DKR-STABLE: A Code Which Calculates Stable Isotope Concentrations Using the Linear Chain Method

A. White and W.F. Vogelsang

November 1984

UWFDM-606

FUSION TECHNOLOGY INSTITUTE

UNIVERSITY OF WISCONSIN

MADISON WISCONSIN

DKR-STABLE: A Code Which Calculates Stable Isotope Concentrations Using the Linear Chain Method

A. White and W.F. Vogelsang

Fusion Technology Institute
University of Wisconsin
1500 Engineering Drive
Madison, WI 53706

http://fti.neep.wisc.edu

November 1984

UWFDM-606

DKR-STABLE: A CODE WHICH CALCULATES STABLE ISOTOPE CONCENTRATIONS USING THE LINEAR CHAIN METHOD

A. White and W.F. Vogelsang

Fusion Technology Institute
1500 Johnson Drive
University of Wisconsin-Madison
Madison, Wisconsin 53706

November 1984

UWFDM-606

1. Introduction

In a fusion reactor, there are several reasons why knowledge of the inventory of stable isotopes is necessary. One such reason is the determination of the physical properties of the materials in the system. The strength and ductility of a material can be changed if its composition is changed. The swelling rate of the material could also be altered by a change in the isotopic fractions of the material. The electrical properties of the conductor can be affected greatly by the inclusion of a small amount of impurity. As an example of this, if only 2% of the copper in a pure copper wire transmutes to nickel, the resistivity increases by a factor of 2.9. If 6% transmutes, the resistivity will increase by a factor of 8.9. The additional resistivity would not only have a great effect on the power requirements of the system, but could also generate enough heat to cause the magnets to melt.

The chain construction method used in the DKR-STABLE code is given in the flowcharts in Fig. 1. The method uses the same logic as that of the DKR code, but several modifications have been made in order for it to run properly. The test for the number of stable nuclides per chain has been eliminated. This was done to allow longer chains to form and more stable isotopes to be created. The elimination of this test means that the criterion for chain termination is based entirely on the ratio of the number densities of the initial and final isotopes. If the final isotope is radioactive, the chain will not be terminated. The size of the arrays for chain data storage has been doubled to allow the code to handle twice as many chains and the storage scheme has been modified for more efficient use of memory. Certain arrays, containing data pertinent only to radioactive decay, have been eliminated.

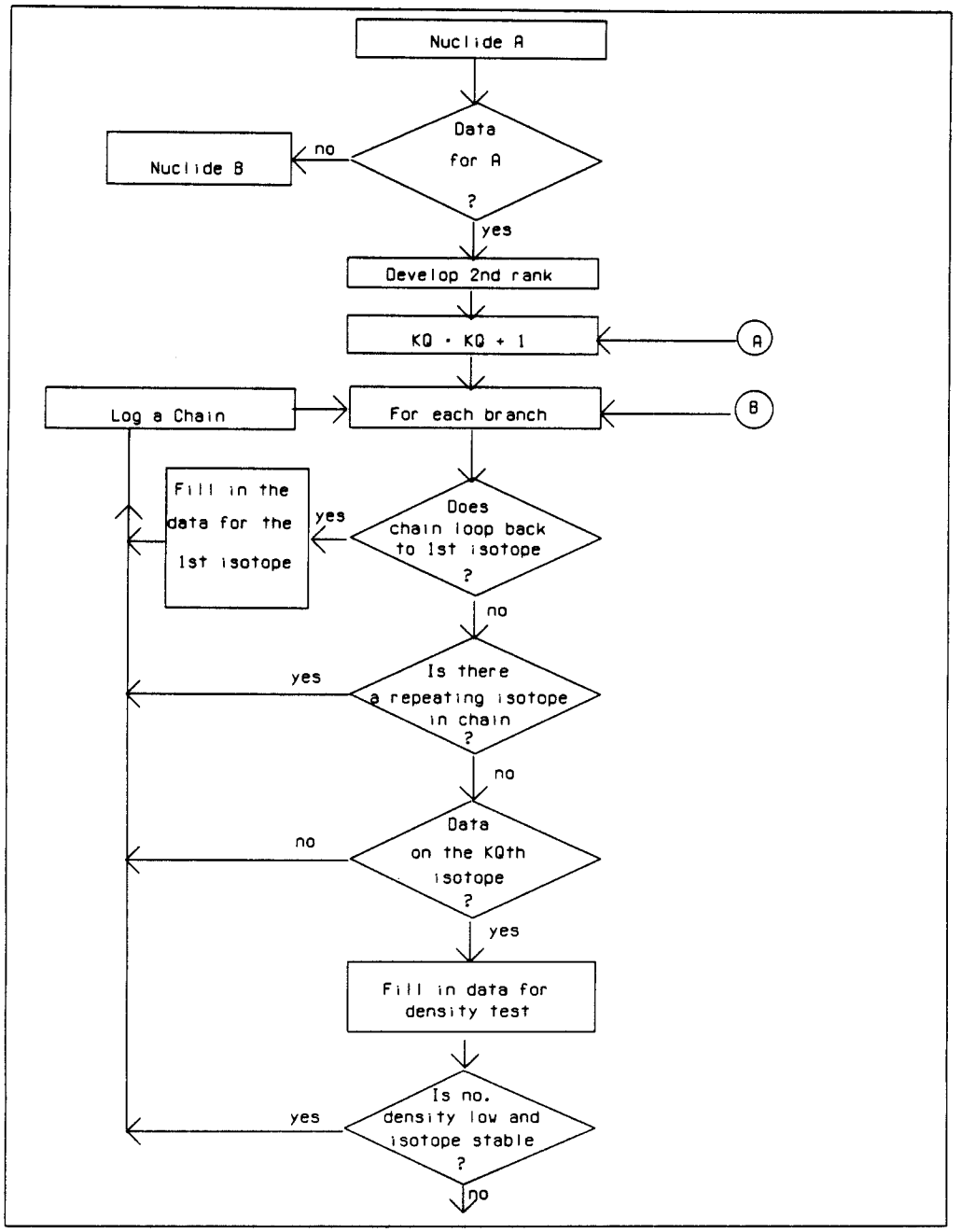


Fig. 1a Stable Chain Construction Flowchart.

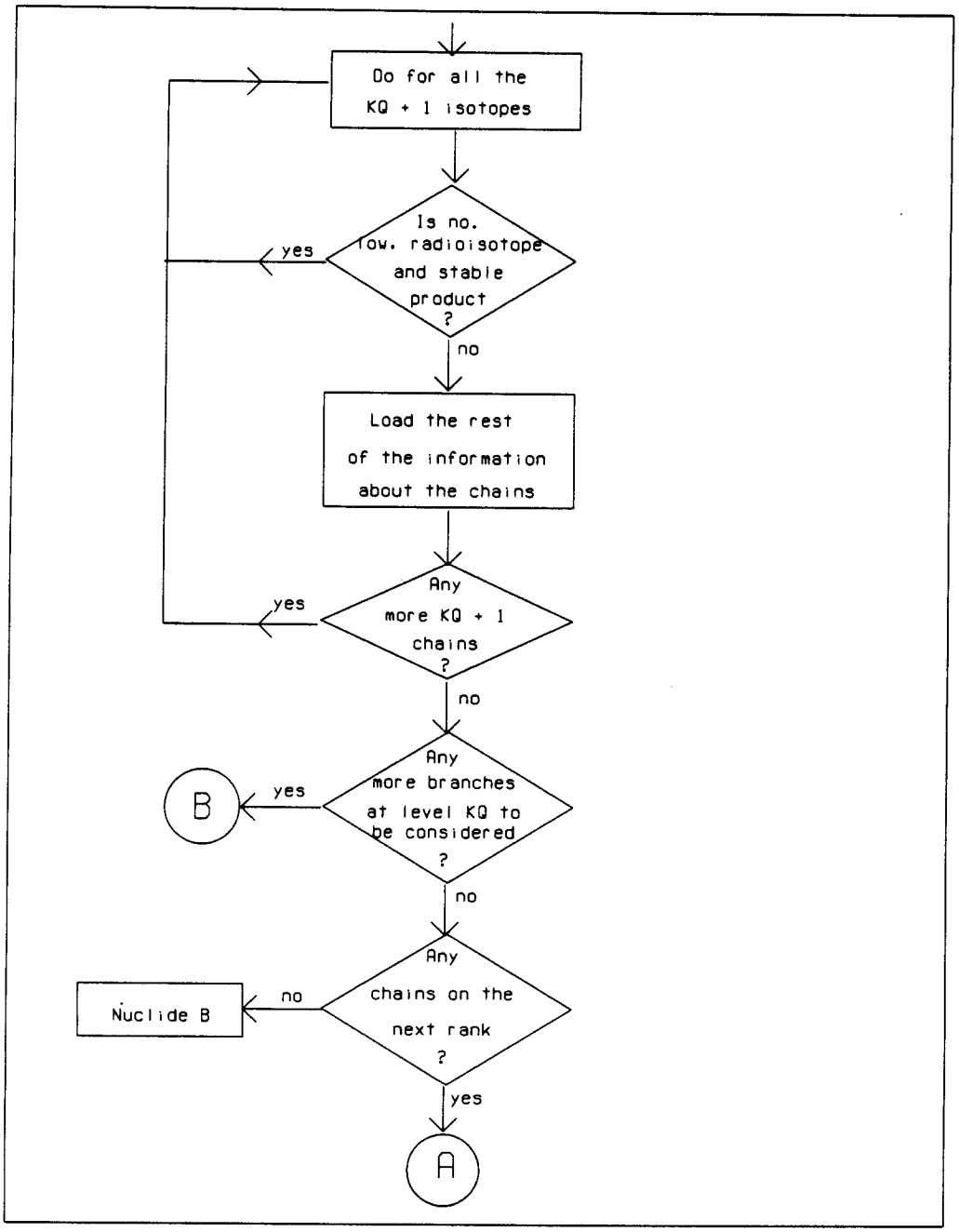

Do for all the KQ + 1 isotopes
Is no. low. radioisotope and stable product ?
yes
no
Load the rest of the information about the chains
Any more KQ + 1 chains ?
yes
no
Any more branches at level KQ to be considered ?
B
yes
no
Any chains on the next rank ?
Nuclide B
no
yes
A

Fig. 1b

It can be seen that the chain formation scheme is based on the forward construction approach, i.e. the starting point is one of the stable nuclides input by the user. The reason that this method is used is that there is no bias toward any resultant isotope (the only response is the final number density). Since the forward method is more efficient in the setup of the chains (only the input nuclides need be considered, instead of all the possible resultant, stable ones), it is the method of choice.

A sample calculation has been done using the DKR-STABLE code to determine the production of stable isotopes in the Mirror Advanced Reactor Study (MARS). The input and output for this case are given in Appendix 1. The input is identical with that of DKR. The output first gives a summary of the input parameters and the cross section and decay data present in the external library. Next, the chains which DKR has constructed are given. This is followed by the stable isotope concentration in each interval. The result is then volume averaged over each zone and the volume averaged isotopic density is output. Finally the atomic parts per million of each isotope is output. Another calculation has been done for the structural portion of the MARS reactor. An analysis of the results showed no significant production of any impurity which would result in structural problems.

2. Calculational Method and Computer Implementation

A complete discussion of the calculation method used in DKR can be found in UWFDM-170. The scheme involves the resolution of the system into many linear chains. These chains are then solved using a recursion coefficient program.

3. DKR-STABLE Input/Output

3.1 Description of Input

A description of the input is given below. A sample of the input can be found in Appendix 1.

Section 1 (SPARSE DATA)

Card 1 (20A4)
  Title
Card 2a (A6)
  The word "PRIOPT"
Card 2b (unformatted)
  LPRT1 - 0 (not used)
  LPRT2 - print option for index file of nuclide
          1 - on
          0 - off
  LPRT3 - 0 (not used)
  LPRT4 - print option for chain results and densities by interval
          1 - on
          0 - off
  LFLX - reference flux option
          0 - uniform flux of 10^{14} n/cm^2-s
          1 - first wall of UWMAK-I
          2 - user supplied reference flux
Card 3a (A6)
  The word "INITAL"
Card 3b (unformatted)
  LID - run ID number
  LNK - link to other solutions
          0 - construction of linear decay chains only
          1 - calculation of stable isotope number densities
          2 - same as 1 except decay chains and data tables from the
              previous run are used
          3 - same as 1
          4 - same as 2
  LGE - problem geometry
          1 - slab
          2 - cylinder
          3 - sphere
          4 - torus
  LFX - flux format
          1 - DKR format
          2 - ANISN format
  IZM - number of zones
  INT - number of intervals
  IORG - organization method used
          0 - neglect lower density input nuclides
          1 - include everything

Card 4a (A6)
The word "TIMCOM"
Card 4b (unformatted)
NOP - number of operating times
NAS - number of after shutdown times
NCMP - number of composition
IGN - number of neutron groups (46)
IGG - number of gamma groups (43)
Card 5a (A6)
The word "REFFOP"
Card 5b (unformatted)
LFCF - flag for flux conversion factor to be multiplied by flux
FCF - flux conversion factor
Card 6a (A6)
The word "HEATNG"
Card 6b (unformatted)
WLLD - total wall loading (MW/m^2)
HTN - neutron heating (MeV)
HTG - gamma heating (MeV)
HTT - total nuclear heating (MeV)
Card 7a (A6)
The word "RADIUS"
Card 7b (unformatted)
RRP - plasma radius
RRW - first wall radius
RRT - toroidal radius (input as 0. if not toroidal)

The order of input of cards 2-7 is arbitrary as long as the correct "b" card follows each "a" card.

Section 2 (Zone-Interval Structure)

Card 1 (IZM cards; unformatted)
IZ - zone number
NINT - number of intervals per zone
LCAL - flag for radioactivity calculation
RI - inner radius of zone
RO - outer radius of zone

Section 3 (Nuclide Composition Data)

Card 1 (unformatted)
COMP - a 4 letter word indicating the type of composition
NUM - number identification of composition
LEN - number of composition cards to follow - LEN=0 if a preprogrammed composition is used.

The input now splits into two cases: (A) if LEN = 0 and (B) if LEN > 0.

Section 3a (LEN=0)

Card 2a (optional, A7)
The word "NEGLECT" if an isotope of the preprogrammed composition is to be neglected or a blank card if all are to be used
Card 3a (unformatted)
ISOT - the isotope to be neglected
Card 4a (blank)

Section 3b (LEN > 0)

Card 2b (LEN cards; unformatted)
ISO - an element (Z) or isotope number (1000*Z+A)
NUMB - a number density

Section 4 (Zone Composition Data)

Card 1 (NCMP cards; unformatted)
COMP(IZ,CMP) - fraction of CMP in zone IZ

Section 5 (Times)

Card 1 (NOP cards; unformatted)
BOP - alphanumeric expression for time of operation
TOP - time of operation in seconds
Card 2 (NAS cards; unformatted)
BAS - alphanumeric expression for time after shutdown
TAS - time after shutdown in seconds

Section 6 (Reference Flux, LFLX=2)

Card 1 (20A4)
REFTL - reference flux title
Card 2 (6E12.3)
REFFLUX(1) - REFFLUX(IGN) - reference flux

3.2 Output

The first section of the output prints out the input data for DKR. The flux data is summarized and flux data title is output.

The second part of the output is the nuclear data library table which shows the reactions included in the external library. Next, if LPRT2=1, the nuclear data index table is printed. These are the reaction rates computed with the reference flux.

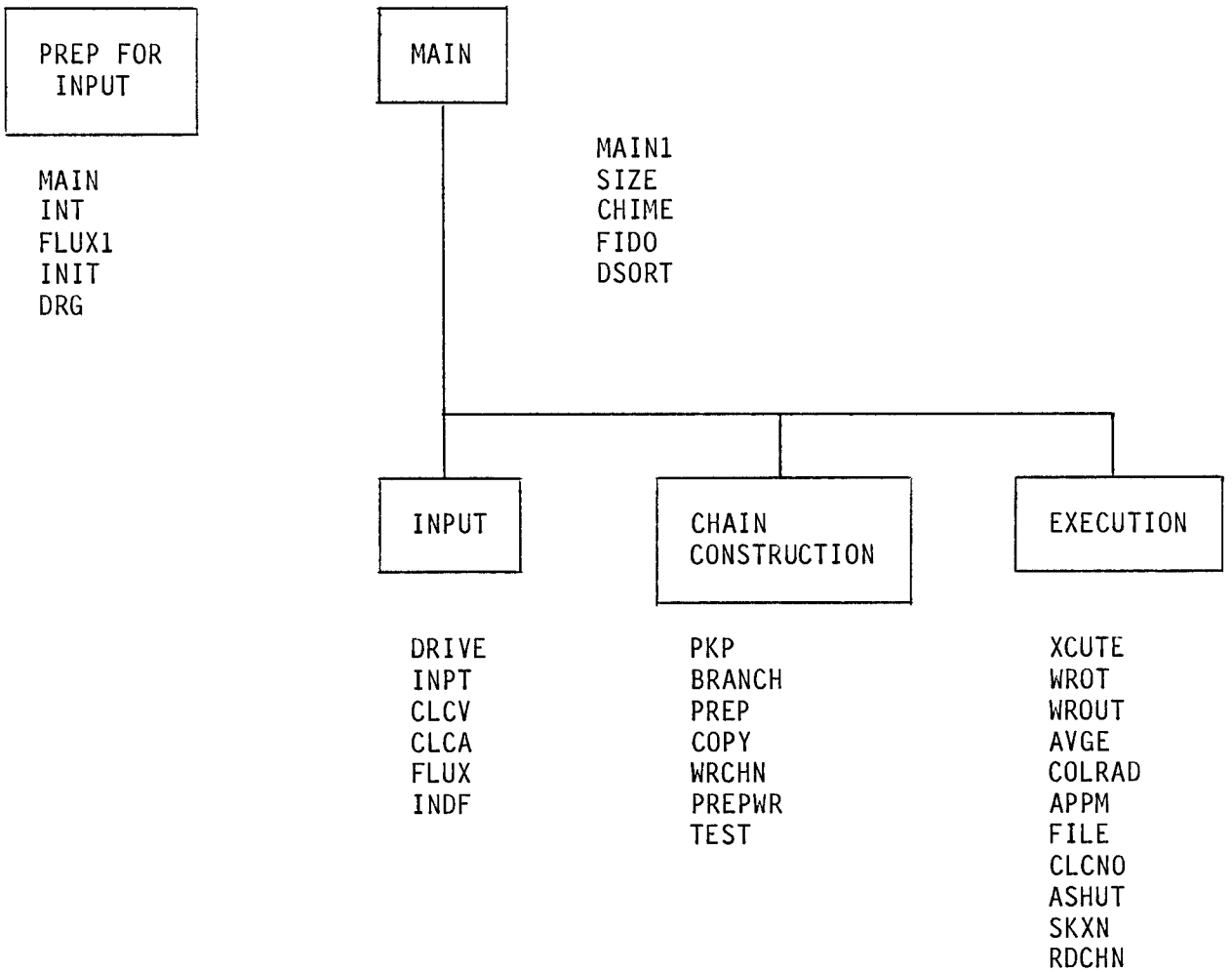
The next section shows the chain construction procedure. The chains constructed are given in the output. If LPRT4=1 the solution of each chain for each interval and the stable isotope density in each interval will be printed out.

Next, the volume averaged density of each zone is output and finally the atomic parts per million in each zone is given.

A sample of the output can be found in Appendix 1.

4. User's Guide

4.1 Program Structure



4.2 Subroutines

A short description of the major subroutines is given here.

MAIN, INT, FLUX1, INIT, ORF: Process the input deck into the old DKR format.

MAIN1: Control routine for the rest of the program. It also sets the dimensioning for the chain construction and execution portions of the program.

SIZE: Approximate core storage requirements are given.

PSORT: General one-dimensional array sorting routine.

DRIVE: Control routine for the input of system parameters, flux and cross sections.

INPT, CLCV, CLCA: Input data is read in to specify the system geometry and composition and the interval volume and first wall area are computed.

FLUX, INDF, FIDO: The flux and cross sections are read and stored on an external unit.

PKP, WRCHN: PKP constructs the linear chain which we then write onto an external unit by WRCHN.

BRANCH: Retrieves and organizes the reaction rate data.

PREP and TEST: Determine whether the chain should continue based on the final isotope density.

XCUTE, SKXN, RDCHN: Subroutine XCUTE computes the isotopic densities using chain data retrieved by RDCHN and cross sections retrieved with SKXN.

WRDT, WROUT, AVGE, APIM: Output respectively the chain solution, interval isotopic densities, volume averaged density, and atomic parts per million.

COLRAD: Collects the chain data computed in XCUTE.

CLCNO, ASHUT: Solve the chains for times of operation and times after shutdown, respectively.

Computer Code Abstract for DKR-STABLE

1. Name of Code:

   DKR-STABLE

2. Coding Language and Computer:

   FORTRAN 66 or FORTRAN 77; Cray with CFT

3. Description of Problem:

   The major purpose of DKR-STABLE is to compute the stable isotope density in a fusion reactor. This is done through the use of the linear chain method and the cross section library DCDLIB.[1] The knowledge of transmutations in the structure of the reactor as well as in the normal or superconducting magnets is necessary in any reactor design study.

4. Method of Solution:

   The activation of a nuclide can be represented by linear chains which can be solved by a recursion coefficient formula. This solution is used to compute the stable nuclide density of all isotopes.

5. Restriction on the Complexity of the Problem:

   Total memory space is unrestricted and variable dimensioning is used to provide flexibility in system parameters. DKR-STABLE accommodates nuclear data in DCDLIB which has a 46 neutron group structure and allows 29 types of transmutations.

6. Typical Running Time:

   Running time is problem dependent. The driving factors are number of initial nuclides and number of intervals in the system.

7. Unique Features of the Program:

   DKR-STABLE constructs the linear decay chain by itself with nuclear data from DCDLIB. Volume averaged stable isotope densities and APPM are output for each zone.

8. Related Programs:

   DKR[2] is a program which computes the activity, biological hazard potential, and afterheat.

9. Machine Requirements:

   DKR-STABLE was written with an updated version of FORTRAN 66 on the Cray-1. In addition to standard input and output units, several logical units are required.

10. References:

   1) T.Y. Sung and W.F. Vogelsang, "Decay Chain Data Library Radioactivity Calculations," University of Wisconsin Fusion Technology Institute Report UWFDM-171 (September 1976).

   2) T.Y. Sung and W.F. Vogelsang, "DKR: A Radioactivity Calculation Code for Fusion Reactors," University of Wisconsin Fusion Technology Institute Report UWFDM-170 (September 1976).

Acknowledgement

Support for this work has been provided by the U.S. Department of Energy.

APPENDIX 1

DKR-STABLE INPUT AND OUTPUT

```
  this is another mars study
priopt
1 0 1 1 1
inital
1 3 2 1 2 68 0
timcom
1 0 1 46 43
reffop
1 7.18e-08
heatng
4.3 7.73 11.73 19.46
radius
44.0 60.0 0.0
1 55 0 60.0 185.0
2 13 1 185.0 200.0
pbal 1 3
82 2.967e+22
13 2.910e+21
12 3.010e+19
0.00
1.00
2yr 6.307e+07
 This is the MARS flux for 8/17/83
int  1
  4.7930e-03  1.0790e-04  4.2311e-05  6.2291e-05  8.6167e-05  8.2294e-05
```

```
     6943 is the number of real variables in segment 1

     6395 is the number of integer variables in segment 1

     9063 is the number of real variables in segment 2

     8483 is the number of integer variables in segment 2

   146559 is the number of real variables in segment 3

     7581 is the number if integer variables in segment 3

     5315 is the number of real variables in segment 4

     6395 is the number of integer variables in segment 4


 If any of the integer amounts should exceed 150,000 or the
 real amounts 250,000 extreme caution is advised
1  this is another mars study

                        problem run id       1
        lnk   link to the other solution          3
        lge   1/2/3   = slab/cyl/sph               2
        lfx   1/2     = tk3/scalar(anisn)         1
        izm   number of zones                      2
        int   number of intervals                 68
        nop   number of operating times           1
        nas   number of after shutdown times     12
        nnc   number of materials(nuclides)       8
        ncmp  number of composition table         1
        ign   number of neutron groups          * 46
        igg   number of photon groups            * 43

                    zone   1   2
                12024          *
                12025          *
                12026          *
                13027          *
                82204          *
                82206          *
                82207          *
                82208          *
1                     reactor system parameters

        radius of the plasma       44.00         cm
        radius of the first wall   60.00         cm
        radius  of the torus        0.00         cm


        first wall area            3.770e-02     m2
        neutron wall loading       4.300e+00     mw/m2
        total operating power      2.237e-01     mw
        flux conversion factor     7.180e-08


        operating time    1
+                                        after shutdown time  12

         2yr      6.307e+07  second
+                                                    0      0.          second
                                                     1 m    6.000e+01   second
                                                    10 m    6.000e+02   second
                                                     1 hr   3.600e+03   second
                                                     6 hr   2.160e+04   second
                                                     1 dy   8.640e+04   second
                                                     1 wk   6.048e+05   second
                                                     1 mo   2.630e+06   second
                                                     1 yr   3.156e+07   second
                                                    10 yr   3.156e+08   second
                                                   100 yr   3.156e+09   second
```

1000yr 3.156e+10 second

volume of zone

zone 1 9.621e+04 cm3
zone 2 1.814e+04 cm3

1 nuclide no. density(10**18)

kza	zone 1	2
12024	0.0	23.7
12025	0.0	3.0
12026	0.0	3.4
13027	0.0	2910.0
82204	0.0	439.1
82206	0.0	7002.0
82207	0.0	6705.0
82208	0.0	15520.0

reference flux
first wall flux of uwmak-i

first interval = 1

flux reading

68 intervals read from flux (68, 46)
This is the MARS flux for 8/17/83

1 nuclear data table

lkza	nkt	kt=	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	21	22	23	24	25	26	27	28	29
80160	5	s	x	x	x		x			x																				
80190	2	r	o	o																										
90180	2	r	o		o																									
90190	8	s	x	x	x	x	x	x		x	x																			
90200	2	r	o	o																										
100230	2	r	o	o																										
110240	2	r	o	o																										
110250	2	r	o	o																										
110260	2	r	o	o																										
120240	3	s	x	x	x																									
120250	3	s	x	x	x																									
120260	4	s	x	x	x					x																				
120270	2	r	o	o																										
130260	2	r	o		o																									
130270	8	s	x	x	x	x	x	x		x											x									
130280	2	r	o	o																										
130290	2	r	o	o																										
130300	2	r	o	o																										
130301	2	r	o				o																							
140280	4	s	x	x	x					x																				
140290	4	s	x	x	x					x																				
140300	4	s	x	x	x					x																				
140310	2	r	o	o																										
170360	3	r	o	o	o																									
170380	2	r	o	o																										
180390	2	r	o	o																										
180410	2	r	o	o																										
190380	2	r	o		o																									
190390	7	s	x	x	x	x	x			x	x																			
190401	3	r	o	o	o																									
190410	3	s	x		x					x																				
200450	2	r	o	o																										
200470	2	r	o	o																										
210450	7	s	x		x	x	x	x		x	x																			
210460	2	r	o	o																										

```
     501160  7   s    x  x  x  x  x        x  x
     501170  8   s    x  x  x  x  x  x     x  x
     501171  2  r     o           o
     501180  7   s    x  x  x  x  x        x  x
     501190  8   s    x  x  x  x  x  x     x  x
     501191  2  r     o           o
     501200  6   s    x  x  x  x  x        x
     501210  2  r     o  o
     501220  6   s    x  x  x  x  x        x
     501230  4   s    x  x
                  r     o  o
     501240  5   s    x  x  x  x  x
     501250  4   s    x  x
                  r     o  o
     501260  4   s    x  x
                  r     o  o
     511250  2  r     o  o
     721810  2  r     o  o
     721830  2  r     o  o
     731810  4   s    x  x  x  x
     731820  2  r     o  o
     731821  2  r     o           o
     731830  2  r     o  o
     731840  2  r     o  o
     731850  2  r     o  o
     731860  2  r     o  o
     741810  2  r     o     o
     741820  6   s    x  x  x  x  x        x
     741830  6   s    x  x  x  x  x        x
     741840  6   s    x  x  x  x  x        x
     741850  2  r     o  o
     741860  6   s    x  x  x  x  x        x
     741870  2  r     o  o
     802030  2  r     o  o
     802050  2  r     o  o
     812040  2  r     o  o
     822030  2  r     o     o
1                              nuclear data table

      lkza  nkt  kt=  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
                       21 22 23 24 25 26 27 28 29

     822040  7   s    x     x  x  x  x     x  x
     822050  2  r     o     o
     822060  7   s    x     x  x  x  x     x  x
     822070  7   s    x     x  x  x  x     x  x
     822080  8   s    x  x  x  x  x  x     x  x
     822090  2  r     o  o
     832080  2  r     o  o
     832090  3   s    x  x     x
     832100  2  r     o  o
     842100  2  r     o        o
   120250
                                                 120250 test    0
     101
        120240   102  0.           2.2446e-11  0.           0.            101  1
        120250   101  1.0156e-21   6.5457e-12  0.           0.            101  2
   110240
                                                 110240 test    0
   120240
     102
        120240   103  1.2870e-05   2.2446e-11  1.2870e-05   0.            102  1
        110240  1022  2.2446e-11   1.2870e-05  0.           1.2870e-05    102  2
   120260
                                                 120260 test    0
     201
        120250   102  0.           6.5457e-12  0.           0.            201  1
        120260   101  1.0156e-21   6.4418e-12  0.           0.            201  2
   110250
                                                 110250 test    0
   120250
     202
```

```
      120250   103  1.1550e-02  6.5457e-12  1.1550e-02  0.            202  1
      110250  1022  6.5457e-12  1.1550e-02  0.          1.1550e-02    202  2
120270
                                           120270 test    0
110260
                                           110260 test    0
100230
                                           100230 test    0
130270
                                           130270 test    0
  301
      120260   102  0.          6.4418e-12  0.          0.            301  1
      120270  1022  1.0156e-21  1.2210e-03  0.          1.2210e-03    301  2
      130270   101  1.2210e-03  3.5110e-11  1.2210e-03  0.            301  3
120260
  302
      120260   103  6.9300e-01  6.4418e-12  6.9300e-01  0.            302  1
      110260  1022  2.8053e-12  6.9300e-01  0.          6.9300e-01    302  2
110230
130280
                                           130280 test    0
120270
                                           120270 test    0
130260
                                           130260 test    0
120260
                                           120260 test    0
  401
      130270   105  0.          3.5110e-11  0.          0.            401  1
      120260   101  2.9933e-12  6.4418e-12  0.          0.            401  2
120250
                                           120250 test    0
  402
      130270   106  0.          3.5110e-11  0.          0.            402  1
      120250   101  1.7697e-14  6.5457e-12  0.          0.            402  2
110240
                                           110240 test    0
130261
140280
                                           140280 test    0
  403
      130270   102  0.          3.5110e-11  0.          0.            403  1
      130280  1022  9.3216e-13  5.0200e-03  0.          5.0200e-03    403  2
      140280   101  5.0200e-03  4.3759e-11  5.0200e-03  0.            403  3
130270
  404
      130270   103  1.2210e-03  3.5110e-11  1.2210e-03  0.            404  1
      120270  1022  1.1072e-11  1.2210e-03  0.          1.2210e-03    404  2
120260
                                           120260 test    0
  405
      130270   104  0.          3.5110e-11  0.          0.            405  1
      130260  1023  2.4416e-12  2.9680e-14  0.          2.9680e-14    405  2
      120260   101  2.9680e-14  6.4418e-12  2.9680e-14  0.            405  3
120240
                                           120240 test    0
  406
      130270   108  0.          3.5110e-11  0.          0.            406  1
      110240  1022  1.5226e-11  1.2870e-05  0.          1.2870e-05    406  2
      120240   101  1.2870e-05  2.2446e-11  1.2870e-05  0.            406  3
812040
                                           812040 test    0
822030
                                           822030 test    0
812030
812020
802010
802000
822040
  501
      822040   103  5.3400e-09  2.9844e-10  5.3400e-09  0.            501  1
      812040  1022  7.3203e-14  5.3400e-09  0.          5.3400e-09    501  2
```

802030 test 0

812050
812030

1 executing procedures for zone 2

lkza	lrz	ai	ak	bi	bk	yo	yt	z,i	op
mxc = 2									
									< 2, 1>2yr
120240	102	0.	2.079e-17	0.	0.	2.369e+19	2.369e+19		
120250	101	1.399e-17	1.500e-17	0.	0.	0.	2.090e+10		
									< 2, 2>2yr
120240	102	0.	2.028e-17	0.	0.	2.369e+19	2.369e+19		
120250	101	1.437e-17	1.525e-17	0.	0.	0.	2.147e+10		
									< 2, 3>2yr
120240	102	0.	1.962e-17	0.	0.	2.369e+19	2.369e+19		
120250	101	1.448e-17	1.525e-17	0.	0.	0.	2.163e+10		
									< 2, 4>2yr
120240	102	0.	1.878e-17	0.	0.	2.369e+19	2.369e+19		
120250	101	1.431e-17	1.498e-17	0.	0.	0.	2.138e+10		
									< 2, 5>2yr
120240	102	0.	1.776e-17	0.	0.	2.369e+19	2.369e+19		
120250	101	1.387e-17	1.446e-17	0.	0.	0.	2.073e+10		
									< 2, 6>2yr
120240	102	0.	1.657e-17	0.	0.	2.369e+19	2.369e+19		
120250	101	1.319e-17	1.370e-17	0.	0.	0.	1.971e+10		
									< 2, 7>2yr
120240	102	0.	1.522e-17	0.	0.	2.369e+19	2.369e+19		
120250	101	1.228e-17	1.273e-17	0.	0.	0.	1.835e+10		
									< 2, 8>2yr
120240	102	0.	1.372e-17	0.	0.	2.369e+19	2.369e+19		
120250	101	1.116e-17	1.155e-17	0.	0.	0.	1.668e+10		
									< 2, 9>2yr
120240	102	0.	1.207e-17	0.	0.	2.369e+19	2.369e+19		
120250	101	9.856e-18	1.019e-17	0.	0.	0.	1.473e+10		
									< 2,10>2yr
120240	102	0.	1.031e-17	0.	0.	2.369e+19	2.369e+19		
120250	101	8.386e-18	8.679e-18	0.	0.	0.	1.253e+10		
									< 2,11>2yr
120240	102	0.	8.439e-18	0.	0.	2.369e+19	2.369e+19		
120250	101	6.776e-18	7.030e-18	0.	0.	0.	1.012e+10		
									< 2,12>2yr
120240	102	0.	6.487e-18	0.	0.	2.369e+19	2.369e+19		
120250	101	5.057e-18	5.276e-18	0.	0.	0.	7.556e+09		
									< 2,13>2yr
120240	102	0.	4.272e-18	0.	0.	2.369e+19	2.369e+19		
120250	101	3.055e-18	3.243e-18	0.	0.	0.	4.565e+09		
									< 2, 1>2yr
120240	103	1.287e-05	2.079e-17	1.287e-05	0.	2.369e+19	2.369e+19		
110240	1022	6.803e-18	1.287e-05	0.	1.287e-05	0.	1.224e+07		
									< 2, 2>2yr
120240	103	1.287e-05	2.028e-17	1.287e-05	0.	2.369e+19	2.369e+19		
110240	1022	5.913e-18	1.287e-05	0.	1.287e-05	0.	1.064e+07		
									< 2, 3>2yr
120240	103	1.287e-05	1.962e-17	1.287e-05	0.	2.369e+19	2.369e+19		
110240	1022	5.143e-18	1.287e-05	0.	1.287e-05	0.	9.252e+06		
									< 2, 4>2yr
120240	103	1.287e-05	1.878e-17	1.287e-05	0.	2.369e+19	2.369e+19		
110240	1022	4.472e-18	1.287e-05	0.	1.287e-05	0.	8.046e+06		
									< 2, 5>2yr
120240	103	1.287e-05	1.776e-17	1.287e-05	0.	2.369e+19	2.369e+19		
110240	1022	3.890e-18	1.287e-05	0.	1.287e-05	0.	6.998e+06		
									< 2, 6>2yr
120240	103	1.287e-05	1.657e-17	1.287e-05	0.	2.369e+19	2.369e+19		
110240	1022	3.382e-18	1.287e-05	0.	1.287e-05	0.	6.085e+06		
									< 2, 7>2yr
120240	103	1.287e-05	1.522e-17	1.287e-05	0.	2.369e+19	2.369e+19		
110240	1022	2.940e-18	1.287e-05	0.	1.287e-05	0.	5.290e+06		
									< 2, 8>2yr
120240	103	1.287e-05	1.372e-17	1.287e-05	0.	2.369e+19	2.369e+19		
110240	1022	2.555e-18	1.287e-05	0.	1.287e-05	0.	4.597e+06		

```
                                                                                                  < 2, 9>2yr
120240  103  1.287e-05 1.207e-17 1.287e-05 0.        2.369e+19 2.369e+19
110240 1022  2.218e-18 1.287e-05 0.        1.287e-05 0.        3.991e+06
                                                                                                  < 2,10>2yr
120240  103  1.287e-05 1.031e-17 1.287e-05 0.        2.369e+19 2.369e+19
110240 1022  1.923e-18 1.287e-05 0.        1.287e-05 0.        3.460e+06
                                                                                                  < 2,11>2yr
120240  103  1.287e-05 8.439e-18 1.287e-05 0.        2.369e+19 2.369e+19
110240 1022  1.663e-18 1.287e-05 0.        1.287e-05 0.        2.992e+06
                                                                                                  < 2,12>2yr
120240  103  1.287e-05 6.487e-18 1.287e-05 0.        2.369e+19 2.369e+19
110240 1022  1.430e-18 1.287e-05 0.        1.287e-05 0.        2.573e+06
                                                                                                  < 2,13>2yr
120240  103  1.287e-05 4.272e-18 1.287e-05 0.        2.369e+19 2.369e+19
110240 1022  1.217e-18 1.287e-05 0.        1.287e-05 0.        2.190e+06


   mxc =  2
                                                                                                  < 2, 1>2yr
120250  102  0.        1.500e-17 0.        0.        3.049e+18 3.049e+18
120260  101  1.399e-17 1.498e-17 0.        0.        0.        2.690e+09
                                                                                                  < 2, 2>2yr
120250  102  0.        1.525e-17 0.        0.        3.049e+18 3.049e+18
120260  101  1.437e-17 1.524e-17 0.        0.        0.        2.764e+09
                                                                                                  < 2, 3>2yr
120250  102  0.        1.525e-17 0.        0.        3.049e+18 3.049e+18
120260  101  1.448e-17 1.524e-17 0.        0.        0.        2.784e+09
                                                                                                  < 2, 4>2yr
120250  102  0.        1.498e-17 0.        0.        3.049e+18 3.049e+18
120260  101  1.431e-17 1.497e-17 0.        0.        0.        2.751e+09
                                                                                                  < 2, 5>2yr
120250  102  0.        1.446e-17 0.        0.        3.049e+18 3.049e+18
120260  101  1.387e-17 1.445e-17 0.        0.        0.        2.668e+09
                                                                                                  < 2, 6>2yr
120250  102  0.        1.370e-17 0.        0.        3.049e+18 3.049e+18
120260  101  1.319e-17 1.369e-17 0.        0.        0.        2.536e+09
                                                                                                  < 2, 7>2yr
120250  102  0.        1.273e-17 0.        0.        3.049e+18 3.049e+18
120260  101  1.228e-17 1.272e-17 0.        0.        0.        2.362e+09
                                                                                                  < 2, 8>2yr
120250  102  0.        1.155e-17 0.        0.        3.049e+18 3.049e+18
120260  101  1.116e-17 1.154e-17 0.        0.        0.        2.147e+09
                                                                                                  < 2, 9>2yr
120250  102  0.        1.019e-17 0.        0.        3.049e+18 3.049e+18
120260  101  9.856e-18 1.019e-17 0.        0.        0.        1.895e+09
                                                                                                  < 2,10>2yr
120250  102  0.        8.679e-18 0.        0.        3.049e+18 3.049e+18
120260  101  8.386e-18 8.674e-18 0.        0.        0.        1.613e+09
                                                                                                  < 2,11>2yr
120250  102  0.        7.030e-18 0.        0.        3.049e+18 3.049e+18
120260  101  6.776e-18 7.026e-18 0.        0.        0.        1.303e+09
                                                                                                  < 2,12>2yr
120250  102  0.        5.276e-18 0.        0.        3.049e+18 3.049e+18
120260  101  5.057e-18 5.273e-18 0.        0.        0.        9.725e+08
                                                                                                  < 2,13>2yr
120250  102  0.        3.243e-18 0.        0.        3.049e+18 3.049e+18
120260  101  3.055e-18 3.240e-18 0.        0.        0.        5.875e+08
                                                                                                  < 2, 1>2yr
120250  103  1.155e-02 1.500e-17 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  1.011e-18 1.155e-02 0.        1.155e-02 0.        2.610e+02
                                                                                                  < 2, 2>2yr
120250  103  1.155e-02 1.525e-17 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  8.825e-19 1.155e-02 0.        1.155e-02 0.        2.277e+02
                                                                                                  < 2, 3>2yr
120250  103  1.155e-02 1.525e-17 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  7.703e-19 1.155e-02 0.        1.155e-02 0.        1.988e+02
                                                                                                  < 2, 4>2yr
120250  103  1.155e-02 1.498e-17 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  6.717e-19 1.155e-02 0.        1.155e-02 0.        1.733e+02
                                                                                                  < 2, 5>2yr
120250  103  1.155e-02 1.446e-17 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  5.858e-19 1.155e-02 0.        1.155e-02 0.        1.511e+02
```

```
                                                                                                    < 2, 6>2yr
120250  103  1.155e-02 1.370e-17 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  5.102e-19 1.155e-02 0.        1.155e-02 0.        1.317e+02
                                                                                                    < 2, 7>2yr
120250  103  1.155e-02 1.273e-17 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  4.445e-19 1.155e-02 0.        1.155e-02 0.        1.147e+02
                                                                                                    < 2, 8>2yr
120250  103  1.155e-02 1.155e-17 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  3.868e-19 1.155e-02 0.        1.155e-02 0.        9.981e+01
                                                                                                    < 2, 9>2yr
120250  103  1.155e-02 1.019e-17 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  3.365e-19 1.155e-02 0.        1.155e-02 0.        8.682e+01
                                                                                                    < 2,10>2yr
120250  103  1.155e-02 8.679e-18 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  2.923e-19 1.155e-02 0.        1.155e-02 0.        7.543e+01
                                                                                                    < 2,11>2yr
120250  103  1.155e-02 7.030e-18 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  2.535e-19 1.155e-02 0.        1.155e-02 0.        6.542e+01
                                                                                                    < 2,12>2yr
120250  103  1.155e-02 5.276e-18 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  2.192e-19 1.155e-02 0.        1.155e-02 0.        5.655e+01
                                                                                                    < 2,13>2yr
120250  103  1.155e-02 3.243e-18 1.155e-02 0.        3.049e+18 3.049e+18
110250 1022  1.882e-19 1.155e-02 0.        1.155e-02 0.        4.857e+01


  mxc =  2
                                                                                                    < 2, 1>2yr
120260  102  0.        1.498e-17 0.        0.        3.362e+18 3.362e+18
120270 1022  1.399e-17 1.221e-03 0.        1.221e-03 0.        3.783e+04
130270  101  1.221e-03 1.051e-16 1.221e-03 0.        0.        2.913e+09
                                                                                                    < 2, 2>2yr
120260  102  0.        1.524e-17 0.        0.        3.362e+18 3.362e+18
120270 1022  1.437e-17 1.221e-03 0.        1.221e-03 0.        3.887e+04
130270  101  1.221e-03 1.057e-16 1.221e-03 0.        0.        2.993e+09
                                                                                                    < 2, 3>2yr
120260  102  0.        1.524e-17 0.        0.        3.362e+18 3.362e+18
120270 1022  1.448e-17 1.221e-03 0.        1.221e-03 0.        3.916e+04
130270  101  1.221e-03 1.047e-16 1.221e-03 0.        0.        3.016e+09
                                                                                                    < 2, 4>2yr
120260  102  0.        1.497e-17 0.        0.        3.362e+18 3.362e+18
120270 1022  1.431e-17 1.221e-03 0.        1.221e-03 0.        3.870e+04
130270  101  1.221e-03 1.021e-16 1.221e-03 0.        0.        2.980e+09
                                                                                                    < 2, 5>2yr
120260  102  0.        1.445e-17 0.        0.        3.362e+18 3.362e+18
120270 1022  1.387e-17 1.221e-03 0.        1.221e-03 0.        3.752e+04
130270  101  1.221e-03 9.792e-17 1.221e-03 0.        0.        2.890e+09
                                                                                                    < 2, 6>2yr
120260  102  0.        1.369e-17 0.        0.        3.362e+18 3.362e+18
120270 1022  1.319e-17 1.221e-03 0.        1.221e-03 0.        3.568e+04
130270  101  1.221e-03 9.234e-17 1.221e-03 0.        0.        2.747e+09
                                                                                                    < 2, 7>2yr
120260  102  0.        1.272e-17 0.        0.        3.362e+18 3.362e+18
120270 1022  1.228e-17 1.221e-03 0.        1.221e-03 0.        3.322e+04
130270  101  1.221e-03 8.644e-17 1.221e-03 0.        0.        2.558e+09
                                                                                                    < 2, 8>2yr
120260  102  0.        1.154e-17 0.        0.        3.362e+18 3.362e+18
120270 1022  1.116e-17 1.221e-03 0.        1.221e-03 0.        3.020e+04
130270  101  1.221e-03 7.734e-17 1.221e-03 0.        0.        2.325e+09
                                                                                                    < 2, 9>2yr
120260  102  0.        1.019e-17 0.        0.        3.362e+18 3.362e+18
120270 1022  9.856e-18 1.221e-03 0.        1.221e-03 0.        2.666e+04
130270  101  1.221e-03 6.814e-17 1.221e-03 0.        0.        2.053e+09
                                                                                                    < 2,10>2yr
120260  102  0.        8.674e-18 0.        0.        3.362e+18 3.362e+18
120270 1022  8.386e-18 1.221e-03 0.        1.221e-03 0.        2.268e+04
130270  101  1.221e-03 5.799e-17 1.221e-03 0.        0.        1.747e+09
                                                                                                    < 2,11>2yr
120260  102  0.        7.026e-18 0.        0.        3.362e+18 3.362e+18
120270 1022  6.776e-18 1.221e-03 0.        1.221e-03 0.        1.833e+04
130270  101  1.221e-03 4.702e-17 1.221e-03 0.        0.        1.411e+09
                                                                                                    < 2,12>2yr
```

1 stable isotopes for zone 2 interval 56

for an operating time of 2yr

nuclide	0	1 m	10 m	1 hr	6 hr	1 dy	1 wk	1 mo	1 yr	10 yr	100 yr	1000yr
120240	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19
120250	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18
120260	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18
130270	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21
140280	1.72e+13	1.72e+13	1.72e+13	1.72e+13	1.72e+13	1.72e+13	1.72e+13	1.72e+13	1.72e+13	1.72e+13	1.72e+13	1.72e+13
822040	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20
822060	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21
822070	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21
822080	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22
832090	1.36e+13	1.36e+13	1.36e+13	1.36e+13	1.36e+13	1.36e+13	1.36e+13	1.36e+13	1.36e+13	1.36e+13	1.36e+13	1.36e+13

1 the volume averaged densities for zone 2

for 2yr after shutdown

nuclide	0	1 m	10 m	1 hr	6 hr	1 dy	1 wk	1 mo	1 yr	10 yr	100 yr	1000yr
120240	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19	2.37e+19
120250	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18	3.05e+18
120260	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18	3.36e+18
130270	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21	2.91e+21
140280	1.29e+13	1.29e+13	1.29e+13	1.29e+13	1.29e+13	1.29e+13	1.29e+13	1.29e+13	1.29e+13	1.29e+13	1.29e+13	1.29e+13
822040	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20	4.39e+20
822060	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21	7.00e+21
822070	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21	6.71e+21
822080	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22	1.55e+22
832090	7.68e+12	7.68e+12	7.68e+12	7.68e+12	7.68e+12	7.68e+12	7.68e+12	7.68e+12	7.68e+12	7.68e+12	7.68e+12	7.68e+12

1 The atomic parts per million of isotopes for zone 2

and 2yr operation time

nuclide	0	1 m	10 m	1 hr	6 hr	1 dy	1 wk	1 mo	1 yr	10 yr	100 yr	1000yr
mg 24	7.27e+02	7.27e+02	7.27e+02	7.27e+02	7.27e+02	7.27e+02	7.27e+02	7.27e+02	7.27e+02	7.27e+02	7.27e+02	7.27e+02
mg 25	9.35e+01	9.35e+01	9.35e+01	9.35e+01	9.35e+01	9.35e+01	9.35e+01	9.35e+01	9.35e+01	9.35e+01	9.35e+01	9.35e+01
mg 26	1.03e+02	1.03e+02	1.03e+02	1.03e+02	1.03e+02	1.03e+02	1.03e+02	1.03e+02	1.03e+02	1.03e+02	1.03e+02	1.03e+02
al 27	8.92e+04	8.92e+04	8.92e+04	8.92e+04	8.92e+04	8.92e+04	8.92e+04	8.92e+04	8.92e+04	8.92e+04	8.92e+04	8.92e+04
si 28	3.96e-04	3.96e-04	3.96e-04	3.96e-04	3.96e-04	3.96e-04	3.96e-04	3.96e-04	3.96e-04	3.96e-04	3.96e-04	3.96e-04
pb 204	1.35e+04	1.35e+04	1.35e+04	1.35e+04	1.35e+04	1.35e+04	1.35e+04	1.35e+04	1.35e+04	1.35e+04	1.35e+04	1.35e+04
pb 206	2.15e+05	2.15e+05	2.15e+05	2.15e+05	2.15e+05	2.15e+05	2.15e+05	2.15e+05	2.15e+05	2.15e+05	2.15e+05	2.15e+05
pb 207	2.06e+05	2.06e+05	2.06e+05	2.06e+05	2.06e+05	2.06e+05	2.06e+05	2.06e+05	2.06e+05	2.06e+05	2.06e+05	2.06e+05
pb 208	4.76e+05	4.76e+05	4.76e+05	4.76e+05	4.76e+05	4.76e+05	4.76e+05	4.76e+05	4.76e+05	4.76e+05	4.76e+05	4.76e+05
bi 209	2.35e-04	2.35e-04	2.35e-04	2.35e-04	2.35e-04	2.35e-04	2.35e-04	2.35e-04	2.35e-04	2.35e-04	2.35e-04	2.35e-04

APPENDIX 2

DKR-STABLE LISTING

```
1  c     this code is designed to make the input for
2  c the dkr code easier
3  c it also checks for input errors that may have occured
4  c this is title
5         common /inte/ iiiia(17),izt,itt,not,nms,niss,iiiib(5890),
6        1 nucl(166),nucl1(118),iiiic(143804)
7         common /pointa/ ivolza,ivola,iwmata,icmpa,
8        1itend,mzma,lcala,miza,minta,jmata,nchna,lkuta,mzmta,
9        1kumzaa,kenda,nop,nas,niso,int,izm,ncmp,niiz,ign
10         common /rel/ dddd(250000)
11         dimension nzi(75), lcc(75), rri(75),rro(75)
12         dimension prep2(30),oprep(30),nprep(30,40),pprep(30,40)
13         dimension compos(25,40)
14         dimension aop(25),aas(25)
15         dimension dens(25,40)
16         dimension track(25),reflux(100),refttl(18)
17         dimension dum(25,40)
18         dimension cmp(50,25)
19         dimension abun(166),abun1(118)
20         dimension a(20), head(6)
21         integer dum,track
22         integer compos
23         real h1, h2, h3, h4, h5, h6
24         integer a1
25         call link(" unit4=dkrin,unit5=dkrinp,print59,
26        1unit59=terminal //")
27         data nucl/ 1001, 1002, 2003, 2004, 3006,3007,4009,5010,
28        15011,6012,6013,7014,7015,8016,8017,8018,9019,10020,10021,
29        110022,11023,12024,12025,12026,13027,14028,14029,14030,
30        115031,16032,16033,16034,16036,17035,17037,18036,18038,
31        118040,19039,19040,19041,20040,20042,20043,20044,20046,
32        120048,21045,22046,22047,22048,22049,22050,23050,23051,
33        124050,24052,24053,24054,25055,26054,26056,26057,26058,
34        127059,28058,28060,28061,28062,28064,29063,29065,30064,
35        130066,30067,30068,30070,31069,31071,32070,32072,32073,
36        132074,32076,33075,34074,34076,34077,34078,34080,
37        134082,35079,35081,36078,36080,36082,36083,36084,36086,
38        137085,37087,38084,38086,38087,38088,39089,40090,40091,
39        140092,40094,40096,41093,42092,42094,42095,42096,42097,
40        142098,42100,44096,44098,44099,44100,44101,44102,44104,
41        145103,46102,46104,46105,46106,46108,46110,47107,47109,
42        148106,48108,48110,48111,48112,48113,48114,48116,49113,
43        149115,50112,50114,50115,50116,50117,50118,50119,50120,
44        150122,50124,51121,51123,52120,52122,52123,52124,52125,
45        152126,52128,52130,53127/
46         data nucl1/54124,54126,54128,54129,54130,
47        154131,54132,54134,54136,55133,56130,56132,56134,56135,
48        156136,56137,56138,57138,57139,58136,58138,58140,58142,
49        159141,60142,60143,60144,60145,60146,62144,62147,62148,
50        162149,62150,62152,62154,63151,63153,64152,64154,64155,
51        164156,64157,64158,64160,65159,66156,66158,66160,66161,
52        166162,66163,66164,67165,68162,68164,68166,68167,68168,
53        168170,69169,70168,70170,70171,70172,70173,70174,70176,
54        171175,71176,72174,72176,72177,72178,72179,72180,73181,
55        174180,74182,74183,74184,74186,75185,75187,76184,76186,
56        176187,76188,76189,76190,76192,77191,77193,78190,78192,
57        178194,78195,78196,78198,79197,80196,80198,80199,80200,
58        180201,80202,80204,81203,81205,82204,82206,82207,82208,
```

```
      183209,90232,92234,92235,92238/
       data abun/99.985,.015,.0001,100.,7.52,92.48,100.,18.45,
      181.55,98.892,1.108,99.634,.366,99.759,.037,.204,100.,
      190.92,.257,8.82,100.,78.7,10.13,11.17,100.,92.21,4.7,3.09,
      1100.,95.,.76,4.22,.014,75.529,24.471,.337,.063,99.6,
      193.1,.012,6.88,96.97,.64,.145,2.06,.003,.185,100.,7.93,
      17.28,73.94,5.38,5.34,0.24,99.76,4.31,83.76,9.55,2.38,
      1100.,5.82,91.66,2.19,.33,100.,67.88,26.23,1.19,3.66,1.08,
      169.1,30.9,48.89,27.81,4.11,18.56,0.62,60.4,39.6,20.52,
      127.43,7.76,36.54,7.76,100.,0.87,9.02,7.58,23.52,49.82,
      19.19,50.54,49.46,0.354,2.27,11.56,11.55,56.90,17.37,
      172.15,27.85,.56,9.86,7.02,82.56,100.,51.46,11.23,17.11,
      117.40,2.8,100.,15.84,9.04,15.72,16.53,9.46,23.78,9.63,
      15.51,1.87,12.72,12.62,17.07,31.61,18.58,
      1100.,.96,10.97,22.23,27.33,26.71,11.81,51.35,
      148.65,1.215,.875,12.39,12.75,24.07,12.26,28.86,7.58,4.28,
      195.72,.96,.66,.35,14.3,7.61,24.03,8.58,32.85,4.72,5.94,
      157.25,42.75,.089,2.46,.87,4.61,6.99,18.71,31.79,34.49,100./
       data abun1/
      1.096,.09,1.919,26.44,4.08,21.18,26.89,10.44,8.87,100.,
      1.101,.097,2.42,6.59,7.81,11.32,71.66,.089,99.911,.193,.25,
      188.48,11.07,100.,27.11,12.17,23.85,8.30,17.22,3.09,14.97,
      111.24,13.83,7.44,26.72,22.71,47.82,52.18,.2,2.15,14.73,20.47,
      115.68,24.87,21.9,100.,.0524,.0902,2.294,18.88,25.53,24.97,
      128.18,100.,.136,1.56,33.41,22.94,27.07,14.88,100.,.135,
      13.03,14.31,21.82,16.13,31.84,12.73,97.40,2.6,.18,5.2,18.5,
      127.14,13.75,35.24,100.,.135,26.41,14.40,30.64,28.41,37.07,
      162.93,.018,1.59,1.64,13.3,16.1,26.4,41.0,37.3,62.7,.0127,
      1.78,32.9,33.8,25.3,7.21,100.,.146,10.02,16.84,23.13,13.22,
      129.80,6.85,29.50,70.50,1.48,23.6,22.6,52.3,100.,100.,
      1.0056,.7205,99.2739/
       iorg=1
c      do 3 i=1,166
c    3 write(6,108) nucl(i), abun(i)
c      do 4 i=1,118
c    4 write(6,108) nucl1(i), abun1(i)
c  108 format(1x,i6,f8.3)
c      stop
       data head/ 6hpriopt, 6hinital, 6htimcom, 6hreffop,
      16hheatng,6hradius/
       read(4,100) (a(i),i=1,20)
  100  format(20a4)
   32  write(5,100) (a(i),i=1,20)
       jk=0
    1  read(4,101) headc
       jk=jk+1
  101  format(a6)
       if (headc .eq.  head(1)) then
       call cint(len,lprt1,lprt2,lprt3,lprt4,lflx,b,v,w,x,y)
       if(len .ne.  5)stop 10
  105  format(6(i6))
       h1=1
       else if (headc .eq.  head(2)) then
       call cint(len,lid,lnk,lge,lfx,izm,int,iorg,a3,a4,a5)
       if(len .ne.  7)stop 10
       h2=1
       else if (headc .eq.  head(3)) then
       call cint(len,nop,nas,ncmp,ign,igg,a2,a3,a4,a5,a6)
```

```
      if(len .ne.  5) stop 10
      h3=1
      else if (headc .eq.  head(4)) then
      call cint(len,lfcf,fcf,a2,a3,a4,a5,a6,a7,a8,a9)
      if(len .ne.  2)stop 10
      h4=1
      else if (headc .eq.  head(5)) then
      call cint(len,wlld,htn,htg,htt,a2,a3,a4,a5,a6,a7)
      if(len .ne.  4) stop 10
      h5=1
       else if (headc .eq.  head(6)) then
      call cint(len,rrp,rrw,rrt,a2,a3,a4,a5,a6,a7,a8)
      if(len .ne.  3)stop 10
      h6=1
      end if
      jh=h1 +h2+h3+h4+h5+h6
      if( jh .ne.  jk) then
       write(59,104) headc
  104 format(a6, 'is not a proper heading')
      stop
      end if
      write(59,101) headc
      if (h2 .ne.  1 .or.  h3 .ne.  1 .or.  h4.ne.  1 .or.  h5 .ne.
     11 .or.  h6 .ne.  1 .or.  h1 .ne.  1) go to 1
      write(59,1000)
 1000 format(' this is the end of the sparse data part')
      nzit=0
      niiz=0
      do 2 i=1,izm
      call cint(lex,iz,nzi(i),lcc(i),rri(i),rro(i),a2,a3,a4,a5,a6)
      if (niiz .lt.  nzi(i))niiz = nzi(i)
      if (iz .ne.  i) stop 14
      if (lex .ne.  5) stop 13
      nzit=nzit+nzi(i)
    2 continue
      if (nzit .ne.  int) stop 7
      write(59,103)
  103 format(' this is the end of the zone part')
c now we enter the part where we must enter compositions
      do 19 io=1,ncmp
      call cint(lex,comp,a1,len,a2,a3,a4,a5,a6,a7,a8)
      if(lex .ne.  3) stop 12
c    if len = 0 it indicates a preprogrammed composition
      if (len .eq.  0) then
c  search through library in prep2 for composition
      do 8 i=1,100
      if (prep2(i) .eq.  comp) go to 9
    8 continue
c    this is the case where the comp.  was not found
      write(59,111) comp
  111 format(1x,a4,1x, 'there is no such mixture in the library')
      stop 8
      nnn=oprep(i)
      do 9 ii=1,nnn
      compos(a1,ii)=nprep(i,ii)
      dens(a1,ii)=pprep(i,ii)
    9 continue
      track(a1)=nnn
```

```
c  we have finished inputing the variables, now we see if
c  any should be neglected
      read(4,112) negl
  112 format(a4)
      if(negl .eq.  4hnegl) then
      read(4,105) numb
      do 12 nu=1,numb
      read(4,105) isot
      do 13 iii=1,nnn
      if(compos(a1,iii) .eq.  isot) go to 14
   13 continue
      write(59,113) isot
  113 format(' we havent found that isotope in our list',i6)
      go to 12
   14 continue
      dens(a1,iii)=0.
   12 continue
      end if
      else
      ll=0
      do 15 ij=1,len
      call cint(lex,int2,densit,a2,a3,a4,a5,a6,a7,a8,a9)
      if(lex .ne.  2) stop 11
      if (int2 .ge.  1000) then
      int1=int2 / 1000
      int2=int2-1000*int1
      else
      int1=int2
      int2=0
      end if
      if (int2 .eq.  0) then
      if(int1 .le.  53) then
      do 16 ik=1,166
      icom=nucl(ik)/1000
      if (icom .eq.  int1) then
      ll=ll+1
      compos(a1,ll)=nucl(ik)
      dens(a1,ll)=abun(ik)*densit/100.
      else
      if(icom .gt.  int1) go to 18
      end if
   16 continue
      else
      do 17 ik=1,118
      icom=nucl1(ik)/1000
      if (icom .eq.  int1) then
      ll=ll+1
      compos(a1,ll)=nucl1(ik)
      dens(a1,ll)=abun1(ik)*densit/100.
      else
      if(icom .gt.  int1) go to 18
      end if
   17 continue
      end if
   18 continue
      else
      ll=ll+1
      int4=int1*1000+int2
```

```
      compos(a1,11)=int4
      dens(a1,11)=densit
      end if
   15 continue
      track(a1)=11
      end if
   19 continue
      do 21 iz=1,izm
      read(4,117)(cmp(iz,ncm),ncm=1,ncmp)
  117 format(12f6.3)
   21 continue
      if(iorg .eq.  1)then
      newcom=10206
      newden=newcom+300
      lista=newden+300*izm
      call org(dens,compos,track,cmp,izm,niso,ncmp,dddd(newcom),
     1dddd(newden),dddd(lista))
      else
      ncou=0
      nco=0
      do 37 j=1,ncmp
      do 37 i=1,track(j)
      ncou=ncou+1
      do 30 jj=j,ncmp
      do 30 ii=1,track(j)
      if ((compos(j,i) .ne.  compos(jj,ii)) .or.
     1(jj .eq.  j .and.  ii .eq.  i)) go to 30
      write(59,122) compos(j,i)
  122 format(' We have a repetition of isotope ',1x,i6,/,
     1'.    The lower density one will be neglected')
      if(dens(jj,ii) .ge.  dens(j,i)) then
      trackn=track(j)-1
      do 36 iiiaa=i,trackn
      ip=iiiaa+1
      compos(j,iiiaa)=compos(j,ip)
   36 dens(j,iiiaa)=dens(j,ip)
      track(j)=track(j)-1
      nco=nco+1
      go to 37
      else
      trackn=track(jj)-1
      do 38 iiiaa=ii,trackn
      ip=iiiaa+1
      compos(jj,iiiaa)=compos(jj,ip)
   38 dens(jj,iiiaa)=dens(jj,ip)
      track(jj)=track(jj)-1
      nco=nco+1
      end if
   30 continue
   37 continue
      niso=ncou - nco
      end if
      write(5,150) lid,lnk,lge,lfx,izm,int,nop,nas,niso,ncmp,ign,igg
  150 format(12i6)
      write(5,150) lprt1,lprt2,lprt3,lprt4,lflx,lfcf
      write(5,156) wlld,htn,htg,htt,fcf
  156 format(4f12.3,e12.3)
      write(5,151) rrp,rrw,rrt
```

```
  151 format(6f12.5)
      write(5,152) (i,nzi(i),lcc(i),rri(i),rro(i),i=1,izm)
  152 format(3i6,6x,f9.2,3x,f9.2)
c   this is the composition prcentage part
c    we use the same format as in previous input edition
      do 22 id=1,ncmp
      write(5,153) (cmp(ie,id),ie=1,izm)
   22 continue
  153 format(12f6.3)
      do 31 j=1,ncmp
      do 31 i=1,track(j)
      dum(j,i) = 1
   31 continue
      do 24 j=1,ncmp
      do 23 i=1,track(j)
      do 25 jj=1,ncmp
      do 26 ii=1,track(jj)
      if (compos(j,i) .gt.compos(jj,ii))
     1dum(j,i)=dum(j,i)+1
   26 continue
   25 continue
   23 continue
   24 continue
      do 27 n=1,niso
      do 28 j=1,ncmp
      do 29 k=1,track(j)
      if (n .eq.  dum(j,k)) then
      inp=4
      if(dens(j,k) .ge.1e+20) inp=3
      if (dens(j,k) .ge.  1e+21) inp=2
      if (dens(j,k) .ge.  1e+22) inp=1
      write(5,155) j,compos(j,k),inp,dens(j,k)
  155 format(3i6,e12.3)
      go to 27
      end if
   29 continue
   28 continue
   27 continue
c  we must now worry about the times(op.  and as.)
      if(nop .ne.  0) then
      do 34 it=1,nop
      call cint(lex,aopp,fopp,a2,a3,a4,a5,a6,a7,a8,a9)
      if (lex .ne.  2) stop 15
   34 write(5,118) aopp,fopp
      else
      nop=9
      end if
      if (nas .ne.  0) then
      do 35 it=1,nas
      call cint(lex,aass,fass,a2,a3,a4,a5,a6,a7,a8,a9)
      if (lex .ne.  2) stop 16
   35 write(5,118) aass,fass
  118 format(a6,e12.3)
      else
      nas=12
      end if
      if(lflx .eq.  2) then
      read(4,201)(refttl(i),i=1,18)
```

```
      write(5,201)(refttl(i),i=1,18)
      read(4,202) (reflux(j),j=1,ign)
      write(5,202)(reflux(j),j=1,ign)
  201 format(18a4)
  202 format(6e12.3)
      end if
      call flux1(int,ign)
c now we spit out the information after checking for errors
c the input has passed the test
      write(59,120)
  120 format('so far it has gone ok')
      call init
      izt=izm
      itt=int
      not=nop
      nms=nas
      niss=niso
      call main1
      end
c
c
c   subroutine int
c     this subroutine reads data in a format free mode
      subroutine cint(len,in1,in2,in3,in4,in5,in6,in7,in8,in9,in10)
      dimension a(10),ap(10),i(10),in(10)
      do 20 j=1,10
   20 a(j)='          '
      read(4,126) (a(l),l=1,10)
  126 format(10a8)
      call getnumb(in,i,len,a,80,10)
      in1=in(1)
      in2=in(2)
      in3=in(3)
      in4=in(4)
      in5=in(5)
       in6=in(6)
      in7=in(7)
      in8=in(8)
      in9=in(9)
      in10=in(10)
      return
      end
c
c
c this subroutine inputs the flux part
      subroutine flux1(int,ign)
      dimension f1(100)
      dimension title(18)
      read(4,100) title
      write(5,100) title
      do 110 n=1,int
      read(4,101) aint,nint
      write(5,101) aint,nint
      if(n .eq.  1) nint1=nint
      if(n .eq.  1)write(59,119) nint1
  119 format(/5x,"first interval =",i3)
      read(4,102)(f1(i),i=1,ign)
      write(5,102)(f1(i),i=1,ign)
```

```
  110 continue
      if(nint .ne.  int)write(59,120)int,nint
  120 format(5x,i3,"intervals required"/5x,"warning:  last interval="
     1,i3/)
      inrd=nint-nint1+1
      if (inrd .ne.  int) stop 50
  100 format(18a4)
  101 format(a3,i3)
  102 format(6e12.3)
      return
      end
c
c
c   this subroutine initializes the array pointers for later use
      subroutine init
      common /rel/ dddd(250000)
      common /inte/ iiii(150000)
      common /pointa/ivolza,ivola,iwmata,icmpa,
     1itend,mzma,lcala,miza,minta,jmata,nchna,lkuta,mzmta,
     1kumzaa,kenda,nop,nas,nnc,int,izm,ncmp,niiz,ign
      ivolza=5149
      ivola=ivolza+nnc
      iwmata=ivola+int
      icmpa=iwmata+nnc*izm
      itend=icmpa+ncmp*izm
      mzma=6197
      lcala=mzma+izm
      miza=lcala+izm
      minta=miza+izm
      jmata=minta+izm
      nchna=jmata+nnc
      lkuta=nchna+nnc
      mzmta=lkuta+nnc
      kumzaa=mzmta+izm*nnc
      kenda=kumzaa+150
      return
      end
      subroutine org(dens,compos,track,cmp,izm,iii,ig,
     1newcom,newden,list)
c      this subroutine organizes the nuclides into various
c      composition groups
      dimension dens(25,40),compos(25,40),track(25),cmp(50,25)
      common /rel/ dddd(250000)
      dimension comnew(1),dennew(300,1),list(1)
      integer comnew,track,compos
      ii=0
      do 1 i=1,ig
      itracx=track(i)
      do 32 j=1,itracx
      ii=ii+1
      comnew(ii)=compos(i,j)
      do 33 ll=1,izm
      dennew(ii,ll)=dens(i,j)*cmp(ll,i)
   33 continue
   32 continue
    1 continue
c     check for duplicity
      ij=0
```

```
 34 ij=ij+1
    ik=ij
  4 ik=ik+1
    if(comnew(ik).  eq.  comnew(ij))then
    flag=1
    write(59,100)comnew(ij)
100 format(' some reorganization is needed since ',i6,
   1' is repeated')
    do 3 ll=1,izm
  3 dennew(ij,ll)=dennew(ij ll)+dennew(ik,ll)
    do 2 il=ik,ii
    im=il+1
    comnew(il)=comnew(im)
    do 2 ll=1,izm
    dennew(il,ll)=dennew(im,ll)
  2 continue
    ii=ii-1
    ik=ik-1
    end if
    if(ik .le.  ii) go to 4
    if (ij .lt.  ii) go to 34
    iii=ii
    if (flag .eq.  0) return
c
c  we have now checked for duplicity and removed it
c    we have also checked to make sure changes are needed
c
c    we now see how many compositions are necessary
c
c     our tolerance is 1   495  c
    ig=0
 10 ia=1
    ig=ig+1
    if (ii .eq.  1) go to 20
    if (izm .eq.  1) then
    ig=1
    track(1)=ii
    cmp(1,1)=1.00
    do 25 je=1,ii
    compos(1,je)=comnew(je)
    dens(1,je)=dennew(je,1)
 25 continue
    ig=1
    return
    end if
    id=1
    iap=ia+1
    do 5 ib=iap,ii
    izmn=izm-1
    do 6 ic=1,izm
  6 if(dennew(ia,ic) .ne.  0.)  go to 26
    do 28 jg=1,izm
c     this is the case of the compositions being zero
 28 if (dennew(ib,jg) .ne.  0.)  go to 5
    go to 30
 26 if(dennew(ib,ic) .eq.  0.)  go to 5
    do 29 jf=1,izm
    if(dennew(ia,jf).eq.0.0.and.dennew(ib,jf) .eq.  0.)  go to 29
```

```
524         if((dennew(ia,jf) .eq.  0.  .and.  dennew(ib,jf) .ne.0.)  .or.
525        1(dennew(ib,jf).eq.  0.  .and.  dennew(ia,jf) .ne.  0.))  go to 5
526         comden=abs((dennew(ia,ic)/dennew(ib,ic))/(dennew(ia,jf)/
527        1dennew(ib,jf))-1.)-.01
528         if (comden .ge.  0) go to 5
529      29 continue
530   c
531   c    check if it compares favorably and if so list it
532   c
533      30 id=id+1
534         list(id)=ib
535       5 continue
536         list(1)=1
537         track(ig)=id
538         do 14 ih=1,id
539         iu=list(ih)
540      14 compos(ig,ih)=comnew(iu)
541         do 13 iq=1,izm
542      13 if (dennew(iu,iq) .ne.  0.)  go to 12
543   c
544   c    we will use this nuclide for comparison
545   c
546         do 16 is=1,id
547      16 dens(ig,is)=0.
548         do 15 ir=1,izm
549      15 cmp(ir,ig)=0.
550         go to 19
551   c
552   c      we use dennew(iu,it) as a comparison
553   c
554      12 do 17 it=1,id
555         iv=list(it)
556      17 dens(ig,it)=dennew(iv,iq)
557         do 18 iw=1,izm
558      18 cmp(iw,ig)=dennew(iu,iw)/dennew(iu,iq)
559   c
560   c     we now remove the isotopes consiered
561   c
562      19 if(id .eq.  ii) return
563         do 8 ie=1,id
564         ib=list(ie)+1-ie
565         do 9 if=ib,ii
566         ifp=if+1
567         comnew(if)=comnew(ifp)
568         do 9 ll=1,izm
569       9 dennew(if,ll)=dennew(ifp,ll)
570         ii=ii-1
571       8 continue
572         if(ii .eq.  0) then
573         return
574         end if
575         go to 10
576   c
577   c      we now consider the case of only one isotope left
578   c
579      20 compos(ig,1)=comnew(1)
580         do 21 ja=1,izm
581         track(ig)=1
```

```
582         if(dennew(1,ja) .ne.  0.)  go to 23
583      21 continue
584   c     this is the case of zero density throughout
585         dens(ig,1)=0.
586         do 22 jb=1,izm
587      22 cmp(jb,ig)=0.
588         ncmp=ig
589         return
590   c      this is the case of one isotope
591      23 dens(ig,1)=dennew(1,ja)
592         do 24 jc=1,izm
593      24 cmp(jc,ig)=dennew(1,jc)/dennew(1,ja)
594         ncmp=ig
595         return
596         end
597         subroutine main1
598   c dkr cray version 11-29-78 ga-san diego, ca.
599   c        *** dkr program listing   ***
600   c          ***july 10 by tak yun sung      ***
601   c
602   c   -              main program provides the data storage
603   c   -                       and initiates the program
604   c
605         common / inte/ ii(150000)
606         common /rel/ dd(250000)
607         common /pointa/ ivolza,ivola,iwmata,icmpa,itend,
608        1mzma,lcala,miza,minta,jmata,nchna,lkuta,mzmta,kumzaa,
609        2kenda,nop,nas,nnc,int,izm,ncmp,niiz,ign
610         call dropfile(4h+dkr)
611   c
612   c   -    n5 and n6 are the standard input and output unit
613   c   -    nt1   punch unit
614   c   -    nt2   constructed chain file unit
615   c   -    nt3   cross section table unit
616   c   -    nt7   nuclide no.  density storage file
617   c   -    nt8   decay gamma source file unit
618   c   -    nt9   master file unit
619   c
620         ii(1) = 5
621         ii(2) = 6
622         ii(3)=7
623         ii(4)= 2
624         ii(5)= 3
625         ii(6)=17
626         ii(7)=18
627         ii(8)= 9
628   c
629         call create(6,6hdkrout,3,900000)
630         call create(7,4hdkr7,3,900000)
631         call create(2,4hdkr2,2,900000)
632         call create(3,4hdkr3,1,900000)
633         call create(17,5hdkr17,1,900000)
634         call create(18,5hdkr18,1,900000)
635         call open(9,4hdkr9,0,len)
636         call size
637         call chime(1)
638   c
639   c   -    manipulation of data and
```

```
c        construction of chains
         call drive
         call chime(2)
c
c        construction of chains
         lkpa=kenda+20
         ktpa=lkpa+10
         jj1a=ktpa+10
         jj2a=jj1a+10
         nbigmaa=jj2a+10
         ibtka=itend+120
         ibtaa=ibtka+10
         isgka=ibtaa+10
         isiga=isgka+10
         icc1a=isiga+10
         icc2a=icc1a+10
         icc3a=icc2a+10
         icc4a=icc3a+10
         ibigma=icc4a+10
         call pkp(ii(nbigmaa),ii(jj2a),ii(jj1a),ii(ktpa),ii(lkpa),
        1ii(jmata),ii(nchna),dd(ibigma),dd(icc4a),dd(icc3a),dd(icc2a),
        2dd(icc1a),dd(isiga),dd(isgka),dd(ibtaa),dd(ibtka))
         call chime(2)
         if(ii(12).eq.0) call exit(1)
c
c   -    calculation of activities
         kxzaa=kenda
         kkzaa=kxzaa+18
         jrnka=kkzaa+150
         ipra=jrnka+66
         kska=ipra+66
         jk1a=kska+66
         jrxa=jk1a+600
         lcla=jrxa+600
         icb1a=100+itend
         icb2a=icb1a+600
         ics1a=icb2a+600
         ics2a=ics1a+600
         iwya=ics2a+600
         iyaa=iwya+nas
         iya=iyaa+9
         icya=iya+nop*9
         isgpa=icya+150*nas
         inda=kenda+2168
         itaba=isgpa+(18*19+150*nas)*niiz
         icwya=itend+2509+nas*151+nop*9+18*19*niiz
         call xcute(ii(mzma),ii(lcala),ii(miza),ii(minta),ii(jmata),
        1ii(nchna),ii(lkuta),ii(mzmta),ii(kumzaa),dd(ivolza),dd(ivola),
        2dd(iwmata),dd(icmpa),ii(kxzaa),ii(kska),ii(kkzaa),ii(jrnka),
        3ii(ipra),ii(lcla),ii(jk1a),ii(jrxa),dd(icya),dd(icb1a),
        4dd(icb2a),dd(ics1a),dd(ics2a),dd(iwya),dd(iyaa),dd(iya),
        5dd(isgpa),dd(icwya),ii(inda),dd(itaba))
c
c   -    summarize the results
         call chime(2)
c
         call exit(1)
         end
```

```
      subroutine ashut(m,ns,s1,s2,x,wy,ts)
c   -    calculate after shutdown radioactivity and bhp etc
      dimension s1(m),s2(m),x(m),wy(ns),ts(ns)
      dimension a(9,9)
c     double precision a(9,9),dxx,sjt,dwy
c
      do 10 ik=1,m
      do 10 ij=1,m
c     a(ik,ij)=0.0d0
      a(ik,ij)=0.0
   10 continue
c
        a(1,1)=x(1)
      if(m.eq.1) go to 55
      do 120 k=2,m
           k1=k-1
      do  40 j=1,k1
      dnom=abs(s2(k)-s2(j))/(s2(k)+s2(j))
      if(dnom.lt.1.0e-4) s2(k)=s2(k)*1.009
      dxx=s1(k)/(s2(k)-s2(j))
      a(k,j)=a(k1,j)*dxx
   40 continue
c
        a(k,k)=x(k)
      do  50 j=1,k1
      dnom=abs(s2(k)-s2(j))/(s2(k)+s2(j))
      if(dnom.lt.1.0e-4) s2(k)=s2(k)*1.009
      dxx=s1(k)/(s2(k)-s2(j))
      a(k,k)=a(k,k)-a(k1,j)*dxx
   50 continue
  120 continue
c
   55 continue
          do 110 i=1,ns
                t=ts(i)
c     dwy=0.0d0
      dwy=0.0
      do  60 j=1,m
      sjt=s2(j)*t
      dwy=a(m,j)*exp(-sjt)+dwy
   60 continue
      wy(i)=dwy
  110 continue
c
      return
      end
      subroutine branch(kq,jk,nkt,ix,btk,sgk,bta,sig,lkp,ktp,mk,mx)
c   -    branch collects the branches of chain
      common /inte/ iiiia(22),jrmax,kkxn(257),kdx(256,3),kdl(256,19),
     1iiiic(1)
      common /rel/ space(134),dcy(256,19),spaceb(1)
      dimension btk(mk),bta(mx),sgk(mk),sig(mx),lkp(mx),ktp(mx)
c   -    initiation of btk and sgk
                btk(kq)=0.
                sgk(kq)=0.
      do 110 ik=1,mx
                sig(ik)=0.
  110           bta(ik)=0.
```

```
c
      ix=0
c
      do 210 i=1,nkt
      lk2=kdl(jk,i)/100
      jj =mod(kdl(jk,i),100)
      d1 =dcy(jk,i)
      if(jj.eq.1.or.jj.eq.21) go to 22
      if(jj.gt.20) go to 13
      ix=ix+1
      lkp(ix)=lk2
      ktp(ix)=jj
      sig(ix)=d1
      go to 210
c
   13 k20=jj-20
      if(ix.eq.0) go to 19
c  -    test the redundancy
      do  30 j=1,ix
      if(lk2.eq.lkp(j)) go to 16
   30 continue
c
   19 continue
      ix=ix+1
      lkp(ix)=lk2
      ktp(ix)=jj
      sig(ix)=d1
      bta(ix)=d1
      go to 210
c
   16 ktp(j)=jj
      sig(j)=sig(j)+d1
      bta(j)=d1
      go to 210
   22 if(jj.eq.1)  sgk(kq)=d1
      if(jj.eq.21) btk(kq)=d1
  210 continue
c
      return
      end
      subroutine chime(n)
c  -    time obtains and writes cpu time ...  univac 1110 ...  t.  y.  sung
      common /inte/ n5, n6, nt1, nt2, nt3, nt7, nt8, nt9,iiii(1)
      now=' now '
      lll=1
c     if(lll.eq.1) go to 99
c     write(n6,601)
c     go to (1,2), n
c   1 x=0.0
c     call urtims(x)
c   2 t=urtimg(now)
c 601 format(1h0,'time record' )
c
   99 return
      end
      subroutine clca(lg,aa,rw,rt)
c  -    calculate the 1st wall area
      pi = 3.14159
```

```
      pi4= 1.25664e+01
      piq= 9.86960
c
      go to (1,2,3,4), lg
c
c   -    slab
    1 aa = 1.e-4
      go to 99
c   -    cylinder
    2 aa = 0.0002*pi*rw
      go to 99
c   -    sphere
    3 aa = 0.0001*pi4*rw**2
      go to 99
c   -    torus
    4 aa = 0.0004*piq*rw*rt
c
   99 return
      end
      subroutine clcno(m,s1,s2,x,y,tt,n)
c   -    calculate no density of nuclide in a chain
      parameter (mk=9)
      dimension s1(m),s2(m),x(m),y(n,mk),tt(n)
c     dimension a(9,9)
      double precision a(9,9),dxx,sjt,at1,at2,dw
      do 20 j=1,m
      do 20 k=j,m
   20 if(abs((s2(k)-s2(j))/(s2(k)+s2(j))).lt.1.e-4)s2(k)=s2(k)*(1.+
     1.009*k)
      do 10 ik=1,m
      do 10 ij=1,m
      a(ik,ij)=0.0d0
c     a(ik,ij)=0.0
   10 continue
c
      if(m.eq.2.and.s1(1).gt.1.0e-18) go to 333
c
c   -    non-loop chain case
           m1=m-1
        a(1,1)=x(1)
      do 120 k=2,m
           k1=k-1
      do  40 j=1,k1
         dxx=s1(k)/(s2(k)-s2(j))
      a(k,j)=a(k1,j)*dxx
   40 continue
        a(k,k)=x(k)
      do 50 j=1,k1
        dxx=s1(k)/(s2(k)-s2(j))
      a(k,k)=a(k,k)-a(k1,j)*dxx
   50 continue
  120 continue
c
      do 70 it=1,n
           t=tt(it)
      do  70 k=1,m
      dw=0.0d0
c     dw=0.0
```

```
872         do  60 j=1,k
873              sjt=s2(j)*t
874              dw=a(k,j)*dexp(-sjt)+dw
875      60 continue
876             y(it,k)=sngl(dw)
877      70 continue
878         go to 99
879   c
880   c    * * * 2nd order loop case
881     333    a11=s2(1)
882            a22=s2(2)
883            a21=s1(1)
884            a12=s1(2)
885         do 80 it=1,n
886                t=tt(it)
887         at1=a11*t
888         at2=a22*t
889   c
890         lx=1
891         dw=0.0d0
892   c     dw=0.0
893         go to (1,2), lx
894   c
895       1 y(it,1)=x(1)*dexp(-at1)
896         elmt=abs(a22-a11)/(a11+a22)
897         if(elmt.gt.1.0e-4) a22=a22*1.005
898         dxx=x(1)*a12/(a22-a11)
899         dw=dxx*(dexp(-at1)-dexp(-at2))
900         y(it,2)=sngl(dw)
901         go to 80
902   c
903       2 tum1=(a11+a22)/2.
904         tum2=sqrt((a11+a22)**2/4.-a11*a22+a12*a21)
905         root1=-tum1+tum2
906         root2=-tum1-tum2
907         if(abs(root1).lt.1.0e-15) root1=(a12*a21-a11*a22)/(a11+a22)
908         dlamd=2.*tum2
909         exlm1=exp(root1*t)
910         exlm2=exp(root2*t)
911   c   -
912         y(it,1)=x(1)*((root1+a22)*exlm1-(root2+a22)*exlm2)/dlamd
913         y(it,2)=x(1)*a12*(exlm1-exlm2)/dlamd
914   c   -    end of loop
915   c
916      80 continue
917      99 return
918         end
919         subroutine clcv(lg,nt,ri,ro,rt,vv)
920   c   -    calculates the volumes of intervals and zone
921         dimension vv(nt)
922         pi = 3.14159
923         piq= 9.86960
924         pi4=12.5664
925         piv= 4.18879
926         xo=ri
927   c
928         do 10 j=1,nt
929         xi=xo
```

```
930         aj=j
931         bj=nt-j
932         an=nt
933         xo=(bj*ri + aj*ro)/an
934         dx=xo-xi
935   c
936         go to (1,2,3,4), lg
937   c   -    slab
938       1 vv(j) = dx
939         go to 10
940   c   -    cylinder
941       2 vv(j) = pi*dx*(xo+xi)
942         go to 10
943   c   -    sphere
944       3 vv(j) = piv*dx*(xo**2+xo*xi+xi**2)
945         go to 10
946   c   -    torus
947       4 vv(j) = 2.*piq*dx*(xo+xi)*rt
948         xi=xo
949      10 continue
950   c
951         return
952         end
953         subroutine colrad(liz,nzi,kint,mzm,lcal,miz,mint,jmat,
954        1nchn,lkut,mzmt,kumza,volz,vol,wmat,cmp,cwy,cy)
955   c   -    colrad collects the radioactivity and prints out
956   c
957   c
958         common / pointa/ ivolza,ivola,iwmata,icmpa,
959        1itend,mzma,lcala,miza,minta,jmata,nchna,lkuta,mzmta,kumzaa,
960        1kenda,nop,nas,nnc,int,izm,ncmp,niiz,ign
961         common /inte/ n5,n6,nt1,nt2,nt3,nt7,nt8,nt9,lid,lge,lfx,
962        1lnk,lprt1,lprt2,lprt3,lprt4,lflx,iizm,iint,inop,inas,innc,jrmax,
963        1kkxn(256),jsmax,kdx(256,3),
964        1kdl(256,19),kks(139795)
965         common /rel/ wlld,area,pwr,fcf,htn,htg,htt,rrp,rrw,rrt,
966        1ttl(18),refttl(18),bop(9),top(9),bas(12),tas(12),
967        1reflux(46),dcy(256,19),anuk(150),ddd(243677)
968         dimension cy(150,1),kkza(150),cwy(niiz,150,nas)
969         dimension mzm(izm),lcal(izm),miz(izm),mint(izm),jmat(nnc),
970        1nchn(nnc),lkut(nnc),mzmt(izm,nnc),kumza(150),npnuk(150)
971         dimension volz(nnc),vol(int),wmat(nnc,izm),cmp(ncmp,izm)
972   c   -    nt7 :  activity file
973   c   -    nt8 :  gamma source file
974         mggnew=21
975         mgx=96
976         do 1000 jop=1,nop
977   c
978         do 300 ij=1,nzi
979         do 300 ik=1,nas
980         do 300 il=1,150
981         cwy(ij,il,ik)=0.
982       300 continue
983   c
984         rewind nt7
985         kmax=0
986        11 read(nt7) jr,jt,kx
987         if(iostatus(nt7,istat).ne.0)go to 999
```

```
      read(nt7) (kkza(kd),(cy(kd,ld),ld=1,nas),kd=1,kx)
c          write(n6,189) liz,jr,jt,kx,(kkza(m),m=1,kx)
c 189 format(' z,i, jt,kx<', 12i6)
      if(jt.ne.jop) go to 11
c
      do 20 k=1,kx
      if(kmax.eq.0) go to 555
      do 25 kk=1,kmax
   25 if(kkza(k).eq.kumza(kk)) go to 556
c
  555 kmax=kmax+1
      kumza(kmax)=kkza(k)
       do 520 l=1,nas
  520 cwy(jr,kmax,l)=cy(k,l)
       go to 20
  556 continue
       do 540 l=1,nas
  540 cwy(jr,kk,l)=cwy(jr,kk,l)+cy(k,l)
   20 continue
c
       go to 11
c
  999 kxm=kmax
       do 256 ia=1,nzi
       ir=nzi+kint
       call wrout(liz,jop,kxm,cwy,bop,nop,nas,ia,niiz,kumza,npnuk,
     1kint,bas,lprt4)
  256 continue
  257 call avge(kumza,npnuk,cwy,bop,vol,volz,nzi,nas,nop,izm,kxm,
     1kint,jop,liz,int,niiz,bas,wmat,nnc)
 1000 continue
       return
       end
       subroutine drive
c   -              drive controls the program
c   -                  data part and chain construction
       common/inte/ii(150000)
       common/rel/dd(250000)
       common/pointa/ivolza,ivola,iwmata,icmpa,itend,mzma,
     1lcala,miza,minta,jmata,nchna,lkuta,mzmta,kumzaa,kenda,
     2nop,nas,nnc,int,izm,ncmp,niiz,ign
       ivolia=itend+100
       ixmata=ivolia+niiz
       iaja=ixmata+nnc
       itita=itend+120
       if1a=18+itita
       iphia=132+ign+if1a
       iadxa=itend+120
       iidxa=iadxa+30
       idxa=iidxa+30
       ikpda=idxa+30
       ilfsa=ikpda+30
       ix1a=ilfsa+30
       isigpa=iphia+int*ign
c   -    read input data
      call input(ii(mzma),ii(lcala),ii(miza),ii(minta),ii(jmata),
     1ii(nchna),ii(lkuta),ii(mzmta),ii(kumza),dd(ivolia),
     2dd(ixmata),dd(iaja),dd(ivolza),dd(ivola),dd(iwmata),
```

```
      3dd(icmpa))
       if(ii(12).eq.2.or.ii(12).eq.4) go to 99
c   -    read and rearrange the flux
       call flux(dd(itita),dd(if1a),dd(iphia))
       call chime(2)
c   -    generate index and xn table file from master file
       call indf(dd(iadxa),dd(iidxa),dd(idxa),dd(ikpda),dd(ilfsa),
      1dd(ix1a),dd(iphia),dd(isigpa))
c
   99 return
      end
      subroutine fido(j5,j6,x1,nn,m)
c   --   fido reads anisn format
      dimension in(6),k(6),v(6)
      dimension x1(nn)
      data lr,lt,lpl,lmi/1hr,1ht,1h+,1h-/
c
      jj=nn
      j=0
c
    1 continue
      read(j5,100) (in(i),k(i),v(i),i=1,6),m
  100 format(6(i2,a1,f9.0),i5)
c
      do 2 i=1,6
      if(k(i).eq.lr)go to 7
      if(k(i).eq.lt)go to 9
      if(j.eq.jj) return
      go to 14
c   -    termination by  t
    9 if(j.eq.jj) return
      write(j6,200)j
  200 format(1x,13h error  141 *,i7,14h  entries read)
c ///
      call exit(1)
    7 l=in(i)
      do 18 ii=1,l
      j=j+1
   18 x1(j)=v(i)
      go to 2
c   -    regular input reading
c ************************************
   14 j=j+1
      x1(j)=v(i)
    2 continue
c   -    termination by reading all the data
      if(j.eq.jj) return
      go to 1
      end
      subroutine flux(title,f1,phi)
c   -    flux reads the scalar flux
      common/pointa/ ivolza,ivola,iwmata,icmpa,
     1itend,mzma,lcala,miza,minta,jmata,nchna,lkuta,mzmta,
     1kumzaa,kenda,nop,nas,nnc,int,izm,ncmp,niiz,ign
      common /inte/ n5,n6,nt1,nt2,nt3,nt7,nt8,nt9,lid,lge,lfx,
     1lnk,lprt1,lprt2,lprt3,lprt4,lflx,iizm,iint,inop,inas,innc,
     1jrmax,kkxn(256),jsmax,kdx(256,3),
     1kdl(256,19),kks(139795)
```

```
      common /rel/ wlld,area,pwr,fcf,htn,htg,htt,rrp,rrw,rrt,
     1 ttl(18),refttl(18),bop(9),top(9),bas(12),tas(12),
     1reflux(46),dcy(256,19),anuk(150),dddd(243677)
      dimension title(18),f1(int),phi(int,ign)
      mng=ign
c   -    nt8 is the file of  flux data
      mmng=mng
      nin= 5
      non=nt8
c
      go to (1,2),lfx
c
c   -    read flux from data file
    1 continue
      read(nin,100) title
      do 110 n=1,int
      read(nin,101) aint,nint
      if(n.eq.1)nint1=nint
      if(n.eq.1)write(n6,119)nint1
  119 format(/5x,"first interval =",i3)
      read(nin,102) (f1(i),i=1,mng)
      do 110 i=1,mng
      phi(n,i)=f1(i)
  110 continue
      if(nint.ne.int)write(n6,120)int,nint
  120 format(5x,i3," intervals required"/5x,"warning:  last interval="
     1,i3/)
      inrd=nint-nint1+1
      if(inrd.ne.int)call exit(1)
c     if(nint.ne.int) call exit(1) 131
      go to 99
c
c   -    read the flux in anisn scalar flux format
    2 continue
      read(nin,100)title
      read(nin,105) signal
      if(signal.ne.3h 3*)call exit(1)
c     if(signal.ne.3h 3*) call exit(1) 132
      do 10 i=1,mng
      call fido(nin,non,f1,int,m)
      do 10 j=1,int
      phi(j,i)=f1(j)
   10 continue
c
   99 continue
c   -    printout the flux
      write(n6,107)
      write(n6,108) int,int,mmng
      write(n6,109) title
c
  100 format(18a4)
  101 format(a3,i3)
  102 format(6e12.4)
  105 format(a3)
  107 format(/,20x,'flux reading',/)
  108 format(22x,i3,' intervals read from flux (',i3,',',i3, ')')
  109 format( 10x,18a4,///)
  810 format(3hint,i3,/,(6e12.4))
```

```
c
      return
      end
      subroutine indf(adx,idx,dx,kpd,lfs,x1,phi,sigp)
c     -    indf reads the master file and
c     -         generates the concise index file
      common /pointa/ ivolza,ivola,iwmata,icmpa,
     1itend,mzma,lcala,miza,minta,jmata,nchna,lkuta,mzmta,
     1kumzaa,kenda,nop,nas,nnc,int,izm,ncmp,niiz,ign
      common /inte/ n5,n6,nt1,nt2,nt3,nt7,nt8,nt9,lid,lge,lfx,
     1lnk,lprt1,lprt2,lprt3,lprt4,lflx,iizm,iint,inop,inas,innc,
     1jrmax,kkxn(256),jsmax,kdx(256,3),
     1kdl(256,19),kks(139795)
      common /rel/ wlld,area,pwr,fcf,htn,htg,htt,rrp,rrw,rrt,
     1ttl(18),refttl(18),bop(9),top(9),bas(12),tas(12),
     1reflux(46),dcy(256,19),anuk(150),dddd(243667)
      dimension adx(29),idx(29),dx(29),kpd(29),lfs(29),
     1akt(29),vkt(29),vls(29),x1(1),sigp(int,19),phi(int,ign)
      integer vkt,vls
      data akt / 'totx', 'g   ', 'p   ', '2n  ', 'np  ', 't   ',
     1           'he3 ', 'a   ', 'na  ', 'a2n ', '2a  ', 't2a ',
     2           '3a  ', 'n3a ', 'n+  ', 'g+  ', '2n+ ', 'p+  ',
     3           'np+ ', '    ', '*tot', '*b- ', '*b+ ', '*a  ',
     4           '*g  ', '*b->', '*b+>', '*n  ', '    ' /
      data vkt /    0,   -1, 1000,    1, 1001, 1002, 2002, 2003, 2004,
     1     2005, 4007, 5010, 6011, 6012,    0,   -1,    1, 1000, 1001,
     2      2*0,-1000, 1000, 2004,    0,-1000, 1000,    1,    0 /
      data vls / 14*0,   -1,   -1,   -1,   -1, -1, 5*0, 1,-1,-1, 2*0 /
      mng=ign
      mxn=19
      mkt=29
c
      nin=nt9
      non=nt3
c
        jjs=0
        jjr=0
      kount=0
c
   11 read(nin,100) a,m,l
      if(iostatus(nin,istat).ne.0)go to 88
c
      do 15 j=1,29
       idx(j)=0
       dx(j) =0.0
   15 continue
c
      read(nin,101) ln,k1,l1,l2,n1,n2,m,l
           lis=ln
           kza=k1
           lsr=l1
           lxn=l2
           nkt=n1
           nwd=n2
           mmm=m
          nktm=nkt
c
           jjr=jjr+1
```

```
      if(lsr.eq.0) go to 22
c   -    read the rad decay part
      read(nin,111) n1,n2,e1,e2,e3,m,l
      nd1=n1+1
      do 20 i=1,nd1
      read(nin,112) d1,z1,m,l
      if(idx(m).eq.1) go to 25
      idx(m)=1
      dx(m) =d1
      go to 20
   25 nktm=nkt-1
      dx(m)=dx(m)+d1
   20 continue
      if(lxn.eq.0) go to 33
      nkt=nkt-nd1
c
c   -    read the xn part
   22 continue
      do 300 mt=1,mxn
      do 300 it=1,int
      sigp(it,mt)=0.
  300 continue
c   -              phi=1.e+14, uwmak-i,   or   supplied reference flux
      do 400 j=1,nkt
      call fido(nin,n6,x1,mng,m)
      idx(m)=1
      do 410 n=1,mng
  410       dx(m)=dx(m)+x1(n)*reflux(n)
      do 420 l=1,int
      do 430 n=1,mng
      sigp(l,m)=sigp(l,m)+x1(n)*phi(l,n)
  430 continue
  420       sigp(l,m)=sigp(l,m)*fcf
  400 continue
c
   33 continue
      kdx(jjr,1)=lis+kza*10
      kdx(jjr,2)=lxn+lsr*10
      kdx(jjr,3)=nktm
c
      kount=kount+1
          if(mod(kount,40).ne.1) go to 44
      write(n6,200) (i,i=1,19),(i,i=21,29)
   44     continue
      if(kdx(jjr,2).eq.10) go to 454
c
      do 450 m=1,mxn
      adx(m)=1h
  450 if(idx(m).eq.1) adx(m)=1hx
      write(n6,201) kdx(jjr,1),kdx(jjr,3),(adx(m),m=1,mxn)
      if(kdx(jjr,2).eq.1) go to 49
c
  454 continue
      do 460 m=21,mkt
      adx(m)=1h
  460 if(idx(m).eq.1) adx(m)=1ho
      if(kdx(jjr,2).eq.10) go to 48
      write(n6,202) (adx(m),m=21,mkt)
```

```
1278         go to 49
1279      48 write(n6,203) kdx(jjr,1),kdx(jjr,3),(adx(m),m=21,mkt)
1280      49                 nx=0
1281         do 40 j=1,mkt
1282         if(idx(j).eq.0) go to 40
1283                         nx=nx+1
1284         kpd(j)=kza-vkt(j)
1285         lfs(j)=lis-vls(j)
1286         if(j.lt.15) lfs(j)=0
1287         if(j.lt.25.and.j.gt.21)  lfs(j)=0
1288         kdl(jjr,nx)=kpd(j)*1000+lfs(j)*100+j
1289         dcy(jjr,nx)=dx(j)
1290      40 continue
1291   c
1292         if(lxn.eq.0) go to 66
1293         jjs=jjs+1
1294         kkxn(jjs)=kdx(jjr,1)
1295   c   -     write on the xn table file
1296         jjza=kdx(jjr,1)
1297         write(non) jjza,(idx(m),m=1,19),((sigp(it,m),m=1,19),it=1,int)
1298   c
1299      66 read(nin,100) a,m,l
1300         if(l.ne.0)call exit(1)
1301   c     if(l.ne.0) call exit(1) 211
1302         go to 11
1303   c
1304   c  88 end file non
1305      88 call wreof(non)
1306         jrmax=jjr
1307         jsmax=jjs
1308   c
1309         if(lprt2.eq.0) go to 99
1310         write(n6,281) refttl
1311         do 800 i=1,jrmax
1312         write(n6,283) (kdx(i,k),k=1,3)
1313         nkt=kdx(i,3)
1314         do 800 j=1,nkt
1315         lkpd=kdl(i,j)/100
1316         ktyp=mod(kdl(i,j),100)
1317         write(n6,285) lkpd,dcy(i,j),ktyp,akt(ktyp)
1318     800 continue
1319   c
1320     100 format(a4,68x,i5,i3)
1321     101 format(6i6,36x,i5,i3)
1322     111 format(2i6,3e12.3,24x,i5,i3)
1323   c 112 format(12x,2(1pe12.3), 36x,i5,i3)
1324     112 format(12x,2e12.3,36x,i5,i3)
1325     200 format(1h1, 30x,' nuclear data table ', //,
1326        1       8x,'lkza  nkt',2x,'kt=',19i3,/,22x,9i3,/)
1327     201 format(7x,i6,i3,2x,' s',2x,19(2x,a1))
1328     202 format(18x,'r ',2x,9(2x,a1))
1329     203 format(7x,i6,i3,2x,'r ',2x,9(2x,a1))
1330     211 format(14x,a4,3i6,1pe12.3,24x,i5,i3)
1331     281 format(1h1,20x,' index table ', //, 10x, 18a4,
1332        1         //,4x, 'lkza', 5x, 'sr', 2x, 'nkt',3x,'product',
1333        2         4x, 't rate',6x, ' kt',/)
1334     283 format(3x,2i6,i4)
1335     285 format(23x,i6,1pe12.3,5x,i3,3x,a4)
```

```
1336  c
1337     99 return
1338        end
1339        subroutine input(mzm,lcal,miz,mint,jmat,nchn,lkut,mzmt,kumza,
1340       1voli,xmat,aj,volz,vol,wmat,cmp)
1341  c   -          inpt reads card input for the program
1342        common / pointa/ ivolza,ivola,iwmata,icmpa,
1343       1itend,mzma,lcala,miza,minta,jmata,nchna,lkuta,mzmta,kumzaa,
1344       1kenda,nop,nas,nnc,int,izm,ncmp,niiz,ign
1345        common /inte/ n5,n6,nt1 nt2,nt3,nt7,nt8,nt9,lid,lge,lfx,
1346       1lnk,lprt1,lprt2,lprt3,lprt4,lflx,iizm,iint,inop,inas,innc,jrmax,
1347       1kkxn(256),jsmax,kdx(256,3),
1348       1kdl(256,19),kks(139795)
1349        common /rel/ wlld,area,pwr,fcf,htn,htg,htt,rrp,rrw,rrt,
1350       1ttl(18),refttl(18),bop(9),top(9),bas(12),tas(12),
1351       1reflux(46),dcy(256,19),anuk(150),ddd(243677)
1352        dimension mzm(izm),lcal(izm),miz(izm),mint(izm),jmat(nnc),
1353       1nchn(nnc),lkut(nnc),mzmt(izm,nnc),inuk(150)
1354        dimension volz(nnc),vol(int),wmat(nnc,izm),cmp(ncmp,izm)
1355        dimension aj(nnc,izm),voli(niiz),xmat(nnc)
1356        data blnk,star/4h    ,4h  * /
1357        data refttl  /  4*4h     , 4hfirs, 4ht wa, 4hll   , 4hflux, 4h of ,
1358       1              4huwma, 4hk-i ,7*4h    /
1359        data reflux/
1360       1  1.039e-10,5.812e-12,3.863e-12,5.580e-12,4.777e-12,3.890e-12,
1361       1  3.163e-12,2.742e-12,2.460e-12,2.363e-12,2.256e-12,2.248e-12,
1362       1  2.303e-12,2.468e-12,2.678e-12,2.951e-12,3.343e-12,3.874e-12,
1363       1  1.343e-11,1.597e-11,1.713e-11,1.841e-11,2.155e-11,2.083e-11,
1364       1  1.630e-11,5.800e-12,1.953e-11,2.604e-11,4.174e-11,3.196e-11,
1365       1  2.383e-11,1.808e-11,1.429e-11,9.229e-12,4.921e-12,2.414e-12,
1366       1  9.770e-13,3.166e-13,8.004e-14,1.558e-14,2.336e-15,2.735e-16,
1367       1  2.565e-17,2.036e-18,1.650e-19,1.612e-20/
1368        data tas/ 0.0, 60.0, 600.0, 3600.0, 21600.0, 8.64e+04,
1369       16.048e+05, 2.630e+06, 3.156e+07, 3.156e+08, 3.156e+09,
1370       23.156e+10/
1371        data top/ 8.64e+04, 1.315e+06, 2.63e+06, 1.578e+07,
1372       13.156e+07, 6.312e+07, 1.262e+08, 2.525e+08, 5.050e+08/
1373        data bop/ 6h  1 d ,6h  2 w ,6h  1 mo,6h  6 mo,6h  1 yr,
1374       16h  2 yr,6h  4 yr,6h  8 yr,6h 16 yr/
1375        data bas/6h  0   ,6h  1 m ,6h 10 m ,6h  1 hr,6h  6 hr,
1376       16h  1 dy,6h  1 wk,6h  1 mo,6h  1 yr,6h 10 yr,6h100 yr,
1377       26h1000yr/
1378        mng=ign
1379        rewind 5
1380  c   -              max no.  of interval in a zone 20
1381  c   -    read title
1382        read(n5,101) ttl
1383        write(n6,300) ttl
1384  c
1385  c   --   read integer parameters
1386        read(n5,111) lid,lnk,lge,lfx,izm,int,nop,nas,nnc,ncmp
1387        read(n5,111) lprt1,lprt2,lprt3,lprt4,lflx,lfcf
1388  c
1389  c   --   read real parameters
1390        read(n5,112) wlld, htn, htg,htt, fcf
1391        read(n5,112) rrp,rrw,rrt
1392  c
1393  c   -    calculate the first wall area
```

```
1394          call clca(lge,area,rrw,rrt)
1395          pwr=wlld*area*htt/14.1
1396          if(lfcf.eq.1) go to 1
1397          fcf=443.e-24*wlld*area
1398    c
1399        1 kint=0
1400    c
1401          do 10 i=1,izm
1402          mzm(i)=0
1403          volz(i) = 0.0
1404          read(n5,121) iz,nzi,lcal(i), rri,rro
1405    c   -    calculate interval volume
1406          call clcv(lge,nzi,rri,rro,rrt,voli)
1407    c
1408          do 15 j=1,nzi
1409          kint=kint+1
1410          vol(kint)=voli(j)
1411          volz(i) = volz(i) + voli(j)
1412       15 continue
1413    c
1414          do 25 j=1,nnc
1415          mzmt(i,j)=0
1416       25 continue
1417    c
1418          miz(i)=nzi
1419          mint(i)=kint
1420       10 continue
1421    c
1422          if(kint.ne.int)call exit(1)
1423    c     if(kint.ne.int) call exit(1) 121
1424    c
1425          do 30 n=1,ncmp
1426          read(n5,113) (cmp(n,i),i=1,izm)
1427       30 continue
1428    c
1429          do  35 j=1,nnc
1430          read(n5,131) m1, jmat(j), lkut(j), w1
1431    c
1432          do  35 m=1,izm
1433          wmat(j,m)=w1*cmp(m1,m)
1434    c   -    rearranging for the zone
1435          if(wmat(j,m).le.1.0) go to  35
1436          num=mzm(m)+1
1437          mzmt(m,num)=jmat(j)
1438          mzm(m)=num
1439       35 continue
1440    c
1441          if(nop.eq.0) go to 41
1442          do 40 i=1,nop
1443          read(n5,151) bop(i), top(i)
1444       40 continue
1445          go to 44
1446       41 nop=9
1447    c
1448       44 if(nas.eq.0) go to 46
1449          do 45 i=1,nas
1450          read(n5,151) bas(i),tas(i)
1451       45 continue
```

```
      go to 49
   46 nas=12
c
   49 if(lflx-1) 53,56,55
c   -    1.e+14 * 1.0e-24
   53 continue
      refttl(10)=4h10**
      refttl(11)=4h14
      refttl(12)=4hflat
      do 50 j=1,mng
      reflux(j)=1.0e-10
   50 continue
      go to 56
   55 read(n5,101) refttl
      read(n5,112) (reflux(j),j=1,mng)
      do 60 i=1,mng
   60 reflux(i)=reflux(i)*fcf
c
   56 write(n6,301) lid,lnk,lge,lfx,izm,int,nop,nas,nnc,ncmp
c
      write(n6,303) (j,j=1,izm)
      do 80 j=1,nnc
      ll=jmat(j)/1000
      do 85 m=1,izm
      aj(j,m)=blnk
      if(wmat(j,m).gt.1.0e-03) aj(j,m)=star
   85 continue
      anuc=4h
      write(n6,305) anuc, jmat(j),(aj(j,m),m=1,izm)
   80 continue
c
   88 continue
      write(n6,310) rrp,rrw,rrt
      write(n6,311) area, wlld, pwr, fcf
      write(n6,320) nop
      write(n6,321) nas
c
      do 65 i=1,nop
      write(n6,322) bop(i),top(i)
      if(nop.lt.i) go to 65
      write(n6,323) bas(i),tas(i)
   65 continue
      if(nop.ge.nas) go to 69
      max2=nop+1
      do 68 i=max2,nas
   68 write(n6,325) bas(i),tas(i)
c
   69 write(n6,340)
      do 90 m=1,izm
      write(n6,341) m,volz(m)
   90 continue
      write(n6,330) (j,j=1,izm)
      do 70 j=1,nnc
      do 75 m=1,izm
      xmat(m)=wmat(j,m)*(1.0e-18)
   75 continue
      write(n6,331) jmat(j),(xmat(m),m=1,izm)
   70 continue
```

```
c
      write(n6,332) (refttl(i),i=1,18)
  341 format(11x, 'zone', i4,1pe12.3, 2x, 'cm3')
  340 format(//,20x, 'volume of zone',/)
  332 format(//,20x,'reference flux',/,10x,18a4)
  331 format( 2x,i6,5x,15f8.1)
  330 format(1h1, 20x,'nuclide no.  density(10**18)', //,4x,'kza',3x,
     1       'zone',15(i4,4x))
  325 format(    43x,a6,1pe12.3,2x, 'second')
  323 format(1h+,42x,a6,1pe12.3,2x, 'second')
  322 format(     9x,a6,1pe12.3,2x, 'second')
  321 format(1h+,36x,20h after shutdown time,i4,/)
  320 format(/,5x,18h   operating time ,i4,5x)
  311 format( /,6x,26h  first wall area          ,1pe12.3,5x, 'm2',
     a        /,6x,26h  neutron wall loading     ,1pe12.3,5x, 'mw/m2',
     b        /,6x,26h  total operating power    ,1pe12.3,5x, 'mw',
     c        /,6x,26h  flux conversion factor   ,1pe12.3,5x,
     z        /,)
  310 format(1h1,20x,'reactor system parameters', /,
     a         /, 6x,26h  radius of the plasma     ,f8.2,9x, 'cm',
     a         /, 6x,26h  radius of the first wall,f8.2,9x, 'cm',
     c         /, 6x,26h  radius  of the torus     ,f8.2,9x, 'cm',
     z          /,)
  305 format(8x,a6,i6,5x,25a4)
  303 format(20x,'zone',25i4,/)
  301 format( /, 20x, 20h   problem run id    ,i5,
     a     /,6x, 40h  lnk   link to the other solution       , i5,
     b     /,6x, 40h  lge   1/2/3   = slab/cyl/sph           , i5, 5x,
     c     /,6x, 40h  lfx   1/2     = tk3/scalar(anisn)      , i5, 5x,
     d     /,6x, 40h  izm   number of zones                  , i5, 5x,
     e     /,6x, 40h  int   number of intervals              , i5, 5x,
     f     /,6x, 40h  nop   number of operating times        , i5, 5x,
     g     /,6x, 40h  nas   number of after shutdown times   , i5, 5x,
     h     /,6x, 40h  nnc   number of materials(nuclides)    , i5, 5x,
     i     /,6x, 40h  ncmp  number of composition table      , i5, 5x,
     j     /,6x, 40h  ign   number of neutron groups         , '* 46',
     k     /,6x, 40h  igg   nuber of photon gruops           , '* 43',
     z        /,)
  300 format(1h1,18a4)
  101 format(18a4)
  111 format(12i6)
  112 format(6e12.3)
  113 format(12f6.3)
  121 format(3i6,6x,f9.2,3x,f9.2)
  131 format(3i6,e12.3)
  132 format(6(i3,f9.2))
  151 format(a6, e12.4)
c
      return
      end
      subroutine rdchn(nki,mxc,jrnk,jkza,jrxt,cbta,cbtk,lcl,mc,mk)
c   -
      common /inte/ n5,n6,nt1,nt2,nt3,nt7,nt8,nt9,kks(149992)
      dimension jkza(mc,mk),jrxt(mc,mk),lcl(mc,mk),jrnk(mc),
     1          cbta(mc,mk),cbtk(mc,mk)
      dimension j1(9),j2(9),c1(9),c2(9),c3(9),c4(9)
c
      nin=nt2
```

```
      non= n6
           rewind nin
c   -
                    jym=0
                     jx=0
c
c   -    read the required chains
   11 continue
      read(nin,201) mmat,nn,m,l
      if(iostatus(nin,istat) ne.0.and.jx.eq.0)return
      if(iostatus(nin,istat).ne.0)go to 33
      if(mmat.gt.nki)  go to 33
      if(mmat.lt.nki) go to 11
c
                     jx=jx+1
                     jy=0
                jrnk(jx)=nn
c
      do 20 i=1,nn
      read(nin,211) j1(i),j2(i),c1(i),c2(i),c3(i),c4(i)
      jy=jy+1
      jkza(jx,jy)=j1(i)
      jrxt(jx,jy)=j2(i)
      cbta(jx,jy)=c3(i)
      cbtk(jx,jy)=c4(i)
   20 continue
      if(jym.lt.jy) jym=jy
       go to 11
c
   33 if(jx.ne.mxc)call exit(1)
c  33 if(jx.ne.mxc)  call exit(1) 321
c
      do 40 ix=1,mxc
      ix1=ix-1
      jr=jrnk(ix)
      do 45 j=1,jr
      lcl(ix,j)=1
   45 continue
      if(ix1.eq.0) go to 40
      do 50 i=1,ix1
      ir1=jrnk(i)
      do 55 j=1,ir1
      if(jkza(ix,j).ne.jkza(i,j)) go to 50
      lcl(ix,j)=0
   55 continue
   50 continue
   40 continue
c
  201 format(6x,i6,i6,54x,i5,i3)
  211 format(2i6,6x,4e12.4)
c
   91 return
      end
      subroutine size
c   -              size determines variable common size
      common /rel/ dddd(250000)
      common /inte/ iiii(150000)
c     this subroutine has been modified to take into
```

```
1626  c     account variable dimensioning.    It will give the
1627  c     legnths of each array in each program segment
1628        common /pointa/ ivolza,ivola,iwmata,icmpa,
1629       1itend,mzma,lcala,miza,minta,jmata,
1630       1nchna,lkuta,mzmta,kumzaa,kenda,nop,nas,nnc,int,
1631       1izm,ncmp,niiz,ign
1632        integer prt1r,prt2r,prt3r,prt4r,prt1i,prt2i,prt3i,prt4i
1633  c
1634  c    part 1 (input)
1635  c
1636        prt1r=itend+270+(1+izm)*ign+int*19
1637        prt1i=kenda
1638  c
1639  c     part 2 (chain construction)
1640  c
1641        prt2r=itend+3820
1642        prt2i=kenda+2088
1643  c
1644  c    part 3 (executing)
1645  c
1646        prt3r=itend+1309+969*nas+9*nop+(166*nas+342)*niiz
1647        prt3i=kenda+1186
1648  c
1649  c    part 4 (summary)
1650  c
1651        prt4r=itend+6*nas
1652        prt4i=prt1i
1653  c
1654  c    now we print these values
1655  c
1656        write(6,100) prt1r,prt1i,prt2r,prt2i,prt3r,prt3i,prt4r,prt4i
1657    100 format(/,1x,i8,' is the number of real variables in ',
1658       1'segment 1',/,/,1x,i8,' is the number of integer variables',
1659       1'in segment 1',/,/,1x,i8,' is the number of real variables',
1660       1'in segment 2',/,/,1x,i8,' is the number of integer variables',
1661       1'in segment 2',/,/,1x,i8,' is the number of real variables',
1662       1'in segment 3',/,/,1x,i8,' is the number if integer variables',
1663       1'in segment 3',/,/,1x,i8,' is the number of real variables',
1664       1'in segment 4',/,/,1x,i8,' is the number of integer variables',
1665       1'in segment 4',/,/,/,' If any of the integer amounts should',
1666       1'exceed 150,000 or the',/,' real amounts 250,000 extreme ',
1667       1'caution is advised')
1668  c
1669        return
1670        end
1671        subroutine skxn(iv,ir,kt,kint,kxza,ind,sgp,tab)
1672  c   -    skxn reads the xn data
1673        common/pointa/ ivolza,ivola,iwmata,icmpa,
1674       1itend,mzma,lcala,miza,minta,jmata,nchna,lkuta,mzmta,
1675       1kumzaa,kenda,nop,nas,nnc,int,izm,ncmp,niiz,ign
1676        common /rel/dddd(250000)
1677        common /inte/ n5,n6,nt1,nt2,nt3,nt7,nt8,nt9,lid,lge,lfx,
1678       1lnk,lprt1,lprt2,lprt3,lprt4,lflx,iizm,iint,inop,inas,innc,
1679       1jrmax,kkxn(256),jsmax,kdx(256,3),
1680       1kdl(256,19),kks(139795)
1681        dimension tab(int,19),ind(19),kxza(18),ksk(18),sgp(18,19,niiz)
1682  c
1683        nin=nt3
```

```
      non=n6
      rewind nin
c
      do 1000 k=1,iv
c
    2 read(nin) la,(ind(m),m=1,19),((tab(it,m),m=1,19),it=1,int)
      if(iostatus(nin,istat).ne.0)go to 77
      if(la.ne.kxza(k)) go to   2
c
      do 40 ii=1,ir
      kok=kint+ii
      do 40 mm=1,19
   40 sgp(k,mm,ii)=tab(kok,mm)
      if(kok.gt.int)call exit(1)
c     if(kok.gt.int) call exit(1) 331
c
 1000 continue
   99 return
c   -    error in the xn table
   77 write(n6,601)
      write(n6,602) (kxza(k),k=1,iv)
      call exit(1)
c     call exit(1) 332
c
  601 format(/,1x, 'error in cross section search ')
  602 format(10i8)
      end
      subroutine test(lpas,ir,ks,j1,j2,c1,c2 )
c   -    test judges whether the chains continue
      dimension j1(ir),j2(ir),c1(ir),c2(ir)
      dimension ram(9),dif(9),y1(9),yz(9)
c
      lpas=0
      acut=4.0e-3
      lk =ir
      lk1=lk-1
      tir=3.156e+7
c   -     check and list for the particular kcheck
      kcheck=0
      if(j1(1).ne.kcheck) go to 5
      do 100 k=1,lk
  100 write(6,701) j1(k),j2(k),c1(k),c2(k)
  701 format(1x,2i6,4e12.4)
c
    5 y1(1)=1.
      ram(1)=c2(1)
c
      do 10 i=2,lk
      y1(i)=0.
      y1(1)=y1(1)*c1(i)
      ram(i)=c2(i)
   10 continue
c   -    calculate the no density of the last nuclide
c   -    using the bateman's equation
      sum=0.
      do 40 j=1,lk
      dif(j)=1.
      do 50 i=1,lk
```

```
      if(i.eq.j) go to 50
      dnom=abs(ram(i)-ram(j))/(ram(i)+ram(j))
      if(dnom.lt.1.0e-4) ram(j)=1.01*ram(j)
      dif(j)=dif(j)*(ram(i)-ram(j))
c
      if(j1(1).ne.kcheck) go to 50
      write(6,711) y1(1),tir,dif(j),ram(i),ram(j)
  711 format(1x,'y1(1) = ','tir   = ','dif   = ', 'ram(i)=','ram(j)= ',
     1       /,10x,6e13.4)
c   *
   50 continue
      sum=sum+exp(-ram(j)*tir)*y1(1)/dif(j)
   40 continue
c   -    test the last nuclide no density
      yz(lk)=sum
      if(yz(lk).gt.acut) lpas=1
c
      write(6,600)j1(ir),lpas
  600 format(42x,i6,' test  ', i3)
c
   99 return
      end
      subroutine wrdt(lr,iz,ir,jt,j,mm,
     1jk1,jrx,lcl,cb1,cb2,cs1,cs2,ya,y)
      parameter (mc=66,mk=9)
      common/pointa/ ivolza,ivola,iwmata,icmpa,
     1itend,mzma,lcala,miza,minta,jmata,nchna,lkuta,mzmta,
     1kumzaa,kenda,nop,nas,nnc,int,izm,ncmp,niiz,ign
      common /rel/wlld,area,pwr,fcf,htn,htg,htt,rrp,rrw,rrt,
     1ttl(18),refttl(18),bop(9),top(9),bas(12),tas(12),
     1reflux(46),dcy(256,19),anuk(150),ddd(243677)
      common /inte/ n5,n6,nt1,nt2,nt3,nt7,nt8,nt9,lid,lge,lfx,
     1lnk,lprt1,lprt2,lprt3,lprt4,lflx,iizm,iint,inop,inas,innc,
     1jrmax,kkxn(256),jsmax,kdx(256,3),
     1kdl(256,19),kks(139795)
      dimension y(nop,9),ya(mk),jk1(mc,mk),jrx(mc,mk),
     1lcl(mc,mk),cb1(mc,mk),cb2(mc,mk),cs1(mc,mk),cs2(mc,mk)
      go to (1,2,3),lr
    1 write(n6,601) iz
        go to 99
    2 write(n6,602) mm
       go to 99
    3 write(6,701) iz,ir,bop(jt)
        do 60 m=1,mm
   60 write(6,702) jk1(j,m),jrx(j,m),cs1(j,m),cs2(j,m),
     1       cb1(j,m),cb2(j,m),ya(m),y(jt,m)
  601 format(1h1,20x,'executing procedures for zone', i3,//,
     1        2x,'lkza', 2x,'lrz',6x,'ai',8x,'ak',8x,'bi',8x,'bk',
     2        8x,'yo',8x,'yt',5x, 'z,i',4x,'op')
  602 format(//,4x,'mxc = ',i2)
  701 format(73x,'<',i2,',',i2,'>',a6)
  702 format(1x,i6,i5,1x,6(1pe10.3))
   99 return
      end
      subroutine xcute(mzm,lcal,miz,mint,jmat,nchn,lkut,nukz,
     1kumza,volz,vol,wmat,cmp,kxza,ksk,kkza,jrnk,ipr,lcl,jk1,
     2jrx,cy,cb1,cb2,cs1,cs2,wy,ya,y,sgp,cwy,ind,tab)
      parameter (mc=66,mk=9)
```

```
      common / pointa/ ivolza,ivola,iwmata,icmpa,
     1itend,mzma,lcala,miza,minta,jmata,nchna,lkuta,mzmta,kumzaa,
     1kenda,nop,nas,nnc,int,izm,ncmp,niiz,ign
      common /inte/ n5,n6,nt1,nt2,nt3,nt7,nt8,nt9,lid,lge,lfx,
     1lnk,lprt1,lprt2,lprt3,lprt4,lflx,iizm,iint,inop,inas,innc,jrmax,
     1kkxn(256),jsmax,kdx(256,3),
     1kdl(256,19),kks(139795)
      common /rel/ wlld,area,pwr,fcf,htn,htg,htt,rrp,rrw,rrt,
     1ttl(18),refttl(18),bop(9),top(9),bas(12),tas(12),
     1reflux(46),dcy(256,19) anuk(150),ddd(243677)
      dimension mzm(izm),lcal(izm),miz(izm),mint(izm),jmat(nnc),
     1nchn(nnc),lkut(nnc),kumza(150),nukz(izm,nnc)
      dimension volz(nnc),vol(int),wmat(nnc,izm),cmp(ncmp,izm)
      dimension kxza(18),ksk(18),sgp(18,19,niiz),kkza(150),cy(150,nas),
     1cwy(niiz,150,nas),jrnk(66),ipr(66),wy(nas),y(nop,9),jk1(mc,mk),
     1jrx(mc,mk),lcl(mc,mk),cb1(mc,mk),cb2(mc,mk),cs1(mc,mk),
     1cs2(mc,mk),b1(9),b2(9),a(9),ya(9)
      mxn=19
      mkt=29
c
c   -             nt7 :  nuclide activity file
c   -             nt8 :  gamma source file
      kint=0
               if(lnk.le.2) go to 1
c     rewind nt8
    1          continue
c   -             for each zone
      do 1000 iz=1,izm
             iz1=iz-1
c   -             check for vaccum or non material zone
      if(lcal(iz).eq.0) go to 1000
c
      rewind nt7
          mm=mzm(iz)
          nzi=miz(iz)
      if(iz.gt.1) kint=mint(iz1)
c
      if(lprt4.eq.1) call wrdt(1,iz,1,1,1,1,
     1jk1,jrx,lcl,cb1,cb2,cs1,cs2,ya,y)
c
c   -    pick the appropriate jmat
      do 1500 ii=1,mm
        do 26 jj=1,nnc
        if(nukz(iz,ii).eq.jmat(jj)) go to 28
   26   continue
c   -    chain number of each kstt
   28 continue
      nki=nukz(iz,ii)
      mxc=nchn(jj)
c      we now check to see if there are any chains corresponding
c      to the present nuclide.    If there are not we list the
c      original stable nuclide density and assume that it under-
c      goes no transformations in time
      if(mxc .eq.  0) then
      kkza(1)=nki*10
      win=wmat(jj,iz)
      do 2501 ir=1,nzi
      mx=1
```

```
1858        do 2503 irrr=1,jop
1859        do 2502 irr=1,nas
1860        cy(1,irr)=win
1861        write(nt7) ir,irrr,mx
1862        write(nt7) kkza(1),(cy(1,irrrr),irrrr=1,nas)
1863     2502 continue
1864     2503 continue
1865     2501 continue
1866        go to 1500
1867        end if
1868  c   -    recall chains in the zone
1869        call rdchn(nki,mxc,jrnk, jk1,jrx,cb1,cb2,lcl,mc,mk)
1870  c   -    number density of lk1
1871        win=wmat(jj,iz)
1872        kwant=nki*10
1873  c   -    read neutron  xn  of lk1
1874  c   -    and other nuclides in the chain
1875        kok=kint
1876  c
1877        iv=1
1878        ksk(iv)=jk1(1,1)
1879        do 160 j=1,mxc
1880        mjr=jrnk(j)
1881        do 160 m=1,mjr
1882        do 165 ik=1,iv
1883        if(jk1(j,m).eq.ksk(ik)) go to 160
1884     165 continue
1885        do 185 ik=1,jsmax
1886     185 if(jk1(j,m).eq.kkxn(ik)) go to 265
1887        go to 160
1888     265 iv=iv+1
1889        ksk(iv)=jk1(j,m)
1890     160 continue
1891        ivm=iv
1892  c
1893        call psort(ipr,ksk,ivm)
1894  c
1895        do 340 ia=1,ivm
1896        do 350 ib=1,ivm
1897     350 if(ia.eq.ipr(ib)) go to 44
1898        call exit(1)
1899  c     call exit(1) 311
1900     44 kxza(ia)=ksk(ib)
1901     340 continue
1902  c
1903        do 364   i2=1,ivm
1904        do 364   i3=1,mxn
1905        do 364   i4=1,nzi
1906     364                sgp(i2,i3,i4)=0.
1907  c
1908        call skxn(ivm,nzi,mkt,kint,kxza,ind,sgp,tab)
1909        if(lprt4.eq.1)call wrdt(2,iz,1,1,1,mxc,jk1,jrx,lcl,
1910       1cb1,cb2,cs1,cs2,ya,y)
1911  c
1912        do 2000  j=1,mxc
1913              mjr=jrnk(j)
1914             ya(1)=win
1915         do 200 im=2,mjr
```

```
  200     ya(im)=0.
c
        cs1(j,1)=cb1(j,1)
c
        do 2500 ir=1,nzi
        do 65 m=1,mjr
        jxn=mod(jrx(j,m),100)
        m7=m+1
        if(jxn.gt.20) go to 85
        do 465 ll=1,ivm
        if(jk1(j,m).eq.kxza(ll)) go to 466
  465 continue
        go to 85
  466 lt=ll
        cs2(j,m)=cb2(j,m)+sgp(lt,1,ir)
        if(m7.gt.mjr) go to 65
        cs1(j,m7)=cb1(j,m7)+sgp(lt,jxn,ir)
        go to 65
   85 continue
        cs2(j,m)=cb2(j,m)
        if(m7.gt.mjr) go to 65
        cs1(j,m7)=cb1(j,m7)
   65 continue
c
        do 220 m=1,mjr
           a(m)=ya(m)
          b1(m)=cs1(j,m)
          b2(m)=cs2(j,m)
        do 220 lt=1,nop
         y(lt,m)=0.
  220 continue
c
        call clcno(mjr,b1,b2,a,y,top,nop)
c
        do 3000 jt=1,nop
        if(lprt4.eq.1) call wrdt(3,iz,ir,jt,j,mjr,
       1jk1,jrx,lcl,cb1,cb2,cs1,cs2,ya,y)
c   -
c   -    calculate aftershutdown radioactivity
c
  333        mx=0
c    first we check for 2nd order loops
         if(cs1(j,1) .gt.0.)  then
c    thyis means that we have a second order loop
c    we treat this case separately
         do 307 jj=1,2
  307 a(jj)=y(jt,jj)
c    first link in chain
         if(lcl(j,1) .eq.  0) then
         do 321 jj=1,nas
  321 wy(jj)=a(2)*(1.-exp(-tas(jj)*cb1(j,1)))
        go to 302
        end if
c    we check to see if the second nuclide is stable
         if(cb1(j,1) .eq.  0) then
        do 303 jj=1,nas
  303 wy(jj)=a(1)
        else
```

```
1974        do 304 jj=1,nas
1975      304 wy(jj)=a(1)+a(2)*(1.-exp(-tas(jj)*cb1(j,1)))
1976        end if
1977  c     we now file the first isotope
1978      302 jkp=jk1(j,1)
1979        call file(wy,nas,mx,kkza,jkp,cy)
1980  c     now for the second link in the chain
1981        if(lcl(j,2) .eq.  0) go to 305
1982        if(jrx(j,2) .gt.  1000) go to 305
1983        do 306 jj=1,nas
1984      306 wy(jj)=a(2)
1985        jkp=jk1(j,2)
1986        call file(wy,nas,mx,kkza,jkp,cy)
1987  c     done with loop case
1988        go to 305
1989        end if
1990  c     the treatment of regular chains
1991        lflag=0
1992        do 301 iia=1,mjr
1993        lsr=jrx(j,iia)/100
1994        a(iia)=y(jt,iia)
1995        iiam=iia-1
1996        if(lsr .gt.  1) then
1997        lflag=lflag+1
1998        go to 301
1999        end if
2000        if(lflag .eq.  0) then
2001        if(lcl(j,iia) .eq.  0) go to 308
2002  c     we have the case of one link in the chain
2003        jkp=jk1(j,iia)
2004        do 309 jj=1,nas
2005      309 wy(jj)=a(iia)
2006        call file(wy,nas,mx,kkza,jkp,cy)
2007        lflag=0
2008        go to 301
2009        end if
2010        if(lcl(j,iia) .eq.  0) go to 308
2011  c     the case where lflag .gt.  0
2012        lflag=lflag+1
2013        do 310 iij=1,lflag
2014        iik=iij+iia-lflag
2015        a(iij)=y(jt,iik)
2016        b1(iij)=cb1(j,iik)
2017      310 b2(iij)=cb2(j,iik)
2018        call ashut(lflag,nas,b1,b2,a,wy,tas)
2019        jkp=jk1(j,iia)
2020        call file(wy,nas,mx,kkza,jkp,cy)
2021      308 lflag=0
2022        go to 301
2023      301 continue
2024      305 continue
2025        write(nt7) ir,jt,mx
2026        write(nt7)(kkza(kd),(cy(kd,ld),ld=1,nas),kd=1,mx)
2027  c     write(6,400) ir,jt,mx
2028  c     write(6,401) (kkza(kd),cy(kd,1),cy(kd,10),kd=1,mx)
2029      400 format(3(i6,1x))
2030      401 format(3(i6,1x,e9.3,1x,e9.3,1x))
2031     3000 continue
```

```
 2500 continue
 2000 continue
c
 1500 continue
c      end file nt7
       call wreof(nt7)
c   -      the checking point of radioactivity in chains
       call colrad(iz,nzi,kint,mzm,lcal,miz,mint,jmat,nchn,lkut,nukz,
      1kumza,volz,vol,wmat,cmp,cwy,cy)
       call chime(2)
 1000 continue
             if(lnk.le.2) go to 99
c
  101 format(i6)
  110 format(6e12.4)
c
   99 return
      end
      subroutine pkp(nbigma,jj2,jj1,ktp,lkp,kkza,nchn,bigma,cc4,
     1cc3,cc2,cc1,sig,sgk,bta,btk)
c     this is a subroutine for thesetup of chains for the stable
c      nuclides.     we begin over for each nuclide
      parameter (mk=9,mx=19,mc=66,nstore=256)
      common /pointa/ ivolza,ivola,iwmata,icmpa,
     1itend,mzma,lcala,miza,minta,jmata,nchna,lkuta,mzmta,kumzaa,
     1kenda,nop,nas,nnc,int,izm,ncmp,niiz,ign
      common /inte/n5,n6,nt1,nt2,nt3,nt7,nt8,nt9,lid,lge,lfx,
     1lnk,lprt1,lprt2,lprt3,lprt4,lflx,iizm,iint,inop,inas,innc,jrmax,
     1kkxn(256),jsmax,kdx(256,3),
     1kdl(256,19),kks(1)
      common /rel/ wlld,area,pwr,fcf,htn,htg,htt,rrp,rrw,rrt,
     1ttl(18),refttl(18),bop(9),top(9),bas(12),tas(12),
     1reflux(46),dcy(256,19),anuk(150),ddd(1)
      dimension nbigma(9,66,4),bigma(9,66,8),sgk(9),btk(9),
     1sig(19),bta(19),cc1(9),cc2(9),cc3(9),cc4(9),jj1(9),jj2(9),
     2ktp(9),lkp(9),kkza(nnc),nchn(nnc)
      do 1 i=1,nnc
      lk1=kkza(i)*10
c      nnc is the number of nuclides initially
c     first find out if there is data for the nuclide in question.
      do 2 j=1,nstore
      if(lk1 .eq.  kdx(j,1)) go to 3
    2 continue
c     this is the case of no data in our library
c     write(6,100) kkza(i)
      go to 1
  100 format(1x, 'there is no data in our library for',1x,i7)
c      kkza is the nuclide we are investigating
c      kdx is the nuclide in the cross section library
    3 kq=1
      nnn=0
      kn=kq+1
      nkt=kdx(j,3)
      lrs=kdx(j,2)
      call branch(kq,j,nkt,ix,btk,sgk,bta,sig,lkp,ktp,mk,mx)
c     this will construct the first rank of all chains
c     j is the nuclide index
c     ix is the number of branches
```

```
2090  c     bta is the beta from i-1 to i
2091  c     btk is the total decay constant
2092  c     sig is the beta+s*phi for the i to i-1
2093  c      sgk is the total decay plus sigma-phi
2094  c
2095  c     we now initialize our big matrix
2096        do 5 ii=1,9
2097        do 5 jj=1,66
2098        do 174 kk=1,8
2099    174 bigma(ii,jj,kk)=0.
2100        do 4 ll=1,4
2101        nbigma(ii,jj,ll)=0.
2102      4 continue
2103      5 continue
2104  c     we now input the information that we have into the big matrix
2105  c      we load bigma(ii,jj,1) with btk
2106  c              bigma(ii,jj,2) with bta
2107  c              bigma(ii,jj,3) with sgk+btk
2108  c              bigma(ii,jj,4) with sig
2109  c              nbigma(ii,jj,1) with the kza
2110  c              nbigma(ii,jj,2) with nkt+100(10*nsr+nxn)
2111  c      we now construct the second and greater ranks of chains
2112        do 7 ia=1,ix
2113        nbigma(1,ia,1)=kdx(j,1)
2114        nbigma(1,ia,2)=ktp(ia)+kdx(j,2)*100
2115        bigma(1,ia,3)=sgk(1)+btk(1)
2116        bigma(1,ia,1)=btk(1)
2117        bigma(1,ia,2)=0.
2118        bigma(1,ia,4)=0.
2119        nbigma(2,ia,1)=lkp(ia)
2120        bigma(2,ia,4)=sig(ia)
2121      7 bigma(2,ia,2)=bta(ia)
2122        nn=ix
2123     24 continue
2124        kq=kq+1
2125        kn=kq+1
2126        ix=nn
2127        nn=0
2128        do 8 iii=1,ix
2129  c     we investigate each of the branches created earlier
2130  c     we first deal with these in the order that they are shown
2131        write(6,101) nbigma(kq,iii,1)
2132    101 format(3x,i6)
2133        if (kn .ge.9) go to 36
2134        nir=kq
2135  c      the first thing to do is to check if there is a loop
2136        if (nbigma(kq,iii,1) .eq.  lk1) go to 6
2137  c      this is the total loop case
2138  c      this is the case of the reappearance of any nuclide
2139  c     we must now see if a nuclide reappears in the chain
2140        km=kq-1
2141        do 9 ic=1,km
2142        if (nbigma(kq,iii,1).eq.nbigma(ic,iii,1)) then
2143        nodata=1
2144        go to 36
2145        end if
2146  c      if a nuclide repeats itself in a chain (other than the starting
2147  c      nuclide), we terminate the chain
```

```
2148      9 continue
2149        kwant=nbigma(kq,iii,1)
2150  c we find out what the product isotope is and build a chain about that
2151  c we first check to see if kwant is in the library
2152        do 10 ie=1,jrmax
2153  c   jrmax is the number of isotopes in the dcdlib
2154        if (kwant .eq.  kdx(ie,1)) go to 11
2155     10 continue
2156  c     write(6,103) kwant
2157    103 format(3x,'the nuclide ',i6,' has no data and the chain is ended')
2158  c this is the case of no data in the file for the product nuclide, kwant
2159        nodata=1
2160        go to 36
2161  c     this is the case of a second level no data
2162     11 lrs=kdx(ie,2)
2163        nkt=kdx(ie,3)
2164  c      we file away the isotope if the last item is stable
2165  c      we now complete the data necessary for the calculation
2166  c      of the isotope number density.    this allows us to test for
2167  c      cutoff density.
2168  c   we note this also allows us to set up the next rank of chains if
2169  c   the test says we should continue
2170        call branch(kq,ie,nkt,ix,btk,sgk,bta,sig,lkp,ktp,mk,mx)
2171  c    we now see what the density will be by considering the total
2172  c    crossections
2173        nbigma(kq,iii,2)=1+lrs*100
2174        bigma(kq,iii,3)=sgk(kq)+btk(kq)
2175        bigma(kq,iii,1)=btk(kq)
2176        call prep(kq,iii,bigma,nbigma,jj1,jj2,cc1,cc2)
2177  c    this subroutine prepares the data for testing
2178        call test(lpass,kq,lk1,jj1,jj2,cc1,cc2)
2179  c    we must not cut off a radioactive nuclide
2180        if(lrs .eq.1 .and.  lpass .eq.  0) go to 36
2181        n=0
2182        do 14 mm=1,ix
2183        if(lpass.eq.  0 .and.  lrs .eq.  11 .and.  ktp(mm) .lt.  21)
2184       1 go to 14
2185        n=n+1
2186        nn=nn+1
2187        if (nn .gt.  65)go to 8
2188  c      we must now copy the bigma and nbigma over in order to
2189  c      include the latest parts of the chain
2190        call copy(nbigma,bigma,kq,nn,n,iii,mc,mk)
2191  c      this subroutine serves that function
2192  c     we must now supplement the next line of the chain
2193  c     since we couldnt copy the whole line kq before,
2194  c     we complete that first and  supplement it.
2195        nbigma(kq,nn,3)=nbigma(kq,iii,1)
2196        nbigma(kq,nn,4)=ktp(mm)+lrs*100
2197        bigma(kq,nn,5)=btk(kq)
2198        bigma(kq,nn,6)=bigma(kq,iii,2)
2199        bigma(kq,nn,7)=sgk(kq)+btk(kq)
2200        bigma(kq,nn,8)=bigma(kq,iii,4)
2201  c     we now begin the next line for the product nuclide
2202        nbigma(kn,nn,3)=lkp(mm)
2203        bigma(kn,nn,8)=sig(mm)
2204        bigma(kn,nn,6)=bta(mm)
2205     14 continue
```

```
c check if there are any more continuations of that particular
c chain branch
      if(n .eq.  0) go to 36
      go to 8
    6 if(kq .gt.3) go to 36
   20 bigma(1,iii,4)=bigma(kq,iii,4)
      bigma(1,iii,2)=bigma(kq,iii,2)
      nir=kq-1
   36 nnn=nnn+1
      nid=nnn+i*100
      call prepwr(nir,iii,nbigma,bigma,jj1,jj2,cc1,cc2,cc3,cc4)
      call wrchn(nnn,nid,nir,lk1,jj1,jj2,cc1,cc2,cc3,cc4,nodata)
c  if not file it away
c      we now consider the next item at that same rank
    8 continue
      if(nn.eq.  0) go to 99
      do 22 ik=1,nn
      do 23 ij=1,kn
      nbigma(ij,ik,1)=nbigma(ij,ik,3)
      nbigma(ij,ik,2)=nbigma(ij,ik,4)
      bigma(ij,ik,1)=bigma(ij,ik,5)
      bigma(ij,ik,2)=bigma(ij,ik,6)
      bigma(ij,ik,3)=bigma(ij,ik,7)
      bigma(ij,ik,4)=bigma(ij,ik,8)
   23 continue
   22 continue
      go to 24
   99 nchn(i)= nnn
    1 continue
      return
      end
c
c
      subroutine prep(kq,iii,bigma,nbigma,jj1,jj2,cc1,cc2)
c    this subroutine sets up the matrix in order to
c    set up the test
      dimension nbigma(9,66,4),bigma(9,66,8),jj1(9),jj2(9),
     1cc1(9),cc2(9)
c         bigma(i,j,4)=cc1(i)
c         bigma(i,j,3)=cc2(i)
c         nbigma(i,j,1)=jj1(i)
c          nbigma(i,j,2)=jj2(i)
c   where i is variable up to kq and  iii is j
      do 1 i=1,kq
      cc1(i)=bigma(i,iii,4)
      cc2(i)=bigma(i,iii,3)
      jj1(i)=nbigma(i,iii,1)
      jj2(i)=nbigma(i,iii,2)
    1 continue
      return
      end
c
c
      subroutine copy(nbigma,bigma,kq,nn,n,iii,mc,mk)
c    this subroutine copies the new sections of the chains
c into a new matrix signified by the index of bigma
c   increased by 4 and that of nbigma increased by 2
c   we are of course referring to the third index
```

```
      dimension nbigma(9,66,4),bigma(9,66,8)
      kqs= kq-1
      do 1 i=1,kqs
      nbigma(i,nn,3)=nbigma(i,iii,1)
      nbigma(i,nn,4)=nbigma(i,iii,2)
      bigma(i,nn,5)=bigma(i,iii,1)
      bigma(i,nn,6)=bigma(i,iii,2)
      bigma(i,nn,7)=bigma(i,iii,3)
      bigma(i,nn,8)=bigma(i,iii,4)
    1 continue
      return
      end
c
c
      subroutine wrchn(nnn,nid,nrank,lk1,jj1,jj2,cc1,cc2,cc3,cc4,nodata)
c     this subroutine writes all the chains down
      common / inte / iiiia(5912),nucl(166),nucl1(118),iiiib(1)
      dimension jj1(9),jj2(9),cc1(9),cc2(9),cc3(9),cc4(9)
      if(nodata .eq.  1) then
      nodata=0
      do 5 ll=nrank,2,-1
      mmat=jj1(ll)
      do 2 ii=1,166
      if(mmat .eq.  nucl(i)) go to 4
    2 continue
      do 3 ii=1,118
      if (mmat .eq.  nucl(i)) go to 4
    3 continue
    5 continue
      nnn=nnn-1
      nid=nid-1
      go to 99
c      the preceding section makes sure that the chain actually
c      ends up on a stable nuclide and not a radioactive one
c      when there is no data.
    4 continue
      nrank=ll
      jj2(nrank)=101
      end if
      non=2
      mmat=lk1/10
c      we can't stop on a radioactive isotope no matter what
c      we therefore write everthing
      write(non,101) mmat,nrank,nid,nnn
      write(6,103) nid
  101 format(6x,i6,i6,54x,i5,i3)
  103 format(2x,i6)
      if (jj2(nrank) .lt.  101) jj2(nrank) = 101
      do 1 i=1,nrank
      write(6,104) jj1(i),jj2(i),cc1(i),cc2(i),cc3(i),cc4(i),nid,i
      write(non,102) jj1(i),jj2(i),cc1(i),cc2(i),cc3(i),cc4(i),nid
    1 continue
  102 format(2i6,6x,4e12.4,i5)
  104 format(9x,2i6,4e12.4,3x,i5,i3)
   99 return
      end
c
      subroutine prepwr(nir,iii,nbigma,bigma,jj1,jj2,cc1,cc2,cc3,cc4)
```

```
c      this subroutine prepares the chains tho be writed in the buffer
       dimension nbigma(9,66,4),bigma(9,66,8),jj1(9),jj2(9),
      1cc1(9),cc2(9),cc3(9),cc4(9)
       do 1 i=1,nir
       jj1(i)=nbigma(i,iii,1)
       jj2(i)=nbigma(i,iii,2)
       cc1(i)=bigma(i,iii,4)
       cc2(i)=bigma(i,iii,3)
       cc3(i)=bigma(i,iii,2)
       cc4(i)=bigma(i,iii,1)
     1 continue
       return
       end
       subroutine psort(ip,nx,n)
c      this subroutine does the pointing
       dimension ip(n),nx(n)
c
       do 10 j=1,n
       ip(j)=1
       jsub=j-1
       jpls=j+1
       if(jsub .eq.  0) go to 22
       do 110 l=1,jsub
       if(nx(j) .ge.  nx(l)) ip(j)=ip(j)+1
   110 continue
    22 if(jpls .gt.  n) go to 10
       do 210 l=jpls,n
       if(nx(j) .gt.  nx(l)) ip(j) =ip(j)+1
   210 continue
    10 continue
c
       return
       end
c
c
       subroutine avge(inuk,npnuk,cwy,bop,vol,volz,nzi,nas,nop,
      1izm,kxm,kint,jop,liz,int,niiz,bas,wmat,nnc)
       dimension inuk(150),npnuk(150),cwy(niiz,150,nas),vol(int),
      1bop(nop),volz(izm),bas(nas),wmat(nnc,izm)
       dimension avg(150,20)
c      this subroutine averages the densities of elements over
c      a zone
       do 2 k=1,nas
       do 2 i=1,kxm
       avg(i,k)=0.
       do 2 j=1,nzi
       knew=kint+j
     2 avg(i,k)=avg(i,k)+cwy(j,i,k)*vol(knew)/volz(liz)
c      write out the answers
       write(6,100) liz,bop(jop),bas
   100 format(1h1,30x,'the volume averaged densities for zone ',i3,
     1/,30x,'for ',a6,' after shutdown',//,1x,'nuclide',5x,6(a6,3x),
     11x,6(a6,2x))
       do 1 i=1,kxm
       do 3 j=1,kxm
     3 if(i .eq.  npnuk(j)) go to 4
       stop 500
     4 write(6,101) inuk(j),(avg(j,l),l=1,nas)
```

```
  101 format(1x,i6,2x,6(1pe9.2),6(1pe9.2))
    1 continue
      call appm(inuk,npnuk,avg,bop,bas,nas,nop,izm,kxm,jop,liz,wmat,
     1nnc)
      return
      end
c
c
      subroutine wrout(liz,jop,kxm,cwy,bop,nop,nas,ia,niiz,inuk,npnuk,
     1kint,bas,lprt4)
      dimension bop(jop),cwy(niiz,150,nas),inuk(150),npnuk(150)
      dimension bas(nas)
      ir=ia+kint
      if (lprt4 .eq.  1) then
      write(6,100) liz,ir,bop(jop),(bas(js),js=1,nas)
      end if
      call psort(npnuk,inuk,kxm)
      do 70 i=1,kxm
      do 60 j=1,kxm
   60 if (i .eq.  npnuk(j)) go to 80
      stop 400
   80 if(lprt4 .eq.  1) then
      write(6,101) inuk(j),(cwy(ia,j,js),js=1,nas)
      end if
  100 format(1h1,30x,'stable isotopes for zone ',i3,' interval ',
     1i3,/,30x,'for an operating time of ',a6,//,1x,'nuclide',
     2 1x,6(3x,a6),6(3x,a6),/)
  101 format(1x,i6,2x,6(1pe9.2),6(1pe9.2))
   70 continue
      return
      end
c
c
      subroutine file(wy,nas,mx,kkza,jkp,cy)
c     this subroutine files away the stable isotopes
c
      dimension wy(nas),kkza(mx),cy(150,nas)
      if(mx .eq.  0) go to 10
c     the case where there is at least one isotope filed
      do 20 i=1,mx
   20 if(jkp .eq.  kkza(i)) go to 30
   10 mx=mx+1
      kkza(mx)=jkp
      do 40 j=1,nas
   40 cy(mx,j)=abs(wy(j))
      return
   30 do 50 j=1,nas
   50 cy(i,j)=cy(i,j)+abs(wy(j))
      return
      end
      subroutine appm(inuk,npnuk,avg,bop,bas,nas,nop,izm,kxm,
     1jop,liz,wmat,nnc)
      character*2 elem(98)
      integer akza(150)
      dimension iatmno(150)
      dimension wmat(nnc,izm),inuk(150),npnuk(150),avg(150,20)
      dimension bop(jop),bas(nas)
      data elem/' h','he','li','be',' b',' c',' n',' o',' f','ne',
```

```
2438       1'na','mg','al','si',' p',' s','cl','ar',' k','ca','sc','ti',
2439       2' v','cr','mn','fe','co','ni','cu','zn','ga','ge','as','se',
2440       3'br','kr','rb','sr',' y','zr','nb','mo','tc','ru','rh','pd',
2441       4'ag','cd','in','sn','sb','te',' i','xe','cs','ba','la','ce',
2442       5'pr','nd','pm','sm','eu','gd','tb','dy','ho','er','tm','yb',
2443       6'lu','hf','ta',' w','re','os','ir','pt','au','hg','tl','pb',
2444       7'bi','po','at','rn','fr','ra','ac','th','pa',' u','np','pu',
2445       8'am','cm','bk','cf'/
2446        factor=1.0e-10
2447        do 10 i=1,nnc
2448      10 factor=factor+wmat(i,liz)
2449        factor=1.0e+6/factor
2450        do 2 k=1,nas
2451        do 2 i=1,kxm
2452       2 avg(i,k)=avg(i,k)*factor
2453        do 20 i=1,150
2454        iscr=inuk(i)/10
2455        akza(i)=iscr/1000
2456      20 iatmno(i)=mod(iscr,1000)
2457        write(6,100) liz,bop(jop),bas
2458     100 format(1h1,20x,' The atomic parts per million of isotopes for
2459       1zone ',i3,/,30x,'and ',a6,' operation time',//1x,'nuclide',5x,
2460       26(a6,3x),1x,6(a6,3x))
2461        do 1 i=1,kxm
2462        do 3 j=1,kxm
2463       3 if(i .eq.  npnuk(j)) go to 4
2464        stop 500
2465       4 write(6,101) elem(akza(j)),iatmno(j),(avg(j,l),l=1,nas)
2466     101 format(1x,a2,1x,i3,2x,6(1pe9.2),6(1pe9.2))
2467       1 continue
2468        return
2469        end
```