 - 3) FOR EARLY FATALITIES
          DO 75 K = 1, 3
            EARLYL(J,I,K) = 0.0
   75     CONTINUE
C . . . . FOR EACH ORGAN USED IN EARLY MORBIDITY CALCULATION
          DO 80 K = 1, NTHRS
```

```
            EARLYD(J,I,K) = 0.0
   80     CONTINUE
          DO 90 J = 1, NPOP
            DTOT(I,J) = 0.0
C           ECNMPT IS DONE AT MATERIAL LEVEL -- NOT SUMMED OVER ISOTOPES
            ECNMPT(I,J) = 0.0
            DO 90 K = 1, NRAD
              TDSDST(K,I,J) = 0.0
   90     CONTINUE
   70   CONTINUE
        RETURN
        END
*DECK,MDT
        SUBROUTINE MDT (CHITAB,AREATB,DEPVEL,NAREAS,CHIVAL,CTMLVL,NV)
C
C       CALCULATES CONCENTRATION OF MATERIAL IN CLOUD AFTER CLOUD
C       HAS BEEN DEPLETED BY SCAVENGING OR GRAVITATIONAL SETTLING.
C
        COMMON /CMDT/ EPS,DLIM,MTAL
        DIMENSION CHITAB(NAREAS),AREATB(NAREAS),CHIVAL(NAREAS),
       *  CTMLVL(NAREAS)
C
        NV = NAREAS
        IF (DEPVEL.EQ.0.) GO TO 60
C       INITIALIZE
        DEPSUM = 0.
        DSAV = 0.
        CHIA = CHITAB(1)
        CHIB = CHITAB(2)
        CHIS = CHIB
        K = 2
        CHIVAL(1) = CHIA
        CTMLVL(1) = CHIA*DEPVEL*1.E6
C       COMPUTE INITIAL ESTIMATE OF DEPOSITION
   10   CONTINUE
        ADIF = AREATB(K)-AREATB(K-1)
        IF (CHIA*CHIB.LT.0.) GO TO 40
        DEPI = DEPVEL*SQRT(CHIA*CHIB)*ADIF
        DEPJ = DEPI
        ITAL = 0
C
        QP = 1.-DSAV-DEPJ
        CHIB = CHIB*QP
   20   CONTINUE
        IF (CHIA*CHIB.LT.0.) GO TO 40
        DEPJ = DEPVEL*SQRT(CHIA*CHIB)*ADIF
        IF (ABS((DEPI-DEPJ)/DEPI).LT.EPS) GO TO 30
        ITAL = ITAL+1
        CHIB = .5*(CHIS+CHIB)
        IF (ITAL.LT.MTAL) GO TO 20
        STOP 100
```

```
   30 CONTINUE
      CHIVAL(K) = CHIB
      CTMLVL(K) = DEPVEL*CHIB*1.E6
      DEPSUM = DEPSUM+DEPJ
      IF (DEPSUM.GT.1.) GO TO 40
      IF (K.EQ.NAREAS) RETURN
      K = K+1
      IF (AREATB(K).GT.DLIM) GO TO 40
      CHIA = CHIB
      CHIB = CHITAB(K)*(1.-DEPSUM)
      CHIS = CHIB
      DSAV = DEPSUM
      GO TO 10
   40 CONTINUE
C ... THE REST OF THE ISOPLETH AREAS DON'T HAVE CONTAMINATION FOR THIS
ISOTOPE
      NV = K-1
      DO 50 I = K, NAREAS
        CHIVAL(I) = 0.0
        CTMLVL(I) = 0.0
   50 CONTINUE
      RETURN
C     ZERO DEP. VEL.
   60 CONTINUE
      DO 70 K = 1, NAREAS
        CHIVAL(K) = CHITAB(K)
        CTMLVL(K) = 0.
        IF (AREATB(K).GT.DLIM) GO TO 40
   70 CONTINUE
      RETURN
      END
*DECK,NEWPAG
      SUBROUTINE NEWPAG
C
C     NEWPAG (NEW PAGE) PRINTS A FORM FEED AND AT THE TOP OF THE PAGE
PRINTS
C     THE DATE, TIME, PAGE NUMBER, AND TITLE IF AVAILABLE.
C
      COMMON /HEADER/ CDATE, KLOCK, TITLE
      CHARACTER CDATE*9, KLOCK*8, TITLE*80
      INTEGER PAGE
      SAVE PAGE
      DATA PAGE /1/

C ... CHAR(12) IS AN ASCII FORM FEED
      IF (PAGE.NE.1)  WRITE (4, '(1X,A1)') CHAR(12)

      WRITE (4, 10) CDATE, KLOCK, PAGE
   10 FORMAT (24X,'RUN DATE: [ ',A9,' AT ',A8,' ]',12X,'PAGE ',I4,/)

C ... For the first 2 pages the title has not been read in
```

```
      IF (PAGE .GT. 2)  WRITE (4, '(A80,///)') TITLE
      PAGE = PAGE + 1

      RETURN
      END
*DECK,NEXT
      SUBROUTINE NEXT
C
C     THIS ROUTINE READS INPUT CARDS AND RETURNS IN "TEST" THE NEXT
C     FIELD ON THE CARD
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /CARD/ IPOINT,KBYTE,CONTUF,NRFLAG,NCS
      LOGICAL CONTUF,NRFLAG
      COMMON /CCARD/ ICARD,TEST,ICARDI,NVY(10)
      CHARACTER*10 ICARD*80, TEST,ICARDI,NVY*1
      INTEGER NUMLIN
      SAVE NUMLIN
      DATA NUMLIN /0/
C
C ... IF TEST IS EOF THEN HAD COMPLETED A CALL TO RDINFR AND STARTING A NEW
ONE
      IF (TEST.EQ.'EOF')  NUMLIN = 0
   10 CALL SCAN
      IF (TEST.NE.'&&') GO TO 20
      IPOINT = 81
      GO TO 30
   20 CONTINUE
      IF (TEST.EQ.' ') GO TO 30
      RETURN
   30 READ (2,50,END=40) ICARD(1:80)
      NUMLIN = NUMLIN + 1
      IF (MOD(NUMLIN,52).EQ.0)  CALL NEWPAG
      WRITE (4,60) ICARD(1:80)
      IPOINT = 1
      GO TO 10
   40 TEST(1:10) = 'EOI'
      IPOINT = 81
      RETURN
   50 FORMAT(A)
   60 FORMAT(1X,(A))
      END
*DECK,NORLNK
      SUBROUTINE NORLNK (DIS,VDN,SPD,PDN,MOD,LTYP,ZONE,
     *                   STERM,PV,DOS5,DOS6)
      CHARACTER ZONE
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
```

```
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
C    *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
      COMMON /SENDOS/ DOSE2(3), DOSE5(13), DOSE6(9)
      EXTERNAL FINDI
C
C     DIS IS IN KILOMETERS
C     SPD IS IN METERS/SECOND
C
C     OFF LINK DOSE

C ... Pedestrian Ratio = Ratio of Pedestrians
      PDR = RPD
C ... If IUOPT is 3 pedestrains are not considered separately
      IF (IUOPT .EQ. 3)  PDR = 1

      X3 = 800.
C ... TRUCK AND ALL VANS ARE CONSIDERED MODE 1
      IF (MOD.EQ.1) THEN
        X1 = 27.
        X2 = 30.
C ...   If the road type is freeway then there are no sidewalks with
pedestrians
        IF (LTYP .EQ. 1)  X1 = 30.
      ELSEIF (MOD.EQ.2) THEN
        X1 = 30.
        X2 = 30.
      ENDIF

C ... Shielding factor (SF) determined by population density
      IF (ZONE.EQ.'R') THEN
        SF = RR
        INDEX = 1
      ELSEIF (ZONE.EQ.'S') THEN
        SF = RS
        INDEX = 2
      ELSE
        SF = RU
        INDEX = 3
C ...   In urban areas: If non-freeway, narrow streets with pedestrian
sidewalks
```

```
            IF ((MOD .EQ. 1) .AND. (LTYP .EQ. 2)) THEN
                X1 = 5
                X2 = 8
            ENDIF
          ENDIF
          U = COEF(1,1)
          A(1) = COEF(1,2)
          A(2) = COEF(1,3)
          A(3) = COEF(1,4)
          CALL GAUS8 (FINDI,X1,X2,1E-5,ANS1,IERR1)
          CALL GAUS8 (FINDI,X2,X3,1E-5,ANS2,IERR2)
          U = COEF(2,1)
          A(1) = COEF(2,2)
          A(2) = COEF(2,3)
          A(3) = COEF(2,4)
          CALL GAUS8 (FINDI,X1,X2,1E-5,ANS3,IERR3)
          CALL GAUS8 (FINDI,X2,X3,1E-5,ANS4,IERR4)
          IF (IERR1.EQ.2.OR.IERR2.EQ.2.OR.IERR3.EQ.2.OR.IERR4.EQ.2) THEN
            WRITE (4,100) IERR1,IERR2,IERR3,IERR4
            STOP
          ENDIF
          F = FG*(ANS1*PDR+ANS2*SF)+FN*(ANS3*PDR+ANS4*SF)
          DOS5 = 4*2.8E-10*DIS*STERM*PDN*F/SPD

C ... GET SENSITIVITY DATA
          IF (ISEN .GE. 3) THEN
             DOSE5(INDEX) = DOSE5(INDEX) + DOS5
             IF ((IUOPT .NE. 3) .AND. (F .NE. 0)) THEN
C .......  GET DOSE USING SHIELDING FACTOR VARIABLE
             DOSE5(INDEX+10) = DOSE5(INDEX+10)+DOS5/F*(FG*ANS2+FN*ANS4)*SF
C .......  GET DOSE USING RPD VARIABLE
             IF (MOD .EQ. 1)
     *              DOSE5(10) = DOSE5(10) + DOS5/F*(FG*ANS1+FN*ANS3)*RPD
          ENDIF
          ENDIF
C
C     ON LINK DOSE
          IF (MOD.EQ.2) THEN
              F1 = 0
              F2 = 0
              W = 3.0
          ELSE
              XMIN = 2.*SPD
              CALL QUAD (1,XMIN,ANS1)
              CALL QUAD (2,XMIN,ANS2)
              X = 4.0
              W = 15.0
              IF (LTYP.EQ.2)  W = 3.0
              F1 = FG*ANS1+FN*ANS2
              F2 = (1./X-(FG*ANS1+FN*ANS2))/2.
          ENDIF
```

```
      U = COEF(1,1)
      A(1) = COEF(1,2)
      A(2) = COEF(1,3)
      A(3) = COEF(1,4)
      F = FG*FINDI(W)
      U = COEF(2,1)
      A(1) = COEF(2,2)
      A(2) = COEF(2,3)
      A(3) = COEF(2,4)
      F = F + FN*FINDI(W)
      DOS6 = 7.7E-8*STERM*DIS*PV*VDN/(SPD**2)*2.0*(F+(F1+F2))

C ... GET SENSITIVITY DATA
      IF (ISEN .GE. 3) THEN
         DOSE6(INDEX) = DOSE6(INDEX) + DOS6
         IF (MOD .EQ. 1) THEN
C .........  GET DOSE USING FREEWAY VARIABLE
            IF ((ZONE.NE.'U') .AND. (LTYP.EQ.1)) THEN
                DOSE5(7) = DOSE5(7) + DOS5
                DOSE6(7) = DOSE6(7) + DOS6
            ENDIF
C .........  GET DOSE USING CITY STREETS VARIABLE
            IF ((ZONE.EQ.'U') .AND. (LTYP.EQ.2)) THEN
                DOSE5(8) = DOSE5(8) + DOS5
                DOSE6(8) = DOSE6(8) + DOS6
             ENDIF
             IF (LTYP .EQ. 2) THEN
C ............   GET DOSE USING VELOCITY VARIABLES
                DOSE5(INDEX+3) = DOSE5(INDEX+3) + DOS5
                DOSE6(INDEX+3) = DOSE6(INDEX+3) + DOS6
             ELSE
C ............   GET DOSE USING RURAL VEL. VARIABLE (ALL RURAL ON FREEWAY)
                DOSE5(4) = DOSE5(4) + DOS5
                DOSE6(4) = DOSE6(4) + DOS6
             ENDIF
         ENDIF
      ENDIF

  100 FORMAT (//,'   ERROR TOLERANCE NOT MET IN GAUS8.',/,
     *                20X,'IERR1 =',I2,/,20X,'IERR2 =',I2,/,
     *                20X,'IERR3 =',I2,/,20X,'IERR4 =',I2)
      RETURN
      END
*DECK,NORMAL
      SUBROUTINE NORMAL (LEG, MATNUM)
C
C     THIS ROUTINE CALCULATES THE INCIDENT-FREE DOSE TO ALL
C     POPULATION SUBROUPS FOR ALL TRANSPORTATION MODES.  PACKAGE
C     EXPOSURE RATES ARE CHECKED AGAINST REGULATORY LIMITS
C     AND AN EXCLUSIVE USE CHECK IS MADE FOR THE INDIVIDUAL
C     SHIPMENTS.
```

```
C
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
C
      COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
      REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
      LOGICAL RSPEC
      COMMON /NSEN/ PRMNDS(10,21),STERM,ATERM,BTERM,D4,RVELR,RVELS,RVELU
      COMMON /CON/ TOL,NDERV,NA,NC,TWOPI,PI,Q6,WS,GAM,FMU,DZ,VV
      COMMON /SENDOS/ DOSE2(3), DOSE5(13), DOSE6(9)
      DIMENSION FCS(2),FF(2),FRUSH(2),FFRWY(2)
      CHARACTER CZONE(3)
      DATA CZONE / 'R', 'S', 'U' /
C
C     RADIOLOGICAL CONSEQUENCES OF ACCIDENT-FREE TRANSPORTATION
C
      HDPKG = .5*DPKG
      DRAT1 = (1.0+HDPKG)/HDPKG
      ATERM = TI*PPS
C ... IF TRUCK, VAN, OR RAIL MODE COMPUTE MAXIMUM INDIVIDUAL DOSE
      IF (MODE.NE.3.AND.MODE.NE.4.AND.MODE.NE.5.AND.MODE.NE.6) THEN
        CALL BESSL (FMU*DZ,BK0,FK0)
        DOSMAX(LEG) = DOSMAX(LEG) +
     *                2.0E-6*CAYZER/VV*TI*PPS*SPY*(.009*BK0+FK0/DZ)
      ENDIF
      DO 20 I = 1, NDOSP1
   20   DOSEN(I) = 0.0
      DO 30 I = 1, 3
   30   DOSE2(I) = 0.0
      DO 40 I = 1, 13
   40   DOSE5(I) = 0.0
```

```
      DO 50 I = 1, 9
   50   DOSE6(I) = 0.0
      IF (DSTRVL.EQ.0.0) RETURN
      RVELR = 0.0
      RVELS = 0.0
      RVELU = 0.0
      IF (VELR.NE.0.)  RVELR = 1./VELR
      IF (VELS.NE.0.)  RVELS = 1./VELS
      IF (VELU.NE.0.)  RVELU = 1./VELU
      IF (LEG.EQ.1.AND.MATNUM.EQ.1)
     *                      WRITE (4,'(32X,''REGULATORY CHECKS'',//)')
C     CHECK TI PER PACKAGE
      IF (TI.GT.10.) GO TO 60
C     CHECK SURFACE DOSE RATE
      IF (TI*DRAT1.GT.200.) GO TO 60
C     CHECK TOTAL TI PER SHIPMENT
      IF (ATERM.GT.50.) GO TO 60
C     SHIPMENT IS NOT REQUIRED TO BE EXCLUSIVE USE
      IF (IEXCLU(MODE).EQ.0) GO TO 100
      WRITE (4,340) LABMAT(MAT), MODE
      GO TO 80
   60 CONTINUE
C     EXCLUSIVE USE CHECKS
      IF (IEXCLU(MODE).EQ.1) GO TO 70
      WRITE (4,350) MODE
      IEXCLU(MODE) = 1
   70 CONTINUE
   80 CONTINUE
      IF (MODE.EQ.5.OR.MODE.EQ.6) WRITE (4,360)
C     CHECK DOSE RATE AT THE SURFACE OF THE VEHICLE
      IF (ATERM*DRAT1.LE.200.) GO TO 90
      ATERM = 200./DRAT1
      WRITE (4,390) LABMAT(MAT),MODE,ATERM
   90 CONTINUE
C     CHECK DOSE RATE AT 2 METERS
C     CORRECTION TO REGULATORY CHECKS SEP 83
      DRAT2 = (1.0+HDPKG)/(2.0+HDPKG)
      IF (ATERM*DRAT2.LE.10.) GO TO 100
      ATERM = 10./DRAT2
      WRITE (4,400) LABMAT(MAT),MODE,ATERM
  100 CONTINUE
      BTERM = ATERM*SPY
      STERM = CAYZER*BTERM
C     DOSE TO WAREHOUSE PERSONNEL WHILE PACKAGE IS IN STORAGE
C     ALL MODES
      IF (RSTOR.EQ.0.0) THEN
        DOSEN(8) = 0.0
      ELSE
        DOSEN(8) = 1.E-3*CAYZER*SPY*TI*PPS*DTSTOR*PDSTOR/(RSTOR*RSTOR)
      ENDIF
C     DOSE TO HANDLERS
```

```
C     NORMAL DOSE TO HANDLERS -- DOSEN(4) IF EXCLUSIVE USE DOSEN(10)
      D4 = 0.
      IF (DPKG.GT.PKGSZ1) GO TO 110
      D4 = 2.5E-4*BTERM*HANDNO
      GO TO 130
  110 CONTINUE
      IF (DPKG.GT.PKGSZ2) GO TO 120
      D4 = 1.E-3*STERM/(DIHD*DIHD)*DTHD*PPH*HANDNO
      GO TO 130
  120 D4 = 1.E-3*BTERM*(SQRT(CAYZER)/DIHD)*DTHD*PPH*HANDNO
  130 CONTINUE
      IF (IEXCLU(MODE).EQ.1) GO TO 140
      DOSEN(4) = D4
      GO TO 150
C     EXCLUSIVE USE DOSE TO HANDLERS
  140 DOSEN(10) = D4
  150 CONTINUE
C     DOSE TO POPULATION DURING SHIPMENT STOPS
      TEST = DSTRVL*DTST+TIMZR
      IF (TEST.LT.STPMIN) TEST = STPMIN
      TRNSFR = FG*TRANS(1,RST)+FN*TRANS(2,RST)
      DOSEN(7) = 1.E-3*STERM*PDST*TEST/(RST*RST)
      DOSEN(7) = DOSEN(7)*TRNSFR
      RSHLD = 0.1
      IF (MODE.EQ.2) DOSEN(7) = 1.E-3*STERM*RSHLD*POPDS*1.E-6*TEST*23.0
C
      GO TO (160,270,320,320,260,260,160,160,160,160), MODE
      CALL GOTOER ('NORMAL-1')
C
C     TRUCK/VAN
  160 CONTINUE
      IF (RSPEC) THEN
          CALL NORLNK (LDIST(LEG),LVDEN(LEG),LSPED(LEG)/3.6,LPOPD(LEG),1
     *                ,LTYPE(LEG),CZONE(LZONE(LEG)),STERM,PPV,DOS5,DOS6)
          DOSEN(5)=DOS5
          DOSEN(6)=DOS6
      ELSE
C  ...     FCS is Fraction of travel on Urban Non-City Streets/City Streets
C  ...     FF is Fraction of travel Rural and Suburban Freeway/Non Freeway
C  ...     FFRWY is Fraction of Travel for any zone on Freeway/Non Freeway
C  ...     FRUSH is Fraction of Travel during Non-Rush Hour/Rush Hour
C  ...     IZONE = 1 => RURAL
C                  2 => SUBURBAN
C                  3 => URBAN
C          IRUSH = 1 => NON-RUSH HOUR
C                  2 => RUSH HOUR
C          IRDTYP = 1 => FREEWAY
C                   2 => NON-FREEWAY
C  ...
          FCS(2) = DNORML(MODE,21)
          FCS(1) = 1.-FCS(2)
```

```
            FF(1) = DNORML(MODE,22)
            FF(2) = 1.-FF(1)
            FRUSH(2) = DNORML(MODE,20)
            FRUSH(1) = 1.-FRUSH(2)
            DOSEN(5) = 0.0
            DOSEN(6) = 0.0
            DO 170 IZONE = 1,3
            DO 170 IRUSH = 1,2
            DO 170 IRDTYP = 1,2
               IF (IZONE .NE. 3) THEN
                   FFRWY(IRDTYP) = FF(IRDTYP)
               ELSE
                   FFRWY(IRDTYP) = FCS(IRDTYP)
               ENDIF
               DIS = DSTRVL*DNORML(MODE,IZONE)*FFRWY(IRDTYP)*FRUSH(IRUSH)
C  ...         On freeways the speed is rural velocity
               IF (IRDTYP .EQ. 1) THEN
                   SPD = DNORML(MODE,4)
               ELSE
                   SPD = DNORML(MODE,3+IZONE)
               ENDIF
               IF (IZONE .NE. 1) THEN
C  ...              For non-rural, the speed slows by half if rush hour
(IRUSH=2)
C  ...              and the vehicle density doubles if rush hour
                    SPD = SPD/IRUSH
                    VDN = DNORML(MODE,22+IZONE)*IRUSH
               ELSE
                    VDN = DNORML(MODE,22+IZONE)
               ENDIF

C  ...         Speed must be converted from km/hr to meters/sec
               CALL NORLNK (DIS,VDN,SPD/3.6,POPDEN(IZONE),1,IRDTYP,
     *                      CZONE(IZONE),STERM,PPV,DOS5,DOS6)

C  .......     GET SENSITIVITY DATA FOR RUSH HOUR VARIABLE
               IF ((ISEN .GE. 3) .AND. (IRUSH .EQ. 2)) THEN
                  DOSE5(9) = DOSE5(9) + DOS5
                  DOSE6(9) = DOSE6(9) + DOS6
               ENDIF

               DOSEN(5)=DOSEN(5)+DOS5
               DOSEN(6)=DOSEN(6)+DOS6
  170      CONTINUE
           ENDIF
  210 CONTINUE
C     DOSE TO CREWMEN -- TRUCK, VAN, AIR
      IF (MODE.EQ.5) THEN
        DOSEN(5) = 0.0
        DOSEN(6) = 0.0
      ENDIF
```

```
C     NORMAL DOSE TO CREW -- DOSEN(2);IF EXCLUSIVE USE, DOSEN(9)
      IF (ADSTCW.EQ.0.) RETURN
      IF (VELM.EQ.0.) RETURN
      R1 = ADSTCW
      R = ADSTCW-HDPKG
      CHKTRM = CAYZER*TI*PPS/(R1*R1)*(FG*TRANS(1,R)+FN*TRANS(2,R))
      IF (CHKTRM.LE.2.) GO TO 230
      IF (IEXCLU(MODE).EQ.0) GO TO 220
C     EXCLUSIVE-USE SHIPMENT DOSE RATE .GT. 2.
      WRITE (4,370) LABMAT(MAT),MODE,CHKTRM
      CHKTRM = 2.
      GO TO 240
C     NON-EXCLUSIVE-USE SHIPMENT DOSE RATE .GT. 2.
  220 WRITE (4,380) LABMAT(MAT),MODE,CHKTRM
      GO TO 250
  230 CONTINUE
      IF (IEXCLU(MODE).EQ.0) GO TO 250
  240 CONTINUE
      DOSEN(9) = 2.8E-4*CHKTRM*SPY*CREWNO*DSTRVL/VELM
      GO TO 255
  250 CONTINUE
      DOSEN(2) = 2.8E-4*CHKTRM*SPY*CREWNO*DSTRVL/VELM
  255 CONTINUE
C ... GET SENSITIVITY DATA
      IF (ISEN .GE. 3) THEN
          DOS2 = DOSEN(2) + DOSEN(9)
          DOSE2(1) = DOS2*VELM*FTZNR*RVELR
          DOSE2(2) = DOS2*VELM*FTZNS*RVELS
          DOSE2(3) = DOS2*VELM*FTZNU*RVELU
      ENDIF
      RETURN
C
  260 CONTINUE
      IF (MODE.EQ.5) GO TO 210
C     PASSENGER AIR
C     DOSE TO AIRLINE PASSENGERS
      RAT = DSTRVL/VELM
      DOSEN(1) = 8.3E-6*BTERM*RAT*PPV
C     DOSE TO FLIGHT ATTENDANTS
      DOSEN(3) = 8.3E-6*BTERM*RAT*FNOATT
      GO TO 210
C
C     RAIL
  270 CONTINUE
      IF (RSPEC) THEN
          CALL NORLNK (LDIST(LEG),LVDEN(LEG),LSPED(LEG)/3.6,LPOPD(LEG),2
     *                ,LTYPE(LEG),CZONE(LZONE(LEG)),STERM,PPV,DOS5,DOS6)
          DOSEN(5)=DOS5
          DOSEN(6)=DOS6
      ELSE
          DOSEN(5)=0.0
```

```
          DOSEN(6)=0.0
          DO 280 IZONE= 1,3
             DIS=DSTRVL*DNORML(MODE,IZONE)
C  ....     Speed entered is km/hr, must convert to meters/sec
             SPD=DNORML(MODE,3+IZONE)/3.6
             VDN=DNORML(MODE,22+IZONE)
             CALL NORLNK (DIS,VDN,SPD,POPDEN(IZONE),MODE,0,
     *                    CZONE(IZONE),STERM,PPV,DOS5,DOS6)
             DOSEN(5)=DOSEN(5)+DOS5
             DOSEN(6)=DOSEN(6)+DOS6
  280     CONTINUE
      ENDIF
C     NORMAL DOSE TO CREW -- RAIL
      CH = CH1
      IF (ITRAIN.EQ.2) CH = CH2
      IF (IEXCLU(MODE).EQ.1) GO TO 310
      DOSEN(2) = 1.E-3*CH*(1.+HDPKG)*BTERM*(FMINCL+.0018*DSTRVL)
      RETURN
  310 DOSEN(9) = 1.E-3*CH*(1.+HDPKG)*BTERM*(FMINCL+.0018*DSTRVL)
      RETURN
C
C     WATER
  320 CONTINUE
C     DOSE TO PERSONS SURROUNDING THE TRANSPORT LINK WHILE THE
C     SHIPMENT IS MOVING (OFF LINK)
      DOSEN(5) = 2.8E-10*TWOPI*DSTRVL*STERM*1.61*(RR*FTZNR*POPDR*RVELR
     *           +RS*FTZNS*POPDS*RVELS+RU*FTZNU*POPDU*RVELU)
C     NORMAL DOSE TO CREW -- WATER
      IF (IEXCLU(MODE).EQ.1) GO TO 330
      DOSEN(2) = 1.E-7*(1.+HDPKG)*BTERM*DSTRVL/VELM
      RETURN
  330 DOSEN(9) = 1.E-7*(1.+HDPKG)*BTERM*DSTRVL/VELM
      RETURN
C
  340 FORMAT (//5X,'THE SHIPMENT OF ',A10,' BY MODE',I3,
     *        ' IS DESIGNATED AS EXCLUSIVE USE',/5X,'BUT IS NOT ',
     *        'REQUIRED TO BE SO DESIGNATED BY REGULATIONS')
  350 FORMAT (//5X,'MODE',I3,' HAS BEEN REDESIGNATED AS EXCLUSIVE USE')
  360 FORMAT (//,5X,'!!!! WARNING !!!! 49CFR173 DOES NOT PERMIT ',
     *      'EXCLUSIVE USE AIR SHIPMENTS',/5X,'COMPUTATIONS PROCEED...')
  370 FORMAT (//5X,'FOR THE SHIPMENT OF ',A10,' BY MODE',I3,/5X,
     *        'THE DOSE RATE IN THE CREW COMPARTMENT COULD EXCEED ',
     *        '2 MREM/HR',/5X,'THE DOSE RATE HAS BEEN RESET FROM',F6.2,
     *        ' TO 2 FOR CREW CALCULATIONS')
  380 FORMAT (//5X,'FOR THE SHIPMENT OF ',A10,' BY MODE',I3,/5X,
     *        'WARNING -- THE DOSE RATE IN THE CREW COMPARTMENT COULD ',
     *        'EXCEED 2 MREM/HR',/5X,'THE COMPUTED VALUE OF',1PE10.2,
     *        ' WILL BE USED FOR THE CREW CALCULATIONS')
  390 FORMAT (//5X,'FOR THE SHIPMENT OF ',A10,' BY MODE',I3,/5X,
     *        'THE SURFACE DOSE RATE COULD EXCEED 200 MREM/HR',/5X,
     *        'PPS*TI HAS BEEN RESET TO EQUAL ',F6.2)
```

```
  400 FORMAT (//5X,'FOR THE SHIPMENT OF ',A10,' BY MODE',I3,/5X,
     *        'THE DOSE RATE AT 2 METERS COULD EXCEED 10 MR/HR',/5X,
     *        'PPS*TI HAS BEEN RESET TO EQUAL ',F6.2)
      END
*DECK,PAIRS
      SUBROUTINE PAIRS
C      PROGRAM PAIRS
C
C     THIS SUBROUTINE (PROGRAM) READS IN DATA FROM FILE6 PRODUCED BY
C     RADTRAN 4 AND PRODUCES 3 FILES OF ORDERED PAIRS OF NUMBERS.
C     THE FIRST NUMBER IS A CONSEQUENCE AND THE SECOND IS A SUMMED
C     PROBABILITY.  THE SUMMED PROBABILITIES INDICATE THE PROBABILTY
C     OF A PARTICULAR CONSEQUENCE OR WORSE.  CONSEQUENCES WITH
C     ZERO PROBABILTIES ARE IGNORED AND NOT PRINTED.
C     1. FILE10 CONTAINS SORTED LCF OR PERSON-REM CONSEQUENCES WITH
C        SUMMED PROBABILITIES.
C     2. FILE11 CONTAINS SORTED GENETIC EFFECTS CONSEQUENCES WITH
C        SUMMED PROBABILITIES.
C     3. FILE12 CONTAINS SORTED ECONOMIC IMPACT CONSEQUENCES WITH
C        SUMMED PROBABILITIES.
C

      REAL CONLCF(14400), CONGE(14400), CONEC(14400)

C .... CONLCF > CONSEQUENCE OF LCF (OR PERSON-REM)
C .... CONGE  > CONSEQUENCE OF GENETIC EFFECTS
C .... CONEC  > CONSEQUENCE OF ECONOMIC IMPACT
C .... 20*3*40*6=14400     SEVERITIES*POP ZONES*LINKS*PASQUILL CATEGORIES

      REAL CLCF(60), CGE(60)
C .... TEMPORARY VARS TO HOLD EACH ISOTOPE'S NSEV*3 VALUES FOR LCF AND GE

      REAL PRLCF(14400), PRGE(14400), PRECON(14400)
C .... PRLCF  > PROBABILITY OF LCFs (OR PERSON-REM)
C .... PRGE   > PROBABILITY OF GENETIC EFFECTS
C .... PRECON > PROBABILITY OF ECONOMIC IMPACT

      CHARACTER TEST*10
      LOGICAL EOF / .FALSE. /

      REAL DUMMY(60), PROB(60), PASQPR
C .... PROB HOLDS THE PROBABILTY FOR A LEG
C .... PASQPR HOLDS THE PASQUILL PROBABILITY (WILL BE 1.0 IF
C .... PASQUILL NOT USED IN RADTRAN)

C .... COMMENT OUT NEXT LINE IF USED AS A SUBROUTINE INSIDE RADTRAN
C      OPEN (UNIT=6, STATUS='OLD', FILE='FILE6.DAT')
      REWIND (6)
      OPEN (UNIT=10, STATUS='NEW', FILE='FILE10.DAT')
      OPEN (UNIT=11, STATUS='NEW', FILE='FILE11.DAT')
      OPEN (UNIT=12, STATUS='NEW', FILE='FILE12.DAT')
```

```
   10  CONTINUE
C ...  CONTINUE READING FILE6 - FOR SEPARATE RUNS

       READ (6,'(I5)',END=1000) NSEV
       MAX = NSEV*3
C .... MAX EQUALS THE NUMBER OF VALUES IN THE ARRAYS IN FILE6
       LOOP = -1
C .... LOOP KEEPS TRACK OF THE NUMBER OF DIFFERENT OUTPUT GROUPS (DIFFERENT
C           MODES AND/OR DIFFERENT PASQUILL CATEGORIES) -- STARTS AT ZERO
       DO 15 I = 1, 14400
         CONLCF(I) = 0.0
         CONGE(I) = 0.0
   15  CONTINUE

       IF (EOF) THEN
C ..... WRITE EOF TO THE OUTPUT FILES --- VARIABLE EOF WILL NOT BE TRUE
UNLESS
C ..... A DATA SET HAS BEEN READ -- IF THERE IS ONLY ONE DATA SET OR IT IS
C ..... THE LAST DATA SET THEN THE ABOVE READ WILL HAVE FOUND THE END OF
FILE
C ..... MARKER AND WOULD SKIP THIS AND GO TO THE END -- IN THIS WAY THERE
WILL
C ..... BE NO ENDING EOF AND NO EOF AT ALL FOR JUST ONE DATA SET
         WRITE (10,'('' EOF'')')
         WRITE (11,'('' EOF'')')
         WRITE (12,'('' EOF'')')
         EOF = .FALSE.
       ENDIF

   20  CONTINUE
C ...  CONTINUE READING DATA FROM FILE6 WHILE NOT EOF

        READ (6,'(A10)',END=1000) TEST

        IF (TEST.EQ.'       EOF') THEN
          EOF = .TRUE.

        ELSEIF (TEST.EQ.'       LEG') THEN
          READ (6,'()')
          READ (6,'()')
C . . . . SKIP THE ACCRAT ARRAY -- NOT USED FOR THIS APPLICATION
          READ (6, 100) (DUMMY(I),I = 1, MAX)
          READ (6,'()')
          READ (6, 100) (PROB(I),I = 1, MAX)

        ELSEIF (TEST.EQ.'  PASQUILL') THEN
          READ (6,'(I6,1PE10.2)') IDUM, PASQPR
          LOOP = LOOP + 1

        ELSEIF (TEST.EQ.'   ISOTOPE') THEN
```

```
            READ (6,'()')
            READ (6,'()')
            READ (6,'()')
            READ (6, 100) (CLCF(I), I = 1, MAX)
            READ (6,'()')
            READ (6, 100) (CGE(I), I = 1, MAX)

            DO 30 I = 1, MAX
              PP = PROB(I)*PASQPR
              INDEX = LOOP*MAX+I
              PRLCF(INDEX) = PP
              PRGE(INDEX) = PP
C . . . . . SUM THE COSEQUENCES OVER ISOTOPES
              CONLCF(INDEX) = CONLCF(INDEX) + CLCF(I)
              CONGE(INDEX) = CONGE(INDEX) + CGE(I)
   30       CONTINUE

          ELSEIF (TEST.EQ.'  ECONOMIC') THEN
            READ (6, 100) (CONEC(LOOP*MAX+I),I = 1, MAX)

            DO 40 I = 1, MAX
              PRECON(LOOP*MAX+I) = PROB(I)*PASQPR
   40       CONTINUE

          ELSE
            WRITE (10,'(A)') '  ERROR IN FILE6'
            WRITE (11,'(A)') '  ERROR IN FILE6'
            WRITE (12,'(A)') '  ERROR IN FILE6'
            STOP
          ENDIF

        IF (.NOT.EOF) GOTO 20

C .... FOUND END OF A DATA SET IN FILE6

C
        LENGTH = (LOOP+1)*MAX
C .... SORT BY CONSEQUENCE IN DECREASING ORDER CARRYING THE PROBABILITY
ALONG
        CALL SSORT (CONLCF, PRLCF, LENGTH, -2)
        CALL SSORT (CONGE, PRGE, LENGTH, -2)
        CALL SSORT (CONEC, PRECON, LENGTH, -2)

C .... SUM1 AND SUM2 HOLD PROBABILITY SUMS
        SUM1 = 0.0
        SUM2 = 0.0
        DO 50 I = 1, LENGTH
          IF (PRLCF(I).NE.0.0) THEN
C . . . . .SKIP IT IF THE PROBABILTY IS ZERO
            SUM1 = SUM1 + PRLCF(I)
            WRITE (10,200) CONLCF(I), SUM1
```

```
          ENDIF
          IF (PRGE(I).NE.0.0) THEN
C . . . . .SKIP IT IF THE PROBABILTY IS ZERO
            SUM2 = SUM2 + PRGE(I)
            WRITE (11,200) CONGE(I), SUM2
          ENDIF
   50   CONTINUE

        SUM1 = 0.0
        DO 60 I = 1, LENGTH
          IF (PRECON(I).NE.0.0) THEN
C . . . . .SKIP IT IF THE PROBABILTY IS ZERO
            SUM1 = SUM1 + PRECON(I)
            WRITE (12,200) CONEC(I), SUM1
          ENDIF
   60   CONTINUE

        GOTO 10

 1000   CONTINUE
C .... END OF FILE MARKER WAS READ ....

   100  FORMAT (8(E10.3))
   200  FORMAT (2(1PE10.2))
C        STOP
        RETURN
        END
*DECK,PASQ
      SUBROUTINE PASQ (JC)
C
C     THIS ROUTINE LOADS THE AREAS AND CONCENTRATIONS FOR THE JC-TH
C     PASQUILL CATEGORY INTO THE AREADA AND DFLEV ARRAYS.
C
C
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
```

```
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
C
      COMMON /CON/ TOL,NDERV,NA,NC,TWOPI,PI,Q6,WS,GAM,FMU,DZ,VV
C
C     PASQUILL AREA DATA
C     SET NAREAS TO NA BECAUSE NAREAS MIGHT HAVE BEEN CHANGED BY USER
      NAREAS = NA
      DO 10 I = 1, NA
        AREADA(I) = PSAREA(I)
        DFLEV(I) = PSQLCT(I,JC)
   10 CONTINUE
      ASIZE(1) = AREADA(1)
      DO 20 I = 2, NA
        ASIZE(I) = AREADA(I) - AREADA(I-1)
   20 CONTINUE
      RETURN
      END
*DECK,PRCHRD
      SUBROUTINE PRCHRD
C
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
      COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
      REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
      LOGICAL RSPEC
C
      CHARACTER*3 NOYES(0:1)
      DATA NOYES / ' NO', 'YES' /

      CALL NEWPAG
C     POPULATION ZONES
```

```
      IF (.NOT.RSPEC)  WRITE (4,40) (LABPOP(I),POPDEN(I),I=1,NPOP)
C     PACKAGE SIZE AND COEFFICIENTS
      WRITE (4,50) (LABMAT(I),PKGSIZ(I),EFFSIZ(I),PKGCOE(I),I=1,MCOUNT)
      WRITE (4,60)
      WRITE (4,70) PKGSZ1,PKGSZ2
      WRITE (4,80) (LABMAT(I),FRGAMA(I),FRNEUT(I),I=1,MCOUNT)
      CALL NEWPAG
      IF (RSPEC) THEN
        WRITE (4,90)
        DO 10 I = 1, NMODE
          M = MODSAV(I)
          WRITE (4,100) LABMOD(M),NOYES(IEXCLU(M)),TABSPY(M)
          WRITE (4,110) (LABMAT(MATSAV(M,J)),TIPKG(M,MATSAV(M,J)),
     *                   PKGSHP(M,MATSAV(M,J)),J=1,NMAT(M))
   10   CONTINUE
      ELSE
        WRITE (4,120)
        DO 20 I = 1, NMODE
          M = MODSAV(I)
          WRITE (4,130) LABMOD(M),DISTKM(M),NOYES(IEXCLU(M)),TABSPY(M)
          WRITE (4,140) (LABMAT(MATSAV(M,J)),TIPKG(M,MATSAV(M,J)),
     *                   PKGSHP(M,MATSAV(M,J)),J=1,NMAT(M))
   20   CONTINUE
      ENDIF
      IF (NMODE .EQ. 6)  WRITE (4,150) FNOATT
      WRITE (4,160) IUOPT,RPD,RR,RS,RU
      RETURN
   40 FORMAT (///12X,'ZONE',5X,'POPULATION DENSITY'/21X,
     *               '(PERSONS PER SQ KM)'/,(10X,A10,4X,F10.0))
   50 FORMAT (///28X,'PACKAGE CHARACTERISTICS'//14X,
     *  '  FOR          DIMENSION      EFFECTIVE       K(0)'/14X,
     *  'MATERIAL       (METERS)       DIMENSION      METERS SQ.'/
     *   (12X,A10,3(5X,1PE10.3)))
   60 FORMAT (/5X,'K(0) IS TI TO DOSE RATE CONVERSION FACTOR')
   70 FORMAT (///5X,'PACKAGE HANDLING THRESHOLDS (METERS)'/
     *        10X,'PKGSZ1=',1PE10.3/10X,'PKGSZ2=',1PE10.3/
     *        10X,'PACKAGES .LE. PKGSZ1 ARE HAND CARRIED'/
     *        10X,'PACKAGES .GT. PKGSZ1 AND .LE. PKGSZ2 ARE HANDLED ',
     *         'BY SMALL EQUIPMENT'/10X,'PACKAGES .GT. PKGSZ2 ARE ',
     *         'HANDLED BY HEAVY EQUIPMENT')
   80 FORMAT (///28X,'MATERIAL CHARACTERISTICS'//19X,
     *        '                FRACTION       FRACTION'/19X,
     *        'MATERIAL        OF GAMMA     OF NEUTRON',/
     *        (17X,A10,2(5X,1PE10.3)))
   90 FORMAT (30X,'MODE CHARACTERISTICS',//12X,
     *       'MODE  EXCLUSIVE  NUMBER OF  MATERIALS  ',
     *       'TRANSPORT   PACKAGES/'/12X,'          USE     ',
     *       'SHIPMENTS              INDEX (TI)   SHIPMENT'/)
  100 FORMAT (6X,A10,5X,A3,4X,1PE10.2)
  110 FORMAT (39X,A10,2X,1PE10.2,1X,1PE10.2)
  120 FORMAT (30X,'MODE CHARACTERISTICS',//7X,
```

```
     *       'MODE   DISTANCE  EXCLUSIVE  NUMBER OF  MATERIALS  ',
     *       'TRANSPORT   PACKAGES/'/7X,'       TRAVELED     USE     ',
     *       'SHIPMENTS              INDEX (TI)   SHIPMENT'/)
  130 FORMAT (1X,A10,1X,1PE10.2,5X,A3,4X,1PE10.2)
  140 FORMAT (45X,A10,2X,1PE10.2,1X,1PE10.2)
  150 FORMAT (///' NUMBER OF ATTENDANTS PER PASSENGER FLIGHT IS',F3.0)
  160 FORMAT (//' BUILDING SHIELDING OPTION=',I2/' (1=TOTAL SHIELDING, '
     *   ,'2=PARTIAL SHIELDING, 3=NO SHIELDING)'//' RPD=',1PE10.3/
     *   ' (RATIO OF PEDESTRIAN DENSITY (PEDESTRIAN/KM SQ OF SIDEWALK)'/
     *   '  TO POPULATION DENSITY (PEOPLE/KM SQ IN URBAN AREAS)'
     *   //' RR =',1PE10.3/' (TRANSMISSION FACTOR FOR RURAL AREAS)'
     *   //' RS =',1PE10.3/' (TRANSMISSION FACTOR FOR SUBURBAN AREAS)'
     *   //' RU =',1PE10.3/' (TRANSMISSION FACTOR FOR URBAN AREAS)')
      END
*DECK,PRCONS
      SUBROUTINE PRCONS (AOUT,BOUT,COUT,DOUT,EOUT)
C
C     PRINTS CONSEQUENCE DATA TABLES
C
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
      REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
```

```
      LOGICAL RSPEC

      REAL AOUT(20,40,10),BOUT(20,40,10),COUT(20,40,10),DOUT(20,40,10),
     *     EOUT(20,40,10)

      CALL NEWPAG
      WRITE (4,'(32X,''ACCIDENT SUMMARY'',/,32X,''******** *******'')')

      IF (RSPEC) THEN
        CALL PROUTC ('NUMBER OF EXPECTED ACCIDENTS$',' $',AOUT(1,1,1))
        IF (NLNK .GT. 14)  CALL NEWPAG
        CALL PROUTC ('EARLY FATALITY CONSEQUENCES$',' $',BOUT(1,1,1))
        CALL NEWPAG
        CALL PROUTC ('ECONOMIC CONSEQUENCES$',' $',EOUT(1,1,1))
        IF (NLNK .GT. 14)  CALL NEWPAG
        IF (URSKF) THEN
          CALL PROUTC ('RADIOLOGICAL CONSEQUENCES$',
     *              '50 YEAR POPULATION DOSE IN PERSON REM$',COUT(1,1,1))
        ELSE
          CALL PROUTC ('RADIOLOGICAL CONSEQUENCES$',
     *                 'TOTAL LATENT CANCER FATALITIES$',COUT(1,1,1))
          CALL NEWPAG
          CALL PROUTC ('EARLY MORBIDITY CONSEQUENCES$',' $',DOUT(1,1,1))
        ENDIF
      ELSE
C ..... NOT ROUTE-SPECIFIC  (PRINT TABLE FOR EACH MODE)
        KOUNT = 0
        DO 10 K = 1, NMODE
          M = MODSAV(K)
          KOUNT = KOUNT+6+NSEV
          IF (KOUNT.GT.50) THEN
            CALL NEWPAG
            KOUNT = 6+NSEV
          ENDIF
          CALL PROUTC ('NUMBER OF EXPECTED ACCIDENTS -- MODE '//
     *                  LABMOD(M)//'$',' $',AOUT(1,1,M))
   10   CONTINUE
        DO 20 K = 1, NMODE
          M = MODSAV(K)
          KOUNT = KOUNT+6+NSEV
          IF (KOUNT.GT.50) THEN
            CALL NEWPAG
            KOUNT = 6+NSEV
          ENDIF
          CALL PROUTC ('EARLY FATALITY CONSEQUENCES -- '//
     *                 'MODE '//LABMOD(M)//'$',' $',BOUT(1,1,M))
   20   CONTINUE
        DO 30 K = 1, NMODE
          M = MODSAV(K)
          KOUNT = KOUNT+6+NSEV
          IF (KOUNT.GT.50) THEN
```

```
            CALL NEWPAG
            KOUNT = 6+NSEV
          ENDIF
          CALL PROUTC ('ECONOMIC CONSEQUENCES -- MODE '//LABMOD(M)//'$',
     *                 ' $',EOUT(1,1,M))
   30   CONTINUE
        IF (URSKF) THEN
          DO 40 K = 1, NMODE
            M = MODSAV(K)
            KOUNT = KOUNT+6+NSEV
            IF (KOUNT.GT.50) THEN
               CALL NEWPAG
              KOUNT = 6+NSEV
            ENDIF
            CALL PROUTC ('RADIOLOGICAL CONSEQUENCES -- MODE '//
     *                   LABMOD(M)//'$',
     *             '50 YEAR POPULATION DOSE IN PERSON REM$',COUT(1,1,M))
   40     CONTINUE
        ELSE
          DO 50 K = 1, NMODE
            M = MODSAV(K)
            KOUNT = KOUNT+6+NSEV
            IF (KOUNT.GT.50) THEN
              CALL NEWPAG
              KOUNT = 6+NSEV
            ENDIF
            CALL PROUTC ('RADIOLOGICAL CONSEQUENCES -- MODE '//
     *                   LABMOD(M)//'$',
     *                 'TOTAL LATENT CANCER FATALITIES$',COUT(1,1,M))
   50     CONTINUE
          DO 60 K = 1, NMODE
            M = MODSAV(K)
            KOUNT = KOUNT+6+NSEV
            IF (KOUNT.GT.50) THEN
              CALL NEWPAG
              KOUNT = 6+NSEV
            ENDIF
            CALL PROUTC ('EARLY MORBIDITY CONSEQUENCES -- '//
     *                   'MODE '//LABMOD(M)//'$',' $',DOUT(1,1,M))
   60     CONTINUE
        ENDIF
      ENDIF
      RETURN
      END
*DECK,PRISTP
      SUBROUTINE PRISTP
C
C   PRINTS DATA RELATED TO EACH ISOTOPE.
C   IF RUNNING UNIT PRINTS EFFECTIVE INHALATION AND INGESTION REM/CI.
C   IF RUNNING NON-UNIT PRINTS ORGAN SPECIFIC INHALATION AND INGESTION
C   REM PER CURIE.
```

```
C
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
C
C ... EACH PAIR OF CHARACTER STRINGS LABELS COLUMNS IN A TABLE
      CHARACTER*80 FMT1,FMT2,FMT3,FMT4,FMT7,FMT8,FMT9,FMT10
      CHARACTER*32 FMT5,FMT6,TITLE1,TITLE2,TITLE3
      TITLE1(1:32) = '      ISOTOPE RELATED DATA      '
      TITLE2(1:32) = '    INHALATION REM PER CURIE    '
      TITLE3(1:32) = '    INGESTION REM PER CURIE     '
      FMT1(1:80) = '    NUCLIDE   CURIES      RELEASE     '//
     *   'RESUSP   LUNG  DISPERS.  1YR INHAL REM/CI'
      FMT2(1:80) = '                PER PKG       GROUP      '//
     *   'FACTOR   TYPE  CATEGORY   LUNG     MARROW'
      FMT3(1:80) = '    NUCLIDE     HALF      GAMMA     CLOUD '//
     *   '         TRANSFER        DEPOS'
      FMT4(1:80) = '                 LIFE      ENERGY    FACTOR'//
     *   '     CROPS     SOIL     SPEED'
      FMT5(1:32) = '    NUCLIDE     1-YR        1-YR'
```

```
      FMT6(1:32) = '                LUNG      MARROW'
      FMT7(1:80) = '    NUCLIDE   50-YR     50-YR     50-YR  '//
     *   '   50-YR     50-YR     50-YR'
      FMT8(1:80) = '                GONADS     LLI     THYROID '//
     *   '   BONE      LUNG      MARROW'
      FMT9(1:80) = '    NUCLIDE   50-YR EFFECTIVE REM/CI'
      FMT10(1:80)= '                   INHALE    INGEST'

      CALL NEWPAG
      WRITE (4,'(24X,A,//)') TITLE1
      WRITE (4,'(A,/,A,/)') FMT1, FMT2
      DO 20 M = 1, MCOUNT
        WRITE (4,'('' '',A10)') LABMAT(M)
        DO 20 I = 1, NIS(M)
C          SETISO
          K = ISOSAV(M,I)
          L = LIBSAV(K)
          WRITE (4,100) LABISO(L),CIPKG(K),LABGRP(IPCGRP(K)),
     *         TABRDF(L),LNGTAB(L),IDISP(K),RPCVAL(L,1),RPCVAL(L,2)
   20 CONTINUE

      IF (ICOUNT+MCOUNT.GT.20)  CALL NEWPAG

      WRITE (4,'(///A,/,A,/)') FMT3, FMT4
      DO 30 M = 1, MCOUNT
        WRITE (4,'('' '',A10)') LABMAT(M)
        DO 30 I = 1, NIS(M)
C          SETISO
          K = ISOSAV(M,I)
          L = LIBSAV(K)
          WRITE (4,'(1X,A10,6(1PE10.2))') LABISO(L),TABHLF(L),
     *          PHTENG(L),CLDOSF(L),FDTRAN(L),SLTRAN(L),VELDEP(L)
   30 CONTINUE
C
C ... FOR UNIT ONLY PRINT EFFECTIVE (WHOLE BODY) REM PER CURIE
      IF (URSKF) THEN
        IF (MCOUNT+ICOUNT.GT.11) THEN
          CALL NEWPAG
          WRITE (4,'(24X,A,///,A,/,A,/)')  TITLE1, FMT9, FMT10
        ELSE
          WRITE (4,'(///A,/,A,/)') FMT9, FMT10
        ENDIF
        DO 40 M = 1, MCOUNT
          WRITE (4,'('' '',A10)') LABMAT(M)
          DO 40 I = 1, NIS(M)
C            SETISO
            K = ISOSAV(M,I)
            L = LIBSAV(K)
            WRITE (4,'(1X,A10,3X,2(1PE10.2))') LABISO(L),
     *                RPCVAL(L,4),INGVAL(L,4)
   40   CONTINUE
```

```
      ELSE
C ..... NON-UNIT -- PRINT INHALATION AND INGESTION FOR EACH ORGAN
C
C ..... INHALATION TABLES
C
        CALL NEWPAG
        WRITE (4,'(24X,A,//,24X,A,//,A,/,A,/)')
     *               TITLE1, TITLE2, FMT5, FMT6
        DO 45 M = 1, MCOUNT
          WRITE (4,'('' '',A10)') LABMAT(M)
          DO 45 I = 1, NIS(M)
C            SETISO
            K = ISOSAV(M,I)
            L = LIBSAV(K)
            WRITE (4,'(1X,A10,6(1PE10.2))') LABISO(L),
     *              (RPCVAL(L,J), J=1,2)
   45   CONTINUE

        IF (MCOUNT+ICOUNT.GT.20) THEN
          CALL NEWPAG
          WRITE (4,'(24X,A,//,24X,A,//,A,/,A,/)')
     *               TITLE1, TITLE2, FMT7, FMT8
        ELSE
          WRITE (4,'(//,A,/,A,/)') FMT7, FMT8
        ENDIF
        DO 50 M = 1, MCOUNT
          WRITE (4,'('' '',A10)') LABMAT(M)
          DO 50 I = 1, NIS(M)
C            SETISO
            K = ISOSAV(M,I)
            L = LIBSAV(K)
            WRITE (4,'(1X,A10,6(1PE10.2))') LABISO(L),
     *              (RPCVAL(L,J), J=3,8)
   50   CONTINUE
C
C ..... INGESTION TABLES
C
        CALL NEWPAG
        WRITE (4,'(24X,A,//,24X,A,//,A,/,A,/)')
     *              TITLE1, TITLE3, FMT5, FMT6
        DO 60 M = 1, MCOUNT
          WRITE (4,'('' '',A10)') LABMAT(M)
          DO 60 I = 1, NIS(M)
C            SETISO
            K = ISOSAV(M,I)
            L = LIBSAV(K)
            WRITE (4,'(1X,A10,2(1PE10.2))') LABISO(L),
     *              (INGVAL(L,J), J=1,2)
   60   CONTINUE
```

```
      IF (MCOUNT+ICOUNT.GT.20) THEN
        CALL NEWPAG
        WRITE (4,'(24X,A,//,24X,A,//,A,/,A,/)')
     *                 TITLE1, TITLE3, FMT7, FMT8
      ELSE
        WRITE (4,'(//,A,/,A,/)') FMT7, FMT8
      ENDIF

      DO 70 M = 1, MCOUNT
        WRITE (4,'('' '',A10)') LABMAT(M)
        DO 70 I = 1, NIS(M)
C          SETISO
          K = ISOSAV(M,I)
          L = LIBSAV(K)
          WRITE (4,'(1X,A10,6(1PE10.2))') LABISO(L),
     *             (INGVAL(L,J), J=3,8)
   70   CONTINUE
      ENDIF

      RETURN
  100 FORMAT(1X,A10,1PE10.2,2X,A10,2X,1PE10.2,4X,I1,4X,I2,5X,2(1PE9.2))
      END
*DECK,PRNORD
      SUBROUTINE PRNORD
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10

      CALL NEWPAG
      WRITE (4,30) (LABMOD(MODSAV(K)), K = 1, NMODE)
      DO 20 I = 1, NDLAB
        WRITE (4,40) I,LABDOS(I,1),(DNORML(MODSAV(K),I), K = 1,NMODE)
        WRITE (4,50) LABDOS(I,2)
   20 CONTINUE
      WRITE (4,60)
      RETURN
   30 FORMAT (30X,'DNORML INPUT',/2X,2HNO,32X,10A10)
   40 FORMAT (2X,I2,2X,A30,10(1PE10.3))
```

```
   50 FORMAT (6X,A30 )
   60 FORMAT (1X,'*(ONE WAY VEHICLES/HR)')
      END
*DECK,PROUTA
      SUBROUTINE PROUTA
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
C
      COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
      REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
      LOGICAL RSPEC

      COMMON /CRSUM/ RSUM(8)

      DIMENSION STOT(3,8), SUM(8)
      CHARACTER*10 NAME
      SAVE NAME
      DATA NAME / '          ' /

      CALL NEWPAG
      WRITE (4,'(30X,A21)') 'INCIDENT-FREE SUMMARY'
      WRITE (4,'(30X,A21)') '******** **** *******'
      DO 20 J = 1, 8
        SUM(J) = 0.0
        DO 20 I = 1, 3
          STOT(I,J) = 0.0
   20 CONTINUE
      WRITE (4,40)
      WRITE (4,50)
      DO 30 L = 1, NLEG
        IF (RSPEC) THEN
          MODE = LMODE(L)
        ELSE
          MODE = MODSAV(L)
        ENDIF
        IF (MCOUNT.NE.1) WRITE (4,'(1X,A4,I3)') 'LINK',L
```

```
            DO 30 J = 1, NMAT(MODE)
              M = MATSAV(MODE,J)
              S1 = INCFRE(L,M,2)+INCFRE(L,M,9)+INCFRE(L,M,3)
              S2 = INCFRE(L,M,4)+INCFRE(L,M,10)
              IF (MCOUNT.NE.1) THEN
                WRITE (4,'(1X,A7,8(1PE9.2))') LABMAT(M)(4:10),INCFRE(L,M,1),
     *            S1,S2,INCFRE(L,M,5),INCFRE(L,M,6),INCFRE(L,M,7),
     *            INCFRE(L,M,8),INCFRE(L,M,NDOSP1)
              ELSE
                WRITE (4,'(1X,A4,I3,8(1PE9.2))') 'LINK',L,INCFRE(L,M,1),
     *            S1,S2,INCFRE(L,M,5),INCFRE(L,M,6),INCFRE(L,M,7),
     *            INCFRE(L,M,8),INCFRE(L,M,NDOSP1)
              ENDIF
              IF (RSPEC) THEN
                JZ = LZONE(L)
                STOT(JZ,1) = STOT(JZ,1)+INCFRE(L,M,1)
                STOT(JZ,2) = STOT(JZ,2)+S1
                STOT(JZ,3) = STOT(JZ,3)+S2
                STOT(JZ,4) = STOT(JZ,4)+INCFRE(L,M,5)
                STOT(JZ,5) = STOT(JZ,5)+INCFRE(L,M,6)
                STOT(JZ,6) = STOT(JZ,6)+INCFRE(L,M,7)
                STOT(JZ,7) = STOT(JZ,7)+INCFRE(L,M,8)
                STOT(JZ,8) = STOT(JZ,8)+INCFRE(L,M,NDOSP1)
              ENDIF
              SUM(1) = SUM(1)+INCFRE(L,M,1)
              SUM(2) = SUM(2)+S1
              SUM(3) = SUM(3)+S2
              SUM(4) = SUM(4)+INCFRE(L,M,5)
              SUM(5) = SUM(5)+INCFRE(L,M,6)
              SUM(6) = SUM(6)+INCFRE(L,M,7)
              SUM(7) = SUM(7)+INCFRE(L,M,8)
              SUM(8) = SUM(8)+INCFRE(L,M,NDOSP1)
           30 CONTINUE
              IF (RSPEC) THEN
                WRITE (4,'(/)')
                WRITE (4,'(A8,8(1PE9.2))') 'RURAL ',(STOT(1,J),J=1,8)
                WRITE (4,'(A8,8(1PE9.2))') 'SUBURB',(STOT(2,J),J=1,8)
                WRITE (4,'(A8,8(1PE9.2))') 'URBAN ',(STOT(3,J),J=1,8)
              ENDIF
              WRITE (4,'(/,A8,8(1PE9.2))') ' TOTALS:',(SUM(J),J=1,8)
        C
        C ... IF MAXIMUM INDIVIDUAL DOSE
              WRITE (4,60)
              DO 35 L = 1, NLEG
                WRITE (4,70) L, DOSMAX(L)
           35 CONTINUE
        C
        C ... THE FOLLOWING CODE IS USED FOR QA PURPOSES FOR THE 0170 INPUT DECK
        ONLY
              IF (LABMAT(1).NE.NAME) THEN
                IF (NAME.NE.'          ') WRITE (5,'(A10,8(1PE10.2))')
```

```
     *                                        NAME,(RSUM(J),J=1,8)
           RSUM(1) = SUM(1)
           RSUM(2) = SUM(2)
           RSUM(3) = SUM(3)
           RSUM(4) = SUM(4)
           RSUM(5) = SUM(5)
           RSUM(6) = SUM(6)
           RSUM(7) = SUM(7)
           RSUM(8) = SUM(8)
           NAME = LABMAT(1)
         ELSE
           RSUM(1) = RSUM(1)+SUM(1)
           RSUM(2) = RSUM(2)+SUM(2)
           RSUM(3) = RSUM(3)+SUM(3)
           RSUM(4) = RSUM(4)+SUM(4)
           RSUM(5) = RSUM(5)+SUM(5)
           RSUM(6) = RSUM(6)+SUM(6)
           RSUM(7) = RSUM(7)+SUM(7)
           RSUM(8) = RSUM(8)+SUM(8)
         ENDIF
         RETURN
      40 FORMAT(////,17X,'INCIDENT-FREE POPULATION EXPOSURE ',
        * 'IN PERSON-REM',/)
      50 FORMAT(/,'          PASSENGR   CREW   HANDLERS OFF LINK ON LINK',
        * '   STOPS   STORAGE   TOTALS')
      60 FORMAT (////,25X,'MAXIMUM INDIVIDUAL IN-TRANSIT DOSE',//)
      70 FORMAT (30X,'LINK',I3,1PE10.2,' REM')
         END
*DECK,PROUTB
         SUBROUTINE PROUTB
         COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
        *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
        *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
        *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
         REAL INCFRE
         COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
        *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
        *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
        *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
         LOGICAL DIST,PCKG
C
         COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
        *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
         REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
         LOGICAL RSPEC
         COMMON /TALLY/ TSUMRS(65,12),VSUMRS(65,12),VALINK(40,3)
         COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
        *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
        *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
        *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
        *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
```

```
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
      DIMENSION FSUM(12), STOT(3,6)
      CHARACTER STR0*38, STR1*24, STR2*24
C
      DO 10 J = 1, LISTLH
        FSUM(J) = 0.0
   10 CONTINUE
      DO 20 J = 1, 6
        DO 20 K = 1, 3
          STOT(K,J) = 0.0
   20 CONTINUE
      STR0 = 'EXPECTED VALUES OF POPULATION RISK IN '
      IF (URSKF) THEN
        STR1 = 'PERSON REM              '
      ELSE
        STR1 = 'LATENT CANCER FATALITIES'
        STR2 = 'GENETIC EFFECTS         '
      ENDIF
      CALL NEWPAG
      WRITE (4,'(14X,A62)') STR0//STR1
      WRITE (4,'(/,12X,A55)')
     *  'GROUND    INHALED   RESUSPD   CLOUDSH  *INGESTION TOTAL'
      IF (RSPEC) THEN
        DO 30 J = 1, NLNK
          WRITE (4,'(3X,''LINK'',I3,6(1PE10.2))') J,(VSUMRS(J,K),K=2,6),
     *      VSUMRS(J,1)+VSUMRS(J,6)
          JZ = LZONE(J)
          STOT(JZ,1) = STOT(JZ,1)+VSUMRS(J,2)
```

```
            STOT(JZ,2) = STOT(JZ,2)+VSUMRS(J,3)
            STOT(JZ,3) = STOT(JZ,3)+VSUMRS(J,4)
            STOT(JZ,4) = STOT(JZ,4)+VSUMRS(J,5)
            STOT(JZ,5) = STOT(JZ,5)+VSUMRS(J,6)
            STOT(JZ,6) = STOT(JZ,6)+VSUMRS(J,1)+VSUMRS(J,6)
            DO 30 K = 1, LISTLH
              FSUM(K) = FSUM(K)+VSUMRS(J,K)
   30       CONTINUE
         ELSE
           DO 50 M = 1, MCOUNT
             WRITE (4,'(A10)') LABMAT(M)
             DO 50 J = 1, NIS(M)
               I = ISOSAV(M,J)
               L = LIBSAV(I)
               WRITE (4,'(A10,6(1PE10.2))') LABISO(L),
          *       (VSUMRS(I,K),K=2,6),VSUMRS(I,1)+VSUMRS(I,6)
               DO 40 K = 1, LISTLH
                 FSUM(K) = FSUM(K)+VSUMRS(I,K)
   40          CONTINUE
   50      CONTINUE
         ENDIF
         IF (RSPEC) THEN
           WRITE (4,'(/)')
           WRITE (4,'(2X,A8,6(1PE10.2))') 'RURAL ',(STOT(1,K),K=1,6)
           WRITE (4,'(2X,A8,6(1PE10.2))') 'SUBURB',(STOT(2,K),K=1,6)
           WRITE (4,'(2X,A8,6(1PE10.2))') 'URBAN ',(STOT(3,K),K=1,6)
         ENDIF
         WRITE (4,'(/,3X,''TOTALS:'',6(1PE10.2))') (FSUM(K),K=2,6),
        *                 FSUM(1)+FSUM(6)
C
         WRITE (4,'(//)')
         IF (URSKF) GOTO 100
         IF ((.NOT.RSPEC.AND.MCOUNT+ICOUNT.GT.14)
        *     .OR.(RSPEC.AND.NLNK.GT.12))  CALL NEWPAG
         WRITE (4,'(/,14X,A62)') STR0//STR2
         WRITE (4,'(/,12X,A55)')
        *  'GROUND    INHALED   RESUSPD   CLOUDSH  *INGESTION TOTAL'
         IF (RSPEC) THEN
           DO 70 J = 1, NLNK
             WRITE (4,'(3X,''LINK'',I3,6(1PE10.2))') J,
          *          (VSUMRS(J,K),K=8,12),VSUMRS(J,7)+VSUMRS(J,12)
   70      CONTINUE
         ELSE
           DO 90 M = 1, MCOUNT
             WRITE (4,'(A10)') LABMAT(M)
             DO 90 J = 1, NIS(M)
               I = ISOSAV(M,J)
               L = LIBSAV(I)
               WRITE (4,'(A10,6(1PE10.2))') LABISO(L),
          *            (VSUMRS(I,K),K=8,12),VSUMRS(I,7)+VSUMRS(I,12)
   90      CONTINUE
```

```
      ENDIF
      WRITE (4,'(/,3X,''TOTALS:'',6(1PE10.2))')
     *              (FSUM(K),K=8,12),FSUM(7)+FSUM(12)
C
C.....DATA WRITTEN TO FILE7 IS ONLY USED TO TEST 0170 OUTPUT (4TE1)
      WRITE (7,'(3X,A10,11(1PE10.2))') LABMAT(1),(FSUM(K),K=2,5),
     *      FSUM(1),(FSUM(K),K=8,11),FSUM(7),VALINK(1,3)+VALINK(2,3)
      WRITE (4,'(/)')
  100 CONTINUE
      WRITE (4, 110)
  110 FORMAT (' * NOTE THAT INGESTION RISK IS A SOCIETAL RISK;',/,
     *        '   THE USER MAY WISH TO TREAT THIS VALUE SEPARATELY.')
C
C.... EXPECTED RISK -- OTHER
      DO 120 K = 1, 3
        FSUM(K) = 0.0
  120 CONTINUE
      CALL NEWPAG
      WRITE (4,'(22X,A28,/)') 'EXPECTED RISK VALUES - OTHER'
      IF (URSKF) THEN
        WRITE (4,'(23X,A25)') 'LINK    ECON       EARLY'
        WRITE (4,'(23X,A27)') '          $$       FATALITY'
        DO 130 K = 1, NLEG
          WRITE (4,'(25X,I2,1X,2(1PE11.2))') K,VALINK(K,3),VALINK(K,1)
          FSUM(1) = FSUM(1)+VALINK(K,1)
          FSUM(3) = FSUM(3)+VALINK(K,3)
  130   CONTINUE
        WRITE (4,'(/23X,A5,2(1PE11.2))') 'TOTAL',FSUM(3),FSUM(1)
      ELSE
        WRITE (4,'(13X,A41)') 'LINK    ECON       EARLY      EARLY'
        WRITE (4,'(13X,A44)') '          $$       FATALITY   MORBIDITY'
        DO 140 K = 1, NLEG
          WRITE (4,'(20X,I2,1X,3(1PE11.2))')
     *                K,VALINK(K,3),VALINK(K,1),VALINK(K,2)
          FSUM(1) = FSUM(1)+VALINK(K,1)
          FSUM(2) = FSUM(2)+VALINK(K,2)
          FSUM(3) = FSUM(3)+VALINK(K,3)
  140   CONTINUE
        WRITE (4,'(/18X,A5,3(1PE11.2))') 'TOTAL',FSUM(3),FSUM(1),FSUM(2)
      ENDIF
      WRITE (4,'(/)')
      RETURN
      END
*DECK,PROUTC
      SUBROUTINE PROUTC (LAB1,LAB2,DOUT)
C
      COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
      REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
      LOGICAL RSPEC
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
```

```
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      DIMENSION DOUT(20,40)
      CHARACTER*80 LABEL1,LABEL2,LAB1,LAB2,STR
      WRITE (4, '(//)')
      N1 = INDEX(LAB1,'$')
      NS = (81-N1)/2
      DO 10 J = 1, NS
   10   STR(J:J) = ' '
      LABEL1(1:NS+N1-1) = STR(1:NS)//LAB1(1:N1-1)
      WRITE (4,'(A)') LABEL1(1:NS+N1-1)
      N1 = INDEX(LAB2,'$')
      NS = (81-N1)/2
      DO 20 J = 1, NS
   20   STR(J:J) = ' '
      LABEL2(1:NS+N1-1) = STR(1:NS)//LAB2(1:N1-1)
      WRITE (4,'(A)') LABEL2(1:NS+N1-1)
      IF (RSPEC) THEN
        KOUNT = 0
        LM = INT(NLNK/7)
        IF (MOD(NLNK,7).NE.0)  LM = LM + 1
        DO 30 L = 1, LM
          KOUNT = KOUNT + NSEV+2
          IF (KOUNT .GT. 44) THEN
C . . . . . THERE WILL NOT BE ENOUGH ROOM - START A NEW PAGE
            CALL NEWPAG
```

```
            KOUNT = NSEV + 2
          ENDIF
          NZN = 7
          IF (L.EQ.LM) NZN = MOD(NLNK,7)
          IF (NZN.EQ.0) NZN = 7
          WRITE (4,'(/,1X,A8,7(3X,A4,1X,I2))') 'CATEGORY',('LINK',7*(L-1
     *      )+K,K=1,NZN)
          DO 30 J = 1, NSEV
            WRITE (4,'(I5,5X,7(1PE10.2))') J,(DOUT(J,7*(L-1)+K),K=1,NZN)
   30   CONTINUE
        ELSE
          WRITE (4,'(1X,A10,1X,3(A10,10X))') 'CATEGORY  ','RURAL ',
     *    'SUBURB','URBAN '
          DO 40 J = 1, NSEV
            WRITE (4,'(I5,9X,7(1PE10.2,10X))') J,(DOUT(J,K),K=1,NPOP)
   40   CONTINUE
        ENDIF
        RETURN
        END
*DECK,PROUTD
        SUBROUTINE PROUTD (LAB1,LAB2,LAB3,NNN,DOUT)
C
        COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
        REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
        LOGICAL RSPEC
        COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
        LOGICAL DIST,PCKG
        COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
        REAL LCFPYR

        DIMENSION DOUT(30,20)
        CHARACTER*80 LABEL1,LABEL2,LAB1,LAB2,LAB3*8, STR
        WRITE (4,'(/)')
        N1 = INDEX(LAB1,'$')
        NS = (81-N1)/2
        DO 10 J = 1, NS
   10   STR(J:J) = ' '
        LABEL1(1:NS+N1-1) = STR(1:NS)//LAB1(1:N1-1)
        WRITE (4,'(A)') LABEL1(1:NS+N1-1)
```

```
      N1 = INDEX(LAB2,'$')
      NS = (81-N1)/2
      DO 20 J = 1, NS
   20   STR(J:J) = ' '
      LABEL2(1:NS+N1-1) = STR(1:NS)//LAB2(1:N1-1)
      WRITE (4,'(A)') LABEL2(1:NS+N1-1)
      LM = INT(NSEV/7)
      IF (MOD(NSEV,7).NE.0)  LM = LM + 1
      DO 30 L = 1, LM
        IF ((L .EQ. 3) .AND. (NNN .GT. 13))  CALL NEWPAG
        NZN = 7
        IF (L.EQ.LM) NZN = MOD(NSEV,7)
        IF (NZN.EQ.0) NZN = 7
        WRITE (4,'(/,1X,A9,7(2X,A6,I2))') LAB3,('SEVER:',7*(L-1)+K,K=1,
     *    NZN)
        DO 30 J = 1, NNN
          WRITE (4,'(I5,5X,7(1PE10.2))') J,(DOUT(J,7*(L-1)+K),K=1,NZN)
   30 CONTINUE
      RETURN
      END
*DECK,PRRELD
      SUBROUTINE PRRELD
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /MAXSZ/ MAXISO,MAXSEV,MAXGRP,MAXRAD,MAXDSP,MAXARS,
     *  MAXMOD,MAXNDL,MAXMAT,NPRAM,RSPHLF
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
```

Page(s) Missing

from

Original Document

```
            IF (L.EQ.LM) NZN = MOD(NLNK,6)
            IF (NZN.EQ.0) NZN = 6
            WRITE (4,'(/,19X,6(3X,A4,1X,I2),/)') ('LINK',6*(L-1)+K,K=1,NZN)
            WRITE (4,'(1X,A19,6(1PE10.2))') 'PEOPLE IN CREW     ',
           *   (DNORML(LMODE(6*(L-1)+K),7),K=1,NZN)
            WRITE (4,'(1X,A19,6(1PE10.2))') 'CREW EXPOSURE DIST ',
           *   (DNORML(LMODE(6*(L-1)+K),8),K=1,NZN)
            WRITE (4,'(1X,A19,6(1PE10.2))') 'PEOPLE AT STOPS    ',
           *   (DNORML(LMODE(6*(L-1)+K),14),K=1,NZN)
            WRITE (4,'(1X,A19,6(1PE10.2))') 'STOP EXPOSURE DIST ',
           *   (DNORML(LMODE(6*(L-1)+K),15),K=1,NZN)
            WRITE (4,'(1X,A19,6(1PE10.2))') 'STOP TIME PER KM   ',
           *   (DNORML(LMODE(6*(L-1)+K),10),K=1,NZN)
            WRITE (4,'(1X,A19,6(1PE10.2))') 'MINIMUM STOP TIME  ',
           *   (DNORML(LMODE(6*(L-1)+K),11),K=1,NZN)
            WRITE (4,'(1X,A19,6(1PE10.2))') 'END POINT STOP TIME',
           *   (DNORML(LMODE(6*(L-1)+K),12),K=1,NZN)
            WRITE (4,'(1X,A19,6(1PE10.2))') 'PEOPLE AT STORAGE  ',
           *   (DNORML(LMODE(6*(L-1)+K),17),K=1,NZN)
            WRITE (4,'(1X,A19,6(1PE10.2))') 'STOR. EXPOSURE DIST',
           *   (DNORML(LMODE(6*(L-1)+K),18),K=1,NZN)
            WRITE (4,'(1X,A19,6(1PE10.2))') 'NUMBER OF HANDLINGS',
           *   (DNORML(LMODE(6*(L-1)+K),9),K=1,NZN)
         20 CONTINUE
            RETURN
            END
      *DECK,PRSEND
            SUBROUTINE PRSEND
            COMMON /LABELS/ LDESCR(38),LABISO(65),
           *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
           *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
            CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
           *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
           *  LBTRSH*10,LABGRP*10,LABMAT*10
            COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
           *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
           *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
           *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
            REAL INCFRE
            COMMON /MAXSZ/ MAXISO,MAXSEV,MAXGRP,MAXRAD,MAXDSP,MAXARS,
           *  MAXMOD,MAXNDL,MAXMAT,NPRAM,RSPHLF
      C
      C     PRINT NORMAL SENSITIVITY SUMMARY
            DO 30 L = 1, NLEG
              CALL NEWPAG
              WRITE (4,60) L
              DO 10 I = 1, NPRAM
                KI(I) = I
         10   CONTINUE
              CALL SSORT (SENPRM(1,L),KI,NPRAM,-2)
              DO 20 I = 1, NPRAM
```

```
          WRITE (4,70) I,LDESCR(KI(I)),SENPRM(I,L)*.01
   20    CONTINUE
         WRITE (4,80)
   30 CONTINUE
      RETURN
   60 FORMAT (12X,'INCIDENT-FREE IMPORTANCE ANALYSIS SUMMARY FOR LINK',
     * I3,/12X,53('*')/////12X,'INDEX',4X,'DESCRIPTION OF PARAMETER',
     * 17X,'IMPORTANCE',/)
   70 FORMAT (1H ,12X,I5,2X,A40,2X,1PE10.3)
   80 FORMAT (///,'  THE IMPORTANCE VALUE ESTIMATES THE PERSON-REM ',
     *  'INFLUENCE',/,'  OF A ONE PERCENT INCREASE IN THE PARAMETER')
      END
*DECK,QUAD
      SUBROUTINE QUAD (IRTYPE,X,ANS)
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      U = COEF(IRTYPE,1)
      A1 = COEF(IRTYPE,2)
      A2 = COEF(IRTYPE,3)
      A3 = COEF(IRTYPE,4)
      A4 = COEF(IRTYPE,5)
      IF (U.EQ.0.0) THEN
        ANS = 1./X
      ELSE
        P = 1./X+(A2/U+A3/U**2+2.*A4/U**3)+X*(A3/U+2.*A4/U**2)+X*X*A4/U
        UX = U*X
        ANS = P*EXP(-UX)+(A1-U)*E1(UX)
      ENDIF
      RETURN
      END
*DECK,RADTN4
      PROGRAM RADTN4
C
C     A REVISED VERSION OF RADTRAN -- A COMPUTER CODE TO ANALYZE
C     TRANSPORTATION OF RADIOACTIVE MATERIAL. THE CALCULATION IS
C     DESCRIBED IN SAND 84-0036. USER INSTRUCTIONS RESEMBLE RADTRAN II,
C     SAND 82-2681.  FILE2 IS THE INPUT FILE, FILE4 IS THE OUTPUT FILE,
C     FILE6 IS USED FOR PLOTTING. FILE5 IS USED FOR SUMMING INCIDENT-
C     FREE DATA OVER MULTIPLE MODES/LINKS AND RUNS.  FILE7 IS USED FOR
C     SUMMING ACCIDENT DATA OVER MULTIPLE MODES/LINKS.
C
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /CON/ TOL,NDERV,NA,NC,TWOPI,PI,Q6,WS,GAM,FMU,DZ,VV
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
```

```
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      COMMON /TALLY/ TSUMRS(65,12),VSUMRS(65,12),VALINK(40,3)
      COMMON /CARD/ IPOINT,KBYTE,CONTUF,NRFLAG,NCS
      LOGICAL CONTUF,NRFLAG
      COMMON /CCARD/ ICARD,TEST,ICARDI,NVY(10)
      CHARACTER*10 ICARD*80, TEST,ICARDI,NVY*1
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
C
      COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
      REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
      LOGICAL RSPEC
      COMMON /MAXSZ/ MAXISO,MAXSEV,MAXGRP,MAXRAD,MAXDSP,MAXARS,
     *  MAXMOD,MAXNDL,MAXMAT,NPRAM,RSPHLF
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
      COMMON /HEADER/ CDATE, KLOCK, TITLE
```

```
      CHARACTER CDATE*9, KLOCK*8, TITLE*80

      REAL AOUT(20,40,10),BOUT(20,40,10),COUT(20,40,10),DOUT(20,40,10),
     *     EOUT(20,40,10)
      CHARACTER AL*1
      DIMENSION AL(6), CTMLVL(30)
      LOGICAL DONE, PRDATA, USED(10)

      DATA AL / 'A','B','C','D','E','F'/
C     MODES OF TRANSPORT
C     1 - TRUCK
C     2 - RAIL
C     3 - BARGE
C     4 - SHIP
C     5 - CARGO AIR
C     6 - PASSENGER AIR
C     7 - PASSENGER VAN
C     8 - C. VAN-T (USED WITH TRUCK)
C     9 - C. VAN-R (USED WITH RAIL)
C     10- C. VAN-CA (USED WITH CARGO AIR)
C
      OPEN (UNIT=2,STATUS='OLD',FILE='FILE2.DAT')
      OPEN (UNIT=4,STATUS='NEW',FILE='FILE4.DAT')
      OPEN (UNIT=5,STATUS='NEW',FILE='FILE5.DAT')
      OPEN (UNIT=6,STATUS='NEW',FILE='FILE6.DAT')
      OPEN (UNIT=7,STATUS='NEW',FILE='FILE7.DAT')
      OPEN (UNIT=20,STATUS='OLD',FILE='$DISK2:[TTC.RT4_RUN]RT4INT.DAT')
      OPEN (UNIT=21,STATUS='OLD',FILE='$DISK2:[TTC.RT4_RUN]RT4ISO.DAT')
      CALL RDDAT
      CLOSE (UNIT=20,STATUS='KEEP')
      CLOSE (UNIT=21,STATUS='KEEP')
      CALL DATE (CDATE)
      CALL TIME (KLOCK)
      CALL NEWPAG
      WRITE (4,'(10(/))')
      CALL BANNER ('RADTRAN 4$')
      CALL NEWPAG
      WRITE (4,'(/,35X,A10,///)') 'ECHO CHECK'
      CALL RDINFR (DONE)
C ... Set the shielding for shielding option IUOPT
      CALL SHIELD (IUOPT, RR, RS, RU)
      IF (URSKF) THEN
        EXPLCF(1) = 1.0
        EXPLCF(2) = 0.0
        EXPLCF(3) = 0.0
        EXPLCF(4) = 0.0
        EXPLCF(5) = 0.0
        WBLCF = 1.0
        GNDSGE = 0.0
      ENDIF
C
```

```
C ... THIS LOOP CONTINUES UNTIL DONE IS SET TRUE IN SUBROUTINE RDINFR
   45 CONTINUE
      CALL RDINFR (DONE)
C ... WHILE NOT DONE
      IF (.NOT. DONE) THEN
C
C . . . SET THE ASIZE FOR EACH ISOPLETH SIZE IN METERS SQ
        ASIZE(1) = AREADA(1)
        DO 90 I = 2, NA
          ASIZE(I) = AREADA(I) - AREADA(I-1)
   90   CONTINUE
C
C . . . SET DISTKM & PKGSIZ IF ASSIGN WAS NOT USED, AND SET EFFSIZ & PKGCOE
        CALL SETVAL
C
C ..... PRINT INPUT DATA TABLES IF FULL PRINTOUT IS WANTED
        IF (ISEN.GE.2) THEN
          CALL PRCHRD
          IF (RSPEC) THEN
            CALL PRSEGD
          ELSE
            CALL PRNORD
          ENDIF
          CALL PRISTP
          CALL PRRELD
        ENDIF
C
        DO 100 J = 1, MAXMOD
          USED(J) = .FALSE.
  100   CONTINUE

        CALL NEWPAG
        IF (RSPEC) THEN
          NLEG = NLNK
        ELSE
          NLEG = NMODE
        ENDIF
C
        CALL INITLZ(1)
        IF (IANA .EQ. 2) GOTO 195
C ....  INCIDENT-FREE NORMAL DOSE CALCULATIONS...
        DO 190 LEG = 1, NLEG
          CALL SETMOD (LEG)
          DO 190 M = 1, NMAT(MODE)
            MAT = MATSAV(MODE,M)
            CALL SETMAT
            CALL NORMAL (LEG,M)
            IF (ISEN.GE.3)  CALL SENSTV (LEG)
C ..... ACCUMLATE NORMAL DOSE DATA
            DO 170 L = 1, NDOSE
              DOSEN(NDOSP1) = DOSEN(NDOSP1)+DOSEN(L)
```

```
  170       CONTINUE
C ..... ACCUMULATE NORMAL DOSE DATA FOR TABLES 1 AND 2
            DO 180 L = 1, NDOSP1
              INCFRE(LEG,MAT,L) = DOSEN(L)
  180       CONTINUE
  190   CONTINUE
        IF (IANA.EQ.1) GO TO 500
C
C ..... ACCIDENT CALCULATIONS...
  195   CONTINUE
        DO 490 LEG = 1, NLEG
          CALL INITLZ(2)
          CALL SETMOD (LEG)
C . . . . SET ACCRAT AND ACCPYR
          IF (RSPEC) THEN
            J = LZONE(LEG)
            DO 200 I = 1, NSEV
               ACCRAT(I,J) = SEVFRC(MODE,I,J)*LARAT(LEG)
               ACCPYR(I,J) = SPY*DSTRVL*ACCRAT(I,J)
  200       CONTINUE
          ELSE
            DO 210 I = 1, NSEV
              DO 210 J = 1, NPOP
                ACCRAT(I,J) = SEVFRC(MODE,I,J)*ARATMZ(MODE,J)
                ACCPYR(I,J) = SPY*DSTRVL*ACCRAT(I,J)*FRCZON(J)
  210       CONTINUE
          ENDIF

          NRP = 1
          IF (IPSQSB.EQ.1) NRP = NC
          DO 450 JR = 1, NRP
C . . . . . PRINT THE CONTAMINATION DATA AND EARLY EFFECTS DATA IF MODE HAS
NOT
C . . . . . BEEN USED BEFORE (PREVENTS MULTIPLE PRINTS FOR ROUTE-SPECIFIC)
            PRDATA = .NOT.USED(MODE)

            IF (IPSQSB.EQ.1) THEN
C . . . . . . FOR LAST CLOUD CAN SET MODE USED TO TRUE
              IF (JR.EQ.NRP) USED(MODE) = .TRUE.
C . . . . . . IF THE PROBABILITY IS ZERO DON'T BOTHER TO COMPUTE
              IF (PSPROB(JR).EQ.0.0) GO TO 450
C . . . . . . SET CLOUD ARRAYS TO PASQUILL CLOUD JR
              CALL PASQ (JR)
            ELSE
C . . . . . . IF NOT USING PASQUILL ONLY 1 CLOUD WITH PROBABILITY OF 1.0
              PSPROB(1) = 1.0
C . . . . . . THIS MODE HAS NOW BEEN USED
              USED(MODE) = .TRUE.
            ENDIF

            CALL INITLZ(3)
```

```
C            DETERMINE TOTAL DECONTAMINATION FACTOR ARRAY
             DO 310 M = 1, NMAT(MODE)
               MAT = MATSAV(MODE,M)
               CALL SETMAT
               DO 310 NI = 1, NIS(MAT)
                 ISO = ISOSAV(MAT,NI)
                 LIB = LIBSAV(ISO)
                 CALL SETISO
                 CALL MDT(DFLEV,AREADA,VELDEP(LIB),NAREAS,CHIVAL(1,LIB),
     *                    CTMLVL,NAVINT(LIB))
                 DO 300 J = 1, NSEV
                   DO 300 K = 1, NAREAS
C  ..............EQUATION 100 IN RADTRAN III GUIDE
                     CLVL = CI*PPS*RFRAC(IGRP,J)*AERSOL(IDSP,J)*CTMLVL(K)
                     TCLEVL(K,J) = TCLEVL(K,J) + CLVL
C  .............. EQUATION 99 IN RADTRAN III GUIDE
                     DECON(K,J,ISO) = CLVL/CULVL
C  .............. TCDONF=SUM OVER ISOTOPES OF DECON (EQS 99&112 IN RADTRAN
III)
                     TDCONF(K,J) = TDCONF(K,J) + DECON(K,J,ISO)
  300              CONTINUE
  310          CONTINUE
               IACCI = 1
               DO 370 M = 1, NMAT(MODE)
                 MAT = MATSAV(MODE,M)
                 CALL SETMAT
                 DO 370 NI = 1, NIS(MAT)
                   ISO = ISOSAV(MAT,NI)
                   LIB = LIBSAV(ISO)
                   CALL SETISO
                   CALL ACC
C
C                  SUMMARY --
                   DO 360 J = 1, NSEV
C . . . . . . . . . FOOD INGESTION DOSE FROM RURAL ZONE ONLY
                     ACCPSP = ACCPYR(J,1)*PSPROB(JR)
                     TSUMRS(ISO,6) = TSUMRS(ISO,6)+INGLCF(J)*ACCPSP
                     TSUMRS(ISO,12) = TSUMRS(ISO,12)+INGGE(J)*ACCPSP
                     DO 360 I = 1, NPOP
                       ACCPSP = ACCPYR(J,I)*PSPROB(JR)
                       TSUMRS(ISO,1) = TSUMRS(ISO,1)+LCFPYR(J,I)*PSPROB(JR)
                       TSUMRS(ISO,2) = TSUMRS(ISO,2)+GRDLCF(J,I)*ACCPSP
                       TSUMRS(ISO,3) = TSUMRS(ISO,3)+DNHLCF(J,I)*ACCPSP
                       TSUMRS(ISO,4) = TSUMRS(ISO,4)+RDFLCF(J,I)*ACCPSP
                       TSUMRS(ISO,5) = TSUMRS(ISO,5)+CLDLCF(J,I)*ACCPSP
                       TSUMRS(ISO,7) = TSUMRS(ISO,7)+GEPYR(J,I)*PSPROB(JR)
                       TSUMRS(ISO,8) = TSUMRS(ISO,8)+GRDGE(J,I)*ACCPSP
                       TSUMRS(ISO,9) = TSUMRS(ISO,9)+DNHGE(J,I)*ACCPSP
                       TSUMRS(ISO,10) = TSUMRS(ISO,10)+RDFGE(J,I)*ACCPSP
                       TSUMRS(ISO,11) = TSUMRS(ISO,11)+CLDGE(J,I)*ACCPSP
  360              CONTINUE
```

```
                IF (IRNKC.EQ.1) CALL WTRNK (LEG,JR,.FALSE.,.FALSE.)
C . . . . . IS ANYTHING DISPERSED
                IF (IDSP.GT.1)  IACCI = 2
  370         CONTINUE
              IACC = IACCI
              CALL COST
              IF (IRNKC.EQ.1) CALL WTRNK (LEG,JR,.TRUE.,.FALSE.)
              CALL EARLY
              DO 380 I = 1, NSEV
                DO 380 J = 1, NPOP
                  PECMPT(I,J) = PECMPT(I,J)+ECNMPT(I,J)*PSPROB(JR)
                  PSUMEF(I,J) = PSUMEF(I,J)+SUMEF(I,J)*PSPROB(JR)
                  PSUMEM(I,J) = PSUMEM(I,J)+SUMEM(I,J)*PSPROB(JR)
                  PDTOT(I,J) = PDTOT(I,J)+DTOT(I,J)*PSPROB(JR)
  380         CONTINUE
              DO 400 I = 1, NPOP
                DO 400 J = 1, NSEV
                  VALINK(LEG,1) = VALINK(LEG,1)+EFPYR(J,I)*PSPROB(JR)
                  VALINK(LEG,2) = VALINK(LEG,2)+EMPYR(J,I)*PSPROB(JR)
                  VALINK(LEG,3) = VALINK(LEG,3)+ECNPYR(J,I)*PSPROB(JR)
  400         CONTINUE
              IF (ISEN.GE.2.AND.PRDATA) THEN
                CALL NEWPAG
                IF (IPSQSB.EQ.1)  WRITE (4,'(30X,
     *                                ''PASQUILL CATEGORY '',A1)') AL(JR)
                WRITE (4,'(32X,''MODE '',A10)') LABMOD(MODE)
                CALL PROUTD ('1-YEAR LUNG DOSE - INHALATION PATHWAY$',
     *               'BDF = 1   (REM)$','AREA #  ',NAREAS,EARLYD(1,1,1))
                IF ((NAREAS.GT.20).OR.(NSEV.GT.7))  CALL NEWPAG
                CALL PROUTD ('1-YEAR MARROW DOSE - INHALATION PATHWAY$',
     *               'BDF = 1   (REM)$','AREA #  ',NAREAS,EARLYD(1,1,2))
                IF (.NOT.URSKF) THEN
                  CALL NEWPAG
                  CALL PROUTD ('DOSE TO GONADS$','BDF = 1   (REM)$',
     *                        'AREA #  ',NAREAS,EARLYD(1,1,3))
                  IF ((NAREAS.GT.20).OR.(NSEV.GT.7))  CALL NEWPAG
                  CALL PROUTD ('DOSE TO LARGE LOWER INTESTINE$',
     *               'BDF = 1   (REM)$','AREA #  ',NAREAS,EARLYD(1,1,4))
                  CALL NEWPAG
                  CALL PROUTD ('DOSE TO THYROID$','BDF = 1   (REM)$',
     *                        'AREA #  ',NAREAS,EARLYD(1,1,5))
                ENDIF
                CALL NEWPAG
                IF (IPSQSB.EQ.1)  WRITE (4,'(30X,
     *                                ''PASQUILL CATEGORY '',A1)') AL(JR)
                WRITE (4,'(32X,''MODE '',A10)') LABMOD(MODE)
                CALL PROUTD ('GROUND SURFACE CONTAMINATION TABLE (MICRO'//
     *            ' CI/M**2)$','BEFORE CLEANUP$','AREA #  ',NAREAS,TCLEVL)
              ENDIF
C . . . . END OF PASQUILL CATEGORY LOOP 1 - 6
  450       CONTINUE
```

```
C
            IF (RSPEC) THEN
C . . . .     SUM OVER ISOTOPES FOR EACH LINK
              DO 452 M = 1, NMAT(MODE)
                MAT = MATSAV(MODE,M)
                DO 452 NI = 1, NIS(MAT)
                  ISO = ISOSAV(MAT,NI)
                  DO 452 J = 1, LISTLH
                    VSUMRS(LEG,J) = VSUMRS(LEG,J)+TSUMRS(ISO,J)
  452           CONTINUE
            ELSE
C . . . .     SUM OVER MODES FOR EACH ISOTOPE
              DO 455 M = 1, NMAT(MODE)
                MAT = MATSAV(MODE,M)
                DO 455 NI = 1, NIS(MAT)
                  ISO = ISOSAV(MAT,NI)
                  DO 455 J = 1, LISTLH
                    VSUMRS(ISO,J) = VSUMRS(ISO,J)+TSUMRS(ISO,J)
  455           CONTINUE
            ENDIF
            IF (RSPEC) THEN
              DO 470 J = 1, NSEV
                AOUT(J,LEG,1) = ACCPYR(J,LZONE(LEG))
                BOUT(J,LEG,1) = PSUMEF(J,LZONE(LEG))
                EOUT(J,LEG,1) = PECMPT(J,LZONE(LEG))
                COUT(J,LEG,1) = PDTOT(J,LZONE(LEG))
                DOUT(J,LEG,1) = PSUMEM(J,LZONE(LEG))
  470           CONTINUE
            ELSE
              DO 480 J = 1, NSEV
                DO 480 K = 1, NPOP
                  AOUT(J,K,MODE) = ACCPYR(J,K)
                  BOUT(J,K,MODE) = PSUMEF(J,K)
                  EOUT(J,K,MODE) = PECMPT(J,K)
                  COUT(J,K,MODE) = PDTOT(J,K)
                  DOUT(J,K,MODE) = PSUMEM(J,K)
  480           CONTINUE
            ENDIF
C ......END DO LEG = 1, NLEG
  490     CONTINUE
  500     CONTINUE
          IF (IANA .NE. 2) THEN
C . . . .   PRINT INCIDENT-FREE OUTPUT
            CALL PROUTA
C . . . .   PRINT SENSITIVITY ANALYSIS
            IF (ISEN.GE.3)  CALL PRSEND
          ENDIF
          IF (IANA .NE. 1) THEN
C . . . .   PRINT CONSEQUENCE DATA TABLES
            IF (ISEN.GE.3) CALL PRCONS (AOUT,BOUT,COUT,DOUT,EOUT)
C . . . .   PRINT ACCIDENT OUTPUT
```

```
          CALL PROUTB
        ENDIF

C . . . WRITE EOF TO FILE6
        IF (IRNKC.EQ.1) CALL WTRNK (1,1,.FALSE.,.TRUE.)

C ..... GO READ RDINFR AGAIN
        GO TO 45
C ... END WHILE NOT DONE
      ENDIF

C ... WRITE ORDERED PAIRS TO FILE10, FILE11, AND FILE12
      IF (IRNKC.EQ.1) CALL PAIRS

      WRITE (4,'(A)') ' END OF RUN'
      STOP
      END
*DECK,RDDAT
      SUBROUTINE RDDAT
      COMMON /PROB/ PROBEF(30,4),REMLEV(30),NE,NREMLV
      COMMON /TBD/ TBDIHD(2),TBDTHD(2),TBPPH(2)
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /CARD/ IPOINT,KBYTE,CONTUF,NRFLAG,NCS
      LOGICAL CONTUF,NRFLAG
      COMMON /IKW/ IKW1(18),IKW2(6),IKW3(4),IKW4(32),IKW5(3),IKW6(3),
     *  IKW7(7),IKW8(7)
      COMMON /NIKW/ NKW1,NKW2,NKW3,NKW4,NKW5,NKW6,NKW7,NKW8
      CHARACTER*10 IKW1,IKW2,IKW3,IKW4,IKW5,IKW6,IKW7,IKW8
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
```

```
*  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
 REAL LCFPYR
 COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
*  A1T20(20),A20T40(20),AGT40(20),
*  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
*  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
*  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
*  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
*  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
*  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
*  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
*  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
*  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
 REAL INGVAL,INGLCF,INGGE
 LOGICAL URSKF
 COMMON /CON/ TOL,NDERV,NA,NC,TWOPI,PI,Q6,WS,GAM,FMU,DZ,VV
 COMMON /CCARD/ ICARD,TEST,ICARDI,NVY(10)
 CHARACTER*10 ICARD*80, TEST,ICARDI,NVY*1
 COMMON /CMDT/ EPS,DLIM,MTAL
 COMMON /MAXSZ/ MAXISO,MAXSEV,MAXGRP,MAXRAD,MAXDSP,MAXARS,
*  MAXMOD,MAXNDL,MAXMAT,NPRAM,RSPHLF
 COMMON /P/   H3WTR(11), H3GAS(11), C14ORG(11),C14GAS(11),
*  P32(11),   S35(11),   CA45(11),  CR51(11),  MN54(11),
*  FE55(11),  FE59(11),  CO58(11),  CO60(11),  ZN65(11),
*  GA67(11),  KR85(11),  SR89(11),  SR90(11),  Y91(11),
*  ZR95(11),  NB94(11),  NB95(11),  MO99(11),  TC99(11),
*  RU103(11), RU106(11), SB125(11), TE125M(11),TE127M(11),
*  TE127(11), TE129M(11),TE129(11), I125(11),  I129(11),
*  I131(11),  XE133(11), CS134(11), CS137(11), CE141(11),
*  CE144(11), PM147(11), SM151(11), EU152(11), EU154(11),
*  EU155(11), U233(11),  U235(11),  U238(11),  NP237(11),
*  PU236(11), PU238(11), PU239(11), PU240(11), PU241(11),
*  PU242(11), AM241(11), AM243(11), CM242(11), CM244(11),
*  CF252(11), DUMMY(715),NISOTP,NPRP
 REAL KR85,NB94,NB95,I129,I131,NP237,MN54,MO99,I125
 COMMON /O/ ISONAM(125)
 CHARACTER*10 ISONAM

 READ(20,*) NSEV,NGROUP,NRAD,NAREAS
 READ(20,*) IRNKC,IANA,IUOPT,ISEN,IPSQSB
 READ(20,*) URSKF
 READ(20,*) (COEF(1,I),I=1,5)
 READ(20,*) (COEF(2,I),I=1,5)
 READ(20,*) ECNPYR,ECNMPT
 READ(20,*) IPRAM,NPRAM
 READ(20,*) MAXSEV,MAXGRP,MAXISO,MAXMAT
 READ(20,*) NPOP,NPOP1
 READ(20,*) NDLAB
 READ(20,*) NDOSE,NDOSP1
 READ(20,*) MAXRAD,MAXDSP
 READ(20,*) NORG
```

```
      READ(20,*) RSPHLF
      READ(20,*) MAXARS,MAXMOD,MAXNDL
      READ(20,*) POPDEN
      READ(20,*) TABSPY,FKMPS
      READ(20,*) RPCVAL,RFRAC
      READ(20,*) INGVAL
      READ(20,*) CH1,CH2,ITRAIN
      DO 10 I = 1, 10
        READ(20,*) (DNORML(I,J),J=1,25)
10    CONTINUE
      READ(20,*) (ARATMZ(I,1),I=1,10)
      READ(20,*) (ARATMZ(I,2),I=1,10)
      READ(20,*) (ARATMZ(I,3),I=1,10)
      READ(20,*)
      DO 20 I = 1, 3
        READ(20,*)
        DO 30 J = 1, 10
          READ(20,*) (SEVFRC(J,K,I),K=1,20)
30      CONTINUE
20    CONTINUE
      DO 40 I = 1,14
        READ(20,*)
40    CONTINUE
      DO 50 I = 1,11
        READ(20,*) (AERSOL(I,J),J=1,20)
50    CONTINUE
      READ(20,*)
      READ(20,*)
      DO 60 I = 1,11
        READ(20,*) (RESP(I,J),J=1,20)
60    CONTINUE
      READ(20,*)
      READ(20,*) AREADA
      READ(20,*) DFLEV
      READ(20,*) (RADIST(I,1),I=1,15)
      READ(20,*) (RADIST(I,2),I=1,15)
      READ(20,*) (RADIST(I,3),I=1,15)
      READ(20,*)
      READ(20,*) PKGSZ1,PKGSZ2
      READ(20,*) RPD,RR,RS,RU,FNOATT
      READ(20,*) BDF,XFARM,CULVL,BRATE
      READ(20,*) EXPLCF
      READ(20,*) WBLCF,GNDSGE
      READ(20,*) TOL
      READ(20,*) LISTLH
      READ(20,*) NKW1,NKW2,NKW3,NKW4,NKW5,NKW6,NKW7,NKW8
      READ(20,*) NDERV
      READ(20,*) TBDIHD,TBDTHD,TBPPH
      READ(20,*) NA,NC
      READ(20,*) PSAREA
      DO 70 I = 1, 6
```

```
      READ(20,*) (PSQLCT(J,I),J=1,18)
70 CONTINUE
   READ(20,*) PSPROB
   READ(20,*) TWOPI
   READ(20,*) FMU,DZ,VV
   READ(20,*) PI,Q6,WS,GAM
   READ(20,*) DELTIM
   READ(20,*) EMRCST
   READ(20,*) ONSCST
   READ(20,*) ECONF
   READ(20,*) NTHRS
   READ(20,*) EMTRSH
   READ(20,*) EPS
   READ(20,*) DLIM
   READ(20,*) MTAL
   READ(20,*) NREMLV,NE
   READ(20,*) REMLEV
   READ(20,*) ((PROBEF(I,J),I=1,30),J=1,4)
   READ(20,*)
   READ(20,*) IPOINT,NCS
   READ(20,*)
   READ(20,*)
   DO 80 I = 1, 25
     DO 90 J = 1, 2
       READ(20,'(A)') LABDOS(I,J)
90   CONTINUE
80 CONTINUE
   READ(20,*)
   READ(20,*)
   READ(20,'(8A)') (IKW1(I),I=1,8)
   READ(20,'(8A)') (IKW1(I),I=9,16)
   READ(20,'(2A)') (IKW1(I),I=17,18)
   READ(20,*)
   READ(20,'(6A)') (IKW2(I),I=1,6)
   READ(20,*)
   READ(20,'(4A)') (IKW3(I),I=1,4)
   READ(20,*)
   READ(20,'(8A)') (IKW4(I),I=1,8)
   READ(20,'(8A)') (IKW4(I),I=9,16)
   READ(20,'(8A)') (IKW4(I),I=17,24)
   READ(20,'(8A)') (IKW4(I),I=25,32)
   READ(20,*)
   READ(20,'(3A)') (IKW5(I),I=1,3)
   READ(20,*)
   READ(20,'(3A)') (IKW6(I),I=1,3)
   READ(20,*)
   READ(20,'(7A)') (IKW7(I),I=1,7)
   READ(20,*)
   READ(20,'(7A)') (IKW8(I),I=1,7)
   READ(20,*)
   READ(20,*)
```

```
      DO 100 I = 1, 38
        READ(20,'(A)') LDESCR(I)
  100 CONTINUE
      READ(20,*)
      READ(20,*)
      READ(20,'(8A)') (LABORG(I),I=1,8)
      READ(20,*)
      READ(20,'(3A)') (LABPOP(I),I=1,3)
      READ(20,*)
      READ(20,'(7A)') (LABMOD(I),I=1,7)
      READ(20,'(3A)') (LABMOD(I),I=8,10)
      READ(20,*)
      READ(20,'(5A)') (LBTRSH(I),I=1,5)
      READ(20,*)
      READ(20,'(4A)') (LABEF(I),I=1,4)
      READ(20,*)
      READ(20,'(10A)') (NVY(I),I=1,10)
      READ(20,'(6A)') (WORDS(I),I=1,6)

      READ(21,*) NISOTP,NPRP
      READ(21,*)
      READ(21,*)
      READ(21,'(8A)') (ISONAM(I),I=1,8)
      READ(21,'(8A)') (ISONAM(I),I=9,16)
      READ(21,'(8A)') (ISONAM(I),I=17,24)
      READ(21,'(8A)') (ISONAM(I),I=25,32)
      READ(21,'(8A)') (ISONAM(I),I=33,40)
      READ(21,'(8A)') (ISONAM(I),I=41,48)
      READ(21,'(8A)') (ISONAM(I),I=49,56)
      READ(21,'(4A)') (ISONAM(I),I=57,60)

      DO 110 I = 1, 42
        READ(21,*)
  110 CONTINUE

      READ(21,*)  H3WTR
      READ(21,*)  H3GAS
      READ(21,*)  C14ORG
      READ(21,*)  C14GAS
      READ(21,*)  P32
      READ(21,*)  S35
      READ(21,*)  CA45
      READ(21,*)  CR51
      READ(21,*)  MN54
      READ(21,*)  FE55
      READ(21,*)  FE59
      READ(21,*)  CO58
      READ(21,*)  CO60
      READ(21,*)  ZN65
      READ(21,*)  GA67
      READ(21,*)  KR85
```

```
      READ(21,*)  SR89
      READ(21,*)  SR90
      READ(21,*)  Y91
      READ(21,*)  ZR95
      READ(21,*)  NB94
      READ(21,*)  NB95
      READ(21,*)  MO99
      READ(21,*)  TC99
      READ(21,*)  RU103
      READ(21,*)  RU106
      READ(21,*)  SB125
      READ(21,*)  TE125M
      READ(21,*)  TE127M
      READ(21,*)  TE127
      READ(21,*)  TE129M
      READ(21,*)  TE129
      READ(21,*)  I125
      READ(21,*)  I129
      READ(21,*)  I131
      READ(21,*)  XE133
      READ(21,*)  CS134
      READ(21,*)  CS137
      READ(21,*)  CE141
      READ(21,*)  CE144
      READ(21,*)  PM147
      READ(21,*)  SM151
      READ(21,*)  EU152
      READ(21,*)  EU154
      READ(21,*)  EU155
      READ(21,*)  U233
      READ(21,*)  U235
      READ(21,*)  U238
      READ(21,*)  NP237
      READ(21,*)  PU236
      READ(21,*)  PU238
      READ(21,*)  PU239
      READ(21,*)  PU240
      READ(21,*)  PU241
      READ(21,*)  PU242
      READ(21,*)  AM241
      READ(21,*)  AM243
      READ(21,*)  CM242
      READ(21,*)  CM244
      READ(21,*)  CF252

      RETURN
      END
*DECK,RDINFR
```

```
      SUBROUTINE RDINFR (DONE)
C
C     THIS ROUTINE WILL INPUT THE DATA IN A FREE FORMAT USING KEYWORDS
C     AS INDENTIFIERS
C
      LOGICAL DONE
C
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      COMMON /IKW/ IKW1(18),IKW2(6),IKW3(4),IKW4(32),IKW5(3),IKW6(3),
     *  IKW7(7),IKW8(7)
      COMMON /NIKW/ NKW1,NKW2,NKW3,NKW4,NKW5,NKW6,NKW7,NKW8
      CHARACTER*10 IKW1,IKW2,IKW3,IKW4,IKW5,IKW6,IKW7,IKW8
      COMMON /CARD/ IPOINT,KBYTE,CONTUF,NRFLAG,NCS
      LOGICAL CONTUF,NRFLAG
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
```

```
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
      COMMON /CCARD/ ICARD,TEST,ICARDI,NVY(10)
      CHARACTER*10 ICARD*80, TEST,ICARDI,NVY*1
      COMMON /HEADER/ CDATE, KLOCK, TITLE
      CHARACTER CDATE*9, KLOCK*8, TITLE*80
C
      COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
      REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
      LOGICAL RSPEC
      COMMON /MAXSZ/ MAXISO,MAXSEV,MAXGRP,MAXRAD,MAXDSP,MAXARS,
     *  MAXMOD,MAXNDL,MAXMAT,NPRAM,RSPHLF
      COMMON /P/   H3WTR(11), H3GAS(11), C14ORG(11),C14GAS(11),
     *  P32(11),   S35(11),   CA45(11),  CR51(11),  MN54(11),
     *  FE55(11),  FE59(11),  CO58(11),  CO60(11),  ZN65(11),
     *  GA67(11),  KR85(11),  SR89(11),  SR90(11),  Y91(11),
     *  ZR95(11),  NB94(11),  NB95(11),  MO99(11),  TC99(11),
     *  RU103(11), RU106(11), SB125(11), TE125M(11),TE127M(11),
     *  TE127(11), TE129M(11),TE129(11), I125(11),  I129(11),
     *  I131(11),  XE133(11), CS134(11), CS137(11), CE141(11),
     *  CE144(11), PM147(11), SM151(11), EU152(11), EU154(11),
     *  EU155(11), U233(11),  U235(11),  U238(11),  NP237(11),
     *  PU236(11), PU238(11), PU239(11), PU240(11), PU241(11),
     *  PU242(11), AM241(11), AM243(11), CM242(11), CM244(11),
     *  CF252(11), DUMMY(715),NISOTP,NPRP
      REAL KR85,NB94,NB95,I129,I131,NP237,MN54,MO99,I125
      DIMENSION ACCDNT(11,125)
      EQUIVALENCE (ACCDNT(1,1),H3WTR(1))
      COMMON /O/ ISONAM(125)
      CHARACTER*10 ISONAM

      CHARACTER*10 TEST1
      DIMENSION TMP(25)
      LOGICAL MODES(10), LMODES(10), OLD
      INTEGER NISO
      SAVE NISO

      COMMON /CRSUM/ RSUM(8)

      DONE = .FALSE.
      KBYTE = 10
      DO 10 J = 1, MAXMOD
        MODES(J) = .FALSE.
        LMODES(J) = .FALSE.
   10 CONTINUE

   20 CALL NEXT
   30 DO 40 I = 1, NKW1
        IF (TEST.EQ.IKW1(I)) GO TO 50
   40 CONTINUE
```

```
      WRITE (4,1450) TEST
      STOP
   50 GO TO (60,70,80,90,170,310,490,600,630,860,1150,1260,1320,1370,
     *  1380,1390,1430,1420), I
      CALL GOTOER ('RDINFR-1')
C
C     DIMEN
   60 NISO = VALUE(DMY)
      NSEV = VALUE(DMY)
      NGROUP = VALUE(DMY)
      NPKG = NGROUP
      NRAD = VALUE(DMY)
      NAREAS = VALUE(DMY)
      IF ((NISO.GT.MAXISO).OR.(NSEV.GT.MAXSEV)) GO TO 1440
      IF ((NGROUP.GT.MAXGRP).OR.(NRAD.GT.MAXRAD)) GO TO 1440
      IF (NAREAS.GT.MAXARS) GO TO 1440
      GO TO 20
C
C     PARM
   70 IRNKC = VALUE(DMY)
      IANA = VALUE(DMY)
      IUOPT = VALUE(DMY)
      IF ((IUOPT .LT. 1) .OR. (IUOPT .GT. 3)) THEN
          WRITE (4, 1455) IUOPT
          STOP
      ENDIF
      ISEN = VALUE(DMY)
      IPSQSB = VALUE(DMY)
      GO TO 20
C
C     POPDEN
   80 CALL ARRAY (POPDEN,NPOP)
      GO TO 20
C
C     PACKAGE
   90 CALL NEXT
      DO 100 I = 1, NKW2
        IF (TEST.EQ.IKW2(I)) GO TO 110
  100 CONTINUE
      GO TO 30
  110 GO TO (120,120,140,150,160,1420), I
      CALL GOTOER ('RDINFR-2')
  120 DO 130 I = 1, NGROUP
        CALL NEXT
        CALL SHIFTC (TEST,LABGRP(I))
  130 CONTINUE

      GO TO 90
  140 CALL ARRAY (PKGCDM,NPKG)
      GO TO 90
  150 PKGSZ1 = VALUE(DMY)
```

```
      GO TO 90
  160 PKGSZ2 = VALUE(DMY)
      GO TO 90
C
C     SHIPMENT
  170 CALL NEXT
  180 DO 190 I = 1, NKW3
        IF (TEST.EQ.IKW3(I)) GO TO 200
  190 CONTINUE
      GO TO 30
  200 GO TO (210,210,280,1420), I
      CALL GOTOER ('RDINFR-3')
  210 DO 220 I = 1, NISO
        CALL NEXT
        CALL SHIFTC (TEST,LABISO(I))
  220 CONTINUE
      GO TO 170
  280 CALL NEXT
      IF (TEST.NE.'NMODE') GO TO 180
      NM = VALUE(DMY)
      IF (NM.GT.MAXMOD) GO TO 1435
      CALL ARRAY (FKMPS(1,NM),NISO)
      GO TO 280
C
C     NORMAL
  310 CALL NEXT
  315 DO 320 I = 1, NKW4
        IF (TEST.EQ.IKW4(I)) GO TO 330
  320 CONTINUE
      GO TO 30
  330 IF (I.LE.MAXNDL) GO TO 350
      I = I-MAXNDL
      GO TO (360,390,395,400,410,420,1420), I
      CALL GOTOER ('RDINFR-4')
  350 ND = I
      CALL NEXT
      IF (TEST.NE.'NMODE') GO TO 315
      NM = VALUE(DMY)
      IF (NM.GT.MAXMOD) GO TO 1435
      DNORML(NM,ND) = VALUE(DMY)
      GO TO 310
  360 NM = VALUE(DMY)
      IF (NM.GT.MAXMOD) GO TO 1435
C     THIS CALL TO ARRAY IS NECESSARY SO THAT ONE CAN STILL USE THE
C     REPEAT CHARACTER (*) ON INPUT
      CALL ARRAY (TMP,NDLAB)
      DO 370 I = 1, NDLAB
  370   DNORML(NM,I) = TMP(I)
      GO TO 310
  390 RPD = VALUE(DMY)
      GO TO 310
```

```
  395 RR = VALUE(DMY)
      GO TO 310
  400 RS = VALUE(DMY)
      GO TO 310
  410 RU = VALUE(DMY)
      GO TO 310
  420 FNOATT = VALUE(DMY)
      GO TO 310
C
C     ACCIDENT
  490 CALL NEXT
  500 DO 510 I = 1, NKW5
        IF (TEST.EQ.IKW5(I)) GO TO 520
  510 CONTINUE
      GO TO 30
  520 GO TO (530,540,1420), I
      CALL GOTOER ('RDINFR-5')
  530 CALL NEXT
      IF (TEST.NE.'NMODE') GO TO 500
      NM = VALUE(DMY)
      IF (NM.GT.MAXMOD) GO TO 1435
      ARATMZ(NM,1) = VALUE(DMY)
      ARATMZ(NM,2) = VALUE(DMY)
      ARATMZ(NM,3) = VALUE(DMY)
      GO TO 530
  540 CALL NEXT
  550 IF (TEST.NE.'NPOP') GO TO 500
      NP = VALUE(DMY)
      IF (NP.GT.NPOP) GO TO 590
  560 CALL NEXT
      IF (TEST.NE.'NMODE') GO TO 550
      NM = VALUE(DMY)
      IF (NM.GT.MAXMOD) GO TO 1435
      CALL ARRAY (TMP,NSEV)
      DO 580 I = 1, NSEV
  580   SEVFRC(NM,I,NP) = TMP(I)
      GO TO 560
  590 WRITE (4,1500) NP,NPOP
      STOP
C
C     ECONOMIC
  600 CALL NEXT
      IF (TEST.EQ.'ECONF') THEN
        ECONF = VALUE(DMY)
        GO TO 600
      ELSEIF (TEST.EQ.'ONSCST') THEN
        CALL ARRAY (ONSCST,4)
        GO TO 600
      ELSEIF (TEST.EQ.'EMRCST') THEN
        CALL ARRAY (EMRCST,NSEV)
        GO TO 600
```

```
      ENDIF
      GO TO 30
C
C     MATERIAL
  630 CALL NEXT
  640 DO 650 I = 1, NKW6
        IF (TEST.EQ.IKW6(I)) GO TO 660
  650 CONTINUE
      GO TO 30
  660 GO TO (760,810,1420), I
      CALL GOTOER ('RDINFR-6')
  760 CALL NEXT
      IF (TEST.EQ.'NORG') GO TO 780
      IF (TEST.NE.'NISO'.AND.TEST.NE.'NMAT') GO TO 640
      ISO = VALUE(DMY)
      IF (ISO.GT.NISO) GO TO 800
      CALL ARRAY (TMP,NORG)
      DO 770 I = 1, NORG
  770   RPCVAL(ISO,I) = TMP(I)
      GO TO 760
  780 NO = VALUE(DMY)
      IF (NO.GT.NORG) GO TO 790
      CALL ARRAY (RPCVAL(1,NO),NISO)
      GO TO 760
  790 WRITE (4,1510) NO,NORG
      STOP
  800 WRITE (4,1520) ISO,NISO
      STOP
  810 CALL NEXT
      IF (TEST.EQ.'NORG') GO TO 830
      IF (TEST.NE.'NISO'.AND.TEST.NE.'NMAT') GO TO 640
      ISO = VALUE(DMY)
      IF (ISO.GT.NISO) GO TO 850
      CALL ARRAY (TMP,NORG)
      DO 820 I = 1, NORG
  820   INGVAL(ISO,I) = TMP(I)
      GO TO 810
  830 NO = VALUE(DMY)
      IF (NO.GT.NORG) GO TO 840
      CALL ARRAY (INGVAL(1,NO),NISO)
      GO TO 810
  840 WRITE (4,1510) NO,NORG
      STOP
  850 WRITE (4,1520) ISO,NISO
      STOP
C
C     RELEASE
  860 CALL NEXT
  870 DO 880 I = 1, NKW7
        IF (TEST.EQ.IKW7(I)) GO TO 890
  880 CONTINUE
```

```
      GO TO 30
  890 GO TO (900,910,920,1000,1010,1020,1420), I
      CALL GOTOER ('RDINFR-7')
  900 IFKW = 1
      GO TO 930
  910 IFKW = 2
      GO TO 930
  920 IFKW = 3
  930 CALL NEXT
      IF (TEST.EQ.'GROUP'.OR.TEST.EQ.'NPTYPE') GO TO 940
      IF (TEST.EQ.'DISP'.OR.TEST.EQ.'NIM') GO TO 960
      IF (TEST.NE.'NSEV') GO TO 870
      NS = VALUE(DMY)
      IF (NS.GT.NSEV) GO TO 1090
      IF (IFKW.EQ.1) CALL ARRAY (RFRAC(1,NS),NGROUP)
      IF (IFKW.EQ.2) CALL ARRAY (AERSOL(1,NS),MAXDSP)
      IF (IFKW.EQ.3) CALL ARRAY (RESP(1,NS),MAXDSP)
      GO TO 930
  940 IGRP = VALUE(DMY)
      IF (IGRP.GT.NGROUP) GO TO 1100
      IF (IFKW.NE.1) GO TO 1120
      CALL ARRAY (TMP,NSEV)
      DO 950 I = 1, NSEV
  950   RFRAC(IGRP,I) = TMP(I)
      GO TO 930
  960 IDSP = VALUE(DMY)
      IF (IDSP.GT.MAXDSP) GO TO 1110
      IF (IFKW.EQ.1) GO TO 1130
      CALL ARRAY (TMP,NSEV)
      IF (IFKW.EQ.3) GO TO 980
      DO 970 I = 1, NSEV
  970   AERSOL(IDSP,I) = TMP(I)
      GO TO 930
  980 DO 990 I = 1, NSEV
  990   RESP(IDSP,I) = TMP(I)
      GO TO 930
 1000 CALL ARRAY (AREADA,NAREAS)
      GO TO 860
 1010 CALL ARRAY (DFLEV,NAREAS)
      GO TO 860
 1020 CALL ARRAY (PSPROB,6)
      GO TO 860
C
 1090 WRITE (4,1540) NS,NSEV
      GO TO 1140
 1100 WRITE (4,1550) IGRP,NGROUP
      GO TO 1140
 1110 WRITE (4,1560) IDSP,MAXDSP
      GO TO 1140
 1120 WRITE (4,1570)
      GO TO 1140
```

```
 1130 WRITE (4,1580)
 1140 STOP
C
C     OTHER
 1150 CALL NEXT
 1160 DO 1170 I = 1, NKW8
        IF (TEST.EQ.IKW8(I)) GO TO 1180
 1170 CONTINUE
      GO TO 30
 1180 GO TO (1190,1200,1210,1220,1230,1240,1420), I
      CALL GOTOER ('RDINFR-8')
 1190 CALL NEXT
      IF (TEST.NE.'NPOP') GO TO 1160
      NP = VALUE(DMY)
      IF (NP.GT.NPOP) GO TO 1250
      CALL ARRAY (RADIST(1,NP),NRAD)
      GO TO 1190
 1200 BDF = VALUE(DMY)
      GO TO 1150
 1210 XFARM = VALUE(DMY)
      GO TO 1150
 1220 CULVL = VALUE(DMY)
      GO TO 1150
 1230 BRATE = VALUE(DMY)
      GO TO 1150
 1240 ITRAIN = VALUE(DMY)
      GO TO 1150
 1250 WRITE (4,1590) NP,NPOP
      STOP
C
C     DEFINE
 1260 CALL NEXT
      DO 1270 J = 1, NISOTP
        IF (TEST.EQ.ISONAM(J)) THEN
          JPT = J
          GO TO 1280
        ENDIF
 1270 CONTINUE
      NISOTP = NISOTP+1
      JPT = NISOTP
      ISONAM(JPT) = TEST
 1280 CONTINUE
      CALL ARRAY (TMP,NPRP)
      DO 1290 J = 1, NPRP
        ACCDNT(J,JPT) = TMP(J)
 1290 CONTINUE
      GO TO 20
C
CCCCCCCCCCCCCCC
C     ISOTOPES
 1320 CONTINUE
```

```
      NLNK = 0
      RSPEC = .FALSE.
      NMODE = 0
      MCOUNT = 0
      ICOUNT = 0
      DIST = .FALSE.
      PCKG = .FALSE.
      DO 1322 I = 1, MAXMOD
        NMAT(I) = 0
 1322 CONTINUE
      DO 1324 I = 1, MAXMAT
        NIS(I) = 0
 1324 CONTINUE
 1330 CONTINUE
      IF (TEST.EQ.'ISOTOPES') THEN
        NM = VALUE(DMY)
        IF (ABS(NM).GT.MAXMOD) GO TO 1435
        IEX = 0
        IF (NM .LT. 0.0) IEX = 1
        NM = ABS(NM)
        SPY = VALUE(DMY)
        PPS = VALUE(DMY)
        TI = VALUE(DMY)
        FG = VALUE(DMY)
        FN = VALUE(DMY)
        CALL NEXT
        CALL SHIFTC(TEST,TEST)
C ..... HAS THE MODE BEEN USED BEFORE IN THIS RUN
        IF (MODES(NM)) THEN
C . . . . HAS THE MATERIAL BEEN LISTED WITH THIS MODE
          DO 1334 M = 1, NMAT(NM)
            IF (LABMAT(MATSAV(NM,M)).EQ.TEST) THEN
              WRITE (4,1620) TEST,NM
              STOP
            ENDIF
 1334     CONTINUE
C . . . . . IS THE EXCLUSIVE USE DIFFERENT
          IF (IEX.NE.IEXCLU(NM)) THEN
            WRITE (4,1630) NM
            STOP
          ENDIF
C . . . . . IS THE NUMBER OF SHIPMENTS DIFFERENT
          IF (SPY.NE.TABSPY(NM)) THEN
            WRITE (4,1640) TABSPY(NM), NM
            STOP
          ENDIF
        ELSE
C . . . . . SET THE MODE TO USED AND SAVE IT
          NMODE = NMODE+1
          MODES(NM) = .TRUE.
          MODSAV(NMODE) = NM
```

```
        ENDIF
C ..... HAS THE MATERIAL BEEN LISTED PREVIOUSLY
        MAT = 0
        OLD = .FALSE.
 1338   CONTINUE
        IF ((.NOT.OLD).AND.(MAT.LE.MCOUNT)) THEN
          MAT = MAT+1
          OLD = (MCOUNT.NE.0).AND.(LABMAT(MAT).EQ.TEST)
          GOTO 1338
        ENDIF
C ..... NEW MATERIAL
        IF (.NOT. OLD) THEN
          MCOUNT = MCOUNT+1
          IF (MCOUNT.GT.MAXMAT) THEN
            WRITE (4,'(A,I2)') ' TOO MANY MATERIALS, MAX IS ',MAXMAT
            STOP
          ENDIF
          MAT = MCOUNT
          FRGAMA(MAT) = FG
          FRNEUT(MAT) = FN
          LABMAT(MAT) = TEST
        ELSE
C ....... MAT IS SET TO LABMAT LOCATION OF THIS MATERIAL PREVIOUSLY READ IN
          IF ((FRGAMA(MAT).NE.FG).OR.(FRNEUT(MAT).NE.FN)) THEN
            WRITE (4,1650) LABMAT(MAT),FRGAMA(MAT),FRNEUT(MAT)
            STOP
          ENDIF
        ENDIF
        NMAT(NM) = NMAT(NM)+1
        MATSAV(NM,NMAT(NM)) = MAT
        IEXCLU(NM) = IEX
        TABSPY(NM) = SPY
        PKGSHP(NM,MAT) = PPS
        TIPKG(NM,MAT) = TI
        CALL NEXT
      ELSEIF (TEST.EQ.'EOF') THEN
        IF (RSPEC) THEN
          NLMODE = 0
          DO 1340 I = 1, MAXMOD
            IF (LMODES(I)) THEN
C . . . . . THIS MODE WAS LISTED UNDER LINKS
              IF ((NMODE.EQ.1).AND.(MCOUNT.EQ.1)) THEN
C . . . . . . . "ISOTOPES" LISTED ONLY ONCE
C . . . . . . . . ASSIGN ALL "ISOTOPES" VALUES TO THIS LINKS MODE (MAT=1)
                IEXCLU(I) = IEX
                TABSPY(I) = SPY
                PKGSHP(I,1) = PPS
                TIPKG(I,1) = TI
                NMAT(I) = 1
                MATSAV(I,1) = 1
              ELSE
```

```
C . . . . . "ISOTOPES" WAS LISTED MORE THAN ONCE -- THIS LINKS MODE
C . . . . . . SHOULD BE LISTED UNDER "ISOTOPES"
               IF (.NOT.MODES(I)) THEN
C . . . . . . . THIS MODE WAS NOT ONE LISTED
                 WRITE (4,1610)
                 STOP
               ENDIF
             ENDIF
C . . . . . NLMODE HOLDS THE NUMBER OF MODES ACTUALLY USED IN THE LINKS
             NLMODE = NLMODE + 1
C . . . . . MODSAV HOLDS THE MODE NUMBERS OF MODES ACTUALLY USED IN LINKS
             MODSAV(NLMODE) = I
           ENDIF
 1340    CONTINUE
C . . . . ASSIGN COMMOM VARIABLE NMODE THE NUMBER OF MODES USED
         NMODE = NLMODE
       ENDIF
C . . . IF NOT ROUTE SPECIFIC NMODE HAS THE CORRECT NUMBER OF MODES AND
C . . . MODSAV HAS THE CORRECT MODES USED.
       RETURN
     ELSEIF (TEST.EQ.'LINK') THEN
       RSPEC = .TRUE.
       NLNK = NLNK+1
       NM = VALUE(DMY)
       IF (NM.GT.MAXMOD) GO TO 1435
       LMODES(NM) = .TRUE.
       LMODE(NLNK) = NM
       LDIST(NLNK) = VALUE(DMY)
       LSPED(NLNK) = VALUE(DMY)
       LPOPD(NLNK) = VALUE(DMY)
       LVDEN(NLNK) = VALUE(DMY)
       LARAT(NLNK) = VALUE(DMY)
       CALL NEXT
       IF (TEST(1:1).EQ.'R')  LZONE(NLNK) = 1
       IF (TEST(1:1).EQ.'S')  LZONE(NLNK) = 2
       IF (TEST(1:1).EQ.'U')  LZONE(NLNK) = 3
       LTYPE(NLNK) = VALUE(DMY)
       CALL NEXT
     ELSEIF (TEST.EQ.'DISTKM') THEN
       DIST = .TRUE.
 1344  CONTINUE
       CALL NEXT
       IF (TEST.EQ.'NMODE') THEN
         NM = VALUE(DMY)
         IF (NM.GT.MAXMOD) GO TO 1435
         DISTKM(NM) = VALUE(DMY)
         GOTO 1344
       ENDIF
     ELSEIF (TEST.EQ.'PKGSIZ') THEN
       PCKG = .TRUE.
 1346  CONTINUE
```

```
        CALL NEXT
        CALL SHIFTC(TEST,TEST1)
        DO 1350 M = 1, MCOUNT
          IF (TEST1.EQ.LABMAT(M)) GOTO 1352
 1350   CONTINUE
C . . . NOT A MATERIAL NAME
        GOTO 1354
 1352   CONTINUE
C . . . FOUND THE MATERIAL NAME AT LOCATION M
        PKGSIZ(M) = VALUE(DMY)
        GOTO 1346
 1354   CONTINUE
      ELSE
C ...   IT'S AN ISOTOPE LISTED IN LABISO, GET THE ISOTOPE DATA.
        CALL SHIFTC (TEST,TEST1)
        DO 1358 I = 1, NISO
           IF (TEST1 .EQ. LABISO(I)) GO TO 1360
 1358   CONTINUE
        WRITE (4,*) ' ISOTOPE ',TEST1,' NOT LISTED'
        STOP
 1360   CONTINUE
        DO 1362 N = 1, NISOTP
          IF (TEST.EQ.ISONAM(N)) GO TO 1364
 1362   CONTINUE
        WRITE (4,*) ' ISOTOPE NAMED ',TEST,' NOT IN DATA SET'
        STOP
 1364   CONTINUE
C . . . ISOTOPE IS LISTED IN LABISO WITH INDEX I
        ICOUNT = ICOUNT+1
        IF (ICOUNT.GT.MAXISO) THEN
          WRITE (4,1660)
          STOP
        ENDIF
        NIS(MAT) = NIS(MAT)+1
        LIBSAV(ICOUNT) = I
        ISOSAV(MAT,NIS(MAT)) = ICOUNT
        CIPKG(ICOUNT) = VALUE(DMY)
        CALL NEXT
        CALL SHIFTC (TEST,TEST)
        DO 1366 J = 1, NGROUP
          IF (TEST.EQ.LABGRP(J)) GO TO 1368
 1366   CONTINUE
        WRITE (4,'(A)')  ' ',TEST,' IS NOT LISTED ABOVE'
        STOP
 1368   CONTINUE
        IPCGRP(ICOUNT) = J
        IDISP(ICOUNT) = VALUE(DMY)
C
        TABHLF(I) = ACCDNT(1,N)
        PHTENG(I) = ACCDNT(2,N)
        CLDOSF(I) = ACCDNT(3,N)
```

```
        FDTRAN(I) = ACCDNT(6,N)
        SLTRAN(I) = ACCDNT(7,N)
        VELDEP(I) = ACCDNT(8,N)
        LNGTAB(I) = ACCDNT(9,N)
C ....  RPCVAL( ,4) AND INGVAL( ,4) TAKEN FROM ACCDNT ARE FOR EFFECTIVE
DOSE
C ....  RPCVAL( ,1) AND RPCVAL( ,2) ARE USED IN EARLY EFFECTS CALCULATIONS
        IF (URSKF) THEN
           RPCVAL(I,4) = ACCDNT(4,N)
           INGVAL(I,4) = ACCDNT(5,N)
           RPCVAL(I,1) = ACCDNT(10,N)
           RPCVAL(I,2) = ACCDNT(11,N)
        ENDIF
C       COMPUTE RESUSPENSION DOSE FACTOR
        AR = ALOG(2.0)/RSPHLF
        HLF = TABHLF(I)
        IF (HLF.NE.0.0) THEN
          ALAM2 = 0.693/HLF
          ALAM1 = AR+ALAM2
          IF (HLF.GT.251) THEN
            TAMP = (1.0E-5/ALAM1)*(1.0-EXP(-18250.0*ALAM1))
            TEMP = (1.0E-9/ALAM2)*(1.0-EXP(-18250.0*ALAM2))
          ELSE
            TAMP = (1.0E-5/ALAM1)
            TEMP = (1.0E-9/ALAM2)
          ENDIF
          TABRDF(I) = 1.0+8.64E4*VELDEP(I)*(TEMP+TAMP)
        ELSE
          TABRDF(I) = 1.0
        ENDIF
        CALL NEXT
      ENDIF
      GO TO 1330
C
C     TITLE
 1370 CONTINUE
      TITLE(1:80) = ICARD(1:80)
      CALL CENTER (TITLE,IPOINT,80)
      IPOINT = NCS + 1
      GO TO 20
C
C     FORM
 1380 CALL NEXT
      URSKF = TEST.EQ.'UNIT'
      GO TO 20
C
C     TRANSFER
 1390 CALL NEXT
      IF (TEST.EQ.'GAMMA') THEN
        CALL ARRAY (TMP,5)
        DO 1400 J = 1, 5
```

```
          COEF(1,J) = TMP(J)
 1400     CONTINUE
          GO TO 1390
        ELSEIF (TEST.EQ.'NEUTRON') THEN
          CALL ARRAY (TMP,5)
          DO 1410 J = 1, 5
            COEF(2,J) = TMP(J)
 1410     CONTINUE
          GO TO 1390
        ELSE
          GO TO 30
        ENDIF
C
C       EOF
 1420 CONTINUE
C       ERROR CHECK FOR PASQUILL PROBABILITIES
        IF (IPSQSP.EQ.1) THEN
          SUM = 0.0
          DO 1425 I = 1, 6
            SUM = SUM+PSPROB(I)
 1425     CONTINUE
          IF (ABS(SUM-1.).GT.1.E-5) THEN
            WRITE (4,1530) (PSPROB(I),I=1,6),SUM
            STOP
          ENDIF
        ENDIF
        RETURN
C
C       EOI
 1430 CONTINUE
        WRITE (5,'(A10,8(1PE10.2))') LABMAT(1),(RSUM(J),J=1,8)
        DONE = .TRUE.
        RETURN
C
C ... MODE ENTRY ERROR MESSAGE
 1435 WRITE (4,1470) NM, MAXMOD
        STOP
C
 1440 WRITE (4,1600) NISO,MAXISO,NSEV,MAXSEV,NGROUP,MAXGRP,NRAD,MAXRAD,
     *  NAREAS,MAXARS
        STOP
C
 1450 FORMAT (//,5X,36HUNKNOWN IDENTIFIER DETECTED ON INPUT,/,A20)
 1455 FORMAT (//,5X,'IUOPT SHOULD BE 1, 2, OR 3.   IUOPT =',I3)
 1470 FORMAT (//,5X,'ARRAY INDEX ERROR, MODE ENTERED IS ',I2,
     *  '.  MODES MUST BE BETWEEN 1 AND ',I2)
 1500 FORMAT (//,5X,24HARRAY INDEX ERROR , NP =,I6,7H NPOP =,I6,//)
 1510 FORMAT (//,5X,24HARRAY INDEX ERROR , NO =,I6,7H NORG =,I6,//)
 1520 FORMAT (//,5X,25HARRAY INDEX ERROR , ISO =,I6,7H NISO =,I6,//)
 1530 FORMAT (//,2X,46HERROR IN PROBABILITIES FOR PASQUILL CATEGORIES,/,
     *5X,6(1PE12.5),/,5X,5HSUM =,1PE12.5)
```

```
 1540 FORMAT (//,5X,24HARRAY INDEX ERROR , NS =,I6,7H NSEV =,I6,//)
 1550 FORMAT (//,5X,'ARRAY INDEX ERROR , IGRP =',I6,' NGROUP =',I6,//)
 1560 FORMAT (//,5X,'ARRAY INDEX ERROR , IDSP =',I6,' MAXDSP =',I6,//)
 1570 FORMAT (//,5X,'GROUP OR NPTYPE ARE NOT APPROPRIATE KEYWORDS WITH',
     *        ' AERSOL OR RESP',//)
 1580 FORMAT (//,5X,'DISP OR NIM ARE NOT APPROPRIATE KEYWORDS WITH',
     *        ' RFRAC',//)
 1590 FORMAT (//,5X,24HARRAY INDEX ERROR , NP =,I6,7H NPOP =,I6,//)
 1600 FORMAT (//,3X,74HYOU ARE ATTEMPTING TO DIMENSION AN ARRAY LARGER T
     *HAN THE AVAILABLE STORAGE,//,10X,10HIDENTIFIER,7X,5HVALUE,9X,7HMAX
     *IMUM,/,13X,4HNISO,8X,I6,9X,I6,/,13X,4HNSEV,8X,I6,9X,I6,/,12X,6HNGR
     *OUP,7X,I6,9X,I6,/,13X,4HNRAD,8X,I6,9X,I6,/,12X,6HNAREAS,7X,I6,9X,
     *I6,//)
 1610 FORMAT (/,' WHEN USING LINKS; EITHER USE ONLY ONE "ISOTOPES" ',
     *  'LINE WHICH WILL'/' DEFINE ALL MODES, OR WRITE AN "ISOTOPES" ',
     *  'LINE FOR EACH MODE USED.',/)
 1620 FORMAT (/,' MATERIAL ',A10,' HAS ALREADY BEEN ASSIGNED TO MODE'
     *        ,I3,/)
 1630 FORMAT (/, ' THE EXCLUSIVE USE DESIGNATOR FOR MODE',I3,' HAS ',
     *           'ALREADY BEEN SET.',/,' CANNOT CHANGE IT.',/)
 1640 FORMAT (/,' THE NUMBER OF SHIPMENTS HAS ALREADY BEEN SET TO ',
     *        F7.1,' FOR MODE',I3,/)
 1650 FORMAT (/,' FRACTIONS OF GAMMA AND NEUTRON FOR MATERIALS CANNOT',
     *    ' BE CHANGED.',/,' MATERIAL ',A10,' HAS FRACTION OF GAMMA =',
     *    1PE9.2,/,' AND FRACTION OF NEUTRON =',1PE9.2,/)
 1660 FORMAT (/,' TOO MANY ISOTOPES IN THE MATERIALS, MAXIMUM ',
     *            'ALLOWED IS 65',/)
      END
*DECK,SCAN
      SUBROUTINE SCAN
C
C     THIS ROUTINE SCANS THE CONTENTS OF A CARD AND USING THE DELIMITER
C     CHARACTERS DETERMINES THE LOCATION OF THE NEXT FIELD ITEM AND
C     PLACES IT INTO "TEST"
C
      COMMON /CARD/ IPOINT,KBYTE,CONTUF,NRFLAG,NCS
      LOGICAL CONTUF,NRFLAG
      COMMON /CCARD/ ICARD,TEST,ICARDI,NVY(10)
      CHARACTER*10 ICARD*80, TEST,ICARDI,NVY*1
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
C
      CONTUF = .FALSE.
      IF (IPOINT.GT.NCS) GO TO 30
      DO 20 I = IPOINT, NCS
        ICARDI = ICARD(I:I)
        DO 10 II = 1, 5
```

```
            IF (ICARDI.EQ.WORDS(II)) GO TO 20
      10    CONTINUE
            GO TO 40
      20 CONTINUE
      30 TEST(1:10) = '          '
         IPOINT = NCS+1
         RETURN
      40 ISTART = I
         IF (ISTART.EQ.NCS) GO TO 70
         IWS = ISTART+1
         DO 60 I = IWS, NCS
           ICARDI = ICARD(I:I)
           DO 50 II = 1, 6
             IF (ICARDI.EQ.WORDS(II)) GO TO 80
      50    CONTINUE
      60 CONTINUE
      70 I = NCS+1
C
      80 ISTOP = I-1
         IPOINT = ISTOP+2
         TEST(1:10) = ICARD(ISTART:ISTOP)
         NRFLAG = .FALSE.
         IF (ICARDI.EQ.WORDS(6)) NRFLAG = .TRUE.
         RETURN
         END
*DECK,SCRIPT
         SUBROUTINE SCRIPT (WORD)
         CHARACTER AB*7, ALP*36, LINE*80, WORD*12, BLANK*80
         DIMENSION AB(36,7),LINE(7)
         DATA ALP / 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789$'/
         DATA BLANK / ' '/
         DATA ((AB(I,J),I=1,6),J=1,7) / ' AAA   ','BBBB   ',' CCCC  ',
        *  'DDDD   ','EEEEE  ','FFFFF  ','A   A  ','B   B  ','C      ',
        *  'D   D  ','E      ','F      ','A   A  ','B   B  ','C      ',
        *  'D   D  ','E      ','F      ','A   A  ','BBBB   ','C      ',
        *  'D   D  ','EEEE   ','FFFF   ','AAAAA  ','B   B  ','C      ',
        *  'D   D  ','E      ','F      ','A   A  ','B   B  ','C      ',
        *  'D   D  ','E      ','F      ','A   A  ','BBBB   ',' CCCC  ',
        *  'DDDD   ','EEEEE  ','F      '/
         DATA ((AB(I,J),I=7,12),J=1,7) / ' GGGG  ','H   H  ','IIIII  ',
        *  'JJJJJ  ','K   K  ','L      ','G      ','H   H  ','  I    ',
        *  '   J   ','K  K   ','L      ','G      ','H   H  ','  I    ',
        *  '   J   ','K K    ','L      ','G      ','HHHHH  ','  I    ',
        *  '   J   ','KK     ','L      ','G GGG  ','H   H  ','  I    ',
        *  '   J   ','K K    ','L      ','G   G  ','H   H  ','  I    ',
        *  'J  J   ','K  K   ','L      ',' GGG   ','H   H  ','IIIII  ',
        *  ' JJ    ','K   K  ','LLLLL  '/
         DATA ((AB(I,J),I=13,18),J=1,7) / 'M   M  ','N   N  ',' OOO   ',
        *  'PPPP   ',' QQQ   ','RRRR   ','MM MM  ','NN  N  ','O   O  ',
        *  'P   P  ','Q   Q  ','R   R  ','M M M  ','N N N  ','O   O  ',
        *  'P   P  ','Q   Q  ','R   R  ','M   M  ','N  NN  ','O   O  ',
```

```
     *  'PPPP   ','Q   Q  ','RRRR   ','M   M  ','N   N  ','O   O  ',
     *  'P      ','Q Q Q  ','R R    ','M   M  ','N   N  ','O   O  ',
     *  'P      ','Q  QQ  ','R  R   ','M   M  ','N   N  ',' OOO   ',
     *  'P      ',' QQQ Q ','R   R  '/
      DATA ((AB(I,J),I=19,24),J=1,7) / ' SSSS  ','TTTTT  ','U   U  ',
     *  'V   V  ','W   W  ','X   X  ','S      ','  T    ','U   U  ',
     *  'V   V  ','W   W  ','X   X  ','S      ','  T    ','U   U  ',
     *  'V   V  ','W   W  ',' X X   ',' SSS   ','  T    ','U   U  ',
     *  'V   V  ','W   W  ','  X    ','    S  ','  T    ','U   U  ',
     *  'V   V  ','W W W  ',' X X   ','    S  ','  T    ','U   U  ',
     *  ' V V   ','WW WW  ','X   X  ','SSSS   ','  T    ','UUUUU  ',
     *  '  V    ','W   W  ','X   X  '/
      DATA ((AB(I,J),I=25,30),J=1,7) / 'Y   Y  ','ZZZZZ  ',' 000   ',
     *  ' 11    ',' 222   ',' 333   ','Y   Y  ','    Z  ','0   0  ',
     *  '  1    ','    2  ','    3  ',' Y Y   ','   Z   ','0   0  ',
     *  '  1    ','    2  ','    3  ','  Y    ','  Z    ','0   0  ',
     *  '  1    ','   2   ','  33   ','  Y    ',' Z     ','0   0  ',
     *  '  1    ','  2    ','    3  ','  Y    ','Z      ','0   0  ',
     *  '  1    ',' 2     ','    3  ','  Y    ','ZZZZZ  ',' 000   ',
     *  '11111  ','22222  ',' 333   '/
      DATA ((AB(I,J),I=31,33),J=1,7) / '    4  ','55555  ','6      ',
     *                                 '4   4  ','5      ','6      ',
     *                                 '4   4  ','5      ','6      ',
     *                                 '44444  ','5555   ','6666   ',
     *                                 '    4  ','    5  ','6   6  ',
     *                                 '    4  ','5   5  ','6   6  ',
     *                                 '    4  ','5555   ',' 666   '/
      DO 10 J = 1, 7
        LINE(J) = BLANK
   10 CONTINUE
      LEN = INDEX(WORD,'$')-1
      IF (LEN.NE.0) THEN
        DO 30 J = 1, LEN
          JJ = INDEX(ALP(1:36),WORD(J:J))
          IF (JJ.NE.0) THEN
            DO 20 K = 1, 7
              LINE(K)((J-1)*7+3:J*7) = AB(JJ,K)
   20       CONTINUE
          ENDIF
   30   CONTINUE
      ELSE
        WRITE (4,*) 'THE $ IS ABSENT FROM',WORD
      ENDIF
      DO 40 J = 1, 7
        WRITE (4,'(A)') LINE(J)
   40 CONTINUE
      RETURN
      END
*DECK,SENSTV
      SUBROUTINE SENSTV (LEG)
C
```

```
C     THIS ROUTINE CALCULATES DERIVATIVES OF NORMAL PARAMETERS
C     FOR INCIDENT-FREE SENSITIVITY ANALYSIS FOR EACH POPULATION
C     GROUP AND MODE.
C     INCIDENT-FREE DOSES
C     1 - DOSE TO AIRLINE PASSENGERS
C     2 - DOSE TO CREWMEN (NON-EXCLUSIVE USE)
C     9 - DOSE TO CREWMEN (EXCLUSIVE USE)
C     3 - DOSE TO FLIGHT ATTENDANTS
C     4 - DOSE TO HANDLERS (NON-EXCLUSIVE USE)

C     10 - DOSE TO HANDLERS (EXCLUSIVE USE)
C     5 - DOSE TO PERSONS SURROUNDING THE TRANSPORT
C     LINK WHILE THE SHIPMENT IS MOVING (OFF LINK)
C     6 - DOSE TO PERSONS SHARING THE TRANSPORT LINK
C     WITH THE SHIPMENT (ON LINK)
C     7 - DOSE TO POPULATION DURING SHIPMENT STOPS
C     8 - DOSE TO WAREHOUSE PERSONNEL WHILE PACKAGE IS IN STORAGE
C
C     DOSE2(1) = Fraction of rural dose of DOSEN(2) + DOSEN(9)
C     DOSE2(2) = Fraction of suburban dose of DOSEN(2) + DOSEN(9)
C     DOSE2(3) = Fraction of urban dose of DOSEN(2) + DOSEN(9)
C
C     DOSE5(1) = Fraction of rural dose of DOSEN(5)
C     DOSE5(2) = Fraction of suburban dose of DOSEN(5)
C     DOSE5(3) = Fraction of urban dose of DOSEN(5)
C     DOSE5(4) = Fraction of DOSEN(5) using rural velocity
C     DOSE5(5) = Fraction of DOSEN(5) using suburban velocit
C     DOSE5(6) = Fraction of DOSEN(5) using urban velocity
C     DOSE5(7) = Fraction of DOSEN(5) with rural and suburban freeway
travel
C     DOSE5(8) = Fraction of DOSEN(5) with urban city street travel
C     DOSE5(9) = Fraction of DOSEN(5) travel during rush hour
C     DOSE5(10) = Fraction of DOSEN(5) with RPD variable
C     DOSE5(11) = Fraction of DOSEN(5) with RR variable
C     DOSE5(12) = Fraction of DOSEN(5) with RS variable
C     DOSE5(13) = Fraction of DOSEN(5) with RU variable
C
C     DOSE6(1) = Fraction of rural dose of DOSEN(6)
C     DOSE6(2) = Fraction of suburban dose of DOSEN(6)
C     DOSE6(3) = Fraction of urban dose of DOSEN(6)
C     DOSE6(4) = Fraction of DOSEN(6) using rural velocity
C     DOSE6(5) = Fraction of DOSEN(6) using suburban velocity
C     DOSE6(6) = Fraction of DOSEN(6) using urban velocity
C     DOSE6(7) = Fraction of DOSEN(6) with rural and suburban freeway
travel
C     DOSS6(8) = Fraction of DOSEN(6) with urban city street travel
C     DOSE6(9) = Fraction of DOSEN(6) travel during rush hour
C
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
```

```
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
C
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
C
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
      COMMON /CON/ TOL,NDERV,NA,NC,TWOPI,PI,Q6,WS,GAM,FMU,DZ,VV
C
      COMMON /NSEN/ PRMNDS(10,21),STERM,ATERM,BTERM,D4,RVELR,RVELS,RVELU
C
      COMMON /SENDOS/ DOSE2(3), DOSE5(13), DOSE6(9)
C
      DO 10 I = 1, NDOSE
        DO 10 J = 1, IPRAM
          PRMNDS(I,J) = 0.
   10 CONTINUE
      RVELR2 = RVELR*RVELR
      RVELS2 = RVELS*RVELS
C
      IF (SPY.EQ.0) RETURN
C
C     DOSE IN WAREHOUSE PERSONNEL WHILE PACKAGE IS IN STORAGE
C     ALL MODES
      DOSE = DOSEN(8)
      IF (DOSE.EQ.0.0) GO TO 50
      PRMNDS(8,1) = DOSE/CAYZER
      IF (TI.EQ.0.0) GO TO 30
      PRMNDS(8,2) = DOSE/TI
   30 CONTINUE
      PRMNDS(8,3) = DOSE/PPS
      PRMNDS(8,4) = DOSE/SPY
      IF (DTSTOR.EQ.0.0) GO TO 40
      PRMNDS(8,5) = DOSE/DTSTOR
   40 CONTINUE
      IF (PDSTOR.EQ.0.0) GO TO 50
      PRMNDS(8,6) = DOSE/PDSTOR
   50 CONTINUE
      IF (RSTOR.EQ.0.0) GO TO 60
      PRMNDS(8,7) = -2.*DOSEN(8)/RSTOR
   60 CONTINUE
C
C     DOSE TO HANDLERS
      DOSE = D4
```

```
      IF (DOSE.EQ.0.0) GO TO 90
      IF (TI.EQ.0.0) GO TO 80
      PRMNDS(4,1) = DOSE/TI
   80 CONTINUE
      PRMNDS(4,2) = DOSE/PPS
      PRMNDS(4,3) = DOSE/SPY
      IF (HANDNO.EQ.0.0) GO TO 90
      PRMNDS(4,4) = DOSE/HANDNO
   90 CONTINUE
      IF (DPKG.LE.PKGSZ1) GO TO 110
      PRMNDS(4,5) = DOSE/DTHD
      PRMNDS(4,6) = DOSE/PPH
      IF (DPKG.GT.PKGSZ2) GO TO 100
      PRMNDS(4,7) = -2.*DOSE/DIHD
      PRMNDS(4,8) = DOSE/CAYZER
      GO TO 110
  100 CONTINUE
      PRMNDS(4,7) = -1.*DOSE/DIHD
      PRMNDS(4,8) = .5*DOSE/CAYZER
  110 CONTINUE
C
C     DOSE TO POPULATION DURING SHIPMENT STOPS
      DOSE = DOSEN(7)
      IF (DOSE.EQ.0.0) GO TO 170
      PRMNDS(7,1) = DOSE/CAYZER
      PRMNDS(7,4) = DOSE/SPY
      IF (PDST.NE.0.0) PRMNDS(7,5) = DOSE/PDST
      IF (DSTRVL.NE.0.0) PRMNDS(7,6) = DOSE/DSTRVL
      IF (DTST.NE.0.0) PRMNDS(7,7) = DOSE/DTST
      IF (IEXCLU(MODE).EQ.1) GO TO 160
C     NON-EXCLUSIVE USE
      IF (ATERM.GT.10.) GO TO 170
  140 IF (TI.EQ.0.0) GO TO 150
      PRMNDS(7,2) = DOSE/TI
  150 CONTINUE
      PRMNDS(7,3) = DOSE/PPS
      GO TO 170
C     EXCLUSIVE USE
  160 CONTINUE
      IF (.5*ATERM.GT.10.) GO TO 170
      GO TO 140
  170 CONTINUE
      IF (RST.EQ.0.0) GO TO 180
      PRMNDS(7,8) = -2.*DOSE/RST
  180 CONTINUE
      GO TO (200,480,590,590,360,430,200,200,200,200), MODE
      CALL GOTOER ('SENSTV-1')
C
  200 CONTINUE
C     TRUCK/VAN
C     DOSE TO PERSONS SURROUNDING THE TRANSPORT LINK WHILE THE
```

```
      DOSE = DOSEN(5)
      IF (DOSE.EQ.0.) GO TO 230
      PRMNDS(5,1) = DOSE/DSTRVL
      PRMNDS(5,2) = DOSE/CAYZER
      PRMNDS(5,5) = DOSE/SPY
      IF (IEXCLU(MODE).EQ.1) GO TO 220
C     NON-EXCLUSIVE USE
      IF (ATERM.GT.10.) GO TO 230
  210 PRMNDS(5,4) = DOSE/PPS
      IF (TI.EQ.0.0) GO TO 230
      PRMNDS(5,3) = DOSE/TI
      GO TO 230
  220 CONTINUE
C     EXCLUSIVE USE
      IF (.5*ATERM.GT.10.) GO TO 230
      GO TO 210
  230 CONTINUE
      IF (FTZNR .NE. 0.0)  PRMNDS(5,6) = DOSE5(1)/FTZNR
      IF (POPDR .NE. 0.0)  PRMNDS(5,7) = DOSE5(1)/POPDR
      PRMNDS(5,8) = -DOSE5(4)*RVELR
      IF (RPD .NE. 0.0)  PRMNDS(5,9) = DOSE5(10)/RPD
      IF (FTZNS .NE. 0.0)  PRMNDS(5,10) = DOSE5(2)/FTZNS
      IF (POPDS .NE. 0.0)  PRMNDS(5,11) = DOSE5(2)/POPDS
      PRMNDS(5,12) = -DOSE5(5)*RVELS
      IF (RS .NE. 0.0)  PRMNDS(5,13) = DOSE5(12)/RS
      IF (FTZNU .NE. 0.0)  PRMNDS(5,14) = DOSE5(3)/FTZNU
      IF (POPDU .NE. 0.0)  PRMNDS(5,15) = DOSE5(3)/POPDU
      PRMNDS(5,16) = -DOSE5(6)*RVELU
      IF (FCTST .NE. 0.0)  PRMNDS(5,17) = DOSE5(8)/FCTST
      IF (FCTST .NE. 1.0)  PRMNDS(5,17) =
     *                     PRMNDS(5,17) - (DOSE5(3)-DOSE5(8))/(1.-FCTST)
      IF (RU .NE. 0.0)  PRMNDS(5,18) = DOSE5(13)/RU
      IF (FTLFWY .NE. 0.0)  PRMNDS(5,19) = DOSE5(7)/FTLFWY
      IF (FTLFWY .NE. 1.0)  PRMNDS(5,19) =
     *         PRMNDS(5,19) - ((DOSE5(1)+DOSE5(2))-DOSE5(7))/(1.-FTLFWY)
      IF (FRSHR .NE. 0.0)  PRMNDS(5,20) = DOSE5(9)/FRSHR
      IF (FRSHR .NE. 1.0)  PRMNDS(5,20) =
     *                     PRMNDS(5,20) - (DOSEN(5)-DOSE5(9))/(1.-FRSHR)
      IF (RR .NE. 0.0)  PRMNDS(5,21) = DOSE5(11)/RR
C
C     DOSE TO PERSONS IN VEHICLES SHARING THE TRANSPORT LINK WITH
C     THE SHIPMENT (ON LINK)
      DOSE = DOSEN(6)
      IF (DOSE.EQ.0.0) GO TO 310
      PRMNDS(6,1) = DOSE/CAYZER
      PRMNDS(6,4) = DOSE/SPY
      PRMNDS(6,5) = DOSE/DSTRVL
      PRMNDS(6,6) = DOSE/PPV
      IF (IEXCLU(MODE).EQ.1) GO TO 300
C     NON-EXCLUSIVE USE
      IF (ATERM.GT.10.) GO TO 310
```

```
  290 PRMNDS(6,3) = DOSE/PPS
      IF (TI.EQ.0.0) GO TO 310
      PRMNDS(6,2) = DOSE/TI
      GO TO 310
  300 CONTINUE
C     EXCLUSIVE USE
      IF (.5*ATERM.GT.10.) GO TO 310
      GO TO 290
  310 CONTINUE
      IF (FTZNR .NE. 0.0)  PRMNDS(6,7) = DOSE6(1)/FTZNR
      IF (FTLFWY .NE. 0.0)  PRMNDS(6,8) = DOSE6(7)/FTLFWY
      IF (FTLFWY .NE. 1.0)  PRMNDS(6,8) =
     *          PRMNDS(6,8) - ((DOSE6(1)+DOSE6(2))-DOSE6(7))/(1.-FTLFWY)
      IF (TCNTPR .NE. 0.0)  PRMNDS(6,9) = DOSE6(1)/TCNTPR
      PRMNDS(6,10) = -2*DOSE6(4)*RVELR
      IF (FTZNS .NE. 0.0)  PRMNDS(6,11) = DOSE6(2)/FTZNS
      IF (FRSHR .NE. 0.0)  PRMNDS(6,12) = DOSE6(9)/FRSHR
      IF (FRSHR .NE. 1.0)  PRMNDS(6,12) =
     *                     PRMNDS(6,12) - (DOSEN(6)-DOSE6(9))/(1.-FRSHR)
      IF (TCNTPS .NE. 0.0)  PRMNDS(6,13) = DOSE6(2)/TCNTPS
      PRMNDS(6,14) = -2*DOSE6(5)*RVELS
      IF (FTZNU .NE. 0.0)  PRMNDS(6,15) = DOSE6(3)/FTZNU
      IF (FCTST .NE. 0.0)  PRMNDS(6,16) = DOSE6(8)/FCTST
      IF (FCTST .NE. 1.0)  PRMNDS(6,16) =
     *                     PRMNDS(6,16) - (DOSE6(3)-DOSE6(8))/(1.-FCTST)
      IF (TCNTPU .NE. 0.0)  PRMNDS(6,17) = DOSE6(3)/TCNTPU
      PRMNDS(6,18) = -2*DOSE6(6)*RVELU
C
  360 CONTINUE
C
C     DOSE TO CREWMEN - TRUCK, VAN, AIR
      IF (ADSTCW.EQ.0.0) RETURN
      TEST = CAYZER*ATERM/ADSTCW**2
      DOSE = DOSEN(2)+DOSEN(9)
      IF (DOSE.EQ.0.0) GO TO 390
      IF (TEST.GT.2..AND.(IEXCLU(MODE).EQ.1)) GO TO 380
      PRMNDS(2,1) = DOSE/CAYZER
      IF (TI.EQ.0.0) GO TO 370
      PRMNDS(2,2) = DOSE/TI
  370 PRMNDS(2,3) = DOSE/PPS
      IF (VELM.EQ.0.) GO TO 380
      PRMNDS(2,4) = -2.*2.8E-4*TEST*SPY*CREWNO*DSTRVL/(VELM*ADSTCW)
  380 PRMNDS(2,5) = DOSE/SPY
      PRMNDS(2,7) = DOSE/DSTRVL
      IF (CREWNO.EQ.0.0) GO TO 400
      PRMNDS(2,6) = DOSE/CREWNO
      GO TO 400
  390 IF (TEST.GT.2.) GO TO 400
      IF (VELM.EQ.0.) GO TO 400
      PRMNDS(2,4) = -2.*2.8E-4*TEST*SPY*CREWNO*DSTRVL/(VELM*ADSTCW)
  400 CONTINUE
```

```
      IF (FTZNR .NE. 0)  PRMNDS(2,8) = DOSE2(1)/FTZNR
      PRMNDS(2,9) = -DOSE2(1)*RVELR
      IF (FTZNS .NE. 0)  PRMNDS(2,10) = DOSE2(2)/FTZNS
      PRMNDS(2,11) = -DOSE2(2)*RVELS
      IF (FTZNU .NE. 0)  PRMNDS(2,12) = DOSE2(3)/FTZNU
      PRMNDS(2,13) = -DOSE2(3)*RVELU
      CALL SENSUM (SENPRM(1,LEG),PRMNDS)
      RETURN
C
C     AIR
  430 CONTINUE
C     PASSENGER AIR
C     DOSE TO AIRLINE PASSENGERS
      DOSE = DOSEN(1)
      IF (DOSE.EQ.0.0) GO TO 440
      PRMNDS(1,2) = DOSE/PPS
      PRMNDS(1,3) = DOSE/SPY
      PRMNDS(1,4) = DOSE/DSTRVL
      PRMNDS(1,5) = DOSE/PPV
      IF (TI.EQ.0.0) GO TO 440
      PRMNDS(1,1) = DOSE/TI
  440 CONTINUE
      CC = 8.3E-6*BTERM*DSTRVL*PPV
      PRMNDS(1,6) = -CC*FTZNR*RVELR2
      PRMNDS(1,7) = CC*RVELR
C
C     DOSE TO FLIGHT ATTENDANTS
      DOSE = DOSEN(3)
      IF (DOSE.EQ.0.) GO TO 460
      PRMNDS(3,2) = DOSE/PPS
      PRMNDS(3,3) = DOSE/SPY
      PRMNDS(3,4) = DOSE/DSTRVL
      PRMNDS(3,5) = DOSE/FNOATT
      IF (TI.EQ.0.) GO TO 460
      PRMNDS(3,1) = DOSE/TI
  460 CONTINUE
      DD = 8.3E-6*BTERM*DSTRVL*FNOATT
      PRMNDS(3,6) = -DD*FTZNR*RVELR2
      PRMNDS(3,7) = DD*RVELR
      GO TO 360
C
C     RAIL
  480 CONTINUE
C     DOSE TO CREW (RAIL)
      DOSE = DOSEN(2)+DOSEN(9)
      IF (DOSE.EQ.0) GO TO 500
      PRMNDS(2,1) = DOSE/CAYZER
      IF (TI.EQ.0) GO TO 490
      PRMNDS(2,2) = DOSE/TI
  490 CONTINUE
      PRMNDS(2,3) = DOSE/PPS
```

```
      PRMNDS(2,4) = DOSE/SPY
      IF (DSTRVL.EQ.0) GO TO 500
      PRMNDS(2,5) = DOSE/DSTRVL
  500 CONTINUE
C
C     DOSE TO PERSONS SURROUNDING THE TRANSPORT LINK WHILE THE
C     SHIPMENT IS MOVING (OFF LINK)
      DOSE = DOSEN(5)
      IF (DOSE.EQ.0.0) GO TO 530
      PRMNDS(5,1) = DOSE/DSTRVL
      PRMNDS(5,2) = DOSE/CAYZER
      PRMNDS(5,5) = DOSE/SPY
      IF (IEXCLU(MODE).EQ.1) GO TO 520
C     NON-EXCLUSIVE USE
      IF (ATERM.GT.10.) GO TO 530
  510 PRMNDS(5,4) = DOSE/PPS
      IF (TI.EQ.0.0) GO TO 530
      PRMNDS(5,3) = DOSE/TI
      GO TO 530
*     EXCLUSIVE USE
  520 IF (.5*ATERM.GT.10.) GO TO 530
      GO TO 510
  530 CONTINUE
      IF (FTZNR .NE. 0.0)  PRMNDS(5,6) = DOSE5(1)/FTZNR
      IF (POPDR .NE. 0.0)  PRMNDS(5,7) = DOSE5(1)/POPDR
      PRMNDS(5,8) = -DOSE5(1)*RVELR
      IF (RS .NE. 0.0)  PRMNDS(5,9) = DOSE5(12)/RS
      IF (POPDS .NE. 0.0)  PRMNDS(5,10) = DOSE5(2)/POPDS
      PRMNDS(5,11) = -DOSE5(2)*RVELS
      IF (FTZNS .NE. 0.0)  PRMNDS(5,12) = DOSE5(2)/FTZNS
      IF (FTZNU .NE. 0.0)  PRMNDS(5,13) = DOSE5(3)/FTZNU
      IF (POPDU .NE. 0.0)  PRMNDS(5,14) = DOSE5(3)/POPDU
      PRMNDS(5,15) = -DOSE5(3)*RVELU
      IF (RU .NE. 0.0)  PRMNDS(5,16) = DOSE5(13)/RU
      IF (RR .NE. 0.0)  PRMNDS(5,17) = DOSE5(11)/RR
C
C     DOSE TO PERSONS IN VEHICLES SHARING THE TRANSPORT
C     LINK WITH THE SHIPMENT (ON LINK)
      DOSE = DOSEN(6)
      IF (DOSE.EQ.0.0) GO TO 580
      PRMNDS(6,1) = DOSE/CAYZER
      PRMNDS(6,4) = DOSE/SPY
      PRMNDS(6,5) = DOSE/DSTRVL
      PRMNDS(6,6) = DOSE/PPV
      IF (IEXCLU(MODE).EQ.1) GO TO 570
C     NON-EXCLUSIVE USE
      IF (ATERM.GT.10.) GO TO 580
  560 PRMNDS(6,3) = DOSE/PPS
      IF (TI.EQ.0.0) GO TO 580
      PRMNDS(6,2) = DOSE/TI
      GO TO 580
```

```
  570 IF (.5*ATERM.GT.10.) GO TO 580
      GO TO 560
  580 CONTINUE
      IF (FTZNR .NE. 0.0)  PRMNDS(6,7) = DOSE6(1)/FTZNR
      IF (TCNTPR .NE. 0.0)  PRMNDS(6,8) = DOSE6(1)/TCNTPR
      PRMNDS(6,9) = -2*DOSE6(1)*RVELR
      IF (FTZNS .NE. 0.0)  PRMNDS(6,10) = DOSE6(2)/FTZNS
      IF (TCNTPS .NE. 0.0)  PRMNDS(6,11) = DOSE6(2)/TCNTPS
      PRMNDS(6,12) = -2.*DOSE6(2)*RVELS
      IF (FTZNU .NE. 0.0)  PRMNDS(6,13) = DOSE6(3)/FTZNU
      IF (TCNTPU .NE. 0.0)  PRMNDS(6,14) = DOSE6(3)/TCNTPU
      PRMNDS(6,15) = -2.*DOSE6(3)*RVELU
      CALL SENSUM (SENPRM(1,LEG),PRMNDS)
      RETURN
C
C     WATER
  590 CONTINUE
C     DOSE TO CREW (WATER)
      DOSE = DOSEN(2)+DOSEN(9)
      PRMNDS(2,1) = DOSE/CAYZER
      IF (TI.EQ.0.) GO TO 600
      PRMNDS(2,2) = DOSE/TI
  600 CONTINUE
      PRMNDS(2,3) = DOSE/PPS
      PRMNDS(2,4) = DOSE/SPY
      IF (DOSE.EQ.0) GO TO 610
      IF (DSTRVL.EQ.0) GO TO 610
      PRMNDS(2,5) = DOSE/DSTRVL
  610 CONTINUE
C
C     DOSE TO PERSONS SURROUNDING THE TRANSPORT LINK WHILE
C     THE SHIPMENT IS MOVING (OFF LINK)
      DOSE = DOSEN(5)
      IF (DOSE.EQ.0.0) GO TO 640
      PRMNDS(5,1) = DOSE/DSTRVL
      PRMNDS(5,2) = DOSE/CAYZER
      PRMNDS(5,5) = DOSE/SPY
      IF (IEXCLU(MODE).EQ.1) GO TO 630
C     NON-EXCLUSIVE USE
      IF (ATERM.GT.10.) GO TO 640
  620 PRMNDS(5,4) = DOSE/PPS
      IF (TI.EQ.0.0) GO TO 640
      PRMNDS(5,3) = DOSE/TI
      GO TO 640
C     EXCLUSIVE USE
  630 IF (.5*ATERM.GT.10.) GO TO 640
      GO TO 620
  640 CONTINUE
      OMFR = 1.-FTZNR
      EE = 2.8E-10*TWOPI*DSTRVL*STERM*1.61
      PRMNDS(5,6) = EE*(RR*POPDR*RVELR-RS*POPDS*RVELS)
```

```
      PRMNDS(5,7) = EE*RR*FTZNR*RVELR
      PRMNDS(5,8) = -EE*RR*FTZNR*POPDR*RVELR2
      PRMNDS(5,9) = EE*OMFR*POPDS*RVELS
      PRMNDS(5,10) = EE*RS*OMFR*RVELS
      PRMNDS(5,11) = -EE*RS*OMFR*POPDS*RVELS2
      PRMNDS(5,12) = EE*FTZNR*POPDR*RVELR
      CALL SENSUM (SENPRM(1,LEG),PRMNDS)
      RETURN
      END
*DECK,SENSUM
      SUBROUTINE SENSUM (SS,XX)
C
C     THIS ROUTINE PERFORMS SUMATION OF THE DERIVATIVES
C     CALCULATED IN SENSTV
C
      DIMENSION SS(38),XX(10,21)
C
C     PARAMETER LIST
C     SS( 1)=DSTRVL
C     SS( 2)=CAYZER
C     SS( 3)=TI
C     SS( 4)=PPS
C     SS( 5)=SPY
C     SS( 6)=FTZNR
C     SS( 7)=POPDR
C     SS( 8)=VELR
C     SS( 9)=RPD
C     SS(10)=FTZNS
C     SS(11)=POPDS
C     SS(12)=VELS
C     SS(13)=RS
C     SS(14)=FTZNU
C     SS(15)=POPDU
C     SS(16)=VELU
C     SS(17)=FCTST
C     SS(18)=RU
C     SS(19)=PDST
C     SS(20)=DTST
C     SS(21)=RST
C     SS(22)=DTSTOR
C     SS(23)=PDSTOR
C     SS(24)=RSTOR
C     SS(25)=ADSTCW
C     SS(26)=CREWNO
C     SS(27)=HANDNO
C     SS(28)=DTHD
C     SS(29)=PPH
C     SS(30)=DIHD
C     SS(31)=PPV
C     SS(32)=FTLFWY
C     SS(33)=TCNTPR
```

```
C     SS(34)=FRSHR
C     SS(35)=TCNTPS
C     SS(36)=TCNTPU
C     SS(37)=FNOATT
C     SS(38)=RR
C
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
C
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
C
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
C
C
C     ACCUMULATE SENSITIVITY DATA FOR EACH VARIABLE
C
C
      GO TO (10,30,40,40,20,20,10,10,10,10), MODE
      CALL GOTOER ('SENSUM-1')
C     TRUCK AND VAN
   10 CONTINUE
      SS(1) = SS(1)+(XX(5,1)+XX(7,6)+XX(2,7)+XX(6,5))*DSTRVL
      SS(2) = SS(2)+(XX(5,2)+XX(8,1)+XX(7,1)+XX(2,1)+XX(4,8)+XX(6,1))*
     *  CAYZER
      SS(3) = SS(3)+(XX(5,3)+XX(8,2)+XX(7,2)+XX(2,2)+XX(4,1)+XX(6,2))*TI
      SS(4) = SS(4)+(XX(5,4)+XX(8,3)+XX(7,3)+XX(2,3)+XX(4,2)+XX(6,3))*
     *  PPS
      SS(5) = SS(5)+(XX(5,5)+XX(8,4)+XX(7,4)+XX(2,5)+XX(4,3)+XX(6,4))*
     *  SPY
      SS(6) = SS(6)+(XX(5,6)+XX(2,8)+XX(6,7))*FTZNR
      SS(7) = SS(7)+XX(5,7)*POPDR
      SS(8) = SS(8)+(XX(5,8)+XX(2,9)+XX(6,10))*VELR
      SS(9) = SS(9)+XX(5,9)*RPD
      SS(10) = SS(10)+(XX(5,10)+XX(2,10)+XX(6,11))*FTZNS
      SS(11) = SS(11)+XX(5,11)*POPDS
      SS(12) = SS(12)+(XX(5,12)+XX(2,11)+XX(6,14))*VELS
      SS(13) = SS(13)+XX(5,13)*RS
      SS(14) = SS(14)+(XX(5,14)+XX(2,12)+XX(6,15))*FTZNU
      SS(15) = SS(15)+XX(5,15)*POPDU
      SS(16) = SS(16)+(XX(5,16)+XX(2,13)+XX(6,18))*VELU
```

```
      SS(17) = SS(17)+(XX(5,17)+XX(6,16))*FCTST
      SS(18) = SS(18)+XX(5,18)*RU
      SS(19) = SS(19)+XX(7,5)*PDST
      SS(20) = SS(20)+XX(7,7)*DTST
      SS(21) = SS(21)+XX(7,8)*RST
      SS(22) = SS(22)+XX(8,5)*DTSTOR
      SS(23) = SS(23)+XX(8,6)*PDSTOR
      SS(24) = SS(24)+XX(8,7)*RSTOR
      SS(25) = SS(25)+XX(2,4)*ADSTCW
      SS(26) = SS(26)+XX(2,6)*CREWNO
      SS(27) = SS(27)+XX(4,4)*HANDNO
      SS(28) = SS(28)+XX(4,5)*DTHD
      SS(29) = SS(29)+XX(4,6)*PPH
      SS(30) = SS(30)+XX(4,7)*DIHD
      SS(31) = SS(31)+XX(6,6)*PPV
      SS(32) = SS(32)+(XX(5,19)+XX(6,8))*FTLFWY
      SS(33) = SS(33)+XX(6,9)*TCNTPR
      SS(34) = SS(34)+(XX(5,20)+XX(6,12))*FRSHR
      SS(35) = SS(35)+XX(6,13)*TCNTPS
      SS(36) = SS(36)+XX(6,17)*TCNTPU
      SS(38) = SS(38)+XX(5,21)*RR
      RETURN
C     AIR
   20 CONTINUE
      SS(1) = SS(1)+(XX(1,4)+XX(2,7)+XX(3,4)+XX(7,6))*DSTRVL
      SS(2) = SS(2)+(XX(2,1)+XX(4,8)+XX(7,1)+XX(8,1))*CAYZER
      SS(3) = SS(3)+(XX(1,1)+XX(2,2)+XX(3,1)+XX(4,1)+XX(7,2)+XX(8,2))*TI
      SS(4) = SS(4)+(XX(1,2)+XX(2,3)+XX(3,2)+XX(4,2)+XX(7,3)+XX(8,3))*
     *  PPS
      SS(5) = SS(5)+(XX(1,3)+XX(2,5)+XX(3,3)+XX(4,3)+XX(7,4)+XX(8,4))*
     *  SPY
      SS(6) = SS(6)+(XX(1,7)+XX(2,8)+XX(3,7))*FTZNR
      SS(8) = SS(8)+(XX(1,6)+XX(2,9)+XX(3,6))*VELR
      SS(10) = SS(10)+XX(2,10)*FTZNS
      SS(12) = SS(12)+XX(2,11)*VELS
      SS(14) = SS(14)+XX(2,12)*FTZNU
      SS(16) = SS(16)+XX(2,13)*VELU
      SS(19) = SS(19)+XX(7,5)*PDST
      SS(20) = SS(20)+XX(7,7)*DTST
      SS(21) = SS(21)+XX(7,8)*RST
      SS(22) = SS(22)+XX(8,5)*DTSTOR
      SS(23) = SS(23)+XX(8,6)*PDSTOR
      SS(24) = SS(24)+XX(8,7)*RSTOR
      SS(25) = SS(25)+XX(2,4)*ADSTCW
      SS(26) = SS(26)+XX(2,6)*CREWNO
      SS(27) = SS(27)+XX(4,4)*HANDNO
      SS(28) = SS(28)+XX(4,5)*DTHD
      SS(29) = SS(29)+XX(4,6)*PPH
      SS(30) = SS(30)+XX(4,7)*DIHD
      SS(31) = SS(31)+XX(1,5)*PPV
      SS(37) = SS(37)+XX(3,5)*FNOATT
```

```
      RETURN
   30 CONTINUE
C     RAIL
      SS(1) = SS(1)+(XX(5,1)+XX(6,5)+XX(7,6))*DSTRVL
      SS(2) = SS(2)+(XX(2,1)+XX(4,8)+XX(5,2)+XX(6,1)+XX(7,1)+XX(8,1))*
     *  CAYZER
      SS(3) = SS(3)+(XX(2,2)+XX(4,1)+XX(5,3)+XX(6,2)+XX(7,2)+XX(8,2))*TI
      SS(4) = SS(4)+(XX(2,3)+XX(4,2)+XX(5,4)+XX(6,3)+XX(7,3)+XX(8,3))*
     *  PPS
      SS(5) = SS(5)+(XX(2,4)+XX(4,3)+XX(5,5)+XX(6,4)+XX(7,4)+XX(8,4))*
     *  SPY
      SS(6) = SS(6)+(XX(5,6)+XX(6,7))*FTZNR
      SS(7) = SS(7)+XX(5,7)*POPDR
      SS(8) = SS(8)+(XX(5,8)+XX(6,9))*VELR
      SS(10) = SS(10)+(XX(5,12)+XX(6,10))*FTZNS
      SS(11) = SS(11)+XX(5,10)*POPDS
      SS(12) = SS(12)+(XX(5,11)+XX(6,12))*VELS
      SS(13) = SS(13)+XX(5,9)*RS
      SS(14) = SS(14)+(XX(5,13)+XX(6,13))*FTZNU
      SS(15) = SS(15)+XX(5,14)*POPDU
      SS(16) = SS(16)+(XX(5,15)+XX(6,15))*VELU
      SS(18) = SS(18)+XX(5,16)*RU
      SS(19) = SS(19)+XX(7,5)*PDST
      SS(20) = SS(20)+XX(7,7)*DTST
      SS(21) = SS(21)+XX(7,8)*RST
      SS(22) = SS(22)+XX(8,5)*DTSTOR
      SS(23) = SS(23)+XX(8,6)*PDSTOR
      SS(24) = SS(24)+XX(8,7)*RSTOR
      SS(27) = SS(27)+XX(4,4)*HANDNO
      SS(28) = SS(28)+XX(4,5)*DTHD
      SS(29) = SS(29)+XX(4,6)*PPH
      SS(30) = SS(30)+XX(4,7)*DIHD
      SS(31) = SS(31)+XX(6,6)*PPV
      SS(33) = SS(33)+XX(6,8)*TCNTPR
      SS(35) = SS(35)+XX(6,11)*TCNTPS
      SS(36) = SS(36)+XX(6,14)*TCNTPU
      SS(38) = SS(38)+XX(5,17)*RR
      RETURN
C     WATER
   40 CONTINUE
      SS(1) = SS(1)+(XX(7,6)+XX(5,1))*DSTRVL
      SS(2) = SS(2)+(XX(2,1)+XX(4,8)+XX(5,2)+XX(7,1)+XX(8,1))*CAYZER
      SS(3) = SS(3)+(XX(2,2)+XX(4,1)+XX(5,3)+XX(7,1)+XX(8,2))*TI
      SS(4) = SS(4)+(XX(2,3)+XX(4,2)+XX(5,4)+XX(7,3)+XX(8,3))*PPS
     SS(5) = SS(5)+(XX(2,4)+XX(4,3)+XX(5,5)+XX(7,4)+XX(8,4))*SPY
      SS(6) = SS(6)+XX(5,6)*FTZNR
      SS(7) = SS(7)+XX(5,7)*POPDR
      SS(8) = SS(8)+XX(5,8)*VELR
      SS(11) = SS(11)+XX(5,10)*POPDS
      SS(12) = SS(12)+XX(5,11)*VELS
      SS(13) = SS(13)+XX(5,9)*RS
```

```
      SS(19) = SS(19)+XX(7,5)*PDST
      SS(20) = SS(20)+XX(7,7)*DTST
      SS(21) = SS(21)+XX(7,8)*RST
      SS(22) = SS(22)+XX(8,5)*DTSTOR
      SS(23) = SS(23)+XX(8,6)*PDSTOR
      SS(24) = SS(24)+XX(8,7)*RSTOR
      SS(27) = SS(27)+XX(4,4)*HANDNO
      SS(28) = SS(28)+XX(4,5)*DTHD
      SS(29) = SS(29)+XX(4,6)*PPH
      SS(30) = SS(30)+XX(4,7)*DIHD
      SS(38) = SS(38)+XX(5,12)*RR
      RETURN
      END
*DECK,SETISO
      SUBROUTINE SETISO
C
C ... SETS ISOTOPE DEPENDENT INFORMATION
C
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
```

```
      PHTE = PHTENG(LIB)
      HLIFE = TABHLF(LIB)
      DEPVEL = VELDEP(LIB)
      LNGTYP = LNGTAB(LIB)
      RDF = TABRDF(LIB)
      CDF = CLDOSF(LIB)
C ... ENTRY DEPENDENT INFORMATION
      CI = CIPKG(ISO)

      IGRP = IPCGRP(ISO)
      IDSP = IDISP(ISO)

C     IACC=1 FOR EXPOSURE SOURCE ACCIDENT
      IACC = 2
      IF (IDSP.EQ.1) IACC = 1
C
      RETURN
      END
*DECK,SETMAT
      SUBROUTINE SETMAT
C
C ... SETS MATERIAL DEPENDENT INFORMATION
C
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
      COMMON /CON/ TOL,NDERV,NA,NC,TWOPI,PI,Q6,WS,GAM,FMU,DZ,VV
      COMMON /TBD/ TBDIHD(2),TBDTHD(2),TBPPH(2)
C
      CAYZER = PKGCOE(MAT)
      DPKG = EFFSIZ(MAT)
      IF (DPKG.LE.PKGSZ2) THEN
```

```
        DIHD = TBDIHD(1)
        DTHD = TBDTHD(1)
        PPH = TBPPH(1)
      ELSE
        DIHD = TBDIHD(2)
        DTHD = TBDTHD(2)
        PPH = TBPPH(2)
      ENDIF

      FG = FRGAMA(MAT)
      FN = FRNEUT(MAT)
      IF (ABS(FG+FN-1.0) .GT. TOL) THEN
        WRITE (4,125) MODE, LABMAT(MAT), FG, FN
        STOP
      ENDIF

C ... MODE AND MATERIAL DEPENDENT INFORMATION
      TI = TIPKG(MODE,MAT)
      PPS = PKGSHP(MODE,MAT)
C
  125 FORMAT (//'  FRACTIONS OF GAMMA AND NEUTRON MUST SUM TO 1.0',/,
     *          '  FOR MATERIAL ',A10,' THE FRACTION OF GAMMA IS',
     *         1PE9.2,/,'  AND THE FRACTION OF NUETRON IS',1PE9.2,/)
      RETURN
      END
*DECK,SETMOD
      SUBROUTINE SETMOD (LEG)
C
C     SETS MODE DEPENDENT INFORMATION
C
      COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *  FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *  PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *  DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *  TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
```

```
     *   ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *   TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *   RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
          REAL LCFPYR
          COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *   LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
          REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
          LOGICAL RSPEC
          COMMON /CON/ TOL,NDERV,NA,NC,TWOPI,PI,Q6,WS,GAM,FMU,DZ,VV

          IF (RSPEC) THEN
            MODE = LMODE(LEG)
          ELSE
            MODE = MODSAV(LEG)
          ENDIF

C ... MODE DEPENDENT INFORMATION
          SPY = TABSPY(MODE)

C ... DNORML VALUES 7 THRU 19 ARE USED FOR ROUTE SPECIFIC AND AGGREGATE
          CREWNO = DNORML(MODE,7)
          ADSTCW = DNORML(MODE,8)
          HANDNO = DNORML(MODE,9)
          DTST = DNORML(MODE,10)
          STPMIN = DNORML(MODE,11)
          TIMZR = DNORML(MODE,12)
          FMINCL = DNORML(MODE,13)
          PDST = DNORML(MODE,14)
          RST = DNORML(MODE,15)
          DTSTOR = DNORML(MODE,16)
          PDSTOR = DNORML(MODE,17)
          RSTOR = DNORML(MODE,18)
          PPV = DNORML(MODE,19)

          IF (.NOT.RSPEC) THEN

            POPDEN(4) = POPDEN(3)*RPD*.1
            POPDR = POPDEN(1)
            POPDS = POPDEN(2)
            POPDU = POPDEN(3)
            POPDUP = POPDEN(4)
            DSTRVL = DISTKM(MODE)
            FTZNR = DNORML(MODE,1)
            FTZNS = DNORML(MODE,2)
            FTZNU = DNORML(MODE,3)
            VELR = DNORML(MODE,4)/3.6
            VELS = DNORML(MODE,5)/3.6
            VELU = DNORML(MODE,6)/3.6
            FRCZON(1) = FTZNR
            FRCZON(2) = FTZNS
            FRCZON(3) = FTZNU
```

```
      FSUM = FTZNR+FTZNS+FTZNU
      IF (ABS(FSUM-1.).LT.TOL) GO TO 30
      IF (MODE.EQ.3.OR.MODE.EQ.4) GO TO 20
      WRITE (4,130) MODE,FTZNR,FTZNS,FTZNU
      STOP
   20 WRITE (4,140) MODE,FSUM
   30 CONTINUE
      FRSHR = DNORML(MODE,20)
      FCTST = DNORML(MODE,21)
      FTLFWY = DNORML(MODE,22)
      TCNTPR = DNORML(MODE,23)
      TCNTPS = DNORML(MODE,24)
      TCNTPU = DNORML(MODE,25)
    ELSE
C . . . ROUTE SPECIFIC
      DO 120 J = 1, 3
        FRCZON(J) = 0.0
        POPDEN(J) = 0.0
  120 CONTINUE
      POPDEN(4) = 0.0
      DSTRVL = LDIST(LEG)
      IF (LZONE(LEG).EQ.1) THEN
        FRCZON(1) = 1.0
        POPDEN(1) = LPOPD(LEG)
        POPDR = LPOPD(LEG)
        POPDS = 0.0
        POPDU = 0.0
        POPDUP = 0.0
        FTZNR = 1.0
        FTZNS = 0.0
        FTZNU = 0.0
        TCNTPR = LVDEN(LEG)
        TCNTPS = 0.0
        TCNTPU = 0.0
        VELR = LSPED(LEG)/3.6
        VELS = 0.0
        VELU = 0.0
      ELSEIF (LZONE(LEG).EQ.2) THEN
        FRCZON(2) = 1.0
        POPDEN(2) = LPOPD(LEG)
        POPDR = 0.0
        POPDS = LPOPD(LEG)
        POPDU = 0.0
        POPDUP = 0.0
        FTZNR = 0.0
        FTZNS = 1.0
        FTZNU = 0.0
        TCNTPR = 0.0
        TCNTPS = LVDEN(LEG)
        TCNTPU = 0.0
        VELR = 0.0
```

```
          VELS = LSPED(LEG)/3.6
          VELU = 0.0
        ELSE
          FRCZON(3) = 1.0
          POPDEN(3) = LPOPD(LEG)
          POPDR = 0.0
          POPDS = 0.0
          POPDU = LPOPD(LEG)
          POPDUP = POPDU*.1*RPD
          POPDEN(4) = POPDUP
          FTZNR = 0.0
          FTZNS = 0.0
          FTZNU = 1.0
          TCNTPR = 0.0
          TCNTPS = 0.0
          TCNTPU = LVDEN(LEG)
          VELR = 0.0
          VELS = 0.0
          VELU = LSPED(LEG)/3.6
        ENDIF
C ...   THESE VARIABLES DO NOT APPLY WHEN ROUTE SPECIFIC
        FRSHR = 0.0
        FCTST = 0.0
        FTLFWY = 0.0
      ENDIF

C ... RECIPRICAL VELOCITY
      DR = 0.0
      DS = 0.0
      DU = 0.0
      IF (VELR.NE.0.) DR = FTZNR/VELR
      IF (VELS.NE.0.) DS = FTZNS/VELS
      IF (VELU.NE.0.) DU = FTZNU/VELU
      DEN = DR+DS+DU
      IF (DEN.EQ.0.0) THEN
        WRITE (4,150)
        STOP
      ENDIF
      VELM = 1./DEN
C
  130 FORMAT (////' ERROR ON DATA INPUT --'/5X,'SUM OF FRACTIONAL ',
     *   'TRAVEL VALUES FOR MODE',I3,' IS NOT EQUAL TO 1.0',/10X,
     *   'RURAL=',1PE10.3,5X,'SUBURBAN=',1PE10.3,5X,'URBAN=',1PE10.3)
  140 FORMAT (////5X,'UNINHABITED ZONE TRAVEL FOR MODE NUMBER',I3,
     *' IS ASSUMED TO BE',1PE10.3)
  150 FORMAT (//,'  THE RECIPROCAL VELOCITY IS ZERO IN SETMOD')
      RETURN
      END
*DECK,SETVAL
      SUBROUTINE SETVAL
C
```

```
C       THIS SUBROUTINE SETS THE ARRAYS DISTKM AND PKGSIZ TO FKMPS AND
PKGCDM
C       VALUES.  THE PACKAGE COEFFICIENT PKGCOE IS ALSO COMPUTED HERE.
C
C       VALUES FOR FKMPS, WHICH IS INDEXED BY ISOTOPE AND MODE, ARE STORED
IN
C       ARRAY DISTKM WHICH IS INDEXED BY MODE ONLY.  IF THE VALUES FOR FKMPS
C       DIFFER WITHIN A MODE AN ERROR MESSAGE IS PRINTED AND EXECUTION IS
C       STOPPED.  VALUES FOR PKGCDM, WHICH IS INDEXED BY THE
PHYSICAL/CHEMICAL
C       GROUP TYPE OF AN ISOTOPE, ARE STORED IN PKGSIZ WHICH IS INDEXED BY
THE
C       MATERIAL.  IF PKGCDM DIFFERES WITHIN A MATERIAL, AN ERROR MESSAGE IS
C       PRINTED AND EXECUTION IS STOPPED.
C
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
     *  PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *  IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *  RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
      LOGICAL DIST,PCKG
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /LABELS/ LDESCR(38),LABISO(65),
     *  LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *  WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
      CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *  LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *  LBTRSH*10,LABGRP*10,LABMAT*10
      COMMON /LVAR/ RSPEC,LMODE(40),LDIST(40),LPOPD(40),LSPED(40),
     *  LVDEN(40),LARAT(40),LZONE(40),LTYPE(40),NLNK
      REAL LDIST,LPOPD,LSPED,LVDEN,LARAT
      LOGICAL RSPEC

      IF ((.NOT.RSPEC).AND.(.NOT.DIST)) THEN
C . . . IF NOT ROUTE-SPECIFIC AND IF DISTKM WAS NOT ENTERED UNDER DISTKM
        DO 10 J = 1, NMODE
          MODE = MODSAV(J)
          DO 10 M = 1, NMAT(MODE)
```

```
            MAT = MATSAV(MODE,M)
            DO 10 I = 1, NIS(MAT)
              ISO = ISOSAV(MAT,I)
              LIB = LIBSAV(ISO)
C   .   .   .   SET DISTKM TO THE VALUE OF FKMPS FOR THE FIRST ISOTOPE
C   .   .   .   IN THE FIRST MATERIAL IN THE MODE
              IF ((M.EQ.1).AND.(I.EQ.1)) DISTKM(MODE) = FKMPS(LIB,MODE)
              IF (DISTKM(MODE) .NE. FKMPS(LIB,MODE)) THEN
                WRITE (4,100) LABISO(LIB), MODE
                STOP
              ENDIF
   10   CONTINUE
      ENDIF

      IF (.NOT.PCKG) THEN
        DO 20 MAT = 1, MCOUNT
          DO 20 I = 1, NIS(MAT)
            ISO = ISOSAV(MAT,I)
            LIB = LIBSAV(ISO)
C   .   .   .   SET PKGSIZ TO THE VALUE OF PKGCDM FOR THE FIRST ISOTOPE IN
MATERIAL
            IF (I.EQ.1) PKGSIZ(MAT) = PKGCDM(IPCGRP(ISO))
            IF (PKGSIZ(MAT) .NE. PKGCDM(IPCGRP(ISO))) THEN
              WRITE (4,200) LABISO(LIB), LABMAT(MAT)
              STOP
            ENDIF
   20   CONTINUE
      ENDIF

C . . CHECK PACKAGE DEMINSION AND COMPUTE PACKAGE COEFFICIENT (KO)
      DO 30 MAT = 1, MCOUNT
        IF (PKGSIZ(MAT).GE.4.) THEN
          XX = 1.0+0.5*PKGSIZ(MAT)
          EFFSIZ(MAT) = 2.*XX**0.75-.55
        ELSE
          EFFSIZ(MAT) = PKGSIZ(MAT)
        ENDIF
        PKGCOE(MAT) = (1.0+0.5*EFFSIZ(MAT))**2
   30 CONTINUE

  100 FORMAT (/,'  ISOTOPE ',A10,' WAS ASSIGNED A DISTANCE FOR MODE',
     *          I3,' THAT IS',/,'  DIFFERENT FROM DISTANCES ASSIGNED ',
     *          'TO OTHER ISOTOPES IN THAT MODE',/)
  200 FORMAT (/,'  ISOTOPE ',A10,' WAS ASSIGNED A PHYSICAL/CHEMICAL ',
     *          'GROUP TYPE WITH A',/,'  PACKAGE SIZE THAT IS ',
     *          'DIFFERENT FROM OTHER ISOTOPES IN MATERIAL ',A10,/)
      RETURN
      END
*DECK,SHIELD
      SUBROUTINE SHIELD (IUOPT, RR, RS, RU)
C ... USING SHIELDING OPTION FLAG IUOPT, THE RURAL SHIELDING (RR), THE
```

```
C     SUBURBAN SHIELDING (RS), AND THE URBAN SHIELDING (RU) ARE OVERRIDDEN.
C     IF IUOPT IS 1 THE PEOPLE INSIDE BUILDINGS ARE COMPLETELY SHIELDED AND
C     GET NO DOSE.  IF IUOPT IS 2 RR, RS, AND RU ARE NOT CHANGED.  AND IF
C     IUOPT IS 3 THERE IS NO SHIELDING BENEFIT FROM THE BUILDINGS.

      IF (IUOPT .EQ. 1) THEN
          RR = 0
          RS = 0
          RU = 0
      ELSE IF (IUOPT .EQ. 3) THEN
          RR = 1
          RS = 1
          RU = 1
      ENDIF
      RETURN
      END
*DECK,SHIFTC
      SUBROUTINE SHIFTC (C,A)
      CHARACTER*10 A,B,C
      A = C
      DO 10 J = 1, 10
        IF (A(10:10).EQ.' ') B = ' '//A(1:9)
        A = B
   10 CONTINUE
      RETURN
      END
*DECK,SHIFTL
      SUBROUTINE SHIFTL (C,A)
      CHARACTER*10 A,B,C
      A = C
      DO 10 J = 1, 10
        IF (A(1:1).EQ.' ') B = A(2:10)//' '
        A = B
   10 CONTINUE
      RETURN
      END
*DECK,TRANS
      FUNCTION TRANS (IRTYPE,D)
      COMMON /NORML/ DNORML(10,25),TIPKG(10,12),PKGCOE(12),DOSEN(11),
     *  DOSMAX(40),INCFRE(40,12,11),PKGCDM(15),PKGSIZ(12),MODSAV(10),
     *  SENPRM(38,40),FNOATT,PKGSZ1,PKGSZ2,DPKG,NDOSE,NDOSP1,NDLAB,NLEG,
     *  CH1,CH2,ITRAIN,COEF(2,5),FRGAMA(12),FRNEUT(12),EFFSIZ(12),KI(38)
      REAL INCFRE
      TRANS = 1.
      DO 10 J = 2, 5
        TRANS = TRANS+COEF(IRTYPE,J)*D**(J-1)
   10 CONTINUE
      TRANS = TRANS*EXP(-COEF(IRTYPE,1)*D)
      RETURN
      END
*DECK,VALUE
```

```
      FUNCTION VALUE (DMY)
C
C     THIS FUNCTION LOOKS AT THE NEXT FIELD ON THE CARD.
C     IF THE FIELD IS A NUMERIC FIELD, THE NUMBER IS RETURNED IN "VALUE"
C     IF THE FIELD IS NOT NUMERIC, A CALL TO "ABORT" IS MADE.
C
      COMMON /CARD/ IPOINT,KBYTE,CONTUF,NRFLAG,NCS
      LOGICAL CONTUF,NRFLAG
      COMMON /CCARD/ ICARD,TEST,ICARDI,NVY(10)
      CHARACTER*10 ICARD*80, TEST,ICARDI,NVY*1
C
      CALL NEXT
      CALL CONVRT (WSA)
      VALUE = WSA
      RETURN
      END
*DECK,WTRNK
      SUBROUTINE WTRNK (LEG,IPASQ,COST,EOF)
C
C     THIS ROUTINE WRITES CONSEQUENCE DATA TO FILE6 FOR USE IN PLOTTING.
C
C     LEG   = LINK NUMBER IF ROUTE-SPECIFIC OR ITH MODE IF NOT ROUTE-
SPECIFIC
C     IPASQ = PASQUILL CLOUD (1 TO 6) -- SHOULD BE 1 IF PASQUILL NOT USED
C     COST  = SET TRUE IF PRINTING ECONOMIC EMPACT DATA, OTHERWISE FALSE
C     EOF   = SET TRUE IF A RUN IS COMPLETE, OTHERWISE FALSE
C
      COMMON /ACD/ RADIST(15,3),DOSDST(15,20,3),CIPKG(65),DELTIM(3),
     *  EXPEO(29,4),RFRAC(15,20),SUMEF(20,3),SUMEM(20,3),ASIZE(30),
     *  ACCLCF(20,3),ACCGE(20,3),SEVFRC(10,20,3),FRCZON(3),
     *  ARATMZ(10,3),ACCRAT(20,3),ACCPYR(20,3),EFPYR(20,3),EMPYR(20,3),
     *  LCFPYR(20,3),GEPYR(20,3),ECNMPT(20,3),PHTENG(65),
     *  IDISP(65),EMRCST(20),ONSCST(4),CULVL,BDF,XFARM,WBLCF,GNDSGE
     *  ,IACC,NTHRS,NAREAS,DFLEV(30),AREADA(30),IPCGRP(65),ECONF,
     *  TABRDF(65),TABHLF(65),RESP(11,20),AERSOL(11,20),LNGTAB(65),
     *  RPCVAL(65,8),DNHLCF(20,3),DNHGE(20,3),RDFLCF(20,3),RDFGE(20,3)
      REAL LCFPYR
      COMMON /ACD2/ GRDLCF(20,3),GRDGE(20,3),AP1T1(20),
     *  A1T20(20),A20T40(20),AGT40(20),
     *  ECNPYR(20,3),EMTRSH(5),EXPLCF(5),CHIVAL(30,65),
     *  BRATE,VELDEP(65),IRNKC,NPOP1,IPSQSB,PSAREA(18),PSQLCT(18,6),
     *  PSPROB(6),CLDLCF(20,3),CLDGE(20,3),CLDOSF(65),
     *  TS1(4),TS2(4),TS3(3),TS4,TBAR(4),EVAL1(3),EVAL2(3),EVAL3(3)
     *  ,FDTRAN(65),SLTRAN(65),INGVAL(65,8),
     *  EARLYL(30,20,3),EARLYD(30,20,5),TDSDST(15,20,3),TDCONF(30,20),
     *  TCLEVL(30,20),DECON(30,20,65),NAVINT(65),
     *  DTOT(20,3),PDTOT(20,3),PSUMEF(20,3),PSUMEM(20,3),
     *  INGLCF(20),INGGE(20),PECMPT(20,3),URSKF
      REAL INGVAL,INGLCF,INGGE
      LOGICAL URSKF
      COMMON /MAIN/ TABSPY(10),FKMPS(65,10),ISOSAV(12,65),MATSAV(10,12),
```

```
     *   PKGSHP(10,12),NMAT(10),NIS(12),NSEV,NPOP,DISTKM(10),MODE,NORG,
     *   IUOPT,NRAD,IANA,IPRAM,ISEN,IEXCLU(10),MAT,ISO,LIB,NMODE,MCOUNT,
     *   RPD,RR,RS,RU,LISTLH,POPDEN(4),NGROUP,ICOUNT,LIBSAV(65),DIST,PCKG
          LOGICAL DIST,PCKG
          COMMON /SET/ VELR,VELS,VELU,VELM,TCNTPR,TCNTPS,TCNTPU,FTZNR,FTZNS,
     *   FTZNU,FRSHR,FCTST,POPDR,POPDS,POPDU,PPV,FTLFWY,DSTRVL,CAYZER,TI,
     *   PPS,SPY,ADSTCW,CREWNO,HANDNO,DTSTOR,PDSTOR,RSTOR,DIHD,DTHD,PPH,
     *   DTST,PDST,RST,CI,PHTE,LNGTYP,RDF,HLIFE,POPDUP,STPMIN,CDF,DEPVEL,
     *   TIMZR,FMINCL,IGRP,IDSP,FG,FN,U,A(4)
          COMMON /LABELS/ LDESCR(38),LABISO(65),
     *   LABDOS(25,2),LABORG(8),LABPOP(3),LABMOD(10),LABEF(4),
     *   WORDS(6),LBTRSH(5),LABGRP(15),LABMAT(12)
          CHARACTER LDESCR*40,LABISO*10,LABDOS*30,
     *   LABORG*10,LABPOP*10,LABMOD*10,LABEF*10,WORDS*1,
     *   LBTRSH*10,LABGRP*10,LABMAT*10

          LOGICAL COST, EOF
          INTEGER LEG0, IPASQ0
          SAVE LEG0, IPASQ0
          DATA LEG0 / 0 /, IPASQ0 / 0 /

          IF (.NOT.EOF) THEN

            IF (LEG0 .EQ. 0) WRITE (6,'(I5)') NSEV

            IF (LEG.NE.LEG0) THEN
C .....   MODE AND/OR LINK DEPENDENT INFORMATION
              LEG0 = LEG
              IPASQ0 = 0
              WRITE (6,'(''       LEG'')')
              WRITE (6,10) LEG, LABMOD(MODE), MODE
              WRITE (6,'(''  ACCIDENT RATE'')')
              WRITE (6,60) ((ACCRAT(J,K),K=1,NPOP),J=1,NSEV)
              WRITE (6,'(''  EXPECTED NUMBER OF ACCIDENTS (PROBABILITY)'')')
              WRITE (6,60) ((ACCPYR(J,K),K=1,NPOP),J=1,NSEV)
            ENDIF

            IF (IPASQ.NE.IPASQ0) THEN
C .....   PASQUILL PROBABILTIY
              IPASQ0 = IPASQ
              WRITE (6,'(''  PASQUILL CATEGORY AND PROBABILITY'')')
              WRITE (6,20) IPASQ, PSPROB(IPASQ)
            ENDIF

            IF (.NOT.COST) THEN
C .....   ISOTOPE DEPENDENT INFORMATION
              WRITE (6,'(''   ISOTOPE'')')
              WRITE (6,80) ISO, IDISP(ISO), CI*PPS, HLIFE
              WRITE (6,70) LABISO(LIB), LABGRP(IPCGRP(ISO)), LABMAT(MAT)
              WRITE (6,'(''       LCF (NON-UNIT) OR PERSON-REM (UNIT) '',
     *                                    ''CONSEQUENCE DATA'')')
```

```
          WRITE (6,60) ((ACCLCF(J,K),K=1,NPOP),J=1,NSEV)
          WRITE (6,'(''         GE CONSEQUENCE DATA'')')
          WRITE (6,60) ((ACCGE(J,K),K=1,NPOP),J=1,NSEV)
        ELSE
          WRITE (6,'(''  ECONOMIC IMPACT DATA'')')
          WRITE (6,60) ((ECNMPT(J,K),K=1,NPOP),J=1,NSEV)
        ENDIF
      ELSE
C ...   END OF A RUN
        WRITE (6, '(''        EOF'')')
        LEGO = 0
        IPASQO = 0
      ENDIF
      RETURN
C
   10 FORMAT (1X,I5,1X,A10,1X,I5)
   20 FORMAT (1X,I5,1PE10.2)
   60 FORMAT (8(1PE10.2))
   70 FORMAT (1X,A10,1X,A10,1X,A10)
   80 FORMAT (1X,2I5, 2(1PE10.2))
      END
```

```
C
C   THE FOLLOWING SUBROUTINES ARE USED IN RADTRAN AND ARE TAKEN FROM
SANDIA'S
C   SLATEC MATH LIBRARY.  SUBROUTINE XERR IS BEING USED INSTEAD OF THE
SLATEC
C   SUBROUTINE XERROR.
C
*DECK,AVINT
      SUBROUTINE AVINT (X,Y,N,XLO,XUP,ANS,IERR)
C     ***BEGIN PROLOGUE  AVINT
C     ***DATE WRITTEN   690901   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  D1B
C     ***KEYWORDS  INTEGRATION,QUADRATURE,TABULATED DATA
C     ***AUTHOR  JONES, R. E., (SLA)
C     ***PURPOSE  INTEGRATE A FUNCTION TABULATED AT ARBITRARILY SPACED
C     ABSCISSAS USING OVERLAPPING PARABOLAS
C     ***DESCRIPTION
C     ABSTRACT
C     AVINT INTEGRATES A FUNCTION TABULATED AT ARBITRARILY SPACED
C     ABSCISSAS.  THE LIMITS OF INTEGRATION NEED NOT COINCIDE
C     WITH THE TABULATED ABSCISSAS.
C
C     A METHOD OF OVERLAPPING PARABOLAS FITTED TO THE DATA IS USED
C     PROVIDED THAT THERE ARE AT LEAST 3 ABSCISSAS BETWEEN THE
C     LIMITS OF INTEGRATION.  AVINT ALSO HANDLES TWO SPECIAL CASES.
C     IF THE LIMITS OF INTEGRATION ARE EQUAL, AVINT RETURNS A RESULT
C     OF ZERO REGARDLESS OF THE NUMBER OF TABULATED VALUES.
C     IF THERE ARE ONLY TWO FUNCTION VALUES, AVINT USES THE
C     TRAPEZOID RULE.
C
C     DESCRIPTION OF PARAMETERS
C     THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
C     X(N), Y(N).
C
C     INPUT--
C     X    - REAL ARRAY OF ABSCISSAS, WHICH MUST BE IN INCREASING
C     ORDER.
C     Y    - REAL ARRAY OF FUNCTIONAL VALUES. I.E., Y(I)=FUNC(X(I)).
C     N    - THE INTEGER NUMBER OF FUNCTION VALUES SUPPLIED.
C     N .GE. 2 UNLESS XLO = XUP.
C     XLO  - REAL LOWER LIMIT OF INTEGRATION.
C     XUP  - REAL UPPER LIMIT OF INTEGRATION.
C     MUST HAVE XLO .LE. XUP.
C
C     OUTPUT--
C     ANS  - COMPUTED APPROXIMATE VALUE OF INTEGRAL
C     IERR - A STATUS CODE
C     --NORMAL CODE
C     =1 MEANS THE REQUESTED INTEGRATION WAS PERFORMED.
C     --ABNORMAL CODES
C     =2 MEANS XUP WAS LESS THAN XLO.
C     =3 MEANS THE NUMBER OF X(I) BETWEEN XLO AND XUP
```

```
C     (INCLUSIVE) WAS LESS THAN 3 AND NEITHER OF THE TWO
C     SPECIAL CASES DESCRIBED IN THE ABSTRACT OCCURRED.
C     NO INTEGRATION WAS PERFORMED.
C     =4 MEANS THE RESTRICTION X(I+1) .GT. X(I) WAS VIOLATED.
C     =5 MEANS THE NUMBER N OF FUNCTION VALUES WAS .LT. 2.
C     ANS IS SET TO ZERO IF IERR=2,3,4,OR 5.
C
C     AVINT IS DOCUMENTED COMPLETELY IN SC-M-69-335
C     ORIGINAL PROGRAM FROM "NUMERICAL INTEGRATION" BY DAVIS &
C     RABINOWITZ.
C     ADAPTATION AND MODIFICATIONS FOR SANDIA MATHEMATICAL PROGRAM
C     LIBRARY BY RONDALL E. JONES.
C     ***REFERENCES  JONES R.E., *APPROXIMATE INTEGRATOR OF FUNCTIONS
C     TABULATED AT ARBITRARILY SPACED ABSCISSAS*,
C     SC-M-69-335, SANDIA LABORATORIES, 1969.
C     ***ROUTINES CALLED  XERROR
C     ***END PROLOGUE  AVINT
C
      DOUBLE PRECISION R3,RP5,SUM,SYL,SYL2,SYL3,SYU,SYU2,SYU3,X1,X2,X3,
     *  X12,X13,X23,TERM1,TERM2,TERM3,A,B,C,CA,CB,CC
      DIMENSION X(N),Y(N)
C     ***FIRST EXECUTABLE STATEMENT  AVINT
      IERR = 1
      ANS = 0.0
      IF (XLO-XUP) 10 , 130 , 140
   10 IF (N.LT.2) GO TO 170
      DO 20 I = 2, N
        IF (X(I).LE.X(I-1)) GO TO 160
        IF (X(I).GT.XUP) GO TO 30
   20 CONTINUE
   30 CONTINUE
      IF (N.GE.3) GO TO 40
C
C     SPECIAL N=2 CASE
      SLOPE = (Y(2)-Y(1))/(X(2)-X(1))
      FL = Y(1)+SLOPE*(XLO-X(1))
      FR = Y(2)+SLOPE*(XUP-X(2))
      ANS = 0.5*(FL+FR)*(XUP-XLO)
      RETURN
   40 CONTINUE
      IF (X(N-2).LT.XLO) GO TO 150
      IF (X(3).GT.XUP) GO TO 150
      I = 1
   50 IF (X(I).GE.XLO) GO TO 60
      I = I+1
      GO TO 50
   60 INLFT = I
      I = N
   70 IF (X(I).LE.XUP) GO TO 80
      I = I-1
      GO TO 70
```

```
   80 INRT = I
      IF ((INRT-INLFT).LT.2) GO TO 150
      ISTART = INLFT
      IF (INLFT.EQ.1) ISTART = 2
      ISTOP = INRT
      IF (INRT.EQ.N) ISTOP = N-1
C
      R3 = 3.0D0
      RP5 = 0.5D0
      SUM = 0.0
      SYL = XLO
      SYL2 = SYL*SYL
      SYL3 = SYL2*SYL
C
      DO 120 I = ISTART, ISTOP
        X1 = X(I-1)
        X2 = X(I)
        X3 = X(I+1)
        X12 = X1-X2
        X13 = X1-X3
        X23 = X2-X3
        TERM1 = DBLE(Y(I-1))/(X12*X13)
        TERM2 = -DBLE(Y(I))/(X12*X23)
        TERM3 = DBLE(Y(I+1))/(X13*X23)
        A = TERM1+TERM2+TERM3
        B = -(X2+X3)*TERM1-(X1+X3)*TERM2-(X1+X2)*TERM3
        C = X2*X3*TERM1+X1*X3*TERM2+X1*X2*TERM3
        IF (I-ISTART) 90 , 90 , 100
   90   CA = A
        CB = B
        CC = C
        GO TO 110
  100   CA = 0.5*(A+CA)
        CB = 0.5*(B+CB)
        CC = 0.5*(C+CC)
  110   SYU = X2
        SYU2 = SYU*SYU
        SYU3 = SYU2*SYU
        SUM = SUM+CA*(SYU3-SYL3)/R3+CB*RP5*(SYU2-SYL2)+CC*(SYU-SYL)
        CA = A
        CB = B
        CC = C
        SYL = SYU
        SYL2 = SYU2
        SYL3 = SYU3
  120 CONTINUE
      SYU = XUP
      ANS = SUM+CA*(SYU**3-SYL3)/R3+CB*RP5*(SYU**2-SYL2)+CC*(SYU-SYL)
  130 RETURN
  140 IERR = 2
      CALL XERR ('AVINT- THE UPPER LIMIT OF INTEGRATION WAS NOT GREATER
     *THAN THE LOWER LIMIT.')
      RETURN
```

```
  150 IERR = 3
      CALL XERR ('AVINT- THERE WERE LESS THAN THREE FUNCTION VALUES BETW
     *EEN THE LIMITS OF INTEGRATION.')
      RETURN
  160 IERR = 4
      CALL XERR ('AVINT- THE ABSCISSAS WERE NOT STRICTLY INCREASING.  MU
     *ST HAVE X(I-1) .LT. X(I) FOR ALL I.')
      RETURN
  170 IERR = 5
      CALL XERR ('AVINT- LESS THAN TWO FUNCTION VALUES WERE SUPPLIED.')
      RETURN
      END
*DECK,BDIFF
      SUBROUTINE BDIFF (L,V)
C     ***BEGIN PROLOGUE  BDIFF
C     ***REFER TO  BSKIN
C     ***ROUTINES CALLED  (NONE)
C     ***DESCRIPTION
C
C     BDIFF COMPUTES THE SUM OF B(L,K)*V(K)*(-1)**K WHERE B(L,K)
C     ARE THE BINOMIAL COEFFICIENTS.  TRUNCATED SUMS ARE COMPUTED BY
C     SETTING LAST PART OF THE V VECTOR TO ZERO. ON RETURN, THE BINOMIAL
C     SUM IS IN V(L).
C     ***END PROLOGUE  BDIFF
      INTEGER I,J,K,L
      REAL V
      DIMENSION V(*)
C     ***FIRST EXECUTABLE STATEMENT  BDIFF
      IF (L.EQ.1) RETURN
      DO 20 J = 2, L
        K = L
        DO 10 I = J, L
          V(K) = V(K-1)-V(K)
          K = K-1
   10   CONTINUE
   20 CONTINUE
      RETURN
      END
*DECK,BESI0
      FUNCTION BESI0 (X)
C     ***BEGIN PROLOGUE  BESI0
C     ***DATE WRITTEN   770401   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  C10B1
C     ***KEYWORDS  BESSEL FUNCTION,FIRST KIND,HYPERBOLIC BESSEL FUNCTION
C     MODIFIED BESSEL FUNCTION,ORDER ZERO
C     ***AUTHOR  FULLERTON, W., (LANL)
C     ***PURPOSE  COMPUTES THE HYPERBOLIC BESSEL FUNCTION OF THE FIRST K
C     OF ORDER ZERO
C     ***DESCRIPTION
C
C     BESI0(X) COMPUTES THE MODIFIED (HYPERBOLIC) BESSEL FUNCTION
C     OF THE FIRST KIND OF ORDER ZERO AND REAL ARGUMENT X.
C
```

```
C     SERIES FOR BI0        ON THE INTERVAL  0.          TO  9.00000D+00
C     WITH WEIGHTED ERROR   2.46E-18
C     LOG WEIGHTED ERROR  17.61
C     SIGNIFICANT FIGURES REQUIRED  17.90
C     DECIMAL PLACES REQUIRED  18.15
C     ***REFERENCES  (NONE)
C     ***ROUTINES CALLED  BESI0E,CSEVL,INITS,R1MACH,XERROR
C     ***END PROLOGUE  BESI0
      DIMENSION BI0CS(12)
      DATA BI0CS(1) / -.07660547252839144951E0 /
      DATA BI0CS(2) / 1.92733795399380827OE0 /
      DATA BI0CS(3) / .2282644586920301339E0 /
      DATA BI0CS(4) / .013048914667072904 28E0 /
      DATA BI0CS(5) / .0004344270900816 4874E0 /
      DATA BI0CS(6) / .0000094226576860 0193E0 /
      DATA BI0CS(7) / .0000001434006289 5106E0 /
      DATA BI0CS(8) / .0000000016138490 6966E0 /
      DATA BI0CS(9) / .0000000000139665 0044E0 /
      DATA BI0CS(10) / .0000000000000957 9451E0 /
      DATA BI0CS(11) / .0000000000000005 3339E0 /
      DATA BI0CS(12) / .0000000000000000 245E0 /
      DATA NTI0,XSML,XMAX / 0,0.,0. /
C     ***FIRST EXECUTABLE STATEMENT  BESI0
      IF (NTI0.NE.0) GO TO 10
      NTI0 = INITS(BI0CS,12,0.1*R1MACH(3))
      XSML = SQRT(4.0*R1MACH(3))
      XMAX = ALOG(R1MACH(2))
C
   10 Y = ABS(X)
      IF (Y.GT.3.0) GO TO 20
C
      BESI0 = 1.0
      IF (Y.GT.XSML) BESI0 = 2.75+CSEVL(Y*Y/4.5-1.0,BI0CS,NTI0)
      RETURN
C
   20 IF (Y.GT.XMAX) CALL XERR ('BESI0   ABS(X) SO BIG I0 OVERFLOWS')
C
      BESI0 = EXP(Y)*BESI0E(X)
C
      RETURN
      END
*DECK,BESI0E
      FUNCTION BESI0E (X)
C     ***BEGIN PROLOGUE  BESI0E
C     ***DATE WRITTEN   770701   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  C10B1
C     ***KEYWORDS  BESSEL FUNCTION,EXPONENTIALLY SCALED,FIRST KIND,
C     HYPERBOLIC BESSEL FUNCTION,MODIFIED BESSEL FUNCTION,
C     ORDER ZERO
C     ***AUTHOR  FULLERTON, W., (LANL)
C     ***PURPOSE  COMPUTES THE EXPONENTIALLY SCALED HYPERBOLIC BESSEL
C     FUNCTION OF THE FIRST KIND OF ORDER ZERO.
C     ***DESCRIPTION
```

```
C
C     BESI0E(X) CALCULATES THE EXPONENTIALLY SCALED MODIFIED (HYPERBOLIC
C     BESSEL FUNCTION OF THE FIRST KIND OF ORDER ZERO FOR REAL ARGUMENT
C     I.E., EXP(-ABS(X))*I0(X).
C
C
C     SERIES FOR BI0        ON THE INTERVAL  0.          TO  9.00000D+00
C     WITH WEIGHTED ERROR   2.46E-18
C     LOG WEIGHTED ERROR  17.61
C     SIGNIFICANT FIGURES REQUIRED  17.90
C     DECIMAL PLACES REQUIRED  18.15
C
C
C     SERIES FOR AI0        ON THE INTERVAL  1.25000D-01 TO  3.33333D-01
C     WITH WEIGHTED ERROR   7.87E-17
C     LOG WEIGHTED ERROR  16.10
C     SIGNIFICANT FIGURES REQUIRED  14.69
C     DECIMAL PLACES REQUIRED  16.76
C
C
C     SERIES FOR AI02       ON THE INTERVAL  0.          TO  1.25000D-01
C     WITH WEIGHTED ERROR   3.79E-17
C     LOG WEIGHTED ERROR  16.42
C     SIGNIFICANT FIGURES REQUIRED  14.86
C     DECIMAL PLACES REQUIRED  17.09
C     ***REFERENCES  (NONE)
C     ***ROUTINES CALLED  CSEVL,INITS,R1MACH
C     ***END PROLOGUE  BESI0E
      DIMENSION BI0CS(12),AI0CS(21),AI02CS(22)
      DATA BI0CS(1) / -.07660547252839144951E0 /
      DATA BI0CS(2) / 1.92733795399380827 0E0 /
      DATA BI0CS(3) / .22826445869203013 39E0 /
      DATA BI0CS(4) / .01304891466707290428E0 /
      DATA BI0CS(5) / .00043442709008164874E0 /
      DATA BI0CS(6) / .00009422657686001 93E0 /
      DATA BI0CS(7) / .00000143400628951 06E0 /
      DATA BI0CS(8) / .00000001613849069 66E0 /
      DATA BI0CS(9) / .00000000013966500 44E0 /
      DATA BI0CS(10) / .00000000000095794 51E0 /
      DATA BI0CS(11) / .00000000000000533 39E0 /
      DATA BI0CS(12) / .00000000000000002 45E0 /
      DATA AI0CS(1) / .07575994494023796E0 /
      DATA AI0CS(2) / .00759138081082334E0 /
      DATA AI0CS(3) / .00041531313389237E0 /
      DATA AI0CS(4) / .00010700764634 39E0 /
      DATA AI0CS(5) / -.00000790117997921E0 /
      DATA AI0CS(6) / -.00000078261435014E0 /
      DATA AI0CS(7) / .00000027838499429E0 /
      DATA AI0CS(8) / .00000008252472 60E0 /
      DATA AI0CS(9) / -.00000001204463945E0 /
      DATA AI0CS(10) / .00000000155964859E0 /
      DATA AI0CS(11) / .00000000022925563E0 /
      DATA AI0CS(12) / -.00000000011916228E0 /
```

```
      DATA AI0CS(13) / .00000000001757854E0 /
      DATA AI0CS(14) / .00000000000112822E0 /
      DATA AI0CS(15) / -.00000000000114684E0 /
      DATA AI0CS(16) / .00000000000027155E0 /
      DATA AI0CS(17) / -.00000000000002415E0 /
      DATA AI0CS(18) / -.00000000000000608E0 /
      DATA AI0CS(19) / .00000000000000314E0 /
      DATA AI0CS(20) / -.00000000000000071E0 /
      DATA AI0CS(21) / .00000000000000007E0 /
      DATA AI02CS(1) / .05449041101410882E0 /
      DATA AI02CS(2) / .00336911647825569E0 /
      DATA AI02CS(3) / .00006889758346918E0 /
      DATA AI02CS(4) / .00000289137052082E0 /
      DATA AI02CS(5) / .00000020489185893E0 /
      DATA AI02CS(6) / .00000002266668991E0 /
      DATA AI02CS(7) / .00000000339623203E0 /
      DATA AI02CS(8) / .00000000049406022E0 /
      DATA AI02CS(9) / .00000000001188914E0 /
      DATA AI02CS(10) / -.00000000003149915E0 /
      DATA AI02CS(11) / -.00000000001321580E0 /
      DATA AI02CS(12) / -.00000000000179419E0 /
      DATA AI02CS(13) / .00000000000071801E0 /
      DATA AI02CS(14) / .00000000000038529E0 /
      DATA AI02CS(15) / .00000000000001539E0 /
      DATA AI02CS(16) / -.00000000000004151E0 /
      DATA AI02CS(17) / -.00000000000000954E0 /
      DATA AI02CS(18) / .00000000000000382E0 /
      DATA AI02CS(19) / .00000000000000176E0 /
      DATA AI02CS(20) / -.00000000000000034E0 /
      DATA AI02CS(21) / -.00000000000000027E0 /
      DATA AI02CS(22) / .00000000000000003E0 /
      DATA NTI0,NTAI0,NTAI02,XSML / 3*0,0. /
C     ***FIRST EXECUTABLE STATEMENT  BESI0E
      IF (NTI0.NE.0) GO TO 10
      NTI0 = INITS(BI0CS,12,0.1*R1MACH(3))
      NTAI0 = INITS(AI0CS,21,0.1*R1MACH(3))
      NTAI02 = INITS(AI02CS,22,0.1*R1MACH(3))
      XSML = SQRT(4.0*R1MACH(3))
C
   10 Y = ABS(X)
      IF (Y.GT.3.0) GO TO 20
C
      BESI0E = 1.0
      IF (Y.GT.XSML) BESI0E = EXP(-Y)*(2.75+CSEVL(Y*Y/4.5-1.0,BI0CS,NTI0
     *  ))
      RETURN
C
   20 IF (Y.LE.8.) BESI0E = (.375+CSEVL((48./Y-11.)/5.,AI0CS,NTAI0))/
     *  SQRT(Y)
      IF (Y.GT.8.) BESI0E = (.375+CSEVL(16./Y-1.,AI02CS,NTAI02))/SQRT(Y)
C
      RETURN
      END
```

```
*DECK,BESI1
      FUNCTION BESI1 (X)
C     ***BEGIN PROLOGUE  BESI1
C     ***DATE WRITTEN   770401   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  C10B1
C     ***KEYWORDS  BESSEL FUNCTION,FIRST KIND,HYPERBOLIC BESSEL FUNCTION
C     MODIFIED BESSEL FUNCTION,ORDER ONE
C     ***AUTHOR  FULLERTON, W., (LANL)
C     ***PURPOSE  COMPUTES THE HYPERBOLIC BESSEL FUNCTION OF FIRST KIND
C     ORDER ONE
C     ***DESCRIPTION
C
C     BESI1(X) CALCULATES THE MODIFIED (HYPERBOLIC) BESSEL FUNCTION
C     OF THE FIRST KIND OF ORDER ONE FOR REAL ARGUMENT X.
C
C     SERIES FOR BI1        ON THE INTERVAL  0.          TO  9.00000D+00
C     WITH WEIGHTED ERROR   2.40E-17
C     LOG WEIGHTED ERROR  16.62
C     SIGNIFICANT FIGURES REQUIRED  16.23
C     DECIMAL PLACES REQUIRED  17.14
C     ***REFERENCES  (NONE)
C     ***ROUTINES CALLED  BESI1E,CSEVL,INITS,R1MACH,XERROR
C     ***END PROLOGUE  BESI1
      DIMENSION BI1CS(11)
      DATA BI1CS(1) / -.0019717132610998 59E0 /
      DATA BI1CS(2) / .407348876675464 81E0 /
      DATA BI1CS(3) / .034838994299959456E0 /
      DATA BI1CS(4) / .001545394556300123E0 /
      DATA BI1CS(5) / .000041888521098377E0 /
      DATA BI1CS(6) / .000007649026764 83E0 /
      DATA BI1CS(7) / .000000100424939 24E0 /
      DATA BI1CS(8) / .000000000993220 77E0 /
      DATA BI1CS(9) / .000000000007663 80E0 /
      DATA BI1CS(10) / .000000000000047 41E0 /
      DATA BI1CS(11) / .000000000000000 24E0 /
      DATA NTI1,XMIN,XSML,XMAX / 0,3*0. /
C     ***FIRST EXECUTABLE STATEMENT  BESI1
      IF (NTI1.NE.0) GO TO 10
      NTI1 = INITS(BI1CS,11,0.1*R1MACH(3))
      XMIN = 2.0*R1MACH(1)
      XSML = SQRT(8.0*R1MACH(3))
      XMAX = ALOG(R1MACH(2))
C
   10 Y = ABS(X)
      IF (Y.GT.3.0) GO TO 20
C
      BESI1 = 0.0
      IF (Y.EQ.0.0) RETURN
C
```

```
      IF (Y.LT.XMIN) CALL XERR ('BESI1   ABS(X) SO SMALL I1 UNDERFLOWS')
      IF (Y.GT.XMIN) BESI1 = 0.5*X
      IF (Y.GT.XSML) BESI1 = X*(.875+CSEVL(Y*Y/4.5-1.,BI1CS,NTI1))
      RETURN
C
   20 IF (Y.GT.XMAX) CALL XERR ('BESI1   ABS(X) SO BIG I1 OVERFLOWS')
C
      BESI1 = EXP(Y)*BESI1E(X)
C
      RETURN
      END
*DECK,BESI1E
      FUNCTION BESI1E (X)
C     ***BEGIN PROLOGUE  BESI1E
C     ***DATE WRITTEN   770401   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  C10B1
C     ***KEYWORDS  BESSEL FUNCTION,EXPONENTIALLY SCALED,FIRST KIND,
C     HYPERBOLIC BESSEL FUNCTION,MODIFIED BESSEL FUNCTION,
C     ORDER ONE
C     ***AUTHOR  FULLERTON, W., (LANL)
C     ***PURPOSE  COMPUTES THE EXPONENTIALLY SCALED HYPERBOLIC BESSEL
C     FUNCTION OF THE FIRST KIND OF ORDER ONE.
C     ***DESCRIPTION
C
C     BESI1E(X) CALCULATES THE EXPONENTIALLY SCALED MODIFIED (HYPERBOLIC
C     BESSEL FUNCTION OF THE FIRST KIND OF ORDER ONE FOR REAL ARGUMENT X
C     I.E., EXP(-ABS(X))*I1(X).
C
C     SERIES FOR BI1        ON THE INTERVAL  0.          TO  9.00000D+00
C     WITH WEIGHTED ERROR   2.40E-17
C     LOG WEIGHTED ERROR  16.62
C     SIGNIFICANT FIGURES REQUIRED  16.23
C     DECIMAL PLACES REQUIRED  17.14
C
C     SERIES FOR AI1        ON THE INTERVAL  1.25000D-01 TO  3.33333D-01
C     WITH WEIGHTED ERROR   6.98E-17
C     LOG WEIGHTED ERROR  16.16
C     SIGNIFICANT FIGURES REQUIRED  14.53
C     DECIMAL PLACES REQUIRED  16.82
C
C     SERIES FOR AI12       ON THE INTERVAL  0.          TO  1.25000D-01
C     WITH WEIGHTED ERROR   3.55E-17
C     LOG WEIGHTED ERROR  16.45
C     SIGNIFICANT FIGURES REQUIRED  14.69
C     DECIMAL PLACES REQUIRED  17.12
C     ***REFERENCES  (NONE)
C     ***ROUTINES CALLED  CSEVL,INITS,R1MACH,XERROR
C     ***END PROLOGUE  BESI1E
      DIMENSION BI1CS(11),AI1CS(21),AI12CS(22)
      DATA BI1CS(1) / -.001971713261099859E0 /
      DATA BI1CS(2) / .407348876675464 81E0 /
      DATA BI1CS(3) / .034838994299959456E0 /
```

```
DATA BI1CS(4) / .001545394556300123E0 /
DATA BI1CS(5) / .0000418885210983 77E0 /
DATA BI1CS(6) / .000000764902676483E0 /
DATA BI1CS(7) / .000000010042493924E0 /
DATA BI1CS(8) / .000000000099322077E0 /
DATA BI1CS(9) / .000000000000766380E0 /
DATA BI1CS(10) / .000000000000004741E0 /
DATA BI1CS(11) / .000000000000000024E0 /
DATA AI1CS(1) / -.02846744181881479E0 /
DATA AI1CS(2) / -.01922953231443221E0 /
DATA AI1CS(3) / -.00061151858579437E0 /
DATA AI1CS(4) / -.00002069971253350E0 /
DATA AI1CS(5) / .00000858561914581E0 /
DATA AI1CS(6) / .00000104949824671E0 /
DATA AI1CS(7) / -.00000029183389184E0 /
DATA AI1CS(8) / -.00000001559378146E0 /
DATA AI1CS(9) / .00000001318012367E0 /
DATA AI1CS(10) / -.00000000144842341E0 /
DATA AI1CS(11) / -.00000000029085122E0 /
DATA AI1CS(12) / .00000000012663889E0 /
DATA AI1CS(13) / -.00000000001664947E0 /
DATA AI1CS(14) / -.00000000000166665E0 /
DATA AI1CS(15) / .00000000000124260E0 /
DATA AI1CS(16) / -.00000000000027315E0 /
DATA AI1CS(17) / .00000000000002023E0 /
DATA AI1CS(18) / .00000000000000730E0 /
DATA AI1CS(19) / -.00000000000000333E0 /
DATA AI1CS(20) / .00000000000000071E0 /
DATA AI1CS(21) / -.00000000000000006E0 /
DATA AI12CS(1) / .028576235018280 14E0 /
DATA AI12CS(2) / -.00976109749136147E0 /
DATA AI12CS(3) / -.00011058893876263E0 /
DATA AI12CS(4) / -.00000388256480887E0 /
DATA AI12CS(5) / -.00000025122362377E0 /
DATA AI12CS(6) / -.00000002631468847E0 /
DATA AI12CS(7) / -.00000000383538039E0 /
DATA AI12CS(8) / -.00000000055897433E0 /
DATA AI12CS(9) / -.00000000001897495E0 /
DATA AI12CS(10) / .00000000003252602E0 /
DATA AI12CS(11) / .00000000001412580E0 /
DATA AI12CS(12) / .00000000000203564E0 /
DATA AI12CS(13) / -.00000000000071985E0 /
DATA AI12CS(14) / -.00000000000040836E0 /
DATA AI12CS(15) / -.00000000000002101E0 /
DATA AI12CS(16) / .00000000000004273E0 /
DATA AI12CS(17) / .00000000000001041E0 /
DATA AI12CS(18) / -.00000000000000382E0 /
DATA AI12CS(19) / -.00000000000000186E0 /
DATA AI12CS(20) / .00000000000000033E0 /
DATA AI12CS(21) / .00000000000000028E0 /
DATA AI12CS(22) / -.00000000000000003E0 /
DATA NTI1,NTAI1,NTAI12,XMIN,XSML / 3*0,2*0. /
```

```
C     ***FIRST EXECUTABLE STATEMENT  BESI1E
      IF (NTI1.NE.0) GO TO 10
      NTI1 = INITS(BI1CS,11,0.1*R1MACH(3))
      NTAI1 = INITS(AI1CS,21,0.1*R1MACH(3))
      NTAI12 = INITS(AI12CS,22,0.1*R1MACH(3))
C
      XMIN = 2.0*R1MACH(1)
      XSML = SQRT(8.0*R1MACH(3))
C
   10 Y = ABS(X)
      IF (Y.GT.3.0) GO TO 20
C
      BESI1E = 0.0
      IF (Y.EQ.0.0) RETURN
C
      IF (Y.LT.XMIN) CALL XERR ('BESI1E  ABS(X) SO SMALL I1 UNDERFLOWS')
      IF (Y.GT.XMIN) BESI1E = 0.5*X
      IF (Y.GT.XSML) BESI1E = X*(.875+CSEVL(Y*Y/4.5-1.,BI1CS,NTI1))
      BESI1E = EXP(-Y)*BESI1E
      RETURN
C
   20 IF (Y.LE.8.) BESI1E = (.375+CSEVL((48./Y-11.)/5.,AI1CS,NTAI1))/
     *  SQRT(Y)
      IF (Y.GT.8.) BESI1E = (.375+CSEVL(16./Y-1.0,AI12CS,NTAI12))/SQRT(Y
     *  )
      BESI1E = SIGN(BESI1E,X)
C
      RETURN
      END
*DECK,BESK0
      FUNCTION BESK0 (X)
C     ***BEGIN PROLOGUE  BESK0
C     ***DATE WRITTEN   770401   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  C10B1
C     ***KEYWORDS  BESSEL FUNCTION,HYBERBOLIC BESSEL FUNCTION,
C     MODIFIED BESSEL FUNCTION,ORDER ZERO,THIRD KIND
C     ***AUTHOR  FULLERTON, W., (LANL)
C     ***PURPOSE  COMPUTES THE HYPERBOLIC BESSEL FUNCTION OF THE THIRD K
C     OF ORDER ZERO
C     ***DESCRIPTION
C
C     BESK0(X) CALCULATES THE MODIFIED (HYPERBOLIC) BESSEL FUNCTION
C     OF THE THIRD KIND OF ORDER ZERO FOR REAL ARGUMENT X .GT. 0.0.
C
C     SERIES FOR BK0        ON THE INTERVAL  0.          TO  4.00000D+00
C     WITH WEIGHTED ERROR   3.57E-19
C     LOG WEIGHTED ERROR  18.45
C     SIGNIFICANT FIGURES REQUIRED  17.99
C     DECIMAL PLACES REQUIRED  18.97
C     ***REFERENCES  (NONE)
C     ***ROUTINES CALLED  BESI0,BESK0E,CSEVL,INITS,R1MACH,XERROR
C     ***END PROLOGUE  BESK0
```

```
      DIMENSION BKOCS(11)
      DATA BKOCS(1) / -.035327393233902768 72E0 /
      DATA BKOCS(2) / .344289899924628486 9E0 /
      DATA BKOCS(3) / .035979936515361501 63E0 /
      DATA BKOCS(4) / .001264615411446925 92E0 /
      DATA BKOCS(5) / .000022862121031194 51E0 /
      DATA BKOCS(6) / .000002534791079026 1E0 /
      DATA BKOCS(7) / .000000019045163772 2E0 /
      DATA BKOCS(8) / .000000000103496952 5E0 /
      DATA BKOCS(9) / .000000000000425981 6E0 /
      DATA BKOCS(10) / .000000000000001374 4E0 /
      DATA BKOCS(11) / .000000000000000003 5E0 /
      DATA NTKO,XSML,XMAX / 0,0.,0. /
C     ***FIRST EXECUTABLE STATEMENT  BESKO
      IF (NTKO.NE.0) GO TO 10
      NTKO = INITS(BKOCS,11,0.1*R1MACH(3))
      XSML = SQRT(4.0*R1MACH(3))
      XMAX = -ALOG(R1MACH(1))
      XMAX = XMAX-0.5*XMAX*ALOG(XMAX)/(XMAX+0.5)-0.01
C
   10 IF (X.LE.0.) CALL XERR ('BESKO   X IS ZERO OR NEGATIVE')
      IF (X.GT.2.) GO TO 20
C
      Y = 0.
      IF (X.GT.XSML) Y = X*X
      BESKO = -ALOG(0.5*X)*BESIO(X)-.25+CSEVL(.5*Y-1.,BKOCS,NTKO)
      RETURN
C
   20 BESKO = 0.
      IF (X.GT.XMAX) CALL XERR ('BESKO   X SO BIG KO UNDERFLOWS')
      IF (X.GT.XMAX) RETURN
C
      BESKO = EXP(-X)*BESKOE(X)
C
      RETURN
      END
*DECK,BESKOE
      FUNCTION BESKOE (X)
C     ***BEGIN PROLOGUE  BESKOE
C     ***DATE WRITTEN   770401   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  C10B1
C     ***KEYWORDS  BESSEL FUNCTION,EXPONENTIALLY SCALED,
C     HYPERBOLIC BESSEL FUNCTION,MODIFIED BESSEL FUNCTION,
C     ORDER ZERO,THIRD KIND
C     ***AUTHOR  FULLERTON, W., (LANL)
C     ***PURPOSE  COMPUTES THE EXPONENTIALLY SCALED HYPERBOLIC BESSEL
C     FUNCTION OF THE THIRD KIND OF ORDER ZERO.
C     ***DESCRIPTION
C
C     BESKOE(X) COMPUTES THE EXPONENTIALLY SCALED MODIFIED (HYPERBOLIC)
C     BESSEL FUNCTION OF THIRD KIND OF ORDER ZERO FOR REAL ARGUMENT
C     X .GT. 0.0, I.E., EXP(X)*KO(X).
C
```

```
C     SERIES FOR BK0        ON THE INTERVAL  0.          TO  4.00000D+00
C     WITH WEIGHTED ERROR   3.57E-19
C     LOG WEIGHTED ERROR  18.45
C     SIGNIFICANT FIGURES REQUIRED  17.99
C     DECIMAL PLACES REQUIRED  18.97
C
C     SERIES FOR AK0        ON THE INTERVAL  1.25000D-01 TO  5.00000D-01
C     WITH WEIGHTED ERROR   5.34E-17
C     LOG WEIGHTED ERROR  16.27
C     SIGNIFICANT FIGURES REQUIRED  14.92
C     DECIMAL PLACES REQUIRED  16.89
C
C     SERIES FOR AK02       ON THE INTERVAL  0.          TO  1.25000D-01
C     WITH WEIGHTED ERROR   2.34E-17
C     LOG WEIGHTED ERROR  16.63
C     SIGNIFICANT FIGURES REQUIRED  14.67
C     DECIMAL PLACES REQUIRED  17.20
C     ***REFERENCES  (NONE)
C     ***ROUTINES CALLED  BESI0,CSEVL,INITS,R1MACH,XERROR
C     ***END PROLOGUE  BESK0E
      DIMENSION BK0CS(11),AK0CS(17),AK02CS(14)
      DATA BK0CS(1) / -.03532739323390276872E0 /
      DATA BK0CS(2) / .34428989992462848 69E0 /
      DATA BK0CS(3) / .035979936515361501 63E0 /
      DATA BK0CS(4) / .001264615411446925 92E0 /
      DATA BK0CS(5) / .000022862121031194 51E0 /
      DATA BK0CS(6) / .000000253479107902 61E0 /
      DATA BK0CS(7) / .000000001904516377 22E0 /
      DATA BK0CS(8) / .000000000010349695 25E0 /
      DATA BK0CS(9) / .000000000000042598 16E0 /
      DATA BK0CS(10) / .000000000000000137 44E0 /
      DATA BK0CS(11) / .000000000000000000 35E0 /
      DATA AK0CS(1) / -.07643947903327941E0 /
      DATA AK0CS(2) / -.02235652605699819E0 /
      DATA AK0CS(3) / .00077341811546938E0 /
      DATA AK0CS(4) / -.00004281006688886E0 /
      DATA AK0CS(5) / .00000308170017386E0 /
      DATA AK0CS(6) / -.00000026393672220E0 /
      DATA AK0CS(7) / .00000002563713036E0 /
      DATA AK0CS(8) / -.00000000274270554E0 /
      DATA AK0CS(9) / .00000000031694296E0 /
      DATA AK0CS(10) / -.00000000003902353E0 /
      DATA AK0CS(11) / .00000000000506804E0 /
      DATA AK0CS(12) / -.00000000000068895E0 /
      DATA AK0CS(13) / .00000000000009744E0 /
      DATA AK0CS(14) / -.00000000000001427E0 /
      DATA AK0CS(15) / .00000000000000215E0 /
      DATA AK0CS(16) / -.00000000000000033E0 /
      DATA AK0CS(17) / .00000000000000005E0 /
      DATA AK02CS(1) / -.01201869826307592E0 /
      DATA AK02CS(2) / -.00917485269102569E0 /
      DATA AK02CS(3) / .00014445509317750E0 /
      DATA AK02CS(4) / -.00000401361417543E0 /
```

```
      DATA AK02CS(5) / .000000156783181O8E0 /
      DATA AK02CS(6) / -.00000000777011043E0 /
      DATA AK02CS(7) / .00000000046111825E0 /
      DATA AK02CS(8) / -.00000000003158592E0 /
      DATA AK02CS(9) / .00000000000243501E0 /
      DATA AK02CS(10) / -.00000000000020743E0 /
      DATA AK02CS(11) / .00000000000001925E0 /
      DATA AK02CS(12) / -.00000000000000192E0 /
      DATA AK02CS(13) / .00000000000000020E0 /
      DATA AK02CS(14) / -.00000000000000002E0 /
      DATA NTK0,NTAK0,NTAK02,XSML / 3*0,0. /
C     ***FIRST EXECUTABLE STATEMENT  BESK0E
      IF (NTK0.NE.0) GO TO 10
      NTK0 = INITS(BK0CS,11,0.1*R1MACH(3))
      NTAK0 = INITS(AK0CS,17,0.1*R1MACH(3))
      NTAK02 = INITS(AK02CS,14,0.1*R1MACH(3))
      XSML = SQRT(4.0*R1MACH(3))
C
   10 IF (X.LE.0.) CALL XERR ('BESK0E  X IS ZERO OR NEGATIVE')
      IF (X.GT.2.) GO TO 20
C
      Y = 0.
      IF (X.GT.XSML) Y = X*X
      BESK0E = EXP(X)*(-ALOG(0.5*X)*BESI0(X)-.25+CSEVL(.5*Y-1.,BK0CS,
     *  NTK0))
      RETURN
C
   20 IF (X.LE.8.) BESK0E = (1.25+CSEVL((16./X-5.)/3.,AK0CS,NTAK0))/SQRT
     *  (X)
      IF (X.GT.8.) BESK0E = (1.25+CSEVL(16./X-1.,AK02CS,NTAK02))/SQRT(X)
C
      RETURN
      END
*DECK,BESK1
      FUNCTION BESK1 (X)
C     ***BEGIN PROLOGUE  BESK1
C     ***DATE WRITTEN   770401   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  C10B1
C     ***KEYWORDS  BESSEL FUNCTION,HYPERBOLIC BESSEL FUNCTION,
C     MODIFIED BESSEL FUNCTION,ORDER ONE,THIRD KIND
C     ***AUTHOR  FULLERTON, W., (LANL)
C     ***PURPOSE  COMPUTES THE HYPERBOLIC BESSEL FUNCTION OF THE THIRD K
C     OF ORDER ONE
C     ***DESCRIPTION
C
C     BESK1(X) COMPUTES THE MODIFIED (HYPERBOLIC) BESSEL FUNCTION OF THI
C     KIND OF ORDER ONE FOR REAL ARGUMENT X, WHERE X .GT. 0.
C
C     SERIES FOR BK1        ON THE INTERVAL  0.          TO  4.00000D+00
C     WITH WEIGHTED ERROR   7.02E-18
C     LOG WEIGHTED ERROR  17.15
C     SIGNIFICANT FIGURES REQUIRED  16.73
C     DECIMAL PLACES REQUIRED  17.67
```

```
C     ***REFERENCES  (NONE)
C     ***ROUTINES CALLED  BESI1,BESK1E,CSEVL,INITS,R1MACH,XERROR
C     ***END PROLOGUE  BESK1
      DIMENSION BK1CS(11)
      DATA BK1CS(1) / .025300227338947705E0 /
      DATA BK1CS(2) / -.3531559607765448760E0 /
      DATA BK1CS(3) / -.1226111808226571480E0 /
      DATA BK1CS(4) / -.0069757238596398643E0 /
      DATA BK1CS(5) / -.0001730288957513052E0 /
      DATA BK1CS(6) / -.0000024334061415659E0 /
      DATA BK1CS(7) / -.0000000221338763073E0 /
      DATA BK1CS(8) / -.0000000001411488392E0 /
      DATA BK1CS(9) / -.0000000000006666901E0 /
      DATA BK1CS(10) / -.0000000000000024274E0 /
      DATA BK1CS(11) / -.0000000000000000070E0 /
      DATA NTK1,XMIN,XSML,XMAX / 0,0.,0.,0. /
C     ***FIRST EXECUTABLE STATEMENT  BESK1
      IF (NTK1.NE.0) GO TO 10
      NTK1 = INITS(BK1CS,11,0.1*R1MACH(3))
      XMIN = EXP(AMAX1(ALOG(R1MACH(1)),-ALOG(R1MACH(2)))+.01)
      XSML = SQRT(4.0*R1MACH(3))
      XMAX = -ALOG(R1MACH(1))
      XMAX = XMAX-0.5*XMAX*ALOG(XMAX)/(XMAX+0.5)
C
   10 IF (X.LE.0.) CALL XERR ('BESK1   X IS ZERO OR NEGATIVE')
      IF (X.GT.2.0) GO TO 20
C
      IF (X.LT.XMIN) CALL XERR ('BESK1   X SO SMALL K1 OVERFLOWS')
      Y = 0.
      IF (X.GT.XSML) Y = X*X
      BESK1 = ALOG(0.5*X)*BESI1(X)+(0.75+CSEVL(.5*Y-1.,BK1CS,NTK1))/X
      RETURN
C
   20 BESK1 = 0.
      IF (X.GT.XMAX) CALL XERR ('BESK1   X SO BIG K1 UNDERFLOWS')
      IF (X.GT.XMAX) RETURN
C
      BESK1 = EXP(-X)*BESK1E(X)
C
      RETURN
      END
*DECK,BESK1E
      FUNCTION BESK1E (X)
C     ***BEGIN PROLOGUE  BESK1E
C     ***DATE WRITTEN   770401   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  C10B1
C     ***KEYWORDS  BESSEL FUNCTION,EXPONENTIALLY SCALED,
C     HYPERBOLIC BESSEL FUNCTION,MODIFIED BESSEL FUNCTION,
C     ORDER ONE,THIRD KIND
C     ***AUTHOR  FULLERTON, W., (LANL)
C     ***PURPOSE  COMPUTES THE EXPONENTIALLY SCALED HYPERBOLIC BESSEL
C     FUNCTION OF THE THIRD KIND OF ORDER ONE.
C     ***DESCRIPTION
C
```

```
C     BESK1E(X) COMPUTES THE EXPONENTIALLY SCALED MODIFIED (HYPERBOLIC)
C     BESSEL FUNCTION OF THIRD KIND OF ORDER ONE FOR REAL ARGUMENT
C     X .GT. 0.0, I.E., EXP(X)*K1(X).
C
C     SERIES FOR BK1        ON THE INTERVAL  0.          TO  4.00000D+00
C     WITH WEIGHTED ERROR   7.02E-18
C     LOG WEIGHTED ERROR  17.15
C     SIGNIFICANT FIGURES REQUIRED  16.73
C     DECIMAL PLACES REQUIRED  17.67
C
C     SERIES FOR AK1        ON THE INTERVAL  1.25000D-01 TO  5.00000D-01
C     WITH WEIGHTED ERROR   6.06E-17
C     LOG WEIGHTED ERROR  16.22
C     SIGNIFICANT FIGURES REQUIRED  15.41
C     DECIMAL PLACES REQUIRED  16.83
C
C     SERIES FOR AK12       ON THE INTERVAL  0.          TO  1.25000D-01
C     WITH WEIGHTED ERROR   2.58E-17
C     LOG WEIGHTED ERROR  16.59
C     SIGNIFICANT FIGURES REQUIRED  15.22
C     DECIMAL PLACES REQUIRED  17.16
C     ***REFERENCES  (NONE)
C     ***ROUTINES CALLED  BESI1,CSEVL,INITS,R1MACH,XERROR
C     ***END PROLOGUE  BESK1E
      DIMENSION BK1CS(11),AK1CS(17),AK12CS(14)
      DATA BK1CS(1) / .02530022733894770 5E0 /
      DATA BK1CS(2) / -.3531559607765448 76E0 /
      DATA BK1CS(3) / -.1226111808226571 48E0 /
      DATA BK1CS(4) / -.0069757238596398643E0 /
      DATA BK1CS(5) / -.0001730288957513052E0 /
      DATA BK1CS(6) / -.0000243340614156 59E0 /
      DATA BK1CS(7) / -.0000000221338763073E0 /
      DATA BK1CS(8) / -.0000000014114883 92E0 /
      DATA BK1CS(9) / -.0000000000066669 01E0 /
      DATA BK1CS(10) / -.0000000000000242 74E0 /
      DATA BK1CS(11) / -.0000000000000000 70E0 /
      DATA AK1CS(1) / .2744313406973883E0 /
      DATA AK1CS(2) / .0757198995319936 8E0 /
      DATA AK1CS(3) / -.0014410515564754 0E0 /
      DATA AK1CS(4) / .0006650116955125E0 /
      DATA AK1CS(5) / -.0000436998470952E0 /
      DATA AK1CS(6) / .0000035402774997E0 /
      DATA AK1CS(7) / -.0000003311163779E0 /
      DATA AK1CS(8) / .0000000344597758E0 /
      DATA AK1CS(9) / -.0000000038989323E0 /
      DATA AK1CS(10) / .0000000004720819E0 /
      DATA AK1CS(11) / -.0000000000604783E0 /
      DATA AK1CS(12) / .0000000000081284E0 /
      DATA AK1CS(13) / -.0000000000011386E0 /
      DATA AK1CS(14) / .0000000000001654E0 /
      DATA AK1CS(15) / -.0000000000000248E0 /
      DATA AK1CS(16) / .0000000000000038E0 /
      DATA AK1CS(17) / -.0000000000000006E0 /
```

```
      DATA AK12CS(1) / .06379308343739001E0 /
      DATA AK12CS(2) / .02832887813049721E0 /
      DATA AK12CS(3) / -.00024753706739052E0 /
      DATA AK12CS(4) / .00000577197245160E0 /
      DATA AK12CS(5) / -.00000020689392195E0 /
      DATA AK12CS(6) / .00000000973998344E0 /
      DATA AK12CS(7) / -.00000000055853361E0 /
      DATA AK12CS(8) / .00000000003732996E0 /
      DATA AK12CS(9) / -.00000000000282505E0 /
      DATA AK12CS(10) / .00000000000023720E0 /
      DATA AK12CS(11) / -.00000000000002176E0 /
      DATA AK12CS(12) / .00000000000000215E0 /
      DATA AK12CS(13) / -.00000000000000022E0 /
      DATA AK12CS(14) / .00000000000000002E0 /
      DATA NTK1,NTAK1,NTAK12,XMIN,XSML / 3*0,2*0. /
C     ***FIRST EXECUTABLE STATEMENT  BESK1E
      IF (NTK1.NE.0) GO TO 10
      NTK1 = INITS(BK1CS,11,0.1*R1MACH(3))
      NTAK1 = INITS(AK1CS,17,0.1*R1MACH(3))
      NTAK12 = INITS(AK12CS,14,0.1*R1MACH(3))
C
      XMIN = EXP(AMAX1(ALOG(R1MACH(1)),-ALOG(R1MACH(2)))+.01)
      XSML = SQRT(4.0*R1MACH(3))
C
   10 IF (X.LE.0.) CALL XERR ('BESK1E  X IS ZERO OR NEGATIVE')
      IF (X.GT.2.0) GO TO 20
C
      IF (X.LT.XMIN) CALL XERR ('BESK1E  X SO SMALL K1 OVERFLOWS')
      Y = 0.
      IF (X.GT.XSML) Y = X*X
      BESK1E = EXP(X)*(ALOG(0.5*X)*BESI1(X)+(0.75+CSEVL(.5*Y-1.,BK1CS,
     *  NTK1))/X)
      RETURN
C
   20 IF (X.LE.8.) BESK1E = (1.25+CSEVL((16./X-5.)/3.,AK1CS,NTAK1))/SQRT
     *  (X)
      IF (X.GT.8.) BESK1E = (1.25+CSEVL(16./X-1.,AK12CS,NTAK12))/SQRT(X)
C
      RETURN
      END
*DECK,BKIAS
      SUBROUTINE BKIAS (X,N,KTRMS,T,ANS,IND,MS,GMRN,H,IERR)
C     ***BEGIN PROLOGUE  BKIAS
C     ***REFER TO  BSKIN
C     ***ROUTINES CALLED  BDIFF,GAMRN,HKSEQ,R1MACH
C     ***DESCRIPTION
C
C     BKIAS COMPUTES REPEATED INTEGRALS OF THE K0 BESSEL FUNCTION
C     BY THE ASYMPTOTIC EXPANSION
C     ***END PROLOGUE  BKIAS
      INTEGER I,II,IND,J,JMI,JN,K,KK,KM,KTRMS,MM,MP,MS,N,IERR
      REAL ANS,B,BND,DEN1,DEN2,DEN3,ER,ERR,FJ,FK,FLN,FM1,GMRN,G1,GS,H,HN
```

```
     *   ,HRTPI,RAT,RG1,RXP,RZ,RZX,S,SS,SUMI,SUMJ,T,TOL,V,W,X,XP,Z
          REAL GAMRN,R1MACH
          DIMENSION B(120),XP(16),S(31),H(*),V(52),W(52),T(50),BND(15)
C     ---------------------------------------------------------------
C     COEFFICIENTS OF POLYNOMIAL P(J-1,X), J=1,15
C     ---------------------------------------------------------------
          DATA B(1),B(2),B(3),B(4),B(5),B(6),B(7),B(8),B(9),B(10),B(11),B(12
     *   ),B(13),B(14),B(15),B(16),B(17),B(18),B(19),B(20),B(21),B(22),B(
     *   23),B(24) / 1.0000000000000000E+00,1.0000000000000000E+00,-
     *   2.0000000000000000E+00,1.0000000000000000E+00,-
     *   8.0000000000000000E+00,6.0000000000000000E+00,
     *   1.0000000000000000E+00,-2.2000000000000000E+01,
     *   5.8000000000000000E+01,-2.4000000000000000E+01,
     *   1.0000000000000000E+00,-5.2000000000000000E+01,
     *   3.2800000000000000E+02,-4.4400000000000000E+02,
     *   1.2000000000000000E+02,1.0000000000000000E+00,-
     *   1.1400000000000000E+02,1.4520000000000000E+03,-
     *   4.4000000000000000E+03,3.7080000000000000E+03,-
     *   7.2000000000000000E+02,1.0000000000000000E+00,-
     *   2.4000000000000000E+02,5.6100000000000000E+03 /
          DATA B(25),B(26),B(27),B(28),B(29),B(30),B(31),B(32),B(33),B(34),B
     *   (35),B(36),B(37),B(38),B(39),B(40),B(41),B(42),B(43),B(44),B(45)
     *   ,B(46),B(47),B(48) / -3.2120000000000000E+04,
     *   5.8140000000000000E+04,-3.3984000000000000E+04,
     *   5.0400000000000000E+03,1.0000000000000000E+00,-
     *   4.9400000000000000E+02,1.9950000000000000E+04,-
     *   1.9580000000000000E+05,6.4402000000000000E+05,-
     *   7.8530400000000000E+05,3.4113600000000000E+05,-
     *   4.0320000000000000E+04,1.0000000000000000E+00,-
     *   1.0040000000000000E+03,6.7260000000000000E+04,-
     *   1.0625000000000000E+06,5.7655000000000000E+06,-
     *   1.2440064000000000E+07,1.1026296000000000E+07,-
     *   3.7339200000000000E+06,3.6288000000000000E+05,
     *   1.0000000000000000E+00,-2.0260000000000000E+03,
     *   2.1884800000000000E+05 /
          DATA B(49),B(50),B(51),B(52),B(53),B(54),B(55),B(56),B(57),B(58),B
     *   (59),B(60),B(61),B(62),B(63),B(64),B(65),B(66),B(67),B(68),B(69)
     *   ,B(70),B(71),B(72) / -5.3261600000000000E+06,
     *   4.4765000000000000E+07,-1.5535738400000000E+08,
     *   2.3890490400000000E+08,-1.6218691200000000E+08,
     *   4.4339040000000000E+07,-3.6288000000000000E+06,
     *   1.0000000000000000E+00,-4.0720000000000000E+03,
     *   6.9503800000000000E+05,-2.5243904000000000E+07,
     *   3.1436972000000000E+08,-1.6483843040000000E+09,
     *   4.0026950880000000E+09,-4.6421639520000000E+09,
     *   2.5074812160000000E+09,-5.6835648000000000E+08,
     *   3.9916800000000000E+07,1.0000000000000000E+00,-
     *   8.1660000000000000E+03,2.1706260000000000E+06,-
     *   1.1487637600000000E+08,2.0514827760000000E+09,-
     *   1.5548960784000000E+10 /
```

```
      DATA B(73),B(74),B(75),B(76),B(77),B(78),B(79),B(80),B(81),B(82),B
     *  (83),B(84),B(85),B(86),B(87),B(88),B(89),B(90),B(91),B(92),B(93)
     *  ,B(94),B(95),B(96) / 5.604139878400000000E+10,-
     *  1.011804330240000000E+11,9.219979022400000000E+10,-
     *  4.078830182400000000E+10,7.827719040000000000E+09,-
     *  4.790016000000000000E+08,1.000000000000000000E+00,-
     *  1.635600000000000000E+04,6.699696000000000000E+06,-
     *  5.072592760000000000E+08,1.266981777600000000E+10,-
     *  1.343234202240000000E+11,6.877200463840000000E+11,-
     *  1.818188642304000000E+12,2.549865473424000000E+12,-
     *  1.883079661824000000E+12,6.979294368000000000E+11,-
     *  1.153360857600000000E+11,6.227020800000000000E+09,
     *  1.000000000000000000E+00,-3.273800000000000000E+04,
     *  2.050798800000000000E+07,-2.189829808000000000E+09,
     *  7.501605222800000000E+10 /
      DATA B(97),B(98),B(99),B(100),B(101),B(102),B(103),B(104),B(105),B
     *  (106),B(107),B(108),B(109),B(110),B(111),B(112),B(113),B(114),B(
     *  115),B(116),B(117),B(118) / -1.084676512416000000E+12,
     *  7.634832149392000000E+12,-2.829991006611200000E+13,
     *  5.749437346459200000E+13,-6.472837513987200000E+13,
     *  3.968957805580800000E+13,-1.255090401792000000E+13,
     *  1.810992556800000000E+12,-8.717829120000000000E+10,
     *  1.000000000000000000E+00,-6.550400000000000000E+04,
     *  6.240789000000000000E+07,-9.292526920000000000E+09,
     *  4.298260063400000000E+11,-8.308444327968000000E+12,
     *  7.839138483131200000E+13,-3.943655878155200000E+14,
     *  1.111747472569680000E+15,-1.797171220690560000E+15,
     *  1.666424486271456000E+15,-8.650232532195840000E+14,
     *  2.369082715430400000E+14 /
      DATA B(119),B(120) / -3.019637698560000000E+13,1.307674368000000000E
     *  +12 /
C     ------------------------------------------------------------------
C     BOUNDS B(M,K) , K=M-3
C     ------------------------------------------------------------------
      DATA BND(1),BND(2),BND(3),BND(4),BND(5),BND(6),BND(7),BND(8),BND(9
     *  ),BND(10),BND(11),BND(12),BND(13),BND(14),BND(15) / 1.0E0,1.0E0,
     *  1.0E0,1.0E0,3.10E0,5.18E0,11.7E0,29.8E0,90.4E0,297.0E0,1070.0E0,
     *  4290.0E0,18100.0E0,84700.0E0,408000.0E0 /
      DATA HRTPI / 8.862269254527580l4E-01 /
C
C     ***FIRST EXECUTABLE STATEMENT  BKIAS
      IERR = 0
      TOL = AMAX1(R1MACH(4),1.0E-18)
      FLN = FLOAT(N)
      RZ = 1.0E0/(X+FLN)
      RZX = X*RZ
      Z = 0.5E0*(X+FLN)
      IF (IND.GT.1) GO TO 10
      GMRN = GAMRN(Z)
   10 CONTINUE
      GS = HRTPI*GMRN
      G1 = GS+GS
```

```
      RG1 = 1.0E0/G1
      GMRN = (RZ+RZ)/GMRN
      IF (IND.GT.1) GO TO 70
C     ------------------------------------------------------------------
C     EVALUATE ERROR FOR M-MS
C     ------------------------------------------------------------------
      HN = 0.5E0*FLN
      DEN2 = KTRMS+KTRMS+N
      DEN3 = DEN2-2.0E0
      DEN1 = X+DEN2
      ERR = RG1*(X+X)/(DEN1-1.0E0)
      IF (N.EQ.0) GO TO 20
      RAT = 1.0E0/(FLN*FLN)
   20 CONTINUE
      IF (KTRMS.EQ.0) GO TO 30
      FJ = FLOAT(KTRMS)
      RAT = 0.25E0/(HRTPI*DEN3*SQRT(FJ))
   30 CONTINUE
      ERR = ERR*RAT
      FJ = -3.0E0
      DO 50 J = 1, 15
        IF (J.LE.5) ERR = ERR/DEN1
        FM1 = AMAX1(1.0E0,FJ)
        FJ = FJ+1.0E0
        ER = BND(J)*ERR
        IF (KTRMS.EQ.0) GO TO 40
        ER = ER/FM1
        IF (ER.LT.TOL) GO TO 60
        IF (J.GE.5) ERR = ERR/DEN3
        GO TO 50
   40   CONTINUE
        ER = ER*(1.0E0+HN/FM1)
        IF (ER.LT.TOL) GO TO 60
        IF (J.GE.5) ERR = ERR/FLN
   50 CONTINUE
      GO TO 200
   60 CONTINUE
      MS = J
   70 CONTINUE
      MM = MS+MS
      MP = MM+1
C     ------------------------------------------------------------------
C     H(K)=(-Z)**(K)*(PSI(K-1,Z)-PSI(K-1,Z+0.5))/GAMMA(K) , K=1,2,...,MM
C     ------------------------------------------------------------------
      IF (IND.GT.1) GO TO 80
      CALL HKSEQ (Z,MM,H,IERR)
      GO TO 100
   80 CONTINUE
      RAT = Z/(Z-0.5E0)
      RXP = RAT
      DO 90 I = 1, MM
        H(I) = RXP*(1.0E0-H(I))
        RXP = RXP*RAT
```

```
   90 CONTINUE
  100 CONTINUE
C     ------------------------------------------------------------------
C     SCALED S SEQUENCE
C     ------------------------------------------------------------------
      S(1) = 1.0E0
      FK = 1.0E0
      DO 120 K = 2, MP
        SS = 0.0E0
        KM = K-1
        I = KM
        DO 110 J = 1, KM
          SS = SS+S(J)*H(I)
          I = I-1
  110   CONTINUE
        S(K) = SS/FK
        FK = FK+1.0E0
  120 CONTINUE
C     ------------------------------------------------------------------
C     SCALED S-TILDA SEQUENCE
C     ------------------------------------------------------------------
      IF (KTRMS.EQ.0) GO TO 160
      FK = 0.0E0
      SS = 0.0E0
      RG1 = RG1/Z
      DO 130 K = 1, KTRMS
        V(K) = Z/(Z+FK)
        W(K) = T(K)*V(K)
        SS = SS+W(K)
        FK = FK+1.0E0
  130 CONTINUE
      S(1) = S(1)-SS*RG1
      DO 150 I = 2, MP
        SS = 0.0E0
        DO 140 K = 1, KTRMS
          W(K) = W(K)*V(K)
          SS = SS+W(K)
  140   CONTINUE
        S(I) = S(I)-SS*RG1
  150 CONTINUE
  160 CONTINUE
C     ------------------------------------------------------------------
C     SUM ON J
C     ------------------------------------------------------------------
      SUMJ = 0.0E0
      JN = 1
      RXP = 1.0E0
      XP(1) = 1.0E0
      DO 190 J = 1, MS
        JN = JN+J-1
        XP(J+1) = XP(J)*RZX
        RXP = RXP*RZ
C       ------------------------------------------------------------------
```

```
C       SUM ON I
C       ------------------------------------------------------------
        SUMI = 0.0E0
        II = JN
        DO 180 I = 1, J
          JMI = J-I+1
          KK = J+I+1
          DO 170 K = 1, JMI
            V(K) = S(KK)*XP(K)
            KK = KK+1
  170     CONTINUE
          CALL BDIFF (JMI,V)
          SUMI = SUMI+B(II)*V(JMI)*XP(I+1)
          II = II+1
  180   CONTINUE
        SUMJ = SUMJ+SUMI*RXP
  190 CONTINUE
      ANS = GS*(S(1)-SUMJ)
      RETURN
  200 CONTINUE
      IERR = 2
      RETURN
      END
*DECK,BKISR
      SUBROUTINE BKISR (X,N,SUM,IERR)
C     ***BEGIN PROLOGUE  BKISR
C     ***REFER TO  BSKIN
C     ***ROUTINES CALLED  PSIXN,R1MACH
C     ***DESCRIPTION
C
C     BKISR COMPUTES REPEATED INTEGRALS OF THE K0 BESSEL FUNCTION
C     BY THE SERIES FOR N=0,1, AND 2.
C     ***END PROLOGUE  BKISR
      INTEGER I,IERR,K,KK,KKN,K1,N,NP
      REAL AK,ATOL,BK,C,FK,FN,HX,HXS,POL,PR,SUM,TKP,TOL,TRM,X,XLN
      REAL PSIXN,R1MACH
      DIMENSION C(2)
      DATA C(1),C(2) / 1.57079632679489662E+00,1.0E0 /
C
C     ***FIRST EXECUTABLE STATEMENT  BKISR
      IERR = 0
      TOL = AMAX1(R1MACH(4),1.0E-18)
      IF (X.LT.TOL) GO TO 50
      PR = 1.0E0
      POL = 0.0E0
      IF (N.EQ.0) GO TO 20
      DO 10 I = 1, N
        POL = -POL*X+C(I)
        PR = PR*X/FLOAT(I)
   10 CONTINUE
   20 CONTINUE
      HX = X*0.5E0
      HXS = HX*HX
```

```
      XLN = ALOG(HX)
      NP = N+1
      TKP = 3.0E0
      FK = 2.0E0
      FN = FLOAT(N)
      BK = 4.0E0
      AK = 2.0E0/((FN+1.0E0)*(FN+2.0E0))
      SUM = AK*(PSIXN(N+3)-PSIXN(3)+PSIXN(2)-XLN)
      ATOL = SUM*TOL*0.75E0
      DO 30 K = 2, 20
        AK = AK*(HXS/BK)*((TKP+1.0E0)/(TKP+FN+1.0E0))*(TKP/(TKP+FN))
        K1 = K+1
        KK = K1+K
        KKN = KK+N
        TRM = (PSIXN(K1)+PSIXN(KKN)-PSIXN(KK)-XLN)*AK
        SUM = SUM+TRM
        IF (ABS(TRM).LE.ATOL) GO TO 40
        TKP = TKP+2.0E0
        BK = BK+TKP
        FK = FK+1.0E0
   30 CONTINUE
      GO TO 70
   40 CONTINUE
      SUM = (SUM*HXS+PSIXN(NP)-XLN)*PR
      IF (N.EQ.1) SUM = -SUM
      SUM = POL+SUM
      RETURN
C     ------------------------------------------------------------------
C     SMALL X CASE, X.LT.WORD TOLERANCE
C     ------------------------------------------------------------------
   50 CONTINUE
      IF (N.GT.0) GO TO 60
      HX = X*0.5E0
      SUM = PSIXN(1)-ALOG(HX)
      RETURN
   60 CONTINUE
      SUM = C(N)
      RETURN
   70 CONTINUE
      IERR = 2
      RETURN
      END
*DECK,BSKIN
      SUBROUTINE BSKIN (X,N,KODE,M,Y,NZ,IERR)
C     ***BEGIN PROLOGUE  BSKIN
C     ***DATE WRITTEN   820601   (YYMMDD)
C     ***REVISION DATE  851111   (YYMMDD)
C     ***CATEGORY NO.  C10F
C     ***KEYWORDS  BICKLEY FUNCTIONS,EXPONENTIAL INTEGRALS,
C     INTEGRALS OF BESSEL FUNCTIONS,K-ZERO BESSEL FUNCTION
C     ***AUTHOR  AMOS, DONALD E., SANDIA NATIONAL LABORATORIES
C     ***PURPOSE  COMPUTE REPEATED INTEGRALS OF THE K-ZERO BESSEL
C     FUNCTION.
C     ***DESCRIPTION
```

```
C
C     THE FOLLOWING DEFINITIONS ARE USED IN BSKIN:
C
C     DEFINITION 1
C     KI(0,X) = K-ZERO BESSEL FUNCTION.
C
C     DEFINITION 2
C     KI(N,X) = BICKLEY FUNCTION
C     =  INTEGRAL FROM X TO INFINITY OF KI(N-1,T)DT
C     FOR X .GE. 0 AND N = 1,2,...
C
C     ------------------------------------------------------------------
C     BSKIN COMPUTES SEQUENCES OF BICKLEY FUNCTIONS (REPEATED INTEGRALS
C     OF THE K0 BESSEL FUNCTION); I.E. FOR FIXED X AND N AND K=1,...,
C     BSKIN COMPUTES THE M-MEMBER SEQUENCE
C
C     Y(K) =        KI(N+K-1,X) FOR KODE=1
C     OR
C     Y(K) = EXP(X)*KI(N+K-1,X) FOR KODE=2,
C
C     FOR N.GE.0 AND X.GE.0 (N AND X CANNOT BE ZERO SIMULTANEOUSLY).
C
C     INPUT
C     X       - ARGUMENT, X .GE. 0.0E0
C     N       - ORDER OF FIRST MEMBER OF THE SEQUENCE N .GE. 0
C     KODE    - SELECTION PARAMETER
C     KODE = 1 RETURNS Y(K)=       KI(N+K-1,X), K=1,M
C     = 2 RETURNS Y(K)=EXP(X)*KI(N+K-1,X), K=1,M
C     M       - NUMBER OF MEMBERS IN THE SEQUENCE, M.GE.1
C
C     OUTPUT
C     Y       - A VECTOR OF DIMENSION AT LEAST M CONTAINING THE
C     SEQUENCE SELECTED BY KODE.
C     NZ      - UNDERFLOW FLAG
C     NZ = 0 MEANS COMPUTATION COMPLETED
C     = M MEANS AN EXPONENTIAL UNDERFLOW OCCURRED ON
C     KODE=1.  Y(K)=0.0E0, K=1,...,M IS RETURNED
C     IERR    - ERROR FLAG
C     IERR = 0, NORMAL RETURN, COMPUTATION COMPLETED.
C     = 1, INPUT ERROR,    NO COMPUTATION.
C     = 2, ERROR,          NO COMPUTATION.  THE
C     TERMINATION CONDITION WAS NOT MET.
C
C     THE NOMINAL COMPUTATIONAL ACCURACY IS THE MAXIMUM OF UNIT
C     ROUNDOFF (=R1MACH(4)) AND 1.0E-18 SINCE CRITICAL CONSTANTS
C     ARE GIVEN TO ONLY 18 DIGITS.
C
C     DBSKIN IS THE DOUBLE PRECISION VERSION OF BSKIN.
C     ***LONG DESCRIPTION
C
C     NUMERICAL RECURRENCE ON
C
C     (L-1)*KI(L,X) = X(KI(L-3,X) - KI(L-1,X)) + (L-2)*KI(L-2,X)
C
```

```
C     IS STABLE WHERE RECURRENCE IS CARRIED FORWARD OR BACKWARD
C     AWAY FROM INT(X+0.5).  THE POWER SERIES FOR INDICES 0,1 AND 2
C     ON 0.LE.X.LE. 2 STARTS A STABLE RECURRENCE FOR INDICES
C     GREATER THAN 2.  IF N IS SUFFICIENTLY LARGE (N.GT.NLIM), THE
C     UNIFORM ASYMPTOTIC EXPANSION FOR N TO INFINITY IS MORE
C     ECONOMICAL.  ON X.GT.2 THE RECURSION IS STARTED BY EVALUATING
C     THE UNIFORM EXPANSION FOR THE THREE MEMBERS WHOSE INDICES ARE
C     CLOSEST TO INT(X+0.5) WITHIN THE SET N,...,N+M-1.  FORWARD
C     RECURRENCE, BACKWARD RECURRENCE OR BOTH, COMPLETE THE
C     SEQUENCE DEPENDING ON THE RELATION OF INT(X+0.5) TO THE
C     INDICES N,...,N+M-1.
C     ***REFERENCES  UNIFORM ASYMPTOTIC EXPANSIONS FOR EXPONENTIAL INTEG
C     E(N,X) AND BICKLEY FUNCTIONS KI(N,X) BY D. E. AMOS,
C     ACM TRANS. MATH SOFTWARE, 1983.
C     A PORTABLE FORTRAN SUBROUTINE FOR THE BICKLEY FUNCTIONS
C     KI(N,X) BY D. E. AMOS, ACM TRANS. MATH SOFTWARE, 1983
C     ALGORITHM 609
C     ***ROUTINES CALLED  BKIAS,BKISR,EXINT,GAMRN,I1MACH,R1MACH
C     ***END PROLOGUE  BSKIN
      INTEGER I,ICASE,IERR,IL,I1M,K,KK,KODE,KTRMS,M,M3,N,NE,NFLG,NL,NLIM
     *  ,NN,NP,NS,NT,NZ
      INTEGER I1MACH
      REAL A,ENLIM,EXI,FN,GR,H,HN,HRTPI,SS,TOL,T1,T2,W,X,XLIM,XNLIM,XP,Y
     *  ,YS,YSS
      REAL GAMRN,R1MACH
      DIMENSION EXI(102),A(50),YS(3),YSS(3),H(31),Y(*)
C     ------------------------------------------------------------------
C     COEFFICIENTS IN SERIES OF EXPONENTIAL INTEGRALS
C     ------------------------------------------------------------------
      DATA A(1),A(2),A(3),A(4),A(5),A(6),A(7),A(8),A(9),A(10),A(11),A(12
     *  ),A(13),A(14),A(15),A(16),A(17),A(18),A(19),A(20),A(21),A(22),A(
     *  23),A(24) / 1.00000000000000000E+00,5.00000000000000000E-01,
     *  3.75000000000000000E-01,3.12500000000000000E-01,
     *  2.73437500000000000E-01,2.46093750000000000E-01,
     *  2.25585937500000000E-01,2.09472656250000000E-01,
     *  1.96380615234375000E-01,1.85470581054687500E-01,
     *  1.76197052001953125E-01,1.68188095092773438E-01,
     *  1.61180257797241211E-01,1.54981017112731934E-01,
     *  1.49445980787277222E-01,1.44464448094367981E-01,
     *  1.39949934091418982E-01,1.35833759559318423E-01,
     *  1.32060599571559578E-01,1.28585320635465905E-01,
     *  1.25370687619579257E-01,1.22385671247684513E-01,
     *  1.19604178719328047E-01,1.17004087877603524E-01 /
      DATA A(25),A(26),A(27),A(28),A(29),A(30),A(31),A(32),A(33),A(34),A
     *  (35),A(36),A(37),A(38),A(39),A(40),A(41),A(42),A(43),A(44),A(45)
     *  ,A(46),A(47),A(48) / 1.14566502713486784E-01,
     *  1.12275172659217048E-01,1.10116034723462874E-01,
     *  1.08076848895250599E-01,1.06146905164978267E-01,
     *  1.04316786110409676E-01,1.02578173008569515E-01,
     *  1.00923686347140974E-01,9.93467537479668965E-02,
     *  9.78414999033007314E-02,9.64026543164874854E-02,
     *  9.50254735405376642E-02,9.37056752969190855E-02,
     *  9.24393823875012600E-02,9.12230747245078224E-02,
```

```
     *  9.00535481254756708E-02,8.89278787739072249E-02,
     *  8.78433924473961612E-02,8.67976377754033498E-02,
     *  8.57883629175498224E-02,8.48134951571231199E-02,
     *  8.38711229887106408E-02,8.29594803475290034E-02,
     *  8.20769326842574183E-02 /
      DATA A(49),A(50) / 8.12219646354630702E-02,8.03931690779583449E-02
     *   /
C     ------------------------------------------------------------------
C     SQRT(PI)/2
C     ------------------------------------------------------------------
      DATA HRTPI / 8.86226925452758014E-01 /
C
C     ***FIRST EXECUTABLE STATEMENT  BSKIN
      IERR = 0
      NZ = 0
      IF (X.LT.0.0E0) IERR = 1
      IF (N.LT.0) IERR = 1
      IF (KODE.LT.1.OR.KODE.GT.2) IERR = 1
      IF (M.LT.1) IERR = 1
      IF (X.EQ.0.0E0.AND.N.EQ.0) IERR = 1
      IF (IERR.NE.0) RETURN
      IF (X.EQ.0.0E0) GO TO 240
      I1M = -I1MACH(12)
      T1 = 2.3026E0*R1MACH(5)*FLOAT(I1M)
      XLIM = T1-3.228086E0
      T2 = T1+FLOAT(N+M-1)
      IF (T2.GT.1000.0E0) XLIM = T1-0.5E0*(ALOG(T2)-0.451583E0)
      IF (X.GT.XLIM.AND.KODE.EQ.1) GO TO 260
      TOL = AMAX1(R1MACH(4),1.0E-18)
      I1M = I1MACH(11)
C     ------------------------------------------------------------------
C     LN(NLIM) = 0.125*LN(EPS),   NLIM = 2*KTRMS+N
C     ------------------------------------------------------------------
      XNLIM = 0.287823E0*FLOAT(I1M-1)*R1MACH(5)
      ENLIM = EXP(XNLIM)
      NLIM = INT(ENLIM)+2
      NLIM = MIN0(100,NLIM)
      NLIM = MAX0(20,NLIM)
      M3 = MIN0(M,3)
      NL = N+M-1
      IF (X.GT.2.0E0) GO TO 70
      IF (N.GT.NLIM) GO TO 220
C     ------------------------------------------------------------------
C     COMPUTATION BY SERIES FOR 0.LE.X.LE.2
C     ------------------------------------------------------------------
      NFLG = 0
      NN = N
      IF (NL.LE.2) GO TO 10
      M3 = 3
      NN = 0
      NFLG = 1
   10 CONTINUE
      XP = 1.0E0
      IF (KODE.EQ.2) XP = EXP(X)
```

```
      DO 30 I = 1, M3
        CALL BKISR (X,NN,W,IERR)
        IF (IERR.NE.0) RETURN
        W = W*XP
        IF (NN.LT.N) GO TO 20
        KK = NN-N+1
        Y(KK) = W
   20   CONTINUE
        YS(I) = W
        NN = NN+1
   30 CONTINUE
      IF (NFLG.EQ.0) RETURN
      NS = NN
      XP = 1.0E0
   40 CONTINUE
C     ------------------------------------------------------------------
C     FORWARD RECURSION SCALED BY EXP(X) ON ICASE=0,1,2
C     ------------------------------------------------------------------
      FN = FLOAT(NS-1)
      IL = NL-NS+1
      IF (IL.LE.0) RETURN
      DO 60 I = 1, IL
        T1 = YS(2)
        T2 = YS(3)
        YS(3) = (X*(YS(1)-YS(3))+(FN-1.0E0)*YS(2))/FN
        YS(2) = T2
        YS(1) = T1
        FN = FN+1.0E0
        IF (NS.LT.N) GO TO 50
        KK = NS-N+1
        Y(KK) = YS(3)*XP
   50   CONTINUE
        NS = NS+1
   60 CONTINUE
      RETURN
C     ------------------------------------------------------------------
C     COMPUTATION BY ASYMPTOTIC EXPANSION FOR X.GT.2
C     ------------------------------------------------------------------
   70 CONTINUE
      W = X+0.5E0
      NT = INT(W)
      IF (NL.GT.NT) GO TO 210
C     ------------------------------------------------------------------
C     CASE NL.LE.NT, ICASE=0
C     ------------------------------------------------------------------
      ICASE = 0
      NN = NL
      NFLG = MIN0(M-M3,1)
   80 CONTINUE
      KK = (NLIM-NN)/2
      KTRMS = MAX0(0,KK)
      NS = NN+1
      NP = NN-M3+1
```

```
      XP = 1.0E0
      IF (KODE.EQ.1) XP = EXP(-X)
      DO 90 I = 1, M3
        KK = I
        CALL BKIAS (X,NP,KTRMS,A,W,KK,NE,GR,H,IERR)
        IF (IERR.NE.0) RETURN
        YS(I) = W
        NP = NP+1
   90 CONTINUE
C     ------------------------------------------------------------------
C     SUM SERIES OF EXPONENTIAL INTEGRALS BACKWARD
C     ------------------------------------------------------------------
      IF (KTRMS.EQ.0) GO TO 100
      NE = KTRMS+KTRMS+1
      NP = NN-M3+2
      CALL EXINT (X,NP,2,NE,TOL,EXI,NZ,IERR)
      IF (NZ.NE.0) GO TO 260
      IF (IERR.EQ.2) RETURN
  100 CONTINUE
      DO 130 I = 1, M3
        SS = 0.0E0
        IF (KTRMS.EQ.0) GO TO 120
        KK = I+KTRMS+KTRMS-2
        IL = KTRMS
        DO 110 K = 1, KTRMS
          SS = SS+A(IL)*EXI(KK)
          KK = KK-2
          IL = IL-1
  110   CONTINUE
  120   CONTINUE
        YS(I) = YS(I)+SS
  130 CONTINUE
      IF (ICASE.EQ.1) GO TO 140
      IF (NFLG.NE.0) GO TO 160
  140 CONTINUE
      DO 150 I = 1, M3
        Y(I) = YS(I)*XP
  150 CONTINUE
      IF (ICASE.EQ.1.AND.NFLG.EQ.1) GO TO 40
      RETURN
  160 CONTINUE
C     ------------------------------------------------------------------
C     BACKWARD RECURSION SCALED BY EXP(X) ICASE=0,2
C     ------------------------------------------------------------------
      KK = NN-N+1
      K = M3
      DO 170 I = 1, M3
        Y(KK) = YS(K)*XP
        YSS(I) = YS(I)
        KK = KK-1
        K = K-1
  170 CONTINUE
      IL = KK
      IF (IL.LE.0) GO TO 190
```

```
      FN = FLOAT(NN-3)
      DO 180 I = 1, IL
        T1 = YS(2)
        T2 = YS(1)
        YS(1) = YS(2)+((FN+2.0E0)*YS(3)-(FN+1.0E0)*YS(1))/X
        YS(2) = T2
        YS(3) = T1
        Y(KK) = YS(1)*XP
        KK = KK-1
        FN = FN-1.0E0
  180 CONTINUE
  190 CONTINUE
      IF (ICASE.NE.2) RETURN
      DO 200 I = 1, M3
        YS(I) = YSS(I)
  200 CONTINUE
      GO TO 40
  210 CONTINUE
      IF (N.LT.NT) GO TO 230
C     ------------------------------------------------------------------
C     ICASE=1, NT.LE.N.LE.NL WITH FORWARD RECURSION
C     ------------------------------------------------------------------
  220 CONTINUE
      NN = N+M3-1
      NFLG = MIN0(M-M3,1)
      ICASE = 1
      GO TO 80
C     ------------------------------------------------------------------
C     ICASE=2, N.LT.NT.LT.NL WITH BOTH FORWARD AND BACKWARD RECURSION
C     ------------------------------------------------------------------
  230 CONTINUE
      NN = NT+1
      NFLG = MIN0(M-M3,1)
      ICASE = 2
      GO TO 80
C     ------------------------------------------------------------------
C     X=0 CASE
C     ------------------------------------------------------------------
  240 CONTINUE
      FN = FLOAT(N)
      HN = 0.5E0*FN
      GR = GAMRN(HN)
      Y(1) = HRTPI*GR
      IF (M.EQ.1) RETURN
      Y(2) = HRTPI/(HN*GR)
      IF (M.EQ.2) RETURN
      DO 250 K = 3, M
        Y(K) = FN*Y(K-2)/(FN+1.0E0)
        FN = FN+1.0E0
  250 CONTINUE
      RETURN
C     ------------------------------------------------------------------
```

```
C     UNDERFLOW ON KODE=1, X.GT.XLIM
C     ----------------------------------------------------------------
  260 CONTINUE
      NZ = M
      DO 270 I = 1, M
        Y(I) = 0.0E0
  270 CONTINUE
      RETURN
      END
*DECK,CSEVL
      FUNCTION CSEVL (X,CS,N)
C     ***BEGIN PROLOGUE  CSEVL
C     ***DATE WRITTEN   770401   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  C3A2
C     ***KEYWORDS  CHEBYSHEV,FNLIB,SPECIAL FUNCTION
C     ***AUTHOR  FULLERTON, W., (LANL)
C     ***PURPOSE  EVALUATE THE N-TERM CHEBYSHEV SERIES CS AT X.
C     ***DESCRIPTION
C
C     EVALUATE THE N-TERM CHEBYSHEV SERIES CS AT X.  ADAPTED FROM
C     R. BROUCKE, ALGORITHM 446, C.A.C.M., 16, 254 (1973). ALSO SEE FOX
C     AND PARKER, CHEBYSHEV POLYNOMIALS IN NUMERICAL ANALYSIS, OXFORD PR
C     PAGE 56.
C
C     INPUT ARGUMENTS --
C     X    VALUE AT WHICH THE SERIES IS TO BE EVALUATED.
C     CS   ARRAY OF N TERMS OF A CHEBYSHEV SERIES.  IN EVAL-
C     UATING CS, ONLY HALF THE FIRST COEFFICIENT IS SUMMED.
C     N    NUMBER OF TERMS IN ARRAY CS.
C     ***REFERENCES  (NONE)
C     ***ROUTINES CALLED  XERROR
C     ***END PROLOGUE  CSEVL
C
      DIMENSION CS(1)
C     ***FIRST EXECUTABLE STATEMENT  CSEVL
      IF (N.LT.1) CALL XERR ('CSEVL   NUMBER OF TERMS LE 0')
      IF (N.GT.1000) CALL XERR ('CSEVL   NUMBER OF TERMS GT 1000')
      IF (X.LT.-1.0.OR.X.GT.1.0) CALL XERR ('CSEVL   X OUTSIDE (-1,+1)')
C
      B1 = 0.
      B0 = 0.
      TWOX = 2.*X
      DO 10 I = 1, N
        B2 = B1
        B1 = B0
        NI = N+1-I
        B0 = TWOX*B1-B2+CS(NI)
   10 CONTINUE
C
      CSEVL = 0.5*(B0-B2)
C
      RETURN
      END
```

```
*DECK,E1
      FUNCTION E1(X)
C***BEGIN PROLOGUE  E1
C***DATE WRITTEN   770701   (YYMMDD)
C***REVISION DATE  861211   (YYMMDD)
C***CATEGORY NO.  C5
C***KEYWORDS  LIBRARY=SLATEC(FNLIB),TYPE=SINGLE PRECISION(E1-S DE1-D),
C             E1 FUNCTION,EXPONENTIAL INTEGRAL,SPECIAL FUNCTIONS
C***AUTHOR  FULLERTON, W., (LANL)
C***PURPOSE  Computes the exponential integral E1(X).
C***DESCRIPTION
C
C E1(X) calculates the single precision exponential integral
C for positive, single precision argument X.  It returns the
C Cauchy principal value for negative X.  If principal values are
C used everywhere, then for all X
C
C     EI(X) = -E1(-X)
C or
C     E1(X) = -EI(-X).
C
C
C Series for AE11       on the interval -1.00000D-01 to  0.
C                                       with weighted error   1.76E-17
C                                        log weighted error  16.75
C                              significant figures required  15.70
C                                   decimal places required  17.55
C
C
C Series for AE12       on the interval -2.50000D-01 to -1.00000D-01
C                                       with weighted error   5.83E-17
C                                        log weighted error  16.23
C                              significant figures required  15.76
C                                   decimal places required  16.93
C
C
C Series for E11        on the interval -4.00000D+00 to -1.00000D+00
C                                       with weighted error   1.08E-18
C                                        log weighted error  17.97
C                              significant figures required  19.02
C                                   decimal places required  18.61
C
C
C Series for E12        on the interval -1.00000D+00 to  1.00000D+00
C                                       with weighted error   3.15E-18
C                                        log weighted error  17.50
C                         approx significant figures required  15.8
C                                   decimal places required  18.10
C
C
C Series for AE13       on the interval  2.50000D-01 to  1.00000D+00
C                                       with weighted error   2.34E-17
C                                        log weighted error  16.63
C                              significant figures required  16.14
C                                   decimal places required  17.33
```

```
C
C
C Series for AE14       on the interval  0.          to  2.50000D-01
C                                       with weighted error   5.41E-17
C                                        log weighted error  16.27
C                              significant figures required  15.38
C                                   decimal places required  16.97
C***REFERENCES  (NONE)
C***ROUTINES CALLED  CSEVL,INITS,R1MACH,XERROR
C***END PROLOGUE  E1
      DIMENSION AE11CS(39), AE12CS(25), E11CS(19), E12CS(16),
     1  AE13CS(25), AE14CS(26)
      SAVE AE11CS, AE12CS, E11 CS, E12 CS, AE13CS, AE14CS,
     1 NTAE11, NTAE12, NTE11, NTE12, NTAE13, NTAE14, XMAX
      DATA AE11CS( 1) /    .1215032397 1606579E0 /
      DATA AE11CS( 2) /   -.0650887785 13550150E0 /
      DATA AE11CS( 3) /    .0048976513 57459670E0 /
      DATA AE11CS( 4) /   -.0006492378 43027216E0 /
      DATA AE11CS( 5) /    .0000938404 34587471E0 /
      DATA AE11CS( 6) /    .0000004202 36380882E0 /
      DATA AE11CS( 7) /   -.0000081133 74735904E0 /
      DATA AE11CS( 8) /    .0000028042 47688663E0 /
      DATA AE11CS( 9) /    .0000000564 87164441E0 /
      DATA AE11CS(10) /   -.0000003448 09174450E0 /
      DATA AE11CS(11) /    .0000000582 09273578E0 /
      DATA AE11CS(12) /    .0000000387 11426349E0 /
      DATA AE11CS(13) /   -.0000000124 53235014E0 /
      DATA AE11CS(14) /   -.0000000051 18504888E0 /
      DATA AE11CS(15) /    .0000000021 48771527E0 /
      DATA AE11CS(16) /    .0000000008 68459898E0 /
      DATA AE11CS(17) /   -.0000000003 43650105E0 /
      DATA AE11CS(18) /   -.0000000001 79796603E0 /
      DATA AE11CS(19) /    .0000000000 47442060E0 /
      DATA AE11CS(20) /    .0000000000 40423282E0 /
      DATA AE11CS(21) /   -.0000000000 03543928E0 /
      DATA AE11CS(22) /   -.0000000000 08853444E0 /
      DATA AE11CS(23) /   -.0000000000 00960151E0 /
      DATA AE11CS(24) /    .0000000000 01692921E0 /
      DATA AE11CS(25) /    .0000000000 00607990E0 /
      DATA AE11CS(26) /   -.0000000000 00224338E0 /
      DATA AE11CS(27) /   -.0000000000 00200327E0 /
      DATA AE11CS(28) /   -.0000000000 00006246E0 /
      DATA AE11CS(29) /    .0000000000 00045571E0 /
      DATA AE11CS(30) /    .0000000000 00016383E0 /
      DATA AE11CS(31) /   -.0000000000 00005561E0 /
      DATA AE11CS(32) /   -.0000000000 00006074E0 /
      DATA AE11CS(33) /   -.0000000000 00000862E0 /
      DATA AE11CS(34) /    .0000000000 00001223E0 /
      DATA AE11CS(35) /    .0000000000 00000716E0 /
      DATA AE11CS(36) /   -.0000000000 00000024E0 /
      DATA AE11CS(37) /   -.0000000000 00000201E0 /
      DATA AE11CS(38) /   -.0000000000 00000082E0 /
      DATA AE11CS(39) /    .0000000000 00000017E0 /
```

```
DATA AE12CS( 1) /    .5824174951 3472674E0 /
DATA AE12CS( 2) /   -.1583488509 0578275E0 /
DATA AE12CS( 3) /   -.0067642755 90323141E0 /
DATA AE12CS( 4) /    .0051258439 50185725E0 /
DATA AE12CS( 5) /    .0004352324 92169391E0 /
DATA AE12CS( 6) /   -.0001436133 66305483E0 /
DATA AE12CS( 7) /   -.0000418013 20556301E0 /
DATA AE12CS( 8) /   -.0000027133 95758640E0 /
DATA AE12CS( 9) /    .0000011513 81913647E0 /
DATA AE12CS(10) /    .0000004206 50022012E0 /
DATA AE12CS(11) /    .0000000665 81901391E0 /
DATA AE12CS(12) /    .0000000006 62143777E0 /
DATA AE12CS(13) /   -.0000000028 44104870E0 /
DATA AE12CS(14) /   -.0000000009 40724197E0 /
DATA AE12CS(15) /   -.0000000001 77476602E0 /
DATA AE12CS(16) /   -.0000000000 15830222E0 /
DATA AE12CS(17) /    .0000000000 02905732E0 /
DATA AE12CS(18) /    .0000000000 01769356E0 /
DATA AE12CS(19) /    .0000000000 00492735E0 /
DATA AE12CS(20) /    .0000000000 00093709E0 /
DATA AE12CS(21) /    .0000000000 00010707E0 /
DATA AE12CS(22) /   -.0000000000 00000537E0 /
DATA AE12CS(23) /   -.0000000000 00000716E0 /
DATA AE12CS(24) /   -.0000000000 00000244E0 /
DATA AE12CS(25) /   -.0000000000 00000058E0 /
DATA E11 CS( 1) / -16.1134616555 71494026E0 /
DATA E11 CS( 2) /   7.7940727787 426802769E0 /
DATA E11 CS( 3) /  -1.9554058188 631419507E0 /
DATA E11 CS( 4) /    .3733729386 6277945612E0 /
DATA E11 CS( 5) /   -.0569250319 1092901938E0 /
DATA E11 CS( 6) /    .0072110777 6966009185E0 /
DATA E11 CS( 7) /   -.0007810490 1449841593E0 /
DATA E11 CS( 8) /    .0000738809 3356262168E0 /
DATA E11 CS( 9) /   -.0000062028 6187580820E0 /
DATA E11 CS(10) /    .0000004681 6002303176E0 /
DATA E11 CS(11) /   -.0000000320 9288853329E0 /
DATA E11 CS(12) /    .0000000020 1519974874E0 /
DATA E11 CS(13) /   -.0000000001 1673686816E0 /
DATA E11 CS(14) /    .0000000000 0627627066E0 /
DATA E11 CS(15) /   -.0000000000 0031481541E0 /
DATA E11 CS(16) /    .0000000000 0001479904E0 /
DATA E11 CS(17) /   -.0000000000 0000065457E0 /
DATA E11 CS(18) /    .0000000000 0000002733E0 /
DATA E11 CS(19) /   -.0000000000 0000000108E0 /
DATA E12 CS( 1) /  -0.0373902147 92202795E0 /
DATA E12 CS( 2) /   0.0427239860 62209577E0 /
DATA E12 CS( 3) /   -.1303182079 849700544E0 /
DATA E12 CS( 4) /    .0144191240 2469889073E0 /
DATA E12 CS( 5) /   -.0013461707 8051068022E0 /
DATA E12 CS( 6) /    .0001073102 9253063780E0 /
DATA E12 CS( 7) /   -.0000074299 9951611943E0 /
DATA E12 CS( 8) /    .0000004537 7325690753E0 /
DATA E12 CS( 9) /   -.0000000247 6417211390E0 /
```

```
DATA E12 CS(10) /    .0000000012 2076581374E0 /
DATA E12 CS(11) /   -.0000000000 5485141480E0 /
DATA E12 CS(12) /    .0000000000 0226362142E0 /
DATA E12 CS(13) /   -.0000000000 0008635897E0 /
DATA E12 CS(14) /    .0000000000 0000306291E0 /
DATA E12 CS(15) /   -.0000000000 0000010148E0 /
DATA E12 CS(16) /    .0000000000 0000000315E0 /
DATA AE13CS( 1) /   -.6057732466 4060346E0 /
DATA AE13CS( 2) /   -.1125352434 8366090E0 /
DATA AE13CS( 3) /    .0134322662 47902779E0 /
DATA AE13CS( 4) /   -.0019268451 87381145E0 /
DATA AE13CS( 5) /    .0003091183 37720603E0 /
DATA AE13CS( 6) /   -.0000535641 32129618E0 /
DATA AE13CS( 7) /    .0000098278 12880247E0 /
DATA AE13CS( 8) /   -.0000018853 68984916E0 /
DATA AE13CS( 9) /    .0000003749 43193568E0 /
DATA AE13CS(10) /   -.0000000768 23455870E0 /
DATA AE13CS(11) /    .0000000161 43270567E0 /
DATA AE13CS(12) /   -.0000000034 66802211E0 /
DATA AE13CS(13) /    .0000000007 58754209E0 /
DATA AE13CS(14) /   -.0000000001 68864333E0 /
DATA AE13CS(15) /    .0000000000 38145706E0 /
DATA AE13CS(16) /   -.0000000000 08733026E0 /
DATA AE13CS(17) /    .0000000000 02023672E0 /
DATA AE13CS(18) /   -.0000000000 00474132E0 /
DATA AE13CS(19) /    .0000000000 00112211E0 /
DATA AE13CS(20) /   -.0000000000 00026804E0 /
DATA AE13CS(21) /    .0000000000 00006457E0 /
DATA AE13CS(22) /   -.0000000000 00001568E0 /
DATA AE13CS(23) /    .0000000000 00000383E0 /
DATA AE13CS(24) /   -.0000000000 00000094E0 /
DATA AE13CS(25) /    .0000000000 00000023E0 /
DATA AE14CS( 1) /   -.1892918000 753017E0 /
DATA AE14CS( 2) /   -.0864811785 5259871E0 /
DATA AE14CS( 3) /    .0072241015 4374659E0 /
DATA AE14CS( 4) /   -.0008097559 4575573E0 /
DATA AE14CS( 5) /    .0001099913 4432661E0 /
DATA AE14CS( 6) /   -.0000171733 2998937E0 /
DATA AE14CS( 7) /    .0000029856 2751447E0 /
DATA AE14CS( 8) /   -.0000005659 6491457E0 /
DATA AE14CS( 9) /    .0000001152 6808397E0 /
DATA AE14CS(10) /   -.0000000249 5030440E0 /
DATA AE14CS(11) /    .0000000056 9232420E0 /
DATA AE14CS(12) /   -.0000000013 5995766E0 /
DATA AE14CS(13) /    .0000000003 3846628E0 /
DATA AE14CS(14) /   -.0000000000 8737853E0 /
DATA AE14CS(15) /    .0000000000 2331588E0 /
DATA AE14CS(16) /   -.0000000000 0641148E0 /
DATA AE14CS(17) /    .0000000000 0181224E0 /
DATA AE14CS(18) /   -.0000000000 0052538E0 /
DATA AE14CS(19) /    .0000000000 0015592E0 /
DATA AE14CS(20) /   -.0000000000 0004729E0 /
DATA AE14CS(21) /    .0000000000 0001463E0 /
```

```
      DATA AE14CS(22) /   -.0000000000 0000461E0 /
      DATA AE14CS(23) /    .0000000000 0000148E0 /
      DATA AE14CS(24) /   -.0000000000 0000048E0 /
      DATA AE14CS(25) /    .0000000000 0000016E0 /
      DATA AE14CS(26) /   -.0000000000 0000005E0 /
      DATA NTAE11, NTAE12, NTE11, NTE12, NTAE13, NTAE14 /6*0/
      DATA XMAX /0./
C***FIRST EXECUTABLE STATEMENT  E1
      IF (NTAE11.NE.0) GO TO 10
      ETA = 0.1*R1MACH(3)
      NTAE11 = INITS (AE11CS, 39, ETA)
      NTAE12 = INITS (AE12CS, 25, ETA)
      NTE11 = INITS (E11CS, 19, ETA)
      NTE12 = INITS (E12CS, 16, ETA)
      NTAE13 = INITS (AE13CS, 25, ETA)
      NTAE14 = INITS (AE14CS, 26, ETA)
C
      XMAX = -ALOG (R1MACH(1))
      XMAX = XMAX - ALOG(XMAX)
C
 10   IF (X.GT.(-10.)) GO TO 20
C
C E1(X) = -EI(-X) FOR X .LE. -10.
C
      E1 = EXP(-X)/X * (1.+CSEVL (20./X+1., AE11CS, NTAE11))
      RETURN
C
 20   IF (X.GT.(-4.0)) GO TO 30
C
C E1(X) = -EI(-X) FOR -10. .LT. X .LE. -4.
C
      E1 = EXP(-X)/X * (1.+CSEVL ((40./X+7.)/3., AE12CS, NTAE12))
      RETURN
C
 30   IF (X.GT.(-1.0)) GO TO 40
C
C E1(X) = -EI(-X) FOR -4. .LT. X .LE. -1.
C
      E1 = -ALOG(ABS(X)) + CSEVL ((2.*X+5.)/3., E11CS, NTE11)
      RETURN
C
 40   IF (X.GT.1.) GO TO 50
      IF (X.EQ.0.) CALL XERR ( 'E1      X IS 0')
C
C E1(X) = -EI(-X) FOR -1. .LT. X .LE. 1.,  X .NE. 0.
C
      E1 = (-ALOG(ABS(X)) - 0.6875 + X) + CSEVL (X, E12CS, NTE12)
      RETURN
C
 50   IF (X.GT.4.) GO TO 60
C
C E1(X) = -EI(-X) FOR 1. .LT. X .LE. 4.
C
```

```
      E1 = EXP(-X)/X * (1.+CSEVL ((8./X-5.)/3., AE13CS, NTAE13))
      RETURN
C
 60   IF (X.GT.XMAX) GO TO 70
C
C E1(X) = -EI(-X) FOR 4. .LT. X .LE. XMAX
C
      E1 = EXP(-X)/X * (1. + CSEVL (8./X-1., AE14CS, NTAE14))
      RETURN
C
C E1(X) = -EI(-X) FOR X .GT. XMAX
C
 70   CALL XERR ( 'E1      X SO BIG E1 UNDERFLOWS')
      E1 = 0.
      RETURN
C
      END
*DECK EXINT
      SUBROUTINE EXINT(X,N,KODE,M,TOL,EN,NZ,IERR)
C***BEGIN PROLOGUE  EXINT
C***DATE WRITTEN   800501   (YYMMDD)
C***REVISION DATE  880811   (YYMMDD)
C***CATEGORY NO.  C5
C***KEYWORDS  LIBRARY=SLATEC,TYPE=SINGLE PRECISION(EXINT-S DEXINT-D),
C             EXPONENTIAL INTEGRAL,SPECIAL FUNCTIONS
C***AUTHOR  AMOS, D. E., (SNLA)
C***PURPOSE  EXINT computes M member sequences of exponential integrals
C            E(N+K,X), K=0,1,...,M-1 for N .GE. 1 and X .GE. 0.
C***DESCRIPTION
C
C         EXINT computes M member sequences of exponential integrals
C         E(N+K,X), K=0,1,...,M-1 for N .GE. 1 and X .GE. 0.  The
C         exponential integral is defined by
C
C         E(N,X)=integral on (1,infinity) of EXP(-XT)/T**N
C
C         where X=0.0 and N=1 cannot occur simultaneously.  Formulas
C         and notation are found in the NBS Handbook of Mathematical
C         Functions (ref. 1).
C
C         Input
C           X       X .GT. 0.0 for N=1 and  X .GE. 0.0 for N .GE. 2
C           N       order of the first member of the sequence, N .GE. 1
C                   (X=0.0 and N=1 is an error)
C           KODE    a selection parameter for scaled values
C                   KODE=1   returns        E(N+K,X), K=0,1,...,M-1.
C                       =2   returns EXP(X)*E(N+K,X), K=0,1,...,M-1.
C           M       number of exponential integrals in the sequence,
C                   M .GE. 1
C           TOL     relative accuracy wanted, ETOL .LE. TOL .LE. 0.1
C                   ETOL = single precision unit roundoff = R1MACH(4)
C
```

```
C         Output
C           EN      a vector of dimension at least M containing values
C                   EN(K) = E(N+K-1,X) or EXP(X)*E(N+K-1,X), K=1,M
C                   depending on KODE
C           NZ      underflow indicator
C                   NZ=0    a normal return
C                   NZ=M    X exceeds XLIM and an underflow occurs.
C                             EN(K)=0.0E0 , K=1,M returned on KODE=1
C           IERR    error flag
C                   IERR=0, normal return, computation completed
C                   IERR=1, input error,   no computation
C                   IERR=2, error,         no computation
C                           algoithm termination condition not met
C***REFERENCES  HANDBOOK OF MATHEMATICAL FUNCTIONS BY M. ABRAMOWITZ
C                 AND I. A. STEGUN, NBS AMS SERIES 55, U.S. DEPT. OF
C                 COMMERCE, 1955.
C               COMPUTATION OF EXPONENTIAL INTEGRALS BY D. E. AMOS,
C                 ACM TRANS. MATH SOFTWARE, 6, 1980,PP. 365-377, 420-428
C***ROUTINES CALLED  I1MACH,R1MACH,PSIXN
C***END PROLOGUE  EXINT
C
      REAL             A,AA,AAMS,AH,AK,AT,B,BK,BT,CC,CNORM,CT,EM,EMX,EN,
     1                 ETOL,FNM,FX,PT,P1,P2,S,TOL,TX,X,XCUT,XLIM,XTOL,Y,
     2                 YT,Y1,Y2
      REAL             R1MACH,PSIXN
      INTEGER I,IC,ICASE,ICT,IERR,IK,IND,IX,I1M,JSET,K,KK,KN,KODE,KS,M,
     1        ML,MU,N,ND,NM,NZ
      INTEGER I1MACH
      DIMENSION EN(*), A(99), B(99), Y(2)
C***FIRST EXECUTABLE STATEMENT  EXINT
      IERR=0
      NZ=0
      ETOL = AMAX1(R1MACH(4),0.5E-18)
      IF (X.LT.0.0E0) IERR=1
      IF (N.LT.1) IERR=1
      IF (KODE.LT.1 .OR. KODE.GT.2) IERR=1
      IF (M.LT.1) IERR=1
      IF (TOL.LT.ETOL .OR. TOL.GT.0.1E0) IERR=1
      IF (X.EQ.0.0E0 .AND. N.EQ.1) IERR=1
      IF (IERR.NE.0) RETURN
      I1M = -I1MACH(12)
      PT = 2.3026E0*R1MACH(5)*FLOAT(I1M)
      XLIM = PT - 6.907755E0
      BT = PT + FLOAT(N+M-1)
      IF (BT.GT.1000.0E0) XLIM = PT - ALOG(BT)
C
      XCUT = 2.0E0
      IF (ETOL.GT.2.0E-7) XCUT = 1.0E0
      IF (X.GT.XCUT) GO TO 100
      IF (X.EQ.0.0E0 .AND. N.GT.1) GO TO 80
C-----------------------------------------------------------------------
C     SERIES FOR E(N,X) FOR X.LE.XCUT
C-----------------------------------------------------------------------
```

```
      TX = X + 0.5E0
      IX = INT(TX)
C-----------------------------------------------------------------------
C     ICASE=1 MEANS INTEGER CLOSEST TO X IS 2 AND N=1
C     ICASE=2 MEANS INTEGER CLOSEST TO X IS 0,1, OR 2 AND N.GE.2
C-----------------------------------------------------------------------
      ICASE = 2
      IF (IX.GT.N) ICASE = 1
      NM = N - ICASE + 1
      ND = NM + 1
      IND = 3 - ICASE
      MU = M - IND
      ML = 1
      KS = ND
      FNM = FLOAT(NM)
      S = 0.0E0
      XTOL = 3.0E0*TOL
      IF (ND.EQ.1) GO TO 10
      XTOL = 0.3333E0*TOL
      S = 1.0E0/FNM
   10 CONTINUE
      AA = 1.0E0
      AK = 1.0E0
      IC = 35
      IF (X.LT.ETOL) IC = 1
      DO 50 I=1,IC
        AA = -AA*X/AK
        IF (I.EQ.NM) GO TO 30
        S = S - AA/(AK-FNM)
        IF (ABS(AA).LE.XTOL*ABS(S)) GO TO 20
        AK = AK + 1.0E0
        GO TO 50
   20   CONTINUE
        IF (I.LT.2) GO TO 40
        IF (ND-2.GT.I .OR. I.GT.ND-1) GO TO 60
        AK = AK + 1.0E0
        GO TO 50
   30   S = S + AA*(-ALOG(X)+PSIXN(ND))
        XTOL = 3.0E0*TOL
   40   AK = AK + 1.0E0
   50 CONTINUE
      IF (IC.NE.1) GO TO 340
   60 IF (ND.EQ.1) S = S + (-ALOG(X)+PSIXN(1))
      IF (KODE.EQ.2) S = S*EXP(X)
      EN(1) = S
      EMX = 1.0E0
      IF (M.EQ.1) GO TO 70
      EN(IND) = S
      AA = FLOAT(KS)
      IF (KODE.EQ.1) EMX = EXP(-X)
      GO TO (220, 240), ICASE
   70 IF (ICASE.EQ.2) RETURN
      IF (KODE.EQ.1) EMX = EXP(-X)
```

```
      EN(1) = (EMX-S)/X
      RETURN
   80 CONTINUE
      DO 90 I=1,M
        EN(I) = 1.0E0/FLOAT(N+I-2)
   90 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     BACKWARD RECURSIVE MILLER ALGORITHM FOR
C              E(N,X)=EXP(-X)*(X**(N-1))*U(N,N,X)
C     WITH RECURSION AWAY FROM N=INTEGER CLOSEST TO X.
C     U(A,B,X) IS THE SECOND CONFLUENT HYPERGEOMETRIC FUNCTION
C-----------------------------------------------------------------------
  100 CONTINUE
      EMX = 1.0E0
      IF (KODE.EQ.2) GO TO 130
      IF (X.LE.XLIM) GO TO 120
      NZ = M
      DO 110 I=1,M
        EN(I) = 0.0E0
  110 CONTINUE
      RETURN
  120 EMX = EXP(-X)
  130 CONTINUE
      IX = INT(X+0.5E0)
      KN = N + M - 1
      IF (KN.LE.IX) GO TO 140
      IF (N.LT.IX .AND. IX.LT.KN) GO TO 170
      IF (N.GE.IX) GO TO 160
      GO TO 340
  140 ICASE = 1
      KS = KN
      ML = M - 1
      MU = -1
      IND = M
      IF (KN.GT.1) GO TO 180
  150 KS = 2
      ICASE = 3
      GO TO 180
  160 ICASE = 2
      IND = 1
      KS = N
      MU = M - 1
      IF (N.GT.1) GO TO 180
      IF (KN.EQ.1) GO TO 150
      IX = 2
  170 ICASE = 1
      KS = IX
      ML = IX - N
      IND = ML + 1
      MU = KN - IX
  180 CONTINUE
      IK = KS/2
```

```
      AH = FLOAT(IK)
      JSET = 1 + KS - (IK+IK)
C-----------------------------------------------------------------------
C     START COMPUTATION FOR
C               EN(IND) = C*U( A , A ,X)    JSET=1
C               EN(IND) = C*U(A+1,A+1,X)    JSET=2
C     FOR AN EVEN INTEGER A.
C-----------------------------------------------------------------------
      IC = 0
      AA = AH + AH
      AAMS = AA - 1.0E0
      AAMS = AAMS*AAMS
      TX = X + X
      FX = TX + TX
      AK = AH
      XTOL = TOL
      IF (TOL.LE.1.0E-3) XTOL = 20.0E0*TOL
      CT = AAMS + FX*AH
      EM = (AH+1.0E0)/((X+AA)*XTOL*SQRT(CT))
      BK = AA
      CC = AH*AH
C-----------------------------------------------------------------------
C     FORWARD RECURSION FOR P(IC),P(IC+1) AND INDEX IC FOR BACKWARD
C     RECURSION
C-----------------------------------------------------------------------
      P1 = 0.0E0
      P2 = 1.0E0
  190 CONTINUE
      IF (IC.EQ.99) GO TO 340
      IC = IC + 1
      AK = AK + 1.0E0
      AT = BK/(BK+AK+CC+FLOAT(IC))
      BK = BK + AK + AK
      A(IC) = AT
      BT = (AK+AK+X)/(AK+1.0E0)
      B(IC) = BT
      PT = P2
      P2 = BT*P2 - AT*P1
      P1 = PT
      CT = CT + FX
      EM = EM*AT*(1.0E0-TX/CT)
      IF (EM*(AK+1.0E0).GT.P1*P1) GO TO 190
      ICT = IC
      KK = IC + 1
      BT = TX/(CT+FX)
      Y2 = (BK/(BK+CC+FLOAT(KK)))*(P1/P2)*(1.0E0-BT+0.375E0*BT*BT)
      Y1 = 1.0E0
C-----------------------------------------------------------------------
C     BACKWARD RECURRENCE FOR
C               Y1=              C*U( A ,A,X)
C               Y2= C*(A/(1+A/2))*U(A+1,A,X)
C-----------------------------------------------------------------------
```

```
      DO 200 K=1,ICT
        KK = KK - 1
        YT = Y1
        Y1 = (B(KK)*Y1-Y2)/A(KK)
        Y2 = YT
  200 CONTINUE
C-----------------------------------------------------------------------
C     THE CONTIGUOUS RELATION
C               X*U(B,C+1,X)=(C-B)*U(B,C,X)+U(B-1,C,X)
C     WITH  B=A+1 , C=A IS USED FOR
C               Y(2) = C * U(A+1,A+1,X)
C     X IS INCORPORATED INTO THE NORMALIZING RELATION
C-----------------------------------------------------------------------
      PT = Y2/Y1
      CNORM = 1.0E0 - PT*(AH+1.0E0)/AA
      Y(1) = 1.0E0/(CNORM*AA+X)
      Y(2) = CNORM*Y(1)
      IF (ICASE.EQ.3) GO TO 210
      EN(IND) = EMX*Y(JSET)
      IF (M.EQ.1) RETURN
      AA = FLOAT(KS)
      GO TO (220, 240), ICASE
C-----------------------------------------------------------------------
C     RECURSION SECTION  N*E(N+1,X) + X*E(N,X)=EMX
C-----------------------------------------------------------------------
  210 EN(1) = EMX*(1.0E0-Y(1))/X
      RETURN
  220 K = IND - 1
      DO 230 I=1,ML
        AA = AA - 1.0E0
        EN(K) = (EMX-AA*EN(K+1))/X
        K = K - 1
  230 CONTINUE
      IF (MU.LE.0) RETURN
      AA = FLOAT(KS)
  240 K = IND
      DO 250 I=1,MU
        EN(K+1) = (EMX-X*EN(K))/AA
        AA = AA + 1.0E0
        K = K + 1
  250 CONTINUE
      RETURN
  340 CONTINUE
      IERR=2
      RETURN
      END
*DECK,GAMRN
      REAL FUNCTION GAMRN (X)
C     ***BEGIN PROLOGUE  GAMRN
C     ***REFER TO  BSKIN
C     ***ROUTINES CALLED  I1MACH,R1MACH
C     ***DESCRIPTION
C
```

```
C     REFERENCES
C     THE SPECIAL FUNCTIONS AND THEIR APPROXIMATIONS, VOL. 1 BY
C     Y.L. LUKE, MATH IN SCI. AND ENG. SERIES 53, ACADEMIC PRESS,
C     NEW YORK, 1969, PP 34-35.
C
C     ABSTRACT
C     GAMRN COMPUTES THE GAMMA FUNCTION RATIO GAMMA(X)/GAMMA(X+0.5)
C     FOR REAL X.GT.0. IF X.GE.XMIN, AN ASYMPTOTIC EXPANSION IS
C     EVALUATED. IF X.LT.XMIN, AN INTEGER IS ADDED TO X TO FORM A
C     NEW VALUE OF X.GE.XMIN AND THE ASYMPTOTIC EXPANSION IS EVAL-
C     UATED FOR THIS NEW VALUE OF X. SUCCESSIVE APPLICATION OF THE
C     RECURRENCE RELATION
C
C     W(X)=W(X+1)*(1+0.5/X)
C
C     REDUCES THE ARGUMENT TO ITS ORIGINAL VALUE. XMIN AND COMP-
C     UTATIONAL TOLERANCES ARE COMPUTED AS A FUNCTION OF THE NUMBER
C     OF DIGITS CARRIED IN A WORD BY CALLS TO I1MACH AND R1MACH.
C     HOWEVER, THE COMPUTATIONAL ACCURACY IS LIMITED TO THE MAX-
C     IMUM OF UNIT ROUNDOFF (=R1MACH(4)) AND 1.0E-18 SINCE CRITICAL
C     CONSTANTS ARE GIVEN TO ONLY 18 DIGITS.
C
C     INPUT
C     X       - ARGUMENT, X.GT.0.0
C
C     OUTPUT
C     GAMRN   - RATIO  GAMMA(X)/GAMMA(X+0.5)
C     ***END PROLOGUE  GAMRN
      INTEGER I,I1M11,K,MX,NX
      INTEGER I1MACH
      REAL FLN,GR,RLN,S,TOL,TRM,X,XDMY,XINC,XM,XMIN,XP,XSQ
      REAL R1MACH
      DIMENSION GR(12)
C
      DATA GR(1),GR(2),GR(3),GR(4),GR(5),GR(6),GR(7),GR(8),GR(9),GR(10),
     *  GR(11),GR(12) / 1.00000000000000000E+00,-1.56250000000000000E-02
     *  ,2.56347656250000000E-03,-1.27983093261718750E-03,
     *  1.34351104497909546E-03,-2.43289663922041655E-03,
     *  6.75423753364157164E-03,-2.66369606131178216E-02,
     *  1.41527455519564332E-01,-9.74384543032201613E-01,
     *  8.43686251229783675E+00,-8.97258321640552515E+01 /
C
C     ***FIRST EXECUTABLE STATEMENT  GAMRN
      NX = INT(X)
      TOL = AMAX1(R1MACH(4),1.0E-18)
      I1M11 = I1MACH(11)
      RLN = R1MACH(5)*FLOAT(I1M11)
      FLN = AMIN1(RLN,20.0E0)
      FLN = AMAX1(FLN,3.0E0)
      FLN = FLN-3.0E0
      XM = 2.0E0+FLN*(0.2366E0+0.01723E0*FLN)
      MX = INT(XM)+1
      XMIN = FLOAT(MX)
```

```
      XDMY = X-0.25E0
      XINC = 0.0E0
      IF (X.GE.XMIN) GO TO 10
      XINC = XMIN-FLOAT(NX)
      XDMY = XDMY+XINC
   10 CONTINUE
      S = 1.0E0
      IF (XDMY*TOL.GT.1.0E0) GO TO 30
      XSQ = 1.0E0/(XDMY*XDMY)
      XP = XSQ
      DO 20 K = 2, 12
        TRM = GR(K)*XP
        IF (ABS(TRM).LT.TOL) GO TO 30
        S = S+TRM
        XP = XP*XSQ
   20 CONTINUE
   30 CONTINUE
      S = S/SQRT(XDMY)
      IF (XINC.NE.0.0E0) GO TO 40
      GAMRN = S
      RETURN
   40 CONTINUE
      NX = INT(XINC)
      XP = 0.0E0
      DO 50 I = 1, NX
        S = S*(1.0E0+0.5E0/(X+XP))
        XP = XP+1.0E0
   50 CONTINUE
      GAMRN = S
      RETURN
      END
*DECK GAUS8
      SUBROUTINE GAUS8(FUN,A,B,ERR,ANS,IERR)
C***BEGIN PROLOGUE  GAUS8
C***DATE WRITTEN   810223   (YYMMDD)
C***REVISION DATE  890206   (YYMMDD)
C***CATEGORY NO.  H2A1A1
C***KEYWORDS  LIBRARY=SLATEC,TYPE=SINGLE PRECISION(GAUS8-S DGAUS8-D),
C             ADAPTIVE QUADRATURE,AUTOMATIC INTEGRATOR,GAUSS QUADRATURE,
C             INTEGRATION,NUMERICAL INTEGRATION,QUADRATURE
C***AUTHOR  JONES, R. E., (SNLA)
C***PURPOSE  GAUS8 integrates real functions of one variable over finite
C            intervals using an adaptive 8-point Legendre-Gauss
C            algorithm.  GAUS8 is intended primarily for high accuracy
C            integration or integration of smooth functions.
C***DESCRIPTION
C
C     Written by R.E. Jones
C
C     Abstract
C        GAUS8 integrates real functions of one variable over finite
C        intervals using an adaptive 8-point Legendre-Gauss algorithm.
C        GAUS8 is intended primarily for high accuracy integration
C        or integration of smooth functions.
```

```
C
C        GAUS8 calls I1MACH, R1MACH, XERROR
C
C     Description of Arguments
C
C        Input--
C        FUN - name of external function to be integrated.  This name
C              must be in an EXTERNAL statement in the calling program.
C              FUN must be a function of one real argument.  The value
C              of the argument to FUN is the variable of integration
C              which ranges from A to B.
C        A   - lower limit of integral
C        B   - upper limit of integral (may be less than A)
C        ERR - is a requested pseudorelative error tolerance.  Normally
C              pick a value of ABS(ERR) so that STOL .LT. ABS(ERR) .LE.
C              1.0E-3 where STOL is the single precision unit roundoff
C              =R1MACH(4).  ANS will normally have no more error than
C              ABS(ERR) times the integral of the absolute value of
C              FUN(X).  Usually, smaller values for ERR yield more
C              accuracy and require more function evaluations.
C
C              A negative value for ERR causes an estimate of the
C              absolute error in ANS to be returned in ERR.  Note that
C              ERR must be a variable (not a constant) in this case.
C              Note also that the user must reset the value of ERR
C              before making any more calls that use the variable ERR.
C
C        Output--
C        ERR - will be an estimate of the absolute error in ANS if the
C              input value of ERR was negative.  (ERR is unchanged if
C              the input value of ERR was non-negative.)  The estimated
C              error is solely for information to the user and should
C              not be used as a correction to the computed integral.
C        ANS - computed value of integral
C        IERR- a status code
C            --Normal Codes
C               1 ANS most likely meets requested error tolerance,
C                 or A=B.
C              -1 A and B are too nearly equal to allow normal
C                 integration.  ANS is set to zero.
C            --Abnormal Code
C               2 ANS probably does not meet requested error tolerance.
C***REFERENCES  (NONE)
C***ROUTINES CALLED  I1MACH,R1MACH,XERROR
C***END PROLOGUE  GAUS8
C
      INTEGER IERR, K, KML, KMX, L, LMN, LMX, LR, MXL, NBITS,
     1 NIB, NLMN, NLMX
      INTEGER I1MACH
      REAL A, AA, AE, ANIB, ANS, AREA, B, C, CE, EE, EF, EPS, ERR, EST,
     1 GL, GLR, GR, HH, SQ2, TOL, VL, VR, W1, W2, W3, W4, X1, X2, X3,
     2 X4, X, H
      REAL R1MACH, G8, FUN
```

```
      DIMENSION AA(30), HH(30), LR(30), VL(30), GR(30)
      SAVE X1, X2, X3, X4, W1, W2, W3, W4, SQ2,
     1 NLMN, KMX, KML
      DATA X1, X2, X3, X4/
     1     1.83434642495649805E-01,     5.25532409916328986E-01,
     2     7.96666477413626740E-01,     9.60289856497536232E-01/
      DATA W1, W2, W3, W4/
     1     3.62683783378361983E-01,     3.13706645877887287E-01,
     2     2.22381034453374471E-01,     1.01228536290376259E-01/
      DATA SQ2/1.41421356E0/
      DATA NLMN/1/,KMX/5000/,KML/6/
      G8(X,H)=H*((W1*(FUN(X-X1*H) + FUN(X+X1*H))
     1           +W2*(FUN(X-X2*H) + FUN(X+X2*H)))
     2          +(W3*(FUN(X-X3*H) + FUN(X+X3*H))
     3           +W4*(FUN(X-X4*H) + FUN(X+X4*H))))
C
C     INITIALIZE
C
C***FIRST EXECUTABLE STATEMENT  GAUS8
      K = I1MACH(11)
      ANIB = R1MACH(5)*FLOAT(K)/0.30102000E0
      NBITS = INT(ANIB)
      NLMX = MIN0(30,(NBITS*5)/8)
      ANS = 0.0E0
      IERR = 1
      CE = 0.0E0
      IF (A.EQ.B) GO TO 140
      LMX = NLMX
      LMN = NLMN
      IF (B.EQ.0.0E0) GO TO 10
      IF (SIGN(1.0E0,B)*A.LE.0.0E0) GO TO 10
      C = ABS(1.0E0-A/B)
      IF (C.GT.0.1E0) GO TO 10
      IF (C.LE.0.0E0) GO TO 140
      ANIB = 0.5E0 - ALOG(C)/0.69314718E0
      NIB = INT(ANIB)
      LMX = MIN0(NLMX,NBITS-NIB-7)
      IF (LMX.LT.1) GO TO 130
      LMN = MIN0(LMN,LMX)
   10 TOL = AMAX1(ABS(ERR),2.0E0**(5-NBITS))/2.0E0
      IF (ERR.EQ.0.0E0) TOL = SQRT(R1MACH(4))
      EPS = TOL
      HH(1) = (B-A)/4.0E0
      AA(1) = A
      LR(1) = 1
      L = 1
      EST = G8(AA(L)+2.0E0*HH(L),2.0E0*HH(L))
      K = 8
      AREA = ABS(EST)
      EF = 0.5E0
      MXL = 0
C
```

```
C     COMPUTE REFINED ESTIMATES, ESTIMATE THE ERROR, ETC.
C
   20 GL = G8(AA(L)+HH(L),HH(L))
      GR(L) = G8(AA(L)+3.0E0*HH(L),HH(L))
      K = K + 16
      AREA = AREA + (ABS(GL)+ABS(GR(L))-ABS(EST))
C     IF (L.LT.LMN) GO TO 11
      GLR = GL + GR(L)
      EE = ABS(EST-GLR)*EF
      AE = AMAX1(EPS*AREA,TOL*ABS(GLR))
      IF (EE-AE) 40, 40, 50
   30 MXL = 1
   40 CE = CE + (EST-GLR)
      IF (LR(L)) 60, 60, 80
C
C     CONSIDER THE LEFT HALF OF THIS LEVEL
C
   50 IF (K.GT.KMX) LMX = KML
      IF (L.GE.LMX) GO TO 30
      L = L + 1
      EPS = EPS*0.5E0
      EF = EF/SQ2
      HH(L) = HH(L-1)*0.5E0
      LR(L) = -1
      AA(L) = AA(L-1)
      EST = GL
      GO TO 20
C
C     PROCEED TO RIGHT HALF AT THIS LEVEL
C
   60 VL(L) = GLR
   70 EST = GR(L-1)
      LR(L) = 1
      AA(L) = AA(L) + 4.0E0*HH(L)
      GO TO 20
C
C     RETURN ONE LEVEL
C
   80 VR = GLR
   90 IF (L.LE.1) GO TO 120
      L = L - 1
      EPS = EPS*2.0E0
      EF = EF*SQ2
      IF (LR(L)) 100, 100, 110
  100 VL(L) = VL(L+1) + VR
      GO TO 70
  110 VR = VL(L+1) + VR
      GO TO 90
C
C      EXIT
C
  120 ANS = VR
      IF ((MXL.EQ.0) .OR. (ABS(CE).LE.2.0E0*TOL*AREA)) GO TO 140
```

```
      IERR = 2
      CALL XERR( 'GAUS8- ANS IS PROBABLY INSUFFICIENTLY ACCURATE.')
      GO TO 140
  130 IERR = -1
      CALL XERR( 'GAUS8- THE FOLLOWING TEMPORARY DIAGNOSTIC WILL APPEA
     1R ONLY ONCE.  A AND B ARE TOO NEARLY EQUAL TO ALLOW NORMAL INTEGRA
     2TION.  ANS IS SET TO ZERO, AND IERR=-1.')
  140 CONTINUE
      IF (ERR.LT.0.0E0) ERR = CE
      RETURN
      END
*DECK,HKSEQ
      SUBROUTINE HKSEQ (X,M,H,IERR)
C     ***BEGIN PROLOGUE  HKSEQ
C     ***REFER TO  BSKIN
C     ***ROUTINES CALLED  I1MACH,R1MACH
C     ***DESCRIPTION
C
C     HKSEQ IS AN ADAPTATION OF SUBROUTINE PSIFN, ACM TRANS. MATH.
C     SOFTWARE, 1983 TO GENERATE H(K,X)=(-X)**(K+1)*(PSI(K,X)-
C     -PSI(K,X+0.5))/GAMMA(K+1) FOR K=0,...,M.
C     ***END PROLOGUE  HKSEQ
      INTEGER I,IERR,J,K,M,MX,NX
      INTEGER I1MACH
      REAL B,FK,FLN,FN,FNP,H,HRX,RLN,RXSQ,R1M5,S,SLOPE,T,TK,TRM,TRMH,
     *  TRMR,TST,U,V,WDTOL,X,XDMY,XH,XINC,XM,XMIN,YINT
      REAL R1MACH
      DIMENSION B(22),TRM(22),TRMR(25),TRMH(25),U(25),V(25),H(1)
C     ----------------------------------------------------------------
C     SCALED BERNOULLI NUMBERS 2.0*B(2K)*(1-2**(-2K))
C     ----------------------------------------------------------------
      DATA B(1),B(2),B(3),B(4),B(5),B(6),B(7),B(8),B(9),B(10),B(11),B(12
     *  ),B(13),B(14),B(15),B(16),B(17),B(18),B(19),B(20),B(21),B(22) /
     *  1.00000000000000000E+00,-5.00000000000000000E-01,
     *  2.50000000000000000E-01,-6.25000000000000000E-02,
     *  4.68750000000000000E-02,-6.64062500000000000E-02,
     *  1.51367187500000000E-01,-5.06103515625000000E-01,
     *  2.33319091796875000E+00,-1.41840972900390625E+01,
     *  1.09941936492919922E+02,-1.05824747562408447E+03,
     *  1.23842434241771698E+04,-1.73160495905935764E+05,
     *  2.85103429084961116E+06,-5.45964619322445132E+07,
     *  1.20316174668075304E+09,-3.02326315271452307E+10,
     *  8.59229286072319606E+11,-2.74233104097776039E+13,
     *  9.76664637943633248E+14,-3.85931586838450360E+16 /
C
C
C     ***FIRST EXECUTABLE STATEMENT  HKSEQ
      IERR = 0
      WDTOL = AMAX1(R1MACH(4),1.0E-18)
      FN = FLOAT(M-1)
      FNP = FN+1.0E0
C     ----------------------------------------------------------------
```

```
C     COMPUTE XMIN
C     ------------------------------------------------------------------
      R1M5 = R1MACH(5)
      RLN = R1M5*FLOAT(I1MACH(11))
      RLN = AMIN1(RLN,18.06E0)
      FLN = AMAX1(RLN,3.0E0)-3.0E0
      YINT = 3.50E0+0.40E0*FLN
      SLOPE = 0.21E0+FLN*(0.0006038E0*FLN+0.008677E0)
      XM = YINT+SLOPE*FN
      MX = INT(XM)+1
      XMIN = FLOAT(MX)
C     ------------------------------------------------------------------
C     GENERATE H(M-1,XDMY)*XDMY**(M) BY THE ASYMPTOTIC EXPANSION
C     ------------------------------------------------------------------
      XDMY = X
      XINC = 0.0E0
      IF (X.GE.XMIN) GO TO 10
      NX = INT(X)
      XINC = XMIN-FLOAT(NX)
      XDMY = X+XINC
   10 CONTINUE
      RXSQ = 1.0E0/(XDMY*XDMY)
      HRX = 0.5E0/XDMY
      TST = 0.5E0*WDTOL
      T = FNP*HRX
C     ------------------------------------------------------------------
C     INITIALIZE COEFFICIENT ARRAY
C     ------------------------------------------------------------------
      S = T*B(3)
      IF (ABS(S).LT.TST) GO TO 30
      TK = 2.0E0
      DO 20 K = 4, 22
        T = T*((TK+FN+1.0E0)/(TK+1.0E0))*((TK+FN)/(TK+2.0E0))*RXSQ
        TRM(K) = T*B(K)
        IF (ABS(TRM(K)).LT.TST) GO TO 30
        S = S+TRM(K)
        TK = TK+2.0E0
   20 CONTINUE
      GO TO 110
   30 CONTINUE
      H(M) = S+0.5E0
      IF (M.EQ.1) GO TO 70
C     ------------------------------------------------------------------
C     GENERATE LOWER DERIVATIVES, I.LT.M-1
C     ------------------------------------------------------------------
      DO 60 I = 2, M
        FNP = FN
        FN = FN-1.0E0
        S = FNP*HRX*B(3)
        IF (ABS(S).LT.TST) GO TO 50
        FK = FNP+3.0E0
        DO 40 K = 4, 22
          TRM(K) = TRM(K)*FNP/FK
```

```
            IF (ABS(TRM(K)).LT.TST) GO TO 50
            S = S+TRM(K)
            FK = FK+2.0E0
   40     CONTINUE
          GO TO 110
   50     CONTINUE
          MX = M-I+1
          H(MX) = S+0.5E0
   60   CONTINUE
   70   CONTINUE
        IF (XINC.EQ.0.0E0) RETURN
C       ------------------------------------------------------------------
C       RECUR BACKWARD FROM XDMY TO X
C       ------------------------------------------------------------------
        XH = X+0.5E0
        S = 0.0E0
        NX = INT(XINC)
        DO 80 I = 1, NX
          TRMR(I) = X/(X+FLOAT(NX-I))
          U(I) = TRMR(I)
          TRMH(I) = X/(XH+FLOAT(NX-I))
          V(I) = TRMH(I)
          S = S+U(I)-V(I)
   80   CONTINUE
        MX = NX+1
        TRMR(MX) = X/XDMY
        U(MX) = TRMR(MX)
        H(1) = H(1)*TRMR(MX)+S
        IF (M.EQ.1) RETURN
        DO 100 J = 2, M
          S = 0.0E0
          DO 90 I = 1, NX
            TRMR(I) = TRMR(I)*U(I)
            TRMH(I) = TRMH(I)*V(I)
            S = S+TRMR(I)-TRMH(I)
   90     CONTINUE
          TRMR(MX) = TRMR(MX)*U(MX)
          H(J) = H(J)*TRMR(MX)+S
  100   CONTINUE
        RETURN
  110   CONTINUE
        IERR = 2
        RETURN
        END
*DECK I1MACH
        INTEGER FUNCTION I1MACH(I)
C***BEGIN PROLOGUE  I1MACH
C***DATE WRITTEN   750101   (YYMMDD)
C***REVISION DATE  890213   (YYMMDD)
C***CATEGORY NO.  R1
C***KEYWORDS  LIBRARY=SLATEC,TYPE=INTEGER(I1MACH-I),MACHINE CONSTANTS
C***AUTHOR  FOX, P. A., (BELL LABS)
C           HALL, A. D., (BELL LABS)
C           SCHRYER, N. L., (BELL LABS)
```

```
C***PURPOSE  Returns integer machine dependent constants
C***DESCRIPTION
C
C     I1MACH can be used to obtain machine-dependent parameters
C     for the local machine environment.  It is a function
C     subroutine with one (input) argument, and can be called
C     as follows, for example
C
C          K = I1MACH(I)
C
C     where I=1,...,16.  The (output) value of K above is
C     determined by the (input) value of I.  The results for
C     various values of I are discussed below.
C
C  I/O unit numbers.
C    I1MACH( 1) = the standard input unit.
C    I1MACH( 2) = the standard output unit.
C    I1MACH( 3) = the standard punch unit.
C    I1MACH( 4) = the standard error message unit.
C
C  Words.
C    I1MACH( 5) = the number of bits per integer storage unit.
C    I1MACH( 6) = the number of characters per integer storage unit.
C
C  Integers.
C    assume integers are represented in the S-digit, base-A form
C
C                sign ( X(S-1)*A**(S-1) + ... + X(1)*A + X(0) )
C
C                where 0 .LE. X(I) .LT. A for I=0,...,S-1.
C    I1MACH( 7) = A, the base.
C    I1MACH( 8) = S, the number of base-A digits.
C    I1MACH( 9) = A**S - 1, the largest magnitude.
C
C  Floating-Point Numbers.
C    Assume floating-point numbers are represented in the T-digit,
C    base-B form
C                sign (B**E)*( (X(1)/B) + ... + (X(T)/B**T) )
C
C                where 0 .LE. X(I) .LT. B for I=1,...,T,
C                0 .LT. X(1), and EMIN .LE. E .LE. EMAX.
C    I1MACH(10) = B, the base.
C
C  Single-Precision
C    I1MACH(11) = T, the number of base-B digits.
C    I1MACH(12) = EMIN, the smallest exponent E.
C    I1MACH(13) = EMAX, the largest exponent E.
C
C  Double-Precision
C    I1MACH(14) = T, the number of base-B digits.
C    I1MACH(15) = EMIN, the smallest exponent E.
C    I1MACH(16) = EMAX, the largest exponent E.
C
```

```
C  To alter this function for a particular environment,
C  the desired set of DATA statements should be activated by
C  removing the C from column 1.  Also, the values of
C  I1MACH(1) - I1MACH(4) should be checked for consistency
C  with the local operating system.
C
C***REFERENCES  FOX P.A., HALL A.D., SCHRYER N.L.,*FRAMEWORK FOR A
C                 PORTABLE LIBRARY*, ACM TRANSACTIONS ON MATHEMATICAL
C                 SOFTWARE, VOL. 4, NO. 2, JUNE 1978, PP. 177-188.
C***ROUTINES CALLED  (NONE)
C***END PROLOGUE  I1MACH
C
      INTEGER IMACH(16),OUTPUT
      SAVE IMACH
      EQUIVALENCE (IMACH(4),OUTPUT)
C
C     MACHINE CONSTANTS FOR THE AMIGA
C     ABSOFT COMPILER
C
C     DATA IMACH(1) /    5 /
C     DATA IMACH(2) /    6 /
C     DATA IMACH(3) /    5 /
C     DATA IMACH(4) /    6 /
C     DATA IMACH(5) /   32 /
C     DATA IMACH(6) /    4 /
C     DATA IMACH(7) /    2 /
C     DATA IMACH(8) /   31 /
C     DATA IMACH(9) / 2147483647 /
C     DATA IMACH(10)/    2 /
C     DATA IMACH(11)/   24 /
C     DATA IMACH(12)/ -126 /
C     DATA IMACH(13)/  127 /
C     DATA IMACH(14)/   53 /
C     DATA IMACH(15)/ -1022 /
C     DATA IMACH(16)/  1023 /
C
C     MACHINE CONSTANTS FOR THE APOLLO
C
C     DATA IMACH(1) /    5 /
C     DATA IMACH(2) /    6 /
C     DATA IMACH(3) /    6 /
C     DATA IMACH(4) /    6 /
C     DATA IMACH(5) /   32 /
C     DATA IMACH(6) /    4 /
C     DATA IMACH(7) /    2 /
C     DATA IMACH(8) /   31 /
C     DATA IMACH(9) / 2147483647 /
C     DATA IMACH(10)/    2 /
C     DATA IMACH(11)/   24 /
C     DATA IMACH(12)/ -125 /
C     DATA IMACH(13)/  129 /
C     DATA IMACH(14)/   53 /
C     DATA IMACH(15)/ -1021 /
C     DATA IMACH(16)/  1025 /
```

```
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 1700 SYSTEM
C
C     DATA IMACH( 1) /    7 /
C     DATA IMACH( 2) /    2 /
C     DATA IMACH( 3) /    2 /
C     DATA IMACH( 4) /    2 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    4 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   33 /
C     DATA IMACH( 9) / Z1FFFFFFFF /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   24 /
C     DATA IMACH(12) / -256 /
C     DATA IMACH(13) /  255 /
C     DATA IMACH(14) /   60 /
C     DATA IMACH(15) / -256 /
C     DATA IMACH(16) /  255 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 5700 SYSTEM
C
C     DATA IMACH( 1) /   5 /
C     DATA IMACH( 2) /   6 /
C     DATA IMACH( 3) /   7 /
C     DATA IMACH( 4) /   6 /
C     DATA IMACH( 5) /  48 /
C     DATA IMACH( 6) /   6 /
C     DATA IMACH( 7) /   2 /
C     DATA IMACH( 8) /  39 /
C     DATA IMACH( 9) / O0007777777777777 /
C     DATA IMACH(10) /   8 /
C     DATA IMACH(11) /  13 /
C     DATA IMACH(12) / -50 /
C     DATA IMACH(13) /  76 /
C     DATA IMACH(14) /  26 /
C     DATA IMACH(15) / -50 /
C     DATA IMACH(16) /  76 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 6700/7700 SYSTEMS
C
C     DATA IMACH( 1) /   5 /
C     DATA IMACH( 2) /   6 /
C     DATA IMACH( 3) /   7 /
C     DATA IMACH( 4) /   6 /
C     DATA IMACH( 5) /  48 /
C     DATA IMACH( 6) /   6 /
C     DATA IMACH( 7) /   2 /
C     DATA IMACH( 8) /  39 /
C     DATA IMACH( 9) / O0007777777777777 /
C     DATA IMACH(10) /   8 /
C     DATA IMACH(11) /  13 /
C     DATA IMACH(12) / -50 /
C     DATA IMACH(13) /  76 /
```

```
C     DATA IMACH(14) /  26 /
C     DATA IMACH(15) / -32754 /
C     DATA IMACH(16) /  32780 /
C
C     MACHINE CONSTANTS FOR THE CDC 170/180 SERIES USING NOS/VE
C
C     DATA IMACH( 1) /     5 /
C     DATA IMACH( 2) /     6 /
C     DATA IMACH( 3) /     7 /
C     DATA IMACH( 4) /     6 /
C     DATA IMACH( 5) /    64 /
C     DATA IMACH( 6) /     8 /
C     DATA IMACH( 7) /     2 /
C     DATA IMACH( 8) /    63 /
C     DATA IMACH( 9) / 9223372036854775807 /
C     DATA IMACH(10) /     2 /
C     DATA IMACH(11) /    47 /
C     DATA IMACH(12) / -4095 /
C     DATA IMACH(13) /  4094 /
C     DATA IMACH(14) /    94 /
C     DATA IMACH(15) / -4095 /
C     DATA IMACH(16) /  4094 /
C
C     MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    7 /
C     DATA IMACH( 4) /6LOUTPUT/
C     DATA IMACH( 5) /   60 /
C     DATA IMACH( 6) /   10 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   48 /
C     DATA IMACH( 9) / 00007777777777777777B /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   47 /
C     DATA IMACH(12) / -929 /
C     DATA IMACH(13) / 1070 /
C     DATA IMACH(14) /   94 /
C     DATA IMACH(15) / -929 /
C     DATA IMACH(16) / 1069 /
C
C     MACHINE CONSTANTS FOR THE CELERITY C1260
C
C     DATA IMACH(1) /    5 /
C     DATA IMACH(2) /    6 /
C     DATA IMACH(3) /    6 /
C     DATA IMACH(4) /    0 /
C     DATA IMACH(5) /   32 /
C     DATA IMACH(6) /    4 /
C     DATA IMACH(7) /    2 /
C     DATA IMACH(8) /   31 /
C     DATA IMACH(9) / Z'7FFFFFFF' /
```

```
C     DATA IMACH(10)/    2 /
C     DATA IMACH(11)/   24 /
C     DATA IMACH(12)/ -126 /
C     DATA IMACH(13)/  127 /
C     DATA IMACH(14)/   53 /
C     DATA IMACH(15)/ -1022 /
C     DATA IMACH(16)/  1023 /
C
C     MACHINE CONSTANTS FOR THE CONVEX C-1
C
C     DATA IMACH( 1) /     5/
C     DATA IMACH( 2) /     6/
C     DATA IMACH( 3) /     7/
C     DATA IMACH( 4) /     6/
C     DATA IMACH( 5) /    32/
C     DATA IMACH( 6) /     4/
C     DATA IMACH( 7) /     2/
C     DATA IMACH( 8) /    31/
C     DATA IMACH( 9) /2147483647/
C     DATA IMACH(10) /     2/
C     DATA IMACH(11) /    24/
C     DATA IMACH(12) /  -128/
C     DATA IMACH(13) /   127/
C     DATA IMACH(14) /    53/
C     DATA IMACH(15) / -1024/
C     DATA IMACH(16) /  1023/
C
C     MACHINE CONSTANTS FOR THE CRAY-1
C
C     DATA IMACH( 1) /   100 /
C     DATA IMACH( 2) /   101 /
C     DATA IMACH( 3) /   102 /
C     DATA IMACH( 4) /   101 /
C     DATA IMACH( 5) /    64 /
C     DATA IMACH( 6) /     8 /
C     DATA IMACH( 7) /     2 /
C     DATA IMACH( 8) /    63 /
C     DATA IMACH( 9) /  777777777777777777777B /
C     DATA IMACH(10) /     2 /
C     DATA IMACH(11) /    47 /
C     DATA IMACH(12) / -8189 /
C     DATA IMACH(13) /  8190 /
C     DATA IMACH(14) /    94 /
C     DATA IMACH(15) / -8099 /
C     DATA IMACH(16) /  8190 /
C
C     MACHINE CONSTANTS FOR THE DATA GENERAL ECLIPSE S/200
C
C     DATA IMACH( 1) /   11 /
C     DATA IMACH( 2) /   12 /
C     DATA IMACH( 3) /    8 /
C     DATA IMACH( 4) /   10 /
C     DATA IMACH( 5) /   16 /
C     DATA IMACH( 6) /    2 /
```

```
C      DATA IMACH( 7) /    2 /
C      DATA IMACH( 8) /   15 /
C      DATA IMACH( 9) /32767 /
C      DATA IMACH(10) /   16 /
C      DATA IMACH(11) /    6 /
C      DATA IMACH(12) /  -64 /
C      DATA IMACH(13) /   63 /
C      DATA IMACH(14) /   14 /
C      DATA IMACH(15) /  -64 /
C      DATA IMACH(16) /   63 /
C
C      MACHINE CONSTANTS FOR THE ELXSI 6400
C
C      DATA IMACH( 1) /     5/
C      DATA IMACH( 2) /     6/
C      DATA IMACH( 3) /     6/
C      DATA IMACH( 4) /     6/
C      DATA IMACH( 5) /    32/
C      DATA IMACH( 6) /     4/
C      DATA IMACH( 7) /     2/
C      DATA IMACH( 8) /    32/
C      DATA IMACH( 9) /2147483647/
C      DATA IMACH(10) /     2/
C      DATA IMACH(11) /    24/
C      DATA IMACH(12) /  -126/
C      DATA IMACH(13) /   127/
C      DATA IMACH(14) /    53/
C      DATA IMACH(15) / -1022/
C      DATA IMACH(16) /  1023/
C
C      MACHINE CONSTANTS FOR THE HARRIS 220
C
C      DATA IMACH( 1) /       5 /
C      DATA IMACH( 2) /       6 /
C      DATA IMACH( 3) /       0 /
C      DATA IMACH( 4) /       6 /
C      DATA IMACH( 5) /      24 /
C      DATA IMACH( 6) /       3 /
C      DATA IMACH( 7) /       2 /
C      DATA IMACH( 8) /      23 /
C      DATA IMACH( 9) / 8388607 /
C      DATA IMACH(10) /       2 /
C      DATA IMACH(11) /      23 /
C      DATA IMACH(12) /    -127 /
C      DATA IMACH(13) /     127 /
C      DATA IMACH(14) /      38 /
C      DATA IMACH(15) /    -127 /
C      DATA IMACH(16) /     127 /
C
C      MACHINE CONSTANTS FOR THE HONEYWELL 600/6000 SERIES
C
C      DATA IMACH( 1) /    5 /
C      DATA IMACH( 2) /    6 /
C      DATA IMACH( 3) /   43 /
```

```
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    6 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / O377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -127 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   63 /
C     DATA IMACH(15) / -127 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR THE HP 2100
C     3 WORD DOUBLE PRECISION OPTION WITH FTN4
C
C     DATA IMACH(1) /      5/
C     DATA IMACH(2) /      6 /
C     DATA IMACH(3) /      4 /
C     DATA IMACH(4) /      1 /
C     DATA IMACH(5) /     16 /
C     DATA IMACH(6) /      2 /
C     DATA IMACH(7) /      2 /
C     DATA IMACH(8) /     15 /
C     DATA IMACH(9) /  32767 /
C     DATA IMACH(10)/      2 /
C     DATA IMACH(11)/     23 /
C     DATA IMACH(12)/   -128 /
C     DATA IMACH(13)/    127 /
C     DATA IMACH(14)/     39 /
C     DATA IMACH(15)/   -128 /
C     DATA IMACH(16)/    127 /
C
C     MACHINE CONSTANTS FOR THE HP 2100
C     4 WORD DOUBLE PRECISION OPTION WITH FTN4
C
C     DATA IMACH(1) /      5 /
C     DATA IMACH(2) /      6 /
C     DATA IMACH(3) /      4 /
C     DATA IMACH(4) /      1 /
C     DATA IMACH(5) /     16 /
C     DATA IMACH(6) /      2 /
C     DATA IMACH(7) /      2 /
C     DATA IMACH(8) /     15 /
C     DATA IMACH(9) /  32767 /
C     DATA IMACH(10)/      2 /
C     DATA IMACH(11)/     23 /
C     DATA IMACH(12)/   -128 /
C     DATA IMACH(13)/    127 /
C     DATA IMACH(14)/     55 /
C     DATA IMACH(15)/   -128 /
C     DATA IMACH(16)/    127 /
C
```

```
C     MACHINE CONSTANTS FOR THE HP 9000
C
C     DATA IMACH(1)  /    5 /
C     DATA IMACH(2)  /    6 /
C     DATA IMACH(3)  /    6 /
C     DATA IMACH(3)  /    7 /
C     DATA IMACH(5)  /   32 /
C     DATA IMACH(6)  /    4 /
C     DATA IMACH(7)  /    2 /
C     DATA IMACH(8)  /   32 /
C     DATA IMACH(9)  /2147483647 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   24 /
C     DATA IMACH(12) / -126 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   53 /
C     DATA IMACH(15) /-1015 /
C     DATA IMACH(16) / 1017 /
C
C     MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C     THE XEROX SIGMA 5/7/9, THE SEL SYSTEMS 85/86, AND
C     THE PERKIN ELMER (INTERDATA) 7/32.
C
C     DATA IMACH( 1) /   5 /
C     DATA IMACH( 2) /   6 /
C     DATA IMACH( 3) /   7 /
C     DATA IMACH( 4) /   6 /
C     DATA IMACH( 5) /  32 /
C     DATA IMACH( 6) /   4 /
C     DATA IMACH( 7) /  16 /
C     DATA IMACH( 8) /  31 /
C     DATA IMACH( 9) / Z7FFFFFFF /
C     DATA IMACH(10) /  16 /
C     DATA IMACH(11) /   6 /
C     DATA IMACH(12) / -64 /
C     DATA IMACH(13) /  63 /
C     DATA IMACH(14) /  14 /
C     DATA IMACH(15) / -64 /
C     DATA IMACH(16) /  63 /
C
C     MACHINE CONSTANTS FOR THE IBM PC
C
C     DATA IMACH( 1) /     5 /
C     DATA IMACH( 2) /     6 /
C     DATA IMACH( 3) /     0 /
C     DATA IMACH( 4) /     0 /
C     DATA IMACH( 5) /    32 /
C     DATA IMACH( 6) /     4 /
C     DATA IMACH( 7) /     2 /
C     DATA IMACH( 8) /    31 /
C     DATA IMACH( 9) / 2147483647 /
C     DATA IMACH(10) /     2 /
C     DATA IMACH(11) /    24 /
C     DATA IMACH(12) /  -125 /
```

```
C     DATA IMACH(13) /   127 /
C     DATA IMACH(14) /    53 /
C     DATA IMACH(15) / -1021 /
C     DATA IMACH(16) /  1023 /
C
C     MACHINE CONSTANTS FOR THE IBM RS 6000
C
C     DATA IMACH( 1) /          5 /
C     DATA IMACH( 2) /          6 /
C     DATA IMACH( 3) /          6 /
C     DATA IMACH( 4) /          0 /
C     DATA IMACH( 5) /         32 /
C     DATA IMACH( 6) /          4 /
C     DATA IMACH( 7) /          2 /
C     DATA IMACH( 8) /         31 /
C     DATA IMACH( 9) / 2147483647 /
C     DATA IMACH(10) /          2 /
C     DATA IMACH(11) /         24 /
C     DATA IMACH(12) /       -125 /
C     DATA IMACH(13) /        128 /
C     DATA IMACH(14) /         53 /
C     DATA IMACH(15) /      -1021 /
C     DATA IMACH(16) /       1024 /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KA PROCESSOR)
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    5 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / "377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -128 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   54 /
C     DATA IMACH(15) / -101 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KI PROCESSOR)
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    5 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / "377777777777 /
```

```
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -128 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   62 /
C     DATA IMACH(15) / -128 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C     32-BIT INTEGER ARITHMETIC.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   32 /
C     DATA IMACH( 6) /    4 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   31 /
C     DATA IMACH( 9) / 2147483647 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   24 /
C     DATA IMACH(12) / -127 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   56 /
C     DATA IMACH(15) / -127 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C     16-BIT INTEGER ARITHMETIC.
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    5 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   16 /
C     DATA IMACH( 6) /    2 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   15 /
C     DATA IMACH( 9) / 32767 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   24 /
C     DATA IMACH(12) / -127 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   56 /
C     DATA IMACH(15) / -127 /
C     DATA IMACH(16) /  127 /
C
C     MACHINE CONSTANTS FOR THE SILICON GRAPHICS IRIS
C
C     DATA IMACH( 1) /     5 /
C     DATA IMACH( 2) /     6 /
C     DATA IMACH( 3) /     6 /
C     DATA IMACH( 4) /     0 /
```

```
C     DATA IMACH( 5) /    32 /
C     DATA IMACH( 6) /     4 /
C     DATA IMACH( 7) /     2 /
C     DATA IMACH( 8) /    31 /
C     DATA IMACH( 9) / 2147483647 /
C     DATA IMACH(10) /     2 /
C     DATA IMACH(11) /    23 /
C     DATA IMACH(12) /  -126 /
C     DATA IMACH(13) /   127 /
C     DATA IMACH(14) /    52 /
C     DATA IMACH(15) / -1022 /
C     DATA IMACH(16) /  1023 /
C
C     MACHINE CONSTANTS FOR THE SUN
C
C     DATA IMACH(1) /    5 /
C     DATA IMACH(2) /    6 /
C     DATA IMACH(3) /    6 /
C     DATA IMACH(4) /    6 /
C     DATA IMACH(5) /   32 /
C     DATA IMACH(6) /    4 /
C     DATA IMACH(7) /    2 /
C     DATA IMACH(8) /   31 /
C     DATA IMACH(9) /2147483647 /
C     DATA IMACH(10)/    2 /
C     DATA IMACH(11)/   24 /
C     DATA IMACH(12)/ -125 /
C     DATA IMACH(13)/  128 /
C     DATA IMACH(14)/   53 /
C     DATA IMACH(15)/ -1021 /
C     DATA IMACH(16)/  1024 /
C
C     MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES FTN COMPILER
C
C
C     DATA IMACH( 1) /    5 /
C     DATA IMACH( 2) /    6 /
C     DATA IMACH( 3) /    1 /
C     DATA IMACH( 4) /    6 /
C     DATA IMACH( 5) /   36 /
C     DATA IMACH( 6) /    4 /
C     DATA IMACH( 7) /    2 /
C     DATA IMACH( 8) /   35 /
C     DATA IMACH( 9) / O377777777777 /
C     DATA IMACH(10) /    2 /
C     DATA IMACH(11) /   27 /
C     DATA IMACH(12) / -128 /
C     DATA IMACH(13) /  127 /
C     DATA IMACH(14) /   60 /
C     DATA IMACH(15) /-1024 /
C     DATA IMACH(16) / 1023 /
C
C
```

```
C     MACHINE CONSTANTS FOR THE VAX 11/780
C
      DATA IMACH(1) /    5 /
      DATA IMACH(2) /    6 /
      DATA IMACH(3) /    5 /
      DATA IMACH(4) /    6 /
      DATA IMACH(5) /   32 /
      DATA IMACH(6) /    4 /
      DATA IMACH(7) /    2 /
      DATA IMACH(8) /   31 /
      DATA IMACH(9) /2147483647 /
      DATA IMACH(10)/    2 /
      DATA IMACH(11)/   24 /
      DATA IMACH(12)/ -127 /
      DATA IMACH(13)/  127 /
      DATA IMACH(14)/   56 /
      DATA IMACH(15)/ -127 /
      DATA IMACH(16)/  127 /
C
C     MACHINE CONSTANTS FOR THE Z80 MICROPROCESSOR
C
C     DATA IMACH( 1) /     1/
C     DATA IMACH( 2) /     1/
C     DATA IMACH( 3) /     0/
C     DATA IMACH( 4) /     1/
C     DATA IMACH( 5) /    16/
C     DATA IMACH( 6) /     2/
C     DATA IMACH( 7) /     2/
C     DATA IMACH( 8) /    15/
C     DATA IMACH( 9) / 32767/
C     DATA IMACH(10) /     2/
C     DATA IMACH(11) /    24/
C     DATA IMACH(12) /  -127/
C     DATA IMACH(13) /   127/
C     DATA IMACH(14) /    56/
C     DATA IMACH(15) /  -127/
C     DATA IMACH(16) /   127/
C
C
C***FIRST EXECUTABLE STATEMENT  I1MACH
      IF (I .LT. 1  .OR.  I .GT. 16) GO TO 10
C
      I1MACH = IMACH(I)
      RETURN
C
   10 CONTINUE
      WRITE (UNIT = OUTPUT, FMT = 9000)
 9000 FORMAT ('1ERROR    1 IN I1MACH - I OUT OF BOUNDS')
C
C     CALL FDUMP
C
C
      STOP
      END
```

```
*DECK,INITS
      FUNCTION INITS (OS,NOS,ETA)
C     ***BEGIN PROLOGUE  INITS
C     ***DATE WRITTEN   770401   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  C3A2
C     ***KEYWORDS  INITIALIZE,ORTHOGONAL SERIES,SPECIAL FUNCTION
C     ***AUTHOR  FULLERTON, W., (LANL)
C     ***PURPOSE  INITIALIZES AN ORTHOGONAL SERIES SO THAT IT DEFINES TH
C     NUMBER OF TERMS TO CARRY IN THE SERIES TO MEET A SPECIFIED
C     ERROR.
C     ***DESCRIPTION
C
C     INITIALIZE THE ORTHOGONAL SERIES SO THAT INITS IS THE NUMBER OF TE
C     NEEDED TO INSURE THE ERROR IS NO LARGER THAN ETA.  ORDINARILY, ETA
C     WILL BE CHOSEN TO BE ONE-TENTH MACHINE PRECISION.
C
C     INPUT ARGUMENTS --
C     OS     ARRAY OF NOS COEFFICIENTS IN AN ORTHOGONAL SERIES.
C     NOS    NUMBER OF COEFFICIENTS IN OS.
C     ETA    REQUESTED ACCURACY OF SERIES.
C     ***REFERENCES  (NONE)
C     ***ROUTINES CALLED  XERROR
C     ***END PROLOGUE  INITS
      DIMENSION OS(NOS)
C     ***FIRST EXECUTABLE STATEMENT  INITS
      IF (NOS.LT.1) CALL XERR ('INITS   NUMBER OF COEFFICIENTS LT 1')
C
      ERR = 0.
      DO 10 II = 1, NOS
        I = NOS+1-II
        ERR = ERR+ABS(OS(I))
        IF (ERR.GT.ETA) GO TO 20
   10 CONTINUE
C
   20 IF (I.EQ.NOS) CALL XERR ('INITS   ETA MAY BE TOO SMALL')
      INITS = I
C
      RETURN
      END
*DECK,PSIXN
      FUNCTION PSIXN (N)
C     ***BEGIN PROLOGUE  PSIXN
C     ***REFER TO  EXINT
C
C     THIS SUBROUTINE RETURNS VALUES OF PSI(X)=DERIVATIVE OF LOG
C     GAMMA(X), X .GT. 0.0 AT INTEGER ARGUMENTS. A TABLE LOOK-UP IS
C     PERFORMED FOR N .LE. 100, AND THE ASYMPTOTIC EXPANSION IS
C     EVALUATED FOR N .GT. 100.
C     ***ROUTINES CALLED  R1MACH
C     ***END PROLOGUE  PSIXN
C
      INTEGER N,K
      REAL AK,AX,B,C,FN,RFN2,TRM,S,WDTOL
```

```
      REAL R1MACH
      DIMENSION B(6),C(100)
      DATA C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8),C(9),C(10),C(11),C(12
     *  ),C(13),C(14),C(15),C(16),C(17),C(18),C(19),C(20),C(21),C(22),C(
     *  23),C(24) / -5.77215664901532861E-01,4.22784335098467139E-01,
     *  9.22784335098467139E-01,1.25611766843180047E+00,
     *  1.50611766843180047E+00,1.70611766843180047E+00,
     *  1.87278433509846714E+00,2.01564147795561000E+00,
     *  2.14064147795561000E+00,2.25175258906672111E+00,
     *  2.35175258906672111E+00,2.44266167997581202E+00,
     *  2.52599501330914535E+00,2.60291809023222227E+00,
     *  2.67434666166079370E+00,2.74101332832746037E+00,
     *  2.80351332832746037E+00,2.86233685773922507E+00,
     *  2.91789241329478063E+00,2.97052399224214905E+00,
     *  3.02052399224214905E+00,3.06814303986119667E+00,
     *  3.11359758531574212E+00,3.15707584618530734E+00 /
      DATA C(25),C(26),C(27),C(28),C(29),C(30),C(31),C(32),C(33),C(34),C
     *  (35),C(36),C(37),C(38),C(39),C(40),C(41),C(42),C(43),C(44),C(45)
     *  ,C(46),C(47),C(48) / 3.19874251285197401E+00,
     *  3.23874251285197401E+00,3.27720405131351247E+00,
     *  3.31424108835054951E+00,3.34995537406483522E+00,
     *  3.38443813268552488E+00,3.41777146601885821E+00,
     *  3.45002953053498724E+00,3.48127953053498724E+00,
     *  3.51158256083801755E+00,3.54099432554389990E+00,
     *  3.56956575411532847E+00,3.59734353189310625E+00,
     *  3.62437055892013327E+00,3.65068634839381748E+00,
     *  3.67632737403484313E+00,3.70132737403484313E+00,
     *  3.72571761793728215E+00,3.74952714174680596E+00,
     *  3.77278295570029433E+00,3.79551022842756706E+00,
     *  3.81773245064978928E+00,3.83947158108457189E+00,
     *  3.86074817682925274E+00 /
      DATA C(49),C(50),C(51),C(52),C(53),C(54),C(55),C(56),C(57),C(58),C
     *  (59),C(60),C(61),C(62),C(63),C(64),C(65),C(66),C(67),C(68),C(69)
     *  ,C(70),C(71),C(72) / 3.88158151016258607E+00,
     *  3.90198967342789220E+00,3.92198967342789220E+00,
     *  3.94159751656514710E+00,3.96082828579591633E+00,
     *  3.97969621032421822E+00,3.99821472884273674E+00,
     *  4.01639654702455492E+00,4.03425368988169777E+00,
     *  4.05179754953082058E+00,4.06903892884116541E+00,
     *  4.08598808138353829E+00,4.10265474805020496E+00,
     *  4.11904819067315578E+00,4.13517722293122029E+00,
     *  4.15105023880423617E+00,4.16667523880423617E+00,
     *  4.18205985418885155E+00,4.19721136934036670E+00,
     *  4.21213674247469506E+00,4.22684262482763624E+00,
     *  4.24133537845082464E+00,4.25562109273653893E+00,
     *  4.26970559977879245E+00 /
      DATA C(73),C(74),C(75),C(76),C(77),C(78),C(79),C(80),C(81),C(82),C
     *  (83),C(84),C(85),C(86),C(87),C(88),C(89),C(90),C(91),C(92),C(93)
     *  ,C(94),C(95),C(96) / 4.28359448866768134E+00,
     *  4.29729311880466764E+00,4.31080663231818115E+00,
     *  4.32413996565151449E+00,4.33729786038835659E+00,
     *  4.35028487337536958E+00,4.36310538619588240E+00,
     *  4.37576361404398366E+00,4.38826361404398366E+00,
     *  4.40060929305632934E+00,4.41280441500754886E+00,
```

```
     *  4.42485260777863319E+00,4.43675736968339510E+00,
     *  4.44852207556574804E+00,4.46014998254249223E+00,
     *  4.47164423541605544E+00,4.48300787177969181E+00,
     *  4.49424382683587158E+00,4.50535493794698269E+00,
     *  4.51634394893599368E+00,4.52721351415338499E+00,
     *  4.53796620232542800E+00,4.54860450019776842E+00,
     *  4.55913081598724211E+00 /
      DATA C(97),C(98),C(99),C(100) / 4.56954748265390877E+00,
     *  4.57985676100442424E+00,4.59006084263707730E+00,
     *  4.60016185273808740E+00 /
      DATA B(1),B(2),B(3),B(4),B(5),B(6) / 8.33333333333333333E-02,-
     *  8.33333333333333333E-03,3.96825396825396825E-03,-
     *  4.16666666666666666E-03,7.57575757575757576E-03,-
     *  2.10927960927960928E-02 /
C     ***FIRST EXECUTABLE STATEMENT  PSIXN
      IF (N.GT.100) GO TO 10
      PSIXN = C(N)
      RETURN
   10 CONTINUE
      WDTOL = R1MACH(4)
      FN = FLOAT(N)
      AX = 1.0E0
      AK = 2.0E0
      S = -0.5E0/FN
      RFN2 = 1.0E0/(FN*FN)
      DO 20 K = 1, 6
        AX = AX*RFN2
        TRM = -B(K)*AX
        IF (ABS(TRM).LT.WDTOL) GO TO 30
        S = S+TRM
        AK = AK+2.0E0
   20 CONTINUE
   30 CONTINUE
      PSIXN = S+ALOG(FN)
      RETURN
      END
*DECK R1MACH
      REAL FUNCTION R1MACH(I)
C***BEGIN PROLOGUE  R1MACH
C***DATE WRITTEN   790101   (YYMMDD)
C***REVISION DATE  890213   (YYMMDD)
C***CATEGORY NO.  R1
C***KEYWORDS  LIBRARY=SLATEC,TYPE=SINGLE PRECISION(R1MACH-S D1MACH-D),
C             MACHINE CONSTANTS
C***AUTHOR  FOX, P. A., (BELL LABS)
C           HALL, A. D., (BELL LABS)
C           SCHRYER, N. L., (BELL LABS)
C***PURPOSE  Returns single precision machine dependent constants
C***DESCRIPTION
C
C   R1MACH can be used to obtain machine-dependent parameters
C   for the local machine environment.  It is a function
C   subroutine with one (input) argument, and can be called
C   as follows, for example
```

```
C
C        A = R1MACH(I)
C
C   where I=1,...,5.  The (output) value of A above is
C   determined by the (input) value of I.  The results for
C   various values of I are discussed below.
C
C   Single-Precision Machine Constants
C   R1MACH(1) = B**(EMIN-1), the smallest positive magnitude.
C   R1MACH(2) = B**EMAX*(1 - B**(-T)), the largest magnitude.
C   R1MACH(3) = B**(-T), the smallest relative spacing.
C   R1MACH(4) = B**(1-T), the largest relative spacing.
C   R1MACH(5) = LOG10(B)
C
C   Assume single precision numbers are represented in the T-digit,
C   base-B form
C
C              sign (B**E)*( (X(1)/B) + ... + (X(T)/B**T) )
C
C   where 0 .LE. X(I) .LT. B for I=1,...,T, 0 .LT. X(1), and
C   EMIN .LE. E .LE. EMAX.
C
C   The values of B, T, EMIN and EMAX are provided in I1MACH as
C   follows:
C   I1MACH(10) = B, the base.
C   I1MACH(11) = T, the number of base-B digits.
C   I1MACH(12) = EMIN, the smallest exponent E.
C   I1MACH(13) = EMAX, the largest exponent E.
C
C   To alter this function for a particular environment,
C   the desired set of DATA statements should be activated by
C   removing the C from column 1.  Also, the values of
C   R1MACH(1) - R1MACH(4) should be checked for consistency
C   with the local operating system.
C
C***REFERENCES  FOX, P.A., HALL, A.D., SCHRYER, N.L, *FRAMEWORK FOR
C                 A PORTABLE LIBRARY*, ACM TRANSACTIONS ON MATHE-
C                 MATICAL SOFTWARE, VOL. 4, NO. 2, JUNE 1978,
C                 PP. 177-188.
C***ROUTINES CALLED  XERROR
C***END PROLOGUE  R1MACH
C
      INTEGER SMALL(2)
      INTEGER LARGE(2)
      INTEGER RIGHT(2)
      INTEGER DIVER(2)
      INTEGER LOG10(2)
C
      REAL RMACH(5)
      SAVE RMACH
C
      EQUIVALENCE (RMACH(1),SMALL(1))
      EQUIVALENCE (RMACH(2),LARGE(1))
      EQUIVALENCE (RMACH(3),RIGHT(1))
```

```
      EQUIVALENCE (RMACH(4),DIVER(1))
      EQUIVALENCE (RMACH(5),LOG10(1))
C
C     MACHINE CONSTANTS FOR THE AMIGA
C     ABSOFT FORTRAN COMPILER USING THE 68020/68881 COMPILER OPTION
C
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7F7FFFFF' /
C     DATA RIGHT(1) / Z'33800000' /
C     DATA DIVER(1) / Z'34000000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE AMIGA
C     ABSOFT FORTRAN COMPILER USING SOFTWARE FLOATING POINT
C
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7EFFFFFF' /
C     DATA RIGHT(1) / Z'33800000' /
C     DATA DIVER(1) / Z'34000000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE APOLLO
C
C     DATA SMALL(1) / 16#00800000 /
C     DATA LARGE(1) / 16#7FFFFFFF /
C     DATA RIGHT(1) / 16#33800000 /
C     DATA DIVER(1) / 16#34000000 /
C     DATA LOG10(1) / 16#3E9A209B /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 1700 SYSTEM
C
C     DATA RMACH(1) / Z400800000 /
C     DATA RMACH(2) / Z5FFFFFFFF /
C     DATA RMACH(3) / Z4E9800000 /
C     DATA RMACH(4) / Z4EA800000 /
C     DATA RMACH(5) / Z500E730E8 /
C
C     MACHINE CONSTANTS FOR THE BURROUGHS 5700/6700/7700 SYSTEMS
C
C     DATA RMACH(1) / O1771000000000000 /
C     DATA RMACH(2) / O0777777777777777 /
C     DATA RMACH(3) / O1311000000000000 /
C     DATA RMACH(4) / O1301000000000000 /
C     DATA RMACH(5) / O1157163034761675 /
C
C     MACHINE CONSTANTS FOR THE CDC 170/180 SERIES USING NOS/VE
C
C     DATA RMACH(1) / Z"3001800000000000" /
C     DATA RMACH(2) / Z"4FFEFFFFFFFFFFFE" /
C     DATA RMACH(3) / Z"3FD2800000000000" /
C     DATA RMACH(4) / Z"3FD3800000000000" /
C     DATA RMACH(5) / Z"3FFF9A209A84FBCF" /
C
```

```
C     MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES
C
C     DATA RMACH(1) / 00564000000000000000B /
C     DATA RMACH(2) / 37767777777777777776B /
C     DATA RMACH(3) / 16414000000000000000B /
C     DATA RMACH(4) / 16424000000000000000B /
C     DATA RMACH(5) / 17164642023241175720B /
C
C     MACHINE CONSTANTS FOR THE CELERITY C1260
C
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7F7FFFFF' /
C     DATA RIGHT(1) / Z'33800000' /
C     DATA DIVER(1) / Z'34000000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE CONVEX C-1
C
C     DATA SMALL(1) / '00800000'X /
C     DATA LARGE(1) / '7FFFFFFF'X /
C     DATA RIGHT(1) / '34800000'X /
C     DATA DIVER(1) / '35000000'X /
C     DATA LOG10(1) / '3F9A209B'X /
C
C     MACHINE CONSTANTS FOR THE CRAY-1
C
C     DATA RMACH(1) / 200034000000000000000B /
C     DATA RMACH(2) / 577767777777777777776B /
C     DATA RMACH(3) / 377224000000000000000B /
C     DATA RMACH(4) / 377234000000000000000B /
C     DATA RMACH(5) / 377774642023241175720B /
C
C     MACHINE CONSTANTS FOR THE DATA GENERAL ECLIPSE S/200
C
C     NOTE - IT MAY BE APPROPRIATE TO INCLUDE THE FOLLOWING CARD -
C     STATIC RMACH(5)
C
C     DATA SMALL /    20K,       0 /
C     DATA LARGE / 77777K, 177777K /
C     DATA RIGHT / 35420K,       0 /
C     DATA DIVER / 36020K,       0 /
C     DATA LOG10 / 40423K,  42023K /
C
C     MACHINE CONSTANTS FOR THE HARRIS 220
C
C     DATA SMALL(1), SMALL(2) / '20000000, '00000201 /
C     DATA LARGE(1), LARGE(2) / '37777777, '00000177 /
C     DATA RIGHT(1), RIGHT(2) / '20000000, '00000352 /
C     DATA DIVER(1), DIVER(2) / '20000000, '00000353 /
C     DATA LOG10(1), LOG10(2) / '23210115, '00000377 /
C
```

```
C     MACHINE CONSTANTS FOR THE HONEYWELL 600/6000 SERIES
C
C     DATA RMACH(1) / O402400000000 /
C     DATA RMACH(2) / O376777777777 /
C     DATA RMACH(3) / O714400000000 /
C     DATA RMACH(4) / O716400000000 /
C     DATA RMACH(5) / O776464202324 /
C
C     MACHINE CONSTANTS FOR THE HP 2100
C     3 WORD DOUBLE PRECISION WITH FTN4
C
C     DATA SMALL(1), SMALL(2) / 40000B,       1 /
C     DATA LARGE(1), LARGE(2) / 77777B, 177776B /
C     DATA RIGHT(1), RIGHT(2) / 40000B,    325B /
C     DATA DIVER(1), DIVER(2) / 40000B,    327B /
C     DATA LOG10(1), LOG10(2) / 46420B,  46777B /
C
C     MACHINE CONSTANTS FOR THE HP 2100
C     4 WORD DOUBLE PRECISION WITH FTN4
C
C     DATA SMALL(1), SMALL(2) / 40000B,       1 /
C     DATA LARGE91), LARGE(2) / 77777B, 177776B /
C     DATA RIGHT(1), RIGHT(2) / 40000B,    325B /
C     DATA DIVER(1), DIVER(2) / 40000B,    327B /
C     DATA LOG10(1), LOG10(2) / 46420B,  46777B /
C
C     MACHINE CONSTANTS FOR THE HP 9000
C
C     DATA SMALL(1) / 000040000000B /
C     DATA LARGE(1) / 17677777777B /
C     DATA RIGHT(1) / 06340000000B /
C     DATA DIVER(1) / 06400000000B /
C     DATA LOG10(1) / 07646420233B /
C
C     MACHINE CONSTANTS FOR THE ELXSI 6400
C       ASSUMING REAL*4 IS THE DEFAULT REAL
C
C     DATA SMALL(1) / '00800000'X /
C     DATA LARGE(1) / '7F7FFFFF'X /
C     DATA RIGHT(1) / '33800000'X /
C     DATA DIVER(1) / '34000000'X /
C     DATA LOG10(1) / '3E9A209B'X /
C
C     MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C     THE XEROX SIGMA 5/7/9, THE SEL SYSTEMS 85/86  AND
C     THE PERKIN ELMER (INTERDATA) 7/32.
C
C     DATA RMACH(1) / Z00100000 /
C     DATA RMACH(2) / Z7FFFFFFF /
C     DATA RMACH(3) / Z3B100000 /
C     DATA RMACH(4) / Z3C100000 /
C     DATA RMACH(5) / Z41134413 /
C
```

```
C     MACHINE CONSTANTS FOR THE IBM PC
C
C     DATA SMALL(1) /     8420761 /
C     DATA LARGE(1) /  2139081118 /
C     DATA RIGHT(1) /   863997169 /
C     DATA DIVER(1) /   872385777 /
C     DATA LOG10(1) /  1050288283 /
C
C     MACHINE CONSTANTS FOR THE IBM RS 6000
C
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7F7FFFFF' /
C     DATA RIGHT(1) / Z'33800000' /
C     DATA DIVER(1) / Z'34000000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE PDP-10 (KA OR KI PROCESSOR)
C
C     DATA RMACH(1) / "000400000000 /
C     DATA RMACH(2) / "377777777777 /
C     DATA RMACH(3) / "146400000000 /
C     DATA RMACH(4) / "147400000000 /
C     DATA RMACH(5) / "177464202324 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C     32-BIT INTEGERS (EXPRESSED IN INTEGER AND OCTAL).
C
C     DATA SMALL(1) /    8388608 /
C     DATA LARGE(1) / 2147483647 /
C     DATA RIGHT(1) /  880803840 /
C     DATA DIVER(1) /  889192448 /
C     DATA LOG10(1) / 1067065499 /
C
C     DATA RMACH(1) / O000040000000 /
C     DATA RMACH(2) / O017777777777 /
C     DATA RMACH(3) / O006440000000 /
C     DATA RMACH(4) / O006500000000 /
C     DATA RMACH(5) / O007746420233 /
C
C     MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C     16-BIT INTEGERS  (EXPRESSED IN INTEGER AND OCTAL).
C
C     DATA SMALL(1), SMALL(2) /   128,     0 /
C     DATA LARGE(1), LARGE(2) / 32767,    -1 /
C     DATA RIGHT(1), RIGHT(2) / 13440,     0 /
C     DATA DIVER(1), DIVER(2) / 13568,     0 /
C     DATA LOG10(1), LOG10(2) / 16282,  8347 /
C
C     DATA SMALL(1), SMALL(2) / O000200, O000000 /
C     DATA LARGE(1), LARGE(2) / O077777, O177777 /
C     DATA RIGHT(1), RIGHT(2) / O032200, O000000 /
C     DATA DIVER(1), DIVER(2) / O032400, O000000 /
C     DATA LOG10(1), LOG10(2) / O037632, O020233 /
C
```

```
C     MACHINE CONSTANTS FOR THE SILICON GRAPHICS IRIS
C
c     data rmach(1) / 1.17549 424 e-38 /
c     data rmach(2) / 3.40282 356 e+38 /
c     data rmach(3) / 1.19209 290 e-07 /
c     data rmach(4) / 2.38418 579 e-07 /
c     data rmach(5) / 0.30103 001 /
C
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7F7FFFFF' /
C     DATA RIGHT(1) / Z'34000000' /
C     DATA DIVER(1) / Z'34800000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE SUN
C
C     DATA SMALL(1) / Z'00800000' /
C     DATA LARGE(1) / Z'7F7FFFFF' /
C     DATA RIGHT(1) / Z'33800000' /
C     DATA DIVER(1) / Z'34000000' /
C     DATA LOG10(1) / Z'3E9A209B' /
C
C     MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES
C
C     DATA RMACH(1) / O000400000000 /
C     DATA RMACH(2) / O377777777777 /
C     DATA RMACH(3) / O146400000000 /
C     DATA RMACH(4) / O147400000000 /
C     DATA RMACH(5) / O177464202324 /
C
C     MACHINE CONSTANTS FOR THE VAX 11/780
C     (EXPRESSED IN INTEGER AND HEXADECIMAL)
C     THE HEX FORMAT BELOW MAY NOT BE SUITABLE FOR UNIX SYSTEMS
C     THE INTEGER FORMAT SHOULD BE OK FOR UNIX SYSTEMS
C
      DATA SMALL(1) /       128 /
      DATA LARGE(1) /    -32769 /
      DATA RIGHT(1) /     13440 /
      DATA DIVER(1) /     13568 /
      DATA LOG10(1) / 547045274 /
C
C     DATA SMALL(1) / Z00000080 /
C     DATA LARGE(1) / ZFFFF7FFF /
C     DATA RIGHT(1) / Z00003480 /
C     DATA DIVER(1) / Z00003500 /
C     DATA LOG10(1) / Z209B3F9A /
C
C     MACHINE CONSTANTS FOR THE Z80 MICROPROCESSOR
C
C     DATA SMALL(1), SMALL(2) /     0,    256/
C     DATA LARGE(1), LARGE(2) /    -1,   -129/
C     DATA RIGHT(1), RIGHT(2) /     0,  26880/
C     DATA DIVER(1), DIVER(2) /     0,  27136/
C     DATA LOG10(1), LOG10(2) /  8347,  32538/
C
C
```

```
C***FIRST EXECUTABLE STATEMENT  R1MACH
      IF (I .LT. 1  .OR.  I .GT. 5)
     1   CALL XERR ('R1MACH -- I OUT OF BOUNDS')
C
      R1MACH = RMACH(I)
      RETURN
C
      END
*DECK,SSORT
      SUBROUTINE SSORT (X,Y,N,KFLAG)
C     ***BEGIN PROLOGUE  SSORT
C     ***DATE WRITTEN   761101   (YYMMDD)
C     ***REVISION DATE  820801   (YYMMDD)
C     ***CATEGORY NO.  M1
C     ***KEYWORDS  QUICKSORT,SINGLETON QUICKSORT,SORT,SORTING
C     ***AUTHOR  JONES, R. E., (SLA)
C     WISNIEWSKI, J. A., (SLA)
C     ***PURPOSE  SSORT SORTS ARRAY X AND OPTIONALLY MAKES THE SAME
C     INTERCHANGES IN ARRAY Y.  THE ARRAY X MAY BE SORTED IN
C     INCREASING ORDER OR DECREASING ORDER.  A SLIGHTLY MODIFIED
C     QUICKSORT ALGORITHM IS USED.
C     ***DESCRIPTION
C     WRITTEN BY RONDALL E. JONES
C     MODIFIED BY JOHN A. WISNIEWSKI TO USE THE SINGLETON QUICKSORT
C     ALGORITHM.  DATE 18 NOVEMBER 1976.
C
C     ABSTRACT
C     SSORT SORTS ARRAY X AND OPTIONALLY MAKES THE SAME
C     INTERCHANGES IN ARRAY Y.  THE ARRAY X MAY BE SORTED IN
C     INCREASING ORDER OR DECREASING ORDER.  A SLIGHTLY MODIFIED
C     QUICKSORT ALGORITHM IS USED.
C
C     REFERENCE
C     SINGLETON, R. C., ALGORITHM 347, AN EFFICIENT ALGORITHM FOR
C     SORTING WITH MINIMAL STORAGE, CACM,12(3),1969,185-7.
C
C     DESCRIPTION OF PARAMETERS
C     X - ARRAY OF VALUES TO BE SORTED   (USUALLY ABSCISSAS)
C     Y - ARRAY TO BE (OPTIONALLY) CARRIED ALONG
C     N - NUMBER OF VALUES IN ARRAY X TO BE SORTED
C     KFLAG - CONTROL PARAMETER
C     =2  MEANS SORT X IN INCREASING ORDER AND CARRY Y ALONG.
C     =1  MEANS SORT X IN INCREASING ORDER (IGNORING Y)
C     =-1 MEANS SORT X IN DECREASING ORDER (IGNORING Y)
C     =-2 MEANS SORT X IN DECREASING ORDER AND CARRY Y ALONG.
C     ***REFERENCES  SINGLETON,R.C., ALGORITHM 347, AN EFFICIENT ALGORIT
C     FOR SORTING WITH MINIMAL STORAGE, CACM,12(3),1969,
C     185-7.
C     ***ROUTINES CALLED  XERROR
C     ***END PROLOGUE  SSORT
      DIMENSION X(N),Y(N),IL(21),IU(21)
C     ***FIRST EXECUTABLE STATEMENT  SSORT
      NN = N
      IF (NN.GE.1) GO TO 10
```

```
      CALL XERR (
     *  'SSORT- THE NUMBER OF VALUES TO BE SORTED WAS NOT POSITIVE.')
      RETURN
   10 KK = IABS(KFLAG)
      IF ((KK.EQ.1).OR.(KK.EQ.2)) GO TO 20
      CALL XERR (
     *  'SSORT- THE SORT CONTROL PARAMETER, K, WAS NOT 2, 1, -1, OR -2.'
     *  )
      RETURN
C
C     ALTER ARRAY X TO GET DECREASING ORDER IF NEEDED
C
   20 IF (KFLAG.GE.1) GO TO 40
      DO 30 I = 1, NN
   30   X(I) = -X(I)
   40 GO TO (50,180), KK
C
C     SORT X ONLY
C
   50 CONTINUE
      M = 1
      I = 1
      J = NN
      R = .375
   60 IF (I.EQ.J) GO TO 140
      IF (R.GT..5898437) GO TO 70
      R = R+3.90625E-2
      GO TO 80
   70 R = R-.21875
   80 K = I
C     SELECT A CENTRAL ELEMENT OF THE
C     ARRAY AND SAVE IT IN LOCATION T
      IJ = I+IFIX(FLOAT(J-I)*R)
      T = X(IJ)
C     IF FIRST ELEMENT OF ARRAY IS GREATER
C     THAN T, INTERCHANGE WITH T
      IF (X(I).LE.T) GO TO 90
      X(IJ) = X(I)
      X(I) = T
      T = X(IJ)
   90 L = J
C     IF LAST ELEMENT OF ARRAY IS LESS THAN
C     T, INTERCHANGE WITH T
      IF (X(J).GE.T) GO TO 110
      X(IJ) = X(J)
      X(J) = T
      T = X(IJ)
C     IF FIRST ELEMENT OF ARRAY IS GREATER
C     THAN T, INTERCHANGE WITH T
      IF (X(I).LE.T) GO TO 110
      X(IJ) = X(I)
      X(I) = T
      T = X(IJ)
      GO TO 110
```

```
  100 TT = X(L)
      X(L) = X(K)
      X(K) = TT
C     FIND AN ELEMENT IN THE SECOND HALF OF
C     THE ARRAY WHICH IS SMALLER THAN T
  110 L = L-1
      IF (X(L).GT.T) GO TO 110
C     FIND AN ELEMENT IN THE FIRST HALF OF
C     THE ARRAY WHICH IS GREATER THAN T
  120 K = K+1
      IF (X(K).LT.T) GO TO 120
C     INTERCHANGE THESE ELEMENTS
      IF (K.LE.L) GO TO 100
C     SAVE UPPER AND LOWER SUBSCRIPTS OF
C     THE ARRAY YET TO BE SORTED
      IF (L-I.LE.J-K) GO TO 130
      IL(M) = I
      IU(M) = L
      I = K
      M = M+1
      GO TO 150
  130 IL(M) = K
      IU(M) = J
      J = L
      M = M+1
      GO TO 150
C     BEGIN AGAIN ON ANOTHER PORTION OF
C     THE UNSORTED ARRAY
  140 M = M-1
      IF (M.EQ.0) GO TO 310
      I = IL(M)
      J = IU(M)
  150 IF (J-I.GE.1) GO TO 80
      IF (I.EQ.1) GO TO 60
      I = I-1
  160 I = I+1
      IF (I.EQ.J) GO TO 140
      T = X(I+1)
      IF (X(I).LE.T) GO TO 160
      K = I
  170 X(K+1) = X(K)
      K = K-1
      IF (T.LT.X(K)) GO TO 170
      X(K+1) = T
      GO TO 160
C
C     SORT X AND CARRY Y ALONG
C
  180 CONTINUE
      M = 1
      I = 1
      J = NN
      R = .375
```

```
  190 IF (I.EQ.J) GO TO 270
      IF (R.GT..5898437) GO TO 200
      R = R+3.90625E-2
      GO TO 210
  200 R = R-.21875
  210 K = I
C     SELECT A CENTRAL ELEMENT OF THE
C     ARRAY AND SAVE IT IN LOCATION T
      IJ = I+IFIX(FLOAT(J-I)*R)
      T = X(IJ)
      TY = Y(IJ)
C     IF FIRST ELEMENT OF ARRAY IS GREATER
C     THAN T, INTERCHANGE WITH T
      IF (X(I).LE.T) GO TO 220
      X(IJ) = X(I)
      X(I) = T
      T = X(IJ)
      Y(IJ) = Y(I)
      Y(I) = TY
      TY = Y(IJ)
  220 L = J
C     IF LAST ELEMENT OF ARRAY IS LESS THAN
C     T, INTERCHANGE WITH T
      IF (X(J).GE.T) GO TO 240
      X(IJ) = X(J)
      X(J) = T
      T = X(IJ)
      Y(IJ) = Y(J)
      Y(J) = TY
      TY = Y(IJ)
C     IF FIRST ELEMENT OF ARRAY IS GREATER
C     THAN T, INTERCHANGE WITH T
      IF (X(I).LE.T) GO TO 240
      X(IJ) = X(I)
      X(I) = T
      T = X(IJ)
      Y(IJ) = Y(I)
      Y(I) = TY
      TY = Y(IJ)
      GO TO 240
  230 TT = X(L)
      X(L) = X(K)
      X(K) = TT
      TTY = Y(L)
      Y(L) = Y(K)
      Y(K) = TTY
C     FIND AN ELEMENT IN THE SECOND HALF OF
C     THE ARRAY WHICH IS SMALLER THAN T
  240 L = L-1
      IF (X(L).GT.T) GO TO 240
C     FIND AN ELEMENT IN THE FIRST HALF OF
C     THE ARRAY WHICH IS GREATER THAN T
  250 K = K+1
      IF (X(K).LT.T) GO TO 250
```

```
C     INTERCHANGE THESE ELEMENTS
      IF (K.LE.L) GO TO 230
C     SAVE UPPER AND LOWER SUBSCRIPTS OF
C     THE ARRAY YET TO BE SORTED
      IF (L-I.LE.J-K) GO TO 260
      IL(M) = I
      IU(M) = L
      I = K
      M = M+1
      GO TO 280
  260 IL(M) = K
      IU(M) = J
      J = L
      M = M+1
      GO TO 280
C     BEGIN AGAIN ON ANOTHER PORTION OF
C     THE UNSORTED ARRAY
  270 M = M-1
      IF (M.EQ.0) GO TO 310
      I = IL(M)
      J = IU(M)
  280 IF (J-I.GE.1) GO TO 210
      IF (I.EQ.1) GO TO 190
      I = I-1
  290 I = I+1
      IF (I.EQ.J) GO TO 270
      T = X(I+1)
      TY = Y(I+1)
      IF (X(I).LE.T) GO TO 290
      K = I
  300 X(K+1) = X(K)
      Y(K+1) = Y(K)
      K = K-1
      IF (T.LT.X(K)) GO TO 300
      X(K+1) = T
      Y(K+1) = TY
      GO TO 290
C
C     CLEAN UP
C
  310 IF (KFLAG.GE.1) RETURN
      DO 320 I = 1, NN
  320   X(I) = -X(I)
      RETURN
      END
*DECK,XERR
      SUBROUTINE XERR (MESG)
      CHARACTER*100 MESG
      WRITE (4,10) MESG
      STOP
   10 FORMAT(A)
      END
```

The file RT4ISO.DAT contains the isotope library data. Other default data are included in RT4INT.DAT. These two files are read upon execution of RADTRAN 4.

```
60,11   && NISOTP,NPRP

&& ISONAM
H3WTR     H3GAS     C14ORG    C14GAS    P32       S35       CA45      CR51
MN54      FE55      FE59      CO58      CO60      ZN65      GA67      KR85
SR89      SR90      Y91       ZR95      NB94      NB95      MO99      TC99
RU103     RU106     SB125     TE125M    TE127M    TE127     TE129M    TE129
I125      I129      I131      XE133     CS134     CS137     CE141     CE144
PM147     SM151     EU152     EU154     EU155     U233      U235      U238
NP237     PU236     PU238     PU239     PU240     PU241     PU242     AM241
AM243     CM242     CM244     CF252

&& ... Correspondence for isotope arrays:
&&      1. HALF LIFE in days (TABHLF) source: ICRP 38
&&      2. PHOTON ENERGY (PHTENG) source: ICRP 38
&&      3. CLOUD DOSE FACTOR (CLDOSF) rem-m**3/Ci-sec  source: DOE/EH--0070
&&      4. INHALATION VALUE - EFFECTIVE DOSE EQUIVALENT - 50 YR -
(RPCVAL(*,4))
&&                                    rem/Ci    source: WIPP DOE 176 or DOE/EH
71
&&      5. INGESTION VALUE - EFFECTIVE DOSE EQUIVALENT - 50 YR -
(INGVAL(*,4))
&&                                    rem/Ci    source: WIPP DOE 176 or DOE/EH
71
&&      6. FOOD TRANSFER (FDTRAN) source:
&&      7. SOIL TRANSFER (SLTRAN) source:
&&      8. DEPOSTION VELOCITY (VELDEP) meters/sec
&&      9. 1 indicates SHORT HALF-LIFE (< 365 days)     source: ICRP 38
&&         2 indicates LONG HALF-LIFE low L.E.T.
&&         3 indicates LONG HALF-LIFE high L.E.T. (alpha emitters)
&&     10. INHALATION VALUE - LUNG - 1 YR - (RPCVAL(*,1)) rem/Ci
&&                                          source: WIPP DOE 176
&&     11. INHALATION VALUE - MARROW - 1 YR - (RPCVAL(*,2)) rem/Ci
&&                                          source: WIPP DOE 176
&& ...
&&
&& ... ** NOTE THE FOLLOWING **
&&
&& ...  SR90 has no gamma so the photon energy is set to 0.0
&&      MO99 photon energy includes the photon energy of its short half-
life
&&           daughter TC99M  (1.50E-01+1.26E-01)
&&      RU103 photon energy includes the photon energy of its short half-
life
```

```
&&            daughter RH103  (4.68E-01+1.75E-03)
&&      RU106 has no gamma, but does include the photon energy of its short
&&            half-life daughter RH106  (0.0+2.01E-01)
&&      TC99 has no gamma so the photon energy is set to 0.0
&&      CS137 has no gamma, but does include the photon energy of its short
&&            half-life daughter BA137M  (0.0+5.96E-01)
&&      CE144 photon energy includes the photon energy of its short half-
life
&&            daughter PR144  (2.07E-02+3.18E-02)  (PR144M was not included
&&            since it is such a small fraction of the CE144 decay and
PR144M
&&            decays to PR144.)
&&
&& ... Dose values are based on 0.3 micron AMAD particles except H3GAS,
&&     C14GAS, and KR85 which are gases; H3WTR and C14ORG are 1.0
&&     micron AMAD particles.

4.51E+03,0.00E+00,0.00E+00,6.30E+01,6.3E+01,0.00E+00,0.00E+00,0.00E+00,2,
    6.00E+01,6.00E+01   && H3WTR
4.51E+03,0.00E+00,0.00E+00,4.40E-03,0.0E+00,0.00E+00,0.00E+00,0.00E+00,2,
    1.20E+02,1.20E+02   && H3GAS
2.09E+06,0.00E+00,0.00E+00,2.10E+03,2.1E+03,0.00E+00,0.00E+00,1.00E-02,2,
    2.10E+03,2.10E+03   && C14ORG
2.09E+06,0.00E+00,0.00E+00,2.40E+01,0.0E+00,0.00E+00,0.00E+00,0.00E+00,2,
    2.40E+01,2.40E+01   && C14GAS
1.43E+01,0.00E+00,0.00E+00,2.20E+04,7.7E+03,0.00E+00,0.00E+00,1.00E-02,1,
    1.60E+05,1.10E+04   && P32
8.74E+01,0.00E+00,0.00E+00,4.10E+03,6.5E+02,0.00E+00,0.00E+00,1.00E-02,1,
    0.00E+00,0.00E+00   && S35
1.63E+02,4.35E-08,2.59E-11,9.50E+03,3.0E+03,0.00E+00,0.00E+00,1.00E-02,1,
    0.00E+00,0.00E+00   && CA45
2.77E+01,3.26E-02,5.01E-03,4.50E+02,1.3E+02,0.00E+00,0.00E+00,1.00E-02,1,
    0.00E+00,0.00E+00   && CR51
3.13E+02,8.35E-01,1.39E-01,1.00E+04,2.7E+03,0.00E+00,0.00E+00,1.00E-02,1,
    4.10E+04,5.30E+03   && MN54
9.86E+02,1.69E-03,3.64E-06,1.60E+03,5.8E+02,0.00E+00,0.00E+00,1.00E-02,2,
    5.80E+03,1.80E+02   && FE55
4.45E+01,1.19E+00,1.96E-01,1.40E+04,6.6E+03,0.00E+00,0.00E+00,1.00E-02,1,
    8.60E+04,5.40E+03   && FE59
7.08E+01,9.75E-01,1.60E-01,1.30E+04,3.5E+03,0.00E+00,0.00E+00,1.00E-02,1,
    1.00E+05,5.30E+03   && CO58
1.93E+03,2.50E+00,4.12E-01,2.80E+05,2.6E+04,0.00E+00,0.00E+00,1.00E-02,2,
    7.90E+05,3.80E+04   && CO60
2.44E+02,5.84E-01,9.57E-02,2.80E+04,1.4E+04,0.00E+00,0.00E+00,1.00E-02,1,
    9.70E+04,1.10E+04   && ZN65
3.26E+00,1.58E-01,2.34E-02,5.90E+02,7.2E+02,0.00E+00,0.00E+00,1.00E-02,1,
    0.00E+00,0.00E+00   && GA67
3.92E+03,2.21E-03,3.55E-04,0.00E+00,0.0E+00,0.00E+00,0.00E+00,0.00E+00,2,
    0.00E+00,0.00E+00   && KR85
5.05E+01,8.45E-05,2.27E-05,6.70E+04,8.7E+03,0.00E+00,0.00E+00,1.00E-02,1,
    5.30E+05,2.90E+02   && SR89
```

```
1.06E+04,0.00E+00,0.00E+00,2.40E+06,1.3E+05,0.00E+00,0.00E+00,1.00E-02,2,
    4.50E+06,3.80E+03    && SR90
5.85E+01,3.61E-03,5.93E-04,8.00E+04,8.9E+03,0.00E+00,0.00E+00,1.00E-02,1,
    6.20E+05,8.70E+02    && Y91
6.40E+01,7.39E-01,1.21E-01,3.20E+04,3.4E+03,0.00E+00,0.00E+00,1.00E-02,1,
    2.50E+05,7.40E+03    && ZR95
7.42E+06,1.57E+00,2.60E-01,6.00E+05,5.1E+03,0.00E+00,0.00E+00,1.00E-02,2,
    0.00E+00,0.00E+00    && NB94
3.52E+01,7.66E-01,1.26E-01,7.30E+03,2.2E+03,0.00E+00,0.00E+00,1.00E-02,1,
    5.30E+04,2.40E+03    && NB95
2.75E+00,2.76E-01,2.53E-02,4.50E+03,4.4E+03,0.00E+00,0.00E+00,1.00E-02,1,
    2.70E+04,1.80E+02    && MO99
7.78E+07,0.00E+00,8.40E-08,1.40E+04,1.3E+03,0.00E+00,0.00E+00,1.00E-02,2,
    1.00E+05,1.40E+02    && TC99
3.93E+01,4.70E-01,7.67E-02,1.30E+04,2.7E+03,0.00E+00,0.00E+00,1.00E-02,1,
    9.90E+04,1.70E+03    && RU103
3.68E+02,2.01E-01,0.00E+00,8.00E+05,2.1E+04,0.00E+00,0.00E+00,1.00E-02,2,
    4.30E+06,4.50E+03    && RU106
1.01E+03,4.30E-01,6.75E-02,1.70E+04,2.6E+03,0.00E+00,0.00E+00,1.00E-02,2,
    4.40E+04,5.50E+02    && SB125
5.80E+01,3.55E-02,1.53E-03,1.00E+04,3.4E+03,0.00E+00,0.00E+00,1.00E-02,1,
    0.00E+00,0.00E+00    && TE125M
1.09E+02,1.12E-02,4.94E-04,3.00E+04,7.9E+03,0.00E+00,0.00E+00,1.00E-02,1,
    2.10E+05,1.60E+04    && TE127M
3.90E-01,4.86E-03,7.67E-04,3.90E+02,6.9E+02,0.00E+00,0.00E+00,1.00E-02,1,
    2.70E+03,9.90E+00    && TE127
3.36E+01,9.66E-02,5.45E-03,3.40E+04,9.9E+03,0.00E+00,0.00E+00,1.00E-02,1,
    2.60E+05,9.60E+03    && TE129M
4.83E-02,5.91E-02,8.62E-03,1.10E+02,1.9E+02,0.00E+00,0.00E+00,1.00E-02,1,
    9.90E+02,1.80E+00    && TE129
6.01E+01,4.20E-02,1.77E-03,2.30E+04,3.8E+04,0.00E+00,0.00E+00,1.00E-02,1,
    6.80E+02,1.60E+02    && I125
5.73E+09,2.46E-02,1.32E-03,1.70E+05,2.8E+05,0.00E+00,0.00E+00,1.00E-02,2,
    1.70E+03,4.30E+02    && I129
8.04E+00,3.80E-01,6.05E-02,3.10E+04,5.3E+04,0.00E+00,0.00E+00,1.00E-02,1,
    4.00E+03,2.30E+02    && I131
5.25E+00,4.60E-02,5.58E-03,0.00E+00,0.0E+00,0.00E+00,0.00E+00,0.00E+00,1,
    0.00E+00,0.00E+00    && XE133
7.53E+02,1.55E+00,2.54E-01,4.60E+04,7.4E+04,0.00E+00,0.00E+00,1.00E-02,2,
    4.10E+04,3.90E+04    && CS134
1.10E+04,5.96E-01,0.00E+00,3.20E+04,5.0E+04,0.00E+00,0.00E+00,1.00E-02,2,
    3.10E+04,2.60E+04    && CS137
3.25E+01,7.61E-02,1.22E-02,1.40E+04,2.6E+03,0.00E+00,0.00E+00,1.00E-02,1,
    1.10E+05,4.30E+02    && CE141
2.84E+02,5.25E-02,2.88E-03,6.30E+05,2.0E+04,0.00E+00,0.00E+00,1.00E-02,1,
    3.60E+06,4.20E+03    && CE144
9.58E+02,4.37E-06,5.80E-07,6.00E+04,9.5E+02,0.00E+00,0.00E+00,1.00E-02,2,
    2.20E+05,5.30E+02    && PM147
3.29E+04,1.34E-05,1.45E-07,3.30E+04,3.4E+02,0.00E+00,0.00E+00,1.00E-02,2,
    2.10E+04,2.90E+03    && SM151
4.87E+03,1.14E+00,1.87E-01,2.60E+05,6.0E+03,0.00E+00,0.00E+00,1.00E-02,2,
```

```
    0.00E+00,0.00E+00    && EU154
3.21E+03,1.22E+00,2.06E-01,3.10E+05,9.1E+03,0.00E+00,0.00E+00,1.00E-02,2,
    0.00E+00,0.00E+00    && EU154
1.81E+03,6.05E-02,9.10E-03,4.80E+04,1.3E+03,0.00E+00,0.00E+00,1.00E-02,2,
    0.00E+00,0.00E+00    && EU155
5.79E+07,1.31E-03,3.80E-05,2.40E+08,2.7E+05,0.00E+00,0.00E+00,1.00E-02,3,
    4.00E+08,6.10E+03    && U233
2.57E+11,1.54E-01,2.44E-02,2.20E+08,2.5E+05,0.00E+00,0.00E+00,1.00E-02,3,
    3.60E+08,1.00E+04    && U235
1.63E+12,1.36E-03,1.65E-05,2.20E+08,2.3E+05,0.00E+00,0.00E+00,1.00E-02,3,
    3.50E+08,6.30E+03    && U238
7.82E+08,3.43E-02,3.64E-03,5.60E+08,3.9E+06,0.00E+00,0.00E+00,1.00E-02,3,
    1.00E+08,1.50E+07    && NP237
1.04E+03,2.09E-03,1.89E-05,2.40E+08,1.3E+06,0.00E+00,0.00E+00,1.00E-02,3,
    4.30E+08,1.00E+06    && PU236
3.21E+04,1.81E-03,1.40E-05,5.30E+08,3.8E+06,0.00E+00,0.00E+00,1.00E-02,3,
    4.50E+08,1.10E+06    && PU238
8.79E+06,7.96E-04,1.30E-05,5.70E+08,4.3E+06,0.00E+00,0.00E+00,1.00E-02,3,
    4.20E+08,1.10E+06    && PU239
2.39E+06,1.73E-03,1.37E-05,5.70E+08,4.3E+06,0.00E+00,0.00E+00,1.00E-02,3,
    4.20E+08,1.10E+06    && PU240
5.26E+03,2.54E-06,0.00E+00,9.90E+06,8.6E+04,0.00E+00,0.00E+00,1.00E-02,3,
    3.60E+05,1.30E+03    && PU241
1.37E+08,1.44E-03,1.16E-05,5.30E+08,4.1E+06,0.00E+00,0.00E+00,1.00E-02,3,
    4.00E+08,1.00E+06    && PU242
1.58E+05,3.24E-02,3.01E-03,5.90E+08,4.5E+06,0.00E+00,0.00E+00,1.00E-02,3,
    1.20E+08,1.70E+07    && AM241
2.70E+06,5.59E-02,8.11E-03,5.90E+08,4.5E+06,0.00E+00,0.00E+00,1.00E-02,3,
    1.10E+08,1.60E+07    && AM243
1.63E+02,1.83E-03,1.55E-05,2.30E+07,1.1E+05,0.00E+00,0.00E+00,1.00E-02,1,
    9.90E+07,8.80E+06    && CM242
6.62E+03,1.70E-03,1.33E-05,3.10E+08,2.3E+06,0.00E+00,0.00E+00,1.00E-02,3,
    1.20E+08,1.70E+07    && CM244
9.64E+02,1.20E-03,1.19E-05,2.40E+08,9.4E+05,0.00E+00,0.00E+00,1.00E-02,3,
    8.60E+08,2.30E+06    && CF252
```

```
20,1,10,18                                  && NSEV,NGROUP,NRAD,NAREAS
1,3,2,3,0                                   && IRNKC,IANA,IUOPT,ISEN,IPSQSB
.TRUE.                                      && URSKF
5*0.0                                       && COEF(1,J),J=1,5
7.42E-3,2.02E-2,6.17E-5,3.17E-8,0.0         && COEF(2,J),J=1,5
120*0.0                                     && ECNPYR,ECNMPT
21,38                                       && IPRAM,NPRAM
20,15,65,12                                 && MAXSEV,MAXGRP,MAXISO,MAXMAT
3,4                                         && NPOP,NPOP1
25                                          && NDLAB
10,11                                       && NDOSE,NDOSP1
15,11                                       && MAXRAD,MAXDSP
8                                           && NORG
365.                                        && RSPHLF
30  10  25                                  && MAXARS  ,MAXMOD ,MAXNDL
6.0, 719.0, 3861.0, 0.0                     && POPDEN
10*0.0  650*0.0                             && TABSPY ,FKMPS
520*0.0  300*0.0                            && RPCVAL ,RFRAC
520*0.0                                     && INGVAL
0.16,0.01,1                                 && CH1,CH2,ITRAIN
0.9,.05,.05,88.49,40.25,24.16,2.0,3.1,0.0,.011,0.0,0.0,0.0,50.0,20.0,0.0,
    100.,100.,2.,.08,.05,.85,470.,780.,2800.        && DNORML(1,J),J=1,25
0.9,.05,.05,64.37,40.25,24.16,5.,152.4,2.0,.033,10.,60.,2.0,100.,20.,4.0,
    100.,100.,3.0,0.0,1.0,0.0,1.0,5.0,5.0           && DNORML(2,J),J=1,25
0.9,.09,.01,16.09,8.06,3.20,2.0,45.7,2.0,.01,10.,0.0,0.0,50.,50.,24.,
    100.,100.,0.0,0.0,0.0,0.0,0.0,0.0,0.0           && DNORML(3,J),J=1,25
0.99,.009,.001,24.16,8.06,3.2,10.,61.,2.,.01,10.,0.0,0.0,50.,50.,48.,
    100.,100.,0.0,0.0,0.0,0.0,0.0,0.0,0.0           && DNORML(4,J),J=1,25
0.88,.1,.02,691.9,691.9,691.9,3.0,6.1,4.0,.0016,1.0,0.0,0.0,10.,50.,0.0,
    100.,100.,0.0,0.0,0.0,0.0,0.0,0.0,0.0           && DNORML(5,J),J=1,25
0.88,0.1,.02,691.9,691.9,691.9,3.0,15.2,4.0,.0008,1.0,0.0,0.0,1000.,50.,0.0
,
    100.,100.,78.,0.0,0.0,0.0,0.0,0.0,0.0           && DNORML(6,J),J=1,25
0.,0.6,0.4,88.49,56.34,24.16,1.0,2.13,2.0,.0004,.15,0.0,0.0,100.,10.,4.0,
    100.,100.,2.0,.08,0.65,0.25,470.,780.,2800.     && DNORML(7,J),J=1,25
0.,0.6,0.4,88.49,56.34,24.16,1.0,2.13,6.0,.0004,.15,0.0,0.0,100.,10.,10.,
    100.,100.,2.0,.08,0.65,0.25,470.,780.,2800.     && DNORML(8,J),J=1,25
0.,0.6,0.4,88.49,56.34,24.16,1.0,2.13,6.0,.0004,.15,0.0,0.0,100.,10.,10.,
    100.,100.,2.0,.08,0.65,0.25,470.,780.,2800.     && DNORML(9,J),J=1,25
0.,0.6,0.4,88.49,56.34,24.16,1.0,2.13,6.0,.0004,.15,0.0,0.0,100.,10.,4.0,
    100.,100.,2.0,.08,0.65,0.25,470.,780.,2800.     && DNORML(10,J),J=1,25
1.402E-07,1.451E-07,1.920E-10,1.746E-10,1.131E-09,
                1.131E-09,4*1.060E-06               && ARATMZ(I,1),I=1,10
2.681E-06,2.972E-06,3.410E-05,3.410E-04,1.262E-07,
                1.262E-07,4*7.046E-07               && ARATMZ(I,2),I=1,10
1.599E-05,1.302E-05,2.972E-04,2.972E-03,3.952E-08,
                3.952E-08,4*1.584E-06               && ARATMZ(I,3),I=1,10
&& ... SEVERITY FRACTIONS - SOURCE NUREG 0170 -  RURAL
```

```
0.462,0.302,0.176,0.0403,0.0118,6.47E-03,5.71E-04,1.13E-04,
                                                  12*0.0  && SEVFRC(1,I,1),I=1,20
0.356,0.214,0.385,0.0385,6.41E-03,6.48E-04,3.42E-04,6.41E-05,
                                                  12*0.0  && SEVFRC(2,I,1),I=1,20
0.0, 0.0, 0.0, 0.0, 0.0182, 0.253, 0.684, 4.56E-02,
                                                  12*0.0  && SEVFRC(3,I,1),I=1,20
0.0, 0.0, 0.0, 0.0, 0.0182, 0.253, 0.684, 4.56E-02,
                                                  12*0.0  && SEVFRC(4,I,1),I=1,20
0.323,0.323,0.0628,0.0155,0.121,8.42E-02,6.52E-02,4.26E-03,
                                                  12*0.0  && SEVFRC(5,I,1),I=1,20
0.323,0.323,0.0628,0.0155,0.121,8.42E-02,6.52E-02,4.26E-03,
                                                  12*0.0  && SEVFRC(6,I,1),I=1,20
0.55,0.36,0.07,0.016,2.8E-03,1.1E-03,8.5E-05,1.5E-05,
                                                  12*0.0  && SEVFRC(7,I,1),I=1,20
0.55,0.36,0.07,0.016,2.8E-03,1.1E-03,8.5E-05,1.5E-05,
                                                  12*0.0  && SEVFRC(8,I,1),I=1,20
0.55,0.36,0.07,0.016,2.8E-03,1.1E-03,8.5E-05,1.5E-05,
                                                  12*0.0  && SEVFRC(9,I,1),I=1,20
0.55,0.36,0.07,0.016,2.8E-03,1.1E-03,8.5E-05,1.5E-05,
                                                  12*0.0  &&
SEVFRC(10,I,1),I=1,20
&& ... SEVERITY FRACTIONS - SUBURBAN
0.435,0.285,0.221,0.0506,6.64E-03,1.74E-03,6.72E-05,5.93E-06,
                                                  12*0.0  && SEVFRC(1,I,2),I=1,20
0.313,0.188,0.451,0.0451,3.38E-03,1.63E-04,3.76E-05, 3.13E-06,
                                                  12*0.0  && SEVFRC(2,I,2),I=1,20
0.886,7.88E-02,2.01E-03,3.31E-02,9.24E-06,1.28E-04,3.47E-04,2.31E-05,
                                                  12*0.0  && SEVFRC(3,I,2),I=1,20
0.886,7.88E-02,2.01E-03,3.31E-02,9.24E-06,1.28E-04,3.47E-04,2.31E-05,
                                                  12*0.0  && SEVFRC(4,I,2),I=1,20
0.459,0.459,0.0396,9.77E-03,1.91E-02,1.33E-02,5.25E-05,3.42E-06,
                                                  12*0.0  && SEVFRC(5,I,2),I=1,20
0.459,0.459,0.0396,9.77E-03,1.91E-02,1.33E-02,5.25E-05,3.42E-06,
                                                  12*0.0  && SEVFRC(6,I,2),I=1,20
0.547,0.358,6.96E-02,2.04E-02,3.57E-03,1.40E-03,1.3E-04,2.30E-05,
                                                  12*0.0  && SEVFRC(7,I,2),I=1,20
0.547,0.358,6.96E-02,2.04E-02,3.57E-03,1.40E-03,1.3E-04,2.30E-05,
                                                  12*0.0  && SEVFRC(8,I,2),I=1,20
0.547,0.358,6.96E-02,2.04E-02,3.57E-03,1.40E-03,1.3E-04,2.30E-05,
                                                  12*0.0  && SEVFRC(9,I,2),I=1,20
0.547,0.358,6.96E-02,2.04E-02,3.57E-03,1.40E-03,1.3E-04,2.30E-05,
                                                  12*0.0  &&
SEVFRC(10,I,2),I=1,20
&& ... SEVERITY FRACTIONS - URBAN
0.583,0.382,2.78E-02,6.36E-03,7.42E-04,1.46E-04,1.13E-05,9.94E-07,
                                                  12*0.0  && SEVFRC(1,I,3),I=1,20
0.572,0.343,7.72E-02,7.72E-02,5.14E-04,1.86E-05,8.57E-06,7.15E-07,
                                                  12*0.0  && SEVFRC(2,I,3),I=1,20
0.915,8.14E-02,2.30E-04,7.79E-03, 0.0, 0.0, 0.0, 0.0,
                                                  12*0.0  && SEVFRC(3,I,3),I=1,20
0.915,8.14E-02,2.30E-04,7.79E-03, 0.0, 0.0, 0.0, 0.0,
```

```
                                                    12*0.0  && SEVFRC(4,I,3),I=1,20
0.407,0.407,7.91E-02,1.95E-02,5.08E-02,3.53E-02,8.328E-04,5.47E-05,
                                                    12*0.0  && SEVFRC(5,I,3),I=1,20
0.407,0.407,7.91E-02,1.95E-02,5.08E-02,3.53E-02,8.328E-04,5.47E-05,
                                                    12*0.0  && SEVFRC(6,I,3),I=1,20
0.552,0.361,7.03E-02,1.31E-02,2.30E-03,9.02E-04,5.55E-05,9.79E-06,
                                                    12*0.0  && SEVFRC(7,I,3),I=1,20
0.552,0.361,7.03E-02,1.31E-02,2.30E-03,9.02E-04,5.55E-05,9.79E-06,
                                                    12*0.0  && SEVFRC(8,I,3),I=1,20
0.552,0.361,7.03E-02,1.31E-02,2.30E-03,9.02E-04,5.55E-05,9.79E-06,
                                                    12*0.0  && SEVFRC(9,I,3),I=1,20
0.552,0.361,7.03E-02,1.31E-02,2.30E-03,9.02E-04,5.55E-05,9.79E-06,
                                                    12*0.0  &&
SEVFRC(10,I,3),I=1,20
&& ... DISPERSABILITY CATEGORIES
&&         1. NON-DISPERSABLE (SHIELDING ONLY)
&&         2. IMMOBILIZED EX: SPENT FUEL, HLW
&&         3. SINTERED-LOOSE CHUNKS
&&         4. LOOSE POWDER-LARGE EX: UO2, U308
&&         5. LOOSE POWDER-SMALL EX: UO2, U308
&&         6. SPENT FUEL PARTICULATE
&&         7. VOLATILE (SUBLIME & SOLUTION) EX: CRUD, CS137, UF6
&&         8. LIQUID EX: H3, RADIO-PHARMACEUTICALS
&&         9. OTHERS
&&        10. GAS EX: KR85
&&        11. FLAMMABLE EX: BITUMIN-ASPHALT MATRIX

&& ... AEROSOL FRACTIONS  - SOURCE: MODIFICATION 11/15/82
20*0.0                  &&  AERSOL(1,I),I=1,20
20*1.0E-0               &&  AERSOL(2,I),I=1,20
20*1.0E-02              &&  AERSOL(3,I),I=1,20
20*5.0E-02              &&  AERSOL(4,I),I=1,20
20*0.1                  &&  AERSOL(5,I),I=1,20
20*1.                   &&  AERSOL(6,I),I=1,20
20*1.                   &&  AERSOL(7,I),I=1,20
20*1.                   &&  AERSOL(8,I),I=1,20
20*1.                   &&  AERSOL(9,I),I=1,20
20*1.                   &&  AERSOL(10,I),I=1,20
20*1.                   &&  AERSOL(11,I),I=1,20
&& ... RESPIRABLE FRACTIONS  - SOURCE: MODIFICATION 11/15/82
20*0.0                  && RESP(1,I),I=1,20
20*.05                  && RESP(2,I),I=1,20
20*.05                  && RESP(3,I),I=1,20
20*.05                  && RESP(4,I),I=1,20
20*.05                  && RESP(5,I),I=1,20
20*.05                  && RESP(6,I),I=1,20
20*1.0                  && RESP(7,I),I=1,20
20*1.0                  && RESP(8,I),I=1,20
20*1.0                  && RESP(9,I),I=1,20
20*1.0                  && RESP(10,I),I=1,20
20*1.0                  && RESP(11,I),I=1,20
```

```
4.59E+02,1.53E+03,3.94E+03,1.25E+04,3.04E+04,6.85E+04,1.76E+05,4.45E+05,
   8.59E+05,2.55E+06,4.45E+06,1.03E+07,2.16E+07,5.52E+07,1.77E+08,4.89E+08,
   8.12E+08,1.35E+09,12*0.0    && AREADA
3.42E-03,1.72E-03,8.58E-04,3.42E-04,1.72E-04,8.58E-05,3.42E-05,1.72E-05,
   8.58E-06,3.42E-06,1.72E-06,8.58E-07,3.42E-07,1.72E-07,8.58E-08,5.42E-08,
   4.30E-08,3.42E-08,12*0.0    && DFLEV
3.05,6.1,9.1,12.2,15.2,30.5,61.,91.4,152.4,305.,5*0.    && RADIST RURAL
3.05,6.1,9.1,12.2,15.2,30.5,61.,91.4,152.4,305.,5*0.    && RADIST SUBURBAN
3.05,6.1,9.1,12.2,15.2,30.5,61.,91.4,152.4,305.,5*0.    && RADIST URBAN
0.5  1.0                          && PKGSZ1  PKGSZ2
6.0  1.0  0.87  0.018  4.0        && RPD RR RS RU FNOATT
8.6E-3  0.5   0.2   3.3E-4        && BDF XFARM CULVL BRATE
3.E-6 1.3E-5 7.E-6 2.2E-5 2.8E-4  && EXPLCF
1.2E-4,1.7E-4                     && WBLCF GNDSGE
1.E-5                             && TOL
12                                && LISTLH
18  6  4  32  3  3  7  7          && NKW1 NKW2 NKW3 NKW4 NKW5 NKW6 NKW7
NKW8
38                                && NDREV
1.,1.,   0.25,0.5,   2.,5.        && TBDIHD,TBDTHD,TBPPH
18, 6                             && NA NC
460.,1500.,3900.,1.3E4,3.E4,6.9E4,1.8E5,4.5E5,8.6E5,2.6E6,
         4.5E6,1.7E7,2.2E7,5.5E7,1.8E8,5.0E8,8.0E8,1.4E9  && PSAREA
6.E-3,1.7E-3,8.4E-4,1.7E-4,7.8E-5,2.8E-5,8.E-6,2.2E-6,9.E-7,
1.4E-7,7.E-8,1.1E-8,7.76E-9,2.24E-9,4.5E-10,1.13E-10,5.96E-11,2.76E-11, &&
A
4.E-3,1.3E-3,5.5E-4,1.3E-4,6.E-5,2.7E-5,1.E-5,3.5E-6,1.6E-6,
4.1E-7,2.2E-7,5.E-8,3.2E-8,1.1E-8,2.5E-9,7.24E-10,4.09E-10,2.08E-10,    &&
B
4.E-3,1.1E-3,5.7E-4,1.3E-4,6.7E-5,3.E-5,1.E-5,5.E-6,2.8E-6,
1.E-6,6.E-7,1.7E-7,1.3E-7,5.7E-8,1.7E-8,6.32E-9,4.01E-9,2.33E-9,        &&
C
4.3E-3,1.3E-3,6.5E-4,1.8E-4,9.5E-5,4.3E-5,1.8E-5,8.5E-6,5.E-6,
1.9E-6,1.3E-6,4.E-7,3.E-7,1.5E-7,5.5E-8,2.41E-8,1.65E-8,1.05E-8,        &&
D
9.6E-3,3.2E-3,1.6E-3,4.E-4,2.1E-4,1.4E-4,4.4E-5,2.1E-5,1.2E-5,
4.8E-6,3.6E-6,1.4E-6,1.2E-6,6.E-7,2.8E-7,1.38E-7,9.97E-8,6.77E-8,       &&
E
6.2E-2,1.8E-2,8.4E-3,2.E-3,9.2E-4,4.4E-4,2.E-4,1.E-4,6.2E-5,
2.6E-5,1.9E-5,8.4E-6,7.0E-6,4.E-6,2.E-6,1.09E-6,8.22E-7,5.89E-7   && F
PSQLCT
6*0.0                                          && PSPROB
6.2831853071796                                && TWOPI
.0081  30.  24.                                && FMU,DZ,VV
3.1415926535898  0.5  3.0  0.5772156649        && PI Q6 WS GAM
1.  1.  .25                                    && DELTIM
20*0.0                                         && EMRCST
 4*0.0                                         && ONSCST
```

```
0.0                              && ECONF
5                                && NTHRS
3000.  75.  50.  1000.  7.       && EMTRSH
.001                             && EPS
2.E+9                            && DLIM
100                              && MTAL
30  4                            && NREMLV  NE
100000.,80000.,70000.,40000.,30000.,25000.,20000.,10000.,8000.,6000.,
    4000.,3000.,2000.,1000.,800.,700.,600.,500.,400.,300.
    100.,75.,50.,30.,15.,5.,1.,.1,.01,.01                    && REMLEV
(MARROW)
7*1.,6.E-1,1.E-1,6.E-2,3.E-2,19*0.,1.,8.5E-1,8.E-1,7.E-1,5.E-1,2.E-1,8.E-
2,23*0.
1.,8.E-1,5.E-1,27*0.,14*1.0,9.96E-1,9.E-1,4.E-1,5.E-2,12*0.  && PROBEF

81  80   && IPOINT  NCS

&& (LABDOS(I,J),J=1,2),I=1,25
FRACTION OF TRAVEL
IN RURAL POPULATION ZONE
FRACTION OF TRAVEL
IN SUBURBAN POPULATION ZONE
FRACTION OF TRAVEL
IN URBAN POPULATION ZONE
VELOCITY IN RURAL POPULATION
ZONE (KILOMETERS/HOUR)
VELOCITY IN SUBURBAN POP. ZONE
(KILOMETERS/HOUR)
VELOCITY IN URBAN POPULATION
ZONE (KILOMETERS/HOUR)
NUMBER OF CREWMEN

DISTANCE FROM SOURCE TO CREW
(METERS)
NUMBER OF HANDLINGS

STOP TIME PER KM (HR/KM)

MINIMUM STOP TIME PER TRIP
(HR)
ZERO STOP TIME PER TRIP (HR)

MINIMUM NUMBER OF RAIL CLASSIF
ICATIONS/INSPECTIONS
PERSONS EXPOSED WHILE STOPPED

AVERAGE EXPOSURE DISTANCE
WHILE STOPPED (METERS)
STORAGE TIME PER SHIPMENT
(HR)
NUMBER OF EXPOSED PERSONS
```

```
DURING STORAGE
AVERAGE EXPOSURE DISTANCE
WHILE IN STORAGE (METERS)
NUMBER OF PEOPLE PER VEHICLE
ON LINK
FRACTION OF URBAN TRAVEL
DURING RUSH HOUR TRAFFIC
FRACTION OF URBAN TRAVEL
ON CITY STREETS
FRACTION OF RURAL-SUBURBAN
TRAVEL ON FREEWAYS
*TRAFFIC COUNT PASSING A
SPECIFIC POINT-RURAL ZONE
*TRAFFIC COUNT PASSING A
SPECIFIC POINT-SUBURBAN ZONE
*TRAFFIC COUNT PASSING A
SPECIFIC POINT-URBAN ZONE

&& IKW1
DIMEN     PARM      POPDEN    PACKAGE   SHIPMENT  NORMAL    ACCIDENT
ECONOMIC
MATERIAL  RELEASE   OTHER     DEFINE    ISOTOPES  TITLE     FORM
TRANSFER
EOI       EOF
&& IKW2
LABGRP    LABPKG    PKGCDM    PKGSZ1    PKGSZ2    EOF
&& IKW3
LABISO    LABMAT    FKMPS     EOF
&& IKW4
FTZNR     FTZNS     FTZNU     VELR      VELS      VELU      CREWNO
ADSTCW
HANDNO    STOPTIM   MINST     TIMZR     FMINCL    PDST      RST
DTSTOR
PDSTOR    RSTOR     PPV       FRSHR     FCTST     FTLFWY    TCNTPR
TCNTPS
TCNTPU    NMODE     RPD       RR        RS        RU        FNOATT    EOF
&& IKW5
ARATMZ    SEVFRC    EOF
&& IKW6
RPCVAL    INGVAL    EOF
&& IKW7
RFRAC     AERSOL    RESP      AREADA    DFLEV     PSPROB    EOF
&& IKW8
RADIST    BDF       XFARM     CULVL     BRATE     ITRAIN    EOF

&& LDESCR
DISTANCE TRAVELED
K ZERO
DOSE RATE (TRANSPORT INDEX)
PACKAGES PER SHIPMENT
NUMBER OF SHIPMENTS
```

```
FRACTION OF TRAVEL - RURAL
POPULATION DENSITY - RURAL
VELOCITY - RURAL
RATIO OF PEDESTRIAN DENSITY (RPD)
FRACTION OF TRAVEL - SUBURBAN
POPULATION DENSITY - SUBURBAN
VELOCITY - SUBURBAN
SUBURBAN SHIELDING FACTOR (RS)
FRACTION OF TRAVEL - URBAN
POPULATION DENSITY - URBAN
VELOCITY - URBAN
FRACTION OF TRAVEL ON CITY STREETS
URBAN SHIELDING FACTOR (RU)
PERSONS EXPOSED WHILE STOPPED
STOP TIME
EXPOSURE DISTANCE WHILE STOPPED
STORAGE TIME PER SHIPMENT
NUMBER OF PERSONS EXPOSED DURING STORAGE
STORAGE EXPOSURE DISTANCE
DISTANCE FROM SOURCE TO CREW
NUMBER OF CREW MEMBERS
NUMBER OF HANDLINGS
EXPOSURE TIME FOR HANDLERS
PERSONS EXPOSED PER HANDLING
HANDLER EXPOSURE DISTANCE
NUMBER OF PEOPLE PER VEHICLE
FRACTION OF TRAVEL ON FREEWAYS
TRAFFIC COUNT - RURAL
FRACTION OF RUSH HOUR TRAVEL
TRAFFIC COUNT - SUBURBAN
TRAFFIC COUNT - URBAN
NUMBER OF FLIGHT ATTENDANTS
RURAL SHIELDING FACTOR (RR)

&& LABORG
1YR/LUNG  1YR/MARROW50/GONADS 50/LLI    50/THYROID50/BONE   50/LUNG
50/MARROW
&& LABPOP
RURAL     SUBURBAN  URBAN
&& LABMOD
     TRUCK      RAIL     BARGE      SHIP CARGO-AIR PASS.-AIR PASS.-VAN
    CVAN-T    CVAN-R   CVAN-CA
 && LBTRSH
  1YR/LUNG1YR/MARROW 50/GONADS    50/LLI50/THYROID
&& LABEF
    LUNG-1    LUNG-2    LUNG-3    MARROW

0123456789 && NVY
 ,=()*     && WORDS
```

4.0 VARIABLE AND SUBROUTINE GLOSSARIES

4.1 Variable Glossary

Table 4-1 lists the variables in common in the RADTRAN 4 code. The first column in Table 4-1 contains the variable names in alphabetical order. The second column contains the data type, such as an integer, real, logical, or character. The third column contains the unit of measure. Unitless is given if the variable has no unit of measure. The fourth column contains the definition and description of the variable. Table 4-2 groups the variables in common blocks. Variables included in Tables 4-1 and 4-2 are important to theoretical applications. Variables not included serve as holders of intermediate mathematical values during calculations. The programmer can obtain information on the line numbers of each variable by compiling RADTRAN 4 with the CROSS_REFERENCE FORTRAN command qualifier. For further information on FORTRAN command qualifiers, see VAX FORTRAN User Manual (DEC, 1988).

Table 4-1

Common Block Variables in RADTRAN 4

Variable	Type	Units/ (Unitless)	Description
A	Real	(unitless)	Coefficient values for calls to subroutine FIND1.
A1T20	Real	m^2	Area involving decontamination factors between 1 and 20.
A20T40	Real	m^2	Area involving decontamination factors between 20 and 40.
ACCGE	Real	genetic effects	Genetic effects per accident for each accident-severity category and population-density zone.
ACCLCF	Real	LCFs	Latent cancer fatalities (LCFs) per accident by accident-severity category and population-density zone.
ACCPYR	Real	accidents	Expected number of accidents by accident-severity category and population-density zone.
ACCRAT	Real	accidents/ km	Accident rates.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
ADSTCW	Real	m	Average distance from the radiation source to the crew.
AERSOL	Real	(unitless)	Fraction of isotope that is released from a package in aerosol form.
AGT40	Real	m^2	Area involving decontamination factors greater than 40.
AM241	Real	*	Array of properties of americium-241.
AM243	Real	*	Array of properties of americium-243.
AP1T1	Real	m^2	Area with decontamination factors between 0.1 and 1.
ARATMZ	Real	accidents/ km	Overall accident rate for each mode in each population-density zone.
AREADA	Real	m^2	Dispersal accident isopleth areas.
ASIZE	Real	m^2	Size of each AREADA.
ATERM	Real	mrem/hr	Dose rate multiplied by the number of packages per shipment (TI • PPS).
BDF	Real	(unitless)	Building dose factor (BDF), which is a measure of air filtration by heating/cooling/ventilation systems.

*Common Block P contains data arrays for all isotopes in the internal data library. Each array contains 11 parameters: half-life (days) (ICRP 38, 1983); photon energy (MeV/disintegration) (ICRP 38, 1983); cloud dose factor (rem-m^3/Ci-sec) (DOE, 1988a); 50-yr committed effective dose equivalent for inhalation (rem/Ci) (Dunning, 1983; DOE, 1988b); 50-yr committed effective dose equivalent for ingestion (rem/Ci) (Dunning, 1983; DOE, 1988b); food transfer factor; soil transfer factor; deposition velocity of aerosol particles (m/sec); lung type for early effects calculations (ICRP 38, 1983); 1-yr lung dose for inhalation (rem/Ci) (Dunning, 1983); 1-yr marrow dose for inhalation (rem/Ci) (Dunning, 1983).

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
BRATE	Real	m^3/min	Breathing rate of persons exposed to aerosol releases.
BTERM	Real	mrem/hr	Transport Index (TI) multiplied by package per shipment and total number of shipments (TI • PPS • SPY).
C14GAS	Real	*	Array of properties of carbon-14 dioxide gas.
C14ORG	Real	*	Array of properties of carbon-14 in organic form.
CA45	Real	*	Array of properties of calcium-45.
CAYZER	Real	m^2	Transport Index (TI)-to-dose rate conversion factor.
CDF	Real	rem-m^2/ Ci-sec	Cloudshine dose factor.
CE141	Real	*	Array of properties of cerium-141.
CE144	Real	*	Array of properties of cerium-144 [photon energy includes the short half-life daughter praseodymium-144].
CF252	Real	*	Array of properties of californium-252.
CH1	Real	hr/m	Exposure factor for general freight railcrew (.16).
CH2	Real	hr/m	Exposure factor for dedicated railcrew (.01).
CHIVAL	Real	Ci-sec/ m^3/Ci released	Dilution factors for each annualar zone.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
CI	Real	Ci	Amount of radioactivity per package of an isotope.
CIPKG	Real	Ci	Radioactivity level for each isotope per package.
CLDGE	Real	genetic effects	Genetic effects from cloudshine for each accident-severity category and population-density zone.
CLDLCF	Real	LCFs	Latent cancer fatalities (LCFs) from cloudshine for each accident-severity category.
CLDOSF	Real	rem-m^2/ Ci-sec	Cloudshine dose factor for each isotope.
CM242	Real	*	Array of properties of curium-242.
CM244	Real	*	Array of properties of curium-244.
CO58	Real	*	Array of properties of cobalt-58.
CO60	Real	*	Array of properties of cobalt-60.
COEF	Real	(unitless)	Coefficients for gamma and neutron calculations.
CONTUF	Logical	(unitless)	Byte flag.
CR51	Real	*	Array of properties of chromium-51.
CREWNO	Real	persons	Number of crew members per shipment.
CS134	Real	*	Array of properties of cesium-134.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
CS137	Real	*	Array of properties of cesium-137 [photon energy includes the short half-life daughter barium-137m].
CULVL	Real	$\mu Ci/m^2$	Clean-up level.
D4	Real	person-rem	Dose to handlers.
DECON	Real	(unitless)	Ratio of deposited curies to clean-up criterion (CULVL) for each accident-severity category, isodose area, and isotope.
DELTIM	Real	hr	Exposure times (before cleanup) of dispersed material deposited on the ground.
DEPVEL	Real	m/sec	Deposition velocity for aerosol particles.
DFLEV	Real	Ci-sec/ m^3-Ci released	Time-integrated concentrations of radionuclides in an annular area normalized to the initial released inventory.
DIHD	Real	m	Average exposure distance from the radiation source to handlers.
DIST	Logical	(unitless)	Set to TRUE if the keyword DISTKM is used; otherwise, set to FALSE.
DISTKM	Real	km	Distance traveled per trip for each model.
DLIM	Real	m^2	Maximum allowable size of an isopleth; used in the MDT subroutine (2E+09).
DNHGE	Real	genetic effects	Genetic effects from inhalation exposure by accident-severity category and population-density zone.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
DNHLCF	Real	LCFs	Latent cancer fatalities (LCFs) from inhalation exposure by accident-severity category.
DNORML	Real	(various units)	Array holding 25 variables pertaining to the incident-free dose calculation. [See the "RADTRAN 4, Volume III: User Guide" (Neuhauser and Kanipe, in preparation).]
DOSDST	Real	rem	Dose-vs-distance relationship for nondispersal accidents.
DOSE2	Real	person-rem	Crew dose by population-density zone for sensitivity analysis.
DOSE5	Real	person-rem	Off-link dose by employed variables for the importance analysis summary.
DOSE6	Real	person-rem	On-link dose by employed variable for the importance analysis summary.
DOSEN	Real	person-rem	Dose to 10 exposed groups for the incident-free dose calculation.
DOSMAX	Real	rem	Maximum incident-free individual dose.
DPKG	Real	m	Package dimension.
DSTRVL	Real	km	Distance traveled per shipment.
DTHD	Real	hr	Average exposure time for handlers.
DTOT	Real	person-rem or LCFs	For the UNIT option, the 50-yr population dose commitment; for the NONUNIT option, latent cancer fatalities (LCFs).

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
DTST	Real	hr/km	Stop time per unit distance traveled.
DTSTOR	Real	hr	Storage time per shipment.
DUMMY	Real	(unitless)	Storage for isotopes defined with the DEFINE keyword.
DZ	Real	m	Source-to-receptor distance (x) used in the maximum individual dose calculation; equal to 30 m.
EARLYD	Real	rem	Early effects calculation of the inhalation dose to five specific organs by accident-severity category.
EARLYL	Real	rem	Early-category fatalities calculation of the inhalation dose by lung type for each accident-severity category.
ECNMPT	Real	U.S. dollars	Economic impact for each accident-severity category and population-density zone.
ECNPYR	Real	U.S. dollars	Expected economic impact for each accident-severity category and population-density zone.
EFFSIZ	Real	m	Effective package size to compute the package coefficient if the actual package size is greater than or equal to 4 m.
EFPYR	Real	(unitless)	Total expected early fatalities for each accident-severity category and population-density zone.
EMPYR	Real	(unitless)	Total expected early morbidities for each accident-severity category and population-density zone.
EMRCST	Real	U.S. dollars	Emergency clean-up cost.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
EMTRSH	Real	rem	Thresholds for early morbidities for five organ types.
EPS	Real	(unitless)	Tolerance level used in the MDT subroutine (.001).
EU152	Real	*	Array of properties of europium-152.
EU154	Real	*	Array of properties of europium-154.
EU155	Real	*	Array of properties of europium-155.
EVAL1	Real	U.S. dollars	Economic option 1 [See the "RADTRAN 4, Volume II: Technical Manual (Neuhauser and Kanipe, in preparation)].
EVAL2	Real	U.S. dollars	Economic option 2 [See the "RADTRAN 4, Volume II: Technical Manual (Neuhauser and Kanipe, in preparation)].
EVAL3	Real	U.S. dollars	Economic option 3 [See the "RADTRAN 4, Volume II: Technical Manual (Neuhauser and Kanipe, in preparation)].
EXPEO	Real	persons	Expected number of persons in each annular area.
EXPLCF	Real	LCFs	Expected latent cancer fatalities (LCFs) by the impact to five specific organs.
FCTST	Real	(unitless)	Fraction of urban travel on city streets.
FDTRAN	Real	(unitless)	Food transfer factor for ingestion pathway for each isotope.
FE55	Real	*	Array of properties of iron-55.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
FE59	Real	*	Array of properties of iron-59.
FG	Real	(unitless)	Fraction of gamma radiation from material.
FKMPS	Real	km	Array of distances traveled per shipment for each isotope.
FMINCL	Real	(unitless)	Minimum number of rail inspections and classifications (rail mode only).
FMU	Real	m^{-1}	Linear attenuation coefficient of 1-MeV gamma radiation in the air; used in maximum individual dose.
FN	Real	(unitless)	Fraction of neutron radiation from material.
FNOATT	Real	persons	Number of flight attendants on commercial passenger flights.
FRCZON	Real	(unitless)	Fraction of travel in each population-density zone.
FRGAMA	Real	(unitless)	Fraction of dose rate attributed to gamma radiation for each material.
FRNEUT	Real	(unitless)	Fraction of dose rate attributed to neutron radiation for each material.
FRSHR	Real	(unitless)	Fraction of urban travel during rush-hour traffic.
FTLFWY	Real	(unitless)	Fraction of rural and suburban travel on freeways.
FTZNR	Real	(unitless)	Fraction of travel in rural zones.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
FTZNS	Real	(unitless)	Fraction of travel in suburban zones.
FTZNU	Real	(unitless)	Fraction of travel in urban zones.
GA67	Real	*	Array of properties of gallium-67.
GAM	Real	(unitless)	Euler's constant that is equal to 0.5772156649; used in the Bessel function calculation in the BESSL subroutine.
GEPYR	Real	genetic effects	Expected genetic effects for each accident-severity category and population-density zone.
GNDSGE	Real	genetic effects	Genetic effects per million person-rem from groundshine.
GRDGE	Real	genetic effects	Genetic effects from groundshine for each accident-severity category and population-density zone.
GRDLCF	Real	LCFs	Latent cancer fatalities (LCFs) from groundshine for each accident-severity category and population-density zone.
H3GAS	Real	*	Array of properties of tritium gas.
H3WTR	Real	*	Array of properties of tritiated water.
HANDNO	Real	(unitless)	Number of handlings of a package during shipment.
HLIFE	Real	days	Half-life for radioactive decay of an isotope.
I125	Real	*	Array of properties of iodine-125.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
I129	Real	*	Array of properties of iodine-129.
I131	Real	*	Array of properties of iodine-131.
IACC	Integer	(unitless)	Flag for dispersal-type accidents (1 = no; 2 = yes).
IANA	Integer	(unitless)	Incident-free or accident analysis flag.
ICARD	Character	(unitless)	Last input line read.
ICARDI	Character	(unitless)	Holds individual characters from the TEST and ICARD variables for comparison with expected characters.
ICOUNT	Integer	(unitless)	Total number of isotopes in an analysis.
IDISP	Integer	(unitless)	Dispersibility category for each isotope.
IDSP	Integer	(unitless)	Dispersion category of an isotope.
IEXCLU	Integer	(unitless)	Flag to designate exclusive-use or nonexclusive-use for each mode (1 = yes).
IGRP	Integer	(unitless)	Pointer to the current physical-chemical group.
IKW1	Character	(unitless)	First-level keywords.
IKW2	Character	(unitless)	Second-level keywords under PACKAGE.
IKW3	Character	(unitless)	Second-level keywords under SHIPMENT.
IKW4	Character	(unitless)	Second-level keywords under NORMAL.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
IKW5	Character	(unitless)	Second-level keywords under ACCIDENT.
IKW6	Character	(unitless)	Second-level keywords under MATERIAL.
IKW7	Character	(unitless)	Second-level keywords under RELEASE.
IKW8	Character	(unitless)	Second-level keywords under OTHER.
INCFRE	Real	person-rem	Incident-free doses for each material in each link.
INGGE	Real	genetic effects	Genetic effects from ingestion for each accident-severity category in rural zones.
INGLCF	Real	LCFs	Latent cancer fatalities (LCFs) from ingestion for each accident-severity category in rural zones.
INGVAL	Real	rem/Ci	Number of rems per curie ingested for each isotope.
IPCGRP	Integer	(unitless)	Index of physical-chemical group for each isotope.
IPOINT	Integer	(unitless)	Pointer to current input character.
IPRAM	Integer	(unitless)	Index for the PRMNDS array (21).
IPSQSB	Integer	(unitless)	Flag for Pasquill category treatment of dispersion (1 = yes).
IRNKC	Integer	(unitless)	Flag for placing data in file 6 (1 = yes).
ISEN	Integer	(unitless)	Flag designating full printout and importance analysis calculations.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
ISO	Integer	(unitless)	Indexes the number of isotopes as determined by the input order.
ISONAM	Character	(unitless)	Array of isotope names in the internal library plus the names defined with the DEFINE keyword (125 names).
ISOSAV	Integer	(unitless)	Isotope indices for each isotope in each material.
ITRAIN	Integer	(unitless)	Flag to select railcrew exposure factor.
IUOPT	Integer	(unitless)	Flag designating a desired shielding model.
KBYTE	Integer	(unitless)	Maximum length of an input word or number (10 characters).
KI	Integer	(unitless)	Pointers to print the importance analysis summary in rank order (38 pointers).
KLOCK	Character	(unitless)	Current time retrieved from system call.
KR852	Real	*	Array of properties of krypton-85.
LABDOS	Character	(unitless)	Labels for DNORML entries.
LABEF	Character	(unitless)	Labels for early-fatality input tables.
LABGRP	Character	(unitless)	Labels for user-designated physical-chemical groups of isotopes.
LABISO	Character	(unitless)	Labels for isotopes.
LABMAT	Character	(unitless)	Labels for user-selected material names.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
LABMOD	Character	(unitless)	Labels for the 10 transportation modes.
LABORG	Character	(unitless)	Labels for eight organ types.
LABPOP	Character	(unitless)	Labels of population-density zones (rural, suburban, urban).
LARAT	Real	accident/ km	Accident rate for each link.
LBTRSH	Character	(unitless)	Labels for dose thresholds.
LCFPYR	Real	LCFs	Expected latent cancer fatalities (LCFs) for each accident-severity category and population-density zone.
LDESCR	Character	(unitless)	Descriptions of variables for sensitivity output.
LDIST	Real	km	Distance traveled for each link.
LIB	Integer	(unitless)	Index of current isotope in the data library.
LIBSAV	Integer	(unitless)	Array index for internal and DEFINE keyword data for all isotopes in an analysis.
LISTLH	Integer	(unitless)	Represents the index for expected risk values (12).
LMODE	Integer	(unitless)	Represents the mode for each link.
LNGTAB	Integer	(unitless)	Represents the table of acute pulmonary response curves.
LNGTYP	Integer	(unitless)	Represents the current lung type.
LPOPD	Real	people/ km^2	Represents the population density for each link.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
LSPED	Real	km/hr	Speed of shipment for each link.
LTYPE	Integer	(unitless)	Represent the road type for each link (1 = freeway; 2 = nonfreeway; 3 = not applicable).
LVDEN	Real	vehicles/hr	Vehicle density for each link; measured in one-way traffic passing a point per hour.
LZONE	Integer	(unitless)	Population zone for each link (1 = rural; 2 = suburban; and 3 = urban).
MAT	Integer	(unitless)	Index of material as determined by the input order.
MATSAV	Integer	(unitless)	Indices for each material in each mode.
MAXARS	Integer	(unitless)	Maximum number of areas in AREADA calculation (30 areas).
MAXDSP	Integer	(unitless)	Maximum number of dispersion categories (11 dispersion categories).
MAXGRP	Integer	(unitless)	Maximum number of physical-chemical groups (15 groups).
MAXISO	Integer	(unitless)	Maximum number of isotopes (65 isotopes).
MAXMAT	Integer	(unitless)	Maximum number of materials (12 materials).
MAXMOD	Integer	(unitless)	Maximum number of transportation modes (10 modes).
MAXNDL	Integer	(unitless)	Maximum number of entries in the DNORML array (25 entries).
MAXRAD	Integer	(unitless)	Maximum number of radial distances (15 distances).

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
MAXSEV	Integer	(unitless)	Maximum number of accident-severity categories (20 accident-severity categories).
MAXTYP	Integer	(unitless)	Maximum number of values entered in PKGCDM (15 values).
MCOUNT	Integer	(unitless)	Total number of materials in an analysis.
MN54	Real	*	Array of properties of manganese-54.
MO99	Real	*	Array of properties of molybdenum-99 [photon energy includes the daughter technetium-99m].
MODE	Integer	(unitless)	Current mode number.
MODSAV	Integer	(unitless)	Modes used in analysis.
MTAL	Integer	(unitless)	Number of iterations allowed to find the tolerance level in the MDT subroutine (100 iterations).
NA	Integer	(unitless)	Number of Pasquill isopleth areas for a dispersal accident (18 isopleth areas).
NAREAS	Integer	(unitless)	Number of annular isodose areas used in the analysis of dispersal accidents.
NAVINT	Integer	(unitless)	Number of isopleth areas used to find CHIVAL for each isotope.
NB94	Real	*	Array of properties of niobium-94.
NB95	Real	*	Array of properties of niobium-95.
NC	Integer	(unitless)	Number of Pasquill stability categories (six categories)

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
NCS	Integer	(unitless)	Maximum length of an input line (80 characters).
NDLAB	Integer	(unitless)	Number of parameters in incident-free transportation model (25 parameters).
NDOSE	Integer	(unitless)	Number of incident-free dose categories (10 dose categories).
NDOSP1	Integer	(unitless)	Total columns for incident-free dose categories (11 columns).
NDERV	Integer	(unitless)	Number of variables printed in the importance analysis summary (38 variables).
NE	Integer	(unitless)	Number of columns in the PROBEF array (four columns).
NGROUP	Integer	(unitless)	Number of physical-chemical groups.
NIS	Integer	(unitless)	Total number of isotopes for each material.
NISOTP	Integer	(unitless)	Number of defined isotopes (60 plus the number defined with the DEFINE keyword to a maximum of 125 isotopes).
NKW1	Integer	(unitless)	Number of keywords in IKW1.
NKW2	Integer	(unitless)	Number of keywords in IKW2.
NKW3	Integer	(unitless)	Number of keywords in IKW3.
NKW4	Integer	(unitless)	Number of keywords in IKW4.
NKW5	Integer	(unitless)	Number of keywords in IKW5.
NKW6	Integer	(unitless)	Number of keywords in IKW6.
NKW7	Integer	(unitless)	Number of keywords in IKW7.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
NKW8	Integer	(unitless)	Number of keywords in IKW8.
NLEG	Integer	(unitless)	Number of modes or number of links.
NLNK	Integer	(unitless)	Number of links.
NMAT	Integer	(unitless)	Total number of materials in each mode.
NMODE	Integer	(unitless)	Number of modes used in the analysis.
NORG	Integer	(unitless)	Total number of organ types (eight organic types).
NP237	Real	*	Array of properties of neptunium-237.
NPKG	Integer	(unitless)	Total number of packages.
NPOP	Integer	(unitless)	Number of population-density zones (three zones).
NPOP1	Integer	(unitless)	Number of population-density zones plus one pedestrian zone (four zones).
NPRAM	Integer	(unitless)	Number of variables used in the importance analysis summary (38 variables).
NPRP	Integer	(unitless)	Number of properties for library isotopes (11 properties).
NRAD	Integer	(unitless)	Number of radial distances for nondispersal accidents.
NREMLV	Integer	(unitless)	Number of dose levels used in the look-up table for early fatalities (30 dose levels).
NRFLAG	Logical	(unitless)	Set to true when a delimiter is encountered.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
NSEV	Integer	(unitless)	Number of accident-severity categories.
NTHRS	Integer	(unitless)	Number of early morbidity thresholds (five thresholds).
NVY	Character	(unitless)	Array of characters representing the numbers 0 through 9 for comparison with ICARDI.
ONSCST	Real	U.S. dollars	On-scene clean-up costs for each accident-severity category and dispersibility category.
P32	Real	*	Array of properties of phosphorus-32.
PCKG	Logical	(unitless)	Set to TRUE if the PKGSIZ keyword is used; otherwise, set to FALSE.
PDST	Real	persons	Number of persons exposed during storage.
PDSTOR	Real	persons	Number of persons exposed during storage.
PDTOT	Real	person-rem or LCFs	For the UNIT option, total 50-yr population dose commitment; for the NONUNIT option, latent cancer fatalities (LCFs) by accident-severity category and population-density zone.
PECMPT	Real	U.S. dollars	Economic impact over all accident-severity categories for each population-density zone.
PHTE	Real	MeV/disintegration	Photon energy of an isotope.
PHTENG	Real	MeV/disintegration	Photon energy values for each isotope.
PKGCDM	Real	m	Characteristic dimension of a given type of package.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
PKGCOE	Real	m^2	Conversion factor for point-source dose rate from dose rate at 1 m for each material.
PKGSHP	Real	(unitless)	Number of packages per shipment for each mode and material.
PKGSIZ	Real	m	Characteristic package dimension for each material.
PKGSZ1	Real	m	Threshold dimension for packages handled by forklifts.
PKGSZ2	Real	m	Threshold dimension for packages handled by cranes.
PM147	Real	*	Array of properties of promethium-147.
POPDEN	Real	person/ km^2	Population densities of zones.
POPDR	Real	person/ km^2	Rural population density.
POPDS	Real	person/ km^2	Suburban population density.
POPDU	Real	person/ km^2	Urban population density.
POPDUP	Real	POPDU • 0.1 • RPD	Pedestrian population density.
PPH	Real	persons	Number of exposed persons per handling.
PPS	Real	(unitless)	Number of packages of a material per shipment.
PPV	Real	persons	Number of persons per vehicle sharing the transport link.
PRMNDS	Real	(unitless)	Partial derivative of normal dose concerning DNORML parameters.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
PROBEF	Real	(unitless)	Array of the probability of early fatality from an exposure to marrow or lung.
PSAREA	Real	m	Areas used in Pasquill dispersion calculation.
PSPROB	Real	(unitless)	Probability of occurrence of each of six Pasquill stability categories.
PSQLCT	Real	Ci-sec/ m^3-Ci released	Integrated air-concentration values for Pasquill stability categories.
PSUMEF	Real	(unitless)	Sum of early fatalities for each accident-severity category and population-density zone.
PSUMEM	Real	(unitless)	Sum of early morbidities for each accident-severity category and population-density zone.
PU236	Real	*	Array of properties of plutonium-236.
PU238	Real	*	Array of properties of plutonium-238.
PU239	Real	*	Array of properties of plutonium-239.
PU240	Real	*	Array of properties of plutonium-240.
PU241	Real	*	Array of properties of plutonium-241.
PU242	Real	*	Array of properties of plutonium-242.
Q6	Real	0.5 rem-m^2/hr-Ci-MeV	Constant used in dose-rate formula.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
RADIST	Real	m	Radial distances defining exposure annuli for the nondispersal accident model.
RDF	Real	(unitless)	Resuspension dose factor.
RDFGE	Real	(unitless)	Genetic effects from resuspension exposure.
RDFLCF	Real	(unitless)	Latent cancer fatalities from resuspension exposure.
REMLEV	Real	rem	Dose levels for the look-up table for early fatalities.
RESP	Real	(unitless)	Respirable fractions of an isotope in aerosol form.
RFRAC	Real	(unitless)	Package release fraction for each physical-chemical group and accident-severity category.
RPCVAL	Real	rem/Ci	Conversion factors for each isotope for eight organ types.
RPD	Real	(unitless)	Ratio of pedestrian density to overall population density.
RR	Real	(unitless)	Shielding effectiveness factor for rural areas.
RS	Real	(unitless)	Shielding effectiveness factor for suburban areas.
RSPEC	Logical	(unitless)	Set true if LINKS is used; otherwise false.
RSPHLF	Real	days	Resuspension half-time [time for half of the material to re-deposit (365 days)].
RST	Real	m	Exposure distance during shipment stops.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
TBPPH	Real	persons	Number of persons exposed per handling of intermediate and large packages.
TC99	Real	*	Array of properties of technetium-99.
TCLEVL	Real	Ci/m^2	Total contamination level over all isotopes for each accident-severity category.
TCNTPR	Real	vehicles/ hr	One-way traffic count in rural zones.
TCNTPS	Real	vehicles/ hr	One-way traffic count in suburban zones.
TCNTPU	Real	vehicles/ hr	One-way traffic count in urban zones.
TDCONF	Real	(unitless)	Total decontamination factor of all isotopes for each accident-severity category.
TDSDST	Real	rem	Total dose vs distance for each value of RADIST.
TE125M	Real	*	Array of properties of tellurium-125M.
TE127	Real	*	Array of properties of tellurium-127.
TE127M	Real	*	Array of properties of tellurium-127M.
TE129	Real	*	Array of properties of tellurium-129.
TE129M	Real	*	Array of properties of tellurium-129M.
TEST	Character	(unitless)	Holds the last input word read.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
TI	Real	mrem/hr	Dose rate; equals dose in mrem/hr at 1 m (Transport Index).
TIMZR	Real	hr	Zero stop time per trip.
TIPKG	Real	mrem/hr	Dose rate at 1 m for each package of a given material by each mode (mrem/hr) at 1 m (Transport Index).
TITLE	Character	(unitless)	Title read in from the input file.
TOL	Real	(unitless)	Tolerance level for comparing equivalence of real numbers (1E-5).
TS1	Real	day	Lesser of 50 yr or the time required to reduce the contamination level to the criterion level.
TS2	Real	day	Lesser of 10 days or the time required to reduce the contamination level to the criterion level.
TS3	Real	day	Lesser of adjusted clean-up the time or the time required to reduce the contamination level.
TS4	Real	day	Time required to raze and rebuild a large area.
TSUMRS	Real	(see VSUMRS)	Intermediate sums of the accident output.
TWOPI	Real	(unitless)	2 times PI (6.2831853071796).
U	Real	(unitless)	Holds μ values for calls to subroutine FINDI.
U233	Real	*	Array of properties of uranium-233.
U235	Real	*	Array of properties of uranium-235.

Table 4-1

Common Block Variables in RADTRAN 4 (Continued)

Variable	Type	Units/ (Unitless)	Description
U238	Real	*	Array of properties of uranium-238.
URSKF	Logical	(unitless)	Set to TRUE for effective dose output; set to FALSE for individual organ doses.
VALINK	Real	fatalities, morbidities, and U.S. dollars	Shipment-specific values for early fatalities, early morbidities, and economic costs.
VELDEP	Real	m/sec	Deposition velocity of particles in aerosol form for each isotope.
VELM	Real	m/sec	Reciprocal of the average velocity for a mode.
VELR		sec/m	Rural velocity.
VELS	Real	sec/m	Suburban velocity.
VELU	Real	sec/m	Urban velocity.
VSUMRS	Real	person-rem or LCFs and genetic effects	For the non-LINK option, the array of accident output for each isotope; for the LINK option, the array of accident output for each link.
VV	Real	km/hr	Velocity used in maximum individual dose calculation (24 km/hr).
WBLCF	Real	LCF/person-rem	Latent cancer fatalities (LCFs) per million person-rem effective dose.
WORDS	Character	(unitless)	Characters used as delimiters for the RADTRAN input.
WS	Real	3 m	Sidewalk width used in early effects calculation; equal to 3 m.
XE133	Real	*	Array of properties of xenon-133.

Table 4-1

Common Block Variables in RADTRAN 4 (Concluded)

Variable	Type	Units/ (Unitless)	Description
XFARM	Real	(unitless)	Fraction of rural land under cultivation.
Y91	Real	*	Array of properties of yttrium-91.
ZN65	Real	*	Array of properties of zinc-65.
ZR95	Real	*	Array of properties of zirconium-95.

Table 4-2

Variables in Common Blocks

COMMON BLOCK - ACD
Values used to compute accident dose and economic impact

ACCGE	GNDSGE
ACCLCF	IACC
ACCPYR	IDISP
ACCRAT	IPCGRP
AERSOL	LCFPYR
ARATMZ	LNGTAB
AREADA	NAREAS
ASIZE	NTHRS
BDF	ONSCST
CIPKG	PHTENG
CULVL	RADIST
DELTIM	RDFGE
DFLEV	RDFLCF
DNHGE	RESP
DNHLCF	RFRAC
DOSDST	RPCVAL
ECNMPT	SEVFRC
EFPYR	SUMEF
EMPYR	SUMEM
EMRCST	TABHLF
EXPEO	TABRDF
FRCZON	WBLCF
GEPYR	XFARM

COMMON BLOCK - ACD2
Additional values used to compute accident dose and economic impact

AGT40	INGVAL
AP1T1	IPSQSB
A1T20	IRNKC
A20T40	NAVINT
BRATE	NPOP1
CHIVAL	PDTOT
CLDGE	PECMPT
CLDLCF	PSAREA
CLDOSF	PSPROB
DECON	PSQLCT
DTOT	PSUMEF
EARLYD	PSUMEM
EARLYL	SLTRAN
ECNPYR	TBAR
EMTRSH	TCLEVL
INGGE	

Table 4-2

Variables in Common Blocks (Continued)

COMMON BLOCK - ACD2 (Concluded)
Additional values used to compute
accident dose and economic impact

EVAL1	TDCONF
EVAL2	TDSDST
EVAL3	TS1
EXPLCF	TS2
FDTRAN	TS3
GRDGE	TS4
GRDLCF	URSKF
INGLCF	VELDEP

COMMON BLOCK - CARD
Variable used when reading the input deck

CONTUF	NCS
IPOINT	NRFLAG
KBYTE	

COMMON BLOCK - CCARD
Character variables used when reading the input deck

ICARD	NVY
ICARDI	TEST

COMMON BLOCK - CMDT
Variables used in subroutine MDT,
which calculates concentration of
material in cloud after depletion

DLIM	MTAL
EPS	

COMMON BLOCK - CON
Mathematical constants

DZ	PI
FMU	Q6
GAM	TOL
NA	TWOPI
NDREV	VV
NC	WS

Table 4-2

Variables in Common Blocks (Continued)

COMMON BLOCK - HEADER
Character arrays printed at the top of each page of output

CDATE	TITLE
KLOCK	

COMMON BLOCK - IKW
Character strings containing input keywords

IKW1	IKW5
IKW2	IKW6
IKW3	IKW7
IKW4	IKW8

COMMON BLOCK - LABELS
Arrays used to store character strings that are read in or used to label output tables

LABDOS	LABORG
LABEF	LABPOP
LABGRP	LDESCR
LABISO	LBTRSH
LABMAT	WORDS
LABMOD	

COMMON BLOCK - LVAR
Link variables, used only when LINKS option is used

LARAT	LTYPE
LDIST	LVDEN
LMODE	LZONE
LPOPD	NLNK
LSPED	RSPEC

COMMON BLOCK - MAIN
Main common block containing many of the most commonly used parameters

DIST	NGROUP
DISTKM	NIS
FKMPS	NMAT
IANA	NMODE

Table 4-2

Variables in Common Blocks (Continued)

COMMON BLOCK - MAIN (Concluded)
Main common block containing many of the most commonly used parameters

ICOUNT	NORG
IEXCLU	NPKG
IPRAM	NPOP
ISEN	NRAD
ISO	NSEV
IUOPT	PCKG
LIB	PKGSHP
LIBSAV	POPDEN
LISTLH	RPD
MAT	RR
MATSAV	RS
MCOUNT	RU
MODE	TABSPY

COMMON BLOCK - MAXSZ
Maximum permitted array sizes

MAXARS	MAXNLD
MAXDSP	MAXRAD
MAXGRP	MAXSEV
MAXISO	NPRAM
MAXMAT	RSPHLF
MAXMOD	

COMMON BLOCK - NIKW
Integer variables containing the lengths of the IKW arrays

NKW1	NKW5
NKW2	NKW6
NKW3	NKW7
NKW4	NKW8

COMMON BLOCK - NORML
Values used to compute incident-free doses

CH1	KI
CH2	MODSAV
COEF	NDLAB
DNORML	NDOSE

Table 4-2

Variables in Common Blocks (Continued)

COMMON BLOCK - NORML (Concluded)
Values used to compute incident-free doses

DOSEN	NDOSP1
DOSMAX	NLEG
DPKG	PKGCDM
EFFSIZ	PKGCOE
FNOATT	PKGSZ2
FRGAMA	PKGSIZ
FRNEUT	PKGSZ1
INCFRE	SENPRM
ITRAIN	TIPKG

COMMON BLOCK - NSEN
Variables used in incident-free dose calculations and sensitivity analysis

ATERM	RVELR
BTERM	RVELS
D4	RVELU
PRMNDS	STERM

COMMON BLOCK - O
Character array of isotope names

ISONAM

COMMON BLOCK - P
Arrays of isotope properties

AM241	NISOTP
AM243	NP237
C14GAS	NPRP
C14ORG	P32
CA45	PM147
CE141	PU236
CF252	PU238
CM242	PU239
CM244	PU240
CO58	PU241
CO60	PU242
CR51	RU103
CS134	RU106
CS137	S35

Table 4-2

Variables in Common Blocks (Continued)

COMMON BLOCK - P (concluded)
Arrays of isotope properties

DUMMY	SB125
EU152	SM151
EU154	SR89
EU155	SR90
FE55	TC99
FE59	TE125M
GA67	TE127
H3GAS	TE127M
H3WTR	TE129
I125	TE129M
I129	U233
I131	U235
KR85	U238
MN54	XE133
MO99	Y91
NB94	ZN65
NB95	ZR95

COMMON BLOCK - PROB
Values used to determine the
probabilities of early fatality

NE	PROBEF
NREMLV	REMLEV

COMMON BLOCK - SENDOS
Dose values used for sensitivity analysis

DOSE2	DOSE6
DOSE5	

COMMON BLOCK - SET
Variables with values that are
set in subroutine SETVAL

A	PDST
ADSTCW	PDSTOR
CAYZER	PHTE
CDF	POPDR
CI	POPDS
CREWNO	

Table 4-2

Variables in Common Blocks (Concluded)

COMMON BLOCK - SET (concluded)
Variables with values that are
set in subroutine SETVAL

DEPVEL	POPDU
DSTRVL	POPDUP
DIHD	PPH
DTHD	PPS
DTST	PPV
DTSTOR	RDF
FCTST	RST
FG	RSTOR
FMINCL	SPY
FN	STPMIN
FRSHR	TCNTPR
FTLFWY	TCNTPS
FTNZR	TCNTPU
FTNZS	TI
FTNZU	TIMZR
HANDNO	U
HLIFE	VELM
IDSP	VELR
IGRP	VELS
LNGTYP	VELU

COMMON BLOCK - TALLY
Summations used in analysis

TSUMRS	VSUMRS
VALINK	

COMMON BLOCK - TBD
Values used to calculate handler dose

TBDIHD	TBPPH
TBDTHD	

4.2 Subroutine Glossary

Table 4-3 lists all subroutines in the code. The left column in Table 4-3 contains the subroutines in alphabetical order; the right column contains corresponding definitions. Table 3-2 gives the number of lines and the order of appearance of each subroutine in the computer code printout.

Table 4-3

Subroutines and Functions in RADTRAN 4

Function or Subroutine	Definition
ACC	Calculates nondispersal (IACC=1) or dispersal (IACC=2) accidents.
ARRAY	Reads N values into the a floating-point array.
BANNER	Prints banner page heading.
BESSL	Computes a modified Bessel function of order zero.
CENTER	Centers characters in a string.
CONVRT	Converts the TEST variable into a real number if appropriate; otherwise, sets the results to 4HWORD.
COST	Computes the economic impact by population-density zone and accident-severity category; checks for the route-controlled shipment.
CPBYTE	Converts the TEST variable into individual characters and places the result into the JVY array.
EARLY	Computes the expected number of early fatalities and early morbidities.
ECON	Calculates the economic impacts from dispersal-accident cleanup.
EFFIT	Determines early fatality probabilities using the lookup table.
FINDI	Performs integration using Bessel function calls.
GOTOER	Prints GOTO error message and stops the program.

Table 4-3

Subroutines and Functions in RADTRAN 4 (Continued)

Function or Subroutine	Definition
IARRAY	Reads N values into the IA integer array.
INITLZ	Initializes variable arrays by setting all values to zero.
MDT	Calculates the concentration of material in radioactive dispersion cloud after the cloud has been depleted by deposition.
NEWPAG	Prints a form feed and prints the date, time, page number, and title at the top of each printout page.
NEXT	Reads input lines and in the TEST variable, returns the position of the next field; prints the input echo of each line.
NORLNK	Called by the NORML subroutine to compute off- and on-link incident-free doses by link for truck, van, and rail modes. Package exposure rates are checked against regulatory limits. An exclusive-use check is also made for the shipment.
NORMAL	Calculates incident-free doses to all population subgroups from all transportation modes. Package exposure rates are checked against regulatory limits. An exclusive-use check is also made for the shipment.
PAIRS	Reads data from FILE6 and produces files of ordered pairs of numbers which can be used for plotting.
PASQ	Called if Pasquill stability categories are used; loads the areas and concentrations into the AREADA and DFLEV arrays.
PRCHRD	Prints input tables and variables.
PRCONS	Prints consequence tables.
PRISTP	Prints the table for isotope-related data.
PRNORD	Prints the DNORML array for incident-free input data.
PROUTA	Prints the incident-free summary table.

Table 4-3

Subroutines and Functions in RADTRAN 4 (Continued)

Function or Subroutine	Definition
PROUTB	For the UNIT option, prints the expected value of population risk in person-rem; for NONUNIT, prints the expected numbers of latent cancer fatalities (LCFs) and genetic effects.
PROUTC	For the NONLINK option, prints tables of size NSEV-by-NPOP; for the LINK, prints tables of size NSEV-by-NLNK.
PROUTD	Prints tables of NNN-by-NSEV, where the NNN dimension is determined by the array being printed.
PRRELD	Prints values of variables for accident-related input data.
PRSEGD	Called only for the LINK option; prints values of variables for each link and each mode.
PRSEND	Prints the importance analysis summary.
QUAD	Performs the integration for the on-link dose for travel in the same direction.
RADTN4	Calls routines to print the input data for extended output; calls routines to set and initialize variables; incident-free and accident calculations are summed; calls print routines for incident-free and accident tables.
RDDAT	Reads default data from external files.
RDINFR	Reads data through the use of keyword identifiers.
SCAN	Scans the input line, determines the location of the next field item, and places it into the TEST variable.
SCRIPT	Called by the BANNER subroutine to print large letters.
SENSTV	Calculates derivatives of incident-free transport parameters for sensitivity analysis by population subgroup and mode.
SENSUM	summation of derivatives calculated in SENSTV.
SETISO	Sets isotope-specific variables.
SETMAT	Sets material-specific variables.

Table 4-3

Subroutines and Functions in RADTRAN 4 (Concluded)

Function or Subroutine	Definition
SETMOD	Sets mode-specific variables.
SETVAL	Sets the distance traveled, package size, and size-dependent package coefficients.
SHIELD	Sets rural, suburban, and urban shielding, depending on the user-selected IUOPT shielding option.
SHIFTC	Right-justifies character string.
SHIFTL	Left-justifies character string.
TRANS	Returns exp(Ur) • B(r).
VALUE	Gets the next field in the line; if it is not numeric, the execution is stopped.
WTRNK	Writes probability and consequence data to file 6 for importance analysis summary or plotting.

5.0 SAMPLES OF DATA INPUT FILE, BENCHMARKING, AND TEXT PROBLEMS

5.1 Input Data

RADTRAN programmers should first become familiar with creating input files, as described in the "RADTRAN 4, Volume 3: User Guide" (Neuhauser and Kanipe, 1992). RADTRAN 4 input files are in free format using keyword identifiers; therefore, the order of data input is irrelevant in many cases. There are cases in which certain data must be given before other data. For instance, the DIMEN keyword and associated data must be given before LABISCO, ACCIDENT, RELEASE, or RADIST since DIMEN data supply information on the number of values to be read. Isotopes not listed in the RADTRAN library must be defined (keyword DEFINE) before they are listed under ISOTOPES since, for each isotope listed, the associated library data is accessed. Not as obvious is the requirement that keyword FORM and its data be given before ISOTOPES. If it is not a UNIT run (identifier with keyword FORM) then the library data is not read for the organ doses and the data that has been entered under keyword MATERIAL is used. These input strategies are also discussed in the RADTRAN 4 User Guide (Neuhauser and Kanipe, 1992).

5.2 Expected Outputs

For each run of code, RADTRAN 4 provides three main categories of output: an input echo that reprints exactly what was input into the code; a reformatted data input file that is reorganized for easy, quick lookup of data; and the output of RADTRAN 4 calculations, consisting of incident-free and accident summaries, and an importance analysis summary of incident-free transport. Examples of RADTRAN outputs are included in the RADTRAN 4 User Guide (Neuhauser and Kanipe, 1992).

5.3 Verification Documentation

Verification of RADTRAN 4 is routinely performed to confirm that dosimetric models in the code have been correctly coded and that the code correctly performs the operations specified in the numerical models. Quality-assurance checks are made before the release of each revision of the code. Computer benchmark tests are performed comparing results of the revised RADTRAN 4 code with the NUREG-0170 data set created with the original RADTRAN code and other test files. The NUREG-0170 data set, consisting of a previously compiled data input file, is a reliable source for gauging the consistency of code revisions because of its large scope. The NUREG-0170 findings are substantiated and considered conservative by Fischer et al. (1987) and Lahs (1987). There are seven addition test data files used as benchmarks for RADTRAN 4. These files are named 4TE1.DAT, 4TE2.DAT, 4TE3.DAT, 4TE4.DAT, 4TE5.DAT, 4TE6.DAT, 4TE7.DAT. They test various aspects of RADTRAN. The test files include (a) an aggregate data file using the DEFINE keyword, (b) a rail transport file, (c) a LINK data file, (d) a file that uses twenty accident severity categories, (e) a aggregate data file using the internal isotope library, (f) a data file that uses Pasquill stability categories, and (g) a data file that tests calculation of neutron doses.

Results from RADTRAN 4 have been compared with results from other risk assessment computer codes. For example, accident cases have been compared with cases analyzed with TREC II. RADTRAN and TREC II were found to produce the same answers (Franklin, 1980). The RADTRAN 4 code for calculating the dose from transport under incident-free conditions is not duplicated elsewhere and therefore has not been verified in this way.

RADTRAN 4 is maintained by and in the custody of the SNL Transportation System Development Department in Albuquerque, New Mexico. A full-time computer programmer implements maintenance procedures on RADTRAN 4. These procedures include enhancing code efficiency and correcting minor errors discovered in code operations. All changes are fully recorded in the RADTRAN 4 programmer's logbook for later review.

When major changes are made to the code or when the code is undergoing a new release, results of the revised code are routinely benchmarked by computer against output from older versions of the code. A 10-percent-or-less difference in results is acceptable to allow for hardware differences and new code refinements. A greater-than-10-percent difference is documented for review.

6.0 ERROR AND CODE LIMITATION REPORTING REQUIREMENTS

If errors in the code are discovered, the task leader of RADTRAN 4 operations in the SNL Risk Assessment and Transportation System Analysis Division authorizes the SNL programmer to correct the problem. Any change to the code is documented in the logbook. The logbook contains write ups of changes made to RADTRAN. There is a hard copy version which also contains comparisons of test data files. Any changes to the test files are noted and are referenced to changes in the coding. The changes to the code are also kept in a data base format.

Potential problems or errors in the code that are discovered by users can be brought to SNL's attention for review. Formal questions from regulatory bodies may be directed to Bob Luna or Sieglinde Neuhauser at SNL at the address and telephone number below. Errors in documentation, inputs, model logic, or outputs can also be reported to the following address:

Sandia National Laboratories
Division 6321
Attn: Sieglinde Neuhauser
Albuquerque, NM 87185

RADTRAN personnel may be contacted by phone during business hours:

Sieglinde Neuhauser, (505) 845-8246.

When minor changes in code efficiency or maintenance are made by the SNL programmer, these changes are documented in the RADTRAN 4 logbook but do not require management authorization. When interim changes are made (such as updates and minor corrections) that improve code operating capability, RADTRAN 4 users are notified. Major model or architectual revisions of the code result in a release of a new version of the code. When a new version of the code is released, a new version number is assigned and new documentation is created.

The RADTRAN 4 computer code may not be electronically transmitted. VAX-executable versions of the code may be mailed to users, or users may use RADTRAN 4 on the TRANSNET system by modem. Output of the RADTRAN 4 computer code is produced in an 80-column format and may be downloaded from the TRANSNET system to the user's personal computer.

7.0 SOURCES OF INFORMATION ABOUT RADTRAN 4

7.1 RADTRAN 4 Documentation

Other RADTRAN 4 documentation currently in preparation includes an Executive Summary, a Technical Manual, and a User Manual (Neuhauser and Kanipe, in preparation). The Executive Summary summarizes the code's capabilities, possible applications, and advantages. The Technical Manual documents the mathematical models and numerical methods of RADTRAN 4 and facilitates technical review of the theories and means of solution in the code. The User's Manual describes the required data and control inputs, input sequences, options, program limitations, and other issues necessary for program execution.

7.2 Previous RADTRAN Documentation

Information concerning previous versions of RADTRAN may be found in "RADTRAN III" (Madsen et al., 1986); "RADTRAN II: Revised Computer Code to Analyze Transportation of Radioactive Material" (Taylor and Daniel, 1982); "RADTRAN II User Guide" (Madsen et al., 1983); and "RADTRAN: A Computer Code to Analyze Transportation of Radioactive Material" (Taylor and Daniel, 1977). These are available from the SNL Transportation System Technology Division.

7.3 Inquiries

Inquiries and comments concerning the RADTRAN 4 code are welcome and should be addressed to

Sandia National Laboratories, Div. 6321
Attn: Sieglinde Neuhauser
Albuquerque, NM 87185

Sieglinde Neuhauser may be contacted by phone during business hours at (505) 845-8246.

REFERENCES

Cashwell, J. W., 1989. "TRANSNET--Access to Transportation Models and Databases," SAND89-0982C, Sandia National Laboratories, Albuquerque, NM.

DEC (Digital Equipment Corporation), 1988. VAX FORTRAN User Manual, AA-D035E-TE, Maynard, Massachusetts: Digital Equipment Corporation.

DEC (Digital Equipment Corporation), 1989a. VMS System Manager's Manual, AA-LA00B-TE, Maynard, Massachusetts: Digital Equipment Corporation.

DEC (Digital Equipment Corporation), 1989b. VMS User's Manual, AA-LA98B-TE, Maynard, Massachusetts: Digital Equipment Corporation.

DEC (Digital Equipment Corporation), 1989c. VMS Version 5.2 New Features Manual, AA-LA97B-TE, Maynard, Massachusetts: Digital Equipment Corporation.

DOE (U.S. Department of Energy), 1988a. "External Dose-Rate Conversion Factors for Calculation of Dose to the Public," DOE/EH-0070, U.S. Department of Energy, Washington, DC.

DOE, 1988b. "Internal Dose Conversion Factors for Calculation of Dose to the Public," DOE/EH-0071, U.S. Department of Energy, Washington, DC.

Dunning, D. E., 1983. "Estimates of Internal Dose Equivalent From Inhalation and Ingestion of Selected Radionuclides," WIPP-DOE-176, Evaluation Research Corporation, Oak Ridge, TN.

Finley, N. C., et al., 1980. "Transportation of Radionuclides In Urban Environs: Draft Environmental Assessment," NUREG/R-0743, SAND79-0369, U.S. Nuclear Regulatory Commission, Washington, DC.

Fischer, L. E., C. K. Chou, M. A. Gerhard, C. Y. Kimura, R. W. Martin, R. W. Mensing, M. E. Mount, and M. C. Witte, 1987. "Shipping Container Response to Severe Highway and Railway Accident Conditions," NUREG/CR-4829, Lawrence Livermore National Laboratory, Livermore, CA.

Fong, K. W., T. H. Jefferson, and T. Suyehiro, 1982. "SLATEC Common Mathematical Library Source File Format," UCRL-53313, Lawrence Livermore National Laboratory, Livermore, CA.

Fong, K. W., T. H. Jefferson, T. Suyehiro, and L. Walton, in preparation. "Guide to the SLATEC Common Mathematical Library," Sandia National Laboratories, Livermore, Livermore, CA.

Franklin, A. L., 1980. "TREC II: A Computer Program for Transportation Risk Assessment," PNL-3208, Pacific Northwest Laboratory, Richland, WA.

ICRP 38 (International Commission on Radiological Protection, Publication 38), 1983. "Radionuclide Transformations, Energy and Intensity of Emissions," Publication 38, Annals of the ICRP, Volumes 11-13, Oxford: Pergamon Press.

Lahs, W. R., 1987. "Transporting Spent Fuel: Protection Provided Against Severe Highway and Railroad Accidents," Division of Regulatory Applications, Nuclear Regulatory Commission, Washington, DC.

Madsen, M. M., J. M. Taylor, R. M. Ostmeyer, and P. C. Reardon, 1986. "RADTRAN III," SAND84-0036, TTC-0470, Sandia National Laboratories, Albuquerque, NM.

Madsen, M. M., E. L. Wilmot, and J. M. Taylor, 1983. "RADTRAN II User Guide," SAND82-2681, Sandia National Laboratories, Albuquerque, NM.

Neuhauser, K. S., and F. L. Kanipe, 1992. "RADTRAN 4, Volume III: User Guide," SAND89-2370, TTC-0943, Sandia National Laboratories, Albuquerque, NM.

Neuhauser, K. S., and F. L. Kanipe, in preparation. "RADTRAN 4, Volume II: Technical Manual," Sandia National Laboratories, Albuquerque, NM.

NRC (U.S. Nuclear Regulatory Commission), 1977. "Final Environmental Statement on the Transportation of Radioactive Material by Air and Other Modes," 2 Vols., NUREG-0170, U.S. Nuclear Regulatory Commission, Washington, DC.

Taylor, J. M., and S. L. Daniel, 1977. "RADTRAN: A Computer Code to Analyze Transportation of Radioactive Material," SAND76-0243, Sandia National Laboratories, Albuquerque, NM.

Taylor, J. M., and S. L. Daniel, 1982. "RADTRAN II: Revised Computer Code to Analyze Transportation of Radioactive Material," SAND80-1943, Sandia National Laboratories, Albuquerque, NM.

DISTRIBUTION:

U.S. Department of Energy (2)
1000 Independence SW
Washington, DC 20585
Attn: J. Williams, RW-321
W. Lake, RW-323

U.S. Department of Energy (3)
Washington, DC 20545
Attn: L. Harmon, EM-53
F. Falci, EM-50.1
C. Miller, EM-50.1

U.S. Department of Energy
Albuquerque Operations Office
Albuquerque Headquarters
PO Box 5400
Albuquerque, NM 87115
Attn: K. Golliher

6000 D. L. Hartley
6600 J. B. Woodard
6603 R. E. Luna
Attn: TTC Master File
6641 R. P. Sandoval
6641 K. S. Neuhauser (15)
6641 F. K. Kanipe (5)
7141 S. A. Landenberger (5)
7151 G. C. Claycomb (3)
7613-2 Doc. Processing for DOE/OSTI (8)
8523-2 Central Technical Files