MASTER

Pacific Northwest Laboratories

Battelle Boulevard / Richland, Washington 99352

RACER - A COMPUTER PROGRAM
FOR CALCULATING POTENTIAL EXTERNAL DOSE
FROM AIRBORNE FISSION PRODUCTS
FOLLOWING POSTULATED REACTOR ACCIDENTS

JUNE 1971

BNWL-B-69

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE MEMORIAL INSTITUTE
for the
U. S. ATOMIC ENERGY COMMISSION
Under Contract AT(45-1) 1830

APPENDIX E

PROGRAM LISTING

A complete listing of the program follows.

The order of the program elements is:

RACER (segmented map)	CHAIN1
MAIN	REDUCE
SOURCE	PLUME
BAFFLE	SEMME
EXDOSE	LIB
DECODE	CHAIN
RIBD	FIT
FIRST	PASSIG
SECOND	FINT
THIRD	LINEAR
FOURTH	THERM
FIFTH	FAST
SIXTH	ISOLIB
LAST	FILTER
SUMZ	INDOSE
ZERO	

Q ELT RACER, 1,700630, 45459

1. SEG MAIN-TITLE-DECODE-*(SOURCE,BAFFLE,EXDOSE,INDOSE)
2. BLK TITLE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

```

1.  C MAIN PROGRAM      MAIN READS FIRST TWO CARDS OF EACH PACKET AND CALLS THE
2.  C DESIRED SUBROUTINE.- SOURCE, BAFFLE, OR EXDOSE. A DUMMY CALL TO INDOSE IS
3.  C INCLUDED FOR FUTURE EXPANSION.
4.  C
5.  C SUBROUTINES ETIME AND ETIMEF ARE SYSTEM TIMING ROUTINES
6.  C
7.  C   DIMENSION ITAB(42),LOGIC1(8),WORD(14),LISTM(4),LIST1(5),LIST2(8),
8.  C   1LIST3(6),LIST4(4)
9.  C   COMMON/LOGIC/LOGIC1
10.  C   COMMON/TITLE/TITLE(14)
11.  C   INTEGER WORD,WORDS
12.  C   LOGICAL SKIP,LOGIC1
13.  C   DATA LIST1/6H EXDOSE,6H BAFFLE,6H SOURCE,6H INDOSE/
14.  C   DATA LIST2/6H LIBRAR,4H TAPE,6H BACKSP,6H REWIND,4H SEMI/
15.  C   DATA LIST3/6H LIBRAR,6H TAPEIN,6H TAPEOU,6H BYPASS,6H REWIND,6H BACKSP,
16.  C   12H IN,5H ACEIN/
17.  C   DATA LIST4/6H LIBRAR,6H BYPASS,4H TAPE,6H REWIND,6H BACKSP,6H ACTIVA/
18.  C   DATA LIST5/6H LIBRAR,4H TAPE,6H REWIND,6H BACKSP/
19.  C
20.  C ENABLE POST EXECUTION TAPE HANDLING IN CASE OF ERROR MODE
21.  C CALL TRAP('MAIN') REMOVED
22.  C
23.  C EQUATE UNIT 29 WITH KIRD LIBRARY
24.  C REQUIRES FOLLOWING CARD BEFORE XQT CARD
25.  C 3 DAT THERM,1      OR      4 DAT FAST,1
26.  C   CALL SETDAR(29,1,ITAB3)
27.  C
28.  C EQUATE UNIT 28 WITH SHIELDING LIBRARY
29.  C REQUIRES FOLLOWING CARD BEFORE XQT CARD
30.  C 3 DAT ISOLIB,2
31.  C   CALL SETDAR(28,2,ITAB(8))
32.  C
33.  C EQUATE UNIT 25 WITH FILTER LIBRARY
34.  C REQUIRES FOLLOWING CARD BEFORE XQT CARD
35.  C 3 DAT FILTER,3
36.  C   CALL SETDAR(25,3,ITAB(29))
37.  C
38.  C EQUATE UNIT 24 WITH INDOSE LIBRARY
39.  C REQUIRES FOLLOWING CARD BEFORE XQT CARD
40.  C 3 DAT INLIB,4
41.  C   CALL SETDAR(24,4,ITAB(36))
42.  C
43.  C SETUP DRUM FOR SOURCE OUTPUT
44.  C   CALL SETDR(27,343000,1000,ITAB(15))
45.  C
46.  C SETUP DRUM FOR CONTAINMENT OUTPUT
47.  C   CALL SETDR(26,344000,966000,ITAB(22))
48.  C
49.  C READ 80 COLUMNS OF TEXT AND CATCH END-OF-FILE
50.  C   CALL SETEOF(53)
51.  C
52.  C   100 READ 101, TITLE
53.  C   101 FORMAT(13A6,A2)
54.  C
55.  C READ COMMAND AND LOGIC
56.  C   CALL ETIME
57.  C   CALL DECODE(WORD,WORDS)
58.  C
59.  C FIND WHICH FUNCTION -FIRST WORD-
60.  C   DO 1 I=1,4
61.  C   IF(WORD(I).EQ.LISTM(I)) GO TO 4
62.  C   1 CONTINUE
63.  C   PRINT 2
64.  C   2 FORMAT('01ST WORD IS NOT A MAJOR PROGRAM FUNCTION - STOP')
65.  C   3 STOP
66.  C
67.  C BRANCH TO APPROPRIATE PLACE
68.  C   4 GO TO (5,12,13,24),I
69.  C
70.  C EXDOSE DESIRED - SORT THRU LOGIC
71.  C   5 DO 6 I=1,5
72.  C   6 LOGIC1(I)=.FALSE.
73.  C   IF(WORDS.LE.1) GO TO 211
74.  C   SKIP=.FALSE.
75.  C   DO 11 I=2,WORDS
76.  C   IF(SKIP) GO TO 10
77.  C   DO 7 J=1,5
78.  C   IF(WORD(I).EQ.LIST1(J)) GO TO 9
79.  C   7 CONTINUE
80.  C   PRINT 8, WORD(I)
81.  C   8 FORMAT('03RD WORD -'A6,'- STOP')
82.  C   GO TO 3
83.  C   9 LOGIC1(J)=.TRUE.
84.  C
85.  C WORDS 1 AND 3 HAVE MORE THAN SIX CHARACTERS - SKIP THE SECOND HALF
86.  C   IF(J.EQ.1.OR.J.EQ.3) SKIP=.TRUE.
87.  C   GO TO 11
88.  C   10 SKIP=.FALSE.

```

01/25/71

06/29/70

06/29/70

07/22/70

```

89.      11 CONTINUE
90.      211 CONTINUE
91.      CALL EXDOSE
92.      CALL ETIMEF(SEC)
93.      PRINT 111, SEC
94.      111 FORMAT(' EXECUTION TIME FOR EXDOSE WAS 'F7.3,' SECONDS')
95.      GO TO 100
96.      C
97.      C CONTAINMENT DESIRED - SORT THRU LOGIC
98.      12 DO 13 I=1,8
99.      13 LOGIC1(I)=.FALSE.
100.      IF(WORDS.LE.1) GO TO 217
101.      SKIP=.FALSE.
102.      DO 17 I=2,WORDS
103.      IF(SKIP) GO TO 16
104.      DO 14 J=1,6
105.      IF(WORD(I).EQ.LIST2(J)) GO TO 15
106.      14 CONTINUE
107.      PRINT 8, WORD(I)
108.      GO TO 3
109.      15 IF(J.EQ.5) GO TO 115
110.      IF(J.EQ.6) GO TO 415
111.      LOGIC1(J)=.TRUE.
112.      C
113.      C WORDS 1 AND 3 HAVE MORE THAN SIX CHARACTERS - SKIP THE SECOND HALF
114.      IF(J.EQ.1.OR.J.EQ.3) SKIP=.TRUE.
115.      GO TO 17
116.      C
117.      C WHEN WORD IS REWIND TEST NEXT WORD TO DISTINGUISH BETWEEN REWINDIN, REWINDOUT
118.      115 IF(WORD(I+1).EQ.LIST2(7)) GO TO 215
119.      LOGIC1(7)=.TRUE.
120.      GO TO 315
121.      215 LOGIC1(5)=.TRUE.
122.      315 SKIP=.TRUE.
123.      GO TO 17
124.      C
125.      C WHEN WORD IS BACKSP TEST WORD(I+1) TO IDENTIFY BACKSPACEIN OR BACKSPACEOUT
126.      415 IF(WORD(I+1).EQ.LIST2(8)) GO TO 515
127.      LOGIC1(8)=.TRUE.
128.      GO TO 315
129.      515 LOGIC1(6)=.TRUE.
130.      GO TO 315
131.      16 SKIP=.FALSE.
132.      17 CONTINUE
133.      217 CONTINUE
134.      CALL BAFFLE
135.      CALL ETIMEF(SEC)
136.      PRINT 117, SEC
137.      117 FORMAT(' EXECUTION TIME FOR BAFFLE WAS 'F7.3,' SECONDS')
138.      GO TO 100
139.      C
140.      C SOURCE DESIRED - SORT THRU LOGIC
141.      18 DO 19 I=1,6
142.      19 LOGIC1(I)=.FALSE.
143.      IF(WORDS.LE.1) GO TO 223
144.      SKIP=.FALSE.
145.      DO 23 I=2,WORDS
146.      IF(SKIP) GO TO 22
147.      DO 20 J=1,6
148.      IF(WORD(I).EQ.LIST3(J)) GO TO 21
149.      20 CONTINUE
150.      PRINT 8, WORD(I)
151.      GO TO 3
152.      21 LOGIC1(J)=.TRUE.
153.      C
154.      C WORDS 1,5 AND 6 HAVE MORE THAN SIX CHARACTERS - SKIP THE SECOND HALF
155.      IF(J.EQ.1.OR.J.EQ.5.OR.J.EQ.6) SKIP=.TRUE.
156.      GO TO 23
157.      22 SKIP=.FALSE.
158.      23 CONTINUE
159.      223 CONTINUE
160.      CALL SOURCE
161.      CALL ETIMEF(SEC)
162.      PRINT 123, SEC
163.      123 FORMAT(' EXECUTION TIME FOR SOURCE WAS 'F7.3,' SECONDS')
164.      GO TO 100
165.      C
166.      C THE FOLLOWING CARDS ARE INCLUDED FOR FUTURE EXPANSION TO INHALATION DOSE
167.      C CALCULATION. A DUMMY ROUTINE INDOSE IS SUPPLIED.
168.      C INDOSE DESIRED - SORT THRU LOGIC
169.      24 DO 25 I=1,4
170.      25 LOGIC1(I)=.FALSE.
171.      IF(WORDS.LE.1) GO TO 229
172.      SKIP=.FALSE.
173.      DO 29 I=2,WORDS
174.      IF(SKIP) GO TO 23
175.      DO 26 J=1,4
176.      IF(WORD(I).EQ.LIST4(J)) GO TO 27
177.      26 CONTINUE
178.      PRINT 8, WORD(I)
179.      GO TO 3
180.      27 LOGIC1(J)=.TRUE.

```

07/22/70

07/22/70

07/22/70

07/22/70

07/22/70

07/22/70

07/22/70

07/22/70

07/22/70

```

181. C
182. C WORDS 1 AND 4 HAVE MORE THAN SIX CHARACTERS - SKIP THE SECOND HALF, WORD(I+1)
183. IF(J.EQ.1.OR.J.EQ.4) SKIP=.TRUE.
184. GO TO 29
185. 28 SKIP=.FALSE.
186. 29 CONTINUE
187. 229 CONTINUE
188. CALL INDOSE
189. CALL ETIMEF(SEC)
190. PRINT 129, SEC
191. 129 FORMAT(' EXECUTION TIME FOR INDOSE WAS 'F7.3,' SECONDS')
192. GO TO 100
193. C
194. END

```

07/22/70

07/22/70

07/22/70


```

1.  C SUBROUTINE SOURCE CONTROLS INVENTORY CALCULATION AND SPECIFICATION 03/25/71
2.  C 03/25/71
3.  SUBROUTINE SOURCE
4.  DIMENSION NAME(450),M(450)
5.  COMMON/TITLE/TITLE(14)
6.  COMMON/LOGIC/LIBRARY,BYPASS,TAPE,REWIND,BACKUP,ACTIVA
7.  COMMON NCD,ML(600),FEED3(600),FEED4(450),REC(600),ISO(450),FEED10(
8.  A450),QUANTITY(450),MASS(600),DATA1(10,450)
9.  COMMON SPACE(7935),LA(450),FEED4(450),FEED5(450),FEED6(450),FEED7(
10.  1450),FEED9(450)
11.  COMMON PUN 12/16/70
12.  LOGICAL PUN 12/16/70
13.  EQUIVALENCE (DATA1(1,1),NAME(1))
14.  EQUIVALENCE (DATA1(1,46),M(1))
15.  REAL MASS
16.  INTEGER UNIT,STATUS,REC
17.  LOGICAL LIBRARY,BYPASS,TAPE,REWIND,BACKUP,ACTIVA
18.  PUN = .FALSE. 12/16/70
19.  IF(.NOT.LIBRARY) GO TO 4
20.  C
21.  C READ NUCLIDE DATA LIBRARY 03/25/71
22.  READ (29,1) NCD 03/25/71
23.  1 FORMAT(I3)
24.  READ (29,2) (ML(I),LA(I),FEED3(I),FEED4(I),FEED5(I),FEED6(I),FEED7
25.  1(I),FEED8(I),FEED9(I),FEED10(I),REC(I),ISO(I),I=1,NCD)
26.  2 FORMAT(I3,12,4E9.3,4F5.3,7X,A3,I1)
27.  DO 3 I=1,NCD
28.  FEED3(I)=.69314/FEED3(I)
29.  FEED9(I)=5.9274E-9*FEED9(I)
30.  3 FEED10(I)=5.9274E-9*FEED10(I)
31.  READ(29,1) NACT
32.  N=NCD+1
33.  N1=NCD+NACT
34.  READ(29,103) (ML(L),FEED3(L),REC(L),L=N,N1)
35.  103 FORMAT(I3,2XF9.0,54XA3)
36.  DO 203 I=N,N1
37.  203 FEED3(I)=.69314/FEED3(I)
38.  4 DO 104 I=1,N1
39.  C
40.  C INITIALIZE NUCLIDE MASS ARRAY 03/25/71
41.  104 MASS(I)=0.0 03/25/71
42.  J1=1
43.  IF(BYPASS) GO TO 10
44.  C CALL SUBROUTINE RIBS FOR NUCLIDE INVENTORY CALCULATION 03/25/71
45.  CALL RIBS
46.  DO 5 I=1,450
47.  5 MASS(I)=86400.*DATA1(1,1)
48.  IF(.NOT.ACTIVA) GO TO 17
49.  J1=N
50.  C
51.  C READ NUCLIDE INVENTORY FROM CARDS 03/25/71
52.  10 READ 1,NUMBER 03/25/71
53.  READ 11, (NAME(I),M(I),QUANTITY(I),I=1,NUMBER)
54.  11 FORMAT(4(A3,I3,E14.0))
55.  PRINT 6
56.  6 FORMAT('OINPUT CURIES')
57.  PRINT 7, (NAME(I),M(I),QUANTITY(I),I=1,NUMBER)
58.  7 FORMAT(6(4XA3,I3,1FF10.4)/)
59.  C
60.  C TRY TO IDENTIFY NUCLIDES READ FROM CARDS 03/25/71
61.  DO 16 I=1,NUMBER 03/25/71
62.  DO 13 J=J1,N1
63.  IF(NAME(I).EQ.REC(J).AND.M(I).EQ.ML(J)) GO TO 15
64.  13 CONTINUE
65.  C
66.  C IF CAN NOT IDENTIFY NUCLIDE -STOP 03/25/71
67.  PRINT 14, NAME(I),M(I) 03/25/71
68.  14 FORMAT('UNKNOWN ISOTOPE - 'A3,I3)
69.  STOP
70.  15 MASS(J)=QUANTITY(I)/FEED3(J)*86400.
71.  C
72.  C CONVERT CURIES TO CURIE SECONDS 03/25/71
73.  16 CONTINUE 03/25/71
74.  C
75.  C WRITE HEADING AND NUCLIDE INVENTORY TO TAPE OR DRUM 03/25/71
76.  17 UNIT=27 03/25/71
77.  IF(TAPE) UNIT=13
78.  CALL NTRAC(UNIT,1,14,TITLE,STATUS)
79.  117 IF(STATUS=1) 19,117,217
80.  217 CALL NTRAC(UNIT,1,N1,MASS,STATUS)
81.  18 IF(STATUS=1) 19,18,21
82.  19 PRINT 20,STATUS
83.  20 FORMAT('O.FITE TROUBLES IN SOURCE - 'A3,I3)
84.  STOP
85.  21 IF(.NOT.PUN) GO TO 221 12/16/70
86.  C
87.  C PURCH NUCLIDE INVENTORY ON CARDS FOR FUTURE USE 03/25/71
88.  DO 121 I=1,NCD 03/25/71
121 12/16/70

```

89.	MASS(1)=MASS(1)+FEED3(I)/56400.	12/16/70
90.	121 CONTINUE	12/16/70
91.	PUNCH 1, NCD	12/16/70
92.	PUNCH 321, (REC(1),ML(I),MASS(I),I=1,NCD)	12/16/70
93.	321 FORMAT(4(A3,I3,1PE14.3))	12/16/70
94.	C	03/25/71
95.	C PERFORM NECESSARY TAPE OR DRUM HANDLING OPERATIONS	03/25/71
96.	221 IF(TAPE) GO TO 23	12/16/70
97.	REWIND 27	
98.	22 REWIND 29	
99.	RETURN	
100.	23 END FILE 13	
101.	IF(BACKUP) GO TO 24	
102.	IF(REWIND) REWIND 13	
103.	GO TO 22	
104.	24 CALL NTRAN(13,8,-1)	
105.	CALL NTRAN(13,7,-2)	
106.	GO TO 22	
107.	END	

```

1.  C SUBROUTINE BAFFLE PERFORMS CONTAINMENT BARRIER CALCULATION. 03/25/71
2.  C OUTPUT IS RELEASE RATE FOR EACH NUCLIDE AS A FUNCTION OF TIME. 03/25/71
3.  C 03/25/71
4.  SUBROUTINE BAFFLE 03/25/71
5.  COMMON ISOTOP,CHAINS,NOFNUC(96),DELTAT,MASS(600,10),RATE(50,7,8),
6.  1TIME(50),DIFRCT(450,3),FILTER(600,2),LAMBDA(600),SKIP(96),H(133),
7.  2A(28),Q(105),P(105),G(84),F(84),E(105),D(63),C(63),B(77),FOR(3),
8.  3ALR(7),ILR(7,8),OLR(7),QUANTY(600),EXP0(35),EEXP0(35),REC(600),
9.  4ML(600),JUMP(10),JUMPOF,JUMPUP,FRATES(50,3),OUT(600)
10.  COMMON SAVE(3),JUMP11(600)
11.  DIMENSION ISO(450),FRACT(450),LOSS(600,2)
12.  DIMENSION GAS(5),HAL(4),VOL(6),IFILT(4),RELF(4),DEPF(4),FILT(4) 06/12/70
13.  DIMENSION FALOUT(3),LEAK(7,8)
14.  DIMENSION CONTEN(10)
15.  LOGICAL LEAK,FALOUT,FREAD
16.  COMMON/LOGIC/LIBRARY,TAPEIN,TAPOUT,BYPASS,RIN,BIN,ROUT,BOUT
17.  COMMON/TITLE/TITLE1(14),TITLE2(14)
18.  EQUIVALENCE (MASS(1,2),LOSS),(MASS(1,4),ISO),(MASS(1,5),FRACT)
19.  REAL ILR
20.  INTEGER FLOWS,FALLS
21.  LOGICAL JUMP,JUMPOF,JUMPUP
22.  LOGICAL JUMP11
23.  REAL LOSS,MASS,LAMBDA
24.  INTEGER READS,UNIT,STATUS,SKIP,CHAINS
25.  LOGICAL LIBRARY,TAPEIN,TAPOUT,BYPASS
26.  LOGICAL RIN,BIN,ROUT,BOUT
27.  DATA GAS/2HXR,3HXR*,2HXR*,2HXR*,2HXR*/
28.  DATA HAL/1HI,2HCL,2H3R,3H3R*/
29.  DATA VOL/2HSE,3HSE*,2HTE,3HTE*,2HCS,3HCS*/
30.  IF(.NOT.LIBRARY) GO TO 14
31.  C 03/25/71
32.  C READ NUCLIDE DATA LIBRARY 03/25/71
33.  READ (29,1) NCD
34.  1 FORMAT(I3)
35.  READ (29,2) (ML(J),LAMBDA(J),FRACT(J),REC(J),ISO(J),J=1,NCD)
36.  2 FORMAT(I3,2X9.3,32X5.3,17X43,I1)
37.  READ(29,1) NACT
38.  N=NCD+1
39.  N1=NCD*NACT
40.  READ(29,102) (ML(L),LAMBDA(L),REC(L),L=N,N1)
41.  102 FORMAT(I3,2X9.0,54X43)
42.  C REWIND LIBRARY UNIT
43.  REWIND 29
44.  C 03/25/71
45.  C SET UP DATA FOR NUCLIDE CHAIN DECAY CALCULATION 03/25/71
46.  MP=0
47.  CHAINS=0
48.  DO 4 I=1,96
49.  4 NOFNUC(I)=0
50.  DO 8 J=1,NCD
51.  LAMBDA(J)=.69314/LAMBDA(J)/86400.
52.  IF(MP-ML(J)) 5,6,6
53.  5 MP=ML(J)
54.  C NEW MASS CHAIN
55.  CHAINS=CHAINS+1
56.  C COUNT NUMBER OF NUCLIDES IN THIS CHAIN
57.  6 NOFNUC(CHAINS)=NOFNUC(CHAINS)+1
58.  DO 7 L=1,3
59.  7 DIFRCT(J,L)=0.0
60.  8 CONTINUE
61.  DO 108 I=N,N1
62.  108 LAMBDA(I)=.69314/86400./LAMBDA(I)
63.  C ISO=1 MEANS ISOMERIC STATE, =2 MEANS GROUND STATE
64.  ISOTOP=0
65.  DO 13 K=1,CHAINS
66.  LIM=NOFNUC(K)
67.  J=0
68.  DO 12 I=1,LIM
69.  J=J+1
70.  ISOTOP=ISOTOP+1
71.  IF(J,EG.1) GO TO 11
72.  IF(ISO(ISOTOP),EG.2) GO TO 9
73.  DIFRCT(ISOTOP,1)=FRACT(ISOTOP-1)
74.  GO TO 11
75.  9 DIFRCT(ISOTOP,1)=1.-FRACT(ISOTOP-1)
76.  IF(J,EG.2) GO TO 11
77.  IF(ISO(ISOTOP-1),EG.1) GO TO 10
78.  IF(ISO(ISOTOP-2),EG.2) GO TO 11
79.  DIFRCT(ISOTOP,2)=FRACT(ISOTOP-2)
80.  GO TO 11
81.  10 DIFRCT(ISOTOP,2)=1.-FRACT(ISOTOP-2)
82.  IF(J,LO.3,OR,ISO(ISOTOP-3),EG.2) GO TO 11
83.  DIFRCT(ISOTOP,3)=FRACT(ISOTOP-3)
84.  11 IF(LAMBDA(ISOTOP).LT.1.E-30) J=0
85.  12 CONTINUE
86.  13 CONTINUE
87.  C 03/25/71
88.  C INITIALIZE CHAIN CONTROL VARIABLE SKIP 03/25/71

```

```

69.      14 DO 114 I=1,CHAINS
90.      114 SKIP(I)=1
91.      C
92.      C READ TITLE AND INITIAL INVENTORY FROM DRUM OR TAPE
93.          UNIT=27
94.          IF(TAPEIN) UNIT=13
95.          CALL NTRN(UNIT,2,14,TITLE2,STATUS)
96.      214 IF(STATUS+1) 16,214,314
97.      314 CALL NTRN(UNIT,2,N1,MASS,STATUS)
98.      15 IF(STATUS+1) 16,15,18
99.      16 PRINT 17, STATUS
100.      17 FORMAT('OREAD TROUBLES IN BAFFLE - STATUS -'13)
101.      STOP
102.      18 PRINT 118, TITLE2
103.      118 FORMAT('OSOURCE TITLE - '13A6,A2)
104.      C
105.      C INITIALIZE PARAMETERS
106.          DO 218 I=1,4
107.          IFILT(I) = 0
108.          FILT(I)=1.0
109.          RELF(I)=1.0
110.      218 DEPF(I)=1.0
111.      C
112.      C READ AND PRINT DATA DESCRIBING CONTAINMENT SYSTEM TO BE MODELED
113.          READ 19, DELTAT,FREAD,NUMBER,READS,FLOWS,FALLS,IFILT
114.          19 FORMAT(F9.0,L1,8I5)
115.          PRINT 119, DELTAT,NUMBER,READS,FLOWS,FALLS,IFILT
116.      119 FORMAT('O'1PE12,4,8I5)
117.          IF(.NOT.FREAD) GO TO 2419
118.      C READ FILTER FACTOR LIBRARY WHEN FREAD = .TRUE.
119.          READ(25,1) KCD
120.          READ(25,3) ((FILTER(J,I),I=1,2),J=1,KCD)
121.          3 FORMAT(6X2F10.4)
122.          REWIND 25
123.      C
124.      C THE FOLLOWING STATEMENTS TO 1419 SET SPECIFIED FILTER FACTORS, FUEL
125.      C FACTORS AND SOURCE DEPLETION FACTORS BY NUCLIDE CLASS- NOBLE GASES,
126.      C VOLATILE SOLIDS AND ALL OTHERS.
127.      C
128.      2419 IF(IFILT(1).LT.1) GO TO 819
129.      C
130.      C READ FILTER FACTORS FOR EACH CLASS
131.          READ 919, FILT
132.          919 FORMAT(4F10.0)
133.          PRINT 1019, FILT
134.          1019 FORMAT('O'4E12,4)
135.          819 IF(IFILT(2).LT.1) GO TO 1119
136.      C
137.      C READ FUEL RELEASE FACTORS
138.          READ 919, RELF
139.          PRINT 1019, RELF
140.          1119 IF(IFILT(3).LT.1) GO TO 1219
141.      C
142.      C READ SOURCE DEPLETION FACTORS
143.          READ 919, DEPF
144.          PRINT 1019, DEPF
145.      C
146.      C IDENTIFY NUCLIDES BY CLASS
147.          DO 1619 I=1,N1
148.          DO 1519 L=1,4
149.          IF(REC(I).EQ.GAS(L)) GO TO 1719
150.          IF(REC(I).EQ.HAL(L)) GO TO 1819
151.          IF(REC(I).EQ.VOL(L)) GO TO 1919
152.      1519 CONTINUE
153.          IF(REC(I).EQ.VOL(5)) GO TO 1919
154.          IF(REC(I).EQ.VOL(6)) GO TO 1919
155.          IF(REC(I).EQ.GAS(5)) GO TO 1719
156.          IFP=4
157.          GO TO 2019
158.      1719 IFP=1
159.          GO TO 2019
160.      1819 IFP=2
161.          GO TO 2019
162.      1919 IFP=3
163.      C
164.      C JUMP11(I)=.TRUE. FOR NUCLIDES THAT ARE NOT RESUSPENDED FROM FALLOUT
165.      C FILTERS.
166.      2019 JUMP11(I)=.FALSE.
167.          IF(IFP.EQ.1) JUMP11(I)=.TRUE.
168.      C
169.      C SET FACTORS ACCORDING TO INPUT
170.          IF(IFILT(1).LT.1) GO TO 2119
171.          FILTER(I,2)=FILT(IFP)
172.      2119 IF(IFILT(2).LT.1) GO TO 2219
173.          FILTER(I,1)=RELF(IFP)
174.      2219 CONTINUE
175.          MASS(I,1)=MASS(I,1)*DEPF(IFP)
176.      1619 CONTINUE
177.      1419 CONTINUE
178.          L10=1
179.          LOSS(1,1)=DELTAT
180.          LOSS(2,1)=DELTAT*NUMBER

```

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

05/08/70

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

06/12/70

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

04/23/70

```

181. C
182. C DO NECESSARY TAPE OR DRUM HANDLING OPERATIONS FOR INPUT FILE 03/25/71
183. IF(TAPEIN) GO TO 419 03/25/71
184. REWIND UNIT
185. GO TO 719
186. 419 IF(BIN) GO TO 519
187. IF(RIN) GO TO 619
188. CALL NTRAN(UNIT,3,1)
189. GO TO 719
190. 519 CALL NTRAN(UNIT,8,-1)
191. CALL NTRAN(UNIT,7,-2)
192. GO TO 719
193. 619 CALL NTRAN(UNIT,10)
194. 719 CONTINUE
195. C
196. C INITIALIZE OUTPUT FILE 03/25/71
197. UNIT=26 03/25/71
198. IF(TAPOUT) UNIT=7
199. CALL NTRAN(UNIT,1,28,TITLE1,STATUS)
200. 219 IF(STATUS+1) 16,219,319
201. 319 CALL NTRAN(UNIT,1,2,LCSS,STATUS)
202. 20 IF(STATUS+1) 21,20,23
203. 21 PRINT 22,STATUS
204. 22 FORMAT('WRITE TROUBLES IN BAFFLE - STATUS -'13)
205. STOP
206. C
207. C INITIALIZE PARAMETERS 03/25/71
208. 23 T=0.0 03/25/71
209. T2=T+.5*DELTAT
210. IYZ=1
211. JLAST=0
212. JSTART=2
213. C
214. C ELIMINATE UNUSED CHAINS - SUBROUTINE REDUCE 03/25/71
215. CALL REDUCE(1) 03/25/71
216. IFLIP=NUMBER/10
217. IF(IFLIP.LT.1) IFLIP=1
218. IF(.NOT.BYPASS) GO TO 36
219. C
220. C DO SINGLE CONTAINMENT CALCULATION 03/25/71
221. C READ TIME AND LEAK RATE DATA PAIRS 03/25/71
222. READ 123, (TIME(1),RATE(1,1,1),I=1,READS) 03/25/71
223. 123 FORMAT(8F10.0)
224. PRINT 223, (TIME(1),RATE(1,1,1),I=1,READS)
225. 223 FORMAT(1P3E12.4)
226. K246=NUMBER+1
227. C
228. C CALCULATE NUCLIDE RELEASE FOR EACH TIME INCREMENT 03/25/71
229. DO 32 I=1,K246 03/25/71
230. C INTERPOLATE RELEASE RATE DATA 03/25/71
231. CALL LINEAR(RELEAS,RATE,I,TIME,LIN,READS,$34) 04/23/70
232. 24 IF(STATUS+1) 21,24,25
233. C
234. C CALCULATE NUCLIDE RELEASE RATE AT CURRENT TIME 03/25/71
235. 25 DO 26 J=1,N1 03/25/71
236. 26 QUANTY(J)=PASS(J,1)*RELEAS*FILTER(J,1)
237. CALL NTRAN(UNIT,1,N1,QUANTY,STATUS)
238. IF(MOD(I-1,IFLIP).NE.0) GO TO 30
239. C
240. C PRINT CURRENT NUCLIDE RELEASE RATE AS CURIES/SEC. 03/25/71
241. PRINT 127,T 03/25/71
242. 127 FORMAT('1AT TIME =1PE10.4,' SECONDS, THE STACK RELEASE IS '/0ISO
243. 1TOPE10X,CURIES/SECOND')
244. 27 FORMAT('1AT TIME =1PE10.4,' SECONDS,'/' THE STACK RELEASE IS'20X,
245. X'COMPARTMENT CONTENTS ARE (CURIES)'/0ISOISOTOPE'1X,'CURIES/SECOND'
246. X7,'FUEL'2X,'BARRIER'1'4X,'PLATE'1'3X,'FILTER'1'2X,'BARRIER'2'4X,
247. X'PLATE'2'3X,'FILTER'2'2X,'BARRIER'3'4X,'PLATE'3'3X,'FILTER'3'1)
248. DO 29 K=1,N1
249. ABLE=QUANTY(K)*LAMBDA(K)
250. IF(ABLE.LT.1.E-30) GO TO 29
251. PRINT 28, REC(K),NL(K),ABLE
252. 28 FORMAT(1X23,13.4X,1P11E11.4)
253. 29 CONTINUE
254. 30 IF(I.EQ.K246) GO TO 32
255. C
256. C CALCULATE NUCLIDE RELEASE RATE AT MIDDLE OF CURRENT TIME INTERVAL 03/25/71
257. CALL LINEAR(RELEAS,RATE,T2,TIME,LIN,READS,$34) 03/25/71
258. C
259. C CALCULATE EFFECTIVE REMOVAL RATE CONSTANT FOR EACH NUCLIDE 04/23/70
260. DO 31 K=1,N1 03/25/71
261. 31 LOSS(K,1)=RELEAS*FILTER(K,1)+LAMBDA(K) 03/25/71
262. C
263. C DECA/ NUCLIDES ONE TIME INCREMENT USING EFFECTIVE REMOVAL RATE CONSTA 03/25/71
264. CALL CHAIN1 03/25/71
265. DO 131 K=1,N1
266. IF(MASS(K,1).LT.1.E-30) GO TO 131
267. MASS(K,1)=PASS(K,1)*EXP(-LOSS(K,1)*DELTAT)
268. 131 CONTINUE
269. C ELIMINATE UNUSED NUCLIDE CHAINS 03/25/71
270. CALL REDUCE(1)
271. C INCREMENT TIMES 03/25/71
272. T=T+DELTAT

```

```

273.      T2=T2+DELTAT
274.      32 CONTINUE
275.      C END OF DO LOOP ON TIME INCREMENTS
276.      C
277.      33 IF(STATUS+1)-21,38,133
278.      C
279.      C DO NECESSARY TAPE OR DRUM HANDLING OPERATIONS AND RETURN TO MAIN
280.      133 IF(TAPCUT) GO TO 333
281.      REWIND UNIT
282.      233 RETURN
283.      353 CALL NTRAN(UNIT,9)
284.      IF(BOUT) GO TO 433
285.      IF(RCUT) GO TO 533
286.      GO TO 233
287.      433 CALL NTRAN(UNIT,8,-1)
288.      CALL NTRAN(UNIT,7,-(NUMBER+3))
289.      GO TO 233
290.      533 CALL NTRAN(UNIT,10)
291.      GO TO 233
292.      C
293.      C WHEN RELEASE RATE DATA DOES NOT COVER LONG ENOUGH TIME PERIOD PRINT
294.      C AND RETURN TO MAIN
295.      34 T2=T+DELTAT
296.      PRINT 35, T2
297.      35 FORMAT('0TIME LIBRARY FOR RELEASE TOO SHORT, CURRENT TIME IS '1PE1
298.      10.4, ' SECONDS')
299.      GO TO 33
300.      C
301.      C
302.      C MULTIPLE CONTAINMENT CALCULATION
303.      C
304.      C INITIALIZE PARAMETERS
305.      36 DO 37 I=1,6
306.      JUMP(I)=.TRUE.
307.      DO 37 J=1,7
308.      LEAK(J,I)=.FALSE.
309.      DO 37 K=1,50
310.      37 RATE(K,J,I)=0.0
311.      JUMP(9)=.TRUE.
312.      JUMP(10)=.TRUE.
313.      DO 137 I=1,3
314.      SAVE(I)=0.0
315.      FALCUT(I)=.FALSE.
316.      DO 137 J=1,50
317.      137 FRATES(J,I)=0.0
318.      DO 237 I=1,N1
319.      237 QUANTY(I)=0.0
320.      C
321.      C READ TIMES FOR RELEASE RATE DATA
322.      READ 123, (TIME(I),I=1,READS)
323.      PRINT 223, (TIME(I),I=1,READS)
324.      C
325.      C READ RELEASE RATE DATA
326.      DO 39 I=1,FLOWS
327.      READ 38, L,M
328.      38 FORMAT(2I1)
329.      LEAK(L,M)=.TRUE.
330.      PRINT 138, L,M
331.      138 FORMAT(' FROM '11, ' TO '11)
332.      READ 123, (RATE(J,L,M),J=1,READS)
333.      PRINT 223, (RATE(J,L,M),J=1,READS)
334.      C IDENTIFY RELEASE RATES FROM FILTERS TO NEXT CONTAINER AND SET LOGICAL
335.      C VARIABLES ACCORDINGLY
336.      IF(M.GT.2.AND.M.LT.0) JUMP(M)=.FALSE.
337.      IF(L.EQ.3.AND.M.EQ.4) JUMP(9)=.FALSE.
338.      IF(L.EQ.5.AND.M.EQ.6) JUMP(10)=.FALSE.
339.      39 CONTINUE
340.      IF(FALLS.EQ.0) GO TO 41
341.      C
342.      C READ FALLOUT RATE DATA
343.      DO 40 I=1,FALLS
344.      READ 38, L
345.      PRINT 139, L
346.      FALCUT(L)=.TRUE.
347.      139 FORMAT(' FALL CUT IN CONTAINER '12)
348.      READ 123, (FRATES(J,L),J=1,READS)
349.      PRINT 223, (FRATES(J,L),J=1,READS)
350.      IF(L.EQ.3) L=3
351.      JUMP(L)=.FALSE.
352.      40 CONTINUE
353.      C
354.      C INITIALIZE STORAGE ARRAY FOR NUCLIDE MASSES
355.      41 DO 42 L=2,10
356.      DO 42 K=1,N1
357.      42 MASS(K,L)=0.0
358.      DO 43 I=1,1440
359.      43 H(I)=0.0
360.      LOOP=0
361.      IF(RATE(1,1,8).LT.1.E-30) GO TO 45
362.      C
363.      C CALCULATE AND PRINT INITIAL NUCLIDE RELEASE RATES
364.      DO 44 I=1,N1

```

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

03/25/71

```

365.      44 QUANTITY(I)=MASS(I,1)*FILTER(I,1)*RATE(1,1,8)
366.
367. C STARTING POINT FOR CALCULATION AT CURRENT TIME
368.      144 IF(IFILT(4).GT.0) GO TO 244
369.      IF(MOD(LOOP,IFLIP).NE.0) GO TO 45
370.      244 PRINT 27,I
371.          DO 347 K=1,N1
372.              ADD=0.
373.              DO 1347 ID=1,10
374.                  CONTEN(ID)=MASS(K,ID)*LAMBDA(K)
375.      1347 ADD=ADD+CONTEN(ID)
376.              ABLE=QUANTITY(K)*LAMBDA(K)
377.              ADD=ADD+ABLE
378.              IF(ADD.EQ.0.) GO TO 347
379.              PRINT 23,REC(K),XL(K),ABLE,CONTEN
380.      347 CONTINUE
381.      45 IF(STATUS+1) 21,45,46
382.
383. C WRITE CURRENT NUCLIDE RELEASE RATES TO TAPE OR DRUM
384.      46 DO 47 I=1,N1
385.      47 OUT(I)=QUANTITY(I)
386.          CALL NTRAN(UNIT,1,N1,OUT,STATUS)
387.          IF(LOOP.EQ.NUMBER) GO TO 33
388.          DO 49 L=1,7
389.              M1=L+1
390.
391. C INTERPOLATE LEAK RATE DATA FOR CURRENT TIME
392.      DO 48 M=1,8
393.          IF(.NOT.LEAK(L,M)) GO TO 48
394.          JLAST=0
395.          CALL LINEAR(ILR(L,M),RATE(1,L,M),T2,TIME,LIN,READS,534)
396.          IF(M.EQ.8) ALR(L)=ILR(L,8)
397.      48 CONTINUE
398.      49 CONTINUE
399.
400. C CALCULATE TOTAL GUT LEAK RATE FOR EACH COMPARTMENT
401.      DO 50 L=1,7
402.          M1=L+1
403.          OLR(L)=0.0
404.          DO 50 K=M1,8
405.      50 OLR(L)=OLR(L)+ILR(L,K)
406.          IF(FALLS.EQ.0) GO TO 251
407.
408. C DO CONTAINMENT CALCULATION FOR EACH MEMBER OF EACH NUCLIDE CHAIN FOR
409. C THERE ARE NUCLIDES PRESENT.
410.      DO 151 I=1,3
411.          IF(.NOT.FALCUT(I)) GO TO 151
412.          JLAST=0
413.          CALL LINEAR(SAVE(I),FRATES(1,I),T2,TIME,LIN,READS,534)
414.      151 CONTINUE
415.      251 ISOTOP=0
416.          ADD=.01/DELTAT
417.          DO 1258 IJ=1,3
418.              DO 1257 K=1,3
419.                  TEST=SAVE(K)/ADD
420.                  IF(TEST.LT..99) GO TO 2251
421.                  IF(TEST.GT.1.01) GO TO 2251
422.                  SAVE(K)=SAVE(K)*.98
423.      2251 I=2*K
424.                  IF((OLR(I)*DELTAT).LT.1.E-8) GO TO 1257
425.                  IJ=1
426.                  DO 1256 J=1,K
427.                      TEST=(OLR(I)-OLR(IJ))*DELTAT
428.                      IF(TEST.GE.0.) GO TO 1251
429.                      TEST=-TEST
430.                      IF(TEST.LT..001) OLR(IJ)=OLR(IJ)+ADD
431.                      GO TO 1252
432.      1251 IF(TEST.LT..001) OLR(I)=OLR(I)+ADD
433.      1252 IF(SAVE(K).LT.1.E-10) GO TO 1254
434.                      TEST=(OLR(I)+SAVE(K)-OLR(IJ))*DELTAT
435.                      IF(TEST.GE.0.) GO TO 1253
436.                      TEST=-TEST
437.                      IF(TEST.LT..001) OLR(IJ)=OLR(IJ)+ADD
438.                      GO TO 1254
439.      1253 IF(TEST.LT..001) OLR(I)=OLR(I)+ADD
440.      1254 IF(IJ.LT.2) GO TO 1256
441.                      TEST=(OLR(I)+SAVE(K)-OLR(IJ)-SAVE(J-1))*DELTAT
442.                      IF(TEST.GE.0.) GO TO 1255
443.                      TEST=-TEST
444.                      IF(TEST.LT..001) OLR(IJ)=OLR(IJ)+ADD
445.                      GO TO 1255
446.      1255 IF(TEST.LT..001) OLR(I)=OLR(I)+ADD
447.      1256 IJ=IJ*2
448.      1257 CONTINUE
449.      1258 CONTINUE
450.      DO 62 K=1,CHAINS
451.          LIM=ROFFHSC(K)
452.          IF(SKIP(K).EQ.0) GO TO 61
453.          J=0
454.          I=SKIP(K)
455.          ISOTOP=ISOTOP+I1-1
456.          DO 160 I1=LIM

```

03/25/71
03/25/71

03/25/71
03/25/71

03/25/71
03/25/71

04/23/70

03/25/71
03/25/71

03/25/71
03/25/71
03/25/71

04/23/70

```

457. ISOTOP=ISOTOP+1
458. IP(LAMBDA(ISOTOP).LT.1.E-10) GO TO 59
459. J=J+1
460. JUMPUP=JUMP11(ISOTOP)
461. IF(FILTER(ISOTOP,2).EQ.1.) GO TO 451
462. JUMPOFF=FALSE
463. C SET FALLOUT RATE ACCORDING TO LOGICAL VARIABLE JUMPUP=JUMP11(ISOTOP) 03/25/71
464. IF(JUMPUP) GO TO 551
465. DO 351 LM=1,3
466. 351 FOR(LM)=SAVE(LM)
467. GO TO 651
468. 451 JUMPOFF=TRUE.
469. 551 DO 751 LM=1,3
470. 751 FOR(LM)=0.0
471. 651 CONTINUE
472. GO TO (52,53,54,55,56,57,58),J
473. 52 CALL FIRST
474. GO TO 60
475. 53 CALL SECOND
476. GO TO 60
477. 54 CALL THIRD
478. GO TO 60
479. 55 CALL FOURTH
480. GO TO 60
481. 56 CALL FIFTH
482. GO TO 60
483. 57 CALL SIXTH
484. GO TO 60
485. 58 CALL LAST
486. GO TO 60
487. 59 J=0
488. 60 CONTINUE
489. ADD=0.
490. KUP=5*J
491. DO 260 ID=1,6
492. IF(MASS(ISOTOP,ID).LT.0.) ADD=1.
493. 260 CONTINUE
494. IF(ADD.EQ.0.) GO TO 160
495. PRINT 360, REC(ISOTOP),ML(ISOTOP), (IE,EXPO(IE),EEXPO(IE),IE=1,KUP)
496. 360 FORMAT(1H1,A3,I3,2X,' K',5X,'EXPO(K)',7X,'EEXPO(K)'/
497. X(9X,I2,1P2E15.7))
498. 160 CONTINUE
499. GO TO 62
500. 61 ISOTOP=ISOTOP+LIM
501. 62 CONTINUE
502. C
503. C DO CONTAINMENT CALCULATION FOR EACH ACTIVATION PRODUCT 03/25/71
504. DO 65 KEN,K1 03/25/71
505. IF(MASS(K,1Y2).LT.1.E-30) GO TO 65
506. EXPO(1)=LAMBDA(K)+OLR(1)
507. EEXPO(1)=EXP(-EXPO(1)*DELTAT)
508. A(1)=MASS(K,1)
509. MASS(K,1)=A(1)*EEXPO(1)
510. EXPO(2)=LAMBDA(K)+OLR(2)
511. IF(FILTER(K,2).NE.1.) EXPO(2)=EXPO(2)+SAVE(1)
512. EEXPO(2)=EXP(-EXPO(2)*DELTAT)
513. A(2)=(ILR(1,2)*A(1)/(EXPO(2)-EXPO(1)))
514. A(3)=MASS(K,2)-A(2)
515. MASS(K,2)=A(3)*EEXPO(2)+A(2)*EEXPO(1)
516. IF(JUMP(4)) GO TO 63
517. EXPO(3)=LAMBDA(K)+OLR(4)
518. IF(FILTER(K,2).NE.1.) EXPO(3)=EXPO(3)+SAVE(2)
519. EEXPO(3)=EXP(-EXPO(3)*DELTAT)
520. ABLE=ILR(2,4)+ILR(3,4)*FILTER(K,2)
521. A(4)=(ILR(1,4)*A(1)+ABLE*A(2))/(EXPO(3)-EXPO(1))
522. A(5)=ABLE-A(3)/(EXPO(3)-EXPO(2))
523. A(6)=MASS(K,5)-A(4)-A(5)
524. MASS(K,5)=A(6)*EEXPO(3)+A(5)*EEXPO(2)+A(4)*EEXPO(1)
525. 63 IF(JUMP(6)) GO TO 64
526. EXPO(4)=LAMBDA(K)+OLR(6)
527. IF(FILTER(K,2).NE.1.) EXPO(4)=EXPO(4)+SAVE(3)
528. EEXPO(4)=EXP(-EXPO(4)*DELTAT)
529. ABLE=ILR(4,6)+ILR(5,6)*FILTER(K,2)
530. A(7)=(ILR(1,6)*A(1)+ABLE*A(4))/(EXPO(4)-EXPO(1))
531. A(8)=ABLE*A(5)/(EXPO(4)-EXPO(2))
532. A(9)=ABLE*A(5)/(EXPO(4)-EXPO(3))
533. A(10)=MASS(K,8)-A(7)-A(8)-A(9)
534. MASS(K,8)=A(10)*EEXPO(4)+A(9)*EEXPO(3)+A(8)*EEXPO(2)+A(7)*EEXPO(1)
535. 64 QUANTY(K)=ALR(1)*MASS(K,1)+(ALR(2)+ALR(3)*FILTER(K,2))*MASS(K,2)+
536. 1(ALR(4)+ALR(5)*FILTER(K,2))*MASS(K,5)+(ALR(6)+ALR(7)*FILTER(K,2))*
537. 2MASS(K,8)
538. 65 CONTINUE
539. IYZ=2
540. C INCREMENT CALCULATION 03/25/71
541. LOOP=LOOP+1
542. T=T+DELTAT
543. T2=T2+DELTAT
544. GO TO 144
545. END

```



```

1. C SUBROUTINE EXDOSE PERFORMS CLOUD DOSE CALCULATION FOR NUCLIDES RELEASED 03/25/71
2. C 03/25/71
3. SUBROUTINE EXDOSE
4. DIMENSION NFULL(5,6),JUMP(43),SUM(3,43),DOSEX(43,5),DOSE(5) 08/04/70
5. COMMON/LOGIC/LIBRARY,TAPE,BACKUP,REV,IND,SEMI
6. COMMON NCS,CAPA(16),ALFA(16),ALFA1(16),MU(16),MEV(16),NOFNUC(96),
7. 1SKIP(96),MASS(600),LAMBDA(600),CHAINS,OKFRCT(450,3),N1,N2,OLDDT,
8. 2BURST(600,16),DRDX(43,16),TIME,NXINT,T(43),X(43),TEXPOS(5),GROUPS,08/04/70
9. 3NEXPO,TRELES,UBAR,UELD(6),DELT(6),TAPTIM,INTERP 08/04/70
10. COMMON RTIME,OLD(86),NOT,DK 01/25/71
11. LOGICAL DK 01/25/71
12. COMMON/TITLE/TITLE1(14),TITLE2(14),TITLE3(14)
13. LOGICAL LIBRARY,TAPE,BACKUP,REV,IND,INTERP,THRU,SEMI
14. LOGICAL JUMP,BLANK,ENSKIP,ACTPRD,FISPRD 08/04/70
15. INTEGER ENUM,EN 08/04/70
16. DIMENSION ENSKIP(16),ENUM(16) 08/04/70
17. INTEGER SKIP,UNIT,STATUS,CHAINS,POSIT,ENERGY,GROUPS
18. REAL MASS,MEV,MU
19. REAL LAMBDA 07/16/70
20. IF(.NOT.LIBRARY) GO TO 1
21. C 03/25/71
22. C READ NUCLIDE DATA LIBRARY AND PHOTON DATA LIBRARY - CALCULATE BURST 103/25/71
23. CALL LIB
24. IGROUP=GROUPS 08/04/70
25. C READ SOURCE AND BAFFLE TITLES FROM TAPE OR DRUM 03/25/71
26. 1 UNIT=26
27. IF(TAPE) UNIT=7
28. CALL NTRAX(UNIT,2,28,TITLE2,STATUS)
29. 2 IF(STATUS+1) 3,2,5
30. 3 PRINT 4, STATUS
31. 4 FORMAT(' READ TROUBLES IN EXDOSE'14)
32. GO TO 13
33. C PRINT TITLES FOR SOURCE AND BAFFLE
34. 5 PRINT 6, TITLE3,TITLE2
35. 6 FORMAT(' SOURCE TITLE - '13A6,A2/' BAFFLE TITLE - '13A6,A2)
36. C READ TIME INCREMENT AND TOTAL TIME FROM TAPE OR DRUM 03/25/71
37. CALL NTRAX(UNIT,2,2,MASS,STATUS)
38. STATUS=STATUS
39. 7 IF(STATUS+1) 3,7,8
40. 8 OLDDT=MASS(1)
41. TAPTIM=MASS(2)
42. C 03/25/71
43. C INITIALIZE NUCLIDE MASS ARRAY 03/25/71
44. DO 9 I=1,N2
45. MASS(I)=0.0
46. 9 CONTINUE
47. GROUPS=IGROUP 08/19/70
48. IF(SEMI) GO TO 1000
49. C 03/25/71
50. C START OF FINITE CLOUD CALCULATION 03/25/71
51. C 03/25/71
52. C INITIALIZE PARAMETERS 03/25/71
53. DO 11 I=1,5
54. DO 10 J=1,6 08/04/70
55. NFULL(I,J)=0
56. 10 CONTINUE
57. 11 CONTINUE
58. DO 12 I=1,43 08/04/70
59. JUMP(I)=.FALSE.
60. DO 125 J=1,5 06/03/70
61. DOSEX(I,J)=0.0 06/03/70
62. 125 CONTINUE 06/03/70
63. 12 CONTINUE
64. C 03/25/71
65. C CALL SUBROUTINE PLUME TO READ METEOROLOGICAL PARAMETERS AND CALCULATE 03/25/71
66. C RATE PER CURIE PER CM OF DOWNWIND CLOUD AT EACH X GRID POSITION - DRD 03/25/71
67. CALL PLUME(513,5245) 09/15/70
68. GO TO 15
69. C ERROR RETURN FROM PLUME - RETURN 0 TO STOP CALCULATION 03/25/71
70. 13 RETURN 0
71. C NORMAL RETURN TO MAIN 03/25/71
72. 14 RETURN
73. C SET CONTROL VARIABLES
74. 15 THRU=.FALSE.
75. ITIME=1
76. RTIME=0.0
77. TT=T(1)
78. BLANK=.TRUE.
79. DO 16 I=1,CHAINS
80. SKIP(I)=0
81. 16 CONTINUE
82. C TIME IS TIME TO POSITION ONE PLUS RTIME.
83. C RTIME IS TIME MATERIAL HAS BEEN RELEASED.
84. TIME=TT(1)
85. C START CALCULATION FOR CURRENT CLOUD PUFF
86. 17 CONTINUE
87. THRU=.FALSE. 06/03/70
88. C LOGICAL VARIABLE THRU IS SET TRUE WHEN RELEASE TIME IS REACHED 03/25/71

```

```

89.      IF (TRELES.LT.RTIME) THRU=.TRUE.
90.      NPOSIT=7*NXINT+1
91.      I,C=1
92.
93.      C DO LOOP ON IX=NUMBER OF MAJOR X GRID SEGMENTS (SEVEN MINOR SEGMENTS 03/25/71
94.      C MAJOR SEGMENT), MAXIMUM OF SIX MAJOR SEGMENTS P03/25/71
95.      DO 39 IX=1,NXINT 03/25/71
96.      IF (IX.GT.1) INC=2
97.
98.      C DO LOOP ON IPOSIT=NUMBER OF MINOR X-POSITION WITHIN MAJOR SEGMENT IX 03/25/71
99.      DO 38 IPOSIT=INC,8
100.      POSIT=(IX-1)*7+IPOSIT
101.      C JUMP(POSIT) IS TRUE WHEN CLOUD HAS PASSED POSITION POSIT - SKIP INPUT 03/25/71
102.      IF (JUMP(POSIT)) GO TO 40
103.      IF (POSIT.GT.1) GO TO 30
104.      C READ DATA FOR NEXT PUFF
105.      UNIT=26 05/19/70
106.      IF (TAPE) UNIT=7 05/19/70
107.      N=NDT 05/19/70
108.      IF (N.LT.1) N=1 10/22/70
109.      IF (ITIME.LE.1) N=1 10/22/70
110.      DO 118 NR=1,N 10/22/70
111.      CALL NTRAN(UNIT,2,N2,MASS(1),STATUS) 10/22/70
112.      STATUS=STATUS
113.      18 IF (STATUS+1) 3,18,118 10/22/70
114.      118 CONTINUE 10/22/70
115.      C
116.      C DETERMINE WHICH CHAINS MAY BE IGNORED - SKIP(K).NE.0 FOR BLANK CHAIN 03/25/71
117.      C FISPRD=.TRUE. IF FISSION PRODUCTS ARE PRESENT 03/25/71
118.      C ACTPRD=.TRUE. IF ACTIVATION PRODUCTS ARE PRESENT 03/25/71
119.      19 IF (.NOT.BLANK) GO TO 21
120.      DO 20 K=1,CHAINS
121.      SKIP(K)=0
122.      20 CONTINUE
123.      21 BLANK=.TRUE.
124.      ACTPRD=.FALSE. 08/04/70
125.      FISPRD=.FALSE. 08/04/70
126.      ISOTOP=0
127.      DO 25 K=1,CHAINS
128.      LIM=HOFNUC(K)
129.      IF (SKIP(K).NE.0) GO TO 24
130.      DO 22 I=1,LIM
131.      L=ISOTOP+I
132.      IF (MASS(L).GT.1.E-30.AND.LAMBDA(L).GT.1.E-30) GO TO 23
133.      22 CONTINUE
134.      SKIP(K)=LIM
135.      GO TO 24
136.      23 BLANK=.FALSE.
137.      FISPRD=.TRUE. 08/04/70
138.      24 ISOTOP=ISOTOP+LIM
139.      25 CONTINUE
140.      DO 225 L=1,N2
141.      IF (MASS(L).GT.1.E-30.AND.LAMBDA(L).GT.1.E-30) GO TO 325
142.      225 CONTINUE
143.      GO TO 425
144.      325 BLANK=.FALSE.
145.      ACTPRD=.TRUE. 08/04/70
146.      425 IF (.NOT.BLANK) GO TO 31
147.      IF (ITIME.LE.1) GO TO 31
148.      C NO ACTIVITY IN PUFF. RE-INITIALIZE CALCULATION TO IGNORE BLANK PUFF.
149.      TIME = TIME + OLCOT/NDT 10/22/70
150.      RTIME = RTIME + OLCOT/NDT 10/22/70
151.      IF (RTIME.LT.TRELES) GO TO 27 11/18/70
152.      C 03/25/71
153.      C RELEASE TIME IS TOO SHORT BECAUSE OF BLANK PUFFS AT START - STOP CALC 03/25/71
154.      PRINT 26, RTIME 11/18/70
155.      26 FORMAT('RELEASE TIME TOO SHORT BECAUSE OF BLANK PUFFS AT START - 11/18/70
156.      1RTIME S'F10.1) 11/18/70
157.      GO TO 245 11/18/70
158.      27 IF (TIME.LT.TEXPOS(1)) GO TO 17
159.      C
160.      C EXPOSURE TIME 1 IS TOO SHORT - ELIMINATE 1 AND MOVE OTHER TIMES UP 03/25/71
161.      PRINT 23, TEXPOS(1) 03/25/71
162.      28 FORMAT('EXPOSURE TIME OF 1PE10.2, SECONDS IS TOO SHORT - SKIP')
163.      IF (NEXPO.EQ.1) GO TO 14
164.      K=NEXPO-1
165.      DO 29 I=1,K
166.      TEXPOS(I)=TEXPOS(I+1)
167.      29 CONTINUE
168.      NEXPO=NEXPO-1
169.      GO TO 27
170.      C DECAY INVENTORY BEYOND POSITION ONE.
171.      30 IF (DK) GO TO 54 01/25/71
172.      IF (FISPRD) CALL CHAIN(DELT(IX)) 01/25/71
173.      IF (.NOT.ACTPRD) GO TO 54 01/25/71
174.      DO 51 I=N1,N2
175.      IF (MASS(I).LT.1.E-30) GO TO 51
176.      MASS(I)=MASS(I)*EXP(-LAMBDA(I)*DELT(IX))
177.      51 CONTINUE
178.      GO TO 54
179.      C DECAY NEW PUFF TO FIRST POSITION 05/19/70
180.      31 IF (DK) GO TO 1053 01/25/71

```

```

181.      IF(FISPRD) CALL CHAIN(T(1))
182.      IF(.NOT.ACTPRD) GO TO 1053
183.      DO 53 I=N1,N2
184.      IF(MASS(I).LT.1.E-30) GO TO 53
185.      MASS(I)=MASS(I)*EXP(-LAMBDA(I)*T(1))
186.      53 CONTINUE
187.      1053 CONTINUE
188.
189.      C
190.      C DETERMINE WHICH ENERGY GROUPS TO CONSIDER
191.      C
192.      DO 153 I=1,16
193.      ENSKIP(I)=.TRUE.
194.      ENUM(I)=1
195.      153 CONTINUE
196.      GROUPS=IGROUP
197.      IF(.NOT.FISPRD) GO TO 953
198.      ISOTOP=0
199.      DO 453 K=1,CHAINS
200.      LIM=NOFNUC(K)
201.      IF(SKIP(K).NE.0) GO TO 1153
202.      DO 353 I=1,LIM
203.      L=ISOTOP+I
204.      IF(MASS(L).LT.1.E-30.OR.LAMBDA(L).LT.1.E-30) GO TO 353
205.      DO 253 EN=1,GROUPS
206.      IF(BURST(L,EN).GT.1.E-30) ENSKIP(EN)=.FALSE.
207.      253 CONTINUE
208.      353 CONTINUE
209.      1153 ISOTOP=ISOTOP+LIM
210.      453 CONTINUE
211.      953 IF(.NOT.ACTPRD) GO TO 853
212.      DO 753 K=N1,N2
213.      IF(MASS(K).LT.1.E-30.OR.LAMBDA(K).LT.1.E-30) GO TO 753
214.      DO 653 EN=1,GROUPS
215.      IF(BURST(K,EN).GT.1.E-30) ENSKIP(EN)=.FALSE.
216.      653 CONTINUE
217.      753 CONTINUE
218.      853 K=1
219.      DO 553 I=1,GROUPS
220.      IF(ENSKIP(I)) GO TO 553
221.      ENUM(K)=I
222.      K=K+1
223.      553 CONTINUE
224.      GROUPS=K-1
225.
226.      C
227.      C CALCULATE FOR EACH ENERGY GROUP SUM(I,IX) WHERE I IS TIME INCREMENT
228.      C
229.      54 SUM(ITIME,POSIT)=0.0
230.      ISOTOP=0
231.      IF(.NOT.FISPRD) GO TO 434
232.      DO 34 K=1,CHAINS
233.      LIM=NOFNUC(K)
234.      IF(SKIP(K).NE.0) GO TO 50
235.      DO 33 I=1,LIM
236.      ISO=ISOTOP+I
237.      SUM1=0.0
238.      DO 32 IEN=1,GROUPS
239.      ENERGY=ENUM(IEN)
240.      SUM1=SUM1+DOEX(POSIT,ENERGY)*BURST(ISO,ENERGY)
241.      32 CONTINUE
242.      SUM(ITIME,POSIT)=SUM(ITIME,POSIT)+SUM1*MASS(ISO)
243.      33 CONTINUE
244.      50 ISOTOP=ISOTOP+LIM
245.      34 CONTINUE
246.      434 IF(.NOT.ACTPRD) GO TO 534
247.      DO 234 ISO=N1,N2
248.      IF(MASS(ISO).LT.1.E-30) GO TO 234
249.      SUM1=0.
250.      DO 134 IEN=1,GROUPS
251.      ENERGY=ENUM(IEN)
252.      SUM1=SUM1+DOEX(POSIT,ENERGY)*BURST(ISO,ENERGY)
253.      134 CONTINUE
254.      SUM(ITIME,POSIT)=SUM(ITIME,POSIT)+SUM1*MASS(ISO)
255.      234 CONTINUE
256.      534 CONTINUE
257.
258.      C CALCULATE TIME INTEGRALS WHEN NECESSARY
259.      C
260.      TPOS=T(POSIT)+RTIME
261.      DO 37 IEX=1,NEXPO
262.      IF(TPOS.GE.TEXPOS(IEX)) GO TO 36
263.      IF(.NOT.THRU) GO TO 37
264.      DOSEDX(POSIT,IEX)=DOSEDX(POSIT,NEXPO)+OLDDT*FINT(SUM(1,POSIT),ITIME
265.      1E)
266.      IF(IPOSIT.GT.NFULL(IEX,IX)) NFULL(IEX,IX)=IPOSIT
267.      GO TO 37
268.      35 DOSEA=0.0
269.      ITM1=ITIME-1
270.      IF(ITIME.GT.2) DOSEA=OLDDT*FINT(SUM(1,POSIT),ITM1)
271.      SUM=SUM(ITIME,POSIT)-SUM(ITIME-1,POSIT)
272.      DOSEC=SUM(ITIME-1,POSIT)*(OLDDT-TPATE)
273.      DOSED=0.5*((OLDDT-TPATE)**2)*SUM/OLDDT
274.      DOSED=DOSED+DOSEC
275.      DOSEDX(POSIT,IEX)=DOSEDX(POSIT,NEXPO)+DOSEA+DOSED
276.      IF(IPOSIT.GT.NFULL(IEX,IX)) NFULL(IEX,IX)=IPOSIT

```

```

273.      GO TO 37
274.      36 IF(itime,eq.1) GO TO 37
275.      tpmte=tpos-texpos(iex)
276.      IF(tpmte,ge.olddt) GO TO 37
277.      IF(iex,eq,nexpo) JUMP(POSIT)=.TRUE.
278.      GO TO 35
279.      37 CONTINUE
280.      C IF 8 POSITIONS ARE FILLED INTEGRATE. (IF THRU=.TRUE. SKIP INTEGRAL.
281.      IF(itime,ne.8) GO TO 38
282.      IF(thru) GO TO 38
283.      DOSEDX(POSIT,NEXP0)=DOSEDX(POSIT,NEXP0)+OLDDT*FINT(SUM(1,POSIT),8)
284.      SUM(1,POSIT)=SUM(8,POSIT)
285.      38 CONTINUE
286.      39 CONTINUE
287.      C SET VARIABLES FOR NEXT TIME PERIOD
288.      40 IF(itime,eq.8) itime=1
289.      itime=itime+1
290.      TIME=TIME+CLDDT
291.      RTIME=RTIME+OLDDT
292.      IF(.NOT.THru) GO TO 17
293.      C PERFORM INTEGRATION IN X-DIRECTION
294.      DO 41 I=1,5
295.      DOSE(I)=0.0
296.      TEXPOS(I)=TEXPOS(I)/3600.
297.      41 CONTINUE
298.      DO 45 IEX=1,NEXP0
299.      DO 42 IX=1,NXINT
300.      K=NFULL(IEX,IX)
301.      IF(K,LE.1) GO TO 43
302.      POSIT=(IX-1)*7+1
303.      DOSE(IEX)=DOSE(IEX)+DELD(IX)*FINT(DOSEDX(POSIT,IEX),K)
304.      42 CONTINUE
305.      C
306.      C PRINT RESULTS
307.      43 PRINT 44, TEXPOS(IEX),DOSE(IEX)
308.      44 FORMAT('OAT EXPOSURE TIME = 'F7.2,' HOURS ACCUMULATED DOSE IS '1PE
309.      '110.3,' RADS')
310.      45 CONTINUE
311.      C
312.      C PERFORM NECESSARY TAPE OR DRUM HANDLING OPERATIONS
313.      245 IF(TAPE) GO TO 60
314.      REWIND UNIT
315.      GO TO 61
316.      60 IF(BACKUP) GO TO 63
317.      IF(REWIND) GO TO 62
318.      CALL NTRAN(UNIT,8,1)
319.      61 GO TO 14
320.      62 CALL NTRAN(UNIT,10)
321.      GO TO 14
322.      63 N=-(TAPTIM/OLDDT+3)
323.      CALL NTRA(UNIT,7,N)
324.      C RETURN TO MAIN PROGRAM
325.      GO TO 14
326.      C
327.      C SEMI INFINITE CLOUD DOSE CALCULATION
328.      C
329.      C SEMME CALCULATES DOSE RATE PER CURIE FOR EACH ENERGY GROUP, DRDX(1,EN03/25/71
330.      1000 CALL SEMME
331.      C
332.      C INITIALIZE PARAMETERS
333.      BLANK=.TRUE.
334.      I2=1
335.      IF(TRELES,GT,TAPTIM) TRELES=TAPTIM
336.      TIME=T(1)
337.      OTIME=T(1)
338.      RTIME=1
339.      RTIME=0.0
340.      DO 1001 K=1,CHAINS
341.      1001 SKIP(K)=0
342.      GO TO 1003
343.      C TEST TO SEE IF RELEASE TIME HAS BEEN REACHED
344.      C START OF CALCULATION FOR EACH TIME INCREMENT
345.      1007 IF(RTIME,GT,TRELES) GO TO 1026
346.      C TEST TO SEE IF CURRENT TIME IS GREATER THAN EXPOSURE TIME PLUS ONE
347.      IF(TIME,GT,TEXPOS(1)+DELT(1)) GO TO 1026
348.      1008 IF(.NOT.INTERP) GO TO 1009
349.      C INTERPOLATE INPUT DATA FOR CURRENT TIME
350.      CALL FIT(12)
351.      GO TO 1012
352.      1009 CALL NTRAN(UNIT,2,N2,MASS(1),STATUS)
353.      1010 IF(STATUS+1) 3,1010,1012
354.      C
355.      C DETERMINE WHICH NUCLIDE CHAINS MAY BE SKIPPED
356.      1012 IF(.NOT.BLANK) GO TO 1112
357.      DO 1212 K=1,CHAINS
358.      SKIP(K)=0
359.      1212 CONTINUE
360.      1112 BLANK=.TRUE.
361.      ACTPR=.FALSE.
362.      FISPR=.FALSE.
363.      ISOTOP=0
364.      DO 1214 K=1,CHAINS

```

05/19/70
05/19/70

03/25/71
03/25/71

03/25/71
03/25/71
03/15/70
06/03/70
06/03/70
06/03/70
06/03/70
06/03/70
06/03/70
06/03/70
06/03/70
03/25/71
06/03/70

03/25/71
03/25/71
12/16/70

03/25/71
03/25/71
10/3/25/71
03/25/71

03/25/71
03/25/71
12/16/70
12/16/70
12/16/70
12/16/70
12/16/70
12/16/70

```

365.     LIM=NOFNUC(K)
366.     IF(SKIP(K).NE.0) GO TO 1114                                12/16/70
367.     DO 1013 I=1,LIM
368.     L=ISOTOP+I
369.     IF(MASS(L).GT.1.E-30.AND.LAMBDA(L).GT.1.E-30) GO TO 1014
370. 1013 CONTINUE
371.     SKIP(K)=LIM
372.     GO TO 1114                                                12/16/70
373. 1014 BLANK=.FALSE.                                           12/16/70
374.     FISPRD=.TRUE.                                           12/16/70
375. 1114 ISOTOP=ISOTOP+LIM                                       12/16/70
376. 1214 CONTINUE                                               12/16/70
377.     DO 1314 L=N1,N2                                         12/16/70
378.     IF(MASS(L).GT.1.E-30.AND.LAMBDA(L).GT.1.E-30) GO TO 1414 12/16/70
379. 1314 CONTINUE                                               12/16/70
380.     GO TO 1514                                               12/16/70
381. 1414 BLANK=.FALSE.                                           12/16/70
382.     ACTPRD=.TRUE.                                           12/16/70
383. 1514 IF(.NOT.BLANK) GO TO 1015                               12/16/70
384. C INCREMENT VARIABLES FOR BLANK PUFF AND RETURN TO START OF CALCULATION 03/25/71
385.     IF(itime.NE.1) GO TO 1015                               12/16/70
386.     TIME=TIME+DELT(1)                                       12/16/70
387.     RTIME=RTIME+DELT(1)                                     12/16/70
388.     GO TO 1007                                               12/16/70
389. 1015 I2=2
390.     IF(DK) GO TO 6015                                         01/25/71
391. C DECAY FISSION PRODUCTS FROM RELEASE POINT TO EXPOSURE POINT 03/25/71
392.     IF(FISPRD) CALL CHAIN(T(1))                             01/25/71
393. 6015 CONTINUE                                               01/25/71
394. C X IS DOSE ARRAY FOR SEMI INFINITE CALCULATION
395.     X(itime)=0.0
396.     IF(.NOT.ACTPRD) GO TO 4015                               12/16/70
397. C FOR ACTIVATION PRODUCTS - CALCULATE SUM OF DOSE RATE FOR EACH ENERGY 03/25/71
398. C DECAY ACTIVATION PRODUCTS TO EXPOSURE POINT              03/25/71
399.     DO 3015 K=N1,N2
400.     IF(MASS(K).LT.1.E-30) GO TO 3015
401.     IF(DK) GO TO 5015                                         01/25/71
402.     MASS(K)=MASS(K)*EXP(-LAMBDA(K)*T(1))
403. 5015 DO 2015 ENERGY=1,GROUPS                                01/25/71
404.     2015 X(itime)=X(itime)+MASS(K)*BURST(K,ENERGY)*DRDX(1,ENERGY)
405. 3015 CONTINUE
406. 4015 CONTINUE                                                12/16/70
407.     IF(.NOT.FISPRD) GO TO 1019                               12/16/70
408. C FOR FISSION PRODUCTS - CALCULATE SUM OF DOSE RATE FOR EACH ENERGY GROUP 03/25/71
409. C ISOTOP=0
410.     DO 1018 K=1,CHAINS
411.     LIM=NOFNUC(K)
412.     IF(SKIP(K).NE.0) GO TO 1018
413.     DO 1016 I=1,LIM
414.     L=ISOTOP+I
415.     DO 1016 ENERGY=1,GROUPS
416.     1016 X(itime)=X(itime)+MASS(L)*BURST(L,ENERGY)*DRDX(1,ENERGY)
417. 1018 ISOTOP=ISOTOP+LIM
418. 1019 IF(itime.NE.3) GO TO 1024
419. C INCREMENT DOSE TIME AND INTEGRATE FOR CURRENT DOSE        03/25/71
420.     DTIME=CTIME+DELT(1)
421.     GAMMA=GAMMA+DELT(1)*(36799.*X(1)+139349.*X(2)-121797.*X(3)+123133.
422.     1*X(4)-88547.*X(5)+41499.*X(6)-11351.*X(7)+1375.*X(8))/120960.
423.     PRINT 52, CTIME,X(2),GAMMA
424. 52 FORMAT(' AT TIME=1PE10.3,' DOSE RATE ='E10.3,' ACCUMULATED DOSE
425. 1 ='E10.3)
426.     GO 1023 I=2,8
427. 1023 X(I-1)=X(I)
428.     GO TO 1025
429. 1024 itime=itime+1
430.     IF(itime.NE.2) GO TO 1025
431. C FOR FIRST TIME INCREMENT INITIALIZE DOSE                  03/25/71
432.     GAMMA=0.
433.     PRINT 52, T(1),X(1),GAMMA
434. C INCREMENT TIMES                                           03/25/71
435. 1025 TIME=TIME+DELT(1)
436.     RTIME=RTIME+DELT(1)
437.     GO TO 1007
438. C
439. C THE FOLLOWING STATEMENTS COMPLETE THE DOSE INTEGRATION AFTER RELEASE 03/25/71
440. C HAS BEEN REACHED                                         03/25/71
441. C
442. 1026 DOSE(1)=(19097.*X(1)+65112.*X(2)-45461.*X(3)+375.04.*X(4)-20211.*X(
443. 15)+6312.*X(6)-863.*X(7))*DELT(1)/60480.+GAMMA
444.     DTIME=DTIME+DELT(1)
445.     PRINT 52, DTIME,X(2),DOSE(1)
446.     DOSE(1)=(1139.*X(1)+5640.*X(2)+33.*X(3)+1328.*X(4)-807.*X(5)+264.*
447. 1*X(6)-37.*X(7))*DELT(1)/3730.+GAMMA
448.     DTIME=DTIME+DELT(1)
449.     PRINT 52, DTIME,X(3),DOSE(1)
450.     DOSE(1)=(683.*X(1)+3240.*X(2)+1161.*X(3)+2176.*X(4)-729.*X(5)+216.
451. 1*X(6)-29.*X(7))*DELT(1)/2240.+GAMMA
452.     DTIME=DTIME+DELT(1)
453.     PRINT 52, DTIME,X(4),DOSE(1)
454.     DOSE(1)=(143.*X(1)+696.*X(2)+192.*X(3)+752.*X(4)+87.*X(5)+26.*X(6)
455. 1

```

```

457. 1-4.*X(7))/DELT(1)/945.+GAMMA
458. DTIME=DTIME+DELT(1)
459. PRINT 52, DTIME,X(5),DOSE(1)
460. DOSE(1)=(523.*X(1)+4680.*X(2)-1425.*X(3)+6400.*X(4)+225.*X(5)+1848
461. 1.*X(6)-155.*X(7))*5.*DELT(1)/12096.+GAMMA
462. DTIME=DTIME+DELT(1)
463. PRINT 52, DTIME,X(6),DOSE(1)
464. DOSE(1)=(41.*X(1)+X(7))+216.*X(2)+X(6))+27.*X(3)+X(5))+272.*X(4
465. 1))*DELT(1)/140.+GAMMA
466. DTIME=DTIME+DELT(1)
467. PRINT 52, DTIME,X(7),DOSE(1)
468.
469. C DO NECESSARY TAPE OR DRUM HANDLING OPERATIONS
470. IF(TAPE)GO TO 1027
471. REWIND UNIT
472. GO TO 1028
473. 1027 IF(BACKUP) GO TO 1030
474. IF(REWIND) GO TO 1029
475. CALL NTRAN(UNIT,8,1)
476. 1028 RETURN
477. 1029 CALL NTRAN(UNIT,10)
478. GO TO 1028
479. 1030 N=-(TAPIN/OLDDT+3)
480. CALL NTRAN(UNIT,7,N)
481. GO TO 1028
482. END

```

03/25/71
03/25/71

```

1. C SUBROUTINE DECODE READS A CONTROL WORD CARD 03/25/71
2. SUBROUTINE DECODE(WORD,WORDS) 03/25/71
3. DIMENSION WORD(4)
4. COMMON/TITLE/TITLE(14),A(80)
5. IMPLICIT INTEGER (A,B,C,M)
6. DATA BLANK/1H /
7. C INITIALIZE WORDS TO BLANKS 03/25/71
8. DO 1 I=1,14
9. 1 WORD(I)=BLANK
10. PRINT 9, TITLE
11. 9 FORMAT('1'13A6,A2)
12. C READ 80 CHARACTERS 03/25/71
13. READ 2, A
14. 2 FORMAT(80A1)
15. PRINT 10, A
16. 10 FORMAT('0'80A1)
17. C INITIALIZE PARAMETERS 03/25/71
18. COLUMN=0
19. WORDS=0
20. BIT=0
21. C START ANALYSIS OF CURRENT CHARACTER 03/25/71
22. 3 COLUMN=COLUMN+1
23. IF(COLUMN.GT.80) GO TO 7
24. C SET CHARAC EQUAL TO CURRENT CHARACTER 03/25/71
25. CHARAC=FLD(0,6,A(COLUMN))
26. C CHARAC=5 FOR BLANKS 03/25/71
27. IF(CHARAC.EQ.5) GO TO 3
28. C START FORMATION OF NEXT WORD 03/25/71
29. WORDS=WORDS+1
30. 4 IF(BIT.NE.36) GO TO 5
31. C WHEN WORD HAS MORE THAN 6 CHARACTERS, PLACE SECOND HALF IN NEXT WORD 03/25/71
32. WORDS=WORDS+1
33. BIT=0
34. C PLACE CHARAC IN CURRENT WORD STARTING AT BIT (6 BITS PER CHARACTER) 03/25/71
35. 5 FLD(BIT,6,WORD(WORDS))=CHARAC
36. BIT=BIT+6
37. 6 COLUMN=COLUMN+1
38. IF(COLUMN.GT.80) GO TO 7
39. C START ANALYSIS OF NEXT CHARACTER 03/25/71
40. CHARAC=FLD(0,6,A(COLUMN))
41. IF(CHARAC.NE.5) GO TO 4
42. BIT=0
43. C RETURN TO MAIN 03/25/71
44. GO TO 3
45. 7 RETURN
46. END

```

```

1. C SUBROUTINE RIBD PERFORMS INVENTORY CALCULATION FOR SPECIFIED REACTOR 03/25/71
2. C OPERATING HISTORIES. 03/25/71
3. C 03/25/71
4. SUBROUTINE RIBD 03/25/71
5. COMMON/TITLE/CASE(6)
6. DIMENSION PAR2(21,10,2),PAR3(20,10,2),Y(5),TERM(20),SUM2(20),
7. ISUM3(20),FX(20),IM(20),WL(25),WM(25),FM(20),FR(20),TF(21),TX(21),
8. 2FXT(21),SEMLT(9),EMLT(9,9),DT(9),TLN(9),TLH(9),HTF(40)
9. COMMON NCD,ML(600),FEED3(600),FEED8(450),REC(600),ISU(450),
10. 1FEED10(450),SPACE(1050),DATA1(10,450),DATA2(10,450),TOTL(10,40,3),
11. 2LSTM(10,96),LISTA(25,40),NRM(96),NRA(40),FLAM(9),GROS(10,3),
12. 3OUT(10),SAVEHC(10),SAVEHA(10),SAVEVS(10),ARFP(10),TF(21),WN(25),
13. 4 EXT(4)
14. COMMON LA(450),FEED4(450),FEED5(450),FEED6(450),FEED7(450),FEED9(4
15. 150)
16. COMMON PUN 12/16/70
17. LOGICAL PUN,BSKIP,BDUM 01/12/71
18. DATA ENFISS/200./
19. LOGICAL L6 01501
20. IMPLICIT DOUBLE PRECISION (H)
21. EQUIVALENCE (TOTL,PAR2), (TOTL(421),PAR3), (TOTL(821),Y),
22. X(TOTL(826),TERM), (TOTL(846),SUM2), (TOTL(866),SU43),
23. X(TOTL(886),FX), (TOTL(906),IM), (TOTL(926),WL), (TOTL(952),WM),
24. X(TOTL(977),FM), (TOTL(997),FR), (TOTL(1017),TF), (TOTL(1038),TX),
25. X(TOTL(1059),FXT), (TOTL(1080),SEMLT), (TOTL(1089),EMLT)
26. 1050 FORMAT(1H+102X,4HSTER13) 03301
27. 1060 FORMAT(1H A3,I3,2X,1P12E10.3,I2) 03401
28. 1061 FORMAT(1H0,31HANCESTOR SUMMATION TROUBLE FOR A3,I3,5H, LO=1PE8.2, 05/27/69
29. X1H,) 05/27/69
30. 1062 FORMAT(1H+,51X,33H NEGATIVE ANCESTOR SUM IS IGNORED) 05/27/69
31. 1063 FORMAT(1H+,51X,34H FRACTIONAL ERROR IN CONCENTRATION F7.4) 05/27/69
32. 1064 FORMAT(1H0,25HJUNE,1969 VERSION OF RIBD) 05/27/69
33. 1065 FORMAT(1H0,77HLOW FLUX WILL CAUSE SOME ZERO CONCENTRATIONS. RE-EXA 05/27/69
34. XHINE T, SIGMA, AND THETA.) 05/27/69
35. 1066 FORMAT(1H0,47HLARGE VALUE OF T MAY CAUSE CONVERGENCE TROUBLES) 05/27/69
36. 1067 FORMAT(1H0,67HLARGE NVT MAY CAUSE TROUBLE, EITHER T TOO LARGE OR S 05/27/69
37. XIGMA TOO SMALL.) 05/27/69
38. 1070 FORMAT(2E10.4,30X,I1) 03501
39. 1080 FORMAT(1H 1P2E15.4,45X, I5) 03601
40. 1092 FORMAT(1H1,3X,24HLIBRARY OUT OF ORDER AT A3,I4/(28X,A3,I4)) 04001
41. 5000 FORMAT(1H1/1H 6A6,33H ACTIVITY AFTER SHUTDOWN - CURIESF12.1,7H MWD 04101
42. 1 IN F7.1,6H DAYS.) 04102
43. 5010 FORMAT(1H+106X,7HAT. NO.I3,1X,A2) 04301
44. 5011 FORMAT(1H1/1H 6A6,33X,F12.1,7H MWD INF7.1,5H DAYS) 04401
45. 5012 FORMAT(8H0ISOTOPE) 04501
46. 5013 FORMAT(1H+12X,8HSHUTDOWNF6.1,A5,F6.1,A5,F6.1,A5,F6.1,A5,F6.1,A5, 04601
47. 1F6.1,A5,F6.1,A5,F6.1,A5,F6.1,A5) 04701
48. 5002 FORMAT(2H A3,I3,2X,1P10E11.3) 04801
49. 5003 FORMAT(10H TOTAL 1P10E11.3) 04901
50. 5004 FORMAT(1H+106X,8H SUMMARY) 05001
51. 5015 FORMAT(1H037X,41HCONCENTRATION AFTER SHUTDOWN - GRAM ATOMS) 05301
52. 5016 FORMAT(8H0ELEMENT) 05401
53. 5017 FORMAT(1H I2,1X,A2,4X,1P10E11.3) 05501
54. 5018 FORMAT(1H I2,1X,A5,1X,1P10E11.3) 05601
55. 1021 FORMAT(I3) 05701
56. 6000 FORMAT(1H1/1H 6A6,33H BETA POWER AFTER SHUTDOWN - MW F12.1,7H MWD 05801
57. 1 IN F7.1,6H DAYS.) 05902
58. 6015 FORMAT(1H037X,31HGAMMA POWER AFTER SHUTDOWN - MW) 06001
59. 9000 FORMAT(1H1/1H 6A6, 59H BETA ENERGY RELEASE - MWD, BETWEEN CONSECUT 06101
60. 1IVE TIME POINTS) 06102
61. 9013 FORMAT(1H037X,36HCONCENTRATION AFTER SHUTDOWN - GRAMS) 06301
62. 9002 FORMAT(2H A3,I3,7X,1P9E11.3) 06401
63. 9003 FORMAT(7H TOTAL,8X,1P9E11.3) 06501
64. 9017 FORMAT(1H I2,1X,A2,9X,1P9E11.3) 06601
65. 9015 FORMAT(1H037X,20HGAMMA ENERGY RELEASE) 06701
66. 9018 FORMAT(1H I2,1X,A5,6X,1P10E11.3) 06901
67. 5107 FORMAT(1H09HNOBLE GAS1P10E11.3) 06904
68. 5108 FORMAT(1H09HHALOGENS 1P10E11.3) 06905
69. 5109 FORMAT(1H09HVOL. SOL. 1P10E11.3) 06906
70. 5110 FORMAT(1H09HAR.F.F.P. 1P10E11.3) 06907
71. C SET PRINTER AS OUTPUT UNIT 3 03/25/71
72. CALL SETIO(3,2)
73. C ANALYZE LIBRARY DATA AND SET UP CALCULATION PARAMETERS 03/25/71
74. WRITE (3,1064) 05/27/69
75. LOWA=LA(1)-1 07301
76. DO 3 J=1,ACD 07401
77. IF (MP-ML(J))5,1,6 07601
78. 5 M=ML(J) 07501
79. K=KM+1 03001
80. 1 K=LA(J)-LOWA 03101
81. IF (KA)6,8,2 08201
82. 6 WRITE (3,1002) REC(J-1),ML(J-1),REC(J),ML(J),REC(J+1),ML(J+1) 08301
83. STOP
84. 2 NRM(KM)=NRM(KM)+1 08501
85. NRA(KA)=NRA(KA)+1 08601
86. J=NRM(KM) 08701
87. JA=NRA(KA) 08801
88. LISTM(JM,KM)=J 08901

```


89.	LISTA(IA,KA)=0	09001
90.	NAT=MAX0(NAT,KA)	09101
91.	3 CONTINUE	09201
92.	IMAX=KM	09301
93.	GO TO 809	09401
94.	C READ FIRST POWER LEVEL AND OPERATING PARAMETERS	03/25/71
95.	9 READ 1001, P0, TIME, T, SIGA25, DEL, ASIG, ENF	
96.	1001 FORMAT(8F10.0)	
97.	C START CALCULATION FOR FIRST POWER LEVEL	03/25/71
98.	NSIG=ASIG	
99.	IF(ENF.GT.0.) ENFISS=ENF	
100.	KEY2=0	09503
101.	SIGA25=SIGA25+1.E-10	05/27/69
102.	SCALE=1./TIME	
103.	FLUX=T/(SIGA25*TIME)	09601
104.	XBAR=EXP(-T)	09801
105.	PXBAR=XBAR	09901
106.	P=P0*1.6872E+8/ENFISS	
107.	TFISS=P*TIME	10101
108.	X=FLUX*SIGA25	10201
109.	THETA=TIME*SCALE	
110.	P=P/SCALE	
111.	X=X/SCALE	
112.	PRATIO=FLUX/P0	04/24/69
113.	NSTEP=1	10401
114.	TET=TIME	10501
115.	F=WD=PC*TIME	10601
116.	P25=P/(1.0+DEL)	10701
117.	PRINT 1010, P0, TIME, T, SIGA25, DEL, NSIG, ENFISS	
118.	1010 FORMAT('0POWER='1PE9.3,' PW DAYS='E9.3,' PARAMETER='E9.3,' SIGA2	
119.	15='E9.3,' NON 35 FISS/35 FISS='E9.3,' SIGNAL='I2,' MEV='E9.3)	
120.	IF(FLUX.LT..0E-10) WRITE (3,1065)	05/29/69
121.	IF(T.GT.4.) WRITE (3,1066)	05/29/69
122.	ENVT=FLUX*TIME	05/29/69
123.	IF(ENVT.GT.1.) WRITE (3,1067)	05/29/69
124.	IF(NSIG)31,31,32	11001
125.	31 JS1=2	11101
126.	GO TO 33	11201
127.	32 WRITE (3,1050)NSTEP	11301
128.	JS1=1	11401
129.	C START CALCULATION FOR CURRENT POWER LEVEL	03/25/71
130.	33 ADD=1.	
131.	T(1)=THETA	11701
132.	WL(1)=THETA	11801
133.	SSUM=THETA	11901
134.	DO 8 L=2,20	
135.	ADD=ADD+1.0	12101
136.	T(L)=T(L-1)*THETA/ADD	12201
137.	WL(L)=-T/ADD*WL(L-1)	12401
138.	8 SSUM=SSUM+WL(L)	12501
139.	TX(1)=SSUM	12601
140.	DO 11 L=2,20	
141.	ADD=L	12801
142.	SSUM=0.0	12901
143.	DO 12 I=1,20	
144.	WL(I)=WL(I)*THETA/ADD	13101
145.	ADD=ADD+1.0	13201
146.	12 SSUM=SSUM+WL(I)	13301
147.	11 TX(L)=SSUM	13401
148.	I0=1	13501
149.	I=2	13601
150.	DO 290 K=1,NMA	13701
151.	RET=0.0	13801
152.	13 N1=0	13901
153.	N2=0	14001
154.	N3=0	14101
155.	OPEN UNIT(K)	14201
156.	DO 2900 J=1,JM	14301
157.	Y(1)=0.0	14401
158.	Y(2)=0.0	14501
159.	Y(3)=0.0	14601
160.	Y(4)=0.0	14701
161.	Y(5)=0.0	14801
162.	KI=LISTM(J,K)	14901
163.	IS=ISC(KI)	15001
164.	AT=LA(KI)	15101
165.	KA=LA(KI)-LOWA	15201
166.	LE=NNA(KA)	15301
167.	F=(FLUX*FEEC6(KI)+FEEC3(KI))/SCALE	
168.	I1=KI	15501
169.	I2=KI	15601
170.	I3=KI	15701
171.	I4=KI	15801
172.	I5=KI	15901
173.	MF=ML(KI)-1	16001
174.	PAR2(21,J,I4)=AT	
175.	DO 10 L=1,LM	16201
176.	MF=LISTA(L,KA)	16301
177.	IF(ML(MF)-F)10,100,170	16401
178.	10 CONTINUE	16501
179.	100 I4=MF	16601
180.	I5=I4+1	16701

181.	DO 101 L=1,10	16001
182.	L4=L	16901
183.	IF(PAR2(21,L,10)-AT)101,102,101	
184.	101 CONTINUE	17101
185.	102 L5=L4+1	17201
186.	IF(ISC(1)) GO TO 140	17301
187.	110 IF(ISO(NP)-2)120,130,130	17401
188.	120 Y(4)=FLUX*FEED6(14)*FEED7(14)	17501
189.	GO TO 135	17601
190.	130 IS=NP	17701
191.	L5=L4	17801
192.	135 Y(5)=FLUX*FEED6(15)*FEED7(15)	17901
193.	GO TO 170	18001
194.	140 IF(ISC(NP)-2)150,160,160	18101
195.	150 Y(4)=FLUX*FEED6(14)*(1.0-FEED7(14))	18201
196.	GO TO 165	18301
197.	160 IS=NP	18401
198.	L5=L4	18501
199.	165 Y(5)=FLUX*FEED6(15)*(1.0-FEED7(15))	18601
200.	170 IF(N1-1)240,180,210	18701
201.	180 I1=K1-1	18801
202.	L1=J-1	18901
203.	Y(1)=FEED3(I1)*(1.0-FEED8(I1))	19001
204.	IF(N2-1)240,185,190	19101
205.	185 WRITE (3,1002)REC(KI),ML(KI)	19201
206.	GO TO 240	19301
207.	190 I2=I1-1	19401
208.	L2=L1-1	19501
209.	Y(2)=FEED3(I2)*(1.0-FEED8(I2))	19601
210.	IF(N3-1)240,200,240	19701
211.	200 I3=I2-1	19801
212.	L3=L2-1	19901
213.	Y(3)=FEED3(I3)*FEED8(I3)	20001
214.	GO TO 240	20101
215.	210 I2=K1-1	20201
216.	L2=J-1	20301
217.	IF(ISC(1)) GO TO 230	20401
218.	220 Y(2)=FEED3(I2)*FEED8(I2)	20501
219.	GO TO 240	20601
220.	230 Y(2)=FEED3(I2)*(1.0-FEED8(I2))	20701
221.	IF(N2-1)240,200,240	20801
222.	240 N3=N2	20901
223.	N2=N1	21001
224.	N1=IS	21101
225.	A=P*FEED5(KI)/100.0	21201
226.	B=P25*(FEED4(KI)-FEED5(KI))/100.0	21301
227.	DO 1112 I=1,5	
228.	1112 Y(I)=Y(I)/SCALE	
229.	DO 241 L=1,20	
230.	SUM2(L)=Y(1)*PAR2(L,L1,IN)+	21501
231.	1 Y(2)*PAR2(L,L2,IN)+	21601
232.	2 Y(3)*PAR2(L,L3,IN)+	21701
233.	3 Y(4)*PAR2(L,L4,IO)+	21801
234.	4 Y(5)*PAR2(L,L5,IO)	21901
235.	241 SUM3(L)=Y(1)*PAR3(L,L1,IN)+	22001
236.	1 Y(2)*PAR3(L,L2,IN)+	22101
237.	2 Y(3)*PAR3(L,L3,IN)+	22201
238.	3 Y(4)*PAR3(L,L4,IO)+	22301
239.	4 Y(5)*PAR3(L,L5,IO)	22401
240.	F1=A/F	22501
241.	F2=B/(F-X)	22601
242.	F3=F2*XBAR	22701
243.	F4=DATA1(1,KI)-F1-F2	22801
244.	ARG=F*THETA	22901
245.	SUM=0.0	23001
246.	ERROR=0.	
247.	HARG=F	
248.	FX(1)=-X*F3	
249.	FR(1)=1.0/F	23201
250.	DO 242 L=2,20	
251.	FX(L)=-FX(L-1)*X	
252.	FR(L)=-FR(L-1)/F	23501
253.	242 TERM(L)=0.0	23601
254.	IF(ARG.LT.79.) GO TO 260	05/27/69
255.	243 NT=1	23801
256.	NTT=0	05/27/69
257.	DO 244 L=1,20	
258.	TERM(L)=SUM2(L)*FR(L)	24001
259.	IF(TERM(L).EQ.0.) GO TO 245	
260.	SUM=SUM+TERM(L)	24101
261.	IF(L.EQ.1) GO TO 244	
262.	ERROR=ABS(TERM(L)/SUM)	
263.	IF(ERROR.GT.2.E-9) GO TO 244	05/27/69
264.	NT2=L-NTT	05/27/69
265.	IF(NT2.EQ.1) GO TO 245	05/27/69
266.	NTT=L	05/27/69
267.	244 CONTINUE	24301
268.	245 DATA2(1,KI)=F1+F3+SUM	24401
269.	PAR2(1,J,IN)=DATA2(1,KI)	24501
270.	L=FT=0.0	24601
271.	NTT=0	05/27/69
272.	DO 247 L=2,20	

273.	SSUM=0.0	24601
274.	N=0	24621
275.	DO 246 I=L,20	
276.	N=N+1	25001
277.	ADD=FX(N)*SUM2(I)	25011
278.	SSUM=SSUM+ADD	
279.	ER=ABS(ADD/SSUM)	
280.	IF(ER.GT.2.E-9) GO TO 246	05/27/69
281.	IF(NTT.GT.0) GO TO 247	05/27/69
282.	NTT=7	05/27/69
283.	246 CONTINUE	
284.	247 PAR2(L,J,IN)=FX(L-1)+SSUM	
285.	249 HARG=F*THETA	
286.	HEFT=EXP(-HARG)	
287.	IF(NT.EQ.1) HEFT=0.000	06/06/69
288.	HARG=F	
289.	HNUM=(1.00-HEFT)/HARG	
290.	HAD=1.00	
291.	HFA=1.000	
292.	HEFT=THETA	
293.	SSUM=SUM	
294.	DO 250 L=1,20	
295.	IF(SUM3(L).GT.0.) GO TO 2491	
296.	SSUM=0.	
297.	GO TO 2490	
298.	2491 SSUM=(SUM3(L)-SSUM)/F	
299.	IF(SSUM.GT.0.) GO TO 2490	
300.	SSUM=SUM3(L)/F*(1.-1./ARG)	
301.	N3BI=N3BI+1	
302.	2490 HFA=HFA*HEFT/HAD	
303.	HAD=HAD+1.000	
304.	FA=HFA	
305.	FMULT=HNUM	
306.	PAR3(L,J,IN)=FMULT*F4+FA*F1+F2*TX(L)+SSUM	
307.	250 HNUM=(HFA-HNUM)/HARG	
308.	GO TO 300	26001
309.	260 EFT=EXP(ARG)	26101
310.	EMFT=1.0/EFT	26201
311.	F5=F4*EMFT	26301
312.	IF(ARG.GT.4.) GO TO 291	
313.	NT=2	
314.	267 ERROR=0.	
315.	HTF(1)=1.000	04/25/69
316.	DO 2671 I=2,20	04/25/69
317.	2671 HTF(I)=-HTF(I-1)*HARG	04/25/69
318.	DO 268 L=1,20	
319.	TERM(L)=HTF(L)*DSCE(SUM3(L))	
320.	IF(TERM(L).EQ.0.) GO TO 266	
321.	SUM=SUM+TERM(L)	26801
322.	IF(L.EQ.1) GO TO 265	
323.	ERROR=ABS(TERM(L)/SUM)	
324.	IF(ERROR.LT.2.E-9) GO TO 266	
325.	265 CONTINUE	27001
326.	266 SS2=THETA*A	
327.	SSUM=THETA*B	
328.	Q=SSUM	
329.	SS1=THETA	
330.	WM(1)=SS1	
331.	WL(1)=SSUM	
332.	WV(1)=SS2	
333.	ADD=1.	
334.	DO 262 L=2,25	
335.	ADD=ADD+1.0	27801
336.	Q=-T/ADD*Q	27901
337.	WL(I)=-ARG/ADD*WL(L-1)+Q	28001
338.	SSUM=SSUM+WL(I)	28101
339.	WV(L)=-ARG/ADD*WV(L-1)	28201
340.	WV(L)=-ARG/ADD*WV(L-1)	
341.	SS1=SS1+WV(L)	
342.	262 SS2=SS2+WV(L)	28301
343.	FXT(1)=SSUM	28401
344.	TF(1)=SS2	28501
345.	TTF(1)=SS1	
346.	DO 264 L=2,21	
347.	SSUM=0.0	29701
348.	SS2=0.0	28801
349.	SS1=0.	
350.	ADD=L	28901
351.	DO 263 I=1,25	
352.	WL(I)=WL(I)*THETA/ADD	29101
353.	WV(I)=WV(I)*THETA/ADD	29201
354.	WM(I)=WM(I)*THETA/ADD	
355.	ADD=ADD+1.0	29301
356.	SSUM=SSUM+WL(I)	29401
357.	SS1=SS1+WV(I)	
358.	263 SS2=SS2+WV(I)	29501
359.	FXT(L)=SSUM	29601
360.	TTF(L)=SS1	
361.	264 TF(L)=SS2	29701
362.	DATA2(1,KI)=DATA1(1,KI)*EMFT+TF(1)+FXT(1)+SUM	
363.	PAR2(1,J,IN)=DATA2(1,KI)	29901
364.	HAD=FG	

365.	269	DO 272 L=2,20	
366.		HFA=HFA+HFA	
367.		HFA=0.000	
368.		DO 270 I=2,L	30231
369.		HFA=I+1	30301
370.	270	HFA=HFA+HFA(I-1)*DBLE(SUM2(N))	
371.		SSUM=HFA+HFA(I)*DBLE(SUM)	
372.	272	PAR2(L,J,I)=ADD+FX(L-1)+SSUM	
373.		DO 2820 I=16,20	04/24/69
374.		RATIO = SUM3(I)*FLOAT(I)/SUM3(I-1)	04/24/69
375.		FTD=RATIO-OR	04/24/69
376.		IF(FTD.LT.0.) FTD=0.	04/24/69
377.		STD=FD-FTD	04/24/69
378.		IF(STD.LT.0.) STD=0.	04/24/69
379.		TTC=SG-STD	04/24/69
380.		IF(TTC.LT.0.) TTC=0.	04/24/69
381.		FRD=TTC-TTC	04/24/69
382.		IF(FRD.LT.0.) FRD=0.	04/24/69
383.		TD=TTC	04/24/69
384.		SG=STD	04/24/69
385.		FD=FTD	04/24/69
386.	2820	OR=RATIO	04/24/69
387.		TD=TD-FRD	04/24/69
388.		SG=SG-TD	04/24/69
389.		FD=FD-SG	04/24/69
390.		IF(FD.LT.0.) FD=0.	04/24/69
391.		RATIO=RATIO+FD	04/24/69
392.		EXT(1)=SUM3(20)*RATIO/21.	04/24/69
393.		DO 2821 I=2,4	04/24/69
394.		TD=TD-FRD	04/24/69
395.		SG=SG-TD	04/24/69
396.		FD=FD-SG	04/24/69
397.		IF(FD.LT.0.) FD=0.	04/24/69
398.		RATIO=RATIO+FD	04/24/69
399.	2821	EXT(I)=EXT(I-1)*RATIO/(20.+FLOAT(I))	04/24/69
400.		DO 281 L=1,20	
401.		LIM=20-L	
402.		HFA=0.000	04/24/69
403.		IF(L.LT.4) GO TO 2823	04/24/69
404.		N=MOD(L,2)	04/24/69
405.		IF(N.LC.0) GO TO 2822	04/24/69
406.		LIM=LIM-1	04/24/69
407.		N=20-L	04/24/69
408.		HAD1=HFA(N)*DBLE(SUM3(20)) +HFA(N+1)*DBLE(EXT(1))	04/24/69
409.		HAD2=HFA(N+2)*DBLE(EXT(2))+HFA(N+3)*DBLE(EXT(3))	04/24/69
410.		HFA=HAD1**2/(HAD1-HAD2)	04/24/69
411.		GO TO 2824	04/24/69
412.	2822	L=20-L+1	04/24/69
413.		HAD1=HFA(N)*DBLE(EXT(1))+HFA(N+1)*DBLE(EXT(2))	04/24/69
414.		HAD2=HFA(N+2)*DBLE(EXT(3))+HFA(N+3)*DBLE(EXT(4))	04/24/69
415.		HFA=HAD1**2/(HAD1-HAD2)	04/24/69
416.	2824	IF(HFA.LT.0.000) HFA=0.000	04/24/69
417.		IF(LIM.GT.0) GO TO 2823	04/24/69
418.		SSUM=HFA	04/24/69
419.		GO TO 281	04/24/69
420.	2823	CONTINUE	04/24/69
421.		DO 280 I=1,LIM	
422.		N=1+L	31401
423.	280	HFA=HFA+HFA(I)*DBLE(SUM3(N))	
424.		SSUM=HFA	
425.		IF(SSUM.GT.0.) GO TO 281	04/24/69
426.		IF(SSUM.GT.0..AND.SSUM.LE.SUM3(L+1)) GO TO 281	
427.		IF(SSUM.LT.0..OR.SSUM.GT.SUM3(L+1)) SSUM=SUM3(L+1)	
428.	281	PAR3(L,J,I)=DATA1(I,I)*HFA(I)+HFA(L+1)+FXT(L+1)+SSUM	
429.		GO TO 300	31701
430.	291	FA=1.0	31801
431.		AGE=1.0	31901
432.		NT=3	32001
433.		HARG=F*THETA	
434.		HEFT=EXP(HARG)	
435.		HNUM=HEFT-1.000	
436.		HAD=1.000	
437.		HFA=1.000	
438.		DO 2291 L=1,20	
439.		HFA(L)=HARG/HFA	
440.		HFA=HFA+HARG/HAD	
441.		HNUM=HNUM-HFA	
442.	2291	HAD=HAD+1.000	
443.		SS2=0.0	32201
444.		AL(1)=1.0	32301
445.		DO 294 L=1,20	
446.		TF(L)=HFA(L)	
447.		FA=FA/AGE+.RG	32601
448.		AL(L+1)=AL(L)+FA	32601
449.		SS2=FA*SS2+AL(L)*SUM2(L)	32901
450.		AL(L)=EMFT+SS2	33001
451.	294	AL=AL+1.0	33101
452.		RT=0	05/27/69
453.		DO 292 L=1,20	
454.		TF(L)=FA(L)+TF(L)*SUM2(L)	33301
455.		IF(TF(L).LT.0.) GO TO 293	
456.		SUM=SUM+TF(L)	33401

457.	IF(LT.1) GO TO 292	05/27/69
458.	ERROR=ABS(TERM(L)/SUM)	
459.	IF(ERROR.GT.2.E-9) GO TO 292	05/27/69
460.	MT=L-MT	05/27/69
461.	IF(MT.EQ.1) GO TO 298	05/27/69
462.	MT=L	05/27/69
463.	292 CONTINUE	04/25/69
464.	GO TO 293	
465.	2940 CONTINUE	
466.	MT=4	
467.	SUM=0.0	34001
468.	DO 299 L=1,20	
469.	299 TERM(L)=0.5	34201
470.	GO TO 267	34301
471.	298 DATA2(1,KI)=DATA1(1,KI)*EMFT+F1*(1.0-EMFT)+F2*(XBAR-EMFT)+SUM	34401
472.	PAR2(1,J,IN)=DATA2(1,KI)	34501
473.	EFM=FB	
474.	DO 295 L=2,20	
475.	EFM=EFM*F	
476.	SSUM=0.0	34701
477.	N=0	34721
478.	DO 2950 I=L,20	
479.	NEL=1	34901
480.	ADD=FR(I)*SUM2(I)*TF(I)	34911
481.	SSUM=SSUM+ADD	
482.	ER=ABS(ADD/SSUM)	
483.	IF(EP.LT.1.E-8) GO TO 295	
484.	2950 CONTINUE	
485.	295 PAR2(L,J,IN)=EFM+FX(L-1)+NM(L-1)+SSUM	
486.	GO TO 249	35201
487.	300 CONTINUE	
488.	ERROR=ERROR*SUM/DATA2(1,KI)	
489.	IF(ERROR.LT..001.AND.SUM.GE.0.) GO TO 305	05/27/69
490.	IF(LT.NE.3) GO TO 2921	
491.	IF(ARG.LT.4.5) GO TO 2920	05/27/69
492.	2921 IF(RE1.GT.0.) GO TO 2922	05/27/69
493.	RET=1.	
494.	GO TO 13	
495.	2922 WRITE (3,1061) REC(KI),ML(KI),ARG	05/27/69
496.	IF(SUM.GT.0.) GO TO 2923	05/27/69
497.	WRITE (3,1062)	05/27/69
498.	DATA2(1,KI)=DATA2(1,KI)-SUM	05/27/69
499.	PAR2(1,J,IN)=PAR2(1,J,IN)-SUM	05/27/69
500.	GO TO 305	05/27/69
501.	2923 WRITE (3,1063) ERROR	05/27/69
502.	305 FLAM(J)=FELD3(KI)	36001
503.	SSUM=0.0	36101
504.	LIM=J-1	36201
505.	IF(LIM)402,403,400	36301
506.	400 GO 402 L=LIM	36401
507.	401 DATA(L+1,1)=Y(1)*DATA1(L+1,1)+Y(2)*DATA1(L+1,2)+Y(3)*	36501
508.	XDATA1(L+1,3))*SCALE/(FLAM(J)-FLAM(L))	
509.	402 SSUM=SSUM+DATA1(L+1,KI)	36701
510.	403 DATA1(J+1,KI)=DATA2(1,KI)-SSUM	36801
511.	2960 CONTINUE	36901
512.	ISAV=10	37001
513.	IO=IN	37101
514.	IF=ISAV	37201
515.	290 CONTINUE	37301
516.	THETA=THETA/SCALE	
517.	P=P*SCALE	06/06/69
518.	DO 2901 IX=1,ACD	38201
519.	2901 DATA1(1,IX)=DATA2(1,IX)	38301
520.	2903 IF(JS1.GT.1) GO TO 700	38401
521.	600 NSTEP=NSTEP+1	38501
522.	C READ NEXT POWER LEVEL, TIME AND SIGNAL. IF SIGNAL.LE.0 THIS IS LAST	03/25/71
523.	READ 1001, PO,TIME,ASIG	
524.	NSIG=ASIG	
525.	WRITE (3,1080)PO,TIME,NSIG	38701
526.	WRITE (3,1050)NSTEP	38801
527.	TET=TET+TIME	38901
528.	IF(PC)650,650,610	39001
529.	610 FLUX=PRATIO*PO	04/24/69
530.	FWIDE=FWID*PO+TIME	39201
531.	X=FLUX*SIGA25	39301
532.	TEX=TIME	39401
533.	P=PO+1.667ACD*E/ENFISS	
534.	TFISS=TFISS+P*TIME	39601
535.	P25=PABAR*F/(1.0+DEL)	39701
536.	XANEXF=(-T)	39801
537.	P25AR=P25AR+XJAR	39901
538.	SCALE=1./TIME	
539.	THETA=THETA*SCALE	
540.	P=P/SCALE	
541.	X=X/SCALE	
542.	P25=P25/SCALE	
543.	1110 CONTINUE	
544.	IF(NSIG) 620,620,33	01/12/71
545.	620 JS1=2	40201
546.	GO TO 33	40301
547.	650 GO 670 AND, NPA	40401
548.	J=J+K(K)	40501

549.	DO 670 J=1,JY	40601
550.	KI=LISTM(L,K)	40701
551.	FLAM(J)=FEED3(KI)	40801
552.	ARG=TIME+FEED3(KI)	40901
553.	IF(79.0-ARG)651,651,652	41001
554.	651 SEMLT(J)=0.0	41101
555.	GO TO 655	41201
556.	652 IF(ARG)653,653,654	41301
557.	653 SEMLT(J)=1.0	41401
558.	GO TO 655	41501
559.	654 SEMLT(J)=EXP(-ARG)	41601
560.	655 SSUM=0.0	41701
561.	LIM=J-1	41801
562.	SS2=0.0	41901
563.	DO 660 I=1,LIM	42001
564.	TEST=(FLAM(J)-FLAM(I))*TIME	42101
565.	IF(ABS(TEST).GT..01) GO TO 1660	05/27/69
566.	FA=TEST	05/27/69
567.	SUM=TEST	05/27/69
568.	ADD=2.	05/27/69
569.	DO 1659 II=1,4	05/27/69
570.	FA=-FA*TEST/ADD	05/27/69
571.	ADD=ADD+1.	05/27/69
572.	1659 SUM=SUM+FA	05/27/69
573.	TEST=SUM	05/27/69
574.	GO TO 660	05/27/69
575.	1660 TEST=1.0	42301
576.	SS2=SS2+DATA1(I+1,KI)	42401
577.	660 SSUM=SSUM+SEMLT(I)*DATA1(I+1,KI)*TEST	42501
578.	SSUM=SSUM+SEMLT(J)*(CATA1(1,KI)-SS2)	42601
579.	670 DATA1(1,KI)=SSUM	42701
580.	GO TO 600	43201
581.	C READ DECAY TIMES	03/25/71
582.	700 READ 1090, DT,IJKLMN,PON,SKIP,BDUM	01/12/71
583.	1090 FORMAT(9E8,3,I1,3L2)	01/12/71
584.	C DETERMINE UNITS FOR POWER LEVELS	03/25/71
585.	DO 22 L=1,9	
586.	TRY=DT(L)/100.	
587.	IF(TRY-1.) 14,14,15	
588.	14 TLN(L)=DT(L)	
589.	DATA AONE/' SEC.'/	
590.	TLH(L)=AONE	
591.	GO TO 22	
592.	15 TRY=DT(L)/3600.	
593.	IF(TRY-1.) 16,16,17	
594.	16 TLN(L)=60.*TRY	
595.	DATA ATWO/' MIN.'/	
596.	TLH(L)=ATWO	
597.	GO TO 22	
598.	17 TRY=TRY/24.	
599.	IF(TRY-1.) 18,18,19	
600.	18 TLN(L)=24.*TRY	
601.	DATA ATTHREE/' HR.'/	
602.	TLH(L)=ATTHREE	
603.	GO TO 22	
604.	19 TRY=TRY/365.	
605.	IF(TRY-1.) 20,21,21	
606.	20 TLN(L)=365.*TRY	
607.	DATA AFOUR/' DAYS.'/	
608.	TLH(L)=AFOUR	
609.	GO TO 22	
610.	21 TLN(L)=TRY	
611.	DATA AFIVE/' YRS.'/	
612.	TLH(L)=AFIVE	
613.	22 DT(L)=DT(L)/36400.	
614.	DO 410 K=1,NVA	
615.	J=NRM(K)	46501
616.	DO 410 J=1,JY	46601
617.	KI=LISTM(J,K)	46701
618.	FLAM(J)=FEED3(KI)	46801
619.	DO 408 L=1,9	
620.	ARG=DT(L)+FEED3(KI)	47001
621.	IF(79.0-ARG)409,409,405	47101
622.	409 EMLT(J,L)=0.0	47201
623.	GO TO 408	47301
624.	405 IF(ARG)406,409,411	47401
625.	409 EMLT(J,L)=1.0	47501
626.	GO TO 408	47601
627.	411 EMLT(J,L)=EXP(-ARG)	47701
628.	406 CONTINUE	47801
629.	DO 408 L=1,9	
630.	SSUM=0.0	48001
631.	LIM=J-1	48101
632.	SS2=0.0	48201
633.	DO 407 I=1,LIM	48301
634.	TEST=(FLAM(J)-FLAM(I))*DT(L)	48401
635.	IF(ABS(TEST).GT..01) GO TO 1407	
636.	SUM=TEST	
637.	FA=TEST	
638.	ADD=2.	
639.	DO 1406 II=1,4	
640.	FA=-FA*TEST/ADD	

641.	ADD=ADD+1.	
642.	SSUM=SSUM+FA	
643.	TEST=SSUM	
644.	GO TO 407	
645.	1407 TEST=1.0	48601
646.	SS2=SS2+DATA1(I+1,KI)	48701
647.	407 SSUM=SSUM+EMLT(I,L)*DATA1(I+1,KI)*TEST	48801
648.	SSUM=SSUM+EMLT(J,L)*(DATA1(1,KI)-SS2)	48901
649.	408 DATA2(L+1,1)=SSUM	49001
650.	DATA2(1,KI)=DATA1(1,KI)	49011
651.	410 CONTINUE	49101
652.	702 DO 500 L=1,10	
653.	GROS(L,1)=0.0	52801
654.	GROS(L,2)=0.0	52901
655.	GROS(L,3)=0.0	53001
656.	SAVENO(L)=0.	53002
657.	SAVEHA(L)=0.	53003
658.	500 SAVEVS(L)=0.0	
659.	DO 540 K=1,NAT	53201
660.	LF=LISTA(1,K)	53301
661.	JM=NRA(K)	53401
662.	IF(JM.EQ.0) GO TO 540	53501
663.	590 LAT=LA(LF)	53601
664.	WRITE (3,5000)CASE,FMWD,TET	53701
665.	WRITE (3,5010)LAT,REC(LF)	53801
666.	WRITE (3,5012)	53901
667.	PRINT 5013,(TLN(I),TLH(I),I=1,9)	
668.	L6=LAT.EQ.54.OR.LAT.EQ.62	54101
669.	504 DO 505 L=1,10	
670.	TOTL(L,K,1)=0.0	54701
671.	TOTL(L,K,3)=0.0	54801
672.	505 TOTL(L,K,2)=0.0	54901
673.	DO 520 J=1,JM	55001
674.	KI=LISTA(J,K)	55101
675.	DO 506 L=1,10	
676.	OUT(L)=DATA2(L,KI)*FEED3(KI)	55301
677.	506 TOTL(L,K,1)=TOTL(L,K,1)+OUT(L)	55401
678.	WRITE (3,5002)REC(KI),ML(KI),OUT	55501
679.	520 CONTINUE	56201
680.	PRINT 5003,(TOTL(L,K,1),L=1,10)	
681.	IF(LAT.EQ.36.OR.LAT.EQ.54) GO TO 5200	56302
682.	IF(LAT.EQ.35.OR.LAT.EQ.53) GO TO 5202	56303
683.	IF(LAT.EQ.34.OR.LAT.EQ.52.OR.LAT.EQ.55) GO TO 5204	56304
684.	GO TO 5206	56305
685.	5200 DO 5201 L=1,10	
686.	5201 SAVENO(L)=SAVENO(L)+TOTL(L,K,1)	56307
687.	GO TO 5205	56308
688.	5202 DO 5203 L=1,10	
689.	5203 SAVEHA(L)=SAVEHA(L)+TOTL(L,K,1)	56310
690.	GO TO 5206	56311
691.	5204 DO 5205 L=1,10	
692.	5205 SAVEVS(L)=SAVEVS(L)+TOTL(L,K,1)	56313
693.	5206 WRITE (3,5015)	56401
694.	DO 550 J=1,JM	56501
695.	KI=LISTA(J,K)	56601
696.	DO 551 L=1,10	
697.	OUT(L)=DATA2(L,KI)*.53076E-3	56801
698.	TOTL(L,K,2)=TOTL(L,K,2)+OUT(L)	56901
699.	551 TOTL(L,K,3)=TOTL(L,K,3)+OUT(L)*FLOAT(ML(KI))	57001
700.	550 WRITE (3,5002)REC(KI),ML(KI),OUT	57101
701.	PRINT 5003,(TOTL(L,K,2), L=1,10)	
702.	540 CONTINUE	57301
703.	WRITE (3,5000)CASE,FMWD,TET	57401
704.	WRITE (3,5004)	57501
705.	WRITE (3,5016)	57601
706.	PRINT 5013,(TLN(I),TLH(I),I=1,9)	
707.	DO 560 K=1,NAT	57801
708.	LF=LISTA(1,K)	57901
709.	IF(NRA(K).EQ.0) GO TO 560	58001
710.	561 PRINT 5017,LA(LF),REC(LF),(TOTL(L,K,1),L=1,10)	
711.	DO 562 L=1,10	
712.	GROS(L,1)=GROS(L,1)+TOTL(L,K,1)	58301
713.	GROS(L,2)=GROS(L,2)+TOTL(L,K,2)	58401
714.	562 CONTINUE	58501
715.	560 CONTINUE	58601
716.	PRINT 5003,(GROS(L,1),L=1,10)	
717.	DO 5600 L=1,10	
718.	5600 ARFP(L)=GROS(L,1)-SAVENO(L)-SAVEHA(L)-SAVEVS(L)	58703
719.	WRITE (3,5000) CASE,FMWD,TET	58704
720.	WRITE (3,5004)	58705
721.	WRITE (3,5016)	58706
722.	PRINT 5013,(TLN(I),TLH(I),I=1,9)	
723.	PRINT 5107,(SAVENO(I),I=1,10)	
724.	PRINT 5108,(SAVEHA(I),I=1,10)	
725.	PRINT 5109,(SAVEVS(I),I=1,10)	
726.	PRINT 5110,(ARFP(I),I=1,10)	
727.	WRITE (3,5011)CASE,FMWD,TET	58801
728.	WRITE (3,5004)	58901
729.	WRITE (3,5015)	59001
730.	WRITE (3,5016)	59101
731.	PRINT 5013,(TLN(I),TLH(I),I=1,9)	
732.	DO 570 K=1,NAT	59301

733.	LFELISTA(1,K)	59401
734.	IF(NRA(K).EQ.0) GO TO 570	59501
735.	571 PRINT 5017, LA(LF),REC(LF),(TOTL(L,K,2),L=1,10)	
736.	570 CONTINUE	59701
737.	PRINT 5003, (GROS(L,2),L=1,10)	
738.	WRITE (3,5011)CASE,FMWD,TET	60001
739.	WRITE (3,5004)	60101
740.	WRITE (3,5013)	60201
741.	WRITE (3,5016)	60301
742.	PRINT 5013, (TLN(I),TLH(I),I=1,9)	
743.	DO 580 K=1,NAT	60501
744.	LFELISTA(1,K)	60601
745.	IF(NRA(K).EQ.0) GO TO 580	60701
746.	581 DO 575 L=1,10	
747.	575 GROS(L,3)=GROS(L,3)+TOTL(L,K,3)	60901
748.	PRINT 5017, LA(LF),REC(LF),(TOTL(L,K,3),L=1,10)	
749.	580 CONTINUE	61101
750.	PRINT 5003, (GROS(L,3),L=1,10)	
751.	820 DO 800 L=1,10	
752.	GROS(L,1)=0.0	61501
753.	800 GROS(L,2)=0.0	61601
754.	IF(BSKIP) GO TO 9101	01/12/71
755.	DO 805 K=1,NAT	61701
756.	LFELISTA(1,K)	61801
757.	JMENRA(K)	61901
758.	IF(JM.EQ.0) GO TO 805	62001
759.	8000 WRITE (3,6000)CASE,FMWD,TET	62101
760.	WRITE (3,5010)LA(LF),REC(LF)	62201
761.	WRITE (3,5012)	62301
762.	PRINT 5013, (TLN(I),TLH(I),I=1,9)	
763.	DO 801 L=1,10	
764.	TOTL(L,K,1)=0.0	62601
765.	801 TOTL(L,K,2)=0.0	62701
766.	DO 803 J=1,JM	62801
767.	KI=LISTA(J,K)	62901
768.	DO 802 L=1,10	
769.	OUT(L)=DATA2(L,KI)*FEED3(KI)*FEED9(KI)	63101
770.	802 TOTL(L,K,1)=TOTL(L,K,1)+OUT(L)	63201
771.	803 WRITE (3,5002)REC(KI),ML(KI),OUT	63301
772.	PRINT 5003, (TOTL(L,K,1),L=1,10)	
773.	WRITE (3,5015)	63501
774.	DO 804 J=1,JM	63601
775.	KI=LISTA(J,K)	63701
776.	DO 8040 L=1,10	
777.	OUT(L)=DATA2(L,KI)*FEED3(KI)*FEED10(KI)	63901
778.	8040 TOTL(L,K,2)=TOTL(L,K,2)+OUT(L)	64001
779.	804 WRITE (3,5002)REC(KI),ML(KI),OUT	64101
780.	PRINT 5003, (TOTL(L,K,2),L=1,10)	
781.	805 CONTINUE	64301
782.	WRITE (3,6000)CASE,FMWD,TET	64401
783.	WRITE (3,5004)	64501
784.	WRITE (3,5016)	64601
785.	PRINT 5013, (TLN(I),TLH(I),I=1,9)	
786.	DO 806 K=1,NAT	64801
787.	LFELISTA(1,K)	64901
788.	IF(NRA(K).EQ.0) GO TO 806	65001
789.	8061 PRINT 5017, LA(LF),REC(LF),(TOTL(L,K,1),L=1,10)	
790.	DO 8062 L=1,10	
791.	GROS(L,1)=GROS(L,1)+TOTL(L,K,1)	65301
792.	GROS(L,2)=GROS(L,2)+TOTL(L,K,2)	65401
793.	8062 CONTINUE	65501
794.	806 CONTINUE	65601
795.	8064 PRINT 5003, (GROS(L,1),L=1,10)	
796.	WRITE (3,5011)CASE,FMWD,TET	66901
797.	WRITE (3,5004)	67001
798.	WRITE (3,5015)	67101
799.	WRITE (3,5016)	67201
800.	PRINT 5013, (TLN(I),TLH(I),I=1,9)	
801.	DO 807 K=1,NAT	67401
802.	LFELISTA(1,K)	67501
803.	IF(NRA(K).EQ.0) GO TO 807	67601
804.	8017 PRINT 5017, LA(LF),REC(LF),(TOTL(L,K,2),L=1,10)	
805.	807 CONTINUE	67801
806.	8073 PRINT 5003, (GROS(L,2),L=1,10)	
807.	DO 9001 L=1,NCD	
808.	9001 DATA1(1,L)=DATA2(IJ,KLMN+1,L)	
809.	9101 CONTINUE	01/12/71
810.	C RETURN TO SOURCE	03/25/71
811.	RETURN	
812.	809 DO 610 KL=1,NCD	68901
813.	610 DATA1(1,KL)=0.0	69001
814.	GO TO 9	69301
815.	END	79001


```

1. C SUBROUTINE FIRST DOES CONTAINMENT CALCULATION FOR FIRST CHAIN MEMBER 03/25/71
2. C 03/25/71
3. SUBROUTINE FIRST
4. COMMON ISCTOP,SP,CE(97),DELTAT,MASS(600,10),HOLE(2050),DKFRCT(450,
5. A3),FILTER(600,2),LAMBDA(600),SPOT(96),H(133),A(20),O(105),P(105),
6. 26(84),F(84),E(105),D(63),C(63),B(77),FOR(3),ALR(7),ILR(7,8),OLR(7)
7. 3,QUANTY(600),EXP0(35),EEXP0(35),SKIP(1200),JUMP1,JUMP4,JUMP2,JUMP3
8. 4,JUMPS,JUMP6,JUMP8,JUMP7,JUMP9,JUMP10,JUMPOF,JUMPUP
9. REAL MASS,ILR,LAMBDA
10. LOGICAL JUMP1,JUMP2,JUMP3,JUMP4,JUMP5,JUMP6,JUMP7,JUMP8
11. LOGICAL JUMP9,JUMP10,JUMPOF,JUMPUP
12. EXP0(1)=LAMBDA(ISCTOP)+OLR(1)*FILTER(ISCTOP,1)
13. EEXP0(1)=EXP(-EXP0(1)*DELTAT)
14. C 03/25/71
15. C A REPRESENTS COEFFICIENTS FOR FUEL, B FOR 1ST BARRIER, C FOR 1ST FALLOUT 03/25/71
16. C D FOR 1ST FILTER, E FOR 2ND BARRIER, F FOR 2ND FALLOUT, G FOR 2ND FILCUT 03/25/71
17. C H FOR 3RD BARRIER, O FOR 3RD FALLOUT, P FOR 3RD FILTER. 03/25/71
18. C 03/25/71
19. A(1)=MASS(ISCTOP,1)
20. MASS(ISCTOP,1)=A(1)*EEXP0(1)
21. C CONSIDER BARRIER 1 03/25/71
22. EXP0(2)=LAMBDA(ISCTOP)+FOR(1)+OLR(2)
23. EEXP0(2)=EXP(-EXP0(2)*DELTAT)
24. B(1)=ILR(1,2)*FILTER(ISCTOP,1)*A(1)/(EXP0(2)-EXP0(1))
25. B(2)=MASS(ISCTOP,2)-B(1)
26. MASS(ISCTOP,2)=B(1)*EEXP0(1)+B(2)*EEXP0(2)
27. EXP0(3)=LAMBDA(ISCTOP)
28. EEXP0(3)=EXP(-EXP0(3)*DELTAT)
29. CALL ZERO(C,1,3)
30. CALL ZERO(D,1,3)
31. IF(JUMP1.OR.JUMPUP.OR.JUMPOF) GO TO 1
32. C CONSIDER FALLOUT IN BARRIER 1 03/25/71
33. C(1)=FOR(1)*B(1)/(EXP0(3)-EXP0(1))
34. C(2)=FOR(1)*B(2)/(EXP0(3)-EXP0(2))
35. C(3)=MASS(ISCTOP,3)-C(1)-C(2)
36. MASS(ISCTOP,3)=C(1)*EEXP0(1)+C(2)*EEXP0(2)+C(3)*EEXP0(3)
37. 1 IF(JUMP2.OR.JUMPOF) GO TO 2
38. C CONSIDER FILTRATION IN BARRIER 1 03/25/71
39. ABLE=ILR(2,3)*(1.-FILTER(ISCTOP,2))
40. D(1)=ABLE*B(1)/(EXP0(3)-EXP0(1))
41. D(2)=ABLE*B(2)/(EXP0(3)-EXP0(2))
42. D(3)=MASS(ISCTOP,4)-D(1)-D(2)
43. MASS(ISCTOP,4)=D(1)*EEXP0(1)+D(2)*EEXP0(2)+D(3)*EEXP0(3)
44. 2 IF(JUMP3) GO TO 4
45. C CONSIDER BARRIER 2 03/25/71
46. EXP0(4)=LAMBDA(ISCTOP)+FOR(2)+OLR(4)
47. EEXP0(4)=EXP(-EXP0(4)*DELTAT)
48. ABLE=ILR(2,4)+ILR(3,4)*FILTER(ISCTOP,2)
49. E(1)=(ILR(1,4)*A(1)+ABLE*B(1))/(EXP0(4)-EXP0(1))
50. E(2)=ABLE*B(2)/(EXP0(4)-EXP0(2))
51. E(3)=MASS(ISCTOP,5)-E(1)-E(2)
52. MASS(ISCTOP,5)=E(1)*EEXP0(1)+E(2)*EEXP0(2)+E(3)*EEXP0(4)
53. CALL ZERO(F,1,4)
54. CALL ZERO(G,1,4)
55. IF(JUMP4.OR.JUMPUP.OR.JUMPOF) GO TO 3
56. C CONSIDER FALLOUT IN BARRIER 2 03/25/71
57. F(1)=FOR(2)*E(1)/(EXP0(3)-EXP0(1))
58. F(2)=FOR(2)*E(2)/(EXP0(3)-EXP0(2))
59. F(3)=FOR(2)*E(3)/(EXP0(3)-EXP0(4))
60. F(4)=MASS(ISCTOP,6)-F(1)-F(2)-F(3)
61. MASS(ISCTOP,6)=F(1)*EEXP0(1)+F(2)*EEXP0(2)+F(3)*EEXP0(4)+F(4)*EEXP
62. 10(3)
63. 3 IF(JUMP5.OR.JUMPOF) GO TO 4
64. C CONSIDER FILTRATION IN BARRIER 2 03/25/71
65. ABLE=ILR(4,5)*(1.-FILTER(ISCTOP,2))
66. G(1)=ABLE*E(1)/(EXP0(3)-EXP0(1))
67. G(2)=ABLE*E(2)/(EXP0(3)-EXP0(2))
68. G(3)=ABLE*E(3)/(EXP0(3)-EXP0(4))
69. G(4)=MASS(ISCTOP,7)-G(1)-G(2)-G(3)
70. MASS(ISCTOP,7)=G(1)*EEXP0(1)+G(2)*EEXP0(2)+G(3)*EEXP0(4)+G(4)*EEXP
71. 10(3)
72. 4 IF(JUMP6) GO TO 6
73. C CONSIDER BARRIER 3 03/25/71
74. CALL ZERO(O,1,5)
75. CALL ZERO(P,1,5)
76. EXP0(5)=LAMBDA(ISCTOP)+FOR(3)+OLR(6)
77. EEXP0(5)=EXP(-EXP0(5)*DELTAT)
78. ABLE=ILR(4,6)+ILR(5,6)*FILTER(ISCTOP,2)
79. H(1)=(ILR(1,6)*A(1)+ABLE*E(1))/(EXP0(5)-EXP0(1))
80. H(2)=ABLE*E(2)/(EXP0(5)-EXP0(2))
81. H(3)=ABLE*E(3)/(EXP0(5)-EXP0(4))
82. H(4)=MASS(ISCTOP,8)-H(1)-H(2)-H(3)
83. MASS(ISCTOP,8)=H(1)*EEXP0(1)+H(2)*EEXP0(2)+H(3)*EEXP0(4)+H(4)*EEXP
84. 10(5)
85. IF(JUMP7.OR.JUMPUP.OR.JUMPOF) GO TO 5
86. C CONSIDER FALLOUT IN BARRIER 3 03/25/71
87. O(1)=FOR(3)*H(1)/(EXP0(3)-EXP0(1))
88. O(2)=FOR(3)*H(2)/(EXP0(3)-EXP0(2))

```

```

89.      O(3)=FO(3)*O(3)/(EXP(3)-EXP(4))
90.      O(4)=FO(4)*O(4)/(EXP(3)-EXP(5))
91.      O(5)=F/SS(ISOTOP,9)-O(1)-O(2)-O(3)-O(4)
92.      MASS(ISOTOP,9)=O(1)*EEXP(1)+O(2)*EXP(2)+O(3)*EEXP(4)+O(4)*ELXP
93.      10(5)+O(5)*EXP(5)
94.      5 IF(JOPE,OF,JOPPOF) GO TO 6
95.      C CONSIDER FILTRATION IN BARRIER 3                                03/25/71
96.      ABLE=ILN(0.7)*(1.-FILTER(ISOTOP,2))
97.      P(1)=ABLE*H(1)/(EXP(3)-EXP(1))
98.      P(2)=ABLE*H(2)/(EXP(3)-EXP(2))
99.      P(3)=ABLE*H(3)/(EXP(3)-EXP(4))
100.     P(4)=ABLE*H(4)/(EXP(3)-EXP(5))
101.     P(5)=MASS(ISOTOP,10)-P(1)-P(2)-P(3)-P(4)
102.     MASS(ISOTOP,10)=P(1)*EEXP(1)+P(2)*EEXP(2)+P(3)*EEXP(4)+P(4)*EEX
103.     1P(5)+P(5)*EEXP(3)
104.     C QUANTY IS RELEASE RATE OF NUCLIDE TO ATMOSPHERE - CURIE*SEC/SEC    03/25/71
105.     6 QUANTY(ISOTOP)=ALR(1)*FILTER(ISOTOP,1)*MASS(ISOTOP,1)+ALR(2)*MASS(
106.     1ISOTOP,2)+ALR(4)*MASS(ISOTOP,5)+ALR(6)*MASS(ISOTOP,8)
107.     QUANTY(ISOTOP)=QUANTY(ISOTOP)+FILTER(ISOTOP,2)*(ALR(3)*MASS(ISOTOP
108.     1,2)+ALR(5)*MASS(ISOTOP,5)+ALR(7)*MASS(ISOTOP,8))
109.     C RETURN TO BAFFLE                                                03/25/71
110.     RETURN
111.     END

```

```

1. C SUBROUTINE SECOND DOES CONTAINMENT CALCULATION FOR SECOND CHAIN MEMBE03/25/71
2. SUBROUTINE SECOND
3. COMMON ISOTOP, SPACE(97), DELTAT, MASS(600,10), HOLE(2650), DKFRCT(450,
4. A3), FILTER(600,2), LAMBDA(600), SPOT(96), H(133), A(24), O(105), P(105),
5. 26(84), F(34), E(105), D(63), C(63), B(77), FOR(3), ALR(7), ILR(7,8), OLR(7)
6. 3, QUANTY(600), EXPC(35), EEXPO(35), SKIP(1200), JUMP1, JUMP4, JUMP2, JUMP3
7. 4, JUMP5, JUMP6, JUMP8, JUMP7, JUMP9, JUMP10, JUMPOF, JUMPOP
8. REAL MASS, ILR, LAMBDA
9. LOGICAL JUMP1, JUMP2, JUMP3, JUMP4, JUMP5, JUMP6, JUMP7, JUMP8
10. LOGICAL JUMP9, JUMP10, JUMPOF, JUMPOP
11. C
12. C A REPRESENTS COEFFICIENTS FOR FUEL, B FOR 1ST BARRIER, C FOR 1ST FALL03/25/71
13. C D FOR 1ST FILTER, E FOR 2ND BARRIER, F FOR 2ND FALLOUT, G FOR 2ND FIL03/25/71
14. C H FOR 3RD BARRIER, G FOR 3RD FALLOUT, P FOR 3RD FILTER.
15. C
16. EXPO(6)=LAMBDA(ISOTOP)+OLR(1)*FILTER(ISOTOP,1)
17. EEXPO(6)=EXP(-EXPO(6)*DELTAT)
18. BAKER=LAMBDA(ISOTOP-1)*DKFRCT(ISOTOP,1)
19. A(2)=BAKER*A(1)/(EXPO(6)-EXPO(1))
20. A(3)=MASS(ISOTOP,1)-A(2)
21. MASS(ISOTOP,1)=A(2)*EEXPO(1)+A(3)*EEXPO(6)
22. C CONSIDER BARRIER 1
23. EXPO(7)=LAMBDA(ISOTOP)+FOR(1)+OLR(2)
24. EEXPO(7)=EXP(-EXPO(7)*DELTAT)
25. ABLE=ILR(1,2)*FILTER(ISOTOP,1)
26. B(3)=(ABLE*A(2)+BAKER*B(1))/(EXPO(7)-EXPO(1))
27. B(4)=ABLE+A(3)/(EXPO(7)-EXPO(6))
28. B(5)=BAKER*B(2)/(EXPO(7)-EXPO(2))
29. CALL ZERO(D,4,9)
30. CALL ZERO(C,4,9)
31. IF(.NOT. JUMPOP) GO TO 1
32. B(3)=B(3)+BAKER*C(1)/(EXPO(7)-EXPO(1))
33. B(5)=B(5)+BAKER*C(2)/(EXPO(7)-EXPO(2))
34. B(6)=BAKER*C(3)/(EXPO(7)-EXPO(3))
35. GO TO 2
36. 1 B(6)=0.0
37. 2 B(7)=MASS(ISOTOP,2)-B(3)-B(4)-B(5)-B(6)
38. MASS(ISOTOP,2)=B(3)*EEXPO(1)+B(4)*EEXPO(6)+B(5)*EEXPO(2)+B(6)*EEXP
39. 10(3)+B(7)*EEXPO(7)
40. EXPO(8)=LAMBDA(ISOTOP)
41. EEXPO(8)=EXP(-EXPO(8)*DELTAT)
42. IF(JUMP1,OR,JUMPOP,OR,JUMPOF) GO TO 3
43. C CONSIDER FALLOUT IN BARRIER 1
44. C(4)=(BAKER*C(1)+FOR(1)*B(3))/(EXPO(8)-EXPO(1))
45. C(5)=FOR(1)*B(4)/(EXPO(8)-EXPO(6))
46. C(6)=(FOR(1)*B(5)+BAKER*C(2))/(EXPO(8)-EXPO(2))
47. C(7)=(FOR(1)*B(6)+BAKER*C(3))/(EXPO(8)-EXPO(3))
48. C(8)=FOR(1)*B(7)/(EXPO(8)-EXPO(7))
49. C(9)=MASS(ISOTOP,3)-C(4)-C(5)-C(6)-C(7)-C(8)
50. MASS(ISOTOP,3)=C(4)*EEXPO(1)+C(5)*EEXPO(6)+C(6)*EEXPO(2)+C(7)*EEXP
51. 10(3)+C(8)*EEXPO(7)+C(9)*EEXPO(8)
52. 3 IF(JUMP2,OR,JUMPOF) GO TO 4
53. C CONSIDER FILTRATION IN BARRIER 1
54. ABLE=ILR(2,3)*(1.-FILTER(ISOTOP,2))
55. D(4)=(ABLE*B(3)+BAKER*D(1))/(EXPO(9)-EXPO(1))
56. D(5)=ABLE*B(4)/(EXPO(9)-EXPO(6))
57. D(6)=(ABLE*B(5)+BAKER*D(2))/(EXPO(9)-EXPO(2))
58. D(7)=(ABLE*B(6)+BAKER*D(3))/(EXPO(9)-EXPO(3))
59. D(8)=ABLE*B(7)/(EXPO(9)-EXPO(7))
60. D(9)=MASS(ISOTOP,4)-D(4)-D(5)-D(6)-D(7)-D(8)
61. MASS(ISOTOP,4)=D(4)*EEXPO(1)+D(5)*EEXPO(6)+D(6)*EEXPO(2)+D(7)*EEXP
62. 10(3)+D(8)*EEXPO(7)+D(9)*EEXPO(8)
63. 4 IF(JUMP3) GO TO 9
64. C CONSIDER BARRIER 2
65. CALL ZERO(F,5,12)
66. CALL ZERO(G,5,12)
67. EXPO(9)=LAMBDA(ISOTOP)+FOR(2)+OLR(4)
68. EEXPO(9)=EXP(-EXPO(9)*DELTAT)
69. ABLE=ILR(1,4)*FILTER(ISOTOP,1)
70. CHARLY=ILR(2,4)+ILR(3,4)*FILTER(ISOTOP,2)
71. E(4)=(ABLE*A(2)+BAKER*E(1)+CHARLY*B(3))/(EXPO(9)-EXPO(1))
72. E(5)=(ABLE*A(3)+CHARLY*B(4))/(EXPO(9)-EXPO(6))
73. E(6)=CHARLY*B(5)/(EXPO(9)-EXPO(2))
74. E(7)=(CHARLY*B(6)+BAKER*E(2))/(EXPO(9)-EXPO(3))
75. E(8)=CHARLY*B(7)/(EXPO(9)-EXPO(7))
76. E(9)=BAKER*E(3)/(EXPO(9)-EXPO(4))
77. IF(.NOT. JUMPOP,OR,JUMP4) GO TO 5
78. E(4)=E(4)+BAKER*F(1)/(EXPO(9)-EXPO(1))
79. E(6)=E(6)+BAKER*F(4)/(EXPO(9)-EXPO(3))
80. E(7)=E(7)+BAKER*F(2)/(EXPO(9)-EXPO(2))
81. E(9)=E(9)+BAKER*F(3)/(EXPO(9)-EXPO(4))
82. 5 IF(.NOT. JUMPOF,OR,JUMP9) GO TO 6
83. E(4)=E(4)+BAKER*G(1)/(EXPO(9)-EXPO(1))
84. E(6)=E(6)+BAKER*G(3)/(EXPO(9)-EXPO(3))
85. E(7)=E(7)+BAKER*G(2)/(EXPO(9)-EXPO(2))
86. 6 E(10)=MASS(ISOTOP,5)-E(4)-E(5)-E(6)-E(7)-E(8)-E(9)
87. MASS(ISOTOP,5)=E(4)*EEXPO(1)+E(5)*EEXPO(6)+E(6)*EEXPO(2)+E(7)*EEXP
88. 10(3)+E(8)*EEXPO(7)+E(9)*EEXPO(4)+E(10)*EEXPO(9)

```

```

89.      IF (JUMP4, OR, JUMPOP, OR, JUMPOF) GO TO 8
90. C   CONSIDER FALLOUT IN BARRIER 2
91.      F(5)=(FOR(2)*E(4)+BAKER*F(1))/(EXP0(8)-EXP0(1))
92.      F(6)=(FOR(2)*E(5)/(EXP0(8)-EXP0(6))
93.      F(7)=(FOR(2)*E(6)+BAKER*F(4))/(EXP0(8)-EXP0(3))
94.      F(8)=(FOR(2)*E(7)+BAKER*F(2))/(EXP0(8)-EXP0(2))
95.      F(9)=(FOR(2)*E(8)/(EXP0(8)-EXP0(7))
96.      F(10)=(FOR(2)*E(9)+BAKER*F(3))/(EXP0(8)-EXP0(4))
97.      F(11)=(FOR(2)*E(10)/(EXP0(8)-EXP0(9))
98.      F(12)=MASS(ISOTOP,6)-F(5)-F(6)-F(7)-F(8)-F(9)-F(10)-F(11)
99.      MASS(ISOTOP,6)=F(5)*EEXP0(1)+F(6)*EEXP0(6)+F(7)*EEXP0(3)+F(8)*EEXP
100.     10(2)+F(9)*EEXP0(7)+F(10)*EEXP0(4)+F(11)*EEXP0(9)+F(12)*EEXP0(8)
101.      8 IF (JUMP5, OR, JUMPOP) GO TO 9
102. C   CONSIDER FILTRATION IN BARRIER 2
103.      ABLE=ILR(1,5)*(1.-FILTER(ISOTOP,2))
104.      G(5)=(ABLE*E(4)+BAKER*G(1))/(EXP0(8)-EXP0(1))
105.      G(6)=ABLE*E(5)/(EXP0(8)-EXP0(6))
106.      G(7)=(ABLE*E(6)+BAKER*G(4))/(EXP0(8)-EXP0(3))
107.      G(8)=(ABLE*E(7)+BAKER*G(2))/(EXP0(8)-EXP0(2))
108.      G(9)=ABLE*E(8)/(EXP0(8)-EXP0(7))
109.      G(10)=(ABLE*E(9)+BAKER*G(3))/(EXP0(8)-EXP0(4))
110.      G(11)=ABLE*E(10)/(EXP0(8)-EXP0(9))
111.      G(12)=MASS(ISOTOP,7)-G(5)-G(6)-G(7)-G(8)-G(9)-G(10)-G(11)
112.      MASS(ISOTOP,7)=G(5)*EEXP0(1)+G(6)*EEXP0(6)+G(7)*EEXP0(3)+G(8)*EEXP
113.     10(2)+G(9)*EEXP0(7)+G(10)*EEXP0(4)+G(11)*EEXP0(9)+G(12)*EEXP0(8)
114.      2+G(12)*EEXP0(4)
115.      9 IF (JUMP6) GO TO 13
116. C   CONSIDER BARRIER 3
117.      CALL ZERO(6,15)
118.      CALL ZERO(7,15)
119.      EXPO(10)=LAMBDA(ISOTOP)+FOR(3)+OLR(6)
120.      EEXP0(10)=EXP(-EXPO(10)*DELTAT)
121.      ABLE=ILR(1,6)*FILTER(ISOTOP,1)
122.      CHARLY=ILR(4,6)+ILR(5,6)*FILTER(ISOTOP,2)
123.      H(5)=(ABLE*A(2)+BAKER*H(1)+CHARLY*E(4))/(EXP0(10)-EXP0(1))
124.      H(6)=(ABLE*A(3)+CHARLY*E(5))/(EXP0(10)-EXP0(6))
125.      H(7)=CHARLY*E(6)/(EXP0(10)-EXP0(3))
126.      H(8)=(CHARLY*E(7)+BAKER*H(2))/(EXP0(10)-EXP0(2))
127.      H(9)=CHARLY*E(8)/(EXP0(10)-EXP0(7))
128.      H(10)=(CHARLY*E(9)+BAKER*H(3))/(EXP0(10)-EXP0(4))
129.      H(11)=CHARLY*E(10)/(EXP0(10)-EXP0(9))
130.      H(12)=BAKER*H(4)/(EXP0(10)-EXP0(5))
131.      IF (.NOT. JUMPOP, OR, JUMP7) GO TO 10
132.      H(5)=H(5)+BAKER*H(1)/(EXP0(10)-EXP0(1))
133.      H(7)=H(7)+BAKER*H(3)/(EXP0(10)-EXP0(3))
134.      H(8)=H(8)+BAKER*H(2)/(EXP0(10)-EXP0(2))
135.      H(10)=H(10)+BAKER*H(4)/(EXP0(10)-EXP0(4))
136.      H(12)=H(12)+BAKER*H(5)/(EXP0(10)-EXP0(5))
137.      10 IF (.NOT. JUMPOP, OR, JUMP10) GO TO 11
138.      H(5)=H(5)+BAKER*H(1)/(EXP0(10)-EXP0(1))
139.      H(7)=H(7)+BAKER*H(3)/(EXP0(10)-EXP0(3))
140.      H(8)=H(8)+BAKER*H(2)/(EXP0(10)-EXP0(2))
141.      H(10)=H(10)+BAKER*H(4)/(EXP0(10)-EXP0(4))
142.      11 H(13)=MASS(ISOTOP,8)-H(5)-H(6)-H(7)-H(8)-H(9)-H(10)-H(11)-H(12)
143.      MASS(ISOTOP,8)=H(5)*EEXP0(1)+H(6)*EEXP0(6)+H(7)*EEXP0(3)+H(8)*EEXP
144.     10(2)+H(9)*EEXP0(7)+H(10)*EEXP0(4)+H(11)*EEXP0(9)+H(12)*EEXP0(5)+H
145.     2(13)*EEXP0(10)
146.      IF (JUMPOF) GO TO 13
147.      IF (JUMP7, OR, JUMPOP) GO TO 12
148. C   CONSIDER FALLOUT IN BARRIER 3
149.      O(6)=(FOR(3)*H(5)+BAKER*O(1))/(EXP0(8)-EXP0(1))
150.      O(7)=FOR(3)*H(6)/(EXP0(8)-EXP0(6))
151.      O(8)=(FOR(3)*H(7)+BAKER*O(5))/(EXP0(8)-EXP0(3))
152.      O(9)=(FOR(3)*H(8)+BAKER*O(2))/(EXP0(8)-EXP0(2))
153.      O(10)=FOR(3)*H(9)/(EXP0(8)-EXP0(7))
154.      O(11)=(FOR(3)*H(10)+BAKER*O(3))/(EXP0(8)-EXP0(4))
155.      O(12)=FOR(3)*H(11)/(EXP0(8)-EXP0(9))
156.      O(13)=(FOR(3)*H(12)+BAKER*O(4))/(EXP0(8)-EXP0(5))
157.      O(14)=FOR(3)*H(13)/(EXP0(8)-EXP0(10))
158.      O(15)=MASS(ISOTOP,9)-O(6)-O(7)-O(8)-O(9)-O(10)-O(11)-O(12)-O(13)-O
159.     1(14)
160.      MASS(ISOTOP,9)=O(6)*EEXP0(1)+O(7)*EEXP0(6)+O(8)*EEXP0(3)+O(9)*EEXP
161.     10(2)+O(10)*EEXP0(7)+O(11)*EEXP0(4)+O(12)*EEXP0(9)+O(13)*EEXP0(5)+O
162.     2(14)*EEXP0(10)+O(15)*EEXP0(8)
163.      12 IF (JUMP8) GO TO 13
164. C   CONSIDER FILTRATION IN BARRIER 3
165.      ABLE=ILR(6,7)*(1.-FILTER(ISOTOP,2))
166.      P(6)=(ABLE*H(5)+BAKER*P(1))/(EXP0(8)-EXP0(1))
167.      P(7)=ABLE*H(6)/(EXP0(8)-EXP0(6))
168.      P(8)=(ABLE*H(7)+BAKER*P(5))/(EXP0(8)-EXP0(3))
169.      P(9)=(ABLE*H(8)+BAKER*P(2))/(EXP0(8)-EXP0(2))
170.      P(10)=ABLE*H(9)/(EXP0(8)-EXP0(7))
171.      P(11)=(ABLE*H(10)+BAKER*P(3))/(EXP0(8)-EXP0(4))
172.      P(12)=ABLE*H(11)/(EXP0(8)-EXP0(9))
173.      P(13)=(ABLE*H(12)+BAKER*P(4))/(EXP0(8)-EXP0(5))
174.      P(14)=ABLE*H(13)/(EXP0(8)-EXP0(10))
175.      P(15)=MASS(ISOTOP,10)-P(6)-P(7)-P(8)-P(9)-P(10)-P(11)-P(12)-P(13)
176.      1-P(14)
177.      MASS(ISOTOP,10)=P(6)*EEXP0(1)+P(7)*EEXP0(6)+P(8)*EEXP0(3)+P(9)*EEXP
178.     10(2)+P(10)*EEXP0(7)+P(11)*EEXP0(4)+P(12)*EEXP0(9)+P(13)*EEXP0(5)+
179.     2P(14)*EEXP0(10)+P(15)*EEXP0(8)
180. C   CALCULATE RELEASE RATE TO AT:OSPHERE

```

```

181.      13 QUANTY(ISOTOP)=ALR(1)*FILTER(ISOTOP,1)*MASS(ISOTOP,1)+ALR(2)*MASS(
182.      1ISOTOP,2)+ALR(4)*MASS(ISOTOP,5)+ALR(6)*MASS(ISOTOP,8)
183.      QUANTY(ISOTOP)=QUANTY(ISOTOP) +FILTER(ISOTOP,2)*(ALR(3)*MASS(ISOT
184.      OP,2)+ALR(5)*MASS(ISOTOP,5)+ALR(7)*MASS(ISOTOP,8))
185.      IF(.NOT.JUMPOF) GO TO 14
186.      QUANTY(ISOTOP)=QUANTY(ISOTOP)+BAKER*(ALR(3)/ALR(3)*MASS(ISOTOP-1,4
187.      1)+ALR(5)/ALR(5)*MASS(ISOTOP-1,7)+ALR(7)/ALR(7)*MASS(ISOTOP-1,10))
188.      C RETURN TO BAFFLE
189.      14 RETURN
190.      END

```

03/25/71

```

1. C SUBROUTINE THIRD DOES CONTAINMENT CALCULATION FOR THIRD CHAIN MEMBE03/25/71
2. SUBROUTINE THIRD 03/25/71
3. COMMON ISOTOP,SPACE(97),DELTAT,MASS(600,10),HOLE(2850),DKFRCT(450,
4. A3),FILTER(600,2),LAMBDA(600),SPOT(96),H(133),A(28),O(105),P(105),
5. 2G(84),F(84),E(105),D(63),C(63),B(77),FOR(3),ALR(7),ILR(7,8),OLR(7)
6. 3,QUANTY(600),EXPO(35),EEXPO(35),SKIP(1200),JUMP1,JUMP4,JUMP2,JUMP3
7. 4,JUMP5,JUMP6,JUMP8,JUMP7,JUMP9,JUMP10,JUMPOF,JUMPUP
8. REAL MASS,ILR,LAMBDA
9. LOGICAL JUMP1,JUMP2,JUMP3,JUMP4,JUMP5,JUMP6,JUMP7,JUMP8
10. LOGICAL JUMP9,JUMP10,JUMPOF,JUMPUP
11. C 03/25/71
12. C A REPRESENTS COEFFICIENTS FOR FUEL, B FOR 1ST BARRIER, C FOR 1ST FALL03/25/71
13. C U FOR 1ST FILTER, E FOR 2ND BARRIER, F FOR 2ND FALL03/25/71
14. C H FOR 3RD BARRIER, O FOR 3RD FALL03/25/71
15. C 03/25/71
16. EXPO(11)=LAMBDA(ISOTOP)+OLR(1)*FILTER(ISOTOP,1)
17. EEXPO(11)=EXP(-EXPO(11)*DELTAT)
18. ABLE=LAMBDA(ISOTOP-2)*DKFRCT(ISOTOP,2)
19. BAKER=LAMBDA(ISOTOP-1)*DKFRCT(ISOTOP,1)
20. A(4)=(ABLE*A(1)+BAKER*A(2))/(EXPO(11)-EXPO(1))
21. A(5)=BAKER*A(3)/(EXPO(11)-EXPO(6))
22. A(6)=MASS(ISOTOP,1)-A(4)-A(5)
23. MASS(ISOTOP,1)=A(4)*EEXPO(11)+A(5)*EEXPO(6)+A(6)*EEXPO(11)
24. C CONSIDER BARRIER 1 03/25/71
25. EXPO(12)=LAMBDA(ISOTOP)+FOR(1)+OLR(2)
26. CHARLY=ILR(1,2)*FILTER(ISOTOP,1)
27. B(8)=(ABLE*B(1)+BAKER*B(3)+CHARLY*A(4))/(EXPO(12)-EXPO(1))
28. EEXPO(12)=EXP(-EXPO(12)*DELTAT)
29. B(9)=(BAKER*B(4)+CHARLY*A(5))/(EXPO(12)-EXPO(6))
30. B(10)=CHARLY*A(6)/(EXPO(12)-EXPO(11))
31. B(11)=BAKER*B(6)/(EXPO(12)-EXPO(3))
32. B(13)=(ABLE*B(2)+BAKER*B(5))/(EXPO(12)-EXPO(2))
33. B(14)=BAKER*B(7)/(EXPO(12)-EXPO(7))
34. CALL ZERO(C,10,13)
35. CALL ZERO(C,10,13)
36. IF(.NOT.JUMPUP.OR.JUMP1) GO TO 1
37. B(8)=B(8)+(ABLE*C(1)+BAKER*C(4))/(EXPO(12)-EXPO(1))
38. B(9)=B(9)+BAKER*C(5)/(EXPO(12)-EXPO(6))
39. B(11)=B(11)+(ABLE*C(3)+BAKER*C(7))/(EXPO(12)-EXPO(3))
40. B(12)=BAKER*C(9)/(EXPO(12)-EXPO(8))
41. B(13)=B(13)+(ABLE*C(2)+BAKER*C(6))/(EXPO(12)-EXPO(2))
42. B(14)=B(14)+BAKER*C(8)/(EXPO(12)-EXPO(7))
43. GO TO 2
44. 1 B(12)=0.0
45. 2 U(15)=MASS(ISOTOP,2)-B(6)-B(9)-B(10)-B(11)-B(12)-B(13)-B(14)
46. MASS(ISOTOP,2)=B(8)*EEXPO(1)+U(9)*EEXPO(6)+B(10)*EEXPO(11)+B(11)*E
47. 1EXPO(3)+B(12)*EEXPO(3)+B(13)*EEXPO(2)+B(14)*EEXPO(7)+B(15)*EEXPO(1
48. 22)
49. EXPO(13)=LAMBDA(ISOTOP)
50. EEXPO(13)=EXP(-EXPO(13)*DELTAT)
51. IF(JUMP1.OR.JUMPUP.OR.JUMPOF) GO TO 3
52. C CONSIDER FALLOUT IN BARRIER 1 03/25/71
53. C(10)=(FOR(1)*B(8)+ABLE*C(1)+BAKER*C(4))/(EXPO(13)-EXPO(1))
54. C(11)=(FOR(1)*B(9)+BAKER*C(5))/(EXPO(13)-EXPO(6))
55. C(12)=(FOR(1)*B(10))/(EXPO(13)-EXPO(11))
56. C(13)=(FOR(1)*B(11)+ABLE*C(3)+BAKER*C(7))/(EXPO(13)-EXPO(3))
57. C(14)=(FOR(1)*B(12)+BAKER*C(9))/(EXPO(13)-EXPO(8))
58. C(15)=(FOR(1)*B(13)+ABLE*C(2)+BAKER*C(6))/(EXPO(13)-EXPO(2))
59. C(16)=(FOR(1)*B(14)+BAKER*C(8))/(EXPO(13)-EXPO(7))
60. C(17)=FOR(1)*B(15)/(EXPO(13)-EXPO(12))
61. C(18)=MASS(ISOTOP,3)-C(10)-C(11)-C(12)-C(13)-C(14)-C(15)-C(16)-C(1
62. 17)
63. MASS(ISOTOP,3)=C(10)*EEXPO(1)+C(11)*EEXPO(6)+C(12)*EEXPO(11)+C(13)
64. 1EEXPO(3)+C(14)*EEXPO(8)+C(15)*EEXPO(2)+C(16)*EEXPO(7)+C(17)*EEXPO
65. 10(12)+C(18)*EEXPO(13)
66. 3 IF(JUMP2.OR.JUMPOF) GO TO 4
67. C CONSIDER FILTRATION IN BARRIER 1 03/25/71
68. CHARLY=ILR(2,3)*(1.-FILTER(ISOTOP,2))
69. D(10)=(CHARLY*B(8)+ABLE*D(1)+BAKER*D(4))/(EXPO(13)-EXPO(1))
70. D(11)=(CHARLY*B(9)+BAKER*D(5))/(EXPO(13)-EXPO(6))
71. D(12)=CHARLY*B(10)/(EXPO(13)-EXPO(11))
72. D(13)=(CHARLY*B(11)+ABLE*D(3)+BAKER*D(7))/(EXPO(13)-EXPO(3))
73. D(14)=(CHARLY*B(12)+BAKER*D(9))/(EXPO(13)-EXPO(8))
74. D(15)=(CHARLY*B(13)+ABLE*D(2)+BAKER*D(6))/(EXPO(13)-EXPO(2))
75. D(16)=(CHARLY*B(14)+BAKER*D(8))/(EXPO(13)-EXPO(7))
76. D(17)=CHARLY*B(15)/(EXPO(13)-EXPO(12))
77. D(18)=MASS(ISOTOP,4)-D(10)-D(11)-D(12)-D(13)-D(14)-D(15)-D(16)-D(1
78. 17)
79. MASS(ISOTOP,4)=D(10)*EEXPO(1)+D(11)*EEXPO(6)+D(12)*EEXPO(11)+D(13)
80. 1EEXPO(3)+D(14)*EEXPO(8)+D(15)*EEXPO(2)+D(16)*EEXPO(7)+D(17)*EEXPO
81. 2(12)+D(18)*EEXPO(13)
82. 4 IF(JUMP3) GO TO 8
83. C CONSIDER BARRIER 2 03/25/71
84. CALL ZERO(F,13,24)
85. CALL ZERO(G,13,24)
86. EXPO(14)=LAMBDA(ISOTOP)+FOR(2)+OLR(4)
87. EEXPO(14)=EXP(-EXPO(14)*DELTAT)
88. CHARLY=ILR(1,4)*FILTER(ISOTOP,1)

```

```

89. DCG=ILR(2,4)+ILR(3,4)+FILTER(ISOTOP,2)
90. E(11)=(CHARLY*A(4)+DOG*B(4)+ABLE*F(1)+BAKER*E(4))/(EXPO(14)-EXPO(1
91. 1))
92. L(12)=(CHARLY*A(5)+DOG*B(5)+BAKER*E(5))/(EXPO(14)-EXPO(6))
93. E(13)=(CHARLY*A(6)+DOG*B(10))/(EXPO(14)-EXPO(11))
94. E(14)=(DOG*B(11)+BAKER*E(4))/(EXPO(14)-EXPO(3))
95. L(15)=DOG*B(12)/(EXPO(14)-EXPO(8))
96. E(16)=(DOG*B(13)+ABLE*E(2)+BAKER*E(7))/(EXPO(14)-EXPO(2))
97. E(17)=(DOG*B(14)+BAKER*E(4))/(EXPO(14)-EXPO(7))
98. E(18)=DOG*B(15)/(EXPO(14)-EXPO(12))
99. E(19)=(ABLE*E(3)+BAKER*E(9))/(EXPO(14)-EXPO(4))
100. E(20)=BAKER*E(10)/(EXPO(14)-EXPO(9))
101. IF(.NOT.JUMPUP.OR.JUMP4) GO TO 5
102. E(11)=E(11)+(ABLE*F(1)+BAKER*F(5))/(EXPO(14)-EXPO(1))
103. E(12)=E(12)+BAKER*F(6)/(EXPO(14)-EXPO(6))
104. E(14)=E(14)+(ABLE*F(4)+BAKER*F(7))/(EXPO(14)-EXPO(3))
105. E(15)=E(15)+BAKER*F(12)/(EXPO(14)-EXPO(8))
106. E(16)=E(16)+(ABLE*F(2)+BAKER*F(8))/(EXPO(14)-EXPO(2))
107. E(17)=E(17)+BAKER*F(9)/(EXPO(14)-EXPO(7))
108. E(19)=E(19)+(ABLE*F(3)+BAKER*F(10))/(EXPO(14)-EXPO(4))
109. E(20)=E(20)+BAKER*F(11)/(EXPO(14)-EXPO(9))
110. 5 IF(.NOT.JUMPUP.OR.JUMP9) GO TO 6
111. E(11)=E(11)+(ABLE*D(1)+BAKER*L(4))/(EXPO(14)-EXPO(1))
112. E(12)=E(12)+BAKER*D(5)/(EXPO(14)-EXPO(6))
113. E(14)=E(14)+(ABLE*D(3)+BAKER*L(7))/(EXPO(14)-EXPO(3))
114. F(15)=E(15)+BAKER*D(9)/(EXPO(14)-EXPO(8))
115. E(16)=E(16)+(ABLE*D(2)+BAKER*L(6))/(EXPO(14)-EXPO(2))
116. E(17)=E(17)+BAKER*D(3)/(EXPO(14)-EXPO(7))
117. 6 F(21)=MASS(ISOTOP,5)*E(11)-E(12)-E(13)-E(14)-E(15)-E(16)-E(17)-E(1
118. 18)-E(19)-E(20)
119. MASS(ISOTOP,5)=E(11)*EEXPO(1)+E(12)*EEXPO(6)+E(13)*EEXPO(11)+E(14)
120. 1+EEXPO(3)+E(15)*EEXPO(8)+E(16)*EEXPO(2)+E(17)*EEXPO(7)+E(18)*EEXPO
121. 2(12)+E(19)*EEXPO(4)+E(20)*EEXPO(9)+E(21)*EEXPO(14)
122. IF(JUMP4.OR.JUMPUP.OR.JUMPOF) GO TO 7
123. C CONSIDER FALLOUT IN BARRIER 2 03/25/71
124. F(13)=(FOR(2)*E(11)+ABLE*F(1)+BAKER*F(5))/(EXPO(13)-EXPO(1))
125. F(14)=(FOR(2)*E(12)+BAKER*F(6))/(EXPO(13)-EXPO(6))
126. F(15)=FOR(2)*E(13)/(EXPO(13)-EXPO(11))
127. F(16)=(FOR(2)*E(14)+ABLE*F(4)+BAKER*F(7))/(EXPO(13)-EXPO(3))
128. F(17)=(FOR(2)*E(15)+BAKER*F(12))/(EXPO(13)-EXPO(8))
129. F(18)=(FOR(2)*E(16)+ABLE*F(2)+BAKER*F(8))/(EXPO(13)-EXPO(2))
130. F(19)=(FOR(2)*E(17)+BAKER*F(9))/(EXPO(13)-EXPO(7))
131. F(20)=FOR(2)*E(18)/(EXPO(13)-EXPO(12))
132. F(21)=(FOR(2)*E(19)+ABLE*F(3)+BAKER*F(10))/(EXPO(13)-EXPO(4))
133. F(22)=(FOR(2)*E(20)+BAKER*F(11))/(EXPO(13)-EXPO(9))
134. F(23)=FOR(2)*E(21)/(EXPO(13)-EXPO(14))
135. F(24)=MASS(ISOTOP,6)-F(13)-F(14)-F(15)-F(16)-F(17)-F(18)-F(19)-F(2
136. 0)-F(21)-F(22)-F(23)
137. MASS(ISOTOP,6)=F(13)*EEXPO(1)+F(14)*EEXPO(6)+F(15)*EEXPO(11)+F(16)
138. 1+EEXPO(3)+F(17)*EEXPO(8)+F(18)*EEXPO(2)+F(19)*EEXPO(7)+F(20)*EEXPO
139. 2(12)+F(21)*EEXPO(4)+F(22)*EEXPO(9)+F(23)*EEXPO(14)+F(24)*EEXPO(13)
140. 7 IF(JUMPOF.OR.JUMP5) GO TO 8
141. C CONSIDER FILTRATION IN BARRIER 2 03/25/71
142. CHARLY=ILR(4,5)*(1.-FILTER(ISOTOP,2))
143. G(13)=(CHARLY*E(11)+ABLE*G(1)+BAKER*G(5))/(EXPO(13)-EXPO(1))
144. G(14)=(CHARLY*E(12)+BAKER*G(6))/(EXPO(13)-EXPO(6))
145. G(15)=CHARLY*E(13)/(EXPO(13)-EXPO(11))
146. G(16)=(CHARLY*E(14)+ABLE*G(4)+BAKER*G(7))/(EXPO(13)-EXPO(3))
147. G(17)=(CHARLY*E(15)+BAKER*G(12))/(EXPO(13)-EXPO(8))
148. G(18)=(CHARLY*E(16)+ABLE*G(2)+BAKER*G(8))/(EXPO(13)-EXPO(2))
149. G(19)=(CHARLY*E(17)+BAKER*G(9))/(EXPO(13)-EXPO(7))
150. G(20)=CHARLY*E(18)/(EXPO(13)-EXPO(12))
151. G(21)=(CHARLY*E(19)+ABLE*G(3)+BAKER*G(10))/(EXPO(13)-EXPO(4))
152. G(22)=(CHARLY*E(20)+BAKER*G(11))/(EXPO(13)-EXPO(9))
153. G(23)=CHARLY*E(21)/(EXPO(13)-EXPO(14))
154. G(24)=MASS(ISOTOP,7)-G(13)-G(14)-G(15)-G(16)-G(17)-G(18)-G(19)-G(2
155. 0)-G(21)-G(22)-G(23)
156. MASS(ISOTOP,7)=G(13)*EEXPO(1)+G(14)*EEXPO(6)+G(15)*EEXPO(11)+G(16)
157. 1+EEXPO(3)+G(17)*EEXPO(8)+G(18)*EEXPO(2)+G(19)*EEXPO(7)+G(20)*EEXPO
158. 2(12)+G(21)*EEXPO(4)+G(22)*EEXPO(9)+G(23)*EEXPO(14)+G(24)*EEXPO(13)
159. 8 IF(JUMP6) GO TO 12
160. C CONSIDER BARRIER 3 03/25/71
161. CALL ZERO(0,16,30)
162. CALL ZERO(0,16,30)
163. EXPO(15)=LAM30A(ISOTOP)+FOR(3)+OLR(6)
164. EXXPO(15)=EXP(-EXPO(15)+DELTA)
165. CHARLY=ILR(1,6)+FILTER(ISOTOP,1)
166. DOG=ILR(4,6)+ILR(5,6)+FILTER(ISOTOP,2)
167. H(14)=(CHARLY*A(4)+DOG*E(11)+ABLE*H(1)+BAKER*H(5))/(EXPO(15)-EXPO(
168. 1))
169. H(15)=(CHARLY*A(5)+DOG*E(12)+BAKER*H(6))/(EXPO(15)-EXPO(6))
170. H(16)=(CHARLY*A(6)+DOG*E(13))/(EXPO(15)-EXPO(11))
171. H(17)=(DOG*E(14)+BAKER*H(7))/(EXPO(15)-EXPO(3))
172. H(18)=DOG*E(15)/(EXPO(15)-EXPO(8))
173. H(19)=(DOG*E(16)+ABLE*H(2)+BAKER*H(8))/(EXPO(15)-EXPO(2))
174. H(20)=(DOG*E(17)+BAKER*H(9))/(EXPO(15)-EXPO(7))
175. H(21)=DOG*E(18)/(EXPO(15)-EXPO(12))
176. H(22)=(DOG*E(19)+ABLE*H(3)+BAKER*H(10))/(EXPO(15)-EXPO(4))
177. H(23)=(DOG*E(20)+BAKER*H(11))/(EXPO(15)-EXPO(9))
178. H(24)=DOG*E(21)/(EXPO(15)-EXPO(14))
179. H(25)=(ABLE*H(4)+BAKER*H(12))/(EXPO(15)-EXPO(5))
180. H(26)=BAKER*H(13)/(EXPO(15)-EXPO(10))

```

```

181. IF(.NOT.JUMPUF.OR.JUMPF) GO TO 9
182. H(14)=H(14)+(ABLE*G(1)+BAKER*G(6))/(EXP0(15)-EXP0(1))
183. H(15)=H(15)+(ABLE*G(7)+BAKER*G(11))/(EXP0(15)-EXP0(6))
184. H(17)=H(17)+(ABLE*G(5)+BAKER*G(8))/(EXP0(15)-EXP0(3))
185. H(18)=H(18)+(ABLE*G(15)+BAKER*G(9))/(EXP0(15)-EXP0(8))
186. H(19)=H(19)+(ABLE*G(2)+BAKER*G(13))/(EXP0(15)-EXP0(2))
187. H(20)=H(20)+(ABLE*G(10)+BAKER*G(14))/(EXP0(15)-EXP0(7))
188. H(22)=H(22)+(ABLE*G(3)+BAKER*G(11))/(EXP0(15)-EXP0(4))
189. H(23)=H(23)+(ABLE*G(12)+BAKER*G(13))/(EXP0(15)-EXP0(9))
190. H(25)=H(25)+(ABLE*G(4)+BAKER*G(13))/(EXP0(15)-EXP0(5))
191. H(26)=H(26)+(ABLE*G(14)+BAKER*G(14))/(EXP0(15)-EXP0(10))
192. 9 IF(.NOT.JUMPUF.OR.JUMPF) GO TO 10
193. H(14)=H(14)+(ABLE*G(1)+BAKER*G(5))/(EXP0(15)-EXP0(1))
194. H(15)=H(15)+(ABLE*G(6)+BAKER*G(6))/(EXP0(15)-EXP0(6))
195. H(17)=H(17)+(ABLE*G(4)+BAKER*G(7))/(EXP0(15)-EXP0(3))
196. H(18)=H(18)+(ABLE*G(12)+BAKER*G(12))/(EXP0(15)-EXP0(8))
197. H(19)=H(19)+(ABLE*G(2)+BAKER*G(8))/(EXP0(15)-EXP0(2))
198. H(20)=H(20)+(ABLE*G(9)+BAKER*G(9))/(EXP0(15)-EXP0(7))
199. H(22)=H(22)+(ABLE*G(3)+BAKER*G(10))/(EXP0(15)-EXP0(4))
200. H(23)=H(23)+(ABLE*G(11)+BAKER*G(11))/(EXP0(15)-EXP0(9))
201. 10 H(27)=MASS(ISOTOP,8)-H(14)-H(15)-H(16)-H(17)-H(18)-H(19)-H(20)-H(2
202. 11)-H(22)-H(23)-H(24)-H(25)-H(26)
203. MASS(ISOTOP,8)=H(14)*EEXP0(1)+H(15)*EEXP0(6)+H(16)*EEXP0(11)+H(17)
204. 1+EEXP0(3)+H(18)*EEXP0(8)+H(19)*EEXP0(2)+H(20)*EEXP0(7)+H(21)*EEXP0
205. 2(12)+H(22)*EEXP0(4)+H(23)*EEXP0(9)+H(24)*EEXP0(14)+H(25)*EEXP0(5)+
206. 3H(26)*EEXP0(10)+H(27)*EEXP0(15)
207. IF(JUMPUF.OR.JUMPUF.OR.JUMPUF) GO TO 11
208. C CONSIDER FALLOUT IN BARRIER 3 03/25/71
209. 0(16)=(FOR(3)*H(14)+ABLE*G(1)+BAKER*G(6))/(EXP0(13)-EXP0(1))
210. 0(17)=(FOR(3)*H(15)+ABLE*G(7)+BAKER*G(11))/(EXP0(13)-EXP0(6))
211. 0(18)=(FOR(3)*H(16)+ABLE*G(11)+BAKER*G(11))/(EXP0(13)-EXP0(11))
212. 0(19)=(FOR(3)*H(17)+ABLE*G(5)+BAKER*G(8))/(EXP0(13)-EXP0(3))
213. 0(20)=(FOR(3)*H(18)+ABLE*G(15)+BAKER*G(9))/(EXP0(13)-EXP0(8))
214. 0(21)=(FOR(3)*H(19)+ABLE*G(2)+BAKER*G(13))/(EXP0(13)-EXP0(2))
215. 0(22)=(FOR(3)*H(20)+ABLE*G(10)+BAKER*G(14))/(EXP0(13)-EXP0(7))
216. 0(23)=(FOR(3)*H(21)+ABLE*G(3)+BAKER*G(11))/(EXP0(13)-EXP0(4))
217. 0(24)=(FOR(3)*H(22)+ABLE*G(12)+BAKER*G(13))/(EXP0(13)-EXP0(9))
218. 0(25)=(FOR(3)*H(23)+ABLE*G(4)+BAKER*G(13))/(EXP0(13)-EXP0(5))
219. 0(26)=(FOR(3)*H(24)+ABLE*G(14)+BAKER*G(14))/(EXP0(13)-EXP0(10))
220. 0(27)=(FOR(3)*H(25)+ABLE*G(14)+BAKER*G(14))/(EXP0(13)-EXP0(10))
221. 0(28)=(FOR(3)*H(26)+ABLE*G(14)+BAKER*G(14))/(EXP0(13)-EXP0(10))
222. 0(29)=(FOR(3)*H(27)+ABLE*G(14)+BAKER*G(14))/(EXP0(13)-EXP0(10))
223. 0(30)=MASS(ISOTOP,9)-SUMZ(0,16,29)
224. MASS(ISOTOP,9)=0(16)*EEXP0(1)+0(17)*EEXP0(6)+0(18)*EEXP0(11)+0(19)
225. 1+EEXP0(3)+0(20)*EEXP0(8)+0(21)*EEXP0(2)+0(22)*EEXP0(7)+0(23)*EEXP0
226. 2(12)+0(24)*EEXP0(4)+0(25)*EEXP0(9)+0(26)*EEXP0(14)+0(27)*EEXP0(5)+
227. 30(28)*EEXP0(10)+0(29)*EEXP0(15)+0(30)*EEXP0(13)
228. 11 IF(JUMPUF.OR.JUMPUF) GO TO 12
229. C CONSIDER FILTRATION IN BARRIER 3 03/25/71
230. CHARLY=ILR(6,7)*(1.-FILTER(ISOTOP,2))
231. P(16)=(CHARLY*H(14)+ABLE*P(1)+BAKER*P(6))/(EXP0(13)-EXP0(1))
232. P(17)=(CHARLY*H(15)+ABLE*P(7)+BAKER*P(11))/(EXP0(13)-EXP0(6))
233. P(18)=CHARLY*H(16)/(EXP0(13)-EXP0(11))
234. P(19)=(CHARLY*H(17)+ABLE*P(5)+BAKER*P(8))/(EXP0(13)-EXP0(3))
235. P(20)=(CHARLY*H(18)+ABLE*P(15)+BAKER*P(9))/(EXP0(13)-EXP0(8))
236. P(21)=(CHARLY*H(19)+ABLE*P(2)+BAKER*P(13))/(EXP0(13)-EXP0(2))
237. P(22)=(CHARLY*H(20)+ABLE*P(10)+BAKER*P(14))/(EXP0(13)-EXP0(7))
238. P(23)=CHARLY*H(21)/(EXP0(13)-EXP0(12))
239. P(24)=(CHARLY*H(22)+ABLE*P(3)+BAKER*P(11))/(EXP0(13)-EXP0(4))
240. P(25)=(CHARLY*H(23)+ABLE*P(12)+BAKER*P(12))/(EXP0(13)-EXP0(9))
241. P(26)=CHARLY*H(24)/(EXP0(13)-EXP0(14))
242. P(27)=(CHARLY*H(25)+ABLE*P(4)+BAKER*P(13))/(EXP0(13)-EXP0(5))
243. P(28)=(CHARLY*H(26)+ABLE*P(14)+BAKER*P(14))/(EXP0(13)-EXP0(10))
244. P(29)=CHARLY*H(27)/(EXP0(13)-EXP0(15))
245. P(30)=MASS(ISOTOP,10)-SUMZ(P,16,29)
246. MASS(ISOTOP,10)=P(16)*EEXP0(1)+P(17)*EEXP0(6)+P(18)*EEXP0(11)+P(19)
247. 1+EEXP0(3)+P(20)*EEXP0(8)+P(21)*EEXP0(2)+P(22)*EEXP0(7)+P(23)*EEXP0
248. 20(12)+P(24)*EEXP0(4)+P(25)*EEXP0(9)+P(26)*EEXP0(14)+P(27)*EEXP0(5)
249. 3+P(28)*EEXP0(10)+P(29)*EEXP0(15)+P(30)*EEXP0(13)
250. C CALCULATE RELEASE RATE TO ATMOSPHERE 03/25/71
251. 12 QUANTY(ISOTOP)=ALR(1)*FILTER(ISOTOP,1)*MASS(ISOTOP,1)+(ALR(2)+ALR(
252. 13)*FILTER(ISOTOP,2))*MASS(ISOTOP,2)+(ALR(4)+ALR(5))*FILTER(ISOTOP,2
253. 2))*MASS(ISOTOP,5)+(ALR(6)+ALR(7))*FILTER(ISOTOP,2))*MASS(ISOTOP,8)
254. IF(.NOT.JUMPUF) GO TO 13
255. QUANTY(ISOTOP)=QUANTY(ISOTOP)+ALR(3)/ALR(3)*(ABLE*MASS(ISOTOP-2,4)
256. 1+BAKER*MASS(ISOTOP-1,4))+ALR(5)/ALR(5)*(ABLE*MASS(ISOTOP-2,7)+BAKER
257. 2*MASS(ISOTOP-1,7))+ALR(7)/ALR(7)*(ABLE*MASS(ISOTOP-2,10)+BAKER*MA
258. 3SS(ISOTOP-1,10))
259. C RETURN TO BAFFLE 03/25/71
260. 13 RETURN
261. END

```



```

1. C SUBROUTINE FOURTH DOES CONTAINMENT CALCULATION FOR FOURTH CHAIN MEMBE03/25/71
2. SUBROUTINE FOURTH 03/25/71
3. COMMON ISOTOP,SPACE(97),DELTAT,MASS(600,10),HOLE(2050),DKFRCT(450,
4. A3),FILTER(600,2),LAMBDA(600),SPOT(96),H(133),A(24),O(105),P(105),
5. 2G(84),F(84),E(105),D(63),C(63),B(77),FOR(3),ALR(7),ILR(7,8),OLR(7)
6. 3,QUANTITY(600),EXPO(35),EEXPO(35),SKIP(1200),JUMP1,JUMP4,JUMP2,JUMP3
7. 4,JUMP5,JUMP6,JUMP8,JUMP7,JUMP9,JUMP10,JUMPOF,JUMPUP
8. REAL MASS,ILR,LAMBDA
9. LOGICAL JUMP1,JUMP2,JUMP3,JUMP4,JUMP5,JUMP6,JUMP7,JUMP8
10. LOGICAL JUMP9,JUMP10,JUMPOF,JUMPUP
11. C 03/25/71
12. C A REPRESENTS COEFFICIENTS FOR FUEL, B FOR 1ST BARRIER, C FOR 1ST FALL03/25/71
13. C D FOR 1ST FILTER, E FOR 2ND BARRIER, F FOR 2ND FALL03/25/71
14. C H FOR 3RD BARRIER, O FOR 3RD FALL03/25/71
15. C 03/25/71
16. EXPO(16)=LAMBDA(ISOTOP)+OLR(1)*FILTER(ISOTOP,1)
17. EEXPO(16)=EXP(-EXPO(16)*DELTAT)
18. ABLE=LAMBDA(ISOTOP-3)*DKFRCT(ISOTOP,3)
19. BAKER=LAMBDA(ISOTOP-2)*DKFRCT(ISOTOP,2)
20. CHARLY=LAMBDA(ISOTOP-1)*DKFRCT(ISOTOP,1)
21. A(7)=(ABLE*A(1)+BAKER*A(2)+CHARLY*A(4))/(EXPO(16)-EXPO(1))
22. A(8)=(BAKER*A(3)+CHARLY*A(5))/(EXPO(16)-EXPO(6))
23. A(9)=CHARLY*A(6)/(EXPO(16)-EXPO(11))
24. A(10)=MASS(ISOTOP,1)-A(7)-A(8)-A(9)
25. MASS(ISOTOP,1)=A(7)*EEXPO(1)+A(8)*EEXPO(6)+A(9)*EEXPO(11)+A(10)*EE
26. XPO(16)
27. C CONSIDER BARRIER 1 03/25/71
28. EXPO(17)=LAMBDA(ISOTOP)+FOR(1)+OLR(2)
29. EEXPO(17)=EXP(-EXPO(17)*DELTAT)
30. DOG=ILR(1,2)*FILTER(ISOTOP,1)
31. B(16)=(ABLE*B(1)+BAKER*B(3)+CHARLY*B(8)+DOG*A(7))/(EXPO(17)-EXPO(1)
32. 1))
33. B(17)=(BAKER*B(4)+CHARLY*B(9)+DOG*A(8))/(EXPO(17)-EXPO(6))
34. B(18)=(CHARLY*B(10)+DOG*A(9))/(EXPO(17)-EXPO(11))
35. B(19)=DOG*A(10)/(EXPO(17)-EXPO(16))
36. B(20)=(BAKER*B(6)+CHARLY*B(11))/(EXPO(17)-EXPO(3))
37. B(21)=CHARLY*B(12)/(EXPO(17)-EXPO(8))
38. B(23)=(ABLE*B(2)+BAKER*B(5)+CHARLY*B(13))/(EXPO(17)-EXPO(2))
39. B(24)=(BAKER*B(7)+CHARLY*B(14))/(EXPO(17)-EXPO(7))
40. B(25)=CHARLY*B(15)/(EXPO(17)-EXPO(12))
41. CALL ZERO (C,19,30)
42. CALL ZERO(D,19,30)
43. IF(.NOT. JUMPUP.OR. JUMP1) GO TO 1
44. B(16)=B(16)+(ABLE*C(1)+BAKER*C(4)+CHARLY*C(10))/(EXPO(17)-EXPO(1))
45. B(17)=B(17)+(BAKER*C(5)+CHARLY*C(11))/(EXPO(17)-EXPO(6))
46. B(18)=B(18)+CHARLY*C(12)/(EXPO(17)-EXPO(11))
47. B(20)=B(20)+(ABLE*C(3)+BAKER*C(7)+CHARLY*C(13))/(EXPO(17)-EXPO(3))
48. B(21)=B(21)+(BAKER*C(9)+CHARLY*C(14))/(EXPO(17)-EXPO(8))
49. B(22)=CHARLY*C(18)/(EXPO(17)-EXPO(13))
50. B(23)=B(23)+(ABLE*C(2)+BAKER*C(6)+CHARLY*C(15))/(EXPO(17)-EXPO(2))
51. B(24)=B(24)+(BAKER*C(8)+CHARLY*C(16))/(EXPO(17)-EXPO(7))
52. B(25)=B(25)+CHARLY*C(17)/(EXPO(17)-EXPO(12))
53. GO TO 2
54. 1 B(22)=0.0
55. 2 B(26)=MASS(ISOTOP,2)-SUMZ(B,16,25)
56. MASS(ISOTOP,2)=B(16)*EEXPO(1)+B(17)*EEXPO(6)+B(18)*EEXPO(11)+B(19)
57. *EEXPO(16)+B(20)*EEXPO(3)+B(21)*EEXPO(8)+B(22)*EEXPO(13)+B(23)*EEX
58. 2PO(2)+B(24)*EEXPO(7)+B(25)*EEXPO(12)+B(26)*EEXPO(17)
59. EXPO(18)=LAMBDA(ISOTOP)
60. EEXPO(18)=EXP(-EXPO(18)*DELTAT)
61. IF(JUMP1.OR. JUMPUP.OR. JUMPOF) GO TO 3
62. C CONSIDER FALL0UT IN BARRIER 1 03/25/71
63. C(19)=(FOR(1)*B(16)+ABLE*C(1)+BAKER*C(4)+CHARLY*C(10))/(EXPO(18)-E
64. XPO(1))
65. C(20)=(FOR(1)*B(17)+BAKER*C(5)+CHARLY*C(11))/(EXPO(18)-EXPO(6))
66. C(21)=(FOR(1)*B(18)+CHARLY*C(12))/(EXPO(18)-EXPO(11))
67. C(22)=(FOR(1)*B(19))/(EXPO(18)-EXPO(16))
68. C(23)=(FOR(1)*B(20)+ABLE*C(3)+BAKER*C(7)+CHARLY*C(13))/(EXPO(18)-E
69. XPO(3))
70. C(24)=(FOR(1)*B(21)+BAKER*C(9)+CHARLY*C(14))/(EXPO(18)-EXPO(8))
71. C(25)=(FOR(1)*B(22)+CHARLY*C(18))/(EXPO(18)-EXPO(13))
72. C(26)=(FOR(1)*B(23)+ABLE*C(2)+BAKER*C(6)+CHARLY*C(15))/(EXPO(18)-E
73. XPO(2))
74. C(27)=(FOR(1)*B(24)+BAKER*C(8)+CHARLY*C(16))/(EXPO(18)-EXPO(7))
75. C(28)=(FOR(1)*B(25)+CHARLY*C(17))/(EXPO(18)-EXPO(12))
76. C(29)=FOR(1)*B(26)/(EXPO(18)-EXPO(17))
77. C(30)=MASS(ISOTOP,3)-SUMZ(C,19,29)
78. MASS(ISOTOP,3)=C(19)*EEXPO(1)+C(20)*EEXPO(6)+C(21)*EEXPO(11)+C(22)
79. *EEXPO(16)+C(23)*EEXPO(3)+C(24)*EEXPO(8)+C(25)*EEXPO(13)+C(26)*EEX
80. 2PO(2)+C(27)*EEXPO(7)+C(28)*EEXPO(12)+C(29)*EEXPO(17)+C(30)*EEXPO(1
81. 3)
82. 3 IF(JUMPOF.OR. JUMP2) GO TO 4
83. C CONSIDER FILTRATION IN BARRIER 1 03/25/71
84. DOG=ILR(2,3)*(1.-FILTER(ISOTOP,2))
85. D(19)=(DOG*B(16)+ABLE*D(1)+BAKER*D(4)+CHARLY*D(10))/(EXPO(18)-EXPO
86. 1(1))
87. D(20)=(DOG*B(17)+BAKER*D(5)+CHARLY*D(11))/(EXPO(18)-EXPO(6))
88. D(21)=(DOG*B(18)+CHARLY*D(12))/(EXPO(18)-EXPO(11))

```

```

89.      D(22)=DOG*B(19)/(EXPO(18)-EXPO(16))
90.      D(23)=(DOG*A(20)+ABLE*D(3)+BAKER*D(7)+CHARLY*D(13))/(EXPO(18)-EXPO
91.      1(3))
92.      D(24)=(DOG*B(21)+BAKER*D(3)+CHARLY*D(14))/(EXPO(18)-EXPO(8))
93.      D(25)=(DOG*B(22)+CHARLY*D(18))/(EXPO(18)-EXPO(13))
94.      D(26)=(DOG*B(23)+ABLE*D(2)+BAKER*D(6)+CHARLY*D(15))/(EXPO(18)-EXPO
95.      1(2))
96.      D(27)=(DOG*B(24)+BAKER*D(8)+CHARLY*D(16))/(EXPO(18)-EXPO(7))
97.      D(28)=(DOG*B(25)+CHARLY*D(17))/(EXPO(18)-EXPO(12))
98.      D(29)=DOG*B(26)/(EXPO(18)-EXPO(17))
99.      D(30)=MASS(ISOTOP,4)-SUMZ(D,19,29)
100.     MASS(ISOTOP,4)=D(19)*EEXPO(11)+D(20)*EEXPO(6)+D(21)*EEXPO(11)+D(22)
101.     1+EEXPO(16)+D(23)*EEXPO(3)+D(24)*EEXPO(8)+D(25)*EEXPO(13)+D(26)*EEX
102.     2PO(2)+D(27)*EEXPO(7)+D(28)*EEXPO(12)+D(29)*EEXPO(17)+D(30)*EEXPO(1
103.     38)
104.     4 IF(JUMP3) GO TO 3
105. C   CONSIDER BARRIER 2
106.     CALL ZERO(F,25,40)
107.     CALL ZERO(G,25,40)
108.     EXPO(19)=LAMHDA(ISOTOP)+FOR(2)+OLR(4)
109.     EEXPO(19)=EXP(-EXPO(19)*DELTAT)
110.     DOG=ILP(1,4)*FILTER(ISOTOP,1)
111.     EASY=ILR(2,4)+ILR(3,4)*FILTER(ISOTOP,2)
112.     E(22)=(EASY*B(16)+DOG*A(7)+ABLE*E(1)+BAKER*E(4)+CHARLY*E(11))/(EXP
113.     10(19)-EXPO(1))
114.     E(23)=(EASY*B(17)+DOG*A(8)+BAKER*E(5)+CHARLY*E(12))/(EXPO(19)-EXPO
115.     1(6))
116.     E(24)=(EASY*B(18)+DOG*A(9)+CHARLY*E(13))/(EXPO(19)-EXPO(11))
117.     E(25)=(EASY*B(19)+DOG*A(10))/(EXPO(19)-EXPO(16))
118.     E(26)=(EASY*B(20)+BAKER*E(6)+CHARLY*E(14))/(EXPO(19)-EXPO(3))
119.     E(27)=(EASY*B(21)+CHARLY*E(15))/(EXPO(19)-EXPO(8))
120.     E(28)=EASY*B(22)/(EXPO(19)-EXPO(13))
121.     E(29)=(EASY*B(23)+ABLE*E(2)+BAKER*E(7)+CHARLY*E(16))/(EXPO(19)-EX
122.     1PO(2))
123.     E(30)=(EASY*B(24)+BAKER*E(8)+CHARLY*E(17))/(EXPO(19)-EXPO(7))
124.     E(31)=(EASY*B(25)+CHARLY*E(18))/(EXPO(19)-EXPO(12))
125.     E(32)=EASY*B(26)/(EXPO(19)-EXPO(17))
126.     E(33)=(ABLE*E(3)+BAKER*E(9)+CHARLY*E(19))/(EXPO(19)-EXPO(4))
127.     E(34)=(BAKER*E(10)+CHARLY*E(20))/(EXPO(19)-EXPO(9))
128.     E(35)=CHARLY*E(21)/(EXPO(19)-EXPO(14))
129.     IF(JUMP4,OR,NOT,JUMPUP) GO TO 5
130.     E(22)=E(22)+(ABLE*F(1)+BAKER*F(5)+CHARLY*F(13))/(EXPO(19)-EXPO(1))
131.     E(23)=E(23)+(BAKER*F(6)+CHARLY*F(14))/(EXPO(19)-EXPO(6))
132.     E(24)=E(24)+CHARLY*F(15)/(EXPO(19)-EXPO(11))
133.     E(26)=E(26)+(ABLE*F(4)+BAKER*F(7)+CHARLY*F(16))/(EXPO(19)-EXPO(3))
134.     E(27)=E(27)+(BAKER*F(12)+CHARLY*F(17))/(EXPO(19)-EXPO(8))
135.     E(28)=E(28)+CHARLY*F(24)/(EXPO(19)-EXPO(13))
136.     E(29)=E(29)+(ABLE*F(2)+BAKER*F(9)+CHARLY*F(18))/(EXPO(19)-EXPO(2))
137.     E(30)=E(30)+(BAKER*F(9)+CHARLY*F(19))/(EXPO(19)-EXPO(7))
138.     E(31)=E(31)+CHARLY*F(20)/(EXPO(19)-EXPO(12))
139.     E(33)=E(33)+(ABLE*F(3)+BAKER*F(10)+CHARLY*F(21))/(EXPO(19)-EXPO(4)
140.     1)
141.     E(34)=E(34)+(BAKER*F(11)+CHARLY*F(22))/(EXPO(19)-EXPO(9))
142.     E(35)=E(35)+CHARLY*F(23)/(EXPO(19)-EXPO(14))
143.     5 IF(.NOT.JUMPOF,OR,JUMP4) GO TO 6
144.     E(22)=E(22)+(ABLE*D(1)+BAKER*D(4)+CHARLY*D(10))/(EXPO(19)-EXPO(1))
145.     E(23)=E(23)+(BAKER*D(5)+CHARLY*D(11))/(EXPO(19)-EXPO(6))
146.     E(24)=E(24)+CHARLY*D(12)/(EXPO(19)-EXPO(11))
147.     E(26)=E(26)+(ABLE*D(3)+BAKER*D(7)+CHARLY*D(13))/(EXPO(19)-EXPO(3))
148.     E(27)=E(27)+(BAKER*D(9)+CHARLY*D(14))/(EXPO(19)-EXPO(8))
149.     E(28)=E(28)+CHARLY*D(18)/(EXPO(19)-EXPO(13))
150.     E(29)=E(29)+(ABLE*D(2)+BAKER*D(6)+CHARLY*D(15))/(EXPO(19)-EXPO(2))
151.     E(30)=E(30)+(BAKER*D(8)+CHARLY*D(16))/(EXPO(19)-EXPO(7))
152.     E(31)=E(31)+CHARLY*D(17)/(EXPO(19)-EXPO(12))
153.     6 E(36)=MASS(ISOTOP,5)-SUMZ(E,22,35)
154.     MASS(ISOTOP,5)=E(22)*EEXPO(1)+E(23)*EEXPO(6)+E(24)*EEXPO(11)+E(25)
155.     1+EEXPO(16)+E(26)*EEXPO(3)+E(27)*EEXPO(8)+E(28)*EEXPO(13)+E(29)*EEX
156.     2PO(2)+E(30)*EEXPO(7)+E(31)*EEXPO(12)+E(32)*EEXPO(17)+E(33)*EEXPO(4
157.     3)+E(34)*EEXPO(9)+E(35)*EEXPO(14)+E(36)*EEXPO(19)
158.     IF(JUMP4,OR,JUMPUP,OR,JUMPOF) GO TO 7
159. C   CONSIDER FALLOUT IN BARRIER 2
160.     F(25)=(FOR(2)*E(22)+ABLE*F(1)+BAKER*F(5)+CHARLY*F(13))/(EXPO(18)-E
161.     1XPO(1))
162.     F(26)=(FOR(2)*E(23)+BAKER*F(6)+CHARLY*F(14))/(EXPO(18)-EXPO(6))
163.     F(27)=(FOR(2)*E(24)+CHARLY*F(15))/(EXPO(18)-EXPO(11))
164.     F(28)=FOR(2)*E(25)/(EXPO(18)-EXPO(16))
165.     F(29)=(FOR(2)*E(26)+ABLE*F(4)+BAKER*F(7)+CHARLY*F(16))/(EXPO(18)-E
166.     1XPO(3))
167.     F(30)=(FOR(2)*E(27)+BAKER*F(12)+CHARLY*F(17))/(EXPO(18)-EXPO(8))
168.     F(31)=(FOR(2)*E(28)+CHARLY*F(24))/(EXPO(18)-EXPO(13))
169.     F(32)=(FOR(2)*E(29)+ABLE*F(2)+BAKER*F(9)+CHARLY*F(18))/(EXPO(18)-
170.     1EXPO(2))
171.     F(33)=(FOR(2)*E(30)+BAKER*F(9)+CHARLY*F(19))/(EXPO(18)-EXPO(7))
172.     F(34)=(FOR(2)*E(31)+CHARLY*F(20))/(EXPO(18)-EXPO(12))
173.     F(35)=FOR(2)*E(32)/(EXPO(18)-EXPO(17))
174.     F(36)=(FOR(2)*E(33)+ABLE*F(3)+BAKER*F(10)+CHARLY*F(21))/(EXPO(18)-
175.     1EXPO(4))
176.     F(37)=(FOR(2)*E(34)+BAKER*F(11)+CHARLY*F(22))/(EXPO(18)-EXPO(9))
177.     F(38)=(FOR(2)*E(35)+CHARLY*F(23))/(EXPO(18)-EXPO(14))
178.     F(39)=FOR(2)*E(36)/(EXPO(18)-EXPO(19))
179.     F(40)=MASS(ISOTOP,6)-SUMZ(F,25,39)
180.     MASS(ISOTOP,6)=F(25)*EEXPO(1)+F(26)*EEXPO(6)+F(27)*EEXPO(11)+F(28)

```

```

181. 1=EXPO(16)+F(29)*EEXPO(3)+F(30)*EEXPO(8)+F(31)*EEXPO(13)+F(32)*EEX
182. 2PO(2)+F(33)*EEXPO(7)+F(34)*EEXPO(12)+F(35)*EEXPO(17)+F(36)*EEXPO(4
183. 3)+F(37)*EEXPO(9)+F(38)*EEXPO(14)+F(39)*EEXPO(19)+F(40)*EEXPO(18)
184. 7 IF(JUMPOF,OR,JUMP5) GO TO 8
185. C CONSIDER FILTRATION IN BARRIER 2 03/25/71
186. DOG=ILR(4,5)*(1.-FILTER(ISOTOP,2))
187. G(25)=(DOG+E(22)+ABLE*G(1)+BAKER*G(5)+CHARLY*G(13))/(EXPO(18)-EXPO
188. 1(1))
189. G(26)=(DOG+E(23)+BAKER*G(6)+CHARLY*G(14))/(EXPO(18)-EXPO(6))
190. G(27)=(DOG+E(24)+CHARLY*G(15))/(EXPO(18)-EXPO(11))
191. G(28)=DOG+E(25)/(EXPO(18)-EXPO(16))
192. G(29)=(DOG+E(26)+ABLE*G(4)+BAKER*G(7)+CHARLY*G(16))/(EXPO(18)-EXPO
193. 1(3))
194. G(30)=(DOG+E(27)+BAKER*G(12)+CHARLY*G(17))/(EXPO(18)-EXPO(8))
195. G(31)=(DOG+E(28)+CHARLY*G(24))/(EXPO(18)-EXPO(13))
196. G(32)=(DOG+E(29)+ABLE*G(2)+BAKER*G(8)+CHARLY*G(13))/(EXPO(18)-EXPO
197. 1(2))
198. G(33)=(DOG+E(30)+BAKER*G(9)+CHARLY*G(19))/(EXPO(18)-EXPO(7))
199. G(34)=(DOG+E(31)+CHARLY*G(20))/(EXPO(18)-EXPO(12))
200. G(35)=DOG+E(32)/(EXPO(18)-EXPO(17))
201. G(36)=(DOG+E(33)+ABLE*G(3)+BAKER*G(10)+CHARLY*G(21))/(EXPO(18)-EXP
202. 10(4))
203. G(37)=(DOG+E(34)+BAKER*G(11)+CHARLY*G(22))/(EXPO(18)-EXPO(9))
204. G(38)=(DOG+E(35)+CHARLY*G(23))/(EXPO(18)-EXPO(14))
205. G(39)=DOG+E(36)/(EXPO(18)-EXPO(19))
206. G(40)=MASS(ISOTOP,7)-SUMZ(G,25,39)
207. MASS(ISOTOP,7)=G(25)*EEXPO(1)+G(26)*EEXPO(6)+G(27)*EEXPO(11)+G(28)
208. 1*EEXPO(16)+G(29)*EEXPO(3)+G(30)*EEXPO(8)+G(31)*EEXPO(13)+G(32)*EEX
209. 2PO(2)+G(33)*EEXPO(7)+G(34)*EEXPO(12)+G(35)*EEXPO(17)+G(36)*EEXPO(4
210. 3)+G(37)*EEXPO(9)+G(38)*EEXPO(14)+G(39)*EEXPO(19)+G(40)*EEXPO(18)
211. 8 IF(JUMP6) GO TO 12
212. C CONSIDER BARRIER 3 03/25/71
213. CALL ZERO(0,31,50)
214. CALL ZERO(P,31,50)
215. EXPO(20)=LAM3DA(ISOTOP)+FOR(3)+OLR(6)
216. EEXPO(20)=EXP(-EXPO(20)*DELTA)
217. EASY=ILR(1,6)*FILTER(ISOTOP,1)
218. DOG=ILR(4,6)+ILR(5,6)*FILTER(ISOTOP,2)
219. H(28)=(EASY*A(7)+DOG+E(22)+ABLE*H(1)+BAKER*H(5)+CHARLY*H(14))/(EXP
220. 10(20)-EXPO(1))
221. H(29)=(EASY*A(8)+DOG+E(23)+BAKER*H(6)+CHARLY*H(15))/(EXPO(20)-EXPO
222. 1(6))
223. H(30)=(EASY*A(9)+DOG+E(24)+CHARLY*H(16))/(EXPO(20)-EXPO(11))
224. H(31)=(EASY*A(10)+DOG+E(25))/(EXPO(20)-EXPO(16))
225. H(32)=(DOG+E(26)+BAKER*H(7)+CHARLY*H(17))/(EXPO(20)-EXPO(3))
226. H(33)=(DOG+E(27)+CHARLY*H(18))/(EXPO(20)-EXPO(8))
227. H(34)=DOG+E(28)/(EXPO(20)-EXPO(13))
228. H(35)=(DOG+E(29)+ABLE*H(2)+BAKER*H(8)+CHARLY*H(19))/(EXPO(20)-EXPO
229. 1(2))
230. H(36)=(DOG+E(30)+BAKER*H(9)+CHARLY*H(20))/(EXPO(20)-EXPO(7))
231. H(37)=(DOG+E(31)+CHARLY*H(21))/(EXPO(20)-EXPO(12))
232. H(38)=DOG+E(32)/(EXPO(20)-EXPO(17))
233. H(39)=(DOG+E(33)+ABLE*H(3)+BAKER*H(10)+CHARLY*H(22))/(EXPO(20)-EXP
234. 10(4))
235. H(40)=(DOG+E(34)+BAKER*H(11)+CHARLY*H(23))/(EXPO(20)-EXPO(9))
236. H(41)=(DOG+E(35)+CHARLY*H(24))/(EXPO(20)-EXPO(14))
237. H(42)=DOG+E(36)/(EXPO(20)-EXPO(19))
238. H(43)=(ABLE*H(4)+BAKER*H(12)+CHARLY*H(25))/(EXPO(20)-EXPO(5))
239. H(44)=(BAKER*H(13)+CHARLY*H(26))/(EXPO(20)-EXPO(10))
240. H(45)=CHARLY*H(27)/(EXPO(20)-EXPO(15))
241. IF(.NOT.JUMPOF,OR,JUMP10) GO TO 9
242. H(28)=H(28)+(ABLE*G(1)+BAKER*G(5)+CHARLY*G(13))/(EXPO(20)-EXPO(1))
243. H(29)=H(29)+(BAKER*G(6)+CHARLY*G(14))/(EXPO(20)-EXPO(6))
244. H(30)=H(30)+CHARLY*G(15)/(EXPO(20)-EXPO(11))
245. H(32)=H(32)+(ABLE*G(4)+BAKER*G(7)+CHARLY*G(16))/(EXPO(20)-EXPO(3))
246. H(33)=H(33)+(BAKER*G(12)+CHARLY*G(17))/(EXPO(20)-EXPO(8))
247. H(34)=H(34)+CHARLY*G(24)/(EXPO(20)-EXPO(13))
248. H(35)=H(35)+(ABLE*G(2)+BAKER*G(8)+CHARLY*G(18))/(EXPO(20)-EXPO(2))
249. H(36)=H(36)+(BAKER*G(9)+CHARLY*G(19))/(EXPO(20)-EXPO(7))
250. H(37)=H(37)+CHARLY*G(20)/(EXPO(20)-EXPO(12))
251. H(39)=H(39)+(ABLE*G(3)+BAKER*G(10)+CHARLY*G(21))/(EXPO(20)-EXPO(4)
252. 1)
253. H(40)=H(40)+(BAKER*G(11)+CHARLY*G(22))/(EXPO(20)-EXPO(9))
254. H(41)=H(41)+CHARLY*G(23)/(EXPO(20)-EXPO(14))
255. 9 IF(JUMP7,OR,.NOT.JUMPUP) GO TO 10
256. H(28)=H(28)+(ABLE*O(1)+BAKER*O(6)+CHARLY*O(16))/(EXPO(20)-EXPO(1))
257. H(29)=H(29)+(BAKER*O(7)+CHARLY*O(17))/(EXPO(20)-EXPO(6))
258. H(30)=H(30)+CHARLY*O(18)/(EXPO(20)-EXPO(11))
259. H(32)=H(32)+(ABLE*O(5)+BAKER*O(8)+CHARLY*O(19))/(EXPO(20)-EXPO(3))
260. H(33)=H(33)+(BAKER*O(15)+CHARLY*O(20))/(EXPO(20)-EXPO(8))
261. H(34)=H(34)+CHARLY*O(30)/(EXPO(20)-EXPO(13))
262. H(35)=H(35)+(ABLE*O(2)+BAKER*O(9)+CHARLY*O(21))/(EXPO(20)-EXPO(2))
263. H(36)=H(36)+(BAKER*O(10)+CHARLY*O(22))/(EXPO(20)-EXPO(7))
264. H(37)=H(37)+CHARLY*O(23)/(EXPO(20)-EXPO(12))
265. H(39)=H(39)+(ABLE*O(3)+BAKER*O(11)+CHARLY*O(24))/(EXPO(20)-EXPO(4)
266. 1)
267. H(40)=H(40)+(BAKER*O(12)+CHARLY*O(25))/(EXPO(20)-EXPO(9))
268. H(41)=H(41)+CHARLY*O(26)/(EXPO(20)-EXPO(14))
269. H(43)=H(43)+(ABLE*O(4)+BAKER*O(13)+CHARLY*O(27))/(EXPO(20)-EXPO(5)
270. 1)
271. H(44)=H(44)+(BAKER*O(14)+CHARLY*O(28))/(EXPO(20)-EXPO(10))
272. H(45)=H(45)+CHARLY*O(29)/(EXPO(20)-EXPO(15))

```

```

273. 10 H(46)=MASS(ISOTOP,8)-SUMZ(H,28,45)
274. MASS(ISOTOP,9)=H(28)*EEXPC(1)+H(29)*EEXPC(6)+H(30)*EEXPC(11)+H(31)
275. 1+EEXPC(16)+H(32)*EEXPC(3)+H(33)*EEXPC(8)+H(34)*EEXPC(13)+H(35)*EEX
276. 2PC(2)+H(36)*EEXPC(7)+H(37)*EEXPC(12)+H(38)*EEXPC(17)+H(39)*EEXPC(4
277. 3)+H(40)*EEXPC(9)+H(41)*EEXPC(14)+H(42)*EEXPC(19)+H(43)*EEXPC(5)+H(
278. 44)*EEXPC(10)+H(45)*EEXPC(15)+H(46)*EEXPC(20)
279. IF(JUMPCF,CR,JUMPOF,OR,JUMPOF) GO TO 11
280. C CONSIDER FALLOUT IN BARRIER 3 03/25/71
281. O(31)=(FOR(3)*H(28)+ABLE*O(1)+BAKER*O(6)+CHARLY*O(16))/(EXPO(18)-E
282. 1XPO(1))
283. O(32)=(FOR(3)*H(29)+BAKER*O(7)+CHARLY*O(17))/(EXPO(18)-EXPO(6))
284. O(33)=(FOR(3)*H(30)+CHARLY*O(18))/(EXPO(18)-EXPO(11))
285. O(34)=(FOR(3)*H(31))/(EXPO(18)-EXPO(16))
286. O(35)=(FOR(3)*H(32)+ABLE*O(5)+BAKER*O(8)+CHARLY*O(19))/(EXPO(18)-E
287. 1XPO(3))
288. O(36)=(FOR(3)*H(33)+BAKER*O(15)+CHARLY*O(20))/(EXPO(18)-EXPO(8))
289. O(37)=(FOR(3)*H(34)+CHARLY*O(30))/(EXPO(18)-EXPO(13))
290. O(38)=(FOR(3)*H(35)+ABLE*O(2)+BAKER*O(9)+CHARLY*O(21))/(EXPO(18)-E
291. 1XPO(2))
292. O(39)=(FOR(3)*H(36)+BAKER*O(10)+CHARLY*O(22))/(EXPO(18)-EXPO(7))
293. O(40)=(FOR(3)*H(37)+CHARLY*O(23))/(EXPO(18)-EXPO(12))
294. O(41)=(FOR(3)*H(38))/(EXPO(18)-EXPO(17))
295. O(42)=(FOR(3)*H(39)+ABLE*O(3)+BAKER*O(11)+CHARLY*O(24))/(EXPO(18)-
296. 1XPO(4))
297. O(43)=(FOR(3)*H(40)+BAKER*O(12)+CHARLY*O(25))/(EXPO(18)-EXPO(9))
298. O(44)=(FOR(3)*H(41)+CHARLY*O(26))/(EXPO(18)-EXPO(14))
299. O(45)=(FOR(3)*H(42))/(EXPO(18)-EXPO(19))
300. O(46)=(FOR(3)*H(43)+ABLE*O(4)+BAKER*O(13)+CHARLY*O(27))/(EXPO(18)-
301. 1XPO(5))
302. O(47)=(FOR(3)*H(44)+BAKER*O(14)+CHARLY*O(28))/(EXPO(18)-EXPO(10))
303. O(48)=(FOR(3)*H(45)+CHARLY*O(29))/(EXPO(18)-EXPO(15))
304. O(49)=(FOR(3)*H(46))/(EXPO(18)-EXPO(20))
305. O(50)=MASS(ISOTOP,9)-SUMZ(O,31,49)
306. MASS(ISOTOP,9)=O(31)*EEXPC(1)+O(32)*EEXPC(6)+O(33)*EEXPC(11)+O(34)
307. 1+EEXPC(16)+O(35)*EEXPC(3)+O(36)*EEXPC(8)+O(37)*EEXPC(13)+O(38)*EEX
308. 2PC(2)+O(39)*EEXPC(7)+O(40)*EEXPC(12)+O(41)*EEXPC(17)+O(42)*EEXPC(4
309. 3)+O(43)*EEXPC(9)+O(44)*EEXPC(14)+O(45)*EEXPC(19)+O(46)*EEXPC(5)+O(
310. 47)*EEXPC(10)+O(48)*EEXPC(15)+O(49)*EEXPC(20)+O(50)*EEXPC(18)
311. 11 IF(JUMPCF,CR,JUMPF) GO TO 12
312. C CONSIDER FILTRATION IN BARRIER 3 03/25/71
313. DOG=ILR(6,7)*(1.-FILTER(ISOTOP,2))
314. P(31)=(DOG*H(28)+ABLE*P(1)+BAKER*P(6)+CHARLY*P(16))/(EXPO(18)-EXPO
315. 1(1))
316. P(32)=(DOG*H(29)+BAKER*P(7)+CHARLY*P(17))/(EXPO(18)-EXPO(6))
317. P(33)=(DOG*H(30)+CHARLY*P(18))/(EXPO(18)-EXPO(11))
318. P(34)=(DOG*H(31))/(EXPO(18)-EXPO(16))
319. P(35)=(DOG*H(32)+ABLE*P(5)+BAKER*P(8)+CHARLY*P(19))/(EXPO(18)-EXPO
320. 1(3))
321. P(36)=(DOG*H(33)+BAKER*P(15)+CHARLY*P(20))/(EXPO(18)-EXPO(8))
322. P(37)=(DOG*H(34)+CHARLY*P(30))/(EXPO(18)-EXPO(13))
323. P(38)=(DOG*H(35)+ABLE*P(2)+BAKER*P(9)+CHARLY*P(21))/(EXPO(18)-EXPO
324. 1(2))
325. P(39)=(DOG*H(36)+BAKER*P(10)+CHARLY*P(22))/(EXPO(18)-EXPO(7))
326. P(40)=(DOG*H(37)+CHARLY*P(23))/(EXPO(18)-EXPO(12))
327. P(41)=(DOG*H(38))/(EXPO(18)-EXPO(17))
328. P(42)=(DOG*H(39)+ABLE*P(3)+BAKER*P(11)+CHARLY*P(24))/(EXPO(18)-EXP
329. 10(4))
330. P(43)=(DOG*H(40)+BAKER*P(12)+CHARLY*P(25))/(EXPO(18)-EXPO(9))
331. P(44)=(DOG*H(41)+CHARLY*P(26))/(EXPO(18)-EXPO(14))
332. P(45)=(DOG*H(42))/(EXPO(18)-EXPO(19))
333. P(46)=(DOG*H(43)+ABLE*P(4)+BAKER*P(13)+CHARLY*P(27))/(EXPO(18)-EXP
334. 10(5))
335. P(47)=(DOG*H(44)+BAKER*P(14)+CHARLY*P(28))/(EXPO(18)-EXPO(10))
336. P(48)=(DOG*H(45)+CHARLY*P(29))/(EXPO(18)-EXPO(15))
337. P(49)=(DOG*H(46))/(EXPO(18)-EXPO(20))
338. P(50)=MASS(ISOTOP,10)-SUMZ(P,31,49)
339. MASS(ISOTOP,10)=P(31)*EEXPC(1)+P(32)*EEXPC(6)+P(33)*EEXPC(11)+P(34)
340. 1+EEXPC(16)+P(35)*EEXPC(3)+P(36)*EEXPC(8)+P(37)*EEXPC(13)+P(38)*EEX
341. 2PC(2)+P(39)*EEXPC(7)+P(40)*EEXPC(12)+P(41)*EEXPC(17)+P(42)*EEXPC(4
342. 3)+P(43)*EEXPC(9)+P(44)*EEXPC(14)+P(45)*EEXPC(19)+P(46)*EEXPC(5)+P
343. 4(47)*EEXPC(10)+P(48)*EEXPC(15)+P(49)*EEXPC(20)+P(50)*EEXPC(18)
344. C CALCULATE RELEASE RATE TO ATMOSPHERE 03/25/71
345. 12 QUANTY(ISOTOP)=ALR(1)*FILTER(ISOTOP,1)*MASS(ISOTOP,1)+(ALR(2)+ALR(
346. 13)*FILTER(ISOTOP,2))*MASS(ISOTOP,2)+(ALR(4)+ALR(5)*FILTER(ISOTOP,2
347. 2))*MASS(ISOTOP,5)+(ALR(6)+ALR(7)*FILTER(ISOTOP,2))*MASS(ISOTOP,8)
348. IF(.NOT.JUMPOF,OR,JUMP2) GO TO 13
349. QUANTY(ISOTOP)=QUANTY(ISOTOP)+ALR(3)/ALR(3)*(ABLE*MASS(ISOTOP-3,4)
350. 1+BAKER*MASS(ISOTOP-2,4)+CHARLY*MASS(ISOTOP-1,4))+ALR(5)/ALR(5)*
351. 2ABLE*MASS(ISOTOP-3,7)+BAKER*MASS(ISOTOP-2,7)+CHARLY*MASS(ISOTOP-1,
352. 37))+ALR(7)/ALR(7)*(ABLE*MASS(ISOTOP-3,10)+BAKER*MASS(ISOTOP-2,10)
353. 4+CHARLY*MASS(ISOTOP-1,10))
354. C RETURN TO BAFFLE 03/25/71
355. 13 RETURN
356. END

```

```

1.  C SUBROUTINE FIFTH DOES CONTAINMENT CALCULATION FOR FIFTH CHAIN MEMBE03/25/71
2.  SUBROUTINE FIFTH                                03/25/71
3.  COMMON ISOTOP,SPACE(97),DELTA1,MASS(600,10),HOLE(2850),DKFRCT(450,
4.  A3),FILTER(600,2),LAMBDA(600),SPOT(96),H(153),A(20),O(105),P(105),
5.  ZG(84),F(84),E(105),D(63),C(63),B(77),FOR(3),ALR(7),ILR(7,8),OLR(7)
6.  3,QUANTY(600),EXPO(35),EEXPO(35),SKIP(1200),JUMP1,JUMP4,JUMP2,JUMP3
7.  4,JUMP5,JUMP6,JUMP8,JUMP7,JUMP9,JUMP10,JUMPOF,JUMPUP
8.  REAL MASS,ILR,LAMBDA
9.  LOGICAL JUMP1,JUMP2,JUMP3,JUMP4,JUMP5,JUMP6,JUMP7,JUMP8
10. LOGICAL JUMP9,JUMP10,JUMPOF,JUMPUP
11.
12.  C A REPRESENTS COEFFICIENTS FOR FUEL, B FOR 1ST BARRIER, C FOR 1ST FALL03/25/71
13.  C D FOR 1ST FILTER, E FOR 2ND BARRIER, F FOR 2ND FALL03/25/71
14.  C H FOR 3RD BARRIER, G FOR 3RD FALL03/25/71
15.  C H FOR 3RD BARRIER, G FOR 3RD FALL03/25/71
16.  EXPO(21)=LAMBDA(ISOTOP)+OLR(1)*FILTER(ISOTOP,1)
17.  EEXPO(21)=EXP(-EXPO(21)*DELTA1)
18.  ABLE=LAMBDA(ISOTOP-3)*DKFRCT(ISOTOP,3)
19.  BAKER=LAMBDA(ISOTOP-2)*DKFRCT(ISOTOP,2)
20.  CHARLY=LAMBDA(ISOTOP-1)*DKFRCT(ISOTOP,1)
21.  A(11)=(ABLE*A(2)+BAKER*A(4)+CHARLY*A(7))/(EXPO(21)-EXPO(1))
22.  A(12)=(ABLE*A(3)+BAKER*A(5)+CHARLY*A(8))/(EXPO(21)-EXPO(6))
23.  A(13)=(BAKER*A(6)+CHARLY*A(9))/(EXPO(21)-EXPO(11))
24.  A(14)=CHARLY*A(16)/(EXPO(21)-EXPO(16))
25.  A(15)=MASS(ISOTOP,1)-SUMZ(A,11,14)
26.  MASS(ISOTOP,1)=A(11)*EEXPO(1)+A(12)*EEXPO(6)+A(13)*EEXPO(11)+A(14)
27.  1+EEXPO(16)+A(15)*EEXPO(21)
28.  C CONSIDER BARRIER 1                                03/25/71
29.  EXPO(22)=LAMBDA(ISOTOP)+FOR(1)+OLR(2)
30.  EEXPO(22)=EXP(-EXPO(22)*DELTA1)
31.  DOG=ILR(1,2)*FILTER(ISOTOP,1)
32.  B(27)=(DOG*A(11)+ABLE*B(3)+BAKER*B(8)+CHARLY*B(16))/(EXPO(22)-EXPO
33.  1(1))
34.  B(28)=(DOG*A(12)+ABLE*B(4)+BAKER*B(9)+CHARLY*B(17))/(EXPO(22)-EXPO
35.  1(6))
36.  B(29)=(DOG*A(13)+BAKER*B(10)+CHARLY*B(18))/(EXPO(22)-EXPO(11))
37.  B(30)=(DOG*A(14)+CHARLY*B(19))/(EXPO(22)-EXPO(16))
38.  B(31)=DOG*A(15)/(EXPO(22)-EXPO(21))
39.  B(32)=(ABLE*B(6)+BAKER*B(11)+CHARLY*B(20))/(EXPO(22)-EXPO(3))
40.  B(33)=(BAKER*B(12)+CHARLY*B(21))/(EXPO(22)-EXPO(1))
41.  B(34)=CHARLY*B(22)/(EXPO(22)-EXPO(13))
42.  B(36)=(ABLE*B(5)+BAKER*B(13)+CHARLY*B(23))/(EXPO(22)-EXPO(2))
43.  B(37)=(ABLE*B(7)+BAKER*B(14)+CHARLY*B(24))/(EXPO(22)-EXPO(7))
44.  B(38)=(BAKER*B(15)+CHARLY*B(25))/(EXPO(22)-EXPO(12))
45.  B(39)=CHARLY*B(26)/(EXPO(22)-EXPO(17))
46.  CALL ZERO(C,31,45)
47.  CALL ZERO(B,31,45)
48.  EXPO(23)=LAMBDA(ISOTOP)
49.  EEXPO(23)=EXP(-EXPO(23)*DELTA1)
50.  IF (JUMP1,OR,NOT,JUMPUP) GO TO 1
51.  B(27)=B(27)+(ABLE*C(4)+BAKER*C(10)+CHARLY*C(19))/(EXPO(22)-EXPO(1)
52.  1)
53.  B(28)=B(28)+(ABLE*C(5)+BAKER*C(11)+CHARLY*C(20))/(EXPO(22)-EXPO(6)
54.  1)
55.  B(29)=B(29)+(BAKER*C(12)+CHARLY*C(21))/(EXPO(22)-EXPO(11))
56.  B(30)=B(30)+CHARLY*C(22)/(EXPO(22)-EXPO(16))
57.  B(32)=B(32)+(ABLE*C(7)+BAKER*C(13)+CHARLY*C(23))/(EXPO(22)-EXPO(3)
58.  1)
59.  B(33)=B(33)+(ABLE*C(9)+BAKER*C(14)+CHARLY*C(24))/(EXPO(22)-EXPO(8)
60.  1)
61.  B(34)=B(34)+(BAKER*C(18)+CHARLY*C(25))/(EXPO(22)-EXPO(13))
62.  B(35)=CHARLY*C(30)/(EXPO(22)-EXPO(18))
63.  B(36)=B(36)+(ABLE*C(6)+BAKER*C(15)+CHARLY*C(26))/(EXPO(22)-EXPO(2)
64.  1)
65.  B(37)=B(37)+(ABLE*C(8)+BAKER*C(16)+CHARLY*C(27))/(EXPO(22)-EXPO(7)
66.  1)
67.  B(38)=B(38)+(BAKER*C(17)+CHARLY*C(28))/(EXPO(22)-EXPO(12))
68.  B(39)=B(39)+CHARLY*C(29)/(EXPO(22)-EXPO(17))
69.  GO TO 2
70.  1 B(35)=0.0
71.  2 B(40)=MASS(ISOTOP,2)-SUMZ(B,27,39)
72.  MASS(ISOTOP,2)=B(27)*EEXPO(1)+B(28)*EEXPO(6)+B(29)*EEXPO(11)+
73.  1+B(30)*EEXPO(16)+B(31)*EEXPO(21)+B(32)*EEXPO(3)+B(33)*EEXPO(1)+
74.  2B(34)*EEXPO(13)+B(35)*EEXPO(18)
75.  MASS(ISOTOP,2)=MASS(ISOTOP,2)+B(36)*EEXPO(2)+B(37)*EEXPO(7)+
76.  3+B(38)*EEXPO(12)+B(39)*EEXPO(17)+B(40)*EEXPO(22)
77.  IF (JUMP1,OR,JUMPUP,OR,JUMPOF) GO TO 3
78.  C CONSIDER FALL03/25/71
79.  C(31)=(FOR(1)+B(27)+ABLE*C(4)+BAKER*C(10)+CHARLY*C(19))/(EXPO(23)-
80.  1EXPO(1))
81.  C(32)=(FOR(1)+B(28)+ABLE*C(5)+BAKER*C(11)+CHARLY*C(20))/(EXPO(23)-
82.  1EXPO(6))
83.  C(33)=(FOR(1)+B(29)+BAKER*C(12)+CHARLY*C(21))/(EXPO(23)-EXPO(11))
84.  C(34)=(FOR(1)+B(30)+CHARLY*C(22))/(EXPO(23)-EXPO(16))
85.  C(35)=FOR(1)+B(31)/(EXPO(23)-EXPO(21))
86.  C(36)=(FOR(1)+B(32)+ABLE*C(7)+BAKER*C(13)+CHARLY*C(23))/(EXPO(23)-
87.  1EXPO(3))
88.  C(37)=(FOR(1)+B(33)+ABLE*C(9)+BAKER*C(14)+CHARLY*C(24))/(EXPO(23)-

```

```

89. 1EXP0(6))
90. C(38)=(FOR(1)*E(34)+BAKER*C(11)+CHARLY*C(25))/(EXP0(23)-EXP0(13))
91. C(39)=(FOR(1)*B(35)+CHARLY*C(30))/(EXP0(23)-EXP0(18))
92. C(40)=(FOR(1)*E(36)+ABLE*C(6)+BAKER*C(15)+CHARLY*C(26))/(EXP0(23)-
93. 1EXP0(2))
94. C(41)=(FOR(1)*B(37)+ABLE*C(8)+BAKER*C(16)+CHARLY*C(27))/(EXP0(23)-
95. 1EXP0(7))
96. C(42)=(FOR(1)*B(38)+BAKER*C(17)+CHARLY*C(28))/(EXP0(23)-EXP0(12))
97. C(43)=(FOR(1)*B(39)+CHARLY*C(29))/(EXP0(23)-EXP0(17))
98. C(44)=(FOR(1)*B(40))/(EXP0(23)-EXP0(22))
99. C(45)=MASS(ISOTOP,3)-SUM2(C,31,44)
100. MASS(ISOTOP,3)= C(31)*EEXP0(1)+C(32)*EEXP0(6)+C(33)*EEXP0(11)+
101. 1C(34)*EEXP0(16)+ C(35)*EEXP0(21)+C(36)*EEXP0(3)+C(37)*EEXP0(8)+
102. 2C(38)*EEXP0(13)+ C(39)*EEXP0(18)+C(45)*EEXP0(23)
103. MASS(ISOTOP,3)=MASS(ISOTOP,3)+C(40)*EEXP0(2)+C(41)*EEXP0(7)+
104. 1C(42)*EEXP0(12)+ C(43)*EEXP0(17)+C(44)*EEXP0(22)
105. 3 IF(JUMP2,CR,JUMPOF) GO TO 4
106. C CONSIDER FILTRATION IN BARRIER 1 03/25/71
107. DOG=ILR(2,3)*(1.-FILTER(ISOTOP,2))
108. D(31)=(DOG*B(27)+ABLE*D(4)+BAKER*D(10)+CHARLY*D(19))/(EXP0(23)-EXP
109. 10(1))
110. D(32)=(DOG*B(28)+ABLE*D(5)+BAKER*D(11)+CHARLY*D(20))/(EXP0(23)-EXP
111. 10(6))
112. D(33)=(DOG*B(29)+BAKER*D(12)+CHARLY*D(21))/(EXP0(23)-EXP0(11))
113. D(34)=(DOG*B(30)+CHARLY*D(22))/(EXP0(23)-EXP0(16))
114. D(35)=DOG*B(31)/(EXP0(23)-EXP0(21))
115. D(36)=(DOG*B(32)+ABLE*D(7)+BAKER*D(13)+CHARLY*D(23))/(EXP0(23)-EXP
116. 10(3))
117. D(37)=(DOG*D(33)+ABLE*D(9)+BAKER*D(14)+CHARLY*D(24))/(EXP0(23)-EXP
118. 10(8))
119. D(38)=(DOG*B(34)+BAKER*D(18)+CHARLY*D(25))/(EXP0(23)-EXP0(13))
120. D(39)=(DOG*B(35)+CHARLY*D(30))/(EXP0(23)-EXP0(18))
121. D(40)=(DOG*B(36)+ABLE*D(6)+BAKER*D(15)+CHARLY*D(26))/(EXP0(23)-EXP
122. 10(2))
123. D(41)=(DOG*B(37)+ABLE*D(8)+BAKER*D(16)+CHARLY*D(27))/(EXP0(23)-EXP
124. 10(7))
125. D(42)=(DOG*B(38)+BAKER*D(17)+CHARLY*D(28))/(EXP0(23)-EXP0(12))
126. D(43)=(DOG*B(39)+CHARLY*D(29))/(EXP0(23)-EXP0(17))
127. D(44)=DOG*B(40)/(EXP0(23)-EXP0(22))
128. D(45)=MASS(ISOTOP,4)-SUM2(D,31,44)
129. MASS(ISOTOP,4)= D(31)*EEXP0(1)+D(32)*EEXP0(6)+D(33)*EEXP0(11)+
130. 1D(34)*EEXP0(16)+D(35)*EEXP0(21)+D(36)*EEXP0(3)+D(37)*EEXP0(8)+
131. 2D(38)*EEXP0(13)+D(39)*EEXP0(18)+D(45)*EEXP0(23)
132. MASS(ISOTOP,4)=MASS(ISOTOP,4)+D(40)*EEXP0(2)+D(41)*EEXP0(7)+
133. 1D(42)*EEXP0(12)+D(43)*EEXP0(17)+D(44)*EEXP0(22)
134. 4 IF(JUMP3) GO TO 8
135. C CONSIDER BARRIER 2 03/25/71
136. CALL ZERO(F,41,60)
137. CALL ZFRN(6,41,60)
138. EXP0(24)=LAMBCA(ISOTOP)+FOR(2)+OLR(4)
139. EEXP0(24)=EXP(-EXP0(24)*DELTA)
140. EASY=ILR(1,4)+FILTER(ISOTOP,1)
141. DOG=ILR(2,4)+ILR(3,4)+FILTER(ISOTOP,2)
142. E(37)=(EASY*A(11)+DOG*B(27)+ABLE*E(4)+BAKER*E(11)+CHARLY*E(22))/(E
143. 1XP0(24)-EXP0(1))
144. E(38)=(EASY*A(12)+DOG*B(28)+ABLE*E(5)+BAKER*E(12)+CHARLY*E(23))/(E
145. 1XP0(24)-EXP0(6))
146. E(39)=(EASY*A(13)+DOG*B(29)+BAKER*E(13)+CHARLY*E(24))/(EXP0(24)-EX
147. 1P0(11))
148. E(40)=(EASY*A(14)+DOG*B(30)+CHARLY*E(25))/(EXP0(24)-EXP0(16))
149. E(41)=(EASY*A(15)+DOG*B(31))/(EXP0(24)-EXP0(21))
150. E(42)=(DOG*B(32)+ABLE*E(6)+BAKER*E(14)+CHARLY*E(26))/(EXP0(24)-EXP
151. 10(3))
152. E(43)=(DOG*B(33)+BAKER*E(15)+CHARLY*E(27))/(EXP0(24)-EXP0(8))
153. E(44)=(DOG*B(34)+CHARLY*E(28))/(EXP0(24)-EXP0(13))
154. E(45)=DOG*B(35)/(EXP0(24)-EXP0(18))
155. E(46)=(DOG*B(36)+ABLE*E(7)+BAKER*E(16)+CHARLY*E(29))/(EXP0(24)-EXP
156. 10(2))
157. E(47)=(DOG*B(37)+ABLE*E(8)+BAKER*E(17)+CHARLY*E(30))/(EXP0(24)-EXP
158. 10(7))
159. E(48)=(DOG*B(38)+BAKER*E(18)+CHARLY*E(31))/(EXP0(24)-EXP0(12))
160. E(49)=(DOG*B(39)+CHARLY*E(32))/(EXP0(24)-EXP0(17))
161. E(50)=DOG*B(40)/(EXP0(24)-EXP0(22))
162. E(51)=(ABLE*E(9)+BAKER*E(19)+CHARLY*E(33))/(EXP0(24)-EXP0(4))
163. E(52)=(ABLE*E(10)+BAKER*E(20)+CHARLY*E(34))/(EXP0(24)-EXP0(9))
164. E(53)=(BAKER*E(21)+CHARLY*E(35))/(EXP0(24)-EXP0(14))
165. E(54)=CHARLY*E(36)/(EXP0(24)-EXP0(19))
166. IF(.NOT.JUMPUP,OR,JUMP4) GO TO 5
167. E(37)=E(37)+(ABLE*F(5)+BAKER*F(13)+CHARLY*F(25))/(EXP0(24)-EXP0(1)
168. 1)
169. E(38)=E(38)+(ABLE*F(6)+BAKER*F(14)+CHARLY*F(26))/(EXP0(24)-EXP0(6)
170. 1)
171. E(39)=E(39)+(BAKER*F(15)+CHARLY*F(27))/(EXP0(24)-EXP0(11))
172. E(40)=E(40)+CHARLY*F(28)/(EXP0(24)-EXP0(16))
173. E(42)=E(42)+(ABLE*F(7)+BAKER*F(16)+CHARLY*F(29))/(EXP0(24)-EXP0(3)
174. 1)
175. E(43)=E(43)+(ABLE*F(12)+BAKER*F(17)+CHARLY*F(30))/(EXP0(24)-EXP0(8)
176. 1)
177. E(44)=E(44)+(BAKER*F(24)+CHARLY*F(31))/(EXP0(24)-EXP0(13))
178. E(45)=E(45)+CHARLY*F(40)/(EXP0(24)-EXP0(18))
179. E(46)=E(46)+(ABLE*F(8)+BAKER*F(18)+CHARLY*F(32))/(EXP0(24)-EXP0(2)
180. 1)

```

```

181. L(47)=E(47)+(ABLE*F(5)+BAKER*F(19)+CHARLY*F(33))/(EXPO(24)-EXPO(7)
182. 1)
183. E(48)=E(48)+(BAKER*F(20)+CHARLY*F(34))/(EXPO(24)-EXPO(12))
184. E(49)=E(49)+CHARLY*F(35)/(EXPO(24)-EXPO(17))
185. E(51)=E(51)+(ABLE*F(10)+BAKER*F(21)+CHARLY*F(36))/(EXPO(24)-EXPO(4)
186. 1)
187. E(52)=E(52)+(ABLE*F(11)+BAKER*F(22)+CHARLY*F(37))/(EXPO(24)-EXPO(9)
188. 1)
189. E(53)=E(53)+(BAKER*F(23)+CHARLY*F(38))/(EXPO(24)-EXPO(14))
190. E(54)=E(54)+CHARLY*F(39)/(EXPO(24)-EXPO(19))
191. 5 IF(.NOT.JUMPCF.OR.JUMPP) GO TO 6
192. E(37)=E(37)+(ABLE*D(4)+BAKER*D(10)+CHARLY*D(19))/(EXPO(24)-EXPO(1)
193. 1)
194. E(38)=E(38)+(ABLE*D(5)+BAKER*D(11)+CHARLY*D(20))/(EXPO(24)-EXPO(6)
195. 1)
196. E(39)=E(39)+(BAKER*D(12)+CHARLY*D(21))/(EXPO(24)-EXPO(11))
197. E(40)=E(40)+CHARLY*D(22)/(EXPO(24)-EXPO(16))
198. E(42)=E(42)+(ABLE*D(7)+BAKER*D(13)+CHARLY*D(23))/(EXPO(24)-EXPO(3)
199. 1)
200. E(43)=E(43)+(ABLE*D(9)+BAKER*D(14)+CHARLY*D(24))/(EXPO(24)-EXPO(8)
201. 1)
202. E(44)=E(44)+(BAKER*D(18)+CHARLY*D(25))/(EXPO(24)-EXPO(13))
203. E(45)=E(45)+CHARLY*D(30)/(EXPO(24)-EXPO(18))
204. E(46)=E(46)+(ABLE*D(6)+BAKER*D(15)+CHARLY*D(26))/(EXPO(24)-EXPO(2)
205. 1)
206. E(47)=E(47)+(ABLE*D(8)+BAKER*D(16)+CHARLY*D(27))/(EXPO(24)-EXPO(7)
207. 1)
208. E(48)=E(48)+(BAKER*D(17)+CHARLY*D(28))/(EXPO(24)-EXPO(12))
209. E(49)=E(49)+CHARLY*D(29)/(EXPO(24)-EXPO(17))
210. 6 E(55)=MASS(ISOTOP,5)-SUMZ(E,37,54)
211. MASS(ISOTOP,5)=E(37)*EEXPO(1)+E(38)*EEXPO(6)+E(39)*EEXPO(11)+
212. 1E(40)*EEXPO(16)+E(41)*EEXPO(21)+E(42)*EEXPO(3)+E(43)*EEXPO(8)+
213. 2E(44)*EEXPO(13)+E(45)*EEXPO(18)
214. MASS(ISOTOP,5)=MASS(ISOTOP,5)+E(46)*EEXPO(2)+E(47)*EEXPO(7)+
215. 1E(48)*EEXPO(12)+E(49)*EEXPO(17)+E(50)*EEXPO(22)+E(51)*EEXPO(4)+
216. 2E(52)*EEXPO(9)+E(53)*EEXPO(14)+E(54)*EEXPO(19)+E(55)*EEXPO(24)
217. IF(JUMP4.OR.JUMPCF.OR.JUMPP) GO TO 7
218. C CONSIDER FALLOUT IN BARRIER 2 03/25/71
219. F(41)=(FOR(2)*E(37)+ABLE*F(5)+BAKER*F(13)+CHARLY*F(25))/(EXPO(23)-
220. 1EXPO(1))
221. F(42)=(FOR(2)*E(38)+ABLE*F(6)+BAKER*F(14)+CHARLY*F(26))/(EXPO(23)-
222. 1EXPO(6))
223. F(43)=(FOR(2)*E(39)+BAKER*F(15)+CHARLY*F(27))/(EXPO(23)-EXPO(11))
224. F(44)=(FOR(2)*E(40)+CHARLY*F(28))/(EXPO(23)-EXPO(16))
225. F(45)=FOR(2)*E(41)/(EXPO(23)-EXPO(21))
226. F(46)=(FOR(2)*E(42)+ABLE*F(7)+BAKER*F(16)+CHARLY*F(29))/(EXPO(23)-
227. 1EXPO(3))
228. F(47)=(FOR(2)*E(43)+ABLE*F(12)+BAKER*F(17)+CHARLY*F(30))/(EXPO(23)
229. 1EXPO(8))
230. F(48)=(FOR(2)*E(44)+BAKER*F(24)+CHARLY*F(31))/(EXPO(23)-EXPO(13))
231. F(49)=(FOR(2)*E(45)+CHARLY*F(40))/(EXPO(23)-EXPO(18))
232. F(50)=(FOR(2)*E(46)+ABLE*F(8)+BAKER*F(18)+CHARLY*F(32))/(EXPO(23)
233. 1EXPO(2))
234. F(51)=(FOR(2)*E(47)+ABLE*F(9)+BAKER*F(19)+CHARLY*F(33))/(EXPO(23)
235. 1EXPO(7))
236. F(52)=(FOR(2)*E(48)+BAKER*F(20)+CHARLY*F(34))/(EXPO(23)-EXPO(12))
237. F(53)=(FOR(2)*E(49)+CHARLY*F(35))/(EXPO(23)-EXPO(17))
238. F(54)=FOR(2)*E(50)/(EXPO(23)-EXPO(22))
239. F(55)=(FOR(2)*E(51)+ABLE*F(10)+BAKER*F(21)+CHARLY*F(36))/(EXPO(23)
240. 1EXPO(4))
241. F(56)=(FOR(2)*E(52)+ABLE*F(11)+BAKER*F(22)+CHARLY*F(37))/(EXPO(23)
242. 1EXPO(9))
243. F(57)=(FOR(2)*E(53)+BAKER*F(23)+CHARLY*F(38))/(EXPO(23)-EXPO(14))
244. F(58)=(FOR(2)*E(54)+CHARLY*F(39))/(EXPO(23)-EXPO(19))
245. F(59)=FOR(2)*E(55)/(EXPO(23)-EXPO(24))
246. F(60)=MASS(ISOTOP,6)-SUMZ(F,41,59)
247. MASS(ISOTOP,6)=F(41)*EEXPO(1)+F(42)*EEXPO(6)+F(43)*EEXPO(11)+
248. 1F(44)*EEXPO(16)+F(45)*EEXPO(21)+F(46)*EEXPO(3)+F(47)*EEXPO(8)+
249. 2F(48)*EEXPO(13)+F(49)*EEXPO(18)+F(50)*EEXPO(23)
250. MASS(ISOTOP,6)=MASS(ISOTOP,6)+F(50)*EEXPO(2)+F(51)*EEXPO(7)+
251. 1F(52)*EEXPO(12)+F(53)*EEXPO(17)+F(54)*EEXPO(22)+F(55)*EEXPO(4)+
252. 2F(56)*EEXPO(9)+F(57)*EEXPO(14)+F(58)*EEXPO(19)+F(59)*EEXPO(24)
253. 7 IF(JUMP5.OR.JUMPCF) GO TO 8
254. C CONSIDER FILTRATION IN BARRIER 2 03/25/71
255. DOG=ILR(4,5)*(1.-FILTER(ISOTOP,2))
256. G(41)=(DOG*F(37)+ABLE*G(5)+BAKER*G(13)+CHARLY*G(25))/(EXPO(23)-EXP
257. 10(1))
258. G(42)=(DOG*E(38)+ABLE*G(6)+BAKER*G(14)+CHARLY*G(26))/(EXPO(23)-EXP
259. 10(6))
260. G(43)=(DOG*E(39)+BAKER*G(15)+CHARLY*G(27))/(EXPO(23)-EXPO(11))
261. G(44)=(DOG*E(40)+CHARLY*G(28))/(EXPO(23)-EXPO(16))
262. G(45)=(DOG*E(41))/(EXPO(23)-EXPO(21))
263. G(46)=(DOG*E(42)+ABLE*G(7)+BAKER*G(16)+CHARLY*G(29))/(EXPO(23)-EXP
264. 10(3))
265. G(47)=(DOG*E(43)+ABLE*G(12)+BAKER*G(17)+CHARLY*G(30))/(EXPO(23)-EX
266. 10(8))
267. G(48)=(DOG*E(44)+BAKER*G(24)+CHARLY*G(31))/(EXPO(23)-EXPO(13))
268. G(49)=(DOG*E(45)+CHARLY*G(40))/(EXPO(23)-EXPO(18))
269. G(50)=(DOG*E(46)+ABLE*G(8)+BAKER*G(18)+CHARLY*G(32))/(EXPO(23)-EXP
270. 10(2))
271. G(51)=(DOG*E(47)+ABLE*G(9)+BAKER*G(19)+CHARLY*G(33))/(EXPO(23)-EXP
272. 10(7))

```

```

273. G(52)=(DOG*E(48)+BAKER*G(20)+CHARLY*G(34))/(EXP0(23)-EXP0(12))
274. G(53)=(DOG*E(49)+CHARLY*G(35))/(EXP0(23)-EXP0(17))
275. G(54)=(DOG*E(50))/(EXP0(23)-EXP0(22))
276. G(55)=(DOG*E(51)+ABLE*G(10)+BAKER*G(21)+CHARLY*G(36))/(EXP0(23)-EX
277. 1P0(4))
278. G(56)=(DOG*E(52)+ABLE*G(11)+BAKER*G(22)+CHARLY*G(37))/(EXP0(23)-EX
279. 1P0(9))
280. G(57)=(DOG*E(53)+BAKER*G(23)+CHARLY*G(38))/(EXP0(23)-EXP0(14))
281. G(58)=(DOG*E(54)+CHARLY*G(39))/(EXP0(23)-EXP0(19))
282. G(59)=DOG*E(55)/(EXP0(23)-EXP0(24))
283. G(60)=MASS(ISOTOP,7)-SUMZ(G,41,59)
284. MASS(ISOTOP,7)= G(41)*EEXP0(1)+G(42)*EEXP0(6)+G(43)*EEXP0(11)+
285. 1G(44)*EEXP0(16)+G(45)*EEXP0(21)+G(46)*EEXP0(3)+G(47)*EEXP0(8)+
286. 2G(48)*EEXP0(13)+G(49)*EEXP0(18)+G(60)*EEXP0(23)
287. MASS(ISOTOP,7)=MASS(ISOTOP,7)+G(50)*EEXP0(2)+G(51)*EEXP0(7)+
288. 1G(52)*EEXP0(12)+G(53)*EEXP0(17)+G(54)*EEXP0(22)+G(55)*EEXP0(4)+
289. 2G(56)*EEXP0(9)+G(57)*EEXP0(14)+G(58)*EEXP0(19)+G(59)*EEXP0(24)
290. 8 IF(JUMP6) GO TO 12
291. C CONSIDER BARRIER 3 03/25/71
292. CALL ZERO(0,51,75)
293. CALL ZERO(P,51,75)
294. EXP0(25)=LAMBDA(ISOTOP)+FCR(3)+OLR(6)
295. EEXP0(25)=EXP(-EXP0(25)*DELTAT)
296. EASY=ILR(1,6)*FILTER(ISOTOP,1)
297. DOG=ILR(4,6)+ILR(5,6)*FILTER(ISOTOP,2)
298. H(47)=(EASY*A(11)+DOG*E(37)+ABLE*H(5)+BAKER*H(14)+CHARLY*H(28))/(E
299. 1XP0(25)-EXP0(1))
300. H(48)=(EASY*A(12)+DOG*E(38)+ABLE*H(6)+BAKER*H(15)+CHARLY*H(29))/(E
301. 1XP0(25)-EXP0(6))
302. H(49)=(EASY*A(13)+DOG*E(39)+BAKER*H(16)+CHARLY*H(30))/(EXP0(25)-EX
303. 1P0(11))
304. H(50)=(EASY*A(14)+DOG*E(40)+CHARLY*H(31))/(EXP0(25)-EXP0(16))
305. H(51)=(EASY*A(15)+DOG*E(41))/(EXP0(25)-EXP0(21))
306. H(52)=(DOG*E(42)+ABLE*H(7)+BAKER*H(17)+CHARLY*H(32))/(EXP0(25)-EXP
307. 10(3))
308. H(53)=(DOG*E(43)+BAKER*H(18)+CHARLY*H(33))/(EXP0(25)-EXP0(8))
309. H(54)=(DOG*E(44)+CHARLY*H(34))/(EXP0(25)-EXP0(13))
310. H(55)=DOG*E(45)/(EXP0(25)-EXP0(18))
311. H(56)=(DOG*E(46)+ABLE*H(8)+BAKER*H(19)+CHARLY*H(35))/(EXP0(25)-EXP
312. 10(2))
313. H(57)=(DOG*E(47)+ABLE*H(9)+BAKER*H(20)+CHARLY*H(36))/(EXP0(25)-EXP
314. 10(7))
315. H(58)=(DOG*E(48)+BAKER*H(21)+CHARLY*H(37))/(EXP0(25)-EXP0(12))
316. H(59)=(DOG*E(49)+CHARLY*H(38))/(EXP0(25)-EXP0(17))
317. H(60)=DOG*E(50)/(EXP0(25)-EXP0(22))
318. H(61)=(DOG*E(51)+ABLE*H(10)+BAKER*H(22)+CHARLY*H(39))/(EXP0(25)-EX
319. 1P0(4))
320. H(62)=(DOG*E(52)+ABLE*H(11)+BAKER*H(23)+CHARLY*H(40))/(EXP0(25)-EX
321. 1P0(9))
322. H(63)=(DOG*E(53)+BAKER*H(24)+CHARLY*H(41))/(EXP0(25)-EXP0(14))
323. H(64)=(DOG*E(54)+CHARLY*H(42))/(EXP0(25)-EXP0(19))
324. H(65)=DOG*E(55)/(EXP0(25)-EXP0(24))
325. H(66)=(ABLE*H(12)+BAKER*H(25)+CHARLY*H(43))/(EXP0(25)-EXP0(5))
326. H(67)=(ABLE*H(13)+BAKER*H(26)+CHARLY*H(44))/(EXP0(25)-EXP0(10))
327. H(68)=(BAKER*H(27)+CHARLY*H(45))/(EXP0(25)-EXP0(15))
328. H(69)=CHARLY*H(46)/(EXP0(25)-EXP0(20))
329. IF(.NOT.JUMPOF,OR.JUMP7) GO TO 9
330. H(47)=H(47)+(ABLE*G(6)+BAKER*G(16)+CHARLY*G(31))/(EXP0(25)-EXP0(1)
331. 1)
332. H(48)=H(48)+(ABLE*G(7)+BAKER*G(17)+CHARLY*G(32))/(EXP0(25)-EXP0(6)
333. 1)
334. H(49)=H(49)+(BAKER*G(18)+CHARLY*G(33))/(EXP0(25)-EXP0(11))
335. H(50)=H(50)+CHARLY*G(34)/(EXP0(25)-EXP0(16))
336. H(52)=H(52)+(ABLE*G(8)+BAKER*G(19)+CHARLY*G(35))/(EXP0(25)-EXP0(3)
337. 1)
338. H(53)=H(53)+(ABLE*G(15)+BAKER*G(20)+CHARLY*G(36))/(EXP0(25)-EXP0(8)
339. 1)
340. H(54)=H(54)+(BAKER*G(30)+CHARLY*G(37))/(EXP0(25)-EXP0(13))
341. H(55)=H(55)+CHARLY*G(50)/(EXP0(25)-EXP0(18))
342. H(56)=H(56)+(ABLE*G(9)+BAKER*G(21)+CHARLY*G(38))/(EXP0(25)-EXP0(2)
343. 1)
344. H(57)=H(57)+(ABLE*G(10)+BAKER*G(22)+CHARLY*G(39))/(EXP0(25)-EXP0(7)
345. 1)
346. H(58)=H(58)+(BAKER*G(23)+CHARLY*G(40))/(EXP0(25)-EXP0(12))
347. H(59)=H(59)+CHARLY*G(41)/(EXP0(25)-EXP0(17))
348. H(61)=H(61)+(ABLE*G(11)+BAKER*G(24)+CHARLY*G(42))/(EXP0(25)-EXP0(4)
349. 1)
350. H(62)=H(62)+(ABLE*G(12)+BAKER*G(25)+CHARLY*G(43))/(EXP0(25)-EXP0(9)
351. 1)
352. H(63)=H(63)+(BAKER*G(26)+CHARLY*G(44))/(EXP0(25)-EXP0(14))
353. H(64)=H(64)+CHARLY*G(45)/(EXP0(25)-EXP0(19))
354. H(66)=H(66)+(ABLE*G(13)+BAKER*G(27)+CHARLY*G(46))/(EXP0(25)-EXP0(5)
355. 1)
356. H(67)=H(67)+(ABLE*G(14)+BAKER*G(28)+CHARLY*G(47))/(EXP0(25)-EXP0(1)
357. 10))
358. H(68)=H(68)+(BAKER*G(29)+CHARLY*G(48))/(EXP0(25)-EXP0(15))
359. H(69)=H(69)+CHARLY*G(49)/(EXP0(25)-EXP0(20))
360. 9 IF(.NOT.JUMPOF,OR.JUMP10) GO TO 10
361. H(47)=H(47)+(ABLE*G(5)+BAKER*G(13)+CHARLY*G(25))/(EXP0(25)-EXP0(1)
362. 1)
363. H(48)=H(48)+(ABLE*G(6)+BAKER*G(14)+CHARLY*G(26))/(EXP0(25)-EXP0(6)
364. 1)

```



```

365. H(49)=H(49)+(BAKER*G(15)+CHARLY*G(27))/(EXP0(25)-EXP0(11))
366. H(50)=H(50)+CHARLY*G(28)/(EXP0(25)-EXP0(16))
367. H(52)=H(52)+(ABLE*G(7)+BAKER*G(16)+CHARLY*G(29))/(EXP0(25)-EXP0(3))
368. 1)
369. H(53)=H(53)+(ABLE*G(12)+BAKER*G(17)+CHARLY*G(30))/(EXP0(25)-EXP0(8))
370. 1)
371. H(54)=H(54)+(BAKER*G(24)+CHARLY*G(31))/(EXP0(25)-EXP0(13))
372. H(55)=H(55)+CHARLY*G(40)/(EXP0(25)-EXP0(18))
373. H(56)=H(56)+(ABLE*G(8)+BAKER*G(18)+CHARLY*G(32))/(EXP0(25)-EXP0(2))
374. 1)
375. H(57)=H(57)+(ABLE*G(9)+BAKER*G(19)+CHARLY*G(33))/(EXP0(25)-EXP0(7))
376. 1)
377. H(58)=H(58)+(BAKER*G(20)+CHARLY*G(34))/(EXP0(25)-EXP0(12))
378. H(59)=H(59)+CHARLY*G(35)/(EXP0(25)-EXP0(17))
379. H(61)=H(61)+(ABLE*G(10)+BAKER*G(21)+CHARLY*G(36))/(EXP0(25)-EXP0(4))
380. 1)
381. H(62)=H(62)+(ABLE*G(11)+BAKER*G(22)+CHARLY*G(37))/(EXP0(25)-EXP0(9))
382. 1)
383. H(63)=H(63)+(BAKER*G(23)+CHARLY*G(38))/(EXP0(25)-EXP0(14))
384. H(64)=H(64)+CHARLY*G(39)/(EXP0(25)-EXP0(19))
385. 10 H(70)=MASS(ISOTOP,8)-SUMZ(H,47,69)
386. MASS(ISOTOP,8)= H(47)*EEXP0( 1)+H(48)*EEXP0( 6)+H(49)*EEXP0(11)+
387. 1H(50)*EEXP0(16)+H(51)*EEXP0(21)+H(52)*EEXP0( 3)+H(53)*EEXP0( 8)+
388. 2H(54)*EEXP0(13)+H(55)*EEXP0(18)
389. MASS(ISOTOP,8)=MASS(ISOTOP,8) +H(56)*EEXP0( 2)+H(57)*EEXP0( 7)+
390. 1H(58)*EEXP0(12)+H(59)*EEXP0(17)+H(60)*EEXP0(22)+H(61)*EEXP0( 4)+
391. 2H(62)*EEXP0( 9)+H(63)*EEXP0(14)+H(64)*EEXP0(19)+H(65)*EEXP0(24)
392. MASS(ISOTOP,8)=MASS(ISOTOP,8) +H(66)*EEXP0( 5)+H(67)*EEXP0(10)+
393. 1H(68)*EEXP0(15)+H(69)*EEXP0(20)+H(70)*EEXP0(25)
394. IF(JUMP7,OR,JUMPOP,OR,JUMPOF) GO TO 11
395. C CONSIDER FALL CUT IN BARRIER 3 03/25/71
396. O(51)=(FOR(3)*H(47)+ABLE*O(6)+BAKER*O(16)+CHARLY*O(31))/(EXP0(23)-
397. 1EXP0(1))
398. O(52)=(FOR(3)*H(48)+ABLE*O(7)+BAKER*O(17)+CHARLY*O(32))/(EXP0(23)-
399. 1EXP0(6))
400. O(53)=(FOR(3)*H(49)+BAKER*O(18)+CHARLY*O(33))/(EXP0(23)-EXP0(11))
401. O(54)=(FOR(3)*H(50)+CHARLY*O(34))/(EXP0(23)-EXP0(16))
402. O(55)=FOR(3)*H(51)/(EXP0(23)-EXP0(21))
403. O(56)=(FOR(3)*H(52)+ABLE*O(8)+BAKER*O(19)+CHARLY*O(35))/(EXP0(23)-
404. 1EXP0(3))
405. O(57)=(FOR(3)*H(53)+ABLE*O(15)+BAKER*O(20)+CHARLY*O(36))/(EXP0(23)-
406. 1EXP0(8))
407. O(58)=(FOR(3)*H(54)+BAKER*O(30)+CHARLY*O(37))/(EXP0(23)-EXP0(13))
408. O(59)=(FOR(3)*H(55)+CHARLY*O(50))/(EXP0(23)-EXP0(18))
409. O(60)=(FOR(3)*H(56)+ABLE*O(9)+BAKER*O(21)+CHARLY*O(38))/(EXP0(23)-
410. 1EXP0(2))
411. O(61)=(FOR(3)*H(57)+ABLE*O(10)+BAKER*O(22)+CHARLY*O(39))/(EXP0(23)-
412. 1EXP0(7))
413. O(62)=(FOR(3)*H(58)+BAKER*O(23)+CHARLY*O(40))/(EXP0(23)-EXP0(12))
414. O(63)=(FOR(3)*H(59)+CHARLY*O(41))/(EXP0(23)-EXP0(17))
415. O(64)=FOR(3)*H(60)/(EXP0(23)-EXP0(22))
416. O(65)=(FOR(3)*H(61)+ABLE*O(11)+BAKER*O(24)+CHARLY*O(42))/(EXP0(23)-
417. 1EXP0(4))
418. O(66)=(FOR(3)*H(62)+ABLE*O(12)+BAKER*O(25)+CHARLY*O(43))/(EXP0(23)-
419. 1EXP0(9))
420. O(67)=(FOR(3)*H(63)+BAKER*O(26)+CHARLY*O(44))/(EXP0(23)-EXP0(14))
421. O(68)=(FOR(3)*H(64)+CHARLY*O(45))/(EXP0(23)-EXP0(19))
422. O(69)=FOR(3)*H(65)/(EXP0(23)-EXP0(24))
423. O(70)=(FOR(3)*H(66)+ABLE*O(13)+BAKER*O(27)+CHARLY*O(46))/(EXP0(23)-
424. 1EXP0(5))
425. O(71)=(FOR(3)*H(67)+ABLE*O(14)+BAKER*O(28)+CHARLY*O(47))/(EXP0(23)-
426. 1EXP0(10))
427. O(72)=(FOR(3)*H(68)+BAKER*O(29)+CHARLY*O(48))/(EXP0(23)-EXP0(15))
428. O(73)=(FOR(3)*H(69)+CHARLY*O(49))/(EXP0(23)-EXP0(20))
429. O(74)=FOR(3)*H(70)/(EXP0(23)-EXP0(25))
430. O(75)=MASS(ISOTOP,9)-SUMZ(O,51,74)
431. MASS(ISOTOP,9)= O(51)*EEXP0( 1)+O(52)*EEXP0( 6)+O(53)*EEXP0(11)+
432. 1O(54)*EEXP0(16)+O(55)*EEXP0(21)+O(56)*EEXP0( 3)+O(57)*EEXP0( 8)+
433. 2O(58)*EEXP0(13)+O(59)*EEXP0(18)+O(75)*EEXP0(23)
434. MASS(ISOTOP,9)=MASS(ISOTOP,9) +O(60)*EEXP0( 2)+O(61)*EEXP0( 7)+
435. 1O(62)*EEXP0(12)+O(63)*EEXP0(17)+O(64)*EEXP0(22)+O(65)*EEXP0( 4)+
436. 2O(66)*EEXP0( 9)+O(67)*EEXP0(14)+O(68)*EEXP0(19)+O(69)*EEXP0(24)
437. MASS(ISOTOP,9)=MASS(ISOTOP,9) +O(70)*EEXP0( 5)+O(71)*EEXP0(10)+
438. 1O(72)*EEXP0(15)+O(73)*EEXP0(20)+O(74)*EEXP0(25)
439. 11 IF(JUMP8,OR,JUMPOP,OR,JUMPOF) GO TO 12
440. C CONSIDER FILTRATION IN BARRIER 3 03/25/71
441. DOG=1LR(6,7)*(1.-FILTER(ISOTOP,2))
442. P(51)=(DOG*H(47)+ABLE*P(6)+BAKER*P(16)+CHARLY*P(31))/(EXP0(23)-EXP
443. 1O(1))
444. P(52)=(DOG*H(48)+ABLE*P(7)+BAKER*P(17)+CHARLY*P(32))/(EXP0(23)-EXP
445. 1O(6))
446. P(53)=(DOG*H(49)+BAKER*P(18)+CHARLY*P(33))/(EXP0(23)-EXP0(11))
447. P(54)=(DOG*H(50)+CHARLY*P(34))/(EXP0(23)-EXP0(16))
448. P(55)=DOG*H(51)/(EXP0(23)-EXP0(21))
449. P(56)=(DOG*H(52)+ABLE*P(8)+BAKER*P(19)+CHARLY*P(35))/(EXP0(23)-EXP
450. 1O(3))
451. P(57)=(DOG*H(53)+ABLE*P(15)+BAKER*P(20)+CHARLY*P(36))/(EXP0(23)-EXP
452. 1O(8))
453. P(58)=(DOG*H(54)+BAKER*P(30)+CHARLY*P(37))/(EXP0(23)-EXP0(13))
454. P(59)=(DOG*H(55)+CHARLY*P(50))/(EXP0(23)-EXP0(18))
455. P(60)=(DOG*H(56)+ABLE*P(9)+BAKER*P(21)+CHARLY*P(38))/(EXP0(23)-EXP
456. 1O(2))

```

```

457. P(61)=(DOG*H(57)+ABLE*P(10)+BAKER*P(22)+CHARLY*P(39))/(EXP0(23)-EX
458. 1P0(7))
459. P(62)=(DOG*H(58)+BAKER*P(23)+CHARLY*P(40))/(EXP0(23)-EXP0(12))
460. P(63)=(DOG*H(59)+CHARLY*P(41))/(EXP0(23)-EXP0(17))
461. P(64)=DOG*H(60)/(EXP0(23)-EXP0(22))
462. P(65)=(DOG*H(61)+ABLE*P(11)+BAKER*P(24)+CHARLY*P(42))/(EXP0(23)-EX
463. 1P0(4))
464. P(66)=(DOG*H(62)+ABLE*P(12)+BAKER*P(25)+CHARLY*P(43))/(EXP0(23)-EX
465. 1P0(9))
466. P(67)=(DOG*H(63)+BAKER*P(26)+CHARLY*P(44))/(EXP0(23)-EXP0(14))
467. P(68)=(DOG*H(64)+CHARLY*P(45))/(EXP0(23)-EXP0(19))
468. P(69)=DOG*H(65)/(EXP0(23)-EXP0(24))
469. P(70)=(DOG*H(66)+ABLE*P(13)+BAKER*P(27)+CHARLY*P(46))/(EXP0(23)-EX
470. 1P0(5))
471. P(71)=(DOG*H(67)+ABLE*P(14)+BAKER*P(28)+CHARLY*P(47))/(EXP0(23)-EX
472. 1P0(10))
473. P(72)=(DOG*H(68)+BAKER*P(29)+CHARLY*P(48))/(EXP0(23)-EXP0(15))
474. P(73)=(DOG*H(69)+CHARLY*P(49))/(EXP0(23)-EXP0(20))
475. P(74)=DOG*H(70)/(EXP0(23)-EXP0(25))
476. P(75)=MASS(ISOTOP,10)-SUMZ(P,51,74)
477. MASS(ISOTOP,10)=P(51)*EEXP0(1)+P(52)*EEXP0(6)+P(53)*EEXP0(11)+
478. 1P(54)*EEXP0(16)+P(55)*EEXP0(21)+P(56)*EEXP0(3)+P(57)*EEXP0(8)+
479. 2P(58)*EEXP0(13)+P(59)*EEXP0(18)+P(75)*EEXP0(23)
480. MASS(ISOTOP,10)=MASS(ISOTOP,10)+P(60)*EEXP0(2)+P(61)*EEXP0(7)+
481. 1P(62)*EEXP0(12)+P(63)*EEXP0(17)+P(64)*EEXP0(22)+P(65)*EEXP0(4)+
482. 2P(66)*EEXP0(9)+P(67)*EEXP0(14)+P(68)*EEXP0(19)+P(69)*EEXP0(24)
483. MASS(ISOTOP,10)=MASS(ISOTOP,10)+P(70)*EEXP0(5)+P(71)*EEXP0(10)+
484. 1P(72)*EEXP0(15)+P(73)*EEXP0(20)+P(74)*EEXP0(25)
485. C CALCULATE RELEASE RATE TO ATMOSPHERE 03/25/71
486. 12 QUANTY(ISOTOP)=ALR(1)*FILTER(ISOTOP,1)*MASS(ISOTOP,1)+
487. A (ALR(2)+ALR(3)*FILTER(ISOTO
488. 1P,2))*MASS(ISOTOP,2)+(ALR(4)+ALR(5)*FILTER(ISOTOP,2))*MASS(ISOTOP,
489. 25)+(ALR(6)+ALR(7)*FILTER(ISOTOP,2))*MASS(ISOTOP,5)
490. IF(.NOT.JUMPOF)GO TO 13
491. QUANTY(ISOTOP)=QUANTY(ISOTOP)+ALR(3)/ALR(3)*(ABLE*MASS(ISOTOP-3,4)
492. 1+BAKER*MASS(ISOTOP-2,4)+CHARLY*MASS(ISOTOP-1,4))+ALR(5)/ALR(5)*
493. 2ABLE*MASS(ISOTOP-3,7)+BAKER*MASS(ISOTOP-2,7)+CHARLY*MASS(ISOTOP-1,
494. 37))+ALR(7)/ALR(7)*(ABLE*MASS(ISOTOP-3,10)+BAKER*MASS(ISOTOP-2,10)
495. 4+CHARLY*MASS(ISOTOP-1,10))
496. C RETURN TO BAFFLE 03/25/71
497. 13 RETURN
498. END

```

```

1. C SUBROUTINE SIXTH DOES CONTAINMENT CALCULATION FOR SIXTH CHAIN MEMBE03/25/71
2. SUBROUTINE SIXTH 03/25/71
3. COMMON ISOTOP,SPACE(97),DELTAT,MASS(600,10),HOLE(2850),DKFRCT(450,
4. A3),FILTER(600,2),LAMBDA(600),SPOT(96),H(133),A(26),O(105),P(105),
5. 2G(34),F(84),E(105),D(63),C(63),B(77),FOR(3),ALR(7),ILR(7.8),OLR(7),
6. 3,QUARTY(600),EXP0(35),EEXP0(35),SKIP(1200),JUMP1,JUMP4,JUMP2,JUMP3
7. 4,JUMPS,JUMP6,JUMP8,JUMP7,JUMPS,JUMP10,JUMPOF,JUMPUP
8. REAL MASS,ILR,LAMBDA
9. LOGICAL JUMP1,JUMP2,JUMP3,JUMP4,JUMPS,JUMP6,JUMP7,JUMP8
10. LOGICAL JUMP9,JUMP10,JUMPOF,JUMPUP
11. C A REPRESENTS COEFFICIENTS FOR FUEL, B FOR 1ST BARRIER, C FOR 1ST FALL03/25/71
12. C 03/25/71
13. C D FOR 1ST FILTER, E FOR 2ND BARRIER, F FOR 2ND FALLOUT, G FOR 2ND FIL03/25/71
14. C H FOR 3RD BARRIER, O FOR 3RD FALLOUT, P FOR 3RD FILTER. 03/25/71
15. C 03/25/71
16. EXP0(26)=LAMBDA(ISOTOP)+OLR(1)*FILTER(ISOTOP,1)
17. EEXP0(26)=EXP(-EXP0(26)*DELTAT)
18. ABLE=LAMBDA(ISOTOP-3)*DKFRCT(ISOTOP,3)
19. BAKER=LAMBDA(ISOTOP-2)*DKFRCT(ISOTOP,2)
20. CHARLY=LAMBDA(ISOTOP-1)*DKFRCT(ISOTOP,1)
21. A(16)=(ABLE*A(4)+BAKER*A(7)+CHARLY*A(11))/(EXP0(26)-EXP0(1))
22. A(17)=(ABLE*A(5)+BAKER*A(8)+CHARLY*A(12))/(EXP0(26)-EXP0(6))
23. A(18)=(ABLE*A(6)+BAKER*A(9)+CHARLY*A(13))/(EXP0(26)-EXP0(11))
24. A(19)=(BAKER*A(10)+CHARLY*A(14))/(EXP0(26)-EXP0(16))
25. A(20)=CHARLY*A(15)/(EXP0(26)-EXP0(21))
26. A(21)=MASS(ISOTOP,1)-SUMZ(A,16,20)
27. MASS(ISOTOP,1)=A(16)*EEXP0(1)+A(17)*EEXP0(6)+A(18)*EEXP0(11)+A(19)
28. 1+EEXP0(16)+A(20)*EEXP0(21)+A(21)*EEXP0(26)
29. C CONSIDER BARRIER 1 03/25/71
30. EXP0(27)=LAMBDA(ISOTOP)+FOR(1)+OLR(2)
31. EEXP0(27)=EXP(-EXP0(27)*DELTAT)
32. DOG=ILR(1,2)*FILTER(ISOTOP,1)
33. B(41)=(DOG*A(16)+ABLE*B(3)+BAKER*B(16)+CHARLY*B(27))/(EXP0(27)-EXP
34. 10(1))
35. B(42)=(DOG*A(17)+ABLE*B(9)+BAKER*B(17)+CHARLY*B(28))/(EXP0(27)-EXP
36. 10(6))
37. B(43)=(DOG*A(18)+ABLE*B(10)+BAKER*B(18)+CHARLY*B(29))/(EXP0(27)-EX
38. 10(11))
39. B(44)=(DOG*A(19)+BAKER*B(19)+CHARLY*B(30))/(EXP0(27)-EXP0(16))
40. B(45)=(DOG*A(20)+CHARLY*B(31))/(EXP0(27)-EXP0(21))
41. B(46)=DOG*A(21)/(EXP0(27)-EXP0(26))
42. B(47)=(ABLE*B(11)+BAKER*B(20)+CHARLY*B(32))/(EXP0(27)-EXP0(3))
43. B(48)=(ABLE*B(12)+BAKER*B(21)+CHARLY*B(33))/(EXP0(27)-EXP0(8))
44. B(49)=(BAKER*B(22)+CHARLY*B(34))/(EXP0(27)-EXP0(13))
45. B(50)=CHARLY*B(35)/(EXP0(27)-EXP0(18))
46. B(52)=(ABLE*B(13)+BAKER*B(23)+CHARLY*B(36))/(EXP0(27)-EXP0(2))
47. B(53)=(ABLE*B(14)+BAKER*B(24)+CHARLY*B(37))/(EXP0(27)-EXP0(7))
48. B(54)=(ABLE*B(15)+BAKER*B(25)+CHARLY*B(38))/(EXP0(27)-EXP0(12))
49. B(55)=(BAKER*B(26)+CHARLY*B(39))/(EXP0(27)-EXP0(17))
50. B(56)=CHARLY*B(40)/(EXP0(27)-EXP0(22))
51. CALL ZERO(D,46,63)
52. CALL ZERO(C,46,63)
53. EXP0(28)=LAMBDA(ISOTOP)
54. EEXP0(28)=EXP(-EXP0(28)*DELTAT)
55. IF (JUMP1.OR..NOT.JUMPUP) GO TO 1
56. B(41)=B(41)+(ABLE*C(10)+BAKER*C(19)+CHARLY*C(31))/(EXP0(27)-EXP0(1
57. 1))
58. B(42)=B(42)+(ABLE*C(11)+BAKER*C(20)+CHARLY*C(32))/(EXP0(27)-EXP0(6
59. 1))
60. B(43)=B(43)+(ABLE*C(12)+BAKER*C(21)+CHARLY*C(33))/(EXP0(27)-EXP0(1
61. 1))
62. B(44)=B(44)+(BAKER*C(22)+CHARLY*C(34))/(EXP0(27)-EXP0(16))
63. B(45)=B(45)+CHARLY*C(35)/(EXP0(27)-EXP0(21))
64. B(47)=B(47)+(ABLE*C(13)+BAKER*C(23)+CHARLY*C(36))/(EXP0(27)-EXP0(3
65. 1))
66. B(48)=B(48)+(ABLE*C(14)+BAKER*C(24)+CHARLY*C(37))/(EXP0(27)-EXP0(8
67. 1))
68. B(49)=B(49)+(ABLE*C(18)+BAKER*C(25)+CHARLY*C(38))/(EXP0(27)-EXP0(1
69. 13))
70. B(50)=B(50)+(BAKER*C(30)+CHARLY*C(39))/(EXP0(27)-EXP0(18))
71. B(51)=CHARLY*C(45)/(EXP0(27)-EXP0(23))
72. B(52)=B(52)+(ABLE*C(15)+BAKER*C(26)+CHARLY*C(40))/(EXP0(27)-EXP0(2
73. 1))
74. B(53)=B(53)+(ABLE*C(16)+BAKER*C(27)+CHARLY*C(41))/(EXP0(27)-EXP0(7
75. 1))
76. B(54)=B(54)+(ABLE*C(17)+BAKER*C(28)+CHARLY*C(42))/(EXP0(27)-EXP0(1
77. 12))
78. B(55)=B(55)+(BAKER*C(29)+CHARLY*C(43))/(EXP0(27)-EXP0(17))
79. B(56)=B(56)+CHARLY*C(44)/(EXP0(27)-EXP0(22))
80. GO TO 2
81. 1 U(51)=0.0
82. 2 U(57)=MASS(ISOTOP,2)-SUMZ(B,41,56)
83. MASS(ISOTOP,2)=B(41)*EEXP0(1)+B(42)*EEXP0(6)+B(43)*EEXP0(11)+B(44)
84. 1+EEXP0(16)+B(45)*EEXP0(21)+B(46)*EEXP0(26)+B(47)*EEXP0(3)+B(48)*E
85. 2+EXP0(8)+B(49)*EEXP0(13)+B(50)*EEXP0(18)+B(51)*EEXP0(23)
86. MASS(ISOTOP,2)=MASS(ISOTOP,2)+B(52)*EEXP0(2)+B(53)*EEXP0(7)+B(54)*
87. 1+EEXP0(12)+B(55)*EEXP0(17)+B(56)*EEXP0(22)+B(57)*EEXP0(27)
88. IF (JUMP1.OR.JUMPUP.OR.JUMPOF) GO TO 3

```

89. C CONSIDER FALLOUT IN BARRIER 1 03/25/71
90. C(46)=(FOR(1)*B(41)+ABLE*C(10)+BAKER*C(19)+CHARLY*C(31))/(EXP0(28)
91. 1-EXPC(1))
92. C(47)=(FOR(1)*B(42)+ABLE*C(11)+BAKER*C(20)+CHARLY*C(32))/(EXP0(28)
93. 1-EXPC(6))
94. C(48)=(FOR(1)*B(43)+ABLE*C(12)+BAKER*C(21)+CHARLY*C(33))/(EXP0(28)
95. 1-EXPC(11))
96. C(49)=(FOR(1)*B(44)+BAKER*C(22)+CHARLY*C(34))/(EXP0(28)-EXP0(16))
97. C(50)=(FOR(1)*B(45)+CHARLY*C(35))/(EXP0(28)-EXP0(21))
98. C(51)=(FOR(1)*B(46))/(EXP0(28)-EXP0(26))
99. C(52)=(FOR(1)*B(47)+ABLE*C(13)+BAKER*C(23)+CHARLY*C(36))/(EXP0(28)
100. 1-EXPC(3))
101. C(53)=(FOR(1)*B(48)+ABLE*C(14)+BAKER*C(24)+CHARLY*C(37))/(EXP0(28)
102. 1-EXPC(8))
103. C(54)=(FOR(1)*B(49)+ABLE*C(18)+BAKER*C(25)+CHARLY*C(38))/(EXP0(28)
104. 1-EXPC(13))
105. C(55)=(FOR(1)*B(50)+BAKER*C(30)+CHARLY*C(39))/(EXP0(28)-EXP0(18))
106. C(56)=(FOR(1)*B(51)+CHARLY*C(45))/(EXP0(28)-EXP0(23))
107. C(57)=(FOR(1)*B(52)+ABLE*C(15)+BAKER*C(26)+CHARLY*C(40))/(EXP0(28)
108. 1-EXPC(2))
109. C(58)=(FOR(1)*B(53)+ABLE*C(16)+BAKER*C(27)+CHARLY*C(41))/(EXP0(28)
110. 1-EXPC(7))
111. C(59)=(FOR(1)*B(54)+ABLE*C(17)+BAKER*C(28)+CHARLY*C(42))/(EXP0(28)
112. 1-EXPC(12))
113. C(60)=(FOR(1)*B(55)+BAKER*C(29)+CHARLY*C(43))/(EXP0(28)-EXP0(17))
114. C(61)=(FOR(1)*B(56)+CHARLY*C(44))/(EXP0(28)-EXP0(22))
115. C(62)=(FOR(1)*B(57))/(EXP0(28)-EXP0(27))
116. C(63)=MASS(ISOTOP,3)-SUMZ(C(46),62)
117. MASS(ISOTOP,3)=C(46)*EEXPC(1)+C(47)*EEXPC(6)+C(48)*EEXPC(11)+C(49)
118. 1+EEXPC(16)+C(50)*EEXPC(21)+C(51)*EEXPC(26)+C(52)*EEXPC(3)+C(53)*E
119. 2XPC(8)+C(54)*EEXPC(13)+C(55)*EEXPC(18)+C(56)*EEXPC(23)+C(63)*EEXPC
120. 3(28)
121. MASS(ISOTOP,3)=MASS(ISOTOP,3)+C(57)*EEXPC(2)+C(58)*EEXPC(7)+C(59)*
122. 1EEXPC(12)+C(60)*EEXPC(17)+C(61)*EEXPC(22)+C(62)*EEXPC(27)
123. 3 IF(JUMPOF,OR,JUMP2) GO TO 4
124. C CONSIDER FILTRATION IN BARRIER 1 03/25/71
125. DOG=ILR(2,3)*(1.-FILTER(ISOTOP,2))
126. D(46)=(DOG*B(41)+ABLE*D(10)+BAKER*D(19)+CHARLY*D(31))/(EXP0(28)-EX
127. 1PC(1))
128. D(47)=(DOG*B(42)+ABLE*D(11)+BAKER*D(20)+CHARLY*D(32))/(EXP0(28)-EX
129. 1PC(6))
130. D(48)=(DOG*B(43)+ABLE*D(12)+BAKER*D(21)+CHARLY*D(33))/(EXP0(28)-EX
131. 1PC(11))
132. D(49)=(DOG*B(44)+BAKER*D(22)+CHARLY*D(34))/(EXP0(28)-EXPC(16))
133. D(50)=(DOG*B(45)+CHARLY*D(35))/(EXP0(28)-EXP0(21))
134. D(51)=DOG*B(46)/(EXP0(28)-EXP0(26))
135. D(52)=(DOG*B(47)+ABLE*D(13)+BAKER*D(23)+CHARLY*D(36))/(EXP0(28)-EX
136. 1PC(3))
137. D(53)=(DOG*B(48)+ABLE*D(14)+BAKER*D(24)+CHARLY*D(37))/(EXP0(28)-EX
138. 1PC(8))
139. D(54)=(DOG*B(49)+ABLE*D(18)+BAKER*D(25)+CHARLY*D(38))/(EXP0(28)-EX
140. 1PC(13))
141. D(55)=(DOG*B(50)+BAKER*D(30)+CHARLY*D(39))/(EXP0(28)-EXP0(18))
142. D(56)=(DOG*B(51)+CHARLY*D(45))/(EXP0(28)-EXP0(23))
143. D(57)=(DOG*B(52)+ABLE*D(15)+BAKER*D(26)+CHARLY*D(40))/(EXP0(28)-EX
144. 1PC(2))
145. D(58)=(DOG*B(53)+ABLE*D(16)+BAKER*D(27)+CHARLY*D(41))/(EXP0(28)-EX
146. 1PC(7))
147. D(59)=(DOG*B(54)+ABLE*D(17)+BAKER*D(28)+CHARLY*D(42))/(EXP0(28)-EX
148. 1PC(12))
149. D(60)=(DOG*B(55)+BAKER*D(29)+CHARLY*D(43))/(EXP0(28)-EXPC(17))
150. D(61)=(DOG*B(56)+CHARLY*D(44))/(EXP0(28)-EXP0(22))
151. D(62)=DOG*B(57)/(EXP0(28)-EXP0(27))
152. C(63)=MASS(ISOTOP,4)-SUMZ(D(46),62)
153. MASS(ISOTOP,4)=D(46)*ELXPC(1)+D(47)*EEXPC(6)+D(48)*EEXPC(11)+D(49)
154. 1+EEXPC(16)+D(50)*EEXPC(21)+D(51)*EEXPC(26)+D(52)*EEXPC(3)+D(53)*E
155. 2XPC(8)+D(54)*EEXPC(13)+D(55)*EEXPC(18)+D(56)*EEXPC(23)+D(63)*EEXPC
156. 3(28)
157. MASS(ISOTOP,4)=MASS(ISOTOP,4)+D(57)*EEXPC(2)+D(58)*ELXPC(7)+D(59)*
158. 1EEXPC(12)+D(60)*EEXPC(17)+D(61)*EEXPC(22)+D(62)*EEXPC(27)
159. 4 IF(JUMP3) GO TO 3
160. C CONSIDER BARRIER 2 03/25/71
161. CALL ZERO(F,61,84)
162. CALL ZERO(G,61,84)
163. EXPC(29)=LAMEDA(ISOTOP)+FOR(2)+OLR(4)
164. EEXPC(29)=EXP(-EXPC(29)*DELTA)
165. EASY=ILR(1,4)*FILTER(ISOTOP,1)
166. DOG=ILR(2,4)+ILR(3,4)*FILTER(ISOTOP,2)
167. E(56)=(EASY*A(16)+DOG*B(41)+ABLE*E(11)+BAKER*E(22)+CHARLY*E(37))/(
168. 1EXPC(29)-EXPC(1))
169. E(57)=(EASY*A(17)+DOG*B(42)+ABLE*E(12)+BAKER*E(23)+CHARLY*E(38))/(
170. 1EXPC(29)-EXPC(6))
171. E(58)=(EASY*A(18)+DOG*B(43)+ABLE*E(13)+BAKER*E(24)+CHARLY*E(39))/(
172. 1EXPC(29)-EXPC(11))
173. E(59)=(EASY*A(19)+DOG*B(44)+BAKER*E(25)+CHARLY*E(40))/(EXP0(29)-EX
174. 1PC(16))
175. E(60)=(EASY*A(20)+DOG*B(45)+CHARLY*E(41))/(EXP0(29)-EXP0(21))
176. E(61)=(EASY*A(21)+DOG*B(46))/(EXP0(29)-EXP0(26))
177. E(62)=(DOG*B(47)+ABLE*E(14)+BAKER*E(26)+CHARLY*E(42))/(EXP0(29)-EX
178. 1PC(3))
179. E(63)=(DOG*B(48)+ABLE*E(15)+BAKER*E(27)+CHARLY*E(43))/(EXP0(29)-EX
180. 1PC(8))

```

181. E(64)=(DOG*B(49)+BAKER*E(29)+CHARLY*E(44))/(EXP0(29)-EXP0(13))
182. E(65)=(DOG*B(50)+CHARLY*E(45))/(EXP0(29)-EXP0(18))
183. E(66)=DOG*B(51)/(EXP0(29)-EXP0(23))
184. E(67)=(DOG*B(52)+ABLE*E(16)+BAKER*E(29)+CHARLY*E(46))/(EXP0(29)-EXP0(2))
185. 1P0(2))
186. E(68)=(DOG*B(53)+ABLE*E(17)+BAKER*E(30)+CHARLY*E(47))/(EXP0(29)-EXP0(7))
187. 1P0(7))
188. E(69)=(DOG*B(54)+ABLE*E(18)+BAKER*E(31)+CHARLY*E(48))/(EXP0(29)-EXP0(12))
189. 1P0(12))
190. E(70)=(DOG*B(55)+BAKER*E(32)+CHARLY*E(49))/(EXP0(29)-EXP0(17))
191. E(71)=(DOG*B(56)+CHARLY*E(50))/(EXP0(29)-EXP0(22))
192. E(72)=DOG*B(57)/(EXP0(29)-EXP0(27))
193. E(73)=(ABLE*E(19)+BAKER*E(33)+CHARLY*E(51))/(EXP0(29)-EXP0(4))
194. E(74)=(ABLE*E(20)+BAKER*E(34)+CHARLY*E(52))/(EXP0(29)-EXP0(9))
195. E(75)=(ABLE*E(21)+BAKER*E(35)+CHARLY*E(53))/(EXP0(29)-EXP0(14))
196. E(76)=(BAKER*E(36)+CHARLY*E(54))/(EXP0(29)-EXP0(19))
197. E(77)=CHARLY*E(55)/(EXP0(29)-EXP0(24))
198. IF(.NOT.JUMPOF,OR.JUMP4) GO TO 5
199. E(56)=E(56)+(ABLE*F(13)+BAKER*F(25)+CHARLY*F(41))/(EXP0(29)-EXP0(1))
200. 1))
201. E(57)=E(57)+(ABLE*F(14)+BAKER*F(26)+CHARLY*F(42))/(EXP0(29)-EXP0(6))
202. 1))
203. E(58)=E(58)+(ABLE*F(15)+BAKER*F(27)+CHARLY*F(43))/(EXP0(29)-EXP0(11))
204. 1))
205. E(59)=E(59)+(BAKER*F(28)+CHARLY*F(44))/(EXP0(29)-EXP0(16))
206. E(60)=E(60)+CHARLY*F(45)/(EXP0(29)-EXP0(21))
207. E(62)=E(62)+(ABLE*F(16)+BAKER*F(29)+CHARLY*F(46))/(EXP0(29)-EXP0(3))
208. 1))
209. E(63)=E(63)+(ABLE*F(17)+BAKER*F(30)+CHARLY*F(47))/(EXP0(29)-EXP0(8))
210. 1))
211. E(64)=E(64)+(ABLE*F(24)+BAKER*F(31)+CHARLY*F(48))/(EXP0(29)-EXP0(13))
212. 13))
213. E(65)=E(65)+(BAKER*F(40)+CHARLY*F(49))/(EXP0(29)-EXP0(18))
214. E(66)=E(66)+CHARLY*F(60)/(EXP0(29)-EXP0(23))
215. E(67)=E(67)+(ABLE*F(18)+BAKER*F(32)+CHARLY*F(50))/(EXP0(29)-EXP0(2))
216. 1))
217. E(68)=E(68)+(ABLE*F(19)+BAKER*F(33)+CHARLY*F(51))/(EXP0(29)-EXP0(7))
218. 1))
219. E(69)=E(69)+(ABLE*F(20)+BAKER*F(34)+CHARLY*F(52))/(EXP0(29)-EXP0(12))
220. 12))
221. E(70)=E(70)+(BAKER*F(35)+CHARLY*F(53))/(EXP0(29)-EXP0(17))
222. E(71)=E(71)+CHARLY*F(54)/(EXP0(29)-EXP0(22))
223. E(73)=E(73)+(ABLE*F(21)+BAKER*F(36)+CHARLY*F(55))/(EXP0(29)-EXP0(4))
224. 1))
225. E(74)=E(74)+(ABLE*F(22)+BAKER*F(37)+CHARLY*F(56))/(EXP0(29)-EXP0(9))
226. 1))
227. E(75)=E(75)+(ABLE*F(23)+BAKER*F(38)+CHARLY*F(57))/(EXP0(29)-EXP0(14))
228. 14))
229. E(76)=E(76)+(BAKER*F(39)+CHARLY*F(58))/(EXP0(29)-EXP0(19))
230. E(77)=E(77)+CHARLY*F(59)/(EXP0(29)-EXP0(24))
231. 5 IF(.NOT.JUMPOF,OR.JUMP9) GO TO 6
232. E(56)=E(56)+(ABLE*D(10)+BAKER*D(19)+CHARLY*D(31))/(EXP0(29)-EXP0(1))
233. 1))
234. E(57)=E(57)+(ABLE*D(11)+BAKER*D(20)+CHARLY*D(32))/(EXP0(29)-EXP0(6))
235. 1))
236. E(58)=E(58)+(ABLE*D(12)+BAKER*D(21)+CHARLY*D(33))/(EXP0(29)-EXP0(11))
237. 11))
238. E(59)=E(59)+(BAKER*D(22)+CHARLY*D(34))/(EXP0(29)-EXP0(16))
239. E(60)=E(60)+CHARLY*D(35)/(EXP0(29)-EXP0(21))
240. E(62)=E(62)+(ABLE*D(13)+BAKER*D(23)+CHARLY*D(36))/(EXP0(29)-EXP0(3))
241. 1))
242. E(63)=E(63)+(ABLE*D(14)+BAKER*D(24)+CHARLY*D(37))/(EXP0(29)-EXP0(8))
243. 1))
244. E(64)=E(64)+(ABLE*D(15)+BAKER*D(25)+CHARLY*D(38))/(EXP0(29)-EXP0(13))
245. 13))
246. E(65)=E(65)+(BAKER*D(30)+CHARLY*D(39))/(EXP0(29)-EXP0(18))
247. E(66)=E(66)+CHARLY*D(45)/(EXP0(29)-EXP0(23))
248. E(67)=E(67)+(ABLE*D(15)+BAKER*D(26)+CHARLY*D(40))/(EXP0(29)-EXP0(2))
249. 1))
250. E(68)=E(68)+(ABLE*D(16)+BAKER*D(27)+CHARLY*D(41))/(EXP0(29)-EXP0(7))
251. 1))
252. E(69)=E(69)+(ABLE*D(17)+BAKER*D(28)+CHARLY*D(42))/(EXP0(29)-EXP0(12))
253. 12))
254. E(70)=E(70)+(BAKER*D(29)+CHARLY*D(43))/(EXP0(29)-EXP0(17))
255. E(71)=E(71)+CHARLY*D(44)/(EXP0(29)-EXP0(22))
256. 6 E(78)=MASS(ISOTOP,5)-SUMZ(E,56,77)
257. MASS(ISOTOP,5)=E(56)*EEXP0(1)+E(57)*EEXP0(6)+E(58)*EEXP0(11)+E(59)*
258. 1+EEXP0(16)+E(60)*EEXP0(21)+E(61)*EEXP0(26)+E(62)*EEXP0(3)+E(63)*LE
259. 2XPO(8)+E(64)*EEXP0(13)+E(65)*EEXP0(18)+E(66)*EEXP0(23)
260. MASS(ISOTOP,5)=MASS(ISOTOP,5)+E(67)*EEXP0(2)+E(68)*EEXP0(7)+E(69)*
261. 1+EEXP0(12)+E(70)*EEXP0(17)+E(71)*EEXP0(22)+E(72)*EEXP0(27)+E(73)*LE
262. 2XPO(4)+E(74)*EEXP0(9)+E(75)*EEXP0(14)+E(76)*EEXP0(19)+E(77)*EEXP0(
263. 324)+E(78)*EEXP0(29)
264. IF(JUMP4,OR.JUMPOF,OR.JUMPOF) GO TO 7
265. C CONSIDER FALLOUT IN BARRIER 2
266. F(61)=(FOR(2)*E(56)+ABLE*F(13)+BAKER*F(25)+CHARLY*F(41))/(EXP0(28)-EXP0(1))
267. 1-EXP0(1))
268. F(62)=(FOR(2)*E(57)+ABLE*F(14)+BAKER*F(26)+CHARLY*F(42))/(EXP0(28)-EXP0(6))
269. 1-EXP0(6))
270. F(63)=(FOR(2)*E(58)+ABLE*F(15)+BAKER*F(27)+CHARLY*F(43))/(EXP0(28)-EXP0(11))
271. 1-EXP0(11))
272. F(64)=(FOR(2)*E(59)+BAKER*F(28)+CHARLY*F(44))/(EXP0(28)-EXP0(16))

```

```

273. F(65)=(FOR(2)*E(60)+CHARLY*F(45))/(EXP0(28)-EXP0(21))
274. F(66)=(FOR(2)*E(61))/(EXP0(28)-EXP0(26))
275. F(67)=(FOR(2)*E(62)+ABLE*F(16)+BAKER*F(29)+CHARLY*F(46))/(EXP0(28)
276. 1-EXP0(3))
277. F(68)=(FOR(2)*E(63)+ABLE*F(17)+BAKER*F(30)+CHARLY*F(47))/(EXP0(28)
278. 1-EXP0(8))
279. F(69)=(FOR(2)*E(64)+ABLE*F(24)+BAKER*F(31)+CHARLY*F(48))/(EXP0(28)
280. 1-EXP0(13))
281. F(70)=(FOR(2)*E(65)+BAKER*F(40)+CHARLY*F(49))/(EXP0(28)-EXP0(18))
282. F(71)=(FOR(2)*E(66)+CHARLY*F(60))/(EXP0(28)-EXP0(23))
283. F(72)=(FOR(2)*E(67)+ABLE*F(18)+BAKER*F(32)+CHARLY*F(50))/(EXP0(28)
284. 1-EXP0(2))
285. F(73)=(FOR(2)*E(68)+ABLE*F(19)+BAKER*F(33)+CHARLY*F(51))/(EXP0(28)
286. 1-EXP0(7))
287. F(74)=(FOR(2)*E(69)+ABLE*F(20)+BAKER*F(34)+CHARLY*F(52))/(EXP0(28)
288. 1-EXP0(12))
289. F(75)=(FOR(2)*E(70)+BAKER*F(35)+CHARLY*F(53))/(EXP0(28)-EXP0(17))
290. F(76)=(FOR(2)*E(71)+CHARLY*F(54))/(EXP0(28)-EXP0(22))
291. F(77)=(FOR(2)*E(72))/(EXP0(28)-EXP0(27))
292. F(78)=(FOR(2)*E(73)+ABLE*F(21)+BAKER*F(36)+CHARLY*F(55))/(EXP0(28)
293. 1-EXP0(4))
294. F(79)=(FOR(2)*E(74)+ABLE*F(22)+BAKER*F(37)+CHARLY*F(56))/(EXP0(28)
295. 1-EXP0(9))
296. F(80)=(FOR(2)*E(75)+ABLE*F(23)+BAKER*F(38)+CHARLY*F(57))/(EXP0(28)
297. 1-EXP0(14))
298. F(81)=(FOR(2)*E(76)+BAKER*F(39)+CHARLY*F(58))/(EXP0(28)-EXP0(19))
299. F(82)=(FOR(2)*E(77)+CHARLY*F(59))/(EXP0(28)-EXP0(24))
300. F(83)=(FOR(2)*E(78))/(EXP0(28)-EXP0(29))
301. F(84)=MASS(ISOTOP,6)-SUM2(F,61,83)
302. MASS(ISOTOP,6)=F(61)*EEXP0(1)+F(62)*EEXP0(6)+F(63)*EEXP0(11)+F(64)
303. 1*EEXP0(16)+F(65)*EEXP0(21)+F(66)*EEXP0(26)+F(67)*EEXP0(3)+F(68)*EE
304. 2XP0(8)+F(69)*EEXP0(13)+F(70)*EEXP0(18)+F(71)*EEXP0(23)+F(84)*EEXP0
305. 3(28)
306. MASS(ISOTOP,6)=MASS(ISOTOP,6)+F(72)*EEXP0(2)+F(73)*EEXP0(7)+F(74)*
307. 1EEXP0(12)+F(75)*EEXP0(17)+F(76)*EEXP0(22)+F(77)*EEXP0(27)+F(78)*EE
308. 2XP0(4)+F(79)*EEXP0(9)+F(80)*EEXP0(14)+F(81)*EEXP0(19)+F(82)*EEXP0(
309. 324)+F(83)*EEXP0(29)
310. 7 IF(JUMPOF,CR,JUMP5) GO TO 8
311. C CONSIDER FILTRATION IN BARRIER 2 03/25/71
312. DOG=ILR(4,5)*(1.-FILTER(ISOTOP,2))
313. G(61)=(DOG*E(56)+ABLE*G(13)+BAKER*G(25)+CHARLY*G(41))/(EXP0(28)-EX
314. 1P0(1))
315. G(62)=(DOG*E(57)+ABLE*G(14)+BAKER*G(26)+CHARLY*G(42))/(EXP0(28)-EX
316. 1P0(6))
317. G(63)=(DOG*E(58)+ABLE*G(15)+BAKER*G(27)+CHARLY*G(43))/(EXP0(28)-EX
318. 1P0(11))
319. G(64)=(DOG*E(59)+BAKER*G(28)+CHARLY*G(44))/(EXP0(28)-EXP0(16))
320. G(65)=(DOG*E(60)+CHARLY*G(45))/(EXP0(28)-EXP0(21))
321. G(66)=DOG*E(61)/(EXP0(28)-EXP0(26))
322. G(67)=(DOG*E(62)+ABLE*G(16)+BAKER*G(29)+CHARLY*G(46))/(EXP0(28)-EX
323. 1P0(3))
324. G(68)=(DOG*E(63)+ABLE*G(17)+BAKER*G(30)+CHARLY*G(47))/(EXP0(28)-EX
325. 1P0(8))
326. G(69)=(DOG*E(64)+ABLE*G(24)+BAKER*G(31)+CHARLY*G(48))/(EXP0(28)-EX
327. 1P0(13))
328. G(70)=(DOG*E(65)+BAKER*G(40)+CHARLY*G(49))/(EXP0(28)-EXP0(18))
329. G(71)=(DOG*E(66)+CHARLY*G(60))/(EXP0(28)-EXP0(23))
330. G(72)=(DOG*E(67)+ABLE*G(18)+BAKER*G(32)+CHARLY*G(50))/(EXP0(28)-EX
331. 1P0(2))
332. G(73)=(DOG*E(68)+ABLE*G(19)+BAKER*G(33)+CHARLY*G(51))/(EXP0(28)-EX
333. 1P0(7))
334. G(74)=(DOG*E(69)+ABLE*G(20)+BAKER*G(34)+CHARLY*G(52))/(EXP0(28)-EX
335. 1P0(12))
336. G(75)=(DOG*E(70)+BAKER*G(35)+CHARLY*G(53))/(EXP0(28)-EXP0(17))
337. G(76)=(DOG*E(71)+CHARLY*G(54))/(EXP0(28)-EXP0(22))
338. G(77)=DOG*E(72)/(EXP0(28)-EXP0(27))
339. G(78)=(DOG*E(73)+ABLE*G(21)+BAKER*G(36)+CHARLY*G(55))/(EXP0(28)-EX
340. 1P0(4))
341. G(79)=(DOG*E(74)+ABLE*G(22)+BAKER*G(37)+CHARLY*G(56))/(EXP0(28)-EX
342. 1P0(9))
343. G(80)=(DOG*E(75)+ABLE*G(23)+BAKER*G(38)+CHARLY*G(57))/(EXP0(28)-EX
344. 1P0(14))
345. G(81)=(DOG*E(76)+BAKER*G(39)+CHARLY*G(58))/(EXP0(28)-EXP0(19))
346. G(82)=(DOG*E(77)+CHARLY*G(59))/(EXP0(28)-EXP0(24))
347. G(83)=DOG*E(78)/(EXP0(28)-EXP0(29))
348. G(84)=MASS(ISOTOP,7)-SUM2(G,61,83)
349. MASS(ISOTOP,7)=G(61)*EEXP0(1)+G(62)*EEXP0(6)+G(63)*EEXP0(11)+G(64)
350. 1*EEXP0(16)+G(65)*EEXP0(21)+G(66)*EEXP0(26)+G(67)*EEXP0(3)+G(68)*EE
351. 2XP0(8)+G(69)*EEXP0(13)+G(70)*EEXP0(18)+G(71)*EEXP0(23)+G(84)*EEXP0
352. 3(28)
353. MASS(ISOTOP,7)=MASS(ISOTOP,7)+G(72)*EEXP0(2)+G(73)*EEXP0(7)+G(74)*
354. 1EEXP0(12)+G(75)*EEXP0(17)+G(76)*EEXP0(22)+G(77)*EEXP0(27)+G(78)*EE
355. 2XP0(4)+G(79)*EEXP0(9)+G(80)*EEXP0(14)+G(81)*EEXP0(19)+G(82)*EEXP0(
356. 324)+G(83)*EEXP0(29)
357. 8 IF(JUMP6) GO TO 12
358. C CONSIDER BARRIER 3 03/25/71
359. CALL ZERO(0,76,105)
360. CALL ZERO(P,76,105)
361. EXP0(30)=LAMBDA(ISOTOP)+FOR(3)+OLR(6)
362. EEXP0(30)=EXP(-EXP0(30)*DELTA)
363. EASY=ILR(1,6)*FILTER(ISOTOP,1)
364. DOG=ILR(4,6)+ILR(5,6)*FILTER(ISOTOP,2)

```

```

365. H(71)=(EASY*A(16)+DOG*E(56)+ABLE*H(14)+BAKER*H(23)+CHARLY*H(47))/(
366. 1EPO(30)-EXP0(1))
367. H(72)=(EASY*A(17)+DOG*E(57)+ABLE*H(15)+BAKER*H(23)+CHARLY*H(48))/(
368. 1EPO(30)-EXP0(6))
369. H(73)=(EASY*A(18)+DOG*E(58)+ABLE*H(16)+BAKER*H(30)+CHARLY*H(49))/(
370. 1EPO(30)-EXP0(11))
371. H(74)=(EASY*A(19)+DOG*E(59)+BAKER*H(31)+CHARLY*H(50))/(EXP0(30)-EX
372. P0(16))
373. H(75)=(EASY*A(20)+DOG*E(60)+CHARLY*H(51))/(EXP0(30)-EXP0(21))
374. H(76)=(EASY*A(21)+DOG*E(61))/(EXP0(30)-EXP0(26))
375. H(77)=(DOG*E(62)+ABLE*H(17)+BAKER*H(32)+CHARLY*H(52))/(EXP0(30)-EX
376. P0(7))
377. H(78)=(DOG*E(63)+ABLE*H(18)+BAKER*H(33)+CHARLY*H(53))/(EXP0(30)-EX
378. P0(8))
379. H(79)=(DOG*E(64)+BAKER*H(34)+CHARLY*H(54))/(EXP0(30)-EXP0(13))
380. H(80)=(DOG*E(65)+CHARLY*H(55))/(EXP0(30)-EXP0(18))
381. H(81)=DOG*E(66)/(EXP0(30)-EXP0(23))
382. H(82)=(DOG*E(67)+ABLE*H(19)+BAKER*H(35)+CHARLY*H(56))/(EXP0(30)-EX
383. P0(2))
384. H(83)=(DOG*E(68)+ABLE*H(20)+BAKER*H(36)+CHARLY*H(57))/(EXP0(30)-EX
385. P0(7))
386. H(84)=(DOG*E(69)+ABLE*H(21)+BAKER*H(37)+CHARLY*H(58))/(EXP0(30)-EX
387. P0(12))
388. H(85)=(DOG*E(70)+BAKER*H(38)+CHARLY*H(59))/(EXP0(30)-EXP0(17))
389. H(86)=(DOG*E(71)+CHARLY*H(60))/(EXP0(30)-EXP0(22))
390. H(87)=DOG*E(72)/(EXP0(30)-EXP0(27))
391. H(88)=(DOG*E(73)+ABLE*H(22)+BAKER*H(39)+CHARLY*H(61))/(EXP0(30)-EX
392. P0(4))
393. H(89)=(DOG*E(74)+ABLE*H(23)+BAKER*H(40)+CHARLY*H(62))/(EXP0(30)-EX
394. P0(9))
395. H(90)=(DOG*E(75)+ABLE*H(24)+BAKER*H(41)+CHARLY*H(63))/(EXP0(30)-EX
396. P0(14))
397. H(91)=(DOG*E(76)+BAKER*H(42)+CHARLY*H(64))/(EXP0(30)-EXP0(19))
398. H(92)=(DOG*E(77)+CHARLY*H(65))/(EXP0(30)-EXP0(24))
399. H(93)=DOG*E(78)/(EXP0(30)-EXP0(29))
400. H(94)=(ABLE*H(25)+BAKER*H(43)+CHARLY*H(66))/(EXP0(30)-EXP0(5))
401. H(95)=(ABLE*H(26)+BAKER*H(44)+CHARLY*H(67))/(EXP0(30)-EXP0(10))
402. H(96)=(ABLE*H(27)+BAKER*H(45)+CHARLY*H(68))/(EXP0(30)-EXP0(15))
403. H(97)=(BAKER*H(46)+CHARLY*H(69))/(EXP0(30)-EXP0(20))
404. H(98)=CHARLY*H(70)/(EXP0(30)-EXP0(25))
405. IF(.NOT.JUMPOP.OR.JUMP7) GO TO 9
406. H(71)=H(71)+(ABLE*O(16)+BAKER*O(31)+CHARLY*O(51))/(EXP0(30)-EXP0(1
407. 1))
408. H(72)=H(72)+(ABLE*O(17)+BAKER*O(32)+CHARLY*O(52))/(EXP0(30)-EXP0(6
409. 1))
410. H(73)=H(73)+(ABLE*O(18)+BAKER*O(33)+CHARLY*O(53))/(EXP0(30)-EXP0(1
411. 1))
412. H(74)=H(74)+(BAKER*O(34)+CHARLY*O(54))/(EXP0(30)-EXP0(16))
413. H(75)=H(75)+CHARLY*O(55)/(EXP0(30)-EXP0(21))
414. H(77)=H(77)+(ABLE*O(19)+BAKER*O(35)+CHARLY*O(56))/(EXP0(30)-EXP0(3
415. 1))
416. H(78)=H(78)+(ABLE*O(20)+BAKER*O(36)+CHARLY*O(57))/(EXP0(30)-EXP0(8
417. 1))
418. H(79)=H(79)+(ABLE*O(30)+BAKER*O(37)+CHARLY*O(58))/(EXP0(30)-EXP0(1
419. 3))
420. H(80)=H(80)+(BAKER*O(50)+CHARLY*O(59))/(EXP0(30)-EXP0(18))
421. H(81)=H(81)+CHARLY*O(75)/(EXP0(30)-EXP0(23))
422. H(82)=H(82)+(ABLE*O(21)+BAKER*O(38)+CHARLY*O(60))/(EXP0(30)-EXP0(2
423. 1))
424. H(83)=H(83)+(ABLE*O(22)+BAKER*O(39)+CHARLY*O(61))/(EXP0(30)-EXP0(7
425. 1))
426. H(84)=H(84)+(ABLE*O(23)+BAKER*O(40)+CHARLY*O(62))/(EXP0(30)-EXP0(1
427. 2))
428. H(85)=H(85)+(BAKER*O(41)+CHARLY*O(63))/(EXP0(30)-EXP0(17))
429. H(86)=H(86)+(CHARLY*O(64))/(EXP0(30)-EXP0(22))
430. H(88)=H(88)+(ABLE*O(24)+BAKER*O(42)+CHARLY*O(65))/(EXP0(30)-EXP0(4
431. 1))
432. H(89)=H(89)+(ABLE*O(25)+BAKER*O(43)+CHARLY*O(66))/(EXP0(30)-EXP0(9
433. 1))
434. H(90)=H(90)+(ABLE*O(26)+BAKER*O(44)+CHARLY*O(67))/(EXP0(30)-EXP0(1
435. 4))
436. H(91)=H(91)+(BAKER*O(45)+CHARLY*O(68))/(EXP0(30)-EXP0(19))
437. H(92)=H(92)+CHARLY*O(69)/(EXP0(30)-EXP0(24))
438. H(94)=H(94)+(ABLE*O(27)+BAKER*O(46)+CHARLY*O(70))/(EXP0(30)-EXP0(5
439. 1))
440. H(95)=H(95)+(ABLE*O(28)+BAKER*O(47)+CHARLY*O(71))/(EXP0(30)-EXP0(1
441. 0))
442. H(96)=H(96)+(ABLE*O(29)+BAKER*O(48)+CHARLY*O(72))/(EXP0(30)-EXP0(1
443. 5))
444. H(97)=H(97)+(BAKER*O(49)+CHARLY*O(73))/(EXP0(30)-EXP0(20))
445. H(98)=H(98)+CHARLY*O(74)/(EXP0(30)-EXP0(25))
446. 9 IF(.NOT.JUMPOP.OR.JUMP10) GO TO 10
447. H(71)=H(71)+(ABLE*G(13)+BAKER*G(25)+CHARLY*G(41))/(EXP0(30)-EXP0(1
448. 1))
449. H(72)=H(72)+(ABLE*G(14)+BAKER*G(26)+CHARLY*G(42))/(EXP0(30)-EXP0(6
450. 1))
451. H(73)=H(73)+(ABLE*G(15)+BAKER*G(27)+CHARLY*G(43))/(EXP0(30)-EXP0(1
452. 1))
453. H(74)=H(74)+(BAKER*G(28)+CHARLY*G(44))/(EXP0(30)-EXP0(16))
454. H(75)=H(75)+CHARLY*G(45)/(EXP0(30)-EXP0(21))
455. H(77)=H(77)+(ABLE*G(16)+BAKER*G(23)+CHARLY*G(46))/(EXP0(30)-EXP0(3
456. 1))

```

```

457. H(78)=H(73)+(ABLE*G(17)+BAKER*G(30)+CHARLY*G(47))/(EXP0(30)-EXP0(8
458. 1))
459. H(79)=H(73)+(ABLE*G(24)+BAKER*G(31)+CHARLY*G(48))/(EXP0(30)-EXP0(1
460. 13))
461. H(80)=H(80)+(BAKER*G(40)+CHARLY*G(49))/(EXP0(30)-EXP0(18))
462. H(81)=H(81)+CHARLY*G(60)/(EXP0(30)-EXP0(23))
463. H(82)=H(82)+(ABLE*G(18)+BAKER*G(32)+CHARLY*G(50))/(EXP0(30)-EXP0(2
464. 1))
465. H(83)=H(83)+(ABLE*G(19)+BAKER*G(33)+CHARLY*G(51))/(EXP0(30)-EXP0(7
466. 1))
467. H(84)=H(84)+(ABLE*G(20)+BAKER*G(34)+CHARLY*G(52))/(EXP0(30)-EXP0(1
468. 12))
469. H(85)=H(85)+(BAKER*G(35)+CHARLY*G(53))/(EXP0(30)-EXP0(17))
470. H(86)=H(86)+CHARLY*G(54)/(EXP0(30)-EXP0(22))
471. H(88)=H(88)+(ABLE*G(21)+BAKER*G(36)+CHARLY*G(55))/(EXP0(30)-EXP0(4
472. 1))
473. H(89)=H(89)+(ABLE*G(22)+BAKER*G(37)+CHARLY*G(56))/(EXP0(30)-EXP0(9
474. 1))
475. H(90)=H(90)+(ABLE*G(23)+BAKER*G(38)+CHARLY*G(57))/(EXP0(30)-EXP0(1
476. 14))
477. H(91)=H(91)+(BAKER*G(39)+CHARLY*G(58))/(EXP0(30)-EXP0(19))
478. H(92)=H(92)+CHARLY*G(59)/(EXP0(30)-EXP0(24))
479. 10 H(99)=MASS(ISOTOP,8)-SUMZ(H,71,98)
480. MASS(ISOTOP,8)= H(71)*EEXP0(1)+H(72)*EEXP0(6)+H(73)*EEXP0(11)+
481. 1H(74)*EEXP0(16)+H(75)*EEXP0(21)+H(76)*EEXP0(26)+H(77)*EEXP0(31)+
482. 2H(78)*EEXP0(8)+H(79)*EEXP0(13)+H(80)*EEXP0(18)+H(81)*EEXP0(23)+
483. MASS(ISOTOP,8)=MASS(ISOTOP,8) H(82)*EEXP0(8)+H(83)*EEXP0(7)+
484. 1H(84)*EEXP0(12)+H(85)*EEXP0(17)+H(86)*EEXP0(22)+H(87)*EEXP0(27)+
485. 2H(89)*EEXP0(4)+H(89)*EEXP0(9)+H(90)*EEXP0(14)+H(91)*EEXP0(19)+
486. 3H(92)*EEXP0(24)+H(93)*EEXP0(29)
487. MASS(ISOTOP,8)=MASS(ISOTOP,8) H(94)*EEXP0(5)+H(95)*EEXP0(10)+
488. 1H(96)*EEXP0(15)+H(97)*EEXP0(20)+H(98)*EEXP0(25)+H(99)*EEXP0(30)
489. IF(JUMP7,OR,JUMPUP,OR,JUMPOF) GO TO 11
490. C CONSIDER FALLOUT IN BARRIER 3 03/25/71
491. 0(76)=(FOR(3)*H(71)+ABLE*0(16)+BAKER*0(31)+CHARLY*0(51))/(EXP0(28)
492. 1-EXP0(11))
493. 0(77)=(FOR(3)*H(72)+ABLE*0(17)+BAKER*0(32)+CHARLY*0(52))/(EXP0(28)
494. 1-EXP0(8))
495. 0(78)=(FOR(3)*H(73)+ABLE*0(18)+BAKER*0(33)+CHARLY*0(53))/(EXP0(28)
496. 1-EXP0(11))
497. 0(79)=(FOR(3)*H(74)+BAKER*0(34)+CHARLY*0(54))/(EXP0(28)-EXP0(16))
498. 0(80)=(FOR(3)*H(75)+CHARLY*0(55))/(EXP0(28)-EXP0(21))
499. 0(81)=(FOR(3)*H(76))/(EXP0(28)-EXP0(26))
500. 0(82)=(FOR(3)*H(77)+ABLE*0(19)+BAKER*0(35)+CHARLY*0(56))/(EXP0(28)
501. 1-EXP0(3))
502. 0(83)=(FOR(3)*H(78)+ABLE*0(20)+BAKER*0(36)+CHARLY*0(57))/(EXP0(28)
503. 1-EXP0(8))
504. 0(84)=(FOR(3)*H(79)+ABLE*0(30)+BAKER*0(37)+CHARLY*0(58))/(EXP0(28)
505. 1-EXP0(13))
506. 0(85)=(FOR(3)*H(80)+BAKER*0(50)+CHARLY*0(59))/(EXP0(28)-EXP0(18))
507. 0(86)=(FOR(3)*H(81)+CHARLY*0(75))/(EXP0(28)-EXP0(23))
508. 0(87)=(FOR(3)*H(82)+ABLE*0(21)+BAKER*0(38)+CHARLY*0(60))/(EXP0(28)
509. 1-EXP0(2))
510. 0(88)=(FOR(3)*H(83)+ABLE*0(22)+BAKER*0(39)+CHARLY*0(61))/(EXP0(28)
511. 1-EXP0(7))
512. 0(89)=(FOR(3)*H(84)+ABLE*0(23)+BAKER*0(40)+CHARLY*0(62))/(EXP0(28)
513. 1-EXP0(12))
514. 0(90)=(FOR(3)*H(85)+BAKER*0(41)+CHARLY*0(63))/(EXP0(28)-EXP0(17))
515. 0(91)=(FOR(3)*H(86)+CHARLY*0(64))/(EXP0(28)-EXP0(22))
516. 0(92)=(FOR(3)*H(87))/(EXP0(28)-EXP0(27))
517. 0(93)=(FOR(3)*H(88)+ABLE*0(24)+BAKER*0(42)+CHARLY*0(65))/(EXP0(28)
518. 1-EXP0(4))
519. 0(94)=(FOR(3)*H(89)+ABLE*0(25)+BAKER*0(43)+CHARLY*0(66))/(EXP0(28)
520. 1-EXP0(9))
521. 0(95)=(FOR(3)*H(90)+ABLE*0(26)+BAKER*0(44)+CHARLY*0(67))/(EXP0(28)
522. 1-EXP0(14))
523. 0(96)=(FOR(3)*H(91)+BAKER*0(45)+CHARLY*0(68))/(EXP0(28)-EXP0(19))
524. 0(97)=(FOR(3)*H(92)+CHARLY*0(69))/(EXP0(28)-EXP0(24))
525. 0(98)=(FOR(3)*H(93))/(EXP0(28)-EXP0(29))
526. 0(99)=(FOR(3)*H(94)+ABLE*0(27)+BAKER*0(46)+CHARLY*0(70))/(EXP0(28)
527. 1-EXP0(5))
528. 0(100)=(FOR(3)*H(95)+ABLE*0(28)+BAKER*0(47)+CHARLY*0(71))/(EXP0(28)
529. 1-EXP0(10))
530. 0(101)=(FOR(3)*H(96)+ABLE*0(29)+BAKER*0(48)+CHARLY*0(72))/(EXP0(28)
531. 1-EXP0(15))
532. 0(102)=(FOR(3)*H(97)+BAKER*0(49)+CHARLY*0(73))/(EXP0(28)-EXP0(20))
533. 0(103)=(FOR(3)*H(98)+CHARLY*0(74))/(EXP0(28)-EXP0(25))
534. 0(104)=(FOR(3)*H(99))/(EXP0(28)-EXP0(30))
535. 0(105)=MASS(ISOTOP,9)-SUMZ(0,76,104)
536. MASS(ISOTOP,9)= 0(76)*EEXP0(1)+0(77)*EEXP0(6)+0(78)*EEXP0(11)+
537. 10(79)*EEXP0(16)+0(80)*EEXP0(21)+0(81)*EEXP0(26)+0(82)*EEXP0(31)+
538. 20(83)*EEXP0(8)+0(84)*EEXP0(13)+0(85)*EEXP0(18)+0(86)*EEXP0(23)+
539. MASS(ISOTOP,9)=MASS(ISOTOP,9) 0(87)*EEXP0(2)+0(88)*EEXP0(7)+
540. 10(89)*EEXP0(12)+0(90)*EEXP0(17)+0(91)*EEXP0(22)+0(92)*EEXP0(27)+
541. 20(93)*EEXP0(4)+0(94)*EEXP0(9)+0(95)*EEXP0(14)+0(96)*EEXP0(19)+
542. 30(97)*EEXP0(24)+0(98)*EEXP0(29)
543. MASS(ISOTOP,9)=MASS(ISOTOP,9) 0(99)*EEXP0(5)+0(100)*EEXP0(10)+
544. 10(101)*EEXP0(15)+0(102)*EEXP0(20)+0(103)*EEXP0(25)+0(104)*EEXP0(30)
545. 2)+0(105)*EEXP0(23)
546. 11 IF(JUMPOF,OR,JUMP8) GO TO 12
547. C CONSIDER FILTRATION IN BARRIER 3
548. DOG=ILR(6,7)*(1.-FILTER(ISOTOP,2))

```



```

549. P(76)=(D0G*H(71)+ABLE*P(16)+BAKER*P(31)+CHARLY*P(51))/(EXP0(28)-EX
550. 1P(1))
551. P(77)=(D0G*H(72)+ABLE*P(17)+BAKER*P(32)+CHARLY*P(52))/(EXP0(28)-EX
552. 1P(6))
553. P(78)=(D0G*H(73)+ABLE*P(18)+BAKER*P(33)+CHARLY*P(53))/(EXP0(28)-EX
554. 1P(11))
555. P(79)=(D0G*H(74)+BAKER*P(34)+CHARLY*P(54))/(EXP0(28)-EXP0(16))
556. P(80)=(D0G*H(75)+CHARLY*P(55))/(EXP0(28)-EXP0(21))
557. P(81)=(D0G*H(76))/(EXP0(28)-EXP0(26))
558. P(82)=(D0G*H(77)+ABLE*P(19)+BAKER*P(35)+CHARLY*P(56))/(EXP0(28)-EX
559. 1P(3))
560. P(83)=(D0G*H(78)+ABLE*P(20)+BAKER*P(36)+CHARLY*P(57))/(EXP0(28)-EX
561. 1P(8))
562. P(84)=(D0G*H(79)+ABLE*P(30)+BAKER*P(37)+CHARLY*P(58))/(EXP0(28)-EX
563. 1P(13))
564. P(85)=(D0G*H(80)+BAKER*P(50)+CHARLY*P(59))/(EXP0(28)-EXP0(18))
565. P(86)=(D0G*H(81)+CHARLY*P(75))/(EXP0(28)-EXP0(23))
566. P(87)=(D0G*H(82)+ABLE*P(21)+BAKER*P(38)+CHARLY*P(60))/(EXP0(28)-EX
567. 1P(2))
568. P(88)=(D0G*H(83)+ABLE*P(22)+BAKER*P(39)+CHARLY*P(61))/(EXP0(28)-EX
569. 1P(7))
570. P(89)=(D0G*H(84)+ABLE*P(23)+BAKER*P(40)+CHARLY*P(62))/(EXP0(28)-EX
571. 1P(12))
572. P(90)=(D0G*H(85)+BAKER*P(41)+CHARLY*P(63))/(EXP0(28)-EXP0(17))
573. P(91)=(D0G*H(86)+CHARLY*P(64))/(EXP0(28)-EXP0(22))
574. P(92)=(D0G*H(87))/(EXP0(28)-EXP0(27))
575. P(93)=(D0G*H(88)+ABLE*P(24)+BAKER*P(42)+CHARLY*P(65))/(EXP0(28)-EX
576. 1P(4))
577. P(94)=(D0G*H(89)+ABLE*P(25)+BAKER*P(43)+CHARLY*P(66))/(EXP0(28)-EX
578. 1P(9))
579. P(95)=(D0G*H(90)+ABLE*P(26)+BAKER*P(44)+CHARLY*P(67))/(EXP0(28)-EX
580. 1P(14))
581. P(96)=(D0G*H(91)+BAKER*P(45)+CHARLY*P(68))/(EXP0(28)-EXP0(19))
582. P(97)=(D0G*H(92)+CHARLY*P(69))/(EXP0(28)-EXP0(24))
583. P(98)=(D0G*H(93))/(EXP0(28)-EXP0(29))
584. P(99)=(D0G*H(94)+ABLE*H(27)+BAKER*P(46)+CHARLY*P(70))/(EXP0(28)-EX
585. 1P(5))
586. P(100)=(D0G*H(95)+ABLE*H(28)+BAKER*P(47)+CHARLY*P(71))/(EXP0(28)-E
587. 1XP(10))
588. P(101)=(D0G*H(96)+ABLE*H(29)+BAKER*P(48)+CHARLY*P(72))/(EXP0(28)-E
589. 1XP(15))
590. P(102)=(D0G*H(97)+BAKER*P(49)+CHARLY*P(73))/(EXP0(28)-EXP0(20))
591. P(103)=(D0G*H(98)+CHARLY*P(74))/(EXP0(28)-EXP0(25))
592. P(104)=(D0G*H(99))/(EXP0(28)-EXP0(30))
593. P(105)=MASS(ISOTOP,10)-SUM Z(P,76,104)
594. MASS(ISOTOP,10)= P(76)*EEXP0( 1)+P(77)*EEXP0( 6)+P(78)*EEXP0(11)+
595. 1P(79)*EEXP0(16) +P(80)*EEXP0(21)+P(81)*EEXP0(26)+P(82)*EEXP0( 3)+
596. 2P(83)*EEXP0( 8) +P(84)*EEXP0(13)+P(85)*EEXP0(18)+P(86)*EEXP0(23)+
597. 3P(105)*EEXP0(28)
598. MASS(ISOTOP,10)=MASS(ISOTOP,10) +P(87)*EEXP0( 2)+P(88)*EEXP0( 7)+
599. 1 P(89)*EEXP0(12)+P(90)*EEXP0(17)+P(91)*EEXP0(22)+P(92)*EEXP0(27)+
600. 2 P(93)*EEXP0( 4)+P(94)*EEXP0( 9)+P(95)*EEXP0(14)+P(96)*EEXP0(19)+
601. 3 P(97)*EEXP0(24)+P(98)*EEXP0(29)
602. MASS(ISOTOP,10)=MASS(ISOTOP,10)+P(99) *EEXP0(5)+P(100)*EEXP0(10)+
603. 1P(101)*EEXP0(15)+P(102)*EEXP0(20)+P(103)*EEXP0(25)+P(105)*EEXP0(30
604. 2)
605. C CALCULATE RELEASE RATE TO ATMOSPHERE 03/25/71
606. 12 QUANTY(ISOTOP)=ALR(1)*FILTER(ISOTOP,1)*MASS(ISOTOP,1)+(ALR(2)+ALR(
607. 13)*FILTER(ISOTOP,2))*MASS(ISOTOP,2)+(ALR(4)+ALR(5))*FILTER(ISOTOP,2
608. 2))*MASS(ISOTOP,5)+(ALR(6)+ALR(7))*FILTER(ISOTOP,2))*MASS(ISOTOP,8)
609. IF (.NOT. JUMP) GO TO 13
610. QUANTY(ISOTOP)=QUANTY(ISOTOP)+ALR(3)/ALR(3)*(ABLE*MASS(ISOTOP-3,4)
611. 1+BAKER*MASS(ISOTOP-2,4)+CHARLY*MASS(ISOTOP-1,4))+ALR(5)/ALR(5)*
612. 2ABLE*MASS(ISOTOP-3,7)+BAKER*MASS(ISOTOP-2,7)+CHARLY*MASS(ISOTOP-1,
613. 37))+ALR(7)/ALR(7)*(ABLE*MASS(ISOTOP-3,10)+BAKER*MASS(ISOTOP-2,10)
614. 4+CHARLY*MASS(ISOTOP-1,10))
615. C RETURN TO BAFFLE 03/25/71
616. 13 RETURN
617. END

```

```

1. C SUBROUTINE LAST DOES CONTAINMENT CALCULATION FOR SEVENTH CHAIN MEMB03/25/71
2. SUBROUTINE LAST 03/25/71
3. COMMON ISOTOP,SPACE(97),DELTA1,MASS(600,10),HOLE(2850),DKFRCT(450,
4. A3),FILTER(600,2),LAMBDA(600),SPOT(96),H(133),A(23),O(105),P(105),
5. 26(84),F(84),E(105),D(63),C(63),B(77),FOR(3),ALR(7),ILR(7,8),OLR(7)
6. 3,QUANTY(600),EXPC(35),EEXPO(35),SKIP(1200),JUMP1,JUMP4,JUMP2,JUMP3
7. 4,JUMP5,JUMP6,JUMP8,JUMP7,JUMP9,JUMP10,JUMPOF,JUMPOP
8. REAL MASS,ILR,LAMBDA
9. LOGICAL JUMP1,JUMP2,JUMP3,JUMP4,JUMP5,JUMP6,JUMP7,JUMP8
10. LOGICAL JUMP9,JUMP10,JUMPOF,JUMPOP
11. C 03/25/71
12. C A REPRESENTS COEFFICIENTS FOR FUEL, B FOR 1ST BARRIER, C FOR 1ST FALL03/25/71
13. C D FOR 1ST FILTER, E FOR 2ND BARRIER, F FOR 2ND FALLOUT, G FOR 2ND FILOS03/25/71
14. C H FOR 3RD BARRIER, O FOR 3RD FALLOUT, P FOR 3RD FILTER. 03/25/71
15. C 03/25/71
16. EXPO(31)=LAMBDA(ISOTOP)+OLR(1)*FILTER(ISOTOP,1)
17. EEXPO(31)=EXP(-EXPO(31)*DELTA1)
18. ABLE=LAMBDA(ISOTOP-3)*DKFRCT(ISOTOP,3)
19. BAKER=LAMBDA(ISOTOP-2)*DKFRCT(ISOTOP,2)
20. CHARLY=LAMBDA(ISOTOP-1)*DKFRCT(ISOTOP,1)
21. DO 100 I=1,4
22. 100 A(I+21)=(ABLE*A(I+6)+BAKER*A(I+10)+CHARLY*A(I+15))/(EXPO(31)-EXPO(
23. 15*I-4))
24. A(26)=(BAKER*A(15)+CHARLY*A(20))/(EXPO(31)-EXPO(21))
25. A(27)=CHARLY*A(21)/(EXPO(31)-EXPO(26))
26. A(28)=MASS(ISOTOP,1)-SUMZ(A,22,27)
27. MASS(ISOTOP,1)=A(22)*EEXPO(1)+A(23)*EEXPO(6)+A(24)*EEXPO(11)+A(25)
28. 1+EEXPO(16)+A(26)*EEXPO(21)+A(27)*EEXPO(26)+A(28)*EEXPO(31)
29. C CONSIDER BARRIER 1 03/25/71
30. EXPO(32)=LAMBDA(ISOTOP)+FOR(1)+OLR(2)
31. EEXPO(32)=EXP(-EXPC(32)*DELTA1)
32. DOG=ILR(1,2)*FILTER(ISOTOP,1)
33. DO 101 I=1,4
34. 101 B(I+57)=(DOG*A(I+21)+ABLE*B(I+15)+BAKER*B(I+26)+CHARLY*B(I+40))/(E
35. 1XPO(32)-EXPO(5*I-4))
36. B(62)=(DOG*A(26)+BAKER*B(31)+CHARLY*B(45))/(EXPO(32)-EXPO(21))
37. B(63)=(DOG*A(27)+CHARLY*B(46))/(EXPO(32)-EXPO(26))
38. B(64)=DOG*A(28)/(EXPO(32)-EXPO(31))
39. DO 102 I=1,3
40. 102 B(I+64)=(ABLE*B(I+19)+BAKER*B(I+31)+CHARLY*B(I+46))/(EXPO(32)-EXPO
41. 1(5*I-2))
42. B(68)=(BAKER*B(35)+CHARLY*B(50))/(EXPO(32)-EXPO(16))
43. B(69)=CHARLY*B(51)/(EXPO(32)-EXPO(23))
44. DO 103 I=1,4
45. 103 B(I+70)=(ABLE*B(I+22)+BAKER*B(I+35)+CHARLY*B(I+51))/(EXPO(32)-EXPO
46. 1(5*I-3))
47. B(75)=(BAKER*B(40)+CHARLY*B(56))/(EXPO(32)-EXPO(22))
48. B(76)=CHARLY*B(57)/(EXPO(32)-EXPO(27))
49. IF(.NOT.JUMPOP.OR.JUMP1) GO TO 1
50. DO 104 I=1,4
51. 104 B(I+57)=B(I+57)+(ABLE*C(I+18)+BAKER*C(I+30)+CHARLY*C(I+45))/(EXPO(
52. 132)-EXPO(5*I-4))
53. B(62)=B(62)+(BAKER*C(35)+CHARLY*C(50))/(EXPO(32)-EXPO(21))
54. B(63)=B(63)+CHARLY*C(51)/(EXPO(32)-EXPO(26))
55. DO 105 I=1,3
56. 105 B(I+64)=B(I+64)+(ABLE*C(I+22)+BAKER*C(I+35)+CHARLY*C(I+51))/(EXPO(
57. 132)-EXPO(5*I-2))
58. B(68)=B(68)+(ABLE*C(30)+BAKER*C(39)+CHARLY*C(55))/(EXPO(32)-EXPO(1
59. 18))
60. B(69)=B(69)+(BAKER*C(45)+CHARLY*C(56))/(EXPO(32)-EXPO(23))
61. B(70)=B(70)+CHARLY*C(63)/(EXPO(32)-EXPO(28))
62. DO 106 I=1,4
63. 106 B(I+70)=B(I+70)+(ABLE*C(I+25)+BAKER*C(I+39)+CHARLY*C(I+56))/(EXPO(
64. 132)-EXPO(5*I-3))
65. B(75)=B(75)+(BAKER*C(44)+CHARLY*C(61))/(EXPO(32)-EXPO(22))
66. B(76)=B(76)+CHARLY*C(62)/(EXPO(32)-EXPO(27))
67. GO TO 2
68. 1 B(70)=0.0
69. 2 B(77)=MASS(ISOTOP,2)-SUMZ(B,58,76)
70. MASS(ISOTOP,2)=B(58)*EEXPO(1)+B(59)*EEXPO(6)+B(60)*EEXPO(11)+B(61)
71. 1+EEXPO(16)+B(62)*EEXPO(21)+B(63)*EEXPO(26)+B(64)*EEXPO(31)+B(65)*E
72. 2EXPO(3)+B(66)*EEXPO(8)+B(67)*EEXPO(13)+B(68)*EEXPO(18)+B(69)*EEXPO
73. 3(23)+B(70)*EEXPO(28)
74. MASS(ISOTOP,2)=MASS(ISOTOP,2)+B(71)*EEXPO(2)+B(72)*EEXPO(7)+B(73)*
75. 1EEXPO(12)+B(74)*EEXPO(17)+B(75)*EEXPO(22)+B(76)*EEXPO(27)+B(77)*E
76. 2XPO(32)
77. IF(JUMP3) GO TO 5
78. C CONSIDER BARRIER 2 03/25/71
79. EXPO(34)=LAMBDA(ISOTOP)+FOR(2)+OLR(4)
80. EEXPO(34)=EXP(-EXPC(34)*DELTA1)
81. EASY=ILR(1,4)*FILTER(ISOTOP,1)
82. DOG=ILR(2,4)+ILR(3,4)*FILTER(ISOTOP,2)
83. DO 107 I=1,4
84. 107 E(I+76)=(EASY*A(I+21)+DOG*B(I+57)+ABLE*E(I+11)+BAKER*E(I+36)+CHARL
85. 1Y*E(I+55))/(EXPO(34)-EXPO(5*I-4))
86. E(83)=(EASY*A(26)+DOG*B(62)+BAKER*E(41)+CHARLY*E(60))/(EXPO(34)-EX
87. 1PC(21))
88. E(84)=(EASY*A(27)+DOG*B(63)+CHARLY*E(61))/(EXPO(34)-EXPO(26))

```

```

89.      E(85)=(EASY*A(28)+DOG*B(64))/(EXPO(34)-EXPO(31))
90.      DO 103 I=1,3
91. 108 E(I+85)=(DOG*B(I+64)+ABLE*E(I+25)+BAKER*E(I+41)+CHARLY*E(I+61))/(E
92.      1XPO(34)-EXPO(5*I-2))
93.      E(89)=(DOG*B(68)+BAKER*E(45)+CHARLY*E(65))/(EXPO(34)-EXPO(18))
94.      E(90)=(DOG*B(69)+CHARLY*E(66))/(EXPO(34)-EXPO(23))
95.      E(91)=DOG*B(70)/(EXPO(34)-EXPO(28))
96.      DO 109 I=1,4
97. 109 E(I+91)=(DOG*B(I+70)+ABLE*E(I+28)+BAKER*E(I+45)+CHARLY*E(I+66))/(E
98.      1XPO(34)-EXPO(5*I-3))
99.      E(96)=(DOG*B(75)+BAKER*E(50)+CHARLY*E(71))/(EXPO(34)-EXPO(22))
100.      E(97)=(DOG*B(76)+CHARLY*E(72))/(EXPO(34)-EXPO(27))
101.      E(98)=DOG*B(77)/(EXPO(34)-EXPO(32))
102.      DO 110 I=1,4
103. 110 E(I+98)=(ABLE*E(I+32)+BAKER*E(I+50)+CHARLY*E(I+72))/(EXPO(34)-EXPO
104.      1(5*I-1))
105.      E(103)=(BAKER*E(55)+CHARLY*E(77))/(EXPO(34)-EXPO(24))
106.      E(104)=CHARLY*E(78)/(EXPO(34)-EXPO(29))
107.      IF(.NOT.JUMPUP.OR.JUMP4) GO TO 3
108.      DO 111 I=1,4
109. 111 E(I+78)=E(I+78)+(ABLE*F(I+24)+BAKER*F(I+40)+CHARLY*F(I+60))/(EXPO(
110.      134)-EXPO(5*I-4))
111.      E(83)=E(83)+(BAKER*F(45)+CHARLY*F(65))/(EXPO(34)-EXPO(21))
112.      E(84)=E(84)+CHARLY*F(66)/(EXPO(34)-EXPO(26))
113.      DO 112 I=1,3
114. 112 E(I+85)=E(I+85)+(ABLE*F(I+28)+BAKER*E(I+45)+CHARLY*E(I+66))/(EXPO(
115.      134)-EXPO(5*I-2))
116.      E(89)=E(89)+(ABLE*F(40)+BAKER*F(49)+CHARLY*F(70))/(EXPO(34)-EXPO(1
117.      18))
118.      E(90)=E(90)+(BAKER*F(60)+CHARLY*F(71))/(EXPO(34)-EXPO(23))
119.      E(91)=E(91)+CHARLY*F(84)/(EXPO(34)-EXPO(28))
120.      DO 113 I=1,4
121. 113 E(I+91)=E(I+91)+(ABLE*F(I+31)+BAKER*F(I+49)+CHARLY*F(I+71))/(EXPO(
122.      134)-EXPO(5*I-3))
123.      E(96)=E(96)+(BAKER*F(54)+CHARLY*F(76))/(EXPO(34)-EXPO(22))
124.      E(97)=E(97)+CHARLY*F(77)/(EXPO(34)-EXPO(27))
125.      DO 114 I=1,4
126. 114 E(I+98)=E(I+98)+(ABLE*F(I+35)+BAKER*F(I+54)+CHARLY*F(I+77))/(EXPO(
127.      134)-EXPO(5*I-1))
128.      E(103)=E(103)+(BAKER*F(59)+CHARLY*F(82))/(EXPO(34)-EXPO(24))
129.      E(104)=E(104)+CHARLY*F(83)/(EXPO(34)-EXPO(29))
130.      3 IF(.NOT.JUMPOF.OR.JUMP9) GO TO 4
131.      DO 115 I=1,4
132. 115 E(I+78)=E(I+78)+(ABLE*D(I+13)+BAKER*D(I+30)+CHARLY*D(I+45))/(EXPO(
133.      134)-EXPO(5*I-4))
134.      E(83)=E(83)+(BAKER*D(35)+CHARLY*D(50))/(EXPO(34)-EXPO(21))
135.      E(84)=E(84)+CHARLY*D(51)/(EXPO(34)-EXPO(26))
136.      DO 116 I=1,3
137. 116 E(I+85)=E(I+85)+(ABLE*F(I+22)+BAKER*F(I+35)+CHARLY*F(I+51))/(EXPO(
138.      134)-EXPO(5*I-2))
139.      E(89)=E(89)+(ABLE*F(30)+BAKER*F(39)+CHARLY*F(55))/(EXPO(34)-EXPO(1
140.      18))
141.      E(90)=E(90)+(BAKER*F(45)+CHARLY*F(56))/(EXPO(34)-EXPO(23))
142.      E(91)=E(91)+CHARLY*F(63)/(EXPO(34)-EXPO(28))
143.      DO 117 I=1,4
144. 117 E(I+91)=E(I+91)+(ABLE*F(I+25)+BAKER*F(I+39)+CHARLY*F(I+56))/(EXPO(
145.      134)-EXPO(5*I-3))
146.      E(96)=E(96)+(BAKER*F(44)+CHARLY*F(61))/(EXPO(34)-EXPO(22))
147.      E(97)=E(97)+CHARLY*F(62)/(EXPO(34)-EXPO(27))
148.      4 E(105)=MASS(ISOTOP,5)-SUMZ(E,79,104)
149.      MASS(ISOTOP,5)=E(79)*EEXPO(1)+E(80)*EEXPO(6)+E(81)*EEXPO(11)+E(82)
150.      1*EEXPO(16)+E(83)*EEXPO(21)+E(84)*EEXPO(26)+E(85)*EEXPO(31)+E(86)*E
151.      2EXPO(3)+E(87)*EEXPO(8)+E(88)*EEXPO(13)+E(89)*EEXPO(18)+E(90)*EEXPO
152.      3(23)+E(91)*EEXPO(28)
153.      MASS(ISOTOP,5)=MASS(ISOTOP,5)+E(92)*EEXPO(2)+E(93)*EEXPO(7)+E(94)*
154.      1EEXPO(12)+E(95)*EEXPO(17)+E(96)*EEXPO(22)+E(97)*EEXPO(27)+E(98)*E
155.      2XPO(32)+E(99)*EEXPO(4)+E(100)*EEXPO(9)+E(101)*EEXPO(14)+E(102)*EEX
156.      3PO(19)+E(103)*EEXPO(24)+E(104)*EEXPO(29)+E(105)*EEXPO(34)
157.      5 IF(JUMP6) GO TO 8
158. C CONSIDER BARRIER 3
159.      EXPO(35)=LAMRCA(ISOTOP)+FOR(3)+OLR(6)
160.      EEXPO(35)=EXP(-EXPO(35)*DELTA)
161.      EASY=ILR(1,6)*FILTER(ISOTOP,1)
162.      DOG=ILR(4,6)+ILR(5,6)*FILTER(ISOTOP,2)
163.      DO 118 I=1,4
164. 118 H(I+99)=(EASY*A(I+21)+DOG*E(I+78)+ABLE*H(I+27)+BAKER*H(I+46)+CHARL
165.      1Y*H(I+70))/(EXPO(35)-EXPO(5*I-4))
166.      H(I+112)=(DOG*E(I+91)+ABLE*H(I+34)+BAKER*H(I+55)+CHARLY*H(I+81))/(
167.      1EXPO(35)-EXPO(5*I-3))
168.      H(I+119)=(DOG*E(I+98)+ABLE*H(I+38)+BAKER*H(I+60)+CHARLY*H(I+87))/(
169.      1EXPO(35)-EXPO(5*I-1))
170. 118 H(I+126)=(ABLE*H(I+42)+BAKER*H(I+65)+CHARLY*H(I+93))/(EXPO(35)-EXP
171.      10(5*I))
172.      DO 119 I=1,3
173. 119 H(I+106)=(DOG*E(I+85)+ABLE*H(I+31)+BAKER*H(I+51)+CHARLY*H(I+76))/(
174.      1EXPO(35)-EXPO(5*I-2))
175.      H(104)=(EASY*A(26)+DOG*E(83)+BAKER*H(51)+CHARLY*H(75))/(EXPO(35)-E
176.      1XPO(21))
177.      H(105)=(EASY*A(27)+DOG*E(84)+CHARLY*H(76))/(EXPO(35)-EXPO(26))
178.      H(106)=(EASY*A(28)+DOG*E(85))/(EXPO(35)-EXPO(31))
179.      H(110)=(DOG*E(89)+BAKER*H(55)+CHARLY*H(80))/(EXPO(35)-EXPO(18))
180.      H(111)=(DOG*E(90)+CHARLY*H(81))/(EXPO(35)-EXPO(23))

```

```

181. H(112)=.006*(91)/(EXP0(35)-EXP0(23))
182. H(117)=(.006*(96)+BAKER*H(60)+CHARLY*H(86))/(EXP0(35)-EXP0(22))
183. H(114)=(.006*(97)+CHARLY*H(87))/(EXP0(35)-EXP0(27))
184. H(119)=.006*(98)/(EXP0(35)-EXP0(32))
185. H(124)=(.006*(103)+BAKER*H(65)+CHARLY*H(92))/(EXP0(35)-EXP0(24))
186. H(125)=(.006*(104)+CHARLY*H(93))/(EXP0(35)-EXP0(29))
187. H(126)=.006*(105)/(EXP0(35)-EXP0(34))
188. H(131)=(BAKER*H(70)+CHARLY*H(98))/(EXP0(35)-EXP0(25))
189. H(132)=CHARLY*H(99)/(EXP0(35)-EXP0(30))
190. IF(.NOT.JUMPUP.OK.JUMP7) GO TO 6
191. DO 120 I=1,4
192. H(I+99)=H(I+99)+(ABLE*(I+30)+BAKER*(I+50)+CHARLY*(I+75))/(EXP0(
193. 135)-EXP0(5*I-4))
194. H(I+112)=H(I+112)+(ABLE*(I+37)+BAKER*(I+59)+CHARLY*(I+86))/(EXP
195. 10(35)-EXP0(5*I-3))
196. H(I+119)=H(I+119)+(ABLE*(I+41)+BAKER*(I+64)+CHARLY*(I+92))/(EXP
197. 10(35)-EXP0(5*I-1))
198. 120 H(I+126)=H(I+126)+(ABLE*(I+45)+BAKER*(I+69)+CHARLY*(I+97))/(EXP
199. 10(35)-EXP0(5*I))
200. DO 121 I=1,3
201. 121 H(I+106)=H(I+106)+(ABLE*(I+34)+BAKER*(I+55)+CHARLY*(I+81))/(EXP
202. 10(35)-EXP0(5*I-2))
203. H(110)=H(110)+(ABLE*(50)+BAKER*(59)+CHARLY*(85))/(EXP0(35)-EXP0
204. 1(18))
205. H(111)=H(111)+(BAKER*(75)+CHARLY*(86))/(EXP0(35)-EXP0(23))
206. H(112)=H(112)+CHARLY*(105)/(EXP0(35)-EXP0(28))
207. H(104)=H(104)+(BAKER*(55)+CHARLY*(80))/(EXP0(35)-EXP0(21))
208. H(117)=H(117)+(BAKER*(64)+CHARLY*(91))/(EXP0(35)-EXP0(22))
209. H(124)=H(124)+(BAKER*(69)+CHARLY*(97))/(EXP0(35)-EXP0(24))
210. H(131)=H(131)+(BAKER*(74)+CHARLY*(103))/(EXP0(35)-EXP0(25))
211. H(105)=H(105)+CHARLY*(81)/(EXP0(35)-EXP0(26))
212. H(115)=H(115)+CHARLY*(92)/(EXP0(35)-EXP0(27))
213. H(125)=H(125)+CHARLY*(98)/(EXP0(35)-EXP0(29))
214. H(132)=H(132)+CHARLY*(104)/(EXP0(35)-EXP0(30))
215. 6 IF(.NOT.JUMPOF.OK.JUMP10) GO TO 7
216. DO 122 I=1,4
217. H(I+99)=H(I+99)+(ABLE*(I+24)+BAKER*(I+40)+CHARLY*(I+60))/(EXP0(
218. 135)-EXP0(5*I-4))
219. H(I+112)=H(I+112)+(ABLE*(I+31)+BAKER*(I+49)+CHARLY*(I+71))/(EXP
220. 10(35)-EXP0(5*I-3))
221. 122 H(I+119)=H(I+119)+(ABLE*(I+35)+BAKER*(I+54)+CHARLY*(I+77))/(EXP
222. 10(35)-EXP0(5*I-1))
223. DO 123 I=1,3
224. 123 H(I+106)=H(I+106)+(ABLE*(I+28)+BAKER*(I+45)+CHARLY*(I+66))/(EXP
225. 10(35)-EXP0(5*I-2))
226. H(110)=H(110)+(ABLE*(40)+BAKER*(49)+CHARLY*(70))/(EXP0(35)-EXP0
227. 1(18))
228. H(104)=H(104)+(BAKER*(45)+CHARLY*(65))/(EXP0(35)-EXP0(21))
229. H(111)=H(111)+(BAKER*(60)+CHARLY*(71))/(EXP0(35)-EXP0(23))
230. H(117)=H(117)+(BAKER*(54)+CHARLY*(76))/(EXP0(35)-EXP0(22))
231. H(124)=H(124)+(BAKER*(59)+CHARLY*(82))/(EXP0(35)-EXP0(24))
232. H(105)=H(105)+CHARLY*(66)/(EXP0(35)-EXP0(26))
233. H(112)=H(112)+CHARLY*(84)/(EXP0(35)-EXP0(28))
234. H(115)=H(115)+CHARLY*(77)/(EXP0(35)-EXP0(27))
235. H(125)=H(125)+CHARLY*(83)/(EXP0(35)-EXP0(29))
236. 7 H(133)=MASS(ISOTOP,8)-SUM2(H,100,132)
237. MASS(ISOTOP,8)=H(100)*EEXP0(1)+H(101)*EEXP0(6)+H(102)*EEXP0(11)+H(
238. 1103)*EEXP0(16)+H(104)*EEXP0(21)+H(105)*EEXP0(26)+H(106)*EEXP0(31)+
239. 2H(107)*EEXP0(3)+H(108)*EEXP0(8)+H(109)*EEXP0(13)+H(110)*EEXP0(18)+
240. 3H(111)*EEXP0(23)+H(112)*EEXP0(28)
241. MASS(ISOTOP,8)=MASS(ISOTOP,8)+H(113)*EEXP0(2)+H(114)*EEXP0(7)+H(11
242. 15)*EEXP0(12)+H(116)*EEXP0(17)+H(117)*EEXP0(22)+H(118)*EEXP0(27)+H(
243. 2119)*EEXP0(32)+H(120)*EEXP0(4)+H(121)*EEXP0(9)+H(122)*EEXP0(14)+H(
244. 3123)*EEXP0(19)+H(124)*EEXP0(24)+H(125)*EEXP0(29)+H(126)*EEXP0(34)
245. MASS(ISOTOP,8)=MASS(ISOTOP,8)+H(127)*EEXP0(5)+H(128)*EEXP0(10)+H(1
246. 129)*EEXP0(15)+H(130)*EEXP0(20)+H(131)*EEXP0(25)+H(132)*EEXP0(30)+H
247. 2(133)*EEXP0(35)
248. C CALCULATE RELEASE RATE TO ATMOSPHERE 03/25/71
249. 8 QUANTY(ISOTOP)=ALR(1)*FILTER(ISOTOP,1)*MASS(ISOTOP,1)+(ALR(2)+ALR(
250. 13)*FILTER(ISOTOP,2))*MASS(ISOTOP,2)+(ALR(4)+ALR(5)*FILTER(ISOTOP,2
251. 2))*MASS(ISOTOP,5)+(ALR(6)+ALR(7)*FILTER(ISOTOP,2))*MASS(ISOTOP,8)
252. IF(.NOT.JUMPOF) GO TO 9
253. QUANTY(ISOTOP)=QUANTY(ISOTOP)+ALR(3)/ALR(3)*(ABLE*MASS(ISOTOP-3,4)
254. 1+BAKER*MASS(ISOTOP-2,4)+CHARLY*MASS(ISOTOP-1,4))+ALR(5)/ALR(5)*(
255. 2ABLE*MASS(ISOTOP-3,7)+BAKER*MASS(ISOTOP-2,7)+CHARLY*MASS(ISOTOP-1,
256. 37))+ALR(7)/ALR(7)*(ABLE*MASS(ISOTOP-3,10)+BAKER*MASS(ISOTOP-2,10)
257. 4+CHARLY*MASS(ISOTOP-1,10))
258. C RETURN TO BAFFLE 03/25/71
259. 9 RETURN
260. END

```

9 ELT ZERO,1,710325, 30841

```
1. C ZERO(A,I,J) SETS ELEMENT I THRU J OF ARRAY A TO ZERO
2. C
3. SUBROUTINE ZERO(A,I,J)
4. DIMENSION A(2)
5. DO 1 K=I,J
6. 1 A(K)=0.0
7. RETURN
8. END
```

9 ELT SUMZ,1,710325, 30840

```
1. C SUMZ(A,I,J) SUMS THE I-TH THRU J-TH ELEMENTS OF ARRAY A
2. C
3. FUNCTION SUMZ(A,I,J)
4. DIMENSION A(2)
5. SUMZ=0.0
6. DO 1 K=I,J
7. 1 SUMZ=SUMZ+A(K)
8. RETURN
9. END
```

```

1.  C SUBROUTINE CHAIN1 DECAYS NUCLIDES BY CHAIN DECAY FOR TIME DELTAT USING
2.  C EFFECTIVE DECAY CONSTANTS FOR SINGLE CONTAINMENT MODEL
3.  SUBROUTINE CHAIN1
4.  COMMON ISOTOP,CHAINS,NCFNUC(96),DELTAT,MASS(600,10),SPACE(2050),
5.  DKFRCT(450,3),SPJCT(1200),LAMBDA(600),SKIP(96),HOL(133),A(20,2),EXP
6.  20(8)
7.  DIMENSION LOSS(600,2)
8.  EQUIVALENCE (MASS(1,2),LOSS)
9.  REAL MASS,LOSS,LAMBDA
10.  INTEGER SKIP,CHAINS
11.  ISOTOP=0
12.  C DO LOOP ON NUMBER OF CHAINS
13.  DO 11 K=1,CHAINS
14.  LIM=NCFNUC(K)
15.  IF (SKIP(K).EQ.0) GO TO 10
16.  C SKIP(K) IS THE FIRST NON-ZERO MEMBER OF THE CHAIN
17.  I1=SKIP(K)
18.  ISOTOP=ISOTOP+I1-1
19.  J=0
20.  C DO LOOP ON ISOTOPES IN THE CHAIN
21.  DO 9 I=I1,LIM
22.  ISOTOP=ISOTOP+1
23.  J=J+1
24.  IF (LAMBDA(ISOTOP).LT.1.E-30) GO TO 8
25.  EXPO(J)=EXP(-LOSS(ISOTOP,1)*DELTAT)
26.  C TRANSFER ON CHAIN MEMBER J
27.  GO TO (1,2,3,4,5,6,7),J
28.  C FIRST CHAIN MEMBER
29.  1 A(1,1)=MASS(ISOTOP,1)
30.  MASS(ISOTOP,1)=A(1,1)*EXPO(1)
31.  GO TO 9
32.  C SECOND CHAIN MEMBER
33.  2 A(2,1)=DKFRCT(ISOTOP,1)*LAMBDA(ISOTOP-1)*A(1,1)/(LOSS(ISOTOP,1)-LO
34.  1SS(ISOTOP-1,1))
35.  A(3,1)=MASS(ISOTOP,1)-A(2,1)
36.  MASS(ISOTOP,1)=A(3,1)*EXPO(2)+A(2,1)*EXPO(1)
37.  GO TO 9
38.  C THIRD CHAIN MEMBER
39.  3 ABLE=DKFRCT(ISOTOP,1)*LAMBDA(ISOTOP-1)
40.  A(4,1)=(ABLE*A(2,1)+DKFRCT(ISOTOP,2)*LAMBDA(ISOTOP-2)*A(1,1))/(LO
41.  1SS(ISOTOP,1)-LOSS(ISOTOP-2,1))
42.  A(5,1)=ABLE*A(3,1)/(LOSS(ISOTOP,1)-LOSS(ISOTOP-1,1))
43.  A(6,1)=MASS(ISOTOP,1)-A(5,1)-A(4,1)
44.  MASS(ISOTOP,1)=A(6,1)*EXPO(3)+A(5,1)*EXPO(2)+A(4,1)*EXPO(1)
45.  GO TO 9
46.  C FOURTH CHAIN MEMBER
47.  4 ABLE=DKFRCT(ISOTOP,1)*LAMBDA(ISOTOP-1)
48.  BAKER=DKFRCT(ISOTOP,2)*LAMBDA(ISOTOP-2)
49.  A(7,1)=(ABLE*A(4,1)+BAKER*A(2,1)+DKFRCT(ISOTOP,3)*LAMBDA(ISOTOP-3)
50.  1*A(1,1))/(LOSS(ISOTOP,1)-LOSS(ISOTOP-3,1))
51.  A(8,1)=(ABLE*A(5,1)+BAKER*A(3,1))/(LOSS(ISOTOP,1)-LOSS(ISOTOP-2,1)
52.  1)
53.  A(9,1)=ABLE*A(6,1)/(LOSS(ISOTOP,1)-LOSS(ISOTOP-1,1))
54.  A(10,1)=MASS(ISOTOP,1)-A(9,1)-A(8,1)-A(7,1)
55.  MASS(ISOTOP,1)=A(10,1)*EXPO(4)+A(9,1)*EXPO(3)+A(8,1)*EXPO(2)+A(7,1)
56.  1*EXPO(1)
57.  GO TO 9
58.  C FIFTH CHAIN MEMBER
59.  5 ABLE=DKFRCT(ISOTOP,1)*LAMBDA(ISOTOP-1)
60.  BAKER=DKFRCT(ISOTOP,2)*LAMBDA(ISOTOP-2)
61.  CHARLY=DKFRCT(ISOTOP,3)*LAMBDA(ISOTOP-3)
62.  A(11,1)=(ABLE*A(7,1)+BAKER*A(4,1)+CHARLY*A(2,1))/(LOSS(ISOTOP,1)-L
63.  1OSS(ISOTOP-4,1))
64.  A(12,1)=(ABLE*A(8,1)+BAKER*A(5,1)+CHARLY*A(3,1))/(LOSS(ISOTOP,1)-L
65.  1OSS(ISOTOP-3,1))
66.  A(13,1)=(ABLE*A(9,1)+BAKER*A(6,1))/(LOSS(ISOTOP,1)-LOSS(ISOTOP-2,1)
67.  1)
68.  A(14,1)=ABLE*A(10,1)/(LOSS(ISOTOP,1)-LOSS(ISOTOP-1,1))
69.  A(15,1)=MASS(ISOTOP,1)-A(14,1)-A(13,1)-A(12,1)-A(11,1)
70.  MASS(ISOTOP,1)=A(15,1)*EXPO(5)+A(14,1)*EXPO(4)+A(13,1)*EXPO(3)+A(12,1)
71.  1*EXPO(2)+A(11,1)*EXPO(1)
72.  GO TO 9
73.  C SIXTH CHAIN MEMBER
74.  6 ABLE=DKFRCT(ISOTOP,1)*LAMBDA(ISOTOP-1)
75.  BAKER=DKFRCT(ISOTOP,2)*LAMBDA(ISOTOP-2)
76.  CHARLY=DKFRCT(ISOTOP,3)*LAMBDA(ISOTOP-3)
77.  A(16,1)=(ABLE*A(11,1)+BAKER*A(7,1)+CHARLY*A(4,1))/(LOSS(ISOTOP,1)-L
78.  1OSS(ISOTOP-5,1))
79.  A(17,1)=(ABLE*A(12,1)+BAKER*A(8,1)+CHARLY*A(5,1))/(LOSS(ISOTOP,1)-L
80.  1OSS(ISOTOP-4,1))
81.  A(18,1)=(ABLE*A(13,1)+BAKER*A(9,1)+CHARLY*A(6,1))/(LOSS(ISOTOP,1)-L
82.  1OSS(ISOTOP-3,1))
83.  A(19,1)=(ABLE*A(14,1)+BAKER*A(10,1))/(LOSS(ISOTOP,1)-LOSS(ISOTOP-2,1)
84.  1)
85.  A(20,1)=ABLE*A(15,1)/(LOSS(ISOTOP,1)-LOSS(ISOTOP-1,1))
86.  A(21,1)=MASS(ISOTOP,1)-A(20,1)-A(19,1)-A(18,1)-A(17,1)-A(16,1)
87.  MASS(ISOTOP,1)=A(21,1)*EXPO(6)+A(20,1)*EXPO(5)+A(19,1)*EXPO(4)+A(18,1)
88.  1*EXPO(3)+A(17,1)*EXPO(2)+A(16,1)*EXPO(1)

```

```

89.      GO TO 9
90.  C SEVENTH CHAIN MEMBER
91.      7 ABLE=DKFRCT(ISO TOP,1)*LAMBDA(ISO TOP-1)
92.      BAKER=DKFRCT(ISO TOP,2)*LAMBDA(ISO TOP-2)
93.      CHARLY=DKFRCT(ISO TOP,3)*LAMBDA(ISO TOP-3)
94.      A(22,1)=(ABLE*A(16,1)+BAKER*A(11,1)+CHARLY*A(7,1))/(LOSS(ISO TOP,1)
95.      1-LOSS(ISO TOP-6,1))
96.      A(23,1)=(ABLE*A(17,1)+BAKER*A(12,1)+CHARLY*A(8,1))/(LOSS(ISO TOP,1)
97.      1-LOSS(ISO TOP-5,1))
98.      A(24,1)=(ABLE*A(18,1)+BAKER*A(13,1)+CHARLY*A(9,1))/(LOSS(ISO TOP,1)
99.      1-LOSS(ISO TOP-4,1))
100.     A(25,1)=(ABLE*A(19,1)+BAKER*A(14,1)+CHARLY*A(10,1))/(LOSS(ISO TOP,1)
101.     1-LOSS(ISO TOP-3,1))
102.     A(26,1)=(ABLE*A(20,1)+BAKER*A(15,1))/(LOSS(ISO TOP,1)-LOSS(ISO TOP-2
103.     1,1))
104.     A(27,1)=ABLE*A(21,1)/(LOSS(ISO TOP,1)-LOSS(ISO TOP-1,1))
105.     A(28,1)=MASS(ISO TOP,1)-A(27,1)-A(26,1)-A(25,1)-A(24,1)-A(23,1)-A(2
106.     12,1)
107.     MASS(ISO TOP,1)=A(28,1)*EXPO(7)+A(27,1)*EXPO(6)+A(26,1)*EXPO(5)+A(2
108.     15,1)*EXPO(4)+A(24,1)*EXPO(3)+A(23,1)*EXPO(2)+A(22,1)*EXPO(1)
109.     GO TO 9
110.  8 J=0
111.  9 CONTINUE
112.     GO TO 11
113. 10 ISO TOP=ISO TOP+LIM
114. 11 CONTINUE
115.  C RETURN TO DAFLE
116.     RETURN
117.     END

```

2. ELT REDUCE,1,710325, 30645

```
1. C SUBROUTINE REDUCE(J) ELIMINATES UNUSED NUCLIDES IN MASS ARRAY J
2.   SUBROUTINE REDUCE(J)
3.     COMMON ISOTOP,CHAINS,NOFNUC(96),DELTAT,MASS(600,10),SPACE(5400),
4.     LAMBDA(600),SKIP(96)
5.     REAL MASS,LAMBDA
6.     INTEGER CHAINS,SKIP
7. C VARIABLE SKIP IS 0 FOR UNUSED CHAINS. FOR USED CHAINS SKIP GIVES THE
8. C NUMBER OF THE FIRST NON-ZERO MEMBER.
9.     ISOTOP=0
10. C DO LOOP ON CHAINS
11.     DO 3 K=1,CHAINS
12.       LIM=NOFNUC(K)
13.       IF(SKIP(K).EQ.0) GO TO 3
14. C DO LOOP ON MEMBERS OF CHAIN K
15.     DO 1 I=1,LIM
16.       L=ISOTOP+1
17. C TEST NUCLIDE MASS
18.       IF(MASS(L,J)*LAMBDA(L).GT.1.E-30) GO TO 2
19.       1 CONTINUE
20.       I=0
21.       2 SKIP(K)=I
22.       3 ISOTOP=ISOTOP+LIM
23. C RETURN TO BAFFLE
24.     RETURN
25.     END
```


ELT PLUME,1,710622, 52057

```

1. C SUBROUTINE PLUME READS METEOROLOGICAL PARAMETERS AND CALCULATES
2. C NOSE RATE PER CURIE PER CM OF DOWNWIND CLOUD FOR EACH ENERGY GROUP.
3. C
4. C ERR1 IS ERROR RETURN FOR INCORRECT VALUE OF VARIABLE TYPE
5. C ERR2 IS ERROR RETURN FOR SHORT EXPOSURE TIME OR DISTANCE TOO LONG.
6. C
7. SUBROUTINE PLUME(ERR1,ERR2)
8. COMMON/LOGIC/LIBRARY,TAPE,BACKUP,REWIND,SEMI
9. COMMON NCD,CAPA(16),ALFA(16),ALFA1(16),MU(16),MEV(16),NOFNUC(96),
10. 1SKIP(96),MASS(600),LAMBDA(600),CHAINS,DKFRCT(450,3),N1,N2,OLDDT,
11. 2BURST(600,16),DRDX(43,16),TIME,NXINT,T(43),X(43),TEXPOS(5),GROUPS,08/04/70
12. 3NEXPO,TRELES,UBAR,DELD(6),DELT(6),TAPTIN 08/04/70
13. COMMON INTERP,RTIME,SY(43),SZ(43) 08/04/70
14. COMMON NDT,DK 01/25/71
15. LOGICAL DK 01/25/71
16. DIMENSION ZLIM(22),YLIM(22),EOVERG(43) 02/26/71
17. DIMENSION ZSUM(16),ZLIST(8,16),YLIST(8,16),YSUM(16) 07/16/70
18. DIMENSION DEP(43),XDIST(20),YSIG(20),ZSIG(20) 09/16/70
19. EQUIVALENCE (XDIST(1),DRDX(1,1)), (YSIG(1),YLIST(1,1)), (ZSIG(1),Z09/16/70
20. 1LIST(1,1)) 09/16/70
21. INTEGER TYPE,ENERGY,GROUPS,VERT,POSIT
22. INTEGER TYPE3 05/19/70
23. REAL MU,MEV,KSQD,MANDIS
24. DIMENSION Y(22),Z(22),YPM(10),ZPM(10)
25. LOGICAL ZDUB,YDUB,DEPLET,METIN,METOUT,PASS
26. LOGICAL LP
27. PASS=.FALSE. 08/19/70
28. C TRANSFER TO STATEMENT 115 ON READING THRU AN END-OF-FILE.
29. CALL SETEOF($115)
30. C SET EXPOSURE TIMES TO ZERO.
31. DEPLET=.FALSE. 06/29/70
32. DO 1 I=1,5
33. TEXPOS(I)=0.0
34. 1 CONTINUE
35. C
36. C READ FIRST METEOROLOGICAL DATA CARD
37. READ 2, TYPE,NDT,LP,DK,METIN,METOUT,DEPLET,UBAR,MANDIS,H,YP,TRELES
38. 2 FORMAT(I2,I3,5L1,6F10.0)
39. PRINT 3, TYPE,UBAR,MANDIS,H,TRELES,YP
40. 3 FORMAT('0METEOROLOGY TYPE 'I3,' WIND SPEED = 'F4.1,' M/SEC, DIST
41. 1ANCE FROM STACK TO MAN = 'E10.2,' METERS',' RELEASE HEIGHT = '
42. 2E10.2,' METERS, RELEASE TIME = 'E10.2,' HOURS, LATERAL DISTANCE
43. 3FROM CENTERLINE = 'E10.2,' METERS')
44. IF(NDT.LT.2) NDT=1 10/22/70
45. IF(NDT.EQ.1) GO TO 1003 10/22/70
46. DT=NDT 10/22/70
47. OLDDT=OLDDT*DT 10/22/70
48. C READ EXPOSURE TIMES
49. 1003 READ 4, TEXPOS 10/22/70
50. 4 FORMAT(8F10.0)
51. PRINT 5, TEXPOS
52. 5 FORMAT('0EXPOSURE TIMES SUBMITTED ARE '1P5E19.3)
53. IF(METIN) METOUT=.FALSE. 08/19/70
54. IF(METOUT) PUNCH 1005, TYPE,NDT,DK,DEPLET,UBAR,MANDIS,H,YP,TRELES
55. 1005 FORMAT(I2,I3,1X,L1,2HFF,L1,5F10.2)
56. IF(METOUT) PUNCH 2005, TEXPOS 08/19/70
57. 2005 FORMAT(8F10.3) 08/19/70
58. NEXPO=0
59. UBAR= 100.*UBAR
60. MANDIS=100.*MANDIS
61. H=100.*H
62. YP=100.*YP
63. TRELES=3600.*TRELES
64. C TEST TO SEE IF RELEASE TIME IS LESS THAN TIME BETWEEN CLOUD PUFFS.
65. IF(TRELES.GE.OLDDT) GO TO 8
66. PRINT 6, TRELES
67. 6 FORMAT(' RELEASE TIME TOO SHORT, THIS CASE BYPASSED'F10.2,' SEC')
68. PASS=.TRUE. 08/19/70
69. IF(METIN) GO TO 25 08/25/70
70. GO TO 43
71. 7 RETURN 2
72. C COUNT EXPOSURE TIMES AND ARRANGE IN INCREASING ORDER.
73. 8 DO 10 I=1,5
74. TEXPOS(I)=TEXPOS(I)*3600.
75. IF(TEXPOS(I)) 11,11,9
76. 9 NEXPO=I
77. 10 CONTINUE
78. 11 K=NEXPO-1
79. IF(NEXPO.LT.2) GO TO 212 07/16/70
80. DO 12 N=1,K 05/19/70
81. DO 12 I=1,K
82. IF(TEXPOS(I).LT.TEXPOS(I+1)) GO TO 12
83. DIFFX=TEXPOS(I)
84. TEXPOS(I)=TEXPOS(I+1)
85. TEXPOS(I+1)=DIFFX
86. 12 CONTINUE
87. 212 CONTINUE 07/16/70
88. C IF RELEASE TIME IS GREATER THAN TAPE TIME, RELEASE TIME = TAPE TIME.

```

```

89.      IF(TRELES.GT.TAPTIM) TRELES=TAPTIM
90.      IF(METIN) GO TO 25
91.      C (UPMAX IS MAXIMUM UPWIND INTEGRATION DISTANCE NECESSARY.
92.      HYP=SQRT(H*H+Y*Y)
93.      UPMAX=40000.+2.*HYP
94.      UPMD=10000.+2.*HYP
95.      C (UPDIS IS UPWIND INTEGRATION DISTANCE.
96.      UPDIS=AMIN1(60000.,MANDIS,UPMAX)
97.      C (DOWNMAX IS DISTANCE FROM MAN TO FRONT OF CLOUD AT END OF EXPOSURE TIME
98.      DOWNMAX=TEXPOS(NEXPO)*UBAR-MANDIS
99.      C (DWNDIS IS DOWNWIND INTEGRATION DISTANCE. (MAY BE NEGATIVE)
100.      DWNDIS=AMIN1(60000.,DOWNMAX,UPMAX)
101.      C (XMIN IS MINIMUM ALLOWABLE 8-POINT INTEGRATION DISTANCE ABOUT MAN.
102.      XMIN=3000.+0.7*HYP
103.      C TEST - UPWIND PLUS DOWNWIND DISTANCES ZERO IMPLIES NO EXPOSURE
104.      IF(UPDIS+DWNDIS) 13,13,15
105.      13 PRINT 14, UPDIS,DWNDIS
106.      14 FORMAT(' NO X-INTEGRATION INTERVAL. UPDIS ='1PE10.2,' METERS, DOWN
107.      DIS ='E10.2,' METERS')
108.      GO TO 43
109.      C TEST - IF DWNDIS LESS THAN 100CM, SKIP DOWNWIND INTEGRATION.
110.      15 IF(DWNDIS.LE.100.) GO TO 22
111.      C TEST - IF MANDIS LESS THAN 100 CM, SKIP UPWIND INTEGRATION.
112.      IF(MANDIS.LT.100.) GO TO 21
113.      C SET UPWIND INTEGRATION
114.      X(1)=MANDIS-UPDIS+10.
115.      IF(UPMD.GT.UPDIS-3000.) GO TO 16
116.      X(0)=MANDIS-UPMD
117.      X(15)=MANDIS-XMIN
118.      X(22)=MANDIS
119.      NXINT=3
120.      GO TO 18
121.      16 IF(XMIN.GT.UPDIS-1000.) GO TO 17
122.      X(8)=MANDIS-XMIN
123.      X(15)=MANDIS
124.      NXINT=2
125.      GO TO 18
126.      17 X(8)=MANDIS
127.      NXINT=1
128.      18 INEXT=7*(NXINT+1)+1
129.      C SET DOWNWIND INTEGRATION LIMITS
130.      IF(UPMD.GT.DWNDIS-3000.) GO TO 19
131.      X(INEXT)=MANDIS+XMIN
132.      X(INEXT+7)=MANDIS+UPMD
133.      X(INEXT+14)=MANDIS+DWNDIS
134.      NXINT=NXINT+3
135.      GO TO 25
136.      19 IF(XMIN.GT.DWNDIS-1000.) GO TO 20
137.      X(INEXT)=MANDIS+XMIN
138.      X(INEXT+7)=MANDIS+DWNDIS
139.      NXINT=NXINT+2
140.      GO TO 25
141.      20 X(INEXT)=MANDIS+DWNDIS
142.      NXINT=NXINT+1
143.      GO TO 25
144.      21 X(1)=MANDIS
145.      NXINT=0
146.      GO TO 18
147.      C SET INTEGRATION LIMITS FOR CLOUD THAT DOESN'T REACH MAN.
148.      22 DIFFX=UPDIS+DWNDIS
149.      X(1)=MANDIS-UPDIS+10.
150.      IF(UPMD.GT.DIFFX-3000.) GO TO 23
151.      X(8)=MANDIS+DWNDIS-UPMD
152.      X(15)=MANDIS+DWNDIS-XMIN
153.      X(22)=MANDIS+DWNDIS
154.      NXINT=3
155.      GO TO 25
156.      23 IF(XMIN.GT.DIFFX-1000.) GO TO 24
157.      X(8)=MANDIS+DWNDIS-XMIN
158.      X(15)=MANDIS+DWNDIS
159.      NXINT=2
160.      GO TO 25
161.      24 X(8)=MANDIS+DWNDIS
162.      NXINT=1
163.      C READ NECESSARY METEOROLOGICAL DATA
164.      25 IF(TYPE.LT.4) GO TO 28
165.      IF(TYPE.LE.9) GO TO 27
166.      PRINT 26, TYPE
167.      26 FORMAT(' METEOROLOGY SPECIFIED INCORRECTLY. TYPE = 'I6)
168.      126 RETURN 1
169.      27 TYPE3=TYPE-3
170.      GO TO 31
171.      28 GO TO (29,30,31), TYPE
172.      C SUTTONS WIND PARAMETERS
173.      29 READ 4, CY,CZ,EN
174.      IF(METOUT) PUNCH 134, CY,CZ,EN
175.      PRINT 129, CY,CZ,EN
176.      129 FORMAT(' CY,CZ,EN ',3F10.4)
177.      IF(METIN) GO TO 117
178.      CY=CY*100.
179.      CZ=CZ*100.
180.      GO TO 31

```

08/25/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

08/04/70

09/15/70

08/25/70

08/25/70

08/25/70

08/25/70

181.	C HANFORD WIND CONDITIONS	
182.	30 READ 4, A,B,C,DH,KSQD,SIGTUB	
183.	PRINT 130, A,B,C,DH,KSQD,SIGTUB	07/16/70
184.	130 FORMAT(' A,B,C,D,KSQD,SIGTUB'1P6E10.3)	07/16/70
185.	IF(METOUT) PUNCH 134, A,B,C,DH,KSQD,SIGTUB	08/25/70
186.	IF(METIN) GO TO 117	08/25/70
187.	A=A*1.E4	
188.	B=B*1.E4	
189.	C=C*1.E4	
190.	DH=100.*DH	
191.	SIGTUB=100.*SIGTUB	
192.	C CALCULATE X-GRID POSITIONS	
193.	31 IF(METIN) GO TO 117	11/24/70
194.	DO 33 I=1,NXINT	09/15/70
195.	J=7*(I-1)+1	
196.	K=J+7	
197.	DIFFX=(X(K)-X(J))/7.	
198.	DELD(I)=DIFFX	
199.	DELT(I)=DIFFX/UBAR	
200.	DO 32 L=2,7	
201.	POSIT=7*(I-1)+L	
202.	X(POSIT)=X(POSIT-1)+DIFFX	
203.	32 CONTINUE	
204.	33 CONTINUE	
205.	C CALCULATE TRAVEL TIME TO EACH POSITION	
206.	INEXT=7*NXINT+1	
207.	DO 34 I=1,INEXT	
208.	T(I)=X(I)/UBAR	
209.	DEP(I)=1.0	06/30/70
210.	34 CONTINUE	
211.	IF(METOUT) PUNCH 2, INEXT	08/19/70
212.	IF(METOUT) PUNCH 134, (X(I),I=1,INEXT)	08/19/70
213.	134 FORMAT(8E10.4)	08/19/70
214.	C TEST EXPOSURE TIMES TO BE SURE CLOUD REACHES SECOND POSITION	
215.	35 IF(TEXPOS(1).GE.T(2)) GO TO 39	
216.	IF(NEXPO.NE.1) GO TO 37	
217.	PRINT 36	
218.	36 FORMAT(' ALL EXPOSURE TIMES TOO SMALL. THIS CASE BYPASSED')	
219.	GO TO 42	
220.	37 K=NEXPO-1	
221.	DO 38 I=1,K	
222.	TEXPOS(I)=TEXPOS(I+1)	
223.	38 CONTINUE	
224.	NEXPO=NEXPO-1	
225.	GO TO 35	
226.	C CALCULATE SIGMA Y AND SIGMA Z FOR EACH POSITION	
227.	39 IF(TYPE.LE.3) GO TO 45	
228.	CALL PASSIG(TYPE3,INEXT,540)	
229.	PRINT 1006, SY	05/19/70
230.	1006 FORMAT(' SY = '10F10.2/(' '10F10.2))	05/19/70
231.	PRINT 1007, SZ	05/19/70
232.	1007 FORMAT(' SZ = '10F10.2/(' '10F10.2))	05/19/70
233.	GO TO 52	
234.	C ERROR RETURN FROM PASSIG - DISTANCE TOO LONG FOR LIBRARY DATA	
235.	C READ REMAINING CARDS FOR THIS CASE AND RETURN 2	
236.	40 PRINT 41, TYPE3	
237.	41 FORMAT(' LIBRARY DISTANCE DATA TO SHORT FOR PASQUILL TYPE 'I2)	
238.	RETURN 2	
239.	42 READ 4, (SY(I),I=1,INEXT)	
240.	READ 4, (SZ(I),I=1,INEXT)	
241.	GO TO 7	
242.	43 IF(TYPE.EQ.1) GO TO 44	
243.	IF(TYPE.EQ.3) GO TO 42	
244.	IF(TYPE.GT.3) GO TO 7	08/04/70
245.	READ 4, A	
246.	GO TO 7	
247.	44 READ 4, CY	
248.	GO TO 7	
249.	45 IF(TYPE.EQ.3) GO TO 50	
250.	IF(TYPE.EQ.2) GO TO 47	
251.	C SUTTON EQUATION	
252.	DO 46 I=1,INEXT	
253.	XPOWR=X(I)**(1.-EN*.5)/1.417	
254.	SY(I)=CY*XPOWR	
255.	SZ(I)=CZ*XPOWR	
256.	46 CONTINUE	
257.	PRINT 1006, SY	08/25/70
258.	PRINT 1007, SZ	08/25/70
259.	GO TO 52	
260.	C HANFORD	
261.	47 BIGA=C+DH*SIGTUB	
262.	ALPHA=.5*BIGA/(SIGTUB**2)	
263.	DO 48 I=1,INEXT	
264.	EXPO=T(I)/ALPHA	06/29/70
265.	IF(EXPO.LT..001) GO TO 147	06/29/70
266.	SIGY2=BIGA*(T(I)-ALPHA*(1.-EXP(-T(I)/ALPHA)))	
267.	GO TO 247	06/29/70
268.	147 SIGY2=12100.	07/16/70
269.	247 CONTINUE	06/29/70
270.	SY(I)=SQRT(SIGY2)	
271.	EXPO=KSQD*T(I)**2	06/29/70
272.	IF(EXPO.LT..001) GO TO 347	06/29/70

273.	SIGZ2=A*(1.-EXP(-KSQD*T(I)**2))+B*T(I)	
274.	GO TO 447	06/29/70
275.	347 SIGZ2=12100.	07/16/70
276.	447 CONTINUE	06/29/70
277.	SZ(I)=SQRT(SIGZ2)	
278.	48 CONTINUE	
279.	PRINT 1006, SY	07/16/70
280.	PRINT 1007, SZ	07/16/70
281.	GO TO 52	
282.	C INPUT SIGMA Y AND SIGMA Z	
283.	50 READ 1110, N	09/16/70
284.	K=1	09/16/70
285.	IF(N.LT.1.OR.N.GT.20) GO TO 250	
286.	READ 4, (XDIST(I),I=1,N)	09/16/70
287.	READ 4, (YSIG(I),I=1,N)	09/16/70
288.	READ 4, (ZSIG(I),I=1,N)	09/16/70
289.	C CONVERT XDIST(I) TO CM	10/27/70
290.	DO 550 I=1,N	10/27/70
291.	XDIST(I)=XDIST(I)*100.	10/27/70
292.	550 CONTINUE	10/27/70
293.	C INTERPOLATE INPUT DATA FOR SY AND SZ AT EACH X POSITION.	
294.	DO 150 I=1,INEXT	09/16/70
295.	CALL LINEAR(SY(I),YSIG,X(I),XDIST,K,N,\$250)	09/16/70
296.	CALL LINEAR(SZ(I),ZSIG,X(I),XDIST,K,N,\$250)	09/16/70
297.	150 CONTINUE	09/16/70
298.	IF(.NOT.METOUT) GO TO 450	09/16/70
299.	C CONVERT XDIST(I) BACK TO METERS FOR PUNCHING	10/27/70
300.	DO 650 I=1,N	10/27/70
301.	XDIST(I)=XDIST(I)*.01	10/27/70
302.	650 CONTINUE	10/27/70
303.	PUNCH 1110, N	09/16/70
304.	PUNCH 134, (XDIST(I),I=1,N)	09/16/70
305.	PUNCH 134, (YSIG(I),I=1,N)	09/16/70
306.	PUNCH 134, (ZSIG(I),I=1,N)	09/16/70
307.	GO TO 450	09/16/70
308.	C ERROR RETURN FROM LINEAR - TROUBLE WITH INPUT VALUES OF SY AND SZ.	
309.	250 PRINT 350, N	
310.	350 FORMAT(' TROUBLE IN SY,SZ INPUT, N=15)	
311.	RETURN 1	09/16/70
312.	C PRINT VALUES OF SY AND SZ	
313.	450 PRINT 1006, SY	09/16/70
314.	PRINT 1007, SZ	08/25/70
315.	IF(METIN) GO TO 1117	11/24/70
316.	DO 51 I=1,INEXT	
317.	SY(I)=SY(I)*100.	
318.	SZ(I)=SZ(I)*100.	
319.	51 CONTINUE	
320.	C CALCULATE AND PRINT E/Q VALUES	02/26/71
321.	52 DO 152 I=1,INEXT	02/26/71
322.	EYP=(YP**2)/(2.*SY(I)**2)	02/26/71
323.	EZP=(H**2)/(2.*SZ(I)**2)	02/26/71
324.	EOVERQ(I)=EXP(-EYP-EZP)/(2.*3.14159*UBAR*SY(I)*SZ(I))*1.E6	02/26/71
325.	152 CONTINUE	02/26/71
326.	PRINT 53, (EOVERQ(I),I=1,INEXT)	02/26/71
327.	53 FORMAT('E/Q VALUES AT EACH DOWNWIND POSITION ARE',(1P10E10,3))	02/26/71
328.	C	
329.	C	
330.	C CALCULATE DOSE FACTORS FOR EACH POSITION AND ENERGY	
331.	DO 110 IX=1,INEXT	
332.	SIGY=SY(IX)	
333.	SIGZ=SZ(IX)	01/12/71
334.	D=MANDIS-X(IX)	
335.	D=ABS(D)	
336.	C SET LIMITS FOR Z-INTEGRATION STARTING AT +3 SIGMA Z AND WORKING DOWN.	
337.	ZLIM(1)=3.*SIGZ	
338.	NZINT=0	
339.	S04=SIGZ/4.	
340.	S3=3.*SIGZ	
341.	IF(D.LT.101.) GO TO 57	07/16/70
342.	IF(D.LT.501.) GO TO 56	07/16/70
343.	IF(D.GT.SIGZ) GO TO 55	07/16/70
344.	54 ZLIM(2)=S04	07/16/70
345.	ZLIM(3)=0.	07/16/70
346.	NZINT=2	07/16/70
347.	GO TO 62	07/16/70
348.	55 ZLIM(2)=0.	07/16/70
349.	NZINT=1	07/16/70
350.	GO TO 62	07/16/70
351.	56 IF(SIGZ.LT.10001.) GO TO 54	07/16/70
352.	GO TO 58	07/16/70
353.	57 IF(SIGZ.LT.1001.) GO TO 61	07/16/70
354.	IF(SIGZ.LT.4001.) GO TO 60	07/16/70
355.	58 IF(SIGZ.LT.20001.) GO TO 59	07/16/70
356.	ZLIM(2)=S04	07/16/70
357.	ZLIM(3)=5000.	07/16/70
358.	ZLIM(4)=1000.	07/16/70
359.	ZLIM(5)=200.	07/16/70
360.	ZLIM(6)=0.	07/16/70
361.	NZINT=5	07/16/70
362.	GO TO 62	07/16/70
363.	59 ZLIM(2)=S04	07/16/70
364.	ZLIM(3)=1000.	07/16/70

```

365.      ZLIM(4)=200.
366.      ZLIM(5)=0.
367.      NZINT=4
368.      GO TO 62
369.      60 ZLIM(2)=SIGZ
370.      ZLIM(3)=S04
371.      ZLIM(4)=200.
372.      ZLIM(5)=0.
373.      NZINT=4
374.      GO TO 62
375.      61 ZLIM(2)=SIGZ
376.      ZLIM(3)=0.
377.      NZINT=2
378.      62 CONTINUE
379.      C SET Z INTEGRATION LIMITS BELOW CLOUD CENTERLINE.
380.      C FOR GROUND LEVEL RELEASE DOUBLE UPPER INTEGRAL (DOUBLE=.TRUE.)
381.      ZDUB=.FALSE.
382.      YDUB=.FALSE.
383.      IF(H.LT.100.) ZDUB=.TRUE.
384.      IF(YP.LT.100.) YDUB=.TRUE.
385.      IF(ZDUB) GO TO 87
386.      HOSZ=H/SIGZ
387.      IF(HOSZ.LT.1.) GO TO 163
388.      SZ3=3.*SIGZ
389.      ZMIN=-SZ3
390.      ZMAX=H+SZ3
391.      ZPMIN=H-5.*SIGZ
392.      IF(ZPMIN.GT.ZMIN) ZMIN=ZPMIN
393.      Z(1)=ZMAX
394.      Z(2)=H
395.      IF(ZMIN.LE.0.) GO TO 162
396.      Z(3)=H-SZ3
397.      Z(4)=ZMIN
398.      NZINT=3
399.      1162 N=NZINT+1
400.      DO 1262 I=1,N
401.      1262 ZLIM(I)=Z(I)-H
402.      GO TO 87
403.      162 ZP02=H/2.
404.      ZMID=SZ3
405.      IF(Z(2).LE.ZMID) ZMID=Z(2)
406.      IF(ZP02.GT.ZMID) ZMID=ZP02
407.      DO 262 J=1,NZINT
408.      262 ZPM(J)= ZLIM(NZINT-J+1)
409.      DO 362 J=1,NZINT
410.      IF(ZPM(J).GT.ZMID*.9) GO TO 462
411.      362 CONTINUE
412.      NZ=NZINT
413.      GO TO 562
414.      462 NZ=J-1
415.      562 IF(NZ.EQ.0) GO TO 762
416.      DO 662 J=1,NZ
417.      662 Z(J+2)=ZPM(NZ-J+1)
418.      762 Z(NZ+3)=0.0
419.      DO 862 J=1,NZINT
420.      I=NZ+J+3
421.      Z(I)=-ZLIM(NZINT+1-J)
422.      IF(Z(I).LT.ZMIN) GO TO 962
423.      862 CONTINUE
424.      GO TO 1062
425.      962 Z(I)=ZMIN
426.      1062 NZINT=I-1
427.      GO TO 1162
428.      163 DO 63 JZ=1,NZINT
429.      Z2=2.*ZLIM(NZINT-JZ+1)
430.      IF(Z2.GT.H) GO TO 64
431.      63 CONTINUE
432.      JZ=JZ+1
433.      64 KZ=JZ-1
434.      LZ=2*JZ+1
435.      INDEX=LZ+NZINT
436.      IF(JZ.GT.NZINT) INDEX=INDEX-1
437.      IF(KZ.LT.1) GO TO 66
438.      DO 65 I=1,KZ
439.      ZLIM(NZINT+I+1)=-ZLIM(NZINT-I+1)
440.      ZLIM(INDEX-I)=-H+ZLIM(NZINT-I+1)
441.      65 CONTINUE
442.      66 ZLIM(INDEX)=-H
443.      IF(JZ.LE.NZINT) ZLIM(NZINT+JZ+1)=-H/2.
444.      DO 67 I=1,NZINT
445.      ZLIM(INDEX+I)=-H-ZLIM(NZINT-I+1)
446.      67 CONTINUE
447.      NZINT=INDEX+NZINT-1
448.      87 CONTINUE
449.      C SET Y INTEGRATION LIMITS STARTING AT 0. AND WORKING UP
450.      DENOM=SIGZ*SIGY*UBAR*2.*3.14159
451.      SIGY2=SIGY*SIGY
452.      SIGZ2=SIGZ*SIGZ
453.      SY3=3.*SIGY
454.      DSIGY=SIGY*SIGY*2.
455.      DO 89 ENERGY=1,GROUPS
456.      ZSUM(ENERGY)=0.0

```

```

457.      89 CONTINUE
458.      DO 108 IZ=1,NZINT
459.      DELZ=(ZLIM(IZ)-ZLIM(IZ+1))/7.
460.      ZZ=ZLIM(IZ)+H
461.      DO 106 VERT=1,8
462.      ZMH2=(ZZ-H)**2
463.      VERTEX=.5*ZMH2/SIGZ2
464.      PERP2=D*D+ZZ*ZZ
465.      PERP=SQRT(PERP2)
466.      YLIM(1)=0.0
467.      IF(PERP.LT.101.) GO TO 93
468.      IF(SIGY.LT.PERP) GO TO 91
469.      PERP4=4.*PERP
470.      IF(SIGY.LT.PERP4) GO TO 92
471.      PERP15=15.*PERP
472.      IF(SIGY.LT.PERP15) GO TO 90
473.      PERP50=50.*PERP
474.      IF(SIGY.GT.PERP50) GO TO 189
475.      NYINT=4
476.      YLIM(2)=PERP4
477.      YLIM(3)=PERP15
478.      YLIM(4)=SIGY
479.      YLIM(5)=SY3
480.      GO TO 96
481.      189 NYINT=5
482.      YLIM(2)=PERP4
483.      YLIM(3)=PERP15
484.      YLIM(4)=PERP50
485.      YLIM(5)=SIGY
486.      YLIM(6)=SY3
487.      GO TO 96
488.      90 NYINT=3
489.      YLIM(2)=PERP4
490.      YLIM(3)=SIGY
491.      YLIM(4)=SY3
492.      GO TO 96
493.      91 YLIM(2)=SY3
494.      NYINT=1
495.      GO TO 96
496.      92 NYINT=2
497.      YLIM(2)=SIGY
498.      YLIM(3)=SY3
499.      GO TO 96
500.      93 SY04=SIGY/4.
501.      IF(SIGY.LT.800.) GO TO 94
502.      IF(SIGY.LT.4000.) GO TO 95
503.      IF(SIGY.GT.20000.) GO TO 193
504.      NYINT=5
505.      YLIM(2)=200.
506.      YLIM(3)=1000.
507.      YLIM(4)=SY04
508.      YLIM(5)=SIGY
509.      YLIM(6)=SY3
510.      GO TO 96
511.      193 NYINT=6
512.      YLIM(2)=200.
513.      YLIM(3)=1000.
514.      YLIM(4)=5000.
515.      YLIM(5)=SY04
516.      YLIM(6)=SIGY
517.      YLIM(7)=SY3
518.      GO TO 96
519.      94 NYINT=3
520.      YLIM(2)=SY04
521.      YLIM(3)=SIGY
522.      YLIM(4)=SY3
523.      GO TO 96
524.      95 NYINT=4
525.      YLIM(2)=200.
526.      YLIM(3)=SY04
527.      YLIM(4)=SIGY
528.      YLIM(5)=SY3
529.      96 DO 97 ENERGY=1,GROUPS
530.      YSUM(ENERGY)=0.0
531.      97 CONTINUE
532.      IF(YDUB) GO TO 2097
533.      C SET Y LIMITS FOR OFF CENTERLINE DOSE.
534.      YPOSY=Y/PERP
535.      IF(YPOSY.LT.1.) GO TO 1297
536.      YMIN=-SY3
537.      YMAX=5.*SIGY
538.      YPMAX=Y/PERP+SY3
539.      IF(YPMAX.LT.YMAX) YMAX=YPMAX
540.      Y(1)=-SY3
541.      Y(2)=0.
542.      IF(YMAX.GT.YP) GO TO 297
543.      Y(3)=SY3
544.      Y(4)=YMAX
545.      NYINT=3
546.      1897 N=NYINT+1
547.      DO 1997 I=1,N
548.      1997 YLIM(I)=Y(I)

```

07/16/70
07/16/70

07/16/70
07/16/70
07/16/70
07/16/70
07/16/70
07/16/70

07/16/70
07/16/70

07/16/70
07/16/70
07/16/70
07/16/70
07/16/70
07/16/70
07/16/70

```

549.      GO TO 2097
550. 297 CONTINUE
551.      YP02=YP/2.
552.      YMID=SY3
553.      IF(YP02.LT.SY3) YMID=YP02
554.      DO 397 J=1,NYINT
555. 397 YPM(J)=YP-YLIM(J+1)
556.      DO 497 J=1,NYINT
557.      IF(YPM(J).LT.YMID) GO TO 597
558. 497 CONTINUE
559.      NY=NYINT
560.      GO TO 697
561. 597 NY=J-1
562. 697 IF(NY.EQ.0) GO TO 897
563.      DO 797 J=1,NY
564. 797 Y(J+2)=YPM(NY+1-J)
565. 897 Y(NY+3)=YP
566.      DO 997 J=1,NYINT
567.      I=NY+3+J
568.      Y(I)=YP+YLIM(J+1)
569.      IF(Y(I).GT.YMAX) GO TO 1097
570. 997 CONTINUE
571.      GO TO 1197
572. 1097 Y(I)=YMAX
573. 1197 NYINT=I-1
574.      GO TO 1897
575. 1297 DO 1397 NY=1,NYINT
576.      Y2=2.*YLIM(NY+1)
577.      IF(Y2.GE.YP*.9) GO TO 1497
578. 1397 CONTINUE
579. 1497 KY=NY-1
580.      LY=2*NY+1
581.      INDEX=LY+NYINT
582.      IF(NY.GT.NYINT) INDEX=INDEX-1
583.      IF(KY.LT.1) GO TO 1697
584.      DO 1597 I=1,KY
585.      Y(NYINT+I+1)=YLIM(I+1)
586. 1597 Y(INDEX-I)=YP-YLIM(I+1)
587. 1697 Y(INDEX)=YP
588.      IF(NY.LE.NYINT) Y(NYINT+1+NY)=YP/2.
589.      DO 1797 I=1,NYINT
590.      Y(INDEX+I)=YP+YLIM(I+1)
591. 1797 Y(NYINT-I+1)=-YLIM(I+1)
592.      Y(NYINT+1)=0.
593.      NYINT=NYINT+INDEX-1
594.      GO TO 1897
595. 2097 VEXPO=EXP(-VERTEX)
596. C DO LOOP ON NUMBER OF MAJOR Y INTEGRATION INTERVALS.
597.      DO 104 IY=1,NYINT
598.      DELY=(YLIM(IY+1)-YLIM(IY))/7.
599.      YY=YLIM(IY)
600. C DO LOOP ON Y POSITIONS WITHIN MAJOR INTERVAL
601.      DO 102 LAT=1,8
602.      Y2=YY*YY
603.      YMYP2=(YY-YP)**2
604.      RSQ=PERP2+YMYP2
605.      R=SQRT(RSQ)
606. C CALCULATE INVERSE SQUARE FACTOR FOR CURRENT X,Y,Z POSITION.
607.      EQG=EXP(-Y2/DSIGY)/DENOM
608.      IF(R.LT.1) GO TO 99
609.      GNUM=1.
610.      IF(R.GT.140.) GO TO 98
611.      GNUM=1.-EXP(-6.283196*RSQ*.0001)
612. 98 GRATIO=GNUM/(12.56639*RSQ)
613.      GO TO 100
614. 99 GRATIO=.00005
615. C CALCULATE DOSE RATE KERNEL FOR EACH ENERGY GROUP AT X,Y,Z.
616. 100 DO 101 ENERGY=1,GROUPS
617.      RMU=MU(ENERGY)*R
618.      ATTEN=EXP(-RMU)
619.      BILDUP=1.+(CAPA(ENERGY)+ALFA(ENERGY)*RMU)*RMU
620.      YLIST(LAT,ENERGY)=VEXPO*GRATIO*EQG*ATTEN*BILDUP*MEV(ENERGY)
621. 101 CONTINUE
622.      YY=YY+DELY
623. 102 CONTINUE
624. C INTEGRATE AND SUM OVER Y LINE FOR X,Z
625.      DO 103 ENERGY=1,GROUPS
626.      IF(YOUB) GO TO 1103
627.      YSUM(ENERGY)=YSUM(ENERGY)+DELY*FINT(YLIST(1,ENERGY),8)
628.      GO TO 103
629. 1103 YSUM(ENERGY)=YSUM(ENERGY)+DELY*FINT(YLIST(1,ENERGY),8)*2.0
630. 103 CONTINUE
631. 104 CONTINUE
632.      DO 105 ENERGY=1,GROUPS
633.      ZLIST(VERT,ENERGY)=YSUM(ENERGY)
634. 105 CONTINUE
635.      ZZ=ZZ-DELZ
636. 106 CONTINUE
637. C INTEGRATE AND SUM OVER Z FOR CURRENT X POSITION
638.      DO 108 ENERGY=1,GROUPS
639.      IF(ZDUB) GO TO 107
640.      ZSUM(ENERGY)=ZSUM(ENERGY)+DELZ*FINT(ZLIST(1,ENERGY),8)

```

11/18/70
11/18/70
11/18/70

```

641.      GO TO 108
642.      107 ZSUM(ENERGY)=ZSUM(ENERGY)+DELZ*FINT(ZLIST(1,ENERGY),8)*2.
643.      108 CONTINUE
644.      DO 109 ENERGY=1, GROUPS
645.      DRDX(IX,ENERGY) = ZSUM(ENERGY)
646.      ZSUM(ENERGY)=0.0
647.      109 CONTINUE
648.      110 CONTINUE
649.      IF(.NOT.DEPLET) GO TO 7110
650.      C
651.      C READ CLOUD DEPLETION DATA
652.      READ 1110, N
653.      IF(N.GT.43.OR.N.LT.2) GO TO 5110
654.      READ 2110, (SY(I),SZ(I),I=1,N)
655.      1110 FORMAT(I3)
656.      2110 FORMAT(8F10.8)
657.      K=1
658.      C INTERPOLATE CLOUD DEPLETION DATA AND CORRECT INTEGRALS ACCORDINGLY.
659.      DO 4110 I=1, INEXT
660.      CALL LINEAR(YY,SZ,X(I),SY,K,N,$5110)
661.      DEP(I)=YY
662.      DO 3110 J=1, GROUPS
663.      DRDX(I,J)=DRDX(I,J)*DEP(I)
664.      3110 CONTINUE
665.      4110 CONTINUE
666.      IF(METOUT) PUNCH 134, (DEP(I),I=1,INEXT)
667.      GO TO 7110
668.      5110 PRINT 6110, N
669.      6110 FORMAT(' TROUBLE IN CLOUD DEPLETION DATA' I4)
670.      RETURN 1
671.      7110 DO 113 I=1, INEXT
672.      IF(.NOT.LP) GO TO 1112
673.      DIFFX=X(I)/100.
674.      PRINT 111, DIFFX, DEP(I)
675.      111 FORMAT(' AT ' F10.2, ' METERS, CLOUD DEPLETION FACTOR =' E12.6, ' THE
676.      1DOSE PER CURIE FACTORS ARE ')
677.      PRINT 112, (DRDX(I,J),J=1, GROUPS)
678.      112 FORMAT(10X, 1P8E13.4)
679.      1112 CONTINUE
680.      IF(METOUT) PUNCH 134, (DRDX(I,J),J=1, GROUPS)
681.      113 CONTINUE
682.      114 RETURN
683.      115 PRINT 116
684.      116 FORMAT(' END OF FILE ')
685.      STOP
686.      C
687.      C READ DOSE RATE/CM FACTORS
688.      117 READ 2, INEXT
689.      READ 134, (X(I),I=1,INEXT)
690.      IF(TYPE.EQ.3) GO TO 50
691.      1117 CONTINUE
692.      IF(.NOT.DEPLET) GO TO 118
693.      READ 134, (DEP(I),I=1,INEXT)
694.      GO TO 120
695.      118 DO 119 I=1, INEXT
696.      119 DEP(I)=1.
697.      120 DO 121 I=1, INEXT
698.      READ 134, (DRDX(I,J),J=1, GROUPS)
699.      DIFFX=X(I)/100.
700.      PRINT 111, DIFFX, DEP(I)
701.      PRINT 112, (DRDX(I,J),J=1, GROUPS)
702.      T(I)=X(I)/UBAR
703.      121 CONTINUE
704.      C WHEN RELEASE TIME IS TOO SHORT PASS=.TRUE. AND NO CALCULATION IS DONE.
705.      IF(PASS) RETURN 2
706.      NXINT=INEXT/7
707.      C FOR INPUT OF DOSE RATE FACTORS - CALCULATE MAJOR TIME AND DISTANCE
708.      C INCREMENTS.
709.      DO 122 I=1, NXINT
710.      J=7*(I-1)+1
711.      K=J+7
712.      DIFFX=(X(K)-X(J))/7
713.      DELD(I)=DIFFX
714.      DELT(I)=DIFFX/UBAR
715.      122 CONTINUE
716.      C RETURN TO EXDOSE
717.      RETURN
718.      END

```

05/19/70

06/29/70

06/29/70

08/04/70

06/29/70

06/29/70

06/29/70

09/16/70

06/29/70

06/29/70

06/29/70

06/29/70

06/29/70

08/19/70

06/29/70

06/29/70

06/29/70

06/29/70

06/29/70

06/29/70

06/29/70

06/29/70

06/29/70

06/29/70

06/29/70

08/19/70

08/19/70

08/19/70

08/19/70

11/25/70

11/24/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

08/19/70

Q ELT SEMME,1,710622, 37201

```

1. C SEMME CALCULATES DOSE RATE PER CURIE FOR EACH ENERGY GROUP AT
2. C EXPOSURE POINT. ERR1 IS ERROR RETURN FROM PASSIG.
3. C
4. SUBROUTINE SEMME(ERR1)
5. COMMON NCD,CAPA(16),ALFA(16),ALFA1(16),MU(16),MEV(16),NOFNUC(96),
6. 1SKIP(96),MASS(600),LAMBDA(600),CHAINS,DKFRCT(450,3),N1,N2,OLDDT,
7. 2BURST(600,16),DRDX(43,16),TIME,NXINT,T(43),X(43),TEXPOS(5),GROUPS,08/04/70.
8. 3NEXPO,TRELES,UBAR,DELD(6),DELT(6),TAPTIM,INTERP 08/04/70
9. COMMON RTIME,DUD(86),NDT,DK 01/25/71
10. LOGICAL DK 01/25/71
11. COMMON/LIBSEM/KEV(15)
12. INTEGER GROUPS,TYPE,TYPE3
13. LOGICAL INTERP
14. REAL MANDIS,MEV,KSQD,KEV
15. C READ FIRST METEOROLOGICAL PARAMETER CARD
16. READ 1, TYPE,DK,UBAR,MANDIS,H,YP,TRELES,TEXPOS(1),DELT(1)
17. 1 FORMAT(I2,4X,L1,3X,7F10.2)
18. PRINT 1000, TYPE,UBAR,MANDIS,H,YP,TRELES
19. 1000 FORMAT(' METEOROLGY TYPE'I2,' UBAR ='F6.2,' M/SEC. DISTANCE FRO
20. 1M RELEASE POINT TO MAN IS 'F10.1,' METERS/' HEIGHT OF RELEASE IS'
21. 2F7.1,' METERS. LATERAL DISPLACEMENT FROM CENTERLINE IS'F7.1,' MET
22. 3ERS')
23. IF(DELT(1).LT.1.E-30) DELT(1)=OLDDT
24. C CONVERT TO CGS UNITS
25. UBAR=100.*UBAR
26. MANDIS=MANDIS*100.
27. H=100.*H
28. YP=100.*YP
29. TRELES=3600.*TRELES
30. TEXPOS(1)=3600.*TEXPOS(1) 09/02/70
31. T(1)=MANDIS/UBAR
32. X(1)=MANDIS
33. IF(TYPE.LT.4) GO TO 1003
34. IF(TYPE.LE.9) GO TO 1002
35. PRINT 1001, TYPE
36. 1001 FORMAT(' METEOROLOGY SPECIFIED INCORRECTLY. TYPE = 'I5)
37. GO TO 2000
38. 1002 TYPE3=TYPE-3
39. GO TO 1007
40. 1003 GO TO (1004,1005,1006), TYPE
41. C SUTTON PARAMETERS
42. 1004 READ 1104, CY,CZ,EN
43. 1104 FORMAT(8F10.0)
44. CY=100.*CY
45. CZ=100.*CZ
46. XPOWER=X(1)**(1.-EN*.5)/1.417
47. SY=CY*XPOWER
48. SZ=CZ*XPOWER
49. GO TO 1010
50. C HANFORD WIND CONDITIONS
51. 1005 READ 1104, A,B,C,DH,KSQD,SIGTUB
52. A=A*1.E4
53. B=B*1.E4
54. C=C*1.E4
55. DH=100.*DH
56. SIGTUB=100.*SIGTUB
57. BIGA=C+DH*SIGTUB
58. ALPHA=.5*BIGA/(SIGTUB**2)
59. SY2=BIGA*(T(1)-ALPHA*(1.-EXP(-T(1)/ALPHA)))
60. SY=SQRT(SY2)
61. SZ2=A*(1.-EXP(-KSQD*T(1)**2))+B*T(1)
62. SZ=SQRT(SZ2)
63. GO TO 1010
64. C INPUT SIGMA Y AND SIGMA Z.
65. 1006 READ 1104, SY,SZ
66. SY=100.*SY
67. SZ=100.*SZ
68. GO TO 1010
69. C PASQUILL PARAMETERS CALCULATED
70. 1007 CALL PASSIG(TYPE3,1,51008)
71. SY=DELD(1)
72. SZ=DELD(2)
73. GO TO 1010
74. 1008 PRINT 1009, TYPE3
75. 1009 FORMAT(' LIBRARY DISTANCE DATA TOO SHORT FOR PASQUILL TYPE 'I2)
76. 2000 RETURN 1
77. C CALCULATE E/Q
78. 1010 EQQ=EXP(-.5*H**2/SZ**2)/(3.14159*SY*SZ*UBAR)
79. EQQ=EQQ*1.EG*EXP(-.5*YP**2/SY**2)
80. C CALCULATE DOSE RATE PER CURIE FOR EACH ENERGY GROUP
81. DRDX(1,1)=.26*EQQ*KEV(1)
82. L=GROUPS-1 12/16/70
83. DRDX(1,GROUPS)=.26*EQQ*KEV(L) 12/16/70
84. DO 1011 1-2,L 12/16/70
85. DRDX(1,I)=EQQ*.26*.5*(KEV(I)+KEV(I-1))
86. 1011 CONTINUE
87. PRINT 1111, (DRDX(1,I),I=1,GROUPS) 09/02/70
88. 1111 FORMAT(' DRDX FACTORS - DOSE RATE '/(1P8E12.4)) 09/02/70

```

```

89.      UBAR=UBAR*.01
90.  C   CONVERT DATA BACK TO ORIGINAL UNITS FOR PRINTING.
91.      CY=CY*.01
92.      CZ=CZ*.01
93.      A=A*1.E-4
94.      B=B*1.E-4
95.      C=C*1.E-4
96.      DH=DH*.01
97.      SIGTUB=SIGTUB*.01
98.      MANDIS=MANDIS*.01
99.      H=H*.01
100.     PRINT 1012, MANDIS,H
101. 1012 FORMAT('0RECEPTOR'F7.0,' METERS FROM'F5.0,' METER HIGH RELEASE POI
102.     INT')
103.     IF(TYPE.GT.3) GO TO 1019
104.     GO TO (1013,1015,1017), TYPE
105. 1013 PRINT 1014, CY,CZ,UBAR
106. 1014 FORMAT('0SUTTON PARAMETERS, CY ='F6.2,' CZ ='F6.2,' UBAR ='F4.1)
107.     GO TO 1021
108. 1015 PRINT 1016, A,B,C,DH,KSQD,SIGTUB,UBAR
109. 1016 FORMAT('0HANFORD PARAMETERS A,B,C,D,KSQD,SIGTUB,UBAR '1P7E10.2)
110.     GO TO 1021
111. 1017 SY=.01*SY
112.     SZ=.01*SZ
113.     PRINT 1018, SY,SZ
114. 1018 FORMAT('0INPUT PARAMETERS SIGMA Y ='F5.1,' SIGMA Z ='F5.1)
115.     GO TO 1021
116. 1019 PRINT 1020, TYPE3,SY,SZ
117. 1020 FORMAT('0PASQUILL TYPE'12,' SIGMA Y ='F6.0,' SIGMA Z ='F6.0)
118.  C   RETURN TO EXDOSE
119. 1021 RETURN
120.     END

```

```

1. C SUBROUTINE LIB READS A RIBD LIBRARY AND A PHOTON DATA LIBRARY.
2. C CALCULATE DKFRCT (FRACTIONAL YIELD FROM PARENTS) AND BURST (FRACTIONAL
3. C YIELD FROM EACH ENERGY GROUP * LAMBDA)
4. C
5. SUBROUTINE LIB
6. COMMON NCD,CAPA(16),ALFA(16),ALFA1(16),MU(16),MEV(16),NOFNUC(96), 07/13/70
7. 1SKIP(96),MASS(600),LAMBDA(600),CHAINS,DKFRCT(450,3),N1,N2,OLDDT, 07/13/70
8. 2BURST(600,16),DRDX(43,16),TIME,NXINT,T(43),X(43),TEXPOS(5),GROUPS,08/04/70
9. 3NEXPO,TRELES,UBAR,DELD(6),DELT(6),TAPTIM,INTERP,RTIME,SY(43), 08/04/70
10. 4SZ(43) 08/04/70
11. DIMENSION ML(450),FRACT(450),ISO(450),KEV(15),ENERGY(16)
12. DIMENSION ENG(16) 11/18/70
13. COMMON/LIBSEM/KEV
14. LOGICAL ERROR
15. INTEGER CHAINS,REREAD,GROUPS
16. REAL LAMBDA,MEV,KEV,MU
17. C
18. C READ RIBD DATA LIBRARY
19. READ (29,1) NCD
20. 1 FORMAT(I3)
21. READ (29,2) (ML(J),LAMBDA(J),FRACT(J),ISO(J),J=1,NCD)
22. 2 FORMAT(I3,2X,E9.3,32X,F5.3,20X,I1)
23. READ(29,1) N2
24. N1=NCD+1
25. N2=N2+NCD
26. READ(29,102) (LAMBDA(L),L=N1,N2)
27. 102 FORMAT(5X,F9.0)
28. DO 202 L=N1,N2
29. 202 LAMBDA(L)=.69314/86400./LAMBDA(L)
30. REWIND 29
31. ERROR=.FALSE.
32. MP=0
33. CHAINS=0
34. DO 3 I=1,96
35. 3 NOFNUC(I)=0
36. C CALCULATE LAMBDA IN SEC-1
37. DO 10 J=1,NCD
38. LAMBDA(J)=.69314/LAMBDA(J)/86400.
39. IF(MP-ML(J)) 4,5,6
40. 4 MP=ML(J)
41. C NEW MASS CHAIN
42. CHAINS=CHAINS+1
43. C COUNT NUMBER OF NUCLIDES IN THIS CHAIN
44. 5 NOFNUC(CHAINS)=NOFNUC(CHAINS)+1
45. GO TO 8
46. 6 ERROR=.TRUE.
47. PRINT 7,J
48. 7 FORMAT('OLIBRARY ENTRY' I4, ' IS OUT OF PLACE')
49. 8 DO 9 L=1,3
50. 9 DKFRCT(J,L)=0.0
51. 10 CONTINUE
52. IF(ERROR) STOP
53. C ISO=1 MEANS ISOMERIC STATE, =2 MEANS GROUND STATE
54. ISOTOP=0
55. DO 15 K=1,CHAINS
56. LIM=NOFNUC(K)
57. J=0
58. C DETERMINE FRACTIONAL DECAY YIELDS
59. DO 14 I=1,LIM
60. J=J+1
61. ISOTOP=ISOTOP+1
62. IF(J.EQ.1) GO TO 13
63. IF(ISO(ISOTOP).EQ.2) GO TO 11
64. DKFRCT(ISOTOP,1)=FRACT(ISOTOP-1)
65. GO TO 13
66. 11 DKFRCT(ISOTOP,1)=1.-FRACT(ISOTOP-1)
67. IF(J.EQ.2) GO TO 13
68. IF(ISO(ISOTOP-1).EQ.1) GO TO 12
69. IF(ISO(ISOTOP-2).EQ.2) GO TO 13
70. DKFRCT(ISOTOP,2)=FRACT(ISOTOP-2)
71. GO TO 13
72. 12 DKFRCT(ISOTOP,2)=1.-FRACT(ISOTOP-2)
73. IF(J.EQ.3.OR.ISO(ISOTOP-3).EQ.2) GO TO 13
74. DKFRCT(ISOTOP,3)=FRACT(ISOTOP-3)
75. 13 IF(LAMBDA(ISOTOP).LT.1.E-30) J=0
76. 14 CONTINUE
77. 15 CONTINUE
78. C
79. C READ PHOTON DATA LIBRARY
80. REREAD=-4
81. READ(28,1) GROUPS
82. LIM=GROUPS-1
83. C KEV IS UPPER MEV LIMIT OF GROUP FOR LOWER 15 GROUPS
84. READ (28,16) (KEV(I),I=1,LIM)
85. 16 FORMAT(16F5.0)
86. READ(28,216) (MEV(I),I=1,GROUPS)
87. 216 FORMAT(8E10.0)
88. IF(GROUPS.GT.1) GO TO 316

```

89.	ENG(1)= .7	11/18/70
90.	GO TO 616	11/18/70
91.	316 IF(GROUPS.GT.2) GO TO 416	11/18/70
92.	ENG(1)=.6667*KEV(1)	11/18/70
93.	ENG(2)=1.2*KEV(1)	11/18/70
94.	GO TO 616	11/18/70
95.	416 ENG(1)= .6667*KEV(1)	11/18/70
96.	ENG(GROUPS)=1.2*KEV(GROUPS-1)	11/18/70
97.	L=GROUPS-1	11/18/70
98.	DO 516 I=2,L	11/18/70
99.	516 ENG(I)=(KEV(I)+KEV(I-1))/2.0	11/18/70
100.	616 CONTINUE	11/18/70
101.	DO 23 ISOTOP=1,N2	
102.	DO 116 L=1,16	
103.	116 BURST(ISOTOP,L)=0.0	
104.	READ(28,17) NO,(FRACT(J),ENERGY(J),J=1,6)	
105.	17 FORMAT(8X,12,12F5.0)	
106.	IF(NO.EQ.0) GO TO 23	
107.	IF(NO.LT.7) GO TO 20	
108.	READ(28,19) (FRACT(J),ENERGY(J),J=7,13)	
109.	IF(NO.LT.14) GO TO 20	
110.	READ(28,19) (FRACT(J),ENERGY(J),J=14,NO)	
111.	19 FORMAT(14F5.0)	
112.	C CALCULATE BURST FOR CURRENT ISOTOPE	
113.	20 DO 22 K=1,NO	
114.	DO 21 I=1,LIM	
115.	IF(ENERGY(K).LE.KEV(I)) GO TO 22	11/18/70
116.	21 CONTINUE	
117.	I=GROUPS	
118.	22 BURST(ISOTOP,I)=BURST(ISOTOP,I)+FRACT(K)*LAMBDA(ISOTOP)*ENERGY(K)/11/18/70	11/24/70
119.	1ENG(I)	
120.	23 CONTINUE	
121.	C READ BUILDUP AND ATTENUATION PARAMETERS	
122.	READ (28,24) (CAPA(I),I=1,GROUPS)	
123.	24 FORMAT(8F10.0)	
124.	READ(28,24) (ALFA(I),I=1,GROUPS)	
125.	READ(28,24) (ALFA1(I),I=1,GROUPS)	
126.	READ(28,24) (MU(I),I=1,GROUPS)	
127.	DO 25 I = 1,GROUPS	
128.	25 MU(I) = MU(I)*1.293E-3	
129.	REWIND 28	
130.	C RETURN TO EXDOSE	
131.	RETURN	
132.	END	

```

1. C SUBROUTINE CHAIN DECAYS NUCLIDES FOR A TIME THETA
2. C
3. SUBROUTINE CHAIN(THETA)
4. DIMENSION EXPO(8)
5. COMMON SPACE(81),NOFNUC(96),SKIP(96),MASS(600),LAMBDA(600),CHAINS,07/13/70
6. IDKFRCT(450,3) 07/13/70
7. REAL LAMBDA,MASS
8. INTEGER CHAINS,SKIP 07/13/70
9. ISOTOP=0
10. C DO LOOP ON NUMBER OF CHAINS
11. DO 11 K=1,CHAINS
12. LIM=NOFNUC(K)
13. IF(SKIP(K).NE.0) GO TO 10
14. J=0
15. C DO LOOP ON ISOTOPES IN THE CHAIN
16. DO 9 I=1,LIM
17. ISOTOP=ISOTOP+1
18. J=J+1
19. IF(LAMBDA(ISOTOP).LT.1.E-30) GO TO 8
20. EXPO(J)=EXP(-LAMBDA(ISOTOP)*THETA)
21. C TRANSFER ON CHAIN MEMBER J
22. GO TO (1,2,3,4,5,6,7),J
23. C FIRST CHAIN MEMBER
24. 1 A1=MASS(ISOTOP) 07/13/70
25. MASS(ISOTOP)=A1*EXPO(1) 07/13/70
26. GO TO 9
27. C SECOND CHAIN MEMBER
28. 2 A2=DKFRCT(ISOTOP,1)*LAMBDA(ISOTOP-1)*A1/(LAMBDA(ISOTOP)-LAMBDA(ISO
29. TOP-1))
30. A3=MASS(ISOTOP)-A2 07/13/70
31. MASS(ISOTOP)=A3*EXPO(2)+A2*EXPO(1) 07/13/70
32. GO TO 9
33. C THIRD CHAIN MEMBER
34. 3 ABLE=DKFRCT(ISOTOP,1)*LAMBDA(ISOTOP-1)
35. A4=(ABLE*A2+DKFRCT(ISOTOP,2)*LAMBDA(ISOTOP-2)*A1)/(LAMBDA(ISOTOP)-
36. LAMBDA(ISOTOP-2))
37. A5=ABLE*A3/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-1))
38. A6=MASS(ISOTOP)-A4-A5 07/13/70
39. MASS(ISOTOP)=A6*EXPO(3)+A5*EXPO(2)+A4*EXPO(1) 07/13/70
40. GO TO 9
41. C FOURTH CHAIN MEMBER
42. 4 ABLE=DKFRCT(ISOTOP,1)*LAMBDA(ISOTOP-1)
43. BAKER=DKFRCT(ISOTOP,2)*LAMBDA(ISOTOP-2)
44. A7=(ABLE*A4+BAKER*A2+DKFRCT(ISOTOP,3)*LAMBDA(ISOTOP-3)*A1)/(LAMBDA
45. (ISOTOP)-LAMBDA(ISOTOP-3))
46. A8=(ABLE*A5+BAKER*A3)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-2)) 11/18/70
47. A9=ABLE*A6/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-1))
48. A10=MASS(ISOTOP)-A9-A8-A7 07/13/70
49. MASS(ISOTOP)=A10*EXPO(4)+A9*EXPO(3)+A8*EXPO(2)+A7*EXPO(1) 07/13/70
50. GO TO 9
51. C FIFTH CHAIN MEMBER
52. 5 ABLE=DKFRCT(ISOTOP,1)*LAMBDA(ISOTOP-1)
53. BAKER=DKFRCT(ISOTOP,2)*LAMBDA(ISOTOP-2)
54. CHARLY=DKFRCT(ISOTOP,3)*LAMBDA(ISOTOP-3)
55. A11=(ABLE*A7+BAKER*A4+CHARLY*A2)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-4))
56. 1)
57. A12=(ABLE*A8+BAKER*A5+CHARLY*A3)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-3))
58. 1)
59. A13=(ABLE*A9+BAKER*A6)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-2))
60. A14=ABLE*A10/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-1))
61. A15=MASS(ISOTOP)-A14-A13-A12-A11 07/13/70
62. MASS(ISOTOP)=A15*EXPO(5)+A14*EXPO(4)+A13*EXPO(3)+A12*EXPO(2)+A11*EXPO(1) 07/13/70
63. EXPO(1) 07/13/70
64. GO TO 9
65. C SIXTH CHAIN MEMBER
66. 6 ABLE=DKFRCT(ISOTOP,1)*LAMBDA(ISOTOP-1)
67. BAKER=DKFRCT(ISOTOP,2)*LAMBDA(ISOTOP-2)
68. CHARLY=DKFRCT(ISOTOP,3)*LAMBDA(ISOTOP-3)
69. A16=(ABLE*A11+BAKER*A7+CHARLY*A4)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-5))
70. 1)
71. A17=(ABLE*A12+BAKER*A8+CHARLY*A5)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-4))
72. 1)
73. A18=(ABLE*A13+BAKER*A9+CHARLY*A6)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-3))
74. 1)
75. A19=(ABLE*A14+BAKER*A10)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-2))
76. A20=ABLE*A15/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-1))
77. A21=MASS(ISOTOP)-A20-A19-A18-A17-A16 07/13/70
78. MASS(ISOTOP)=A21*EXPO(6)+A20*EXPO(5)+A19*EXPO(4)+A18*EXPO(3)+A17*EXPO(2)+A16*EXPO(1) 07/13/70
79. EXPO(1) 07/13/70
80. GO TO 9
81. C SEVENTH CHAIN MEMBER
82. 7 ABLE=DKFRCT(ISOTOP,1)*LAMBDA(ISOTOP-1)
83. BAKER=DKFRCT(ISOTOP,2)*LAMBDA(ISOTOP-2)
84. CHARLY=DKFRCT(ISOTOP,3)*LAMBDA(ISOTOP-3)
85. A22=(ABLE*A16+BAKER*A11+CHARLY*A7)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-6))
86. 1)
87. A23=(ABLE*A17+BAKER*A12+CHARLY*A8)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-5))
88. 1)

```

```

89.      A24=(ABLE*A18+BAKER*A13+CHARLY*A9)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-4
90.      1))
91.      A25=(ABLE*A19+BAKER*A14+CHARLY*A10)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-
92.      13))
93.      A26=(ABLE*A20+BAKER*A15)/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-2))
94.      A27=ABLE*A21/(LAMBDA(ISOTOP)-LAMBDA(ISOTOP-1))
95.      A28=MASS(ISOTOP)-A27-A26-A25-A24-A23-A22
96.      MASS(ISOTOP)=A28*EXP0(7)+A27*EXP0(6)+A26*EXP0(5)+A25*EXP0(4)+A24*EXP0(3)+A23*EXP0(2)+A22*EXP0(1)
97.      GO TO 9
98.
99.      8 J=0
100.     9 CONTINUE
101.     GO TO 11
102.     10 ISOTOP=ISOTOP+LIM
103.     11 CONTINUE
104.     C RETURN TO EXDOSE
105.     RETURN
106.     END

```

07/13/70

07/13/70

07/13/70

```

1.      SUBROUTINE FIT(I1)
2.      C
3.      C FIT CLOUD DATA FROM CONTAINMENT WITH QUADRATIC IN TIME -READING NEW DATA WHEN
4.      C NECESSARY- AND INTERPOLATE FOR DESIRED TIME
5.      C
6.      DIMENSION A(600),B(600),C(600),HOLD(600,3),TOLD(3)
7.      COMMON/LOGIC/SPACE,TAPE
8.      COMMON NCD,CAPA(16),ALFA(16),ALFA1(16),MU(16),MEV(16),NOFNUC(96), 07/13/70
9.      1SKIP(96),MASS(600),LAMBDA(600),CHAINS,DKFRCT(450,3),N1,N2,OLDDT, 07/13/70
10.     2BURST(600,16),DRDX(43,16),TIME,NXINT,T(43),X(43),TEXPOS(5),GROUPS,08/04/70
11.     3NEXPO,TRELES,UBAR,DELD(6),DELT(6),TAPTIM,INTERP,RTIME,SY(43), 08/04/70
12.     4SZ(43) 08/04/70
13.     REAL MASS
14.     INTEGER UNIT,STATUS,SKIP,CHAINS
15.     LOGICAL TAPE
16.     GO TO (1,6),I1
17.     C READ FIRST THREE CLOUD RECORDS AND SETUP ROUTINE FOR FURTHER ENTRIES
18.     1 ODT2=2.*OLDDT*OLDDT
19.     UNIT=26
20.     IF(TAPE) UNIT=7
21.     DO 5 I=1,3
22.     CALL NTRAN(UNIT,2,N2,HOLD(1,I),STATUS)
23.     STATUS=STATUS
24.     2 IF(STATUS+1) 3,2,5
25.     5 CONTINUE
26.     TOLD(1)=0.0
27.     TOLD(2)=OLDDT
28.     TOLD(3)=TOLD(2)+OLDDT
29.     GO TO 12
30.     C CHECK TO SEE IF STILL IN FITTED REGION
31.     6 IF(RTIME.LT.TOLD(3)) GO TO 15
32.     C NEED MORE DATA
33.     7 TOLD(1)=TOLD(2)
34.     TOLD(2)=TOLD(3)
35.     TOLD(3)=TOLD(3)+OLDDT
36.     DO 8 I=2,3
37.     DO 8 J=1,N2
38.     8 HOLD(J,I-1)=HOLD(J,I)
39.     CALL NTRAN(UNIT,2,N2,HOLD(1,3),STATUS)
40.     STATUS=STATUS
41.     9 IF(STATUS+1) 3,9,12
42.     12 IF(RTIME.GT.TOLD(3)) GO TO 7
43.     C GET COEFFICIENTS
44.     ISOTOP=0
45.     TERM=TOLD(2)**2-TOLD(1)**2
46.     DO 14 K=1,CHAINS
47.     LIM=NOFNUC(K)
48.     IF(SKIP(K).NE.0) GO TO 14
49.     DO 13 I=1,LIM
50.     L=ISOTOP+I
51.     C(L)=(HOLD(L,3)+HOLD(L,1)-2.*HOLD(L,2))/ODT2
52.     B(L)=(HOLD(L,2)-HOLD(L,1)-TERM*C(L))/OLDDT
53.     13 A(L)=HOLD(L,1)-TOLD(1)*(B(L)+TOLD(1)*C(L))
54.     14 ISOTOP=ISOTOP+LIM
55.     DO 114 K=N1,N2
56.     C(K)=(HOLD(K,3)+HOLD(K,1)-2.*HOLD(K,2))/ODT2
57.     B(K)=(HOLD(K,2)-HOLD(K,1)-TERM*C(K))/OLDDT
58.     114 A(K)=HOLD(K,1)-TOLD(1)*(B(K)+TOLD(1)*C(K))
59.     C INTERPOLATE
60.     15 ISOTOP=0
61.     DO 17 K=1,CHAINS
62.     LIM=NOFNUC(K)
63.     IF(SKIP(K).NE.0) GO TO 17
64.     DO 16 I=1,LIM
65.     L=ISOTOP+I
66.     16 MASS(L)=A(L)+RTIME*(B(L)+RTIME*C(L)) 07/13/70
67.     17 ISOTOP=ISOTOP+LIM
68.     DO 117 K=N1,N2
69.     117 MASS(K)=A(K)+RTIME*(B(K)+RTIME*C(K)) 07/13/70
70.     C RETURN TO EXDOSE
71.     RETURN
72.     3 PRINT 4, STATUS
73.     4 FORMAT('O READ TROUBLES IN FIT - STATUS -',I3)
74.     STOP
75.     END

```

```

1.  C PASSIG INTERPOLATES STORED PASQUILL CURVES FOR SY AND SZ FOR TYPE TYPE
2.  C AND NUM X POSITIONS
3.  SUBROUTINE PASSIG(TYPE,NUM,ERR)
4.  COMMON/LOGIC/LIBRARY,TAPE,BACKUP,REWIND,SEMI
5.  COMMON NCD,CAPA(16),ALFA(16),ALFA1(16),MU(16),MEV(16),NOFNUC(96),
6.  1SKIP(96),MASS(600),LAMBDA(600),CHAINS,DKFRCT(450,3),N1,N2,OLDDT,
7.  2BURST(600,16),DRDX(43,16),TIME,NXINT,T(43),X(43),TEXPOS(5),GROUPS,08/04/70
8.  3NEXPO,TRELES,UBAR,DELD(6),DELT(6),TAPTIM,INTERP,RTIME,SY(43),08/04/70
9.  4SZ(43)08/04/70
10. LOGICAL SEMI,IDOSE08/26/70
11. INTEGER TYPE
12. DIMENSION SIGY(6,20),SIGZ(6,20),DIST(20)
13. C DISTANCES TO WHICH SY AND SZ DATA CORRESPOND.
14. DATA DIST/0.0,1.E4,1.5E4,2.5E4,3.5E4,5.E4,7.E4,1.E5,1.5E5,2.5E5,
15. 13.5E5,5.E5,7.E5,1.E6,1.5E6,2.5E6,3.5E6,5.E6,7.E6,1.E7/
16. C SIGMA Y DATA - 1 FOR TYPE A, 2 FOR TYPE B, ETC.
17. DATA (SIGY(1,I),I=1,20)/20.,2.75E3,3.8E3,6.1E3,8.3E3,1.15E4,1.54E4
18. 1,2.14E4,3.05E4,4.7E4,6.3E4,8.7E4,1.14E5,1.56E5,2.2E5,3.32E5,4.4E5,
19. 25.95E5,8.E5,1.15E6/
20. DATA (SIGY(2,I),I=1,20)/90.,1.95E3,2.75E3,4.4E3,6.E3,8.3E3,1.12E4,
21. 11.56E4,2.25E4,3.6E4,4.9E4,6.6E4,8.8E4,1.18E5,1.68E5,2.6E5,3.45E5,4
22. 2.E5,6.1E5,8.2E5/
23. DATA (SIGY(3,I),I=1,20)/90.,1.26E3,1.8E3,2.9E3,4.E3,5.5E3,7.5E3,1.
24. 104E4,1.51E4,2.44E4,3.3E4,4.5E4,6.1E4,8.4E4,1.19E5,1.87E5,2.5E5,3.3
25. 25E5,4.5E5,6.2E5/
26. DATA (SIGY(4,I),I=1,20)/90.,8.2E2,1.18E3,1.9E3,2.6E3,3.6E3,4.9E3,6
27. 1.9E3,1.E4,1.6E4,2.17E4,3.E4,4.05E4,5.5E4,7.9E4,1.21E5,1.62E5,2.2E5
28. 2,3.E5,4.1E5/
29. DATA (SIGY(5,I),I=1,20)/90.,6.1E2,8.8E2,1.4E3,1.9E3,2.6E3,3.6E3,5.
30. 105E3,7.3E3,1.18E4,1.6E4,2.22E4,3.E4,4.2E4,5.9E4,9.2E4,1.21E5,1.68E
31. 25,2.25E5,3.E5/
32. DATA (SIGY(6,I),I=1,20)/90.,4.2E2,5.8E2,9.4E2,1.19E3,1.77E3,2.4E3,
33. 13.4E3,4.9E3,7.8E3,1.07E4,1.48E4,2.E4,2.72E4,3.95E4,6.2E4,8.3E4,1.1
34. 22E5,1.5E5,2.E5/
35. C SIGMA Z DATA - 1 FOR TYPE A, 2 FOR TYPE B, ETC.
36. DATA (SIGZ(1,I),I=1,20)/100.,1.4E3,2.2E3,3.8E3,6.E3,1.08E4,2.17E4,
37. 14.6E4,1.08E5,3.15E5,6.0E5,9*-1.0/
38. DATA (SIGZ(2,I),I=1,20)/100.,1.08E3,1.56E3,2.55E3,3.5E3,5.1E3,7.4E
39. 13,1.1E4,1.7E4,3.E4,4.3E4,6.4E4,9.2E4,1.35E5,2.1E5,3.75E5,6.4E5,-1.
40. 2,-1.,-1./
41. DATA (SIGZ(3,I),I=1,20)/100.,7.3E2,1.06E3,1.7E3,2.3E3,3.2E3,4.4E3,
42. 16.1E3,8.9E3,1.42E4,1.95E4,2.65E4,3.65E4,5.E4,7.4E4,1.15E5,1.6E5,2.
43. 22E5,2.9E5,4.E5/
44. DATA (SIGZ(4,I),I=1,20)/100.,4.6E2,6.6E2,1.02E3,1.37E3,1.85E3,2.4E
45. 13,3.15E3,4.2E3,5.8E3,7.1E3,8.8E3,1.1E4,1.37E4,1.72E4,2.3E4,2.75E4,
46. 23.3E4,3.9E4,4.6E4/
47. DATA (SIGZ(5,I),I=1,20)/100.,3.5E2,5.E2,7.6E2,9.9E2,1.3E3,1.65E3,2
48. 1.15E3,2.8E3,3.8E3,4.7E3,5.5E3,6.6E3,7.9E3,9.6E3,1.21E4,1.39E4,1.54
49. 2E4,1.7E4,1.85E4/
50. DATA (SIGZ(6,I),I=1,20)/100.,2.3E2,3.2E2,4.9E2,6.3E2,8.5E2,1.1E3,1
51. 1.4E3,1.8E3,2.4E3,2.86E3,3.4E3,4.E3,4.65E3,5.4E3,6.4E3,7.1E3,7.8E3,
52. 28.6E3,9.3E3/
53. IDOSE=.FALSE.08/26/70
54. IF(NUM.EQ.100) IDOSE=.TRUE.08/26/70
55. IF(IDOSE) NUM=108/26/70
56. IF(IDOSE) X(1)=CAPA(3)08/26/70
57. IDATA=1
58.
59. C
60. C CALCULATE SY AND SZ FOR EACH X POSITION DESIRED
61. DO 5 I=1,NUM
62. 1 IF(IDATA.GT.20) GO TO 7
63. IF(SIGY(TYPE,IDATA)) 7,7,101
64. 1n1 IF(SIGZ(TYPE,IDATA)) 7,7,201
65. 201 IF(X(I)-DIST(IDATA)) 4,3,2
66. 2 IDATA=IDATA+1
67. GO TO 1
68. 3 SY(I)=SIGY(TYPE,IDATA)
69. SZ(I)=SIGZ(TYPE,IDATA)
70. GO TO 5
71. 4 IF(IDATA.EQ.1) GO TO 3
72. SY1=SIGY(TYPE,IDATA-1)
73. SY2=SIGY(TYPE,IDATA)
74. SZ1=SIGZ(TYPE,IDATA-1)
75. SZ2=SIGZ(TYPE,IDATA)
76. D1=DIST(IDATA-1)
77. D2=DIST(IDATA)
78. SY(I)=SY1+(X(I)-D1)*(SY2-SY1)/(D2-D1)
79. SZ(I)=SZ1+(X(I)-D1)*(SZ2-SZ1)/(D2-D1)
80. 5 CONTINUE08/26/70
81. IF(.NOT.IDOSE) GO TO 10508/26/70
82. CAPA(1)=SY(1)08/26/70
83. CAPA(2)=SZ(1)08/26/70
84. GO TO 608/26/70
85. 1n5 CONTINUE08/26/70
86. IF(.NOT.SEMI) GO TO 6
87. DELD(1)=SY(1)
88. DELD(2)=SZ(1)
89. 6 RETURN

```


89. C ERROR RETURN WHEN DISTANCE SPECIFIED IS BEYOND RANGE OF STORED DATA.
90. 7 RETURN 3.
91. END

Q ELT LINAR,1,710519, 34890

```

1.      SUBROUTINE LINEAR(Y, YA, X, XA, N, M, IERROR)
2.      C SUBROUTINE LINEAR INTERPOLATES YA=F(XA) FOR Y AT X
3.      C IERROR IS ERROR RETURN WHEN X IS NOT WITHIN XA
4.      DIMENSION XA(3), YA(3)
5.      C N IS PREVIOUS POSITION IN TABLE
6.      C DATA MUST BE IN ORDER OF INCREASING X
7.      IF(X.LT.XA(M)) GO TO 5
8.      Y=YA(M)
9.      RETURN
10.     5 IF(X.LT.XA(1)) RETURN 7
11.     DO 1 J=N,M
12.     IF(XA(J)-X) 1,2,4
13.     1 CONTINUE
14.     2 Y=YA(J)
15.     3 N=J
16.     RETURN
17.     4 Y=YA(J-1)+(X-XA(J-1))*(YA(J)-YA(J-1))/(XA(J)-XA(J-1))
18.     GO TO 3
19.     END

```

Q ELT FINT,1,710325, 30864

```

1.      FUNCTION FINT(LIST, NUMBER)
2.      C
3.      C THIS FUNCTION RETURNS THE INTEGRAL FROM THE FIRST EQUALLY SPACED INTEGRAND
4.      C IN THE LIST TO THE LAST (NUMBER)
5.      C MAXIMUM NUMBER IS 8
6.      C
7.      REAL LIST
8.      DIMENSION A(35), B(7), C(7), LIST(8)
9.      DATA B/2*1., 3., 2., 5., 1., 7., /C/2., 3., 8., 45., 288., 140., 17280./
10.     DATA A/3*1., 4., 2*1., 2*3., 1., 7., 32., 12., 32., 7., 19., 75., 2*50., 75.,
11.     A 19., 41., 216., 27., 272., 27., 216., 41., 751., 3577., 1323., 2*2989.,
12.     B 1323., 3577., 751./
13.     C
14.     INDEX=(NUMBER*NUMBER)/2-1-NUMBER/2
15.     FINT=0.0
16.     DO 1 I=1, NUMBER
17.     INDEX=INDEX+1
18.     1 FINT=FINT+LIST(I)*A(INDEX)
19.     FINT=FINT*B(NUMBER-1)/C(NUMBER-1)
20.     RETURN
21.     END

```

Q ELT THERM, 1,700610, 71303

450 RIBD LIBRARY UP-DATED THRU 5/5/65									
2.	72301.938E	001.600E-051.200E-040.	0.	0.	.12	1.235			ZN 2 001
3.	72315.875E-010.	0.	0.		.43	2.869			GA 2 002
4.	7232	0.	0.	1.					GE 2 003
5.	73301.390E-039.800E-050.					2.25			ZN 2 004
6.	73312.000E-012.000E-06					.42	.409		GA 2 005
7.	73320.	0.	0.	14.					GE 2 006
8.	74315.555E-033.500E-040.					1.01	3.149		GA 2 007
9.	7432	0.	0.	.5	.4				GE 2 008
10.	75311.389E-034.000E-04					1.43	.58		GA 2 0081
11.	75325.671E-044.000E-04						.139		GE*1 0082
12.	75325.694E-02					.42	.038		GE 2 009
13.	7533	0.	0.	4.5					AS 2 010
14.	76313.704E-04 .001					2.74	1.		GA 2 0101
15.	7632	.001		.2	.5				GE 2 011
16.	76331.104E 000.	0.				1.09	0.404		AS 2 012
17.	7634	0.	0.	85.	.259				SE 2 013
18.	77326.250E-045.400E-030.					0.640	.77	0.096	GE*1 014
19.	77324.583E-013.700E-030.					.64	1.134		GE 2 015
20.	77331.625E 000.	0.				.22	0.022		AS 2 016
21.	77342.083E-04						.16		SE*1 0161
22.	7734	0.	0.	42.					SE 2 017
23.	78328.750E-021.800E-020.					.32	0.300		GE 2 018
24.	78336.319E-032.000E-030.					1.47	0.888		AS 2 019
25.	7834	0.	0.	.41	.875				SE 2 020
26.	79336.250E-034.000E-020.				1.	.90			AS 2 021
27.	79342.708E-030.	0.					0.096		SE*1 022
28.	79342.555E 070.	0.				.04			SE 2 023
29.	79350.	0.	0.	1.140E 010.2540.					BR 2 024
30.	80331.736E-04 .08					2.73	.5		AS 2 0241
31.	8034			.6	.2				SE 2 025
32.	80351.875E-010.	0.					0.085		BR*1 026
33.	80351.250E-020.	0.				.72	.116		BR 2 027
34.	8036	0.	0.	15.					KR 2 028
35.	81333.819E-041.250E-010.					1.66			AS 2 029
36.	81343.958E-028.400E-030.						0.103		SE*1 030
37.	81341.250E-020.	0.				.53			SE 2 031
38.	8135	0.	0.	3.					BR 2 032
39.	8136	0.	0.						KR 2 033
40.	8234	3.800E-050.		5.400E-020.9200.					SE 2 034
41.	82351.471E 00					.14	2.648		BR 2 035
42.	8236	0.	0.	45.	1.				KR 2 036
43.	83347.986E-043.000E-011.200E-01					1.0001	.55	0.200	SE*1 037
44.	83341.736E-021.800E-018.600E-02						.69	1.000	SE 2 038
45.	83351.000E-010.	8.400E-020.		0.	1.000	.34	0.010		BR 2 039
46.	83367.750E-020.	0.	0.	0.	0.		0.041		KR*1 040
47.	8336	0.	0.	180.					KR 2 041
48.	84342.083E-036.100E-023.000E-02					.49			SE 2 042
49.	84354.167E-031.900E-021.000E-02				1.	.71	3.396		BR*1 043
50.	84352.222E-029.200E-014.300E-01					1.23	1.863		BR 2 044
51.	8436	0.	0.	.14	.714				K

89.	91378.338E-042.000E 001.100E 00		2.04 0.4	RB 2	077
90.	91384.042E-010.	0.	.590 .63 0.742	SR 2	078
91.	91393.472E-020.	0.	0.551	Y* 1	079
92.	91395.900E 010.	0.	.61 0.004	Y 2	080
93.	9140 0.	0.		ZR 2	081
94.	92363.471E-052.700E 001.530E 00		2.73	KR 2	082
95.	92375.787E-052.800E 001.540E 00		2.97	RB 2	083
96.	92381.125E-016.000E-017.000E-02		2.26 1.274	SR 2	084
97.	92391.471E-010.	0.	1.44 0.266	Y 2	085
98.	9240 0.	0.		ZR 2	086
99.	93362.315E-051.300E 007.700E-01		3.82	KR 2	087
100.	93376.944E-053.100E 001.820E 00		2.83	RB 2	088
101.	93385.764E-032.000E 001.320E 00		1.40 1.0	SR 2	089
102.	93394.208E-011.000E-016.000E-02		1.18 0.102	Y 2	090
103.	93403.468E 080.	0.		ZR 2	091
104.	93411.351E 030.	0.	.25 .015 0.029	NB*1	092
105.	9341 0.	0.	.909	NB 2	093
106.	94361.620E-056.000E-016.000E-01		2.70	KR 2	094
107.	94373.472E-052.300E 001.500E 00		4.57	RB 2	095
108.	94388.333E-042.900E 001.640E 00		1.46	SR 2	096
109.	94391.389E-027.000E-017.400E-01		2.23 1.	Y 2	097
110.	9440 0.	0.		ZR 2	098
111.	94414.583E-030.	0.	.001 .00050.042	NB*1	099
112.	94417.300E 060.	0.	.16 1.570	NB 2	100
113.	9442 0.	0.		MO 2	101
114.	95361.157E-053.200E 002.900E 00		4.57	KR 2	1011
115.	95372.314E-053.200E 002.900E 00		3.58	RB 2	1012
116.	95385.556E-04		2.72	SR 2	1013
117.	95397.292E-03		2.09	Y 2	102
118.	95406.500E 010.	0.	.02 .12 0.728	ZR 2	103
119.	95413.750E 000.	0.	0.235	NB*1	104
120.	95413.500E 010.	0.	.043 0.760	NB 2	105
121.	9542 0.	0.		MO 2	106
122.	96391.597E-033.200E 002.585E 00		1.55 1.7	Y 2	1061
123.	9640 3.200E 002.585E 00.05			ZR 2	107
124.	96419.583E-015.700E-043.600E-03		.22 2.456	NB 2	108
125.	9642 0.	0.		MO 2	109
126.	97361.157E-052.400E 002.210E 00		4.20	KR 2	1091
127.	97372.315E-052.400E 002.210E 00		4.39	RB 2	1092
128.	97383.472E-05		3.57	SR 2	1093
129.	97394.630E-05		2.31	Y 2	110
130.	97407.083E-011.400E 001.230E 00		.9 .70 0.221	ZR 2	111
131.	97416.944E-040.	0.	0.75	NB*1	112
132.	97415.000E-020.	0.	.47 .673	NB 2	113
133.	9742 0.	0.		MO 2	114
134.	98406.944E-04 .2	1.000E-01	1.99	ZR 2	1141
135.	98413.576E-02 .064	1.000E-01	1. 1.31 2.	NB*1	1142
136.	98411.806E-02 3.	2.845E 00	1.26 1.2	NB 2	115
137.	9842 2.516	2.845E 00.15		MO 2	116
138.	99401.852E-053.050E 003.050E 00		1.55	ZR 2	1161
139.	99411.736E-033.050E 003.050E 00		1.36 .26	NB 2	1162
140.	99422.750E 00		.92 .41 0.124	MO 2	117
141.	99432.500E-010.	0.	.1423	TC*1	118
142.	99437.665E 070.	0.	.084	TC 2	119
143.	9944 0.	0.		RU 2	120
144.	100412.083E-036.300E 007.100E 00		3.39 1.05	NB 2	121
145.	10042 0.	0.		MO 2	122
146.	100431.968E-040.	0.	1.44 .114	TC 2	123
147.	10044 0.	0.		RU 2	124
148.	101416.944E-042.500E 002.950E 00		1.03	NB 2	1241
149.	101421.014E-022.500E 002.950E 00		.42 1.649	MO 2	125
150.	101439.722E-030.	0.	.47 0.362	TC 2	126
151.	10144 0.	0.		RU 2	127
152.	102427.981E-034.200E 005.990E 00		.44 1.3	MO 2	128
153.	102435.787E-050.	0.	1.79	TC 2	129
154.	10244 0.	0.		RU 2	130
155.	103435.787E-041.550E 002.835E 00		1.03	TC 2	1301
156.	103444.000E+011.450E 002.835E 00		.995 .063 0.488	RU 2	131
157.	103453.958E-020.	0.	0.040	RH*1	132
158.	10345 0.	0.	.071	RH 2	133
159.	104421.111E-031.800E 005.930E 00		1.08 1.	MO 2	134
160.	104431.250E-020.	0.	.98 3.1	TC 2	135
161.	10444 0.	0.		RU 2	136
162.	104453.056E-030.	0.	.0013.0002.131	RH*1	137
163.	104454.861E-040.	0.	.98 0.013	RH 2	138
164.	10446 0.	0.		PD 2	139
165.	105424.630E-046.000E-012.600E 00		2.54	MO 2	140
166.	105435.556E-033.000E-011.300E 00		2.09	TC 2	141
167.	105441.846E-010.	0.	.2690.42 .7680	RU 2	142
168.	105453.472E-040.	0.	0.129	RH*1	143
169.	105451.500E 000.	0.	2.100E+04.715 0. .17 0.032	RH 2	144
170.	10546 0.	0.		PD 2	145
171.	106443.650E 023.800E-014.570E 00.1		.01	RU 2	146
172.	106459.167E-02		1. .35 2.627	RH*1	1461
173.	106453.472E-040.	0.	1.40 .222	RH 2	147
174.	10646 0.	0.		PD 2	148
175.	107431.042E-031.600E-013.600E 00		2.71	TC 2	149
176.	107442.917E-034.000E-020.		1.67 .144	RU 2	150
177.	107451.507E-020.	0.	.42 0.328	RH 2	152
178.	107462.555E 090.	0.	.014	PD 2	153
179.	10747 0.	0.	35. 0.0100.	AG 2	154
180.	108443.125E-038.000E-022.100E 00		.54 0.059	RU 2	155

181.	108451.968E-040.	0.			1.74	0.553	RH 2	156
182.	10846	0.	12.2	.016			PD 2	157
183.	108471.825E 030.	0.		.9	1.605		AG*1	158
184.	108471.667E-030.	0.			.62	.017	AG 2	159
185.	10848	0.	10.				CD 2	160
186.	109453.472E-042.800E-021.400E 00	0.		1.	1.16	.31	RH 2	161
187.	109463.333E-030.	0.				0.18	PD*1	162
188.	109465.625E-010.	0.		1.	.36	.0002	PD 2	163
189.	109474.630E-040.	0.				0.088	AG*1	164
190.	10947	0.	92.	.0326			AG 2	165
191.	11046	2.000E-020.	.24	.167			PD 2	166
192.	110472.600E 020.	0.	80.		0.98	.069 2.722	AG*1	167
193.	110472.778E-040.	0.				1.17 0.033	AG 2	168
194.	11048	2.000E-070.	3.1	.0322			CD 2	169
195.	111462.292E-011.000E-031.000E-02	0.		0.32	.048	.676	PD*1	170
196.	111461.528E-021.800E-022.200E-01	0.		1.	.85	.3	PD 2	171
197.	111478.565E-04	0.				.065	AG*1	1711
198.	111477.500E 00	0.				.36 0.025	AG 2	172
199.	111483.403E-020.	0.				0.397	CD*1	173
200.	11148	0.					CD 2	174
201.	112468.750E-011.100E-021.200E-01	0.			.08	0.018	PD 2	175
202.	112471.333E-010.	0.			1.44	0.695	AG 2	176
203.	11248	0.	.03	1.			CD 2	177
204.	113461.042E-031.100E-021.000E-02	0.		1.	1.40		PD 2	178
205.	113478.333E-040.	0.		.1	.08	0.200	AG*1	179
206.	113472.208E-010.	0.			.79	0.030	AG 2	180
207.	113485.110E 030.	0.			0.999.18	.0003	CD*1	181
208.	11348	0.	2.000E 04				CD 2	182
209.	11349	0.	12.	.67			IN 2	1821
210.	114461.667E-031.000E-021.000E-02	0.			.52		PD 2	183
211.	114475.787E-050.	0.			2.02	0.5	AG 2	184
212.	11448	0.	1.24	0.1130.			CD 2	185
213.	114495.000E 01	0.				.192	IN*1	1851
214.	114498.333E-04	0.			.779	.002	IN 2	1852
215.	115465.208E-04 .01	.0038		.28	1.91		PD 2	1853
216.	115472.315E-04	0.		1.	1.49		AG*1	186
217.	115471.458E-02 .0005	3.000E-04		.09	1.21	.2	AG 2	187
218.	115484.300E 010.	0.		1.	.61	0.036	CD*1	188
219.	115482.300E 000.	0.		1.	.32	0.192	CD 2	189
220.	115491.833E-010.	0.		.055	.015	0.317	IN*1	190
221.	11549	0.	203.	.78	.15		IN 2	191
222.	11550	0.					SN 2	192
223.	116463.472E-041.000E-020.	0.			.93		PD 2	193
224.	116471.736E-030.	0.			2.21	1.217	AG 2	194
225.	11648	0.	1.4				CD 2	195
226.	116493.750E-020.	0.		1.	.29	2.495	IN*1	196
227.	116491.620E-040.	0.			1.39		IN 2	197
228.	11650	3.000E-030.	.006	1.			SN 2	198
229.	117477.639E-041.100E-021.100E-02	0.		.5	1.63		AG 2	199
230.	117481.208E-010.	0.		1.	.35	1.6	CD*1	200
231.	117483.472E-020.	0.		1.	.70	0.49	CD 2	201
232.	117497.917E-020.	0.		0.8	.52	.132	IN*1	202
233.	117493.125E-020.	0.			.24	.73	IN 2	203
234.	117501.400E 010.	0.				0.32	SN*1	204
235.	11750	0.					SN 2	205
236.	118483.472E-021.000E-021.000E-03	0.			.27	.2	CD 2	206
237.	118495.903E-050.	0.			1.85	0.244	IN 2	207
238.	11850	0.	.01	1.			SN 2	208
239.	119481.875E-03 .01	4.000E-03		.99	1.56		CD*1	209
240.	119486.507E-03	0.		1.	1.44	.1	CD 2	210
241.	119491.250E-02	0.		.96	1.04	0.066	IN*1	2101
242.	119491.389E-03	0.		.1	.61	.84	IN 2	211
243.	119502.500E+02	0.				.089	SN*1	212
244.	11950	0.					SN 2	213
245.	120495.787E-040.500E-020.500E-03	0.			.88	3.2	IN 2	2131
246.	12050	0.500E-020.500E-03.141		.007			SN 2	214
247.	121482.431E-032.500E-042.500E-03	0.			1.73	0.85	CD 2	2141
248.	121492.153E-032.500E-042.500E-03	0.		1.	1.58		IN*1	2142
249.	121493.472E-04 .0005	.0053			1.20	.94	IN 2	2143
250.	121509.125E+030.500E-030.500E-02	0.		1.	.15	.01	SN*1	215
251.	121501.125E 001.400E-023.000E-02	0.			.11		SN 2	216
252.	12151	0.	6.06	.01			SB 2	217
253.	122498.681E-050.650E-020.500E-03	0.			2.68	2.14	IN 2	2171
254.	12250	0.650E-020.500E-03.201		.005			SN 2	218
255.	122512.847E-030.	0.				0.136	SB*1	219
256.	122512.750E 000.	0.			.58	.405	SB 2	220
257.	12252	0.	3.	.333			TE 2	221
258.	123491.157E-040.600E-030.500E-03	0.			1.39	1.1	IN 2	2211
259.	123501.250E+020.600E-030.500E-03	0.		1.	.52	0.022	SN*1	222
260.	123502.778E-021.400E-020.	0.			.46	0.160	SN 2	223
261.	12351	0.	3.345	.0135			SB 2	224
262.	123521.040E+02	0.				.248	TE*1	2241
263.	12352	0.	400.				TE 2	225
264.	12450	2.000E-020.	.104	.962			SN 2	226
265.	124511.458E-020.	0.		.982	.97	0.427	SB*1	227
266.	124516.020E+010.	0.	2000.		.38	1.899	SB 2	228
267.	12452	0.	7.	.714			TE 2	229
268.	125506.736E-031.100E-021.000E-03	0.		1.	.79	0.361	SN*1	230
269.	125509.400E 001.200E-027.100E-02	0.			.92	0.094	SN 2	231
270.	125519.855E 020.	0.	20.	.26	.096	.415	SB 2	232
271.	125525.800E 010.	0.				0.145	TE*1	233
272.	12552	0.	1.5				TE 2	234

273.	126503.650E+071.000E-012.000E-01	1.	.12 .09	SN 2	235
274.	126511.319E-020.	0.	.01 .00780.117	SB*1	236
275.	126511.250E 010.	0.	.74 1.780	SB 2	237
276.	12652 0.	0.		TE 2	238
277.	127508.750E-022.400E-013.900E-01	1.	1.34 1.	SN 2	239
278.	127513.900E 001.000E-021.000E-02	.2	.38 0.505	SB 2	240
279.	127521.050E 020.	0.	.015 .00410.088	TE*1	241
280.	127523.875E-010.	0.	.23 0.004	TE 2	242
281.	12753 0.	0.		I 2	243
282.	128504.306E-023.700E-014.000E-01	.03	3.85 3.27	SN 2	244
283.	128514.000E-015.000E-020.	1.	.35 3.6	SB*1	245
284.	128516.944E-038.000E-021.000E-01		1.20 1.7	SB 2	246
285.	12852 0.	0.		TE 2	247
286.	128531.736E-023.000E-052.000E-04	.108	.77 0.093	I 2	248
287.	12854 0.	0.		XE 2	249
288.	129511.792E-011.000E 001.000E 00	.36	.542 .916	SB 2	250
289.	129523.300E 010.	0.	0.106	TE*1	251
290.	129524.653E-020.	0.	.45 .238	TE 2	252
291.	129535.840E 090.	0.	.050 .040	I 2	253
292.	12954 0.	0.		XE 2	254
293.	130501.806E-032.000E 002.000E 00		4.23	SN 2	255
294.	130512.292E-020.	0.	1.72 1.0	SB 2	256
295.	13052 0.	0.		TE 2	257
296.	130535.208E-01 .0005 .0052 18.	.13	.27 2.123	I 2	258
297.	13054 0.	0.		XE 2	259
298.	131511.597E-022.700E 003.100E 00	.15	1.33	SB 2	260
299.	131521.200E 004.000E-021.300E 01	.783	.16 1.468	TE*1	261
300.	131521.736E-021.600E-015.400E-01		.74 0.376	TE 2	262
301.	131538.050E 00 .2	0.	.007 .18 0.399	I 2	263
302.	131541.200E 010.	1.000E-02		XE*1	264
303.	13154 0.	0.		XE 2	265
304.	132511.455E-033.400E 005.100E 00		2.60	SB 2	266
305.	132523.250E 001.000E 00 .18		.055 0.280	TE 2	267
306.	132539.583E-020.	0.	.49 2.249	I 2	268
307.	13254 0.	0.		XE 2	269
308.	133512.847E-033.800E 003.000E 00	.72	1.72	SB 2	270
309.	133523.472E-021.100E 001.000E 00	.87	.49 1.613	TE*1	271
310.	133521.389E-031.100E 001.000E 00		.97 0.6	TE 2	272
311.	133538.750E-015.000E-012.000E-01	.02	.46 .53	I 2	273
312.	133542.300E 000.	0.		XE*1	274
313.	133545.270E 000.	1.700E 00190.	.10 .082	XE 2	275
314.	13355 0.	0.		CS 2	276
315.	134515.556E-043.000E 003.500E 00	.085	3.45 0.200	SB 2	277
316.	134522.917E-023.700E 004.000E 00		.52 .1	TE 2	278
317.	134533.681E-029.000E-010.		.78 2.5	I 2	279
318.	13454 0.	0.		XE 2	280
319.	134551.208E-010.	0.	.01 .00170.153	CS*1	281
320.	134557.665E 020.	0.	.16 1.588	CS 2	282
321.	13456 0.	0.		BA 2	283
322.	135512.778E-04 4.3 1.1		3.16	SB 2	2831
323.	135526.944E-04		2.68	TE 2	284
324.	135532.792E-01 1.9 5.700E 00	.27	.32 1.642	I 2	285
325.	135541.111E-020.	0.		XE*1	286
326.	135543.833E-013.000E-01 .47 2.900E+06		.31 0.261	XE 2	287
327.	135557.300E 000.	8.7	.055	CS 2	288
328.	135561.208E 000.	0.		BA*1	289
329.	13556 0.	0.		BA 2	290
330.	136539.606E-043.100E 002.100E 00		1.68 3.237	I 2	291
331.	13654 3.36 4.520E 00.15			XE 2	292
332.	136551.300E 016.800E-031.100E-01		.11 2.466	CS 2	293
333.	13656 0.	0.		BA 2	294
334.	137532.778E-044.900E 005.100E 00	1.01	2.19 .71	I 2	295
335.	137542.708E-031.300E 001.530E 00		1.48 1.	XE 2	296
336.	137551.095E+040.	0.	.92 .20	CS 2	297
337.	137561.806E-030.	0.		BA*1	298
338.	13756 0.	0.		BA 2	299
339.	138537.292E-053.400E 003.700E 00		3.10 1.5	I 2	300
340.	138549.722E-032.100E 002.210E 00		.96 0.420	XE 2	301
341.	138552.236E-023.000E-014.000E-01		1.10 2.153	CS 2	302
342.	13856 0.	0.		BA 2	303
343.	139532.314E-051.800E 001.700E 00		2.40	I 2	304
344.	139544.745E-042.900E 002.800E 00		2.00 .4	XE 2	305
345.	139556.597E-031.200E 001.230E 00		1.74 0.232	CS 2	306
346.	139565.764E-021.000E-011.400E-014.		.91 .053	BA 2	307
347.	13957 0.	0.		LA 2	308
348.	140541.852E-043.700E 003.300E 00		1.09	XE 2	309
349.	140557.639E-042.300E 002.100E 00		2.66 .59	CS 2	310
350.	140561.280E 013.000E-010.	20.	.30 .222	BA 2	311
351.	140571.675E 000.	2.000E-01	.48 2.468	LA 2	312
352.	14058 0.	0.		CE 2	313
353.	141541.968E-051.800E 001.100E 00	.6	2.36	XE 2	314
354.	141552.778E-042.900E 003.700E 00		1.72	CS 2	315
355.	141561.250E-021.200E 009.000E-01		1.14 .46	BA 2	316
356.	141571.625E-011.000E-010.		.96 .0272	LA 2	317
357.	141583.250E 010.	0.	.15 0.101	CE 2	318
358.	14159 0.	0.		PR 2	319
359.	142541.736E-053.500E-013.100E-01		1.70	XE 2	320
360.	142552.662E-053.050E 002.870E 00		2.98	CS 2	321
361.	142567.639E-032.200E 001.830E 00		.78 .112	BA 2	322
362.	142575.833E-023.000E-010.		1.42 1.164	LA 2	323
363.	14258 0.	0.		CE 2	324
364.	142598.000E-010.	0.	.83 .063	PR 2	325

[illegible]

457.	2211	950.
458.	2411	0.63
459.	3114	0.11
460.	3215	14.3
461.	3516	87.1
462.	3617	1.2 E8
463.	3817	0.026
464.	4118	0.0763
465.	4219	.52
466.	4520	164.
467.	4720	4.9
468.	4621	85.
469.	4721	3.43
470.	4821	1.83
471.	4823	16.1
472.	5124	27.8
473.	5225	5.55
474.	5425	300.
475.	5625	.11
476.	5526	1100.
477.	5926	45.1
478.	5727	270.
479.	5827	0.38
480.	5827	72.
481.	6027	1.9 E3
482.	5928	2.9 E7
483.	6328	2.9 E4
484.	6528	.11
485.	6429	.53
486.	6530	245.
487.	6930	.58
488.	6930	.036
489.	7132	12.
490.	7333	76.
491.	7433	17.5
492.	7534	127.
493.	8538	.049
494.	8538	65.
495.	9643	.036
496.	9643	4.3
497.	9743	.25
498.	9743	3.7 E6
499.	9744	2.8
500.	10346	17.
501.	10547	40.
502.	10948	475.
503.	11349	.073
504.	11350	112.
505.	12653	13.3
506.	13155	10.
507.	13156	11.6
508.	15364	236.
509.	16960	9.4
510.	17168	0.31
511.	17069	127.
512.	17169	694.
513.	17570	4.1
514.	17771	6.8
515.	18172	46.
516.	18273	112.
517.	10174	140.
518.	18574	74.
519.	18774	1.
520.	18375	73.
521.	18875	3.79
522.	18775	1.8E13
523.	18875	.71
524.	18576	95.
525.	19176	.58
526.	19176	16.
527.	19376	1.3
528.	19077	12.
529.	19277	74.5
530.	19477	.79
531.	19178	3.
532.	19378	3.4
533.	19378	1.8 E5
534.	19778	.058
535.	19778	.75
536.	19679	5.6
537.	19879	2.7
538.	19979	3.15
539.	19780	1.
540.	19780	2.7
541.	20380	45.8
542.	20081	1.13
543.	20181	3.
544.	20281	12.
545.	20481	1.1 E3
546.	20382	2.17
547.	21082	7.1 E3
548.	21282	.44

NA	2
NA	2
SI	2
P	2
S	2
CL	2
CL	2
AR	2
K	2
CA	2
CA	2
SC	2
SC	2
SC	2
V	2
CR	2
MN	2
MN	2
MN	2
FE	2
FE	2
CO	2
CO*1	
CO	2
CO	2
NI	2
NI	2
NI	2
CU	2
ZN	2
ZN*1	
ZN	2
GE	2
AS	2
AS	2
SE	2
SR*1	
SR	2
TC*1	
TC	2
TC*1	
TC	2
RU	2
PD	2
AG	2
CD	2
IN*1	
SN	2
I	2
CS	2
DA	2
GD	2
ER	2
ER	2
TM	2
TM	2
YB	2
LU	2
HF	2
TA	2
W	2
W	2
W	2
RF	2
RE	2
RE	2
RE	2
OS	2
OS*1	
OS	2
OS	2
IR	2
IR	2
PT	2
PT*1	
PI	2
PT*1	
PT	2
AU	2
AU	2
AU	2
HG*1	
HG	2
HG	2
TL	2
TL	2
TL	2
TL	2
PB	2
PB	2
PB	2

549.	206A3	6.4
550.	207A3	2.9 E3
551.	210A3	5.0
552.	212A3	.042
553.	210A4	138.4
554.	211A5	.3
555.	223A8	11.7
556.	224A8	3.64
557.	226A8	5.9 E5
558.	228A8	2.4 E3
559.	227A9	8.0 E3
560.	228A9	.26
561.	227A0	18.4
562.	228A0	700.
563.	230A0	2.9 E7
564.	231A0	1.07
565.	232A0	5.1E12
566.	234A0	24.1
567.	230A1	17.7
568.	231A1	1.3 E7
569.	233A1	27.4
570.	230A2	20.8
571.	232A2	2.7 E4
572.	233A2	5.9 E7
573.	234A2	9.1 E7
574.	235A2	2.6E11
575.	236A2	8.7 E9
576.	238A2	1.7E12
577.	237A3	8.0 E8
578.	239A3	2.33
579.	238A4	3.3 E4
580.	239A4	8.9 E6
581.	240A4	2.4 E6
582.	241A4	4.8 E3
583.	242A4	1.4 E8
584.	241A5	1.7 E5
585.	242A5	5.6 E4
586.	242A5	0.677
587.	243A5	2.9 E6
588.	242A6	162.5
589.	243A6	1.3 E4
590.	244A6	6.7 E3
591.	245A6	7.3 E6
592.	246A6	2.4 E6
593.	249A7	290.
594.	249A8	1.31 E5
595.	250A8	3.7 E3
596.	252A8	803.
597.		

END CUR

BI	2
BI	2
BI	2
BI	2
PO	2
AT	2
RA	2
RA	2
RA	2
RA	2
AC	2
AC	2
TH	2
TH	2
TH	2
TH	2
TH	2
TH	2
PA	2
PA	2
PA	2
U	2
U	2
U	2
U	2
U	2
U	2
U	2
NP	2
NP	2
PU	2
PU	2
PU	2
PU	2
PU	2
AM	2
AM*1	
AM	2
AM	2
AM	2
CM	2
CM	2
CM	2
CM	2
CM	2
BK	2
CF	2
CF	2
CF	2

ELT EAST-1.700E10, 71362									
FTR EAST FISSION LIBRARY AS EVALUATED BY LARRY O'DELL 10/11/68									
1.	450								
2.	7230	.194+01	.000	.000	.100+00	.000	.000	.122	.151
3.	7231	.587-00	.000	.000	.100+00	.000	.000	.300	.700
4.	7232	.000	.000	.000	.100+00	.000	.000	.000	.000
5.	7230	.139-02	.000	.000	.100+00	.000	.000	.000	.000
6.	7231	.200-00	.000	.000	.100+00	.000	.000	.470	.327
7.	7232	.000	.000	.000	.100+00	.000	.000	.000	.000
8.	7431	.558-02	.000	.000	.100+00	.000	.000	.910	.263
9.	7432	.000	.000	.000	.100+00	.400	.000	.000	.000
10.	7531	.135-02	.000	.000	.100+00	.000	.000	.000	.000
11.	7532	.567-03	.000	.000	.100+00	.000	.000	.000	.139
12.	7532	.568-01	.000	.000	.100+00	.000	.000	.354	.035
13.	7533	.000	.000	.000	.321-00	.000	.000	.000	.000
14.	7631	.370-03	.000	.976-03	.100+00	.000	.000	.000	.252
15.	7632	.000	.000	.000	.100+00	.500	.000	.000	.000
16.	7633	.110+01	.000	.000	.100+00	.000	.000	.041	.465
17.	7634	.000	.000	.000	.100+00	.259	.000	.000	.000
18.	7732	.625-03	.900-02	.164-02	.100+00	.000	.640	.900	.384
19.	7732	.471-00	.900-02	.164-02	.100+00	.000	.000	.523	.302
20.	7733	.163+01	.000	.000	.100+00	.000	.000	.223	.010
21.	7734	.208-03	.000	.000	.100+00	.000	.000	.000	.162
22.	7734	.000	.000	.000	.100+00	.000	.000	.000	.000
23.	7832	.612-01	.356-01	.141-01	.100+00	.000	.000	.300	.260
24.	7833	.632-01	.000	.000	.100+00	.000	.000	.100	.000
25.	7834	.000	.000	.000	.100+00	.875	.000	.000	.000
26.	7933	.625-02	.700-01	.406-01	.100+00	.000	.000	.700	.000
27.	7934	.271-02	.000	.000	.100+00	.000	.000	.000	.096
28.	7934	.237+08	.000	.000	.100+00	.000	.000	.053	.000
29.	7935	.000	.000	.000	.100+00	.254	.000	.000	.000
30.	8033	.174-03	.109+00	.877-01	.100+00	.000	.000	.800	.435
31.	8034	.000	.100-02	.000	.100+00	.200	.000	.000	.000
32.	8035	.100-00	.000	.000	.100+00	.000	.000	.000	.013
33.	8035	.125-01	.000	.000	.100+00	.000	.000	.600	.075
34.	8036	.000	.000	.000	.100+00	.000	.000	.000	.000
35.	8133	.382-03	.173-00	.157-00	.100+00	.000	.000	.170	.000
36.	8134	.396-01	.200-02	.000	.100+00	.000	.000	.000	.008
37.	8134	.129-01	.200-02	.000	.100+00	.000	.000	.517	.006
38.	8135	.000	.000	.000	.126-00	.000	.000	.000	.000
39.	8136	.000	.000	.000	.100+00	.000	.000	.000	.000
40.	8234	.000	.259-00	.261-00	.100+00	.920	.000	.000	.000
41.	8235	.147+01	.000	.000	.100+00	.000	.000	.135	.556
42.	8236	.000	.000	.000	.100+00	.000	.000	.000	.000
43.	8334	.799-03	.167-00	.194-00	.100+00	.000	.000	.102	.004
44.	8334	.170-01	.167-00	.194-00	.100+00	.000	.000	.500	.339
45.	8335	.958-01	.600-02	.000	.100+00	.000	.000	.313	.007
46.	8336	.792-01	.000	.000	.523-00	.000	.000	.000	.042
47.	8336	.000	.000	.000	.523-00	.000	.000	.000	.000
48.	8434	.229-02	.479-00	.018-00	.100+00	.000	.000	.467	.000
49.	8435	.417-02	.206-01	.206-02	.100+00	.000	.000	.502	.357
50.	8435	.221-01	.206-01	.206-02	.100+00	.000	.000	.000	.722
51.	8436	.000	.000	.000	.200-01	.714	.000	.000	.000
52.	8533	.498-05	.110+00	.549-00	.100+00	.000	.000	.900	.000
53.	8534	.451-03	.413-00	.307-00	.100+00	.000	.000	.200	.250
54.	8535	.206-02	.162-00	.503-01	.100+00	.000	.830	.000	.000
55.	8536	.183-00	.300-02	.000	.100+00	.000	.810	.233	.181
56.	8536	.387+04	.300-02	.000	.250-01	.000	.000	.232	.211
57.	8537	.000	.000	.000	.241-00	.100	.000	.000	.000
58.	8634	.185-03	.442-00	.115+01	.108+00	.000	.000	.000	.000
59.	8635	.625-03	.415-00	.180-00	.108+00	.000	.000	.000	.055
60.	8636	.000	.420-01	.201-02	.500-02	.000	.000	.000	.000
61.	8637	.722-03	.000	.000	.108+00	.000	.000	.000	.560
62.	8637	.187+02	.000	.000	.108+00	.000	.000	.562	.095
63.	8638	.000	.000	.000	.710-01	.000	.000	.000	.000
64.	8734	.185-03	.264-00	.127+01	.117+00	.000	.000	.000	.000
65.	8735	.637-03	.691-00	.422-00	.117+00	.000	.000	.410	.780
66.	8736	.528-01	.191-00	.345-01	.117+00	.000	.000	.010	.374
67.	8737	.000	.500-02	.000	.200-01	.000	.000	.000	.000
68.	8738	.117+00	.000	.000	.500-01	.000	.000	.000	.310
69.	8738	.000	.000	.000	.500-01	.000	.000	.000	.000
70.	8835	.185-03	.849-00	.193+01	.126-00	.000	.000	.900	.000
71.	8836	.117+00	.543-00	.171-00	.126-00	.000	.000	.312	.070
72.	8837	.125-01	.300-01	.000	.126-00	.000	.000	.610	.569
73.	8838	.000	.000	.000	.700-02	.000	.000	.000	.000
74.	8935	.521-04	.567-00	.227+01	.136-00	.000	.000	.370	.000
75.	8936	.222-02	.103+01	.519-00	.136-00	.000	.000	.310	.000
76.	8937	.107-01	.195-00	.141-01	.136-00	.000	.000	.510	.225
77.	8938	.520+02	.300-02	.000	.136-00	.000	.000	.407	.000
78.	8939	.185-03	.000	.000	.180-01	.000	.000	.000	.901
79.	8939	.000	.000	.000	.180-01	.000	.000	.000	.000
80.	9035	.185-04	.244-00	.201+01	.147-00	.000	.000	.000	.000
81.	9036	.382-03	.122+01	.934-00	.147-00	.000	.000	.070	.407
82.	9037	.201-02	.627-00	.154-00	.147-00	.000	.000	.900	.188
83.	9038	.102+05	.300-01	.000	.170-01	.000	.000	.200	.000
84.	9039	.132-00	.000	.000	.147-00	.000	.000	.000	.644
85.	9039	.267+01	.000	.000	.147-00	.000	.000	.719	.000
86.	9040	.000	.000	.000	.700-02	.000	.000	.000	.000
87.	9136	.110-03	.167+01	.330+01	.159-00	.000	.000	.200	.000
88.	9137	.972-02	.673-00	.256-00	.159-00	.000	.000	.500	.000

89.	9137	.833-03	.673-00	.250-00	.159-01-000	.0001.833	.000	RB 2	088
90.	9138	.404-00	.191-00	.574-01	.159-00-000	.620.533	.645	SR 2	089
91.	9139	.347-01	.160-02	.000	.159-00-000	.000.000	.551	Y* 1	090
92.	9139	.590+02	.100-02	.000	.159-00-000	.000.512	.004	Y 2	091
93.	9140	-.000	.060	.000	.510-01-000	.000.000	.000	ZR 2	092
94.	9236	.347-04	.546-00	.323+01	.172-00-000	.0001.670	.000	KR 2	093
95.	9237	.613-04	.159+01	.107+01	.172-00-000	.0002.500	.000	RB 2	094
96.	9238	.108+00	.687-00	.581-01	.172-00-000	.000.2151	.258	SR 2	095
97.	9239	.150-00	.220-01	.000	.172-01-000	.0001.200	.229	Y 2	096
98.	9240	-.000	.000	.000	.240-01-000	.000.000	.000	ZR 2	097
99.	9336	.231-04	.184-00	.272+01	.186-00-000	.0002.680	.000	KR 2	098
100.	9337	.648-04	.177+01	.173+01	.186-00-000	.0002.000	.000	RB 2	099
101.	9338	.569-02	.169+01	.369-00	.186-00-000	.0001.4001	.000	SR 2	100
102.	9339	.417-00	.157-00	.000	.186-00-000	.0001.030	.093	Y 2	101
103.	9340	.347+09	.100-02	.000	.103+00-000	.0001.000	.021	ZR 2	102
104.	9341	.135+04	.000	.000	.560-01-000	.000.000	.029	NB*1	103
105.	9341	-.000	.000	.000	.560-01.909	.000.000	.000	NB 2	104
106.	9436	.162-04	.400-01	.167+01	.201-00-000	.0002.140	.000	KR 2	105
107.	9437	.336-04	.113+01	.238+01	.201-00-000	.0003.000	.000	RB 2	106
108.	9438	.903-03	.275+01	.967-00	.201-00-000	.000.900	.000	SR 2	107
109.	9439	.132-01	.700-00	.261-01	.201-00-000	.0001.800	.724	Y 2	108
110.	9440	-.000	.160-01	.000	.240-01-000	.000.000	.000	ZR 2	109
111.	9441	.453-02	.060	.000	.201-00-000	.001.001	.002	NB*1	110
112.	9441	.730+07	.060	.000	.201-00-000	.000.1571	.573	NB 2	111
113.	9442	.000	.060	.000	.201-00-000	.000.000	.000	MO 2	112
114.	9536	.116-04	.500-02	.147+01	.217-00-000	.0003.100	.000	KR 2	113
115.	9537	.231-04	.422-00	.259+01	.217-00-000	.0002.500	.000	RB 2	114
116.	9538	.556-03	.284+01	.136+01	.217-00-000	.0001.800	.000	SR 2	115
117.	9539	.729-02	.191+01	.110+00	.217-00-000	.0001.200	.000	Y 2	116
118.	9540	.650+02	.123+00	.000	.217-00-000	.020.127	.733	ZR 2	117
119.	9541	.375+01	.000	.000	.217-00-000	.000.000	.235	NB*1	118
120.	9541	.350+02	.060	.000	.217-00-000	.000.053	.745	NB 2	119
121.	9542	-.000	.000	.000	.242-00-000	.000.000	.000	MO 2	120
122.	9539	.160-02	.472+01	.575+01	.235-00-000	.0001.1571	.600	Y 2	121
123.	9540	-.000	.526-00	.000	.360-01-000	.000.000	.000	ZR 2	122
124.	9541	.958-00	.000	.000	.235-00-000	.000.2402	.400	NB 2	123
125.	9542	.000	.000	.000	.590-01-000	.000.000	.000	MO 2	124
126.	9736	.116-04	.000	.297-00	.254-00-000	.0003.500	.000	KR 2	125
127.	9737	.231-04	.120-01	.181+01	.254-00-000	.0002.900	.000	RB 2	126
128.	9738	.347-04	.647-00	.270+01	.254-00-000	.0002.300	.000	SR 2	127
129.	9739	.463-04	.303+01	.110+01	.254-00-000	.0001.700	.000	Y 2	128
130.	9740	.708-00	.144+01	.297-01	.254-00-000	.960.640	.081	ZR 2	129
131.	9741	.694-03	.320-01	.000	.254-00-000	.000.000	.747	NB*1	130
132.	9741	.514-01	.320-01	.000	.254-00-000	.000.422	.665	NB 2	131
133.	9742	-.000	.000	.000	.262-00-000	.000.000	.000	MO 2	132
134.	9740	.694-03	.515+01	.600+01	.274-00-000	.00000001	.0001.000	ZR 2	133
135.	9741	.354-01	.171-00	.000	.274-00-000	.0001.000	.9272	NB*1	134
136.	9741	.116-04	.171-00	.000	.274-00-000	.0001.3001	.300	NB 2	135
137.	9742	-.000	.300-02	.000	.990-01-000	.000.000	.000	MO 2	136
138.	9940	.405-03	.459+01	.597+01	.296-00-000	.000.900	.900	ZR 2	137
139.	9941	.167-02	.117+01	.122+00	.296-00-000	.0001.000	.636	NB 2	138
140.	9942	.279+01	.350-01	.000	.296-00-000	.900.376	.137	MO 2	139
141.	9943	.250-00	.000	.000	.734-00-000	.000.000	.141	TC*1	140
142.	9943	.767+00	.000	.000	.734-00-000	.000.000	.000	TC 2	141
143.	9944	-.000	.060	.000	.296-00-000	.000.000	.000	RII 2	142
144.	10041	.205-02	.570+01	.620+01	.320-00-000	.0002.033	.922	NB 2	143
145.	10042	-.000	.219-00	.444-02	.740-01-000	.000.000	.000	MO 2	144
146.	10043	.197-03	.000	.000	.320-00-000	.0001.123	.561	TC 2	145
147.	10044	.000	.000	.000	.272-00-000	.000.000	.000	RU 2	146
148.	10141	.694-03	.531+01	.602+01	.319-00-000	.000.625	.625	NB 2	147
149.	10142	.104-01	.603-00	.317-00	.319-00-000	.000.5001	.147	MO 2	148
150.	10143	.972-02	.110-01	.000	.319-00-000	.000.400	.300	TC 2	149
151.	10144	-.000	.000	.000	.626-00-000	.000.000	.000	RU 2	150
152.	10242	.758-02	.594+01	.635+01	.318-00-000	.000.3101	.300	MO 2	151
153.	10243	.579-04	.600-01	.000	.318-00-000	.0001.100	.000	TC 2	152
154.	10244	-.000	.000	.000	.259-00-000	.000.000	.000	RU 2	153
155.	10343	.375-03	.500+01	.638+01	.318-00-000	.000.700	.044	TC 2	154
156.	10344	.396+02	.280-02	.000	.318-00-000	.994.000	.473	KU 2	155
157.	10345	.396-01	.000	.000	.357-00-000	.000.000	.040	RH*1	156
158.	10345	-.000	.000	.000	.357-00.071	.000.000	.000	RH 2	157
159.	10442	.111-04	.450+01	.603+01	.317-00-000	.0001.1571	.300	MO 2	158
160.	10443	.125-01	.296-00	.296-00	.317-00-000	.000.9402	.980	TC 2	159
161.	10444	-.000	.140-01	.000	.114+00-000	.000.000	.000	RU 2	160
162.	10445	.306-02	.000	.000	.317-00-000	.001.000	.127	RH*1	161
163.	10445	.486-03	.000	.000	.317-00-000	.000.800	.013	RH 2	162
164.	10446	.000	.000	.000	.322-00-000	.000.000	.000	PU 2	163
165.	10442	.463-03	.371+01	.431+01	.316-00-000	.0001.600	.000	MO 2	164
166.	10443	.694-02	.107+01	.426-00	.316-00-000	.0001.000	.000	TC 2	165
167.	10544	.183-00	.700-01	.000	.316-00-000	.206.333	.726	RU 2	166
168.	10545	.347-03	.000	.000	.316-00-000	.000.000	.130	RH*1	167
169.	10545	.150+01	.000	.000	.316-00.715	.000.103	.032	RH 2	168
170.	10546	-.000	.000	.000	.933-00-000	.000.000	.000	PU 2	169
171.	10644	.365+03	.470+01	.453+01	.315-00-000	.000.013	.000	RU 2	170
172.	10645	.517-01	.190-02	.000	.315-00-000	.0001.3262	.374	RH*1	171
173.	10645	.347-03	.100-02	.000	.315-00-000	.0001.074	.200	RH 2	172
174.	10646	-.000	.000	.000	.166-00-000	.000.000	.000	PU 2	173
175.	10743	.336-03	.314+01	.129+01	.314-00-000	.0001.500	.000	TC 2	174
176.	10744	.292-02	.546-00	.193-02	.314-00-000	.0001.027	.218	RU 2	175
177.	10745	.153-01	.120-01	.000	.314-00-000	.000.400	.290	RH 2	176
178.	10746	.256+10	.000	.000	.312-00-000	.000.012	.000	PU 2	177
179.	10747	-.000	.000	.000	.458-00.010	.000.000	.000	AG 2	178
180.	10844	.313-02	.260+01	.627-00	.313-00-000	.000.415	.046	RU 2	179

181.	10845	.197-03	.440-01	.000	.313-00	.000	.0001.261	.374	RH 2	180
182.	10846	.000	.000	.000	.150-00	.016	.000	.000	PD 2	181
183.	10847	.162+04	.000	.000	.313-00	.000	.900	.0001.593	AG*1	182
184.	10847	.167-02	.000	.000	.313-00	.000	.000	.000	AG 2	183
185.	10848	.000	.000	.000	.313-00	.000	.000	.000	CU 2	184
186.	10845	.347-03	.167+01	.310-00	.313-00	.0001.000	.800	.400	RH 2	185
187.	10746	.332-02	.000	.000	.313-00	.000	.000	.000	PD*1	186
188.	10746	.583-00	.000	.000	.313-00	.0001.000	.312	.000	PU 2	187
189.	10747	.451-03	.000	.000	.861-00	.000	.000	.000	AG*1	188
190.	10947	.000	.000	.000	.661-00	.033	.000	.000	AG 2	189
191.	11046	.000	.620-00	.140-00	.163-00	.167	.000	.000	PU 2	190
192.	11047	.260+03	.000	.000	.312-00	.000	.980	.0002.000	AG*1	191
193.	11047	.278-03	.000	.000	.312-00	.000	.000	.917	AG 2	192
194.	11048	.000	.000	.000	.152-00	.032	.000	.000	CU 2	193
195.	11146	.229-00	.250-00	.368-01	.311-00	.000	.320	.138	PD*1	194
196.	11146	.153-01	.250-00	.368-01	.311-00	.0001.000	.713	.107	PD 2	195
197.	11147	.868-03	.000	.000	.311-00	.000	.000	.000	AG*1	196
198.	11147	.750+01	.000	.000	.311-00	.000	.000	.339	AG 2	197
199.	11148	.340-01	.000	.000	.624-00	.000	.000	.000	CD*1	198
200.	11148	.000	.000	.000	.624-00	.000	.000	.000	CU 2	199
201.	11246	.875-00	.150-00	.446-01	.310-00	.000	.000	.070	PD 2	200
202.	11247	.133-00	.000	.000	.310-00	.000	.0001.000	.486	AG 2	201
203.	11248	.000	.000	.000	.172-00	.000	.000	.000	CD 2	202
204.	11346	.972-03	.109+00	.473-01	.309-00	.0001.0001.000	.100	.000	PD 2	203
205.	11347	.833-03	.000	.000	.309-00	.000	.100	.067	AG*1	204
206.	11347	.221-00	.000	.000	.309-00	.000	.000	.525	AG 2	205
207.	11348	.511+04	.000	.000	.514-00	.000	.999	.200	CU*1	206
208.	11348	.000	.000	.000	.514-00	.000	.000	.000	CD 2	207
209.	11349	.000	.000	.000	.309-00	.670	.000	.000	IN 2	208
210.	11446	.167-02	.970-01	.381-01	.309-00	.000	.000	.467	PD 2	209
211.	11447	.579-04	.200-02	.000	.309-00	.000	.0001.370	.570	AG 2	210
212.	11448	.000	.000	.000	.152-00	.113	.000	.000	CU 2	211
213.	11449	.500+02	.000	.000	.309-00	.000	.000	.000	IN*1	212
214.	11449	.833-03	.000	.000	.309-00	.000	.000	.602	IN 2	213
215.	11546	.521-03	.870-01	.357-01	.308-00	.000	.5001.500	.000	PD 2	214
216.	11547	.231-03	.400-02	.300-04	.308-00	.0001.600	.2001.256		AG*1	215
217.	11547	.146-01	.400-02	.300-04	.308-00	.000	.090	.5841.063	AG 2	216
218.	11548	.430+02	.000	.000	.308-00	.0001.000	.528	.031	CU*1	217
219.	11548	.233+01	.000	.000	.308-00	.0001.000	.300	.193	CU 2	218
220.	11549	.189-00	.000	.000	.546-00	.000	.055	.917	IN*1	219
221.	11549	.000	.000	.000	.548-00	.780	.000	.163	IN 2	220
222.	11550	.000	.000	.000	.308-00	.000	.000	.000	SN 2	221
223.	11646	.347-03	.700-01	.339-01	.307-00	.000	.000	.000	PL 2	222
224.	11647	.174-02	.190-01	.690-03	.307-00	.000	.0001.6071.220		AG 2	223
225.	11648	.000	.100-02	.000	.307-00	.000	.000	.000	CU 2	224
226.	11649	.375-01	.000	.000	.307-00	.0001.000	.2032.494		IN*1	225
227.	11649	.162-03	.000	.000	.307-00	.000	.0001.007	.017	IN 2	226
228.	11650	.000	.000	.000	.240-011.000	.000	.000	.000	SN 2	227
229.	11747	.764-03	.840-01	.339-01	.306-00	.000	.500	.950	AG 2	228
230.	11748	.142-00	.100-02	.000	.306-00	.0001.000	.3081.359		CU*1	229
231.	11748	.100+00	.100-02	.000	.306-00	.0001.000	.573	.573	CU 2	230
232.	11749	.804-01	.000	.000	.306-00	.000	.530	.449	IN*1	231
233.	11749	.306-01	.000	.000	.306-00	.000	.000	.247	IN 2	232
234.	11750	.140+02	.000	.000	.281-00	.000	.000	.321	SN*1	233
235.	11750	.000	.000	.000	.281-00	.000	.000	.000	SN 2	234
236.	11848	.347-01	.860-01	.336-01	.305-00	.000	.000	.257	CU 2	235
237.	11849	.579-04	.000	.000	.305-00	.000	.0001.339	.185	IN 2	236
238.	11850	.000	.000	.000	.520-011.000	.000	.000	.000	SN 2	237
239.	11948	.187-02	.420-01	.169-01	.305-00	.0001.000	.8001.167		CU*1	238
240.	11948	.694-02	.420-01	.169-01	.305-00	.0001.000	.933	.170	CD 2	239
241.	11949	.125-01	.000	.000	.305-00	.000	.960	.846	IN*1	240
242.	11949	.160-02	.000	.000	.305-00	.000	.000	.533	IN 2	241
243.	11950	.250+03	.000	.000	.231-00	.000	.000	.090	SN*1	242
244.	11950	.000	.000	.000	.231-00	.000	.000	.000	SN 2	243
245.	12049	.579-03	.890-01	.339-01	.304-00	.000	.000	.6262.831	IN 2	244
246.	12050	.000	.000	.000	.330-01	.007	.000	.000	SN 2	245
247.	12148	.243-02	.820-01	.257-01	.303-00	.000	.0001.1201.120		CD 2	246
248.	12149	.215-02	.500-02	.429-02	.303-00	.0001.0001.233	.000	.000	IN*1	247
249.	12149	.347-03	.500-02	.429-02	.303-00	.000	.000	.957	IN 2	248
250.	12150	.730+04	.000	.350-03	.303-00	.0001.000	.140	.010	SN*1	249
251.	12150	.113+01	.000	.350-03	.303-00	.000	.000	.127	SN 2	250
252.	12151	.000	.000	.000	.303-00	.010	.000	.000	SB 2	251
253.	12249	.860-04	.960-01	.368-01	.302-00	.000	.0001.6571.700		IN 2	252
254.	12250	.000	.100-02	.500-04	.302-00	.005	.000	.000	SN 2	253
255.	12251	.285-02	.000	.000	.302-00	.000	.000	.043	SN*1	254
256.	12251	.275+01	.000	.000	.302-00	.000	.000	.561	SB 2	255
257.	12252	.000	.000	.000	.302-00	.333	.000	.000	TE 2	256
258.	12349	.116-03	.105+00	.308-01	.301-00	.000	.0001.1001.100		IN 2	257
259.	12350	.125+03	.200-02	.000	.301-00	.0001.000	.466	.022	SN*1	258
260.	12350	.285-01	.200-02	.000	.301-00	.000	.000	.473	SN 2	259
261.	12351	.000	.000	.000	.426-00	.013	.000	.000	SB 2	260
262.	12352	.104+03	.000	.000	.301-00	.000	.000	.134	TE*1	261
263.	12352	.000	.000	.000	.301-00	.000	.000	.000	TE 2	262
264.	12450	.000	.129-00	.425-01	.301-00	.962	.000	.000	SN 2	263
265.	12451	.146-01	.000	.000	.301-00	.000	.200	.625	SN*1	264
266.	12451	.600+02	.000	.000	.301-00	.000	.000	.3522.117	SB 2	265
267.	12452	.000	.000	.000	.129-00	.714	.000	.000	TE 2	266
268.	12550	.674-02	.820-01	.237-01	.300-00	.0001.000	.700	.326	SN*1	267
269.	12550	.940+01	.820-01	.237-01	.300-00	.000	.000	.095	SN 2	268
270.	12551	.966+03	.200-02	.720-03	.442-00	.000	.210	.132	SB 2	269
271.	12552	.580+02	.000	.000	.256-00	.000	.000	.145	TE*1	270
272.	12552	.000	.000	.000	.256-00	.000	.000	.000	TE 2	271

273.	12650	.365+06	.266-00	.563-01	.540-01-.0001.000	.109	.077	SN 2	272
274.	12651	.132-01	.700-02	.570-03	.275-00-.000	.010	.006	SB*1	273
275.	12651	.125+02	.700-02	.570-03	.275-00-.000	.000	.6331.870	SB 2	274
276.	12652	.000	.000	.000	.390-01 .100	.000	.000	TE 2	275
277.	12750	.875-01	.367-00	.107+00	.252-00-.000	.0001.610	.000	SN 2	276
278.	12751	.388+01	.620-01	.947-02	.252-00-.000	.160	.400	SB 2	277
279.	12752	.109+03	.000	.000	.252-00-.000	.008	.002	TE*1	278
280.	12752	.392-00	.000	.000	.252-00-.000	.000	.230	TE 2	279
281.	12753	.000	.000	.000	.520-00-.000	.000	.000	I 2	280
282.	12850	.396-01	.481-00	.309-00	.230-00-.000	.030	.246	SN 2	281
283.	12851	.400-00	.109+00	.242-01	.230-00-.0001.000	.3333.300		SB*1	282
284.	12851	.694-02	.109+00	.242-01	.230-00-.000	.000	.8661.808	SB 2	283
285.	12852	.000	.110-01	.540-03	.170-01 .108	.000	.000	TE 2	284
286.	12853	.174-01	.000	.000	.230-00-.000	.000	.634	I 2	285
287.	12854	.000	.000	.000	.230-00-.000	.000	.000	XE 2	286
288.	12951	.179-00	.113+01	.876-00	.211-00-.000	.000	.605	SB 2	287
289.	12952	.340+02	.350-01	.220-02	.211-00-.000	.360	.000	TE*1	288
290.	12952	.479-01	.350-01	.220-02	.211-00-.000	.000	.456	TE 2	289
291.	12953	.584+10	.100-02	.000	.441-00-.000	.000	.050	I 2	290
292.	12954	.000	.000	.000	.211-00-.000	.000	.000	XE 2	291
293.	13050	.181-02	.552-00	.134+01	.193-00-.000-.000	.500	.500	SN 2	292
294.	13051	.694-02	.141+01	.448-00	.193-00-.000	.0001.413	.760	SB 2	293
295.	13052	.000	.380-00	.366-01	.800-02 .130-.000	.000	.000	TE 2	294
296.	13053	.521-00	.000	.000	.193-00-.000-.000	.2762.133		I 2	295
297.	13054	.000	.000	.000	.860-01-.000	.000	.000	XE 2	296
298.	13151	.174-01	.220+01	.295+01	.177-00-.000	.150	.827	SB 2	297
299.	13152	.125+01	.566-00	.716-01	.177-00-.000	.820	.1531.191	TE*1	298
300.	13152	.174-01	.566-00	.716-01	.177-00-.000	.000	.636	TE 2	299
301.	13153	.805+01	.640-01	.000	.177-00-.000	.006	.181	I 2	300
302.	13154	.118+02	.000	.000	.291-00-.000	.000	.000	XE*1	301
303.	13154	.000	.000	.000	.291-00-.000	.000	.000	XE 2	302
304.	13251	.146-02	.189+01	.417+01	.162-00-.000	.0002.600	.000	SB 2	303
305.	13252	.325+01	.234+01	.371-00	.162-00-.000	.000	.073	TE 2	304
306.	13253	.958-01	.311-00	.000	.162-00-.000	.000	.4341.992	I 2	305
307.	13254	.000	.300-02	.000	.670-01 .962	.000	.000	XE 2	306
308.	13351	.292-02	.120+01	.431+01	.148-00-.000	.7202.100	.000	SB 2	307
309.	13352	.347-01	.172+01	.494+00	.148-00-.000	.870	.4391.639	TE*1	308
310.	13352	.866-02	.172+01	.494+00	.148-00-.000	.000	.800	TE 2	309
311.	13353	.875-00	.104+01	.266-01	.148-00-.000	.024	.450	I 2	310
312.	13354	.230+01	.140-01	.000	.148-00-.000	.000	.000	XE*1	311
313.	13354	.530+01	.140-01	.000	.148-00-.000	.000	.115	XE 2	312
314.	13355	.000	.000	.000	.438-00 .085	.000	.000	CS 2	313
315.	13451	.556-03	.523-00	.354+01	.136-00-.000	.0002.900	.000	SB 2	314
316.	13452	.292-01	.346+01	.231+01	.136-00-.000	.000	.217	TE 2	315
317.	13453	.368-01	.232+01	.521-00	.136-00-.000	.000	.6001.937	I 2	316
318.	13454	.000	.148-00	.000	.470-01 .962	.000	.000	XE 2	317
319.	13455	.121+00	.000	.000	.136-00-.000	.010	.002	CS*1	318
320.	13455	.766+03	.000	.000	.136-00-.000	.000	.1661.591	CS 2	319
321.	13456	.000	.000	.000	.157-00 .038	.000	.000	BA 2	320
322.	13551	.231-04	.160-00	.154+01	.124+00-.000	.0001.7501.750		SB 2	321
323.	13552	.116-03	.247+01	.273+01	.124+00-.000	.0002.000	.000	TE 2	322
324.	13553	.275-00	.366+01	.142+01	.124+00-.0000.270	.3081.540		I 2	323
325.	13554	.100-01	.272-00	.581-01	.124+00-.000	.000	.000	XE*1	324
326.	13554	.383-00	.272-00	.581-01	.124+00-.000	.000	.302	CS 2	325
327.	13555	.730+09	.700-02	.000	.218-00-.000-.000	.070	.000	XS 2	326
328.	13556	.121+01	.000	.000	.124+00-.000	.000	.000	BA*1	327
329.	13556	.000	.000	.000	.124+00-.000	.000	.000	BA 2	328
330.	13653	.961-03	.520+01	.493+01	.114+00-.000	.0001.1502.480		I 2	329
331.	13654	.000	.169+01	.775-00	.120-01-.000-.000	.000	.000	XE 2	330
332.	13655	.136+02	.000	.856-02	.114+00-.000	.000	.1202.231	CS 2	331
333.	13656	.000	.000	.000	.270-01 .010	.000	.030	BA 2	332
334.	13753	.276-03	.324+01	.441+01	.104+00-.000-.0001.158	.230		I 2	333
335.	13754	.271-02	.325+01	.147+01	.104+00-.000-.0001.330	.150		XE 2	334
336.	13755	.110+05	.351-00	.120+00	.380-01-.000	.950	.192	CS 2	335
337.	13756	.181-02	.100-02	.000	.139-00-.000	.000	.000	BA*1	336
338.	13756	.000	.100-02	.000	.139-00-.000	.000	.000	BA 2	337
339.	13853	.694-04	.129+01	.337+01	.960-01-.000-.0002.560	.000		I 2	338
340.	13854	.110-01	.360+01	.219+01	.960-01-.000-.0001.000	.420		XE 2	339
341.	13855	.222-01	.110+01	.493-00	.960-01-.000	.0001.0802.010		CS 2	340
342.	13856	.000	.290-01	.000	.800-02-.000	.000	.000	BA 2	341
343.	13953	.231-04	.331-00	.212+01	.880-01-.000-.0002.150	.000		I 2	342
344.	13954	.475-03	.266+01	.271+01	.880-01-.000	.0001.700	.359	XE 2	343
345.	13955	.660-02	.215+01	.110+01	.880-01-.000	.0001.300	.230	CS 2	344
346.	13956	.590-01	.167-00	.298-01	.880-01-.000	.000	.764	BA 2	345
347.	13957	.000	.100-02	.000	.320-01-.000	.000	.000	LA 2	346
348.	14054	.185-03	.144+01	.355+01	.800-01-.000	.0001.300	.000	XE 2	347
349.	14055	.764-03	.299+01	.168+01	.800-01-.000	.0001.5251.525		CS 2	348
350.	14056	.128+02	.656-00	.276-00	.800-01-.000	.000	.268	BA 2	349
351.	14057	.167+01	.120-01	.000	.800-01-.000	.000	.4952.110	LA 2	350
352.	14058	.000	.000	.000	.270-01-.000	.000	.000	CE 2	351
353.	14154	.231-04	.510-00	.242+01	.980-01-.000	.0002.000	.000	XE 2	352
354.	14155	.289-03	.297+01	.229+01	.980-01-.000	.0001.600	.000	CS 2	353
355.	14156	.125-01	.177+01	.739-00	.980-01-.000	.000	.930	BA 2	354
356.	14157	.158-00	.990-01	.821-02	.980-01-.000	.000	.725	LA 2	355
357.	14158	.330+02	.000	.000	.980-01-.000	.000	.163	CE 2	356
358.	14159	.000	.000	.000	.920-01-.000	.000	.000	PR 2	357
359.	14254	.174-04	.111+00	.138+01	.120+00-.000	.0001.730	.000	XE 2	358
360.	14255	.266-04	.186+01	.246+01	.120+00-.000	.0002.300	.000	CS 2	359
361.	14256	.764-02	.295+01	.128+01	.120+00-.000	.000	.750	BA 2	360
362.	14257	.590-01	.473-00	.105+00	.120+00-.000	.000	.8301.084	LA 2	361
363.	14258	.000	.600-02	.000	.340-01-.000	.000	.000	CE 2	362
364.	14259	.800-00	.000	.000	.120+00-.000	.000	.697	PR 2	363

365.	14260	.000	.000	.000	.450-01-.000	.000	.000	.000	ND 2	364
366.	14334	.116-04	.150-01	.201-00	.146-00-.000	.000	.330	.000	XE 2	365
367.	14335	.231-04	.2671-00	.263+01	.146-00-.000	.000	.190	.000	CS 2	366
368.	14336	.139-03	.227+01	.175+01	.146-00-.000	.000	.150	.000	BA 2	367
369.	14337	.972-02	.127+01	.395-00	.146-00-.000	.000	.100	.242	LA 2	368
370.	14338	.138+01	.510-01	.000	.146-00-.000	.000	.355	.334	CE 2	369
371.	14339	.138+02	.000	.000	.146-00-.000	.000	.311	.000	PK 2	370
372.	14340	-.000	.000	.000	.241-00-.000	.000	.000	.000	ND 2	371
373.	14434	.116-04	.000	.219-00	.179-00-.000	.000	.190	.000	XE 2	372
374.	14435	.231-04	.149-00	.133+01	.179-00-.000	.000	.650	.000	CS 2	373
375.	14436	.347-04	.172+01	.193+01	.179-00-.000	.000	.130	.000	BA 2	374
376.	14437	.463-04	.211+01	.305-00	.179-00-.000	.000	.760	.000	LA 2	375
377.	14438	.284+03	.245-00	.218-01	.179-00-.000	.000	.037	.029	CE 2	376
378.	14439	.112-01	.230-02	.000	.179-00-.000	.000	.970	.031	PK 2	377
379.	14460	-.000	.000	.000	.700-01-.000	.000	.000	.000	ND 2	378
380.	14558	.202-02	.353+01	.377+01	.219-00-.000	.000	.890	.000	CE 2	379
381.	14559	.250-00	.200-01	.000	.219-00-.000	.000	.620	.016	PK 2	380
382.	14560	-.000	.000	.000	.455-00-.000	.000	.000	.000	ND 2	381
383.	14658	.972-02	.273+01	.300+01	.268-00-.000	.000	.250	.327	CE 2	382
384.	14659	.174-01	.101+00	.000	.268-00-.000	.000	.0101	.120	PK 2	383
385.	14660	-.000	.100-02	.000	.980-01-.000	.000	.000	.000	ND 2	384
386.	14758	.833-03	.193+01	.250+01	.328-00-.000	.000	.150	.000	CE 2	385
387.	14759	.833-02	.315-00	.126-01	.328-00-.000	.000	.713	.757	PK 2	386
388.	14760	.111+02	.600-02	-.000	.328-00-.000	.000	.228	.236	ND 2	387
389.	14761	.950+03	.000	.000	.692-00-.522	.000	.074	.000	PM 2	388
390.	14762	-.000	.000	.000	.261-00-.000	.000	.000	.000	SM 2	389
391.	14858	.486-03	.110+01	.197+01	.401-00-.000	.000	.500	.500	CE 2	390
392.	14859	.135-02	.609-00	.104+00	.401-00-.000	.000	.400	.300	PK 2	391
393.	14860	-.000	.370-01	.000	.179-00-.000	.000	.000	.000	ND 2	392
394.	14861	.540+01	.000	.000	.401-00-.000	.000	.614	.623	PM*1	393
395.	14862	.410+02	.000	.000	.401-00-.000	.000	.1471	.836	PM 2	394
396.	14862	.000	.000	.000	.187-00-.000	.000	.000	.000	SM 2	395
397.	14900	.750-01	.143+01	.174+01	.491-00-.000	.000	.500	.336	ND 2	396
398.	14901	.221+01	.260-02	.000	.491-00-.000	.000	.350	.008	PM 2	397
399.	14962	-.000	.000	.000	.117+01-.000	.000	.000	.000	SM 2	398
400.	15060	-.000	.114+01	.133+01	.274-00-.000	.000	.000	.000	ND 2	399
401.	15062	-.000	.000	.000	.263-00-.000	.000	.000	.000	SM 2	400
402.	15160	.833-02	.833-00	.947-00	.631-00-.000	.000	.640	.910	ND 2	

457.	2211	950.
458.	2411	0.63
459.	3114	0.11
460.	3215	14.3
461.	3516	87.1
462.	3617	1.2 E8
463.	3817	0.026
464.	4118	0.0763
465.	4219	.52
466.	4520	164.
467.	4720	4.9
468.	4621	85.
469.	4721	3.43
470.	4621	1.83
471.	4823	16.1
472.	5124	27.8
473.	5225	5.55
474.	5425	300.
475.	5625	.11
476.	5526	1100.
477.	5926	45.1
478.	5727	270.
479.	5827	0.38
480.	5627	72.
481.	6027	1.9 E3
482.	5928	2.9 E7
483.	6328	2.9 E4
484.	6528	.11
485.	6429	.53
486.	6530	245.
487.	6930	.58
488.	6930	.036
489.	7132	12.
490.	7333	76.
491.	7433	17.5
492.	7534	127.
493.	8538	.049
494.	8538	65.
495.	9643	.036
496.	9643	4.3
497.	9743	.25
498.	9743	3.7 E6
499.	9744	2.8
500.	10346	17.
501.	10547	40.
502.	10948	475.
503.	11349	.073
504.	11350	112.
505.	12653	13.3
506.	13155	10.
507.	13156	11.6
508.	15384	236.
509.	16968	4.4
510.	17168	0.31
511.	17069	127.
512.	17169	694.
513.	17570	4.1
514.	17771	6.8
515.	18172	46.
516.	18273	112.
517.	18174	140.
518.	18574	74.
519.	18774	1.
520.	18375	73.
521.	18675	3.79
522.	18775	1.8E13
523.	18875	.71
524.	18576	95.
525.	19176	.50
526.	19176	16.
527.	19376	1.3
528.	19377	12.
529.	19277	74.5
530.	19477	.79
531.	19178	3.
532.	19378	3.4
533.	19378	1.0 E5
534.	19778	.058
535.	19778	.75
536.	19679	5.6
537.	19879	2.7
538.	19979	3.15
539.	19780	1.
540.	19780	2.7
541.	20380	45.8
542.	20081	1.13
543.	20181	3.
544.	20281	12.
545.	20481	1.1 E3
546.	20382	2.17
547.	21082	7.1 E3
548.	21282	.44

NA	2
NA	2
SI	2
P	2
S	2
CL	2
CL	2
AR	2
K	2
CA	2
CA	2
SC	2
SC	2
SC	2
V	2
CR	2
MN	2
MN	2
MN	2
FE	2
FE	2
CO	2
CO*	1
CO	2
CO	2
NI	2
NI	2
NI	2
CU	2
ZN	2
ZN*	1
ZN	2
GE	2
AS	2
AS	2
SE	2
SR*	1
SR	2
TC*	1
TC	2
TC*	1
TC	2
RU	2
PD	2
AG	2
CD	2
IN*	1
SN	2
I	2
CS	2
BA	2
GU	2
ER	2
ER	2
TM	2
TM	2
YU	2
LU	2
HF	2
TA	2
W	2
W	2
W	2
RE	2
RE	2
RE	2
RE	2
OS	2
OS*	1
OS	2
OS	2
IR	2
IR	2
IR	2
PT	2
PT*	1
PT	2
PT*	1
PT	2
AU	2
AU	2
AU	2
AU	2
HG*	1
HG	2
HG	2
TL	2
TL	2
TL	2
TL	2
PB	2
PB	2
PB	2

549.	20563	6.4
550.	20753	2.9 E3
551.	21063	5.8
552.	21283	.042
553.	21684	136.4
554.	21125	.3
555.	22388	11.7
556.	22488	3.64
557.	22666	5.9 E5
558.	22888	2.4 E3
559.	22789	8.0 E3
560.	22389	.26
561.	22790	18.4
562.	22890	700.
563.	23090	2.9 E7
564.	23190	1.07
565.	23290	5.1E12
566.	23490	24.1
567.	23691	17.7
568.	23191	1.3 E7
569.	23391	27.4
570.	23692	20.8
571.	23292	2.7 E4
572.	23392	5.9 E7
573.	23492	9.1 E7
574.	23592	2.6E11
575.	23692	8.7 E9
576.	23892	1.7E12
577.	23793	8.0 E8
578.	23993	2.33
579.	23894	3.3 E4
580.	23994	6.9 E6
581.	24094	2.4 E6
582.	24194	4.8 E3
583.	24294	1.4 E3
584.	24195	1.7 E5
585.	24295	5.6 E4
586.	24295	0.677
587.	24395	2.9 E6
588.	24296	162.5
589.	24396	1.3 E4
590.	24496	6.7 E3
591.	24596	7.3 E6
592.	24696	2.4 E6
593.	24997	290.
594.	24998	1.31 E5
595.	25098	3.7 E3
596.	25298	603.
597.		

BI	2
BI	2
BI	2
UI	2
PO	2
AT	2
RA	2
RA	2
RA	2
RA	2
AC	2
AC	2
TH	2
TH	2
TH	2
TH	2
TH	2
TH	2
PA	2
PA	2
PA	2
U	2
U	2
U	2
U	2
U	2
U	2
U	2
NP	2
NP	2
PU	2
PU	2
PU	2
PU	2
PU	2
AM	2
AM*	1
AM	2
AM	2
CM	2
CM	2
CM	2
CM	2
CM	2
BK	2
CF	2
CF	2
CF	2

001K
002K
002
003S
004N
005K
006K
007K
007
008S
009K
010K
011K
012S
013U
014S
015K
016S
017K
018U
018
019K
020K
021S
022K
023U
024S
025K
026K
027K
028S
029K
030S
031K
032K
033S
034K
035K
036K
037K
038N
039S
040K
040
041K
042K
043K
043
044K
045K
046S
047N
048K
049K
049
050S
051N
052N
053K
054K

056S
057N
058U
059S
060U
061K
062S
063N
064U

066K
067K
068S
069U
070K
070
071K
072S
073N
074U
074
074
075K

[illegible]

075
076K
077K
078S
079N
080K
080
081K
081
082K
083K
084K
085S
086K
087N
088N
089K
090K
091K
092S
093N
094N
095K
096K
097S
098N
099N
100U
101K
101
102K
103U
104S
105N
106N
107K
108K
108
109S
110K
111K
112S
113N
114N
115N
116U
117K
118U
119K
120S
121U
122S
123K
123
124S
125N
126N
127N
128U
129U
130K
131K
132S
133N
134N
135U
135
136S
137N
138U
139K
140K
141K
142S
143U
144S
145U
146S
147N
148K
148
149K
150S
151N
152U
153S
154U
155K
156K
157S
158N

181.	10443TC2	5.25	.53	.50	1.37	.20	2.2	.30	3.4	.20	4.4	
182.	10444RU2	0										
183.	10445RH1	5	.002	.77	.002	.56	.026	.10	.025	.08	.47	.051
184.	10445RH2	2	.001	1.24	.02	.56						
185.	10446PD2	0										
186.	10542M02	0										
187.	10543TC2	1	1.0	.11								
188.	10544RU2	10	.48	.73	.16	.67	.20	.48	.06	.40	.11	.32
189.	10545RH1	1	.001	1.72	.006	1.35	.015	.96	.023	.87		
190.	10545RH2	1	1.0	.129								
191.	10546PD2	2	.19	.319	.05	.306						
192.	10644RU2	0										
193.	10645RH1	13	.18	1.56	.17	1.22	.12	1.13	.25	1.05	.35	.82
194.	10645RH2	6	.002	1.55	.005	1.13	.015	1.05	.11	.622	.21	.512
195.	10646PD2	0										
196.	10743TC2	0										
197.	10744RU2	5	.04	1.29	.04	1.03	.04	.93	.07	.86	.14	.20
198.	10745RH2	6	.03	.68	.11	.39	.73	.31	.03	.55	.02	.365
199.	10746PD2	0										
200.	10747AG2	0										
201.	10844RU2	1	.28	.165								
202.	10845RH2	5	.22	.62	.10	.51	.43	.434	.05	1.52	.03	2.0
203.	10846PD2	0										
204.	10847AG1	4	.90	.722	.90	.614	.89	.434	.05	.080		
205.	10847AG2	4	.017	.632	.0018	.615	.006	.511	.005	.434		
206.	10848CD2	0										
207.	10945RH2	0										
208.	10946PD1	1	.58	.188								
209.	10946PD2	5	.0001	.64	.0003	.60	.0001	.41	.0001	.31	.0013	.13
210.	10947AG1	1	.05	.088								
211.	10947AG2	0										
212.	11046PD2	0										
213.	11047AG1	9	.11	1.51	.21	1.38	.32	.937	.71	.885	.08	.818
214.	11047AG2	2	.04	.658	.0002	1.47						
215.	11048CD2	0										
216.	11146PD1	6	.051	1.4	.01	.81	.08	.62	.06	.58	.05	.38
217.	11146PD2	4	.05	.38	.13	.60	.01	.80	.08	1.4		
218.	11147AG1	1	1.0	.065								
219.	11147AG2	2	.06	.342	.01	.247						
220.	11148CD1	2	.94	.247	.30	.15						
221.	11148CD2	0										
222.	11246PD2	1	.20	.019								
223.	11247AG2	11	.02	2.55	.03	2.11	.03	1.63	.05	1.4	.41	.617
224.	11248CD2	0										
225.	11346PD2	0										
226.	11347AG1	5	.25	.39	.05	.70	.05	.56	.25	.14	.25	.30
227.	11347AG2	7	.04	1.18	.05	.98	.04	.88	.17	.67	.05	.56
228.	11348CD1	1	.001	.265								
229.	11348CD2	0										
230.	11349IN2	0										
231.	11446PD2	0										
232.	11447AG2	1	1.0	.57								
233.	11448CD2	0										
234.	11449IN1	3	.035	.724	.035	.558	.17	.192				
235.	11449IN2	1	.0017	1.29								
236.	11546PD2	0										
237.	11547AG1	0										
238.	11547AG2	13	.15	2.12	.10	1.89	.10	1.66	.11	1.48	.04	.64
239.	11548CD1	3	.009	1.29	.019	.935	.003	.485				
240.	11548CD2	4	.26	.53	.10	.49	.02	.262	.006	.23		
241.	11549IN1	1	.50	.335								
242.	11549IN2	0										
243.	11550SN2	0										
244.	11646PD2	0										
245.	11647AG2	2	1.0	.52	.21	.70						
246.	11648CD2	0										
247.	11649IN1	8	.20	2.11	.11	1.51	.80	1.29	.53	1.09	.17	.819
248.	11649IN2	3	.0012	.434	.001	.95	.012	1.29				
249.	11650SN2	0										
250.	11747AG2	0										
251.	11748CD1	17	.03	2.32	.15	2.0	.06	1.56	.10	1.43	.08	1.41
252.	11748CD2	13	.11	1.24	.04	1.12	.09	1.07	.10	.88	.04	.715
253.	11749IN1	2	.31	.314	.14	.158						
254.	11749IN2	2	1.0	.565	.87	.158						
255.	11750SN1	1	.87	.158								
256.	11750SN2	0										
257.	11848CD2	0										
258.	11849IN2	1	.15	1.23								
259.	11948CD1	0										
260.	11948CD2	1	.21	.8								
261.	11949IN1	3	.05	.30	.05	.91	.9	.024				

157U
160S
161K
162K
163S
164N
165U
166K
166
167U
168K
169S
170K
171K
171
172K
173S
174N
175K
176K
177K
178S
179K
180K
1812
182K
183K
184S
185N
186K
187K
188K
189S
190S
191K
191
192K
193S
194U
195U
196K
197K
198K
199S
200K
201K
201
202S
203K
204U
205U
205
206K
207S
208S
209K
210U
211S
212K
213K
214N
215N
216K
216
217K
218K
219K
220S
221S
222N
223U
224S
225K
225
226K
227S
228N
229K
229
229
230K
230
231K
232K
233K
234S
235K
236K
237S
238N
239U
240U

[illegible]

[illegible]

323
324K
325K
326K
327K
328S
329K
330S
331K
331
332S
333N
334K
335K
336K
337S
338N
339U
340K
340
341S
342N
343U
344U
345K
346S
347U
348U
349K
350K
350
351S
352N
353N
354U
355K
356K
357S
358N
359N
360U
361K
361
362S
363K
364S
365H
366N
367N
368U
369K
370K
371S
372N
373N
374N
375N
376K
377K
378S
379U
380U
381S
382U
383K
383
384S
385N
386K
387K
388K
389S
390N
391U
392S
393K
393
394K
395S
396K
397K
398S
399S
400S
401K
401
402K
402
403K
404S
405U
406U

457.	15263EU1	6	.011	1.39	.012	1.32	.12	.963	.13	.842	.025	.344	.08	.122
458.	15263EU2	7	.22	1.41	.26	1.11	.15	.965	.14	.78	.27	.344	.08	.245
459.	.37	.122												
460.	15264GD2	0												
461.	15361PM2	2	.50	.12	.50	.18								
462.	15362SM2	3	.006	.64	.28	.103	.054	.07						
463.	15363EU2	0												
464.	15461PM2	0												
465.	15462SM2	0												
466.	15463EU2	8	.37	1.28	.31	1.0	.12	.876	.26	.73	.06	.593	.07	.248
467.	.38	.123	.03	1.6										
468.	15464GD2	0												
469.	15562SM2	2	.04	.246	.73	.104								
470.	15563EU2	2	.20	.105	.32	.087								
471.	15564GD2	0												
472.	15662SM2	5	.03	.291	.05	.25	.20	.204	.10	.166	.30	.088		
473.	15663EU2	10	.05	2.19	.03	2.1	.07	1.97	.16	1.24	.14	1.15	.11	1.07
474.	.09	.812	.06	.723	.07	.646	.08	.089						
475.	15664GD2	0												
476.	15763EU2	7	.06	.623	.05	.477	.27	.413	.14	.37	.05	.32	.27	.064
477.	.05	.055												
478.	15764GD2	0												
479.	15863EU1	7	.06	1.19	.045	1.11	.032	.605	.02	.20	.10	.52	.40	.08
480.	.38	.95												
481.	15864GD2	0												
482.	15963EU2	9	.05	1.5	.11	1.1	.11	.8	.10	.73	.21	.67	.05	.22
483.	.14	.15	.18	.09	.42	.07								
484.	15964GD2	2	.09	.363	.03	.058								
485.	15965TB2	0												
486.	16063LU2	0												
487.	16064GD2	0												
488.	16065TB2	10	.07	1.27	.15	1.18	.31	.966	.31	.879	.30	.299	.06	.197
489.	.12	.087	.02	1.31	.027	1.2	.02	1.11						
490.	16066UY2	0												
491.	16164GD2	6	.66	.361	.25	.315	.08	.284	.11	.102	.03	.53	.03	.26
492.	16165TB2	4	.10	.075	.05	.057	.19	.049	.21	.026				
493.	16166UY2	0												
494.	16264GD2	3	.20	.5	.10	1.0	.10	1.4						
495.	16265TB2	3	.33	.9	.30	.5	.50	.30						
496.	16266UY2	0												
497.	16365TB2	4	.15	.510	.40	.288	.15	.170	.30	.025				
498.	16366UY2	0												
499.	16466DY2	0												
500.	16566UY1	4	.018	.514	.006	.362	.003	.152	.03	.108				
501.	16566DY2	6	.007	.716	.007	.633	.011	.361	.006	.28	.04	.095	.002	1.08
502.	16567H02	0												
503.	16666UY2	3	.005	.426	.005	.372	.12	.002						
504.	16667H02	4	.054	.081	.009	1.38	.002	1.582	.001	1.66				
505.	16668ER2	0												
506.	16768ER1	1	.43	.208										
507.	16768ER2	0												
508.	3 1H 2	0												
509.	7 4BE2	1	.103	.477										
510.	14 6C 2	0												
511.	18 9F 2	1	1.94	.511										
512.	2211NA2	2	1.80	.511	1.0	1.275								
513.	2411NA2	2	1.0	2.75	1.0	1.37								
514.	3114SI2	1	.0007	1.27										
515.	3215P 2	0												
516.	3516S 2	0												
517.	3617CL2	0												
518.	3817CL2	2	.38	1.6	.47	2.17								
519.	4118AR2	1	1.0	1.29										
520.	4219K 2	2	.002	.31	.18	1.524								
521.	4520CA2	0												
522.	4720CA2	3	.05	.49	.05	.815	.74	1.31						
523.	4621SC2	2	1.0	1.12	1.0	.89								
524.	4721SC2	1	.73	.159										
525.	4821SC2	4	.06	.176	1.0	.903	1.0	1.040	1.0	1.314				
526.	4823V 2	5	1.0	.511	.10	.945	1.0	.983	.97	1.312	.03	2.241		
527.	5124GR2	1	.1	.32										
528.	5225MN2	4	.67	.511	.82	.744	.84	.935	1.0	1.434				
529.	5425MN2	1	1.0	.84										
530.	5625MN2	3	.99	.847	.29	1.811	.15	2.11						
531.	5526FE2	1	4.-5	.23										
532.	5926FE2	4	.01	.1433	.44	1.29	.56	1.1	.028	.192				
533.	5727CO2	4	.09	.014	.87	.122	.11	.136	.0014	.692				
534.	5827CO1	0												
535.	5827CO2	4	.30	.511	.99	.81	.014	.865	.006	1.67				
536.	6027CO2	2	1.0	1.17	1.0	1.33								
537.	5928HI2	0												
538.	6328HI2	0												
539.	6528KI2	3	.045	.368	.16	1.115	.25	1.481						
540.	6429CU2	2	.005	1.34	.38	.511								
541.	6530ZN2	2	.5	1.11	.034	.511								
542.	6930ZN1	1	.95	.439										
543.	6930ZN2	0												
544.	7132GE2	0												
545.	7333AS2	1	.09	.054										
546.	7433AS2	3	.59	.511	.61	.596	.14	.635						
547.	7534SE2	7	1.0	.066	.033	.097	.17	.121	.57	.136	.6	.265	.25	.28
548.	.12	.401												

407K
408K
408
409S
410U
411K
412S
413N
414S
415K
415
416S
417K
418K
419S
420K
421K
421
422S
423K
423
424S
425U
425
426S
427K
427
428K
429S
430K
431S
432K
432
433S
434K
435K
436S
437U
438U
439S
440U
441S
442S
443K
444K
445S
446K
447K
448S
449K
450S

549.	8538SR1	2	.14	.15	.35	.231														
550.	8538SP2	1	1.0	.514																
551.	9643TC1	0																		
552.	9643TC2	5	.05	.32	1.0	.778	.84	.81	1.0	.851	.16	1.12								
553.	9743TC1	0																		
554.	9743TC2	0																		
555.	9744RC2	2	.91	.215	.08	.324														
556.	10346PL2	3	.0001	.297	.0006	.362	.0001	.498												
557.	10547AG2	6	.10	.064	.32	.28	.42	.344	.1	.443	.12	.65	.02	1.080						
558.	10948CC2	1	.05	.088																
559.	11349IN1	1	.64	.393																
560.	11350SN2	1	.018	.255																
561.	12653I1	2	.34	.386	.33	.667														
562.	13155CS2	0																		
563.	13156LA2	8	.28	.124	.19	.216	.05	.25	.13	.373	.48	.496	.03	.60						
564.	.0131.048		.008	.924																
565.	15364G02	2	.024	.07	.55	.099														
566.	16368ER2	1	.003	.038																
567.	17168BK2	4	.25	.112	.09	.124	.28	.296	.63	.308										
568.	17669TM2	1	.033	.084																
569.	17169TM2	0																		
570.	17370YE2	3	.019	.114	.037	.283	.06	.396												
571.	17771LU2	2	.028	.113	.061	.208														
572.	18172HF2	3	.48	.133	.13	.346	.81	.482												
573.	18273TA2	8	.42	.066	.14	.1	.07	.152	.08	.222	.34	1.122	.16	1.189						
574.	.27 1.222		.13	.231																
575.	18174W2	3	.01	.006	.001	.136	.001	.152												
576.	18574W2	2																		
577.	16774W2	7	.11	.072	.09	.134	.23	.479	.05	.522	.06	.618	.27	.686						
578.	.04 .773																			
579.	18375RE2	0																		
580.	16675RE2	3	.09	.137	.0003	.632	.0004	.768												
581.	18775RE2	0																		
582.	18675RE2	5	.10	.155	.006	.478	.009	.633	.003	.029	.004	.932								
583.	185760S2	2	.80	.646	.14	.875														

641.	23993NP2	4	.23	.106	.04	.209	.12	.228	.14	.278
642.	23894PU2	1	.0001	.039						
643.	23994PU2	3	.0001	.039	.0002	.052	.0001	.129		
644.	24094PU2	1	2.-5	.65						
645.	24194PU2	11	6-4	.145						
646.	24294PU2	0								
647.	24195AM2	2	.0004	.101	.36	.06				
648.	24295AM1	4	.002	.049	.0004	.087	.0003	.11	.0003	.163
649.	24295AM2	1	.60	.042						
650.	24395AM2	2	.04	.044	.0004	.087	.0003	.11	.0003	.163
651.	24296CM2	5	.0004	.044	4.-5	.1022	5-5	.1583	2-6	.58 3.-5 .89
652.	24396CM2	3	.04	.209	.12	.228	.14	.278		
653.	24496CM2	1	.0002	.043						
654.	24596CM2	2	.05	.13	.14	.173				
655.	24696CM2	0								
656.	24997HK2	0								
657.	24998CF2	2	.16	.333	.72	.388				
658.	25098CF2	0								
659.	25298CF2	0								
660.										
661.		0.4	2.2	2.75	2.35	1.56	1.0	1.0	0.	
662.		.85	.8	.75	.65					
663.		.0006	.143	.754	1.287	1.287	.831	.402	.211	
664.		.0837	.0373	.0204	.0103					
665.		0.0	0.0	0.0	0.0	0.0	.008	.035	.058	
666.		.074	.083	.088	.094	.098	.103	.106	.110	
667.		.691	.229	.179	.159	.134	.106	.084	.0693	
668.		.057	.048	.042	.0390					

ELT FICHER, 1,691022, 45385

1.	453		
2.	ZN 72	1.0000	.0005 T
3.	GA 72	.0000	.0000 F
4.	GE 72	.0000	.0000 F
5.	ZN 73	1.0000	.0005 T
6.	GA 73	.0000	.0000 F
7.	GE 73	.0000	.0000 F
8.	GA 74	.0000	.0000 F
9.	GE 74	.0000	.0000 F
10.	GA 75	.0000	.0000 F
11.	GE* 75	.0000	.0000 F
12.	GE 75	.0000	.0000 F
13.	AS 75	1.0000	.0005 T
14.	GA 76	.0000	.0000 F
15.	GE 76	.0000	.0000 F
16.	AS 76	1.0000	.0005 T
17.	SE 76	1.0000	.0005 T
18.	GE* 77	.0000	.0000 F
19.	GE 77	.0000	.0000 F
20.	AS 77	1.0000	.0005 T
21.	SE* 77	1.0000	.0005 T
22.	SE 77	1.0000	.0005 T
23.	GE 78	.0000	.0000 F
24.	AS 78	1.0000	.0005 T
25.	SE 78	1.0000	.0005 T
26.	AS 79	1.0000	.0005 T
27.	SE* 79	1.0000	.0005 T
28.	SE 79	1.0000	.0005 T
29.	BR 79	1.0000	.0005 T
30.	AS 80	1.0000	.0005 T
31.	SE 80	1.0000	.0005 T
32.	BR* 80	1.0000	.0005 T
33.	BR 80	1.0000	.0005 T
34.	KR 80	1.0000	1.0000 F
35.	AS 81	1.0000	.0005 T
36.	SE* 81	1.0000	.0005 T
37.	SE 81	1.0000	.0005 T
38.	BR 81	1.0000	.0005 T
39.	KR 81	1.0000	1.0000 F
40.	SE 82	1.0000	.0005 T
41.	BR 82	1.0000	.0005 T
42.	KR 82	1.0000	1.0000 F
43.	SE* 83	1.0000	.0005 T
44.	SE 83	1.0000	.0005 T
45.	BR 83	1.0000	.0005 T
46.	KR* 83	1.0000	1.0000 F
47.	KR 83	1.0000	1.0000 F
48.	SE 84	1.0000	.0005 T
49.	BR* 84	1.0000	.0005 T
50.	BR 84	1.0000	.0005 T
51.	KR 84	1.0000	1.0000 F
52.	AS 85	1.0000	.0005 T
53.	SE 85	1.0000	.0005 T
54.	BR 85	1.0000	.0005 T
55.	KR* 85	1.0000	1.0000 F
56.	KR 85	1.0000	1.0000 F
57.	RB 85	1.0000	.0005 T
58.	SE 86	1.0000	.0005 T
59.	BR 86	1.0000	.0005 T
60.	KR 86	1.0000	1.0000 F
61.	RB* 86	1.0000	.0005 T
62.	RB 86	1.0000	.0005 T
63.	SR 86	1.0000	.0005 T
64.	SE 87	1.0000	.0005 T
65.	BR 87	1.0000	.0005 T
66.	KR 87	1.0000	1.0000 F
67.	RB 87	1.0000	.0005 T
68.	SR* 87	1.0000	.0005 T
69.	SR 87	1.0000	.0005 T
70.	BR 88	1.0000	.0005 T
71.	KR 88	1.0000	1.0000 F
72.	RB 88	1.0000	.0005 T
73.	SR 88	1.0000	.0005 T
74.	BR 89	1.0000	.0005 T
75.	KR 89	1.0000	1.0000 F
76.	RJ 89	1.0000	.0005 T
77.	SR 89	1.0000	.0005 T
78.	Y* 89	.0000	.0000 F
79.	Y 89	.0000	.0000 F
80.	BR 90	1.0000	.0005 T
81.	KR 90	1.0000	1.0000 F
82.	RB 90	1.0000	.0005 T
83.	SR 90	1.0000	.0005 T
84.	Y* 90	.0000	.0000 F
85.	Y 90	.0000	.0000 F
86.	ZR 90	.0000	.0000 F
87.	KR 91	1.0000	1.0000 F
88.	RB* 91	1.0000	.0005 T

89.	RH	91	1.0000	.0005 T
90.	SR	91	1.0000	.0005 T
91.	Y*	91	.0000	.0000 F
92.	Y	91	.0000	.0000 F
93.	ZR	91	.0000	.0000 F
94.	KR	92	1.0000	1.0000 F
95.	RB	92	1.0000	.0005 T
96.	SR	92	1.0000	.0005 T
97.	Y	92	.0000	.0000 F
98.	ZR	92	.0000	.0000 F
99.	KR	93	1.0000	1.0000 F
100.	RB	93	1.0000	.0005 T
101.	SR	93	1.0000	.0005 T
102.	Y	93	.0000	.0000 F
103.	ZR	93	.0000	.0000 F
104.	RB*	93	.0000	.0000 F
105.	NB	93	.0000	.0000 F
106.	KR	94	1.0000	1.0000 F
107.	RB	94	1.0000	.0005 T
108.	SR	94	1.0000	.0005 T
109.	Y	94	.0000	.0000 F
110.	ZR	94	.0000	.0000 F
111.	NB*	94	.0000	.0000 F
112.	NB	94	.0000	.0000 F
113.	MO	94	.0000	.0000 F
114.	KR	95	1.0000	1.0000 F
115.	RB	95	1.0000	.0005 T
116.	SR	95	1.0000	.0005 T
117.	Y	95	.0000	.0000 F
118.	ZR	95	.0000	.0000 F
119.	RB*	95	.0000	.0000 F
120.	NB	95	.0000	.0000 F
121.	MO	95	.0000	.0000 F
122.	Y	96	.0000	.0000 F
123.	ZR	96	.0000	.0000 F
124.	NB	96	.0000	.0000 F
125.	MO	96	.0000	.0000 F
126.	KR	97	1.0000	1.0000 F
127.	RB	97	1.0000	.0005 T
128.	SR	97	1.0000	.0005 T
129.	Y	97	.0000	.0000 F
130.	ZR	97	.0000	.0000 F
131.	NB*	97	.0000	.0000 F
132.	NB	97	.0000	.0000 F
133.	MO	97	.0000	.0000 F
134.	ZR	98	.0000	.0000 F
135.	NB*	98	.0000	.0000 F
136.	NB	98	.0000	.0000 F
137.	MO	98	.0000	.0000 F
138.	ZR	99	.0000	.0000 F
139.	NB	99	.0000	.0000 F
140.	MO	99	.0000	.0000 F
141.	TC*	99	.0000	.0000 F
142.	TC	99	.0000	.0000 F
143.	RH	99	1.0000	.0005 T
144.	RB	100	.0000	.0000 F
145.	MO	100	.0000	.0000 F
146.	TC	100	.0000	.0000 F
147.	RU	100	1.0000	.0005 T
148.	NB	101	.0000	.0000 F
149.	MO	101	.0000	.0000 F
150.	TC	101	.0000	.0000 F
151.	RU	101	1.0000	.0005 T
152.	MO	102	.0000	.0000 F
153.	TC	102	.0000	.0000 F
154.	RU	102	1.0000	.0005 T
155.	TC	103	.0000	.0000 F
156.	RU	103	1.0000	.0005 T
157.	RH*	103	.0000	.0000 F
158.	RH	103	.0000	.0000 F
159.	MO	104	.0000	.0000 F
160.	TC	104	.0000	.0000 F
161.	RU	104	1.0000	.0005 T
162.	RH*	104	.0000	.0000 F
163.	RH	104	.0000	.0000 F
164.	PD	104	.0000	.0000 F
165.	MO	105	.0000	.0000 F
166.	TC	105	.0000	.0000 F
167.	RU	105	1.0000	.0005 T
168.	RH*	105	.0000	.0000 F
169.	RH	105	.0000	.0000 F
170.	PD	105	.0000	.0000 F
171.	RU	106	1.0000	.0005 T
172.	RH*	106	.0000	.0000 F
173.	RH	106	.0000	.0000 F
174.	PD	106	.0000	.0000 F
175.	TC	107	.0000	.0000 F
176.	RU	107	1.0000	.0005 T
177.	RH	107	.0000	.0000 F
178.	PD	107	.0000	.0000 F
179.	AG	107	.0000	.0000 F
180.	RU	108	1.0000	.0005 T

161.	RM 108	.0000	.0000 F
162.	PD 108	.0000	.0000 F
163.	AG*108	.0000	.0000 F
164.	AG 108	.0000	.0000 F
165.	CD 108	1.0000	.0005 T
166.	RM 109	.0000	.0000 F
167.	PD*109	.0000	.0000 F
168.	FD 109	.0000	.0000 F
169.	AG*109	.0000	.0000 F
190.	AG 109	.0000	.0000 F
191.	PD 110	.0000	.0000 F
192.	AG*110	.0000	.0000 F
193.	AG 110	.0000	.0000 F
194.	CD 110	1.0000	.0005 T
195.	PD*111	.0000	.0000 F
196.	PD 111	.0000	.0000 F
197.	AG*111	.0000	.0000 F
198.	AG 111	.0000	.0000 F
199.	CD*111	1.0000	.0005 T
200.	CD 111	1.0000	.0005 T
201.	PD 112	.0000	.0000 F
202.	AG 112	.0000	.0000 F
203.	CD 112	1.0000	.0005 T
204.	PD 113	.0000	.0000 F
205.	AG*113	.0000	.0000 F
206.	AG 113	.0000	.0000 F
207.	CD*113	1.0000	.0005 T
208.	CD 113	1.0000	.0005 T
209.	IN 113	.0000	.0000 F
210.	PD 114	.0000	.0000 F
211.	AG 114	.0000	.0000 F
212.	CD 114	1.0000	.0005 T
213.	IN*114	.0000	.0000 F
214.	IN 114	.0000	.0000 F
215.	PD 115	.0000	.0000 F
216.	AG*115	.0000	.0000 F
217.	AG 115	.0000	.0000 F
218.	CD*115	1.0000	.0005 T
219.	CD 115	1.0000	.0005 T
220.	IN*115	.0000	.0000 F
221.	IN 115	.0000	.0000 F
222.	SN 115	.0000	.0000 F
223.	PD 116	.0000	.0000 F
224.	AG 116	.0000	.0000 F
225.	CD 116	1.0000	.0005 T
226.	IN*116	.0000	.0000 F
227.	IN 116	.0000	.0000 F
228.	SN 116	.0000	.0000 F
229.	AG 117	.0000	.0000 F
230.	CD*117	1.0000	.0005 T
231.	CD 117	1.0000	.0005 T
232.	IN*117	.0000	.0000 F
233.	IN 117	.0000	.0000 F
234.	SN*117	.0000	.0000 F
235.	SN 117	.0000	.0000 F
236.	CD 118	1.0000	.0005 T
237.	IN 118	.0000	.0000 F
238.	SN 118	.0000	.0000 F
239.	CD*119	1.0000	.0005 T
240.	CD 119	1.0000	.0005 T
241.	IN*119	.0000	.0000 F
242.	IN 119	.0000	.0000 F
243.	SN*119	.0000	.0000 F
244.	SN 119	.0000	.0000 F
245.	IN 120	.0000	.0000 F
246.	SN 120	.0000	.0000 F
247.	CD 121	1.0000	.0005 T
248.	IN*121	.0000	.0000 F
249.	IN 121	.0000	.0000 F
250.	SN*121	.0000	.0000 F
251.	SN 121	.0000	.0000 F
252.	SB 121	.0000	.0000 F
253.	IN 122	.0000	.0000 F
254.	SN 122	.0000	.0000 F
255.	SB*122	.0000	.0000 F
256.	SB 122	.0000	.0000 F
257.	TE 122	.0000	.0000 F
258.	IN 123	.0000	.0000 F
259.	SN*123	.0000	.0000 F
260.	SN 123	.0000	.0000 F
261.	SB 123	.0000	.0000 F
262.	TE*123	.0000	.0000 F
263.	TE 123	.0000	.0000 F
264.	SN 124	.0000	.0000 F
265.	SB*124	.0000	.0000 F
266.	SB 124	.0000	.0000 F
267.	TE 124	.0000	.0000 F
268.	SN*125	.0000	.0000 F
269.	SN 125	.0000	.0000 F
270.	SB 125	.0000	.0000 F
271.	TE*125	.0000	.0000 F
272.	TE 125	.0000	.0000 F

273.	SN 126	.0000	.0000 F
274.	SU 126	.0000	.0000 F
275.	SB 126	.0000	.0000 F
276.	TE 126	.0000	.0000 F
277.	SN 127	.0000	.0000 F
278.	SB 127	.0000	.0000 F
279.	TE*127	.0000	.0000 F
280.	TE 127	.0000	.0000 F
281.	I 127	1.0000	.0005 T
282.	SN 128	.0000	.0000 F
283.	SB*128	.0000	.0000 F
284.	SB 128	.0000	.0000 F
285.	TE 128	.0000	.0000 F
286.	I 128	1.0000	.0005 T
287.	XE 128	1.0000	1.0000 F
288.	SB 129	.0000	.0000 F
289.	TE*129	.0000	.0000 F
290.	TE 129	.0000	.0000 F
291.	I 129	1.0000	.0005 T
292.	XE 129	1.0000	1.0000 F
293.	SN 130	.0000	.0000 F
294.	SB 130	.0000	.0000 F
295.	TE 130	.0000	.0000 F
296.	I 130	1.0000	.0005 T
297.	XE 130	1.0000	1.0000 F
298.	SB 131	.0000	.0000 F
299.	TE*131	.0000	.0000 F
300.	TE 131	.0000	.0000 F
301.	I 131	1.0000	.0005 T
302.	XE*131	1.0000	1.0000 F
303.	XE 131	1.0000	1.0000 F
304.	SB 132	.0000	.0000 F
305.	TE 132	.0000	.0000 F
306.	I 132	1.0000	.0005 T
307.	XE 132	1.0000	1.0000 F
308.	SB 133	.0000	.0000 F
309.	TE*133	.0000	.0000 F
310.	TE 133	.0000	.0000 F
311.	I 133	1.0000	.0005 T
312.	XE*133	1.0000	1.0000 F
313.	XE 133	1.0000	1.0000 F
314.	CS 133	1.0000	.0005 T
315.	SB 134	.0000	.0000 F
316.	TE 134	.0000	.0000 F
317.	I 134	1.0000	.0005 T
318.	XE 134	1.0000	1.0000 F
319.	CS*134	1.0000	.0005 T
320.	CS 134	1.0000	.0005 T
321.	BA 134	1.0000	.0005 T
322.	SB 135	.0000	.0000 F
323.	TE 135	.0000	.0000 F
324.	I 135	1.0000	.0005 T
325.	XP*135	1.0000	1.0000 F
326.	XE 135	1.0000	1.0000 F
327.	CS 135	1.0000	.0005 T
328.	BA*135	1.0000	.0005 T
329.	BA 135	1.0000	.0005 T
330.	I 136	1.0000	.0005 T
331.	XE 136	1.0000	1.0000 F
332.	CS 136	1.0000	.0005 T
333.	BA 136	1.0000	.0005 T
334.	I 137	1.0000	.0005 T
335.	XE 137	1.0000	1.0000 F
336.	CS 137	1.0000	.0005 T
337.	BA*137	1.0000	.0005 T
338.	BA 137	1.0000	.0005 T
339.	I 138	1.0000	.0005 T
340.	XE 138	1.0000	1.0000 F
341.	CS 138	1.0000	.0005 T
342.	BA 138	1.0000	.0005 T
343.	I 139	1.0000	.0005 T
344.	XE 139	1.0000	1.0000 F
345.	CS 139	1.0000	.0005 T
346.	BA 139	1.0000	.0005 T
347.	LA 139	.0000	.0000 F
348.	XE 140	1.0000	1.0000 F
349.	CS 140	1.0000	.0005 T
350.	HA 140	1.0000	.0005 T
351.	LA 140	.0000	.0000 F
352.	CE 140	.0000	.0000 F
353.	XE 141	1.0000	1.0000 F
354.	CS 141	1.0000	.0005 T
355.	BA 141	1.0000	.0005 T
356.	LA 141	.0000	.0000 F
357.	CE 141	.0000	.0000 F
358.	FR 141	1.0000	.0005 T
359.	XE 142	1.0000	1.0000 F
360.	CS 142	1.0000	.0005 T
361.	BA 142	1.0000	.0005 T
362.	LA 142	.0000	.0000 F
363.	CE 142	.0000	.0000 F
364.	FR 142	1.0000	.0005 T

365.	ND 142	.0000	.0000 F
366.	XE 143	1.0000	1.0000 F
367.	CS 143	1.0000	.0005 T
368.	BA 143	1.0000	.0005 T
369.	LA 143	.0000	.0000 F
370.	CE 143	.0000	.0000 F
371.	PR 143	1.0000	.0005 T
372.	ND 143	.0000	.0000 F
373.	XE 144	1.0000	1.0000 F
374.	CS 144	1.0000	.0005 T
375.	BA 144	1.0000	.0005 T
376.	LA 144	.0000	.0000 F
377.	CE 144	.0000	.0000 F
378.	PR 144	1.0000	.0005 T
379.	ND 144	.0000	.0000 F
380.	CE 145	.0000	.0000 F
381.	PR 145	1.0000	.0005 T
382.	ND 145	.0000	.0000 F
383.	CE 146	.0000	.0000 F
384.	PR 146	1.0000	.0005 T
385.	ND 146	.0000	.0000 F
386.	CE 147	.0000	.0000 F
387.	PR 147	1.0000	.0005 T
388.	ND 147	.0000	.0000 F
389.	PM 147	.0000	.0000 F
390.	SM 147	.0000	.0000 F
391.	CE 148	.0000	.0000 F
392.	PR 148	1.0000	.0005 T
393.	ND 148	.0000	.0000 F
394.	PM*148	.0000	.0000 F
395.	PM 148	.0000	.0000 F
396.	SM 148	.0000	.0000 F
397.	ND 149	.0000	.0000 F
398.	PM 149	.0000	.0000 F
399.	SM 149	.0000	.0000 F
400.	ND 150	.0000	.0000 F
401.	SM 150	.0000	.0000 F
402.	ND 151	.0000	.0000 F
403.	PM 151	.0000	.0000 F
404.	SM 151	.0000	.0000 F
405.	EU 151	.0000	.0000 F
406.	PM 152	.0000	.0000 F
407.	SM 152	.0000	.0000 F
408.	EU*152	.0000	.0000 F
409.	EU 152	.0000	.0000 F
410.	GD 152	.0000	.0000 F
411.	PM 153	.0000	.0000 F
412.	SM 153	.0000	.0000 F
413.	EU 153	.0000	.0000 F
414.	PM 154	.0000	.0000 F
415.	SM 154	.0000	.0000 F
416.	EU 154	.0000	.0000 F
417.	GD 154	.0000	.0000 F
418.	SM 155	.0000	.0000 F
419.	EU 155	.0000	.0000 F
420.	GD 155	.0000	.0000 F
421.	SM 156	.0000	.0000 F
422.	EU 156	.0000	.0000 F
423.	GD 156	.0000	.0000 F
424.	EU 157	.0000	.0000 F
425.	GD 157	.0000	.0000 F
426.	EU 158	.0000	.0000 F
427.	GD 158	.0000	.0000 F
428.	EU 159	.0000	.0000 F
429.	GD 159	.0000	.0000 F
430.	TB 159	.0000	.0000 F
431.	EU 160	.0000	.0000 F
432.	GD 160	.0000	.0000 F
433.	TB 160	.0000	.0000 F
434.	DY 160	.0000	.0000 F
435.	GD 161	.0000	.0000 F
436.	TB 161	.0000	.0000 F
437.	DY 161	.0000	.0000 F
438.	GD 162	.0000	.0000 F
439.	TB 162	.0000	.0000 F
440.	DY 162	.0000	.0000 F
441.	TB 163	.0000	.0000 F
442.	DY 163	.0000	.0000 F
443.	DY 164	.0000	.0000 F
444.	DY*165	.0000	.0000 F
445.	DY 165	.0000	.0000 F
446.	HO 165	.0000	.0000 F
447.	DY 166	.0000	.0000 F
448.	HO 166	.0000	.0000 F
449.	ER 166	.0000	.0000 F
450.	ER*167	.0000	.0000 F
451.	ER 167	.0000	.0000 F

9 ELT INDOSE,1,710623, 35005

1. C THIS SUBROUTINE IS FOR FUTURE EXPANSION TO INTERNAL DOSE CALCULATIONS.
2. SUBROUTINE INDOSE
3. C RETURN TO MAIN
4. RETURN
5. END

END CUR