LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

36
10/23/71
MR
C
DR-0506-1
ORNL--5912

DE85 001689

ornl

NOTICE
PORTIONS OF THIS REPORT ARE ILLEGIBLE.
It has been reproduced from the best available copy to permit the broadest possible availability.

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

**PRIMUS: A Computer Code for the
Preparation of Radionuclide
Ingrowth Matrices from
User-Specified Sources**

O. William Hermann
Charles F. Baes, III
Charles W. Miller
Connie L. Begovich
Andrea L. Sjoreen

MASTER

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A06; Microfiche A01

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.

ORNL-5912

Contract No. DE-AC05-84OR21400

Health and Safety Research Division

**PRIMUS: A Computer Code for the Preparation of Radionuclide
Ingrowth Matrices from User-Specified Sources**

O. William Hermann
Computer Services

Charles F. Baes, III
Environmental Sciences Division

Charles W. Miller
Health and Safety Research Division

Connie L. Begovich
Andrea L. Sjoreen
Computer Services

Date Published - October 1984

Research sponsored by the Office of Radiation Programs, U. S.
Environmental Protection Agency under Interagency Agreement
AD-89-F-2-A106 (formerly EPA-78-D-X0394).

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

EDB

iii/iv

DISCLAIMER

Although the research described in this report has been funded by the United States Environmental Protection Agency through Interagency Agreement Number AD-89-F-2-A106 (formerly EPA-78-D-X0394) with Oak Ridge National Laboratory, it has not been subjected to the Agency's required peer and policy review and therefore does not necessarily reflect the views of the Agency and no official endorsement should be inferred.

v/vi

CONTENTS

	Page
DISCLAIMER	iii
LIST OF FIGURES	vii
ACKNOWLEDGEMENT	ix
HIGHLIGHTS	xi
1. INTRODUCTION	1
1.1 THE CRRIS SYSTEM	1
1.2 THE PRIMUS CODE	3
2. BASIC TECHNIQUES APPLIED	5
2.1 LIBRARY CONVERSION	5
2.2 PRODUCTION OF PRIMUS DATA SETS	8
2.3 DECAY CALCULATION	10
2.4 THE NUCLIDE ZAS NUMBER	13
3. PROGRAM FLOW AND SUBROUTINE DESCRIPTION	15
3.1 LIBRARY CONVERSION	15
3.2 PRODUCTION OF PRIMUS DATA SETS	16
3.3 UTILITY SUBROUTINES	17
4. INPUT DATA REQUIREMENTS	21
5. PRIMUS FILES	27
5.1 THE DRALIST-CRRIS/1 DECAY LIBRARY	27
5.2 THE PRIMUS DECAY-CHAIN LIBRARY	27
5.3 PRIMUS DATA SETS	29
6. SAMPLE PROBLEMS	31
7. CONCLUSIONS	33
8. REFERENCES	35
APPENDIX A. PRIMUS Output	7
APPENDIX B. PRIMUS JCL	47
APPENDIX C. PRIMUS Code Listing	53
APPENDIX D. Method of Extending Use of the Code	87

LIST OF FIGURES

Figure	Page
1.1 The six computer codes which comprise the CRRIS.	2

ix/12

ACKNOWLEDGEMENT

We wish to express our deepest gratitude to C. E. Nelson, Office of Radiation Programs, U.S. Environmental Protection Agency, for his ideas, support, discussion, and assistance in this work. M. A. Bell and J. L. Bledsoe provided helpful reviews. We also thank S. A. Norman, B. R. Ferguson, and R. J. Patton for their assistance in producing this document.

HIGHLIGHTS

The computer program, PRIMUS, reads a library of radionuclide branching fractions and half-lives and constructs a decay-chain data library and a problem-specific decay-chain data file. PRIMUS reads the decay data compiled for 496 nuclides from the Evaluated Nuclear Structure Data File (ENSDF). The ease of adding radionuclides to the input library allows the CRRIS system to further expand its comprehensive data base. The decay-chain library produced is input to the ANEMOS code. Also, PRIMUS produces a data set reduced to only the decay chains required in a particular problem, for input to the SUMIT, TERRA, MLSOIL, and ANDROS codes. Air concentrations and deposition rates from the RETADD-II code may be processed by latter CRRIS codes using the PRIMUS decay-chain data file. Source term data may be entered directly to PRIMUS to be read by MLSOIL, TERRA, and ANDROS.

The decay-chain data prepared by PRIMUS is needed for a matrix-operator method that computes either time pendent decay products from an initial concentration generated from a constant input source.

This document describes the input requirements and the output obtained. Also, sections are included on methods, applications, subroutines, and sample cases. A short appendix indicates a method of utilizing PRIMUS and the associated decay subroutines from TERRA or ANDROS for applications to other decay problems.

1. INTRODUCTION

1.1 THE CRRIS

The funding for this research was provided by the U.S. Environmental Protection Agency (EPA), which, upon assuming regulatory authority for nuclear effluents covered by the Clean Air Act of 1977, launched an extensive review and revision of models that simulate transport of radionuclides through environmental pathways to man and calculate resulting health impacts. The results of the first phase of this work were undertaken by the Health and Safety Research Division (HASRD) of the Oak Ridge National Laboratory (ORNL) and are published in two reports: AIRDOS-EPA (Moore et al., 1979) and DARTAB (Begovich et al., 1981). While the AIRDOS-EPA and DARTAB computer codes have provided EPA with an interim methodology for performing nuclear assessments, work has continued at ORNL on an integrated set of computer codes and data bases which serves as a state-of-the-art radiological risk assessment methodology. This Computerized Radiological Risk Investigation System, or CRRIS (Baes and Miller, 1981), is based on a review and evaluation of the AIRDOS-EPA-DARTAB system. This report documents one code of the CRRIS, PRIMUS, a computer code for the Preparation of Radionuclide Ingrowth Matrices from User-specified Sources.

The relationship of PRIMUS to the other codes of the CRRIS is shown in Fig. 1.1. PRIMUS produces the decay matrices needed by all the other CRRIS codes, except RETADD-II (Murphy, Ohr, and Begovich, in press) and DFSOIL (Sjoreen et al., in press). RETADD-II is a long-range trajectory atmospheric dispersion code which calculates radionuclide air concentrations and ground deposition rates on a regional or continental rectangular grid scale. ANEMOS (Miller and Hermann, 1981; Miller, Begovich, and Hermann, in press) is a short-range (≤ 80 km) Gaussian plume atmospheric dispersion code which calculates radionuclide air concentrations and ground deposition rates on a circular grid. Output from ANEMOS may be specific for successive downwind distances or averaged over the sector segments. Multiple runs of ANEMOS within a relatively small area may be summed on a master grid by the SUMIT code

Figure 1.1. The six computer codes which comprise the CRRIS

(Begovich et al., in press). SUMIT may also be used simply to transform the results from an ANEMOS calculational grid to another grid that is more convenient for assessment purposes. TERRA (Baes et al., in press) is used to calculate concentrations of radionuclide chains in four vegetable and produce categories, four cattle feed categories, irrigated and nonirrigated soils, beef, and milk. MISOIL (Sjoreen et al., in press) computes an effective soil concentration to be used to compute the external dose from exposure to contaminated soil. MISOIL computes the soil concentration using dose factors computed by DFSOIL. MISOIL contains a multi-compartment model for leaching radionuclides through soil rather than the single compartment model used in TERRA. The output of MISOIL may be merged with a TERRA output file. ANDROS (Begovich, Ohr, and Chester, in press) receives environmental radionuclide concentrations and combines them with population estimates, food and air usage rates, and dose and risk factors to calculate population and average individual doses and risks from the source term supplied to PRIMUS, ANEMOS, or RETADD-II.

As indicated by Fig. 1.1, various combinations of the CRRIS codes may be used to tailor the assessment to the user's needs. For example, if only inhalation and air immersion doses and risks are to be calculated, then the user may wish to eliminate TERRA and MISOIL from the assessment strategy. If ground deposition rates or air concentrations are known for a given location, then the user may wish to make the assessment by using only PRIMUS, TERRA, MISOIL, and ANDROS.

1.2 THE PRIMUS CODE

The PRIMUS code reorganizes radioactive decay data into the appropriate data arrays for efficient access by other codes in the CRRIS. The primary function of PRIMUS is the construction of the transition matrix for each decay chain of radioactive nuclides in a given source term. These decay chain matrices are required in the time-dependent isotopic growth and decay calculations. These calculations apply the matrix operator method developed by Lee (1976). This method of computing nuclide concentrations and activities has been

implemented by the ANEMOS, TERRA, MLSOIL, and ANDROS codes in producing environmental assessment evaluations.

There are two basic stages in the conversion of the input data to the required form. First, the decay matrices for the entire radionuclide library are produced. This information is stored either in a temporary or a permanent data set. Second, the code selects the data required for a particular problem from the full set of decay matrices. Only the decay chains that contain at least one of the released nuclides are selected. The nuclide precursors to all released nuclides of the selected chain are removed. Decay calculations using this smaller size matrix can significantly reduce computer execution time. The data produced by the second stage are saved for use by other CRRIS codes. Supplementary data, e.g., nuclide activities, also may be included with the problem-specific decay matrices.

The problem-dependent decay matrices also may be produced by ANEMOS. When running ANEMOS, the decay matrices for latter codes in the CRRIS may be produced by either ANEMOS or PRIMUS. If the RETADD-II code is run rather than ANEMOS, then the problem-dependent decay matrices must be produced by PRIMUS.

This documentation includes an explanation of the computations done in PRIMUS and instructions for its use. The computational techniques are discussed in Sect. 2. A brief description of the program flow and subroutines is presented in Sect. 3. Input requirements are given in Sect. 4. The contents of the PRIMUS files and the decay chain library are presented in Sect. 5. Section 6 contains examples of input for two sample problems. The printed results of these problems are shown in Appendix A. A sample JCL file for PRIMUS is shown in Appendix B. The PRIMUS code is listed in Appendix C. The application of this methodology outside of the CRRIS framework, using the PRIMUS file and the CRRIS decay subroutines, is discussed in Appendix D.

2. BASIC TECHNIQUES APPLIED

The two main functions of PRIMUS are the conversion of data in the DRALIST-CRRIS/1 Decay Library to the PRIMUS Decay-Chain Library and the reduction of the converted library to problem-dependent decay chains for a PRIMUS data set. The basic techniques applied in these two functions are described below. The method of computing the concentrations of decay products is also presented here, even though these computations are not done in PRIMUS. The method described in Sect. 2.3 is used in all the CRRIS code except RETADD-II and DFSOIL and is dependent on the form of the decay matrix data set produced by PRIMUS. A definition of the ZAS number follows. The ZAS number is an integer which uniquely identifies any radionuclide and which is used throughout CRRIS.

2.1 LIBRARY CONVERSION

The most important part of the library conversion is the establishment of the separate decay chains. A preliminary processing of input library data is required before the assignment of nuclides to appropriate chains. The seven different input library variables (see Sect. 5.1) are stored in seven arrays in the order given in the library.

Parent-nuclide names, as chemical symbols, are copied to another array and sorted. An array of consecutive integers, starting with 1, is sorted in precisely the same order as the nuclide names. Each integer is the index of the same parent name in the original library array of parent names. This integer can be used as a pointer to any of the other seven arrays to obtain data for a particular nuclide. This integer array will be called a pointer array.

Nuclides are assigned to particular chains beginning with the nuclide parents with no radionuclide decay product. Chains assignment progresses back up the chains. A nuclide is never assigned to more than one chain. If the nuclide branches during decays, the branches are included in the same chain. If two or more parent nuclides decay to the same nuclide, all parent nuclides are included in the same chain.

All nuclides of decay chains are assigned both a chain number and a position in that chain. Note that no stable decay products are included in these chains. Passes are repeated through a segment of coding which considers all nuclides in the original parent-name array, until the chain-assignment process is completed. During the first pass, only the final radionuclides and their parents are assigned chain and position numbers. The parents of final nuclides may branch to two daughters. In repeated passes through the coding, each parent nuclide is assigned the chain number of a daughter. If only one of its two daughters has a chain number, the other daughter is assigned the same chain number. The chain length is incremented by 1 as each nuclide is added to the chain. The positions of the nuclides in the chains are altered later in the final construction of matrices. The chain data are stored in the order of the original parent-name array by applying the appropriate daughters array pointers, previously prepared.

During each pass, the chain number assignment of a parent nuclide is skipped when: (1) the nuclide has no radionuclide daughter; (2) the nuclide already has a chain number; or (3) neither of its daughters has a chain number and either daughter, in turn, has a daughter. Note that an alternative to the third condition, that neither daughter has a daughter, is the condition that causes both the parent and its daughters to be processed during the first pass. Also, since chain numbers are initialized to zero, they remain zero for all nuclides that have neither radioactive parent nor daughter. Finally, after any pass in which at least one daughter of every parent has a chain number, the chain numbering is complete.

After the establishment of all of the decay chains, the radioactive decay matrices are constructed. The matrices are constructed with column and row positions pertaining to parents and daughters, respectively. First, the order of the nuclide positions in the chain are reversed, so that the last nuclide has no radioactive daughters. Next, the positions are reordered to make any daughter of a parent be located at a higher-index position than the parent and to arrange the ZAS numbers in descending order. This process has the effect of automatically producing lower triangular decay matrices. The triangular

matrices can be processed in the decay calculations in less than half the computer time required for nontriangular matrices.

The decay matrix required by PRIMUS is derived from the rate-of-change balance equation describing the radioactive decay for nuclides $Q_i(0)$:

$$\frac{dQ_i(t)}{dt} = \sum_j \lambda_i b_{ij} Q_j(t) - \lambda_i Q_i(t) , \quad (1)$$

where

- $Q_i(t)$ = activity of nuclide i at time t ,
- $Q_j(t)$ = activity of j , a parent of i , at time t ,
- λ_i = total decay constant of i , and
- b_{ij} = branching fraction of j to i .

Equation 1 is a common form (Lee, 1976; Bell, 1973; Hermann and Westfall, 1981) of the activity balance equation. Equation 1 may be written in matrix notation for a chain of M nuclides ($i = 1, M$) as

$$\dot{\underline{Q}}(t) = \underline{A}\underline{Q}(t) + \underline{S} , \quad (2)$$

where

- $\underline{Q}(t)$ = a column vector of elements $q_i(t)$,
- $\underline{A}$ = the $M \times M$ matrix of coefficients to q_i and q_j in Eq. (3),
- a_{ii} = diagonal elements of $\underline{A}$
- a_{ij} = off-diagonal elements for row i and column j of $\underline{A}$, and
- $\underline{S}$ = a column vector of deposition rates of elements s_i .

Then, from Eqs. 1 and 2,

$$a_{ii} = -\lambda_i \text{ , and} \quad (3)$$

$$a_{ij} = \lambda_i b_{ij} \text{ .} \quad (4)$$

The CRRIS codes apply a matrix operator method of performing decay calculations (Lee, 1976). This methodology is discussed in Sect. 2.3. PRIMUS constructs the transition matrix for each chain from the library input and the pointer array following the element definitions in Eqs. 3 and 4. In MLSOIL the decay products are leached downward through five soil layers and so a more complicated set of matrices is produced. The resulting differences in the decay computation are discussed in detail in Sjoreen et al. (in press).

Finally, the PRIMUS Decay-Chain Library is written. Arrays including data such as nuclide symbols, chain numbers, decay constants and nuclide pointers are written first. Then, data describing all of the individual decay chains are written. These data are discussed in detail in Sect. 5.2. The library converted by this method is input to ANEMOS or applied to the next function for producing problem dependent PRIMUS data sets.

2.2 PRODUCTION OF PRIMUS DATA SETS

Decay-chain data for a particular problem are written to a PRIMUS data set (see Sect. 5.3). The options available in constructing this data set are presented in Sect. 4. A set of released nuclides is always entered to PRIMUS when constructing this data set. The only required data are the released nuclide names. Exposure nuclides, which are defined as decay products in the decay chain of a released nuclide, may be added in the input. The released nuclides are always included as exposure nuclides. Extra data, such as activities of the exposure nuclides, may be entered for inclusion on the PRIMUS data set.

A reduced decay chain is produced for each released nuclide. A nuclide chain in the PRIMUS Decay-Chain Library that contains a released

nuclide may contain more nuclides than the exposure nuclides of the particular released nuclide. The extra nuclides include parents of the released nuclide and any others above the parents in the chain. The parents of the released nuclide are removed from the decay chain. All decay products proceeding down the chain to the final radionuclide must be included. It is possible for released nuclides to decay to different daughter products that, in turn, have a common decay product at some point down the chain. Thus, a complete decay chain combination may contain more than one released nuclide. The minimum reduced decay chain for these more complex chains requires the inclusion of only the chain segments that contain exposure nuclides of the specified released nuclides.

To produce these reduced decay matrices PRINUS first reads the decay chain data of the first released nuclide. It determines all released nuclides for the chain and eliminates nuclides that are positioned before the released nuclide nearest to the beginning of the chain. The fact that the matrix is constructed as lower triangular assures that no exposure nuclides are eliminated. Further reduction of the matrix is performed by inspecting data pertaining to each nuclide represented in the matrix. When a nuclide is not released and has no parent, i.e., no positive element in the row for the nuclide, its row and column are removed from the matrix. This technique ultimately produces the minimum reduced decay matrix.

Another reduction to the decay matrix is performed if needed. When a nuclide has a very short half-life and its parent does not, its activity after any reasonable decay time is essentially equal to that of its parent. The phenomenon is known as secular equilibrium. Furthermore, since the daughter has a large decay constant, excessive computer time is required during the decay calculation. In such cases, the short-lived nuclide is eliminated from the decay matrix and its parent is made to decay directly to the daughter of the short-lived nuclide. The following criteria are required for the elimination of a short-lived daughter, where λ_d and λ_p are the decay constants of the daughter and parent, respectively:

1. $\lambda_d \geq 17.329$,
2. $\lambda_d / \lambda_p \geq 2000$, and
3. The short-lived daughter has no other parent.

A scratch data set is written containing the problem-specific reduced matrices and associated data as each decay-chain is processed. Then, the final PRIMUS data set is written, including header records with general information such as the number of chains in the problem, the activity source term, and the decay-chain data.

2.3 DECAY CALCULATION

The decay matrices in the PRIMUS file are used throughout the CRRIS to do time-dependent decay and buildup calculations for a variety of purposes. In ANEMOS, nuclides decay and buildup for the time it takes the airborne plume to reach each downwind distance. SUMIT does not calculate decay, but it does use PRIMUS files to organize multiple ANEMOS runs, which may have different released nuclides, into one master ANEMOS-like file. TERRA and MLSOIL compute nuclide concentrations due to decay and buildup from a continuously depositing source and, if requested, due to decay and buildup over a following period with no deposition. ANDROS computes the concentrations of nuclides due to decay and buildup during the shelf-life of foods.

The change of the nuclide concentration with time, $\dot{Q}(t)$, is calculated in the CRRIS by solving a matrix equation of the form

$$\dot{Q}(t) = g + AQ(t) \quad (5)$$

where

$\underline{Q}(t)$ = vector of concentration for each nuclide q_i ; for

$i = 1, n$; n = chain length;

$\underline{S}$ = source vector of constant input to system of release rate

for nuclide s_i ; $i = 1, n$;

A = matrix in which each a_{ij} represents the transition (or

decay) rate constants of nuclide j to i . Let b_{ij} be the

branching ratio of j to i . Then,

$$a_{ij} = b_{ij}\lambda_j.$$

Note that $\underline{Q}(t)$ is produced as a result of two conditions. There may be an input source term under steady state conditions.

$$\underline{S} = \text{constant vector} > 0 \quad (6)$$

Also, there may be an initial concentration $\underline{Q}(0)$.

$$\underline{Q}(0) > 0 \quad (7)$$

The models in ANEMOS, TERRA, and MISOIL require the accumulated concentration during the time, t , where there was no concentration at the starting time $t = 0$, or:

$$\underline{Q}(0) = 0 \quad (8)$$

Writing Eq. 5 as

$$\dot{\underline{Q}}(t) - A\underline{Q}(t) = \underline{S} \quad (9)$$

and the integral of $-A$ with respect to Q ,

$$- \int A dt = -At \quad (10)$$

gives the following integrating factor that can be used in solving Eq. 9

$$I. F. = e^{-At} \quad (11)$$

Thus, from standard differential equations techniques:

$$\begin{aligned} e^{-At} Q(t) &= \int S e^{-At} dt + K \\ &= -S(A)^{-1} e^{-At} + K \end{aligned} \quad (12)$$

where

K = the constant of integration.

Using the initial condition of Eq. 8 for $t = 0$ in Eq. 12 to solve for K :

$$\begin{aligned} Q(0) &= -S(A)^{-1} + K = 0 \\ K &= S(A)^{-1} \end{aligned} \quad (13)$$

Or, with K substituted into Eq. 12, the solution to Eq. 9 becomes:

$$Q e^{-At} = -S(A)^{-1} e^{-At} + S(A)^{-1} \quad (14)$$

or,

$$Q(t) = S(A)^{-1} (e^{At} - I), \quad (15)$$

where

I = the identity matrix.

C. E. Lee (1976) has shown that decay equations of the form of Eq. 15 may be recast as:

$$\begin{aligned} Q(t) &= (At)^{-1} (e^{At} - I) S t \\ &= D(At) S t \end{aligned} \quad (16)$$

where

$$D(At) = (At)^{-1} (e^{At} - I). \quad (17)$$

In TERRA, MLSOIL, and ANDROS the computation may be done with an initial concentration, $\underline{Q}(0)$, and no source term, $\underline{S}$. Then, from Eq. 9

$$\dot{\underline{Q}}(t) - \underline{A}\underline{Q}(t) = 0 . \quad (18)$$

From Eq. 13, 14, and 18

$$e^{-\underline{A}t}\underline{Q}(t) = \underline{K} . \quad (19)$$

At $t=0$, $\underline{Q}(t)$ is nonzero.

$$\underline{Q}(0) = \underline{K} , \quad (20)$$

and so

$$\underline{Q}(t) = \underline{Q}(0)e^{\underline{A}t} . \quad (21)$$

In either case, we then apply Lee's matrix operator technique based on the calculus of Volterra involving the multiplicative integral, which avoids the usual difficulties of evaluating $e^{\underline{A}t}$ resulting from large values for the elements in the matrix $\underline{A}t$. This calculational model has been successfully applied in the ORIGEN-S code (Hermann and Westfall, 1981), and it has now been incorporated into the CRRIS.

2.4 THE NUCLIDE ZAS NUMBER

The identifier constructed for each nuclide is known as a ZAS number. The ZAS number is a packed integer composed of three separate integers. The definition of the nuclide ZAS number is

$$\text{Nuclide ZAS Number} = Z \times 10,000 + A \times 10 + S , \quad (22)$$

where

- Z = the atomic number, Z, of the nuclide,
- A = the mass number, A, of the nuclide,
- S = 0 for a nuclide in the ground state,
- S = 1 for the lowest metastable state, and
- S $\geq$ 1 for higher metastable state levels.

3. PROGRAM FLOW AND SUBROUTINE DESCRIPTION

The basic function of each PRIMUS subroutine is presented below. Also, the flow of the program is provided in the heading statements, which specify both the calling subroutine (or subroutines) and those that are called by it. The chief subroutines for the two main functions are at the beginnings of Sects. 3.1 and 3.2 and followed by the shorter utility routines.

3.1 LIBRARY CONVERSION

Routine Name: MAIN
Routines Called: MOVEIN, CHAIN1, SETDEC, LIBMAK, EDIT, WRTLIB
Function: Reads user input in NAMELIST format. Prepares variable-dimensioned arrays and calls major subroutines of the code.

Routine Name: CHAIN1
Calling Routine: MAIN
Routines Called: CHAIN2, CHAIN3, CHAIN4, CHAIN5, CHAIN6
Function: Prepares variable-dimensioned arrays and controls flow to the other library conversion subroutines. Calls CHAIN5 and CHAIN6 for each of the decay chains.

Routine Name: CHAIN2
Calling Routine: CHAIN1
Routine Called: SORTP
Function: Reads DRALIST-CRRIS/1 Decay Library. Initializes to zero the decay-chain data arrays used by CHAIN3. Calls SORTP for sorting the library nuclide names. Produces the two pointer arrays for daughter nuclides.

Routine Name: CHAIN3
Calling Routine: CHAIN1
Routine Called: None
Function: Assigns chain numbers and positions in chain for all nuclides. Determines the length of all decay chains. See Sect. 2.1 for a more complete explanation of the technique applied.

Routine Name: CHAIN4
Calling Routine: CHAIN1
Routine Called: None
Function: Converts nuclide half-lives to decay constants. Writes header records but not decay-chain matrices in the output library.

Routine Name: CHAIN5
Calling Routine: CHAIN1
Routine Called: None
Function: Treats one decay chain each time it is called. Reorders nuclides in chain to cause a lower triangular decay matrix to be produced by CHAIN6.

Routine Name: CHAIN6
Calling Routine: CHAIN1
Routine Called: None
Function: Constructs final decay-chain matrices and associated nuclide pointers. Treats one chain each time it is called and writes the appropriate chain data as a record in the output library. Matrix elements are produced as specified by Eqs. 3 and 4 of Sect. 2.1. The method requires source vectors in units of activities. Note that the matrices may be easily converted to be applied with source vectors in units of atoms by two simple changes in CHAIN6. The revision would require the index of DLAN to be changed to I, i.e., DLAN(M) to DLAN(I), in the two statements for computing C(J,K).

3.2 PRODUCTION OF PRIMUS DATA SETS

Routine Name: SETDEC
Calling Routine: MAIN
Routines Called: REDUCE, ISERCH, ID
Function: Reads decay data from PRIMUS Decay-Chain Library. Obtains nuclide pointers from ISERCH. Determines required decay-chain records and reads data. Calls REDUCE to reduce matrix appropriately. Sets pointers to source term data for the exposure nuclides corresponding to each released nuclide. After processing is completed, writes decay chain data on scratch data set.

Routine Name: REDUCE
Calling Routine: SETDEC
Routines Called: None
Function: Eliminates data from each decay matrix for all nuclides not required in the reduced matrix. Treats one matrix each time it is called.

Routine Name: LIBMAK
Calling Routine: MAIN
Routine Called: RELESE
Function: Produces total source term array and associated data. Writes on the PRIMUS data set the header records of sizes, control parameters and source term nuclides and data. Reads decay chain data written by SETDEC. Obtains pointers to released nuclides of a chain by calling RELESE. Then, writes the complete decay chain data on the PRIMUS data set.

Routine Name: RELESE
Calling Routine: LIBMAK
Routine Called: ID
Function: Determines the nuclide ZAS number of all short-lived daughters. For each decay chain, produces an array of pointers to released nuclides in the chain.

3.3 UTILITY SUBROUTINES

Routine Name: EDIT
Calling Routine: MAIN
Routines Called: ID, WRMAT
Functions: Reads the PRIMUS data set. Prints a table containing exposure nuclides of each released nuclide, listing decay constants and branching fractions to daughters of each nuclide. Prints extra data, if that option is activated. Prints separate tables for each released nuclide in the same decay-chain by beginning with the position of the released nuclide. Does not attempt to remove extraneous exposure nuclides from other released nuclides when listing each released nuclide data of the same decay chain.

Routine Name: ID
Calling Routine: LIBMAX, SETDEC, RELEASE
Routine Called: None
Function: Converts the chemical symbols of nuclides to nuclide ZAS numbers. This function subprogram is an entry point in function UNPACK.

Routine Name: ISERCH
Calling Routine: SETDEC
Routine Called: None
Function: Applies a table splitting method to search for a nuclide in the original array of parent nuclides given as chemical symbol names. Since the array has not been sorted, the sorted pointer array is applied. This function subprogram returns the pointer to the unsorted array, which can be used for other input arrays or developed arrays, e.g., the chain number array.

Routine Name: SORTP
Calling Routine: CHAIN2
Routine Called: None
Function: Sorts an array of double precision floating point numbers into ascending order. An array of consecutive integers is reordered precisely the same as the sorted array, producing a pointer array. The pointer array is used to obtain values from data arrays in various parts of PRIMUS and ANEMOS.

Routine Name: MOVEIN
Calling Routine: MAIN
Routine Called: XCHNG
Function: Moves one-dimensional input arrays of data denoted as extra data to two-dimensional arrays indexed for all exposure nuclides input for each released nuclide. Calls XCHNG to reorder exposure nuclides to assure that the released nuclide is first. When requested, prints the released nuclide index, exposure nuclide index, chemical symbol of exposure nuclide, source value input and supplementary vector value input for each nuclide.

4. INPUT DATA REQUIREMENTS

The various input options of PRIMUS pertain to the different requested input and output data sets and the types of data included on an output PRIMUS data set. The purpose of running PRIMUS may be only the conversion of the DRALIST-CRRIS/1 Decay Library to the PRIMUS Decay-Chain Library for input to ANEMOS. Even so, at least one problem-dependent decay chain must be requested. Any nuclide in the input library may be requested.

Either the DRALIST-CRRIS/1 Decay Library (alphanumeric) or the PRIMUS Decay-Chain Library (binary) may be read to produce a PRIMUS data set. Other decay libraries, produced in the same format, could be used as input to PRIMUS. Nuclides may be added to chains or new decay chains may be added to the input decay library. The only requirements are: new parent nuclide may not already exist in the library; new daughter nuclides must be included as a parent nuclides; and no stable isotope is permitted. Also, the data set should be assigned a new name.

A PRIMUS data set (for input to SUMIT, TERRA, MLSOIL, and ANDROS) may be requested for decay chains pertaining to a specified list of the released nuclides of a particular problem. The complete decay chains may be produced with no supplementary data. A PRIMUS data set containing truncated decay chains may be produced by specifying KTRUNC = 1. The truncated chains can be made to match the truncated chains produced by RETADD-II and will reduce the time needed to do the decay calculations. If a decay chain is longer than the specified maximum chain length, MAXTRU, then the chain is truncated to the length MAXTRU. The order of removal of nuclides begins with the last nuclide in the chain and proceeds up the chain. While it is easy to determine the nuclides removed by the truncation of a simple decay chain prior to executing the PRIMUS case, it may be difficult for complex chains. Determine the nuclides that will be retained after truncation by reading the listing of a previous run for the released nuclides without truncation. The first MAXTRU nuclides in the listing identify the chain that would be produced by requesting MAXTRU in the input.

Set IEXTRA = 1 to write additional source data on a PRIMUS data set. Only the nuclides having nonzero data are required as input. You may specify the activities in XINPUT for the corresponding nuclides in EXPSYN. Also, other data pertaining to conditions of various physical problems may be input as two vectors. One vector is used for the list of exposure nuclides and the other for the released nuclides. The extra data are contained on every PRIMUS data set. The data are set to zero, if not entered. The extra data may be entered when producing either full-length or truncated decay chains.

A table listing the decay constants and branching fractions for all nuclides in a decay chain is printed for each released nuclide when producing a PRIMUS data set. The extra data arrays are printed when the option is requested. Input data, the decay-chain matrices produced for the PRIMUS data sets, and the PRIMUS decay-chain library may be printed.

The standard rules of NAMELIST apply to the user input. The input contains title cards, three required NAMELIST blocks of data and one optional NAMELIST block. Data for each block appear below, between the proper NAMELIST name and \$END. The default for each variable is shown in parentheses.

INPUT:

Title Cards - 80 columns. There may be one to 20 nonblank title and comment cards, followed by a blank card.

\$IODATA

LBMODE	= 0, input library, LBBCD, in alphanumeric; written in binary to LBBIN.
	= 1, input library, LBBIN, in binary mode (1).
LBBCD	= unit number of input alphanumeric library (2).
LBBIN	= unit number of binary library (10).
LIBOUT	= unit number of output PRIMUS data set (14).

NSCR = unit number of unformatted scratch data set (12).
IPRINT = unit number of printed output (6).
IEDIT = 0, input is not printed.
 > 0, input edit is printed.
 > 1, decay matrices printed on unit IEDIT (1).
IDLIB = 0, PRIMUS decay-chain library is not printed.
 = 1, decay chains and single nuclides are printed.
 = 2, same as 1, with a listing of the 'raw' library data (0).

END

CONST

NUMREL = number of released nuclides (1).
KTRUNC = 0, chain truncation not wanted.
 = 1, chain truncation requested. This produces a shorter chain matrix in addition to the full-size matrix (0).
MAXTRU = length of truncated chain (8).
IEXTRA = 0, the optional data block ,EXTRA, is omitted.
 = 1, the optional block of source data requested (0).

END

NAMES

RELSYM The **NUMREL** released nuclide chemical symbols, in any order. They should contain no spaces. A hyphen should separate the element symbol and mass number. The metastable state of an isotope is denoted by following the mass number with M. Two or more excited states are specified by M1, M2, etc., with the integers increasing as the excited state level increases. As examples, ²³⁵U is entered as U-235 and the syntax error file , between

lines 2899 and 2900 excited state of ^{133m}Xe is entered as XE-133M (Ir is input as IR-190M2). Only those nuclides contained in the input library are permitted by PRIMUS. The user may refer to Kocher (1981) or RSIC (1981) for list of nuclides in the DEALIST-CRRIS/1 Decay Library. RELSYM is required.

NUMEXP Number of exposure nuclides for which data are given in IEXTRA for each of the released nuclides, in the same order the nuclides appear in RELSYM. NUMREL values are entered. Required if IEXTRA = 1 (0).

IEND

This ends input, if IEXTRA = 0.

IEXTRA

EXPSYM Exposure nuclide chemical symbols as an array. The length is the sum of all values in NUMEXP. After all exposure nuclides resulting from one released nuclide, enter those for the next released nuclide, etc. For example, for a chain r-s-t-u-v and NUMEXP(1) = 3 if there are data for s and t, then the first three entries into EXPSYM are r, s and t. Never omit r; but, if there are no user data for u and v, they may be omitted. You do not need to know the complete chain or the decay order.

The following data represent initial conditions of various physical problems.

19/10

Routine Name: VRMAT
Calling Routine: EDIT
Routine Called: None
Function: Prints the entire decay matrix on the
user specified unit.

Routine Name: WRILIB
Calling Routine: MAIN
Routine Called: None
Function: Prints the PRIMUS decay chain library.

Routine Name: ICHNG
Calling Routine: MOVEIN
Routine Called: None
Function: Repositions released nuclide data to
the first position in data arrays.

XINPUT An array or matrix of initial activity data for the exposure nuclides of each released nuclide. Values apply to the nuclides in the corresponding position in EXPSYM (0.0).

EVECT A vector array of data that may be multiplied times data in XINPUT. Values apply to the nuclides in the corresponding positions in EXPSYM. Since the data are extracted into a vector of the exposure nuclides, any exposure nuclide repeated for other released nuclides should have equal values each time it appears in EVECT (0.0).

RVECT A vector array of data, similar to EVECT, but applying only to released nuclides. Values apply to nuclides in the corresponding positions in RELSYM (0.0).

END

(End of data.) Cases may be stacked by starting again with the title cards. Note that a different unit number (LIBOUT) must be entered for each case to avoid overwriting the output PRIMUS data set.

5. PRIMUS FILES

There is one input file to the PRIMUS code and two output files. These files are discussed in this section.

5.1 THE DRALIST-CRRIS/1 DECAY LIBRARY

PRIMUS reads the decay data derived from ENSDF (Ewbank et al., 1975; Ewbank, 1976; Martin, 1976) by Kocher (1981) for the Radioactive Decay Data Tables and for the production of the DLC-80/DRALIST data package (RSIC, 1981). The data set read by PRIMUS is identified as the DLC-80/DRALIST file in the format for branching fractions. It is called the DRALIST-CRRIS/1 Decay Library in code documentation for the CRRIS.

There are 496 radionuclides in the DRALIST-CRRIS/1 Decay Library. The following is a list of the seven data entries on each line of the library. The nuclide names are given as isotope symbols (e.g., H-3, U-235, AM-242M, IR-190M2):

1. Parent radionuclide name
2. First daughter radionuclide name
3. Decay branching fraction to first daughter
4. Second daughter radionuclide name
5. Decay branching fraction to second daughter
6. Half-life of parent
7. Units for the half-life

A parent nuclide may have no-, one-, or two-daughter radionuclides. The last radionuclide in a decay chain always decays to either one or two stable nuclides. Stable daughters are not included in the library.

5.2 THE PRIMUS DECAY-CHAIN LIBRARY

The PRIMUS Decay-Chain Library is required as input to the ANEMOS code. Also, it may be input to PRIMUS in place of the DRALIST-CRRIS/1 Decay Library when making a PRIMUS data set as described in the next

subsection. Only PRIMUS and ANEMOS use the PRIMUS Decay-Chain Library. The library is an unformatted (binary) file. It includes all of the data contained in the DRALIST-CRIS/1 Decay Library. The library contains the following data:

First record

NTOT number of nuclides
NCHTOT number of decay chains
MAXMX maximum chain length

Second record

BNC(NTOT) nuclide names (REAL*8) in descending alphabetic order

Third record

NP(NTOT) original position of each nuclide in nuclide library

Fourth record

NCHAIN(NTOT) chain number of each nuclide

Fifth record

DLAN(NTOT) decay constant of each nuclide

Sixth record

ANC(NTOT) nuclide name (REAL*8) in original order

Seventh record

BF1(NTOT) branching fraction to first daughter of each nuclide

Eighth record

BF2(NTOT) branching fraction to second daughter of each nuclide

Ninth record

INDAU1(NTOT) index of first daughter of each nuclide

Tenth record

INDAU2(NTOT) index of second daughter of each nuclide

The following record is written for each (NCHTOT) chain

LSQ L*L
L chain length
N chain number (NCHAIN)
IND(L) index of each nuclide in this chain in order
S(LSQ) decay matrix

5.3 PRIMUS DATA SETS

Data derived by the second function of the code, i.e., the production of decay-chain data for a particular problem, are stored on a PRIMUS data set. This data set is identical in format to one of the data sets produced by ANEMOS. Decay data, in the form of a PRIMUS data set, are a required input to SUMIT, TERRA, MLSOIL, and ANDROS. The PRIMUS data set is written in binary.

The PRIMUS data set contains the following information. The variables listed as '- not used -' are set to zero in PRIMUS. They are all used in ANEMOS, except for the arrays WET and DRY. Note that VELD and RVEC are written before their array sizes are. Note also that the ordering set up in IC and JC also holds for the concentration data computed by the various CRRIS modules.

First Record

NUMCHA	number of chains
KTRUNC	input option; 1, if chains are truncated
NPACK	number of nuclides
NAME	- not used -
NUCOUT	number of nuclides

Second Record

IC(NPACK)	position of the nuclide in the exposure nuclide list
JC(NPACK)	chain number of the nuclide

Third Record

CPAK(NPACK)	input concentration of each nuclide (XINPUT)
VELD(NEXP)	input data on exposure nuclides (EVECT)
RVEC(NREL)	input data on released nuclides (RVECT)

Fourth Record

NREL	number of released nuclide.
NEXP	number of exposure nuclides
IDREL(NREL)	ZAS numbers of released nuclides
IDEXP(NEXP)	ZAS numbers of exposure nuclides

If KTRUNC is 1, then the fifth record is written containing the same information as the third, with the chains truncated to the input MAXTRU.

One record of the below data is then written for each chain.

L	length of the chain as written (excludes KSHORT and JSHORT)
LSQ	L^2/L
NCEXP	number of exposure nuclides in this chain
NCREL	number of released nuclides in this chain
KSHORT	position in the chain following which the first short-lived daughter was removed
JSHORT	position in the chain following which the second short-lived daughter was removed
R(LSQ)	decay matrix
JINL(NCREL)	positions of the released nuclides in the chain
NUCLID(L)	ZAS numbers of the nuclides in this chain
LOUT	- not used -
QTOT	- not used -
QIN(L)	- not used -
ITRAN(2,LOUT)	- not used -
VET(L)	- not used -
NUMX	1
DRY(NUMX,L)	- not used -
KID	ZAS number of nuclide KSHORT (or 0)
JID	ZAS number of nuclide JSHORT (or 0)
DLAMK	decay constant of nuclide KSHORT (or 0)
DLAMJ	decay constant of nuclide JSHORT (or 0)
SYMK	literal (AS) name of nuclide KSHORT (or blank)
SYMJ	literal (AS) name of nuclide JSHORT (or blank)

If KTRUNC is 1, then this record is repeated, truncated according to the input MAXTRU.

6. SAMPLE PROBLEMS

The first problem converts the input library (in alphanumeric) to the PRIMUS Decay Chain Library (in binary). Pu-239 is entered as the released nuclide for testing the library. The decay matrix is printed on unit 6.

***** SAMPLE CASE NO. 1, CONVERTING LIBRARY *****

```

$IODATA  LBMODE=0, IEDIT=6, IDLIB=1, $END
$CONST  NUMREL=1, $END
$NAMES  RELSYN= 'PU-239', $END

```

The second problem produces a PRIMUS data set. Eighteen released nuclides, exposure nuclides, an input activity source and vector data are entered. A listing of the input is requested. Since the case has a large number of nuclides and data, the specifications are simply shown below in the input to the case and, also, are shown in the printed output of the case.

***** SAMPLE CASE NO. 2 *****

(FOR PRIMUS DATA SET, WITH SOURCE DATA INPUT.)

```

$IODATA  LBMODE=1, $END
$CONST  NUMREL=18, IEXTRA=1, $END
$NAMES  RELSYN= 'PU-240', 'PU-239', 'U-238', 'PU-238',
'BA-139', 'XE-138', 'CS-137', 'XE-135M',
'CS-134', 'XE-133M', 'I-131', 'SB-125',
'RU-106', 'SR-90', 'KR-88', 'KR-85M', 'AR-41', 'H-3',
NUMEXP=1,2,2,1, 1,2,1,2, 1,2,3*1, 3*2,1,1, $END
$EXTRA  EXPSYM= 'PU-240', 'PU-239', 'U-235', 'U-238', 'TH-234', 'PU-238',
'BA-139', 'XE-138', 'CS-138', 'CS-137', 'XE-135M', 'XE-135', 'CS-134',
'XE-133M', 'XE-133', 'I-131', 'SB-125', 'RU-106', 'SR-90', 'Y-90', 'KR-88',
'RB-88', 'KR-85M', 'KR-85', 'AR-41', 'H-3',
XINPUT=6.3E-5,3.6E-4,3.8E-4,4.5E-4,5.2E-3,9.9E-4,5.0E+2,1.0E+4,7.2E+1,
2.0E-1,3.6E+3,2.3E+2,1.1E-1,1.5E-1,6.5E+2,1.7E-3,3.4E-2,1.3E-1,
6.2E-2,6.2E-2,4.2E+1,3.4E+3,1.1E+3,1.1E+5,3.5E+3,3.1E+3,
EVECT= 26*1.0, RVECT= 18*1.0,
$END

```

The execution time of these two cases together is about 1.2 seconds on the IBM 370/3033; there are less than 800 I/O counts for block sizes of 2000; and less than 700 lines are printed. Selected pages of the printed output are presented on Appendix A.

8. REFERENCES

- [1] Baes, C. F., III, C. W. Miller, D. C. Kocher, A. L. Sjoeren, and B. D. Murphy, in press, An Introduction to CRRIS: A Computerized Radiological Risk Investigation System for Assessing Atmospheric Releases of Radionuclides, ORNL/TM-8573, Oak Ridge National Laboratory.
- [2] Baes, C. F., III, R. D. Sharp, A. L. Sjoeren, and O. W. Hermann, in press, TERRA: A Computer Code for Simulating the Transport of Environmentally Released Radionuclides Through Agriculture, ORNL-5785, Oak Ridge National Laboratory.
- [3] Begovich, C. L., K. F. Eckerman, E. C. Schlatter, S. Y. Ohr, and R. O. Chester, 1981, DARTAB: A Program to Combine Airborne Radionuclide Environmental Exposure Data with Dosimetric and Health Effects Data to Generate Tabulations of Predicted Health Impacts, ORNL-5692, Oak Ridge National Laboratory.
- [4] Begovich, C. L., S. Y. Ohr, A. L. Sjoeren, and R. O. Chester, in press, ANDROS: A Code for Assessment of Nuclide Doses and Risks with Option Selection, ORNL-5889, Oak Ridge National Laboratory.
- [5] Begovich, C. L., N. J. DeBlienck, J. T. Holdeman, Jr., A. L. Sjoeren, and C. W. Miller, in press, SUMIT: A Computer Code to Interpolate and Sum Single Release Atmospheric Model Results onto a Master Grid, ORNL-5914, Oak Ridge National Laboratory.
- [6] Bell, M. J., 1973, ORIGEN-The ORNL Isotope Generation and Depletion Code, ORNL-4628, Oak Ridge National Laboratory.
- [7] Ewbank, W. B., M. R. Schmorak, F. E. Bertrand, M. Feliciano, and D. J. Horen, 1975, Nuclear Structure Data File: A Manual for Preparation of Data Sets, ORNL-5054, Oak Ridge National Laboratory.
- [8] Ewbank, W. B., 1976, "Evaluated Nuclear Structure Data File (ENSDF) for Basic and Applied Research," presented at the Fifth International CODATA Conference, Boulder, Colorado.
- [9] Hermann, O. W., and R. M. Westfall, 1981, "ORIGEN-S: SCALE System Module to Calculate Fuel Depletion, Actinide Transmutation, Fission Product Buildup and Decay, and Associated Radiation Source Terms," Section F7 of SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation, NUREG/CR-0200, Vol. 2.
- [10] Kocher, D. C., 1981, Radioactive Decay Data Tables, U.S. Department of Energy, DOE/TIC-11026.

- [11] Lee, C. E., 1976, The Calculation of Isotopic Mass and Energy Production by a Matrix Operator Method, LA-6483-MS.
- [12] Martin, M. J. (Ed.), 1976, Nuclear Decay Data for Selected Radionuclides, ORNL-5114, Oak Ridge National Laboratory.
- [13] Miller, C. W., and O. W. Hermann, 1981 "ANEMOS: A Code to Calculate Radionuclide Concentrations in Air," Trans. Am. Nucl. Soc., 38: 97-98.
- [14] Miller, C. W., C. L. Begovich, O. W. Hermann, and A. L. Sjoreen, in press, ANEMOS: A Computer Code to Estimate Air Concentrations and Ground Deposition Rates for Atmospheric Nuclides Emitted from Multiple Operating Sources, ORNL-5913, Oak Ridge National Laboratory.
- [15] Moore, R. E., C. F. Baes, III, L. M. McDowell-Boyer, A. P. Watson, F. O. Hoffman, J. C. Pleasant, and C. W. Miller, 1979, AIRDOS-EPA A Computerized Methodology for Estimating Environmental Concentrations and Dose to Man from Airborne Releases of Radionuclides, ORNL-5532, Oak Ridge National Laboratory.
- [16] Murphy, B. D., S. Y. Ohr, and C. L. Begovich, 1984, RETADD-II: A Long Range Atmospheric Trajectory Model with Consistent Treatment of Deposition Loss and Species Growth and Decay, ORNL/CSD-99, Oak Ridge National Laboratory.
- [17] RSIC (Radiation Shielding Information Center) Data Library Collection, 1981, DRALIST, Radioactive Decay Data for Application to Radiation Dosimetry and Radiological Assessments, Discussion by D. C. Kocher and R. W. Roussin, ORNL, DLC-80.
- [18] Sjoreen, A. L., D. C. Kocher, G. G. Killough, and C. W. Miller, in press, MLSOIL and DFSOIL - Computer codes to Estimate Effective Ground Surface Concentrations for Dose Computations, ORNL-5974, Oak Ridge National Laboratory.

Appendix A

SAMPLE CASE NO. 1, CONVERTING LIBRARY

* EDIT OF INPUT DATA *

810DATA							
LIBMODE=	0, LBBIN=	10, LIBOUT=	14, IPRT=	6, WSCR=	12, IEDIT=	6, LBDCD=	2,
LIBID=	1						
8END							
8CONST							
NUMREL=	1, KTRUNC=	0, MAXTRU=	8, IEXTRA=	0			
8END							
1 PU-239							

38

CONVERTED DECAY LIBRARY WAS WRITTEN ON UNIT 10

NUMBER OF NUCLIDES = 496

NUMBER OF CHAINS = , 101

DECAY CHAIN (NO. 1) FOR RELEASED NUCLIDE PU-239

NUCLIDE NAME	ID NUMBER (Z-A-S)	DECAY CONSTANT (PER SECOND)	RADIOACTIVE DAUGHTERS OF NUCLIDE.....			
			NAME	BRANCHING FRACTION	NAME	BRANCHING FRACTION
1. PU-239	942390	9.1022E-13	U-235	1.000000		
2. U-235	922350	3.1208E-17	TH-231	1.000000		
3. TH-231	902310	7.5447E-06	PA-231	1.000000		
4. PA-231	912310	5.8949E-13	AC-227	1.000000		
5. AC-227	892270	1.0088E-09	FR-223	0.013800	TH-227	0.986200
6. TH-227	902270	4.2840E-07	RA-223	1.000000		
7. FR-223	872230	5.2993E-04	RA-223	0.999940		
8. RA-223	882230	7.0164E-07	RN-219	1.000000		
9. RN-219	862190	1.7504E-01	PO-215	1.000000		
10. PO-215	842150	3.8985E-02	PB-211	0.999998		
11. PB-211	822110	3.2001E-04	BI-211	1.000000		
12. BI-211	832110	5.4237E-03	TL-207	0.997270	PO-211	0.002730
13. PO-211	842110	1.3433E-00	(DAUGHTER IS STABLE)			
14. TL-207	812070	2.4219E-03	(DAUGHTER IS STABLE)			

SINGLE NUCLIDES	
NAME	DECAY CONSTANT PER SECOND
H-3	1.79E-09
He-7	1.50E-07
He-10	1.37E-14
C-11	5.64E-04
C-13	1.16E-03
C-14	3.83E-12
O-15	5.67E-03
N-16	9.7E-02
F-18	1.05E-04
Na-22	8.44E-09
Na-24	1.28E-05
Al-26	3.05E-14
Mg-27	1.22E-03
Si-31	7.34E-05
P-33	3.16E-07
S-35	9.17E-08
Cl-36	2.30E-14
Ar-37	2.29E-07
Cl-38	3.10E-14
Ar-39	8.17E-11
K-40	1.7E-17
Ar-41	1.05E-04
Ca-41	2.13E-13
K-42	1.56E-05
K-43	8.52E-06
Ca-45	4.93E-08
Ti-45	6.25E-05
Sc-48	4.41E-06
V-48	5.02E-07
Ti-51	2.81E-03
Cr-51	2.90E-07
V-52	3.08E-03
Mn-53	5.94E-15
Mn-54	2.57E-08
Fe-55	8.14E-09
Mn-56	7.47E-05
Mn-57	7.84E-03
Fe-59	1.80E-07
Ni-59	2.93E-13
Co-61	1.17E-04
Cu-61	5.65E-05
Ni-63	2.19E-10
Cu-64	1.52E-05

NUCLIDE CHAINS. THE DECAY MATRIX IS PRINTED WITH THE DIAGONAL IN THE COLUMN TO THE FAR RIGHT. MOST ZERO ENTRIES ARE OMITTED.

DECAY CONSTANT		-----DAUGHTERS-----		DECAY MATRIX					
NAME	PER SECOND	NAME	BRANCH. FRAC.	NAME	BRANCH. FRAC.				
CHAIN NUMBER	1								
				D-5	D-4	D-3	D-2	D-1	D
1 CF-253	4.505E-07	CM-249	0.003100						-4.505E-07
				ES-253	0.996900				
2 ES-253	3.920E-07	BK-249	1.000000					3.908E-07	-3.920E-07
3 CM-249	1.801E-04	BK-249	1.000000				5.583E-07	0.0	-1.801E-04
4 BK-249	2.507E-08	CF-249	0.999985				2.507E-08	2.507E-08	-2.507E-08
5 CF-249	6.265E-11	CM-245	1.000000					6.265E-11	-6.265E-11
6 PU-245	1.822E-05	AM-245	1.000000						-1.822E-05
7 AM-245	9.438E-05	CM-245	1.000000					9.438E-05	-9.438E-05
8 CM-245	2.584E-12	PU-241	1.000000			2.584E-12	0.0	2.584E-12	-2.584E-12
9 PU-241	1.525E-09	AM-241	0.999975					1.525E-09	-1.525E-09
10 AM-241	5.082E-11	NP-237	1.000000					5.082E-11	-5.082E-11
11 PU-237	1.771E-07	NP-237	0.999950						-1.771E-07
12 U-237	1.189E-06	NP-237	1.000000						-1.189E-06
13 NP-237	1.026E-14	PA-233	1.000000			1.026E-14	1.026E-14	1.026E-14	-1.026E-14

NUCLIDE CHAINS. THE DECAY MATRIX IS PRINTED WITH THE DIAGONAL IN THE COLUMN TO THE FAR RIGHT. MOST ZERO ENTRIES ARE OMITTED.

DECAY CONSTANT		-----DAUGHTERS-----		DECAY MATRIX	
NAME	PER SECOND	NAME	BRANCH. FRAC.	NAME	BRANCH. FRAC.

CHAIN NUMBER 97

1 OS-193	6.418E-06			D-1	D
		IR-193M	0.003500		-6.418E-06
2 IR-193M	6.742E-07			2.360E-09	-6.742E-07

CHAIN NUMBER 98

1 PT-193M	1.853E-06			D-1	D
		PT-193	1.000000		-1.853E-06
2 PT-193	4.393E-10			4.393E-10	-4.393E-10

CHAIN NUMBER 99

1 AU-195M	2.265E-02			D-1	D
		AU-195	1.000000		-2.265E-02
2 AU-195	4.384E-08			4.384E-08	-4.384E-08

CHAIN NUMBER 100

1 PT-197M	1.224E-04			D-1	D
		PT-197	0.967000		-1.224E-04
2 PT-197	1.052E-05			1.017E-05	-1.052E-05

CHAIN NUMBER 101

1 HG-197M	8.090E-06			D-1	D
		HG-197	0.930000		-8.090E-06
2 HG-197	3.002E-06			2.792E-06	-3.002E-06

SAMPLE CASE NO. 2, FOR PRIMUS DATA SET, WITH SOURCE DATA INPUT.

* EDIT OF INPUT DATA *

210DATA							
LDNODE=	1,LDBIN=	10,LIBOUT=	14,IPRINT=	6,NSCR=	12,IEDIT=	1,LDLCD=	2,
IDLIB=	0						
2END							
2CONST							
NUMREL=	18,KTRUNC=	0,MAXTRU=	8,IEXTRA=	1			
2F77							

1	PU-240	1.000E 00	4.000E 00	1.000E 00
2	PU-239	1.000E 00	3.000E -05	1.000E 00
3	U-239	1.000E 00	3.600E -04	1.000E 00
4	PU-238	1.000E 00	3.800E -04	1.000E 00
5	M-139	1.000E 00	4.500E -04	1.000E 00
6	XE-139	1.000E 00	5.200E -03	1.000E 00
7	CS-137	1.000E 00	9.900E -04	1.000E 00
8	XE-137M	1.000E 00	1.000E 00	1.000E 00
9	CS-134	1.000E 00	1.000E 00	1.000E 00
10	XE-133M	1.000E 00	1.000E 00	1.000E 00
11	I-131	1.000E 00	1.000E 00	1.000E 00
12	SR-125	1.000E 00	1.000E 00	1.000E 00
13	BU-104	1.000E 00	1.000E 00	1.000E 00
14	SR-90	1.000E 00	1.000E 00	1.000E 00
15	IR-88	1.000E 00	1.000E 00	1.000E 00
16	IR-85M	1.000E 00	1.000E 00	1.000E 00
17	IR-41	1.000E 00	1.000E 00	1.000E 00
18	H-3	1.000E 00	1.000E 00	1.000E 00
1	PU-240	4.000E 00	4.000E 00	1.000E 00
2	PU-239	3.000E -05	3.000E -05	1.000E 00
3	U-239	3.600E -04	3.600E -04	1.000E 00
4	PU-238	3.800E -04	3.800E -04	1.000E 00
5	M-139	4.500E -04	4.500E -04	1.000E 00
6	XE-139	5.200E -03	5.200E -03	1.000E 00
7	CS-137	9.900E -04	9.900E -04	1.000E 00
8	XE-137M	1.000E 00	1.000E 00	1.000E 00
9	CS-134	1.000E 00	1.000E 00	1.000E 00
10	XE-133M	1.000E 00	1.000E 00	1.000E 00
11	I-131	1.000E 00	1.000E 00	1.000E 00
12	SR-125	1.000E 00	1.000E 00	1.000E 00
13	BU-104	1.000E 00	1.000E 00	1.000E 00
14	SR-90	1.000E 00	1.000E 00	1.000E 00
15	IR-88	1.000E 00	1.000E 00	1.000E 00
16	IR-85M	1.000E 00	1.000E 00	1.000E 00
17	IR-41	1.000E 00	1.000E 00	1.000E 00
18	H-3	1.000E 00	1.000E 00	1.000E 00

DECAY CHAIN (NO. 1) AND ASSOCIATED DATA FOR RELEASED NUCLIDE PU-240
(RELEASE VECTOR = 1.0000E 00)

NUCLIDE NAME	ID NUMBER (Z-A-S)	ARRAY DATA	EXPOSURE VECTOR	DECAY CONSTANT (PER SECOND)	RADIOACTIVE DAUGHTERS OF NUCLIDE.....			
					NAME	BRANCHING FRACTION	NAME	BRANCHING FRACTION
1. PU-240	942400	6.0000E 00	1.0000E 00	3.3437E-12	U-236	1.000000		
2. U-236	922360	0.0	0.0	9.3005E-16	TH-232	1.000000		
3. TH-232	902320	0.0	0.0	1.5633E-18	RA-228	1.000000		
4. RA-228	882280	0.0	0.0	3.8199E-09	AC-228	1.000000		
5. AC-228	892280	0.0	0.0	3.1410E-05	TH-228	1.000000		
6. TH-228	902280	0.0	0.0	1.1481E-08	RA-224	1.000000		
7. RA-224	882240	0.0	0.0	2.2162E-06	RN-220	1.000000		
8. RN-220	862200	0.0	0.0	1.2444E-02	PO-216	1.000000		
9. PO-216	842160	0.0	0.0	4.7476E 00	PB-212	1.000000		
10. PB-212	822120	0.0	0.0	1.8091E-05	BI-212	1.000000		
11. BI-212	832120	0.0	0.0	1.9079E-04	TL-208	0.359300	PO-212	0.640700
12. PO-212	842120	0.0	0.0	2.3260E 06	(DAUGHTER IS STABLE)			
13. TL-208	812080	0.0	0.0	3.7840E-03	(DAUGHTER IS STABLE)			

DECAY CHAIN (NO. 2) AND ASSOCIATED DATA FOR RELEASED NUCLIDE PU-239
(RELEASE VECTOR = 1.0000E 00)

NUCLIDE NAME	ID NUMBER (Z-A-S)	ARRAY DATA	EXPOSURE VECTOR	DECAY CONSTANT (PER SECOND)	RADIOACTIVE DAUGHTERS OF NUCLIDE.....			
					NAME	BRANCHING FRACTION	NAME	BRANCHING FRACTION
1. PU-239	942390	3.0000E-05	1.0000E 00	9.1022E-13	U-235	1.000000		
2. U-235	922350	3.6000E-04	1.0000E 00	3.1208E-17	TH-231	1.000000		
3. TH-231	902310	0.0	0.0	7.5447E-06	PA-231	1.000000		
4. PA-231	912310	0.0	0.0	5.8949E-13	AC-227	1.000000		
5. AC-227	892270	0.0	0.0	1.0088E-09	FR-223	0.013800	TH-227	0.986200
6. TH-227	902270	0.0	0.0	4.2860E-07	RA-223	1.000000		
7. FR-223	872230	0.0	0.0	5.2993E-04	RA-223	0.999940		
8. RA-223	882230	0.0	0.0	7.0164E-07	RN-219	1.000000		
9. RN-219	862190	0.0	0.0	1.7504E-01	PO-215	1.000000		
10. PO-215	842150	0.0	0.0	3.8985E 02	PB-211	0.999998		
11. PB-211	822110	0.0	0.0	3.2001E-04	BI-211	1.000000		
12. BI-211	832110	0.0	0.0	5.4237E-03	TL-207	0.997270	PO-211	0.002730
13. PO-211	842110	0.0	0.0	1.3433E 00	(DAUGHTER IS STABLE)			
14. TL-207	812070	0.0	0.0	2.4219E-03	(DAUGHTER IS STABLE)			

DECAY CHAIN (NO. 3) AND ASSOCIATED DATA FOR RELEASED NUCLIDE PU-238
(RELEASE VECTOR = 1.0000E 00)

NUCLIDE NAME	ID NUMBER (Z-A-S)	ARRAY DATA	EXPOSURE VECTOR	DECAY CONSTANT (PER SECOND)	RADIOACTIVE DAUGHTERS OF NUCLIDE.....	
					NAME BRANCHING FRACTION	NAME BRANCHING FRACTION
1. PU-238	942380	5.2000E-03	1.0000E 00	2.5031E-10	U-234	1.000000
2. U-238	922380	0.0	1.0000E 00	4.9160E-18	TH-234	1.000000
3. TH-234	902340	0.0	1.0000E 00	3.3289E-07	PA-234M	1.000000
4. PA-234M	912341	0.0	0.0	9.8739E-03	PA-234	0.001600
5. PA-234	912340	0.0	0.0	2.8737E-05	U-234	1.000000
6. U-234	922340	0.0	0.0	8.9834E-14	TH-230	1.000000
7. TH-230	902300	0.0	0.0	2.8525E-13	RA-226	1.000000
8. RA-226	882260	0.0	0.0	1.3728E-11	RN-222	1.000000
9. RN-222	862220	0.0	0.0	2.0982E-06	PO-218	1.000000
10. PO-218	842180	0.0	0.0	3.7877E-03	PB-214	0.999800
11. PB-214	822140	0.0	0.0	4.3106E-04	BI-214	1.000000
12. BI-214	832140	0.0	0.0	5.8053E-04	PO-214	0.999790
13. PO-214	842140	0.0	0.0	4.2343E-03	PB-210	1.000000
14. PB-210	822100	0.0	0.0	9.8673E-10	BI-210	1.000000
15. BI-210	832100	0.0	0.0	1.6003E-06	PO-210	0.999999
16. PO-210	842100	0.0	0.0	5.7976E-08	(DAUGHTER IS STABLE)	

DECAY CHAIN (NO. 3) AND ASSOCIATED DATA FOR RELEASED NUCLIDE U-238
(RELEASE VECTOR = 1.0000E 00)

NUCLIDE NAME	ID NUMBER (Z-A-S)	ARRAY DATA	EXPOSURE VECTOR	DECAY CONSTANT (PER SECOND)	RADIOACTIVE DAUGHTERS OF NUCLIDE.....	
					NAME BRANCHING FRACTION	NAME BRANCHING FRACTION
1. U-238	922380	3.8000E-04	1.0000E 00	4.9160E-18	TH-234	1.000000
2. TH-234	902340	4.5000E-04	1.0000E 00	3.3289E-07	PA-234M	1.000000
3. PA-234M	912341	0.0	0.0	9.8739E-03	PA-234	0.001600
4. PA-234	912340	0.0	0.0	2.8737E-05	U-234	1.000000
5. U-234	922340	0.0	0.0	8.9834E-14	TH-230	1.000000
6. TH-230	902300	0.0	0.0	2.8525E-13	RA-226	1.000000
7. RA-226	882260	0.0	0.0	1.3728E-11	RN-222	1.000000
8. RN-222	862220	0.0	0.0	2.0982E-06	PO-218	1.000000
9. PO-218	842180	0.0	0.0	3.7877E-03	PB-214	0.999800
10. PB-214	822140	0.0	0.0	4.3106E-04	BI-214	1.000000
11. BI-214	832140	0.0	0.0	5.8053E-04	PO-214	0.999790
12. PO-214	842140	0.0	0.0	4.2343E-03	PB-210	1.000000
13. PB-210	822100	0.0	0.0	9.8673E-10	BI-210	1.000000
14. BI-210	832100	0.0	0.0	1.6003E-06	PO-210	0.999999
15. PO-210	842100	0.0	0.0	5.7976E-08	(DAUGHTER IS STABLE)	

7. CONCLUSIONS

In a sense PRIMUS and the decay package subroutines are the core of the CRRIS. It is the use of the PRIMUS file that makes the CRRIS modular and assures that radioactive decay and daughter buildup are handled in a uniform and correct manner. The methodology set up in PRIMUS is the basis of the organization of the CRRIS data files. All environmental concentrations of radionuclides are accessed via the indexing arrays set up in PRIMUS.

It is not necessary to understand all the theory behind the decay calculations, or any other computations in the CRRIS, in order to run the programs and use the results. Some understanding of how the computations in PRIMUS and the other CRRIS codes are done is needed to intelligently interpret the (sometimes) large quantity of numbers produced by the CRRIS.

As is the case with most large computer systems, PRIMUS and the other CRRIS codes may never be "finished." Be aware that future modifications and enhancements are planned. We hope to add a computation to PRIMUS that will determine the maximum "useful" decay chain length. This will eliminate the computations of decay products in amounts so small that they are less than the accuracy achievable with IBM double precision computations.

DECAY CHAIN (NO. 15) AND ASSOCIATED DATA FOR RELEASED NUCLIDE KR-85M
(RELEASE VECTOR = 1.0000E 00)

NUCLIDE NAME	ID NUMBER (Z-A-S)	ARRAY DATA	EXPOSURE VECTOR	DECAY CONSTANT (PER SECOND)	RADIOACTIVE DAUGHTERS OF NUCLIDE.....	
					NAME BRANCHING FRACTION	NAME BRANCHING FRACTION
1. KR-85M	360851	3.4000E 03	1.0000E 00	4.2978E-05	KR-85	0.211000
2. KR-85	360850	1.1000E 03	1.0000E 00	2.0489E-09	(DAUGHTER IS STABLE)	

DECAY CHAIN (NO. 16) AND ASSOCIATED DATA FOR RELEASED NUCLIDE AR-41
(RELEASE VECTOR = 1.0000E 00)

NUCLIDE NAME	ID NUMBER (Z-A-S)	ARRAY DATA	EXPOSURE VECTOR	DECAY CONSTANT (PER SECOND)	RADIOACTIVE DAUGHTERS OF NUCLIDE.....	
					NAME BRANCHING FRACTION	NAME BRANCHING FRACTION
1. AR-41	180410	1.1000E 05	1.0000E 00	1.0539E-04	(DAUGHTER IS STABLE)	

DECAY CHAIN (NO. 17) AND ASSOCIATED DATA FOR RELEASED NUCLIDE H-3
(RELEASE VECTOR = 1.0000E 00)

NUCLIDE NAME	ID NUMBER (Z-A-S)	ARRAY DATA	EXPOSURE VECTOR	DECAY CONSTANT (PER SECOND)	RADIOACTIVE DAUGHTERS OF NUCLIDE.....	
					NAME BRANCHING FRACTION	NAME BRANCHING FRACTION
1. H-3	10030	3.5000E 03	1.0000E 00	1.7886E-09	(DAUGHTER IS STABLE)	

IHC0021 STOP 1

Appendix B


```

//ALSPRIM JOB (22864,I3), 'ALS/B212/XCSD', TIME=1
/*ROUTE PRINT RNT59
//S EXEC FORTHCLG, REGION.GO=270K
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
=PRIMUS.FIN
//GO.FT07F001 DD DUMMY
//GO.FT06F001 DD SYSOUT=A
//GO.FT02F001 DD DISP=SHR, DSN=CAD10.OWH19503.DECAYDAT, LABEL=(,,,IN)
//GO.FT10F001 DD UNIT=SYSDA, DSN=ββX, DISP=(NEW,PASS),
// SPACE=(4000,(30,8),ELSE),
// DCB=(RECFM=VBS,LRECL=X,BLKSIZE=4000)
//GO.FT12F001 DD UNIT=SYSDA, DSN=ββT, DISP=(NEW,PASS),
// SPACE=(4000,(30,8),ELSE),
// DCB=(RECFM=VBS,LRECL=X,BLKSIZE=4000)
//GO.FT14F001 DD DSN=T.ALS22864.PRIMU2, DISP=(NEW,CATLG),
// SPACE=(2000,(20,8),ELSE), UNIT=SPDA,
// DCB=(RECFM=VBS,LRECL=X,BLKSIZE=2000)
//GO.FT09F001 DD SYSOUT=A, DCB=(RECFM=FA,BLKSIZE=133)
//GO.FT05F001 DD *
=PRIMUS.DAT
/*
//
ENDINPUT

```

TYPE APP.PGM

Appendix C

C ----- WRITTEN FOR PRIMUS BY O. W. HERMANN, 12/30/80.
 C
 C (THIS DECK CONTAINS ALL REVISIONS AND MODIFICATIONS THROUGH 2/7/1983.)
 C ----- O. W. H.
 C

```
DOUBLE PRECISION E(1), RELSYM(200), EXPSYM(400)
DIMENSION NUMEXP(200), XINPUT(400), D(45000), EVECT(400), RVECT(400)
DIMENSION T(20)
DIMENSION TI(410)
COMMON /IONUMB/ NR, NREAD, KOUT, NTEMP
COMMON /IEXT/ IEXTRA
COMMON/NUCDAT/ NUMCHA, NEXP, NEXPT, KTRUNC, LMX, NSHORT, NREL
DATA BLANK / 4H      /
EQUIVALENCE (E(1), D(1) )
NAMelist /IODATA/ LBMODE, LBBIN, LIBOUT, IPRINT, NSCR, IEDIT,
- LBBCD, IDLIB
NAMelist /CONST/ NUMREL, KTRUNC, MAXTRU, IEXTRA
NAMelist /NAMES/ RELSYM, NUMEXP
NAMelist /EXTRA/ EXPSYM, XINPUT, EVECT, RVECT
```

C
 1 CONTINUE
 LIM = 18000
 LIM = 45000
 IN = 5
 NSIZR = 200
 NSIZE = 400
 IEDIT = 1
 IDLIB = 0
 LBMODE = 1
 LBBCD = 2
 LBBIN = 10
 NSCR = 12
 LIBOUT = 14
 IPRINT = 6
 NUMREL = 1
 KTRUNC = 0
 MAXTRU = 8
 IEXTRA = 0
 NTI = 0
 2 READ(IN,3,END=1000) T
 3 FORMAT (20A4)
 KT = 0
 DO 4 L = 1,20
 IF (T(L) .NE. BLANK) KT = 1
 4 TI(NTI+L) = T(L)
 IF (NTI .LT. 400) NTI = NTI + 20
 IF (KT .NE. 0) GO TO 2
 DO 6 L = 1, NSIZR
 6 NUMEXP(L) = 1

C
 READ(IN, IODATA)
 C

```

WRITE(IPRINT,8)(TI(L), L=1,NTI)
8 FORMAT(1H1, 20A4/ 22(1H , 20A4/))
IF (IEDIT .GT. 0) WRITE(IPRINT, 10)
10 FORMAT (1H0,10X,  '* EDIT OF INPUT DATA *' /1H )
IF (IEDIT .GT. 0) WRITE(IPRINT, IODATA)
READ(IN, CONST)
IF (IEDIT .GT. 0) WRITE(IPRINT, CONST)
READ(IN,NAMES)
IF (IEDIT .GT. 0 .AND. IEXTRA .EQ. 0)
*WRITE(IPRINT,12) (L, RELSYM(L), L = 1, NUMREL)
12 FORMAT (1X,I3,1X, A8)
NT = LBBCD
NR = LBBIN
NREAD = LIBOUT
KOUT = IPRINT
NTEMP = NSCR
NREL = NUMREL
NRELIN = NREL
LMX = MAXTRU
IF (LBMODE .EQ. 0) CALL CHAIN1 ( D, NT, NR, KOUT, NTOT, LIM)
NT = 0
C MADE ZERO SO THAT IT CAN NOT BE WRITTEN OVER.
MXE = 0
DO 30 L = 1, NUMREL
30 MXE = MAXO (NUMEXP(L), MXE)
LSYM = 1
LXDA = LSYM + 2 * MXE * NUMREL
LEV = LXDA + MXE * NUMREL
LNEX = LEV + MXE * NUMREL
DO 33 I = 1, LIM
33 D(I) = 0.0
DO 40 L = 1, NSIZE
XINPUT(L) = 0.0
RVECT (L) = 0.0
40 ETECT (L) = 0.0
IF (IEXTRA .EQ. 0) GO TO 50
READ (IN,EXTRA)
IF (IEDIT .GT. 0)
*WRITE(IPRINT,42) (L, RELSYM(L), RVECT(L), L = 1, NUMREL)
42 FORMAT (1X,I3,1X,A8,1PE18.4)
CALL MOVEIN (NUMREL, MXE, NUMEXP, D(LSYM), D(LXDA), D(LEV),RELSYM,
* EXPSYM, XINPUT, ETECT, RVECT, IEDIT)
C ....MOVES DATA ARRAYS TO RECTANGULAR ARRYS (FOR LIBMAX)..
C
GO TO 54
50 CONTINUE
DO 52 L = 1, NUMREL
52 E(L) = RELSYM(L)
54 REWIND NR
READ(NR) NTOT, NDUM, MAXMX
NTOT1 = NTOT + 1
MAX2 = MAXMX * MAXMX

```

```

NNUC = NREL
NUCD = NNUC
NUMX = 1
K = (LNEX+1) / 2
K = 2 * K + 1

```

```

LBNC = K
LNN = LBNC + NTOT * 2
LCU = LNN + NTOT1

```

C

```

LINNP = LCU + NNUC
LNDON = LINNP + NUCD
LDLAM = LNDON + NREL
LITR = LDLAM + NTOT1
LJTR = LITR + MAX2 * 2
LQIN = LJTR + 1
LIND = LQIN + MAXMX
LNUCI = LIND + MAXMX
LWET = LNUCI + MAXMX
LDRY = LWET + MAXMX
LWTR = LDRY + MAXMX * NUMX
LDRR = LWTR + NUCD
LS = LDRR + NUCD * NUMX
LICH A = LS + MAX2
LR = LICH A + NNUC
LIDEX = LR + MAX2
LIDET = LIDEX + MAXMX * NREL
LQSHR = LIDET + 8 * NREL
LIC = LQSHR + 20
LJC = LIC + 4000
LNEX = LJC + 4000
IF (LNEX .GT. LIM) CALL ERROR

```

C

```

DO 66 I = 1, NNUC
66 D(LCU + I - 1) = 1.0

```

C

C

C

```

CALL SETDEC(NTOT, NTOT1, MAXMX, MAX2, NUCD, NNUC, NUMX,
* NUCOUT, D(LBNC), D(LBNC), D(LNN), D(LCU), RELSYM,
* D(LINNP), D(LNDON), D(LDLAM),
* D(LITR), D(LJTR), D(LQIN), D(LIND), D(LNUCI), D(LWET), D(LDRY),
* D(LWTR), D(LDRR), D(LS), D(LICH A), D(LR), D(LIDEX), D(LIDET),
* D(LQSHR), D(LIC), D(LJC), NPACK)

```

```

LCONC = LNEX
LVELD = LCONC + NREL * NEXP
LIDRL = LVELD + NEXP
LIDRT = LIDRL + NREL
LJNUM = LIDRT + NREL
LJINL = LJNUM + NREL
LRVEC = LJINL + MAXMX
LCPAK = LRVEC + NREL
LIREF = LCPAK + NPACK

```

```

LNEX = LLEAF + NREL
IF (LNEX .GT. LIM) CALL ERROR
NRELIN = NREL
CALL LIBMAK (NRELIN, NEXP, MXE, D(LIDEX), D(LCONC), D(LVELD),
* D(LXDA), D(LEV), D(LIDET), D(LIDEL), D(LIDRT), D(LQSHR), D(LSYM),
* D(LJNUM), NUMEXP, D(LR), D(LITR), D(LQIN), D(LWET), D(LDRY),
* D(LNUCI), D(LJINL), D(LRVEC), RVECT, D(LIC), D(LJC), D(LCPAK),
* D(LIREF), NPACK, NUCOUT, NUMX, NAME)
LBF1 = LNEX
LBF2 = LBF1 + NTOT
LIN1 = LBF2 + NTOT
LIN2 = LIN1 + NTOT
LNEX = LIN2 + NTOT
IF (LNEX .GT. LIM) CALL ERROR
CALL EDIT (IEDIT, MXE, NEXP, D(LBNC), D(LDLAM), D(LBF1), D(LBF2),
* D(LIN1), D(LIN2), D(LS), D(LJINL), D(LNUCI), D(LIND),
* D(LCONC), D(LVELD), D(LRVEC), D(LIREF) )
IF (IDLIB.NE.0) CALL WRTLIB(NR, IDLIB)
IF (NREL .GT. 0) GO TO 1
1000 STOP 1
END
SUBROUTINE CHAIN1 (D, NT, NR, KOUT, NTOT, LIM)
C
C ** CALLED FROM PRIMUS CODE MAIN **
C
C -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C
C DIMENSION D(1)
C
C REWIND NT
C REWIND NR
C NTOT = 0
3 READ (NT, 4, END=5)
C NTOT = NTOT + 1
C GO TO 3
4 FORMAT (80X)
5 REWIND NT
C
C LANC = 1
C LBNC = LANC + 2 * NTOT
C LAZS = LBNC + 2 * NTOT
C LIN1 = LAZS + NTOT
C LIN2 = LIN1 + NTOT
C K = (LIN2 + NTOT + 1) / 2
C LDA1 = 2 * K + 1
C LDA2 = LDA1 + 2 * NTOT
C LNPO = LDA2 + 2 * NTOT
C LNCH = LNPO + NTOT
C LNP = LNCH + NTOT
C IF (LNP + NTOT .GT. LIM) CALL ERROR
C
C CALL CHAIN2 (D(LANC), D(LBNC), D(LDA1), D(LDA2), D(LNPO),

```

```

      * D(LNCH), D(LIN1), D(LIN2), D(LNP), D(LAZS), NTOT, NT, KOUT)
C
      LMX = LDA1
      IF (LMX .GT. LIM) CALL ERROR
C
      CALL CHAIN3 ( D(LMX), D(LIN1), D(LIN2), D(LNCH), D(LANC), D(LNPO),
      * NTOT, NCHTOT, MAXMX)
C
      LDLA = LMX + NTOT
      LBF1 = LDLA + NTOT
      LBF2 = LBF1 + NTOT
      IF (LBF2 + NTOT .GT. LIV) CALL ERROR
C
      CALL CHAIN4 ( D(LANC), D(LNP), D(LNCH), D(LDLA), D(LBF1), D(LBF2),
      * D(LBNC), D(LIN1), D(LIN2), NT, NR, NTOT, NCHTOT, MAXMX)
C
      LIND = LNP + NTOT
      LS = LIND + NTOT
      IF (LS + NTOT .GT. LIM) CALL ERROR
C
      LOOP TO MAKE EACH CHAIN AND WRITE IT ON NR....
C
      DO 40 N = 1, NCHTOT
C
      CALL CHAIN5 ( D(LMX), D(LNPO), D(LIND), D(LIN1), D(LIN2), D(LNCH),
      * D(LAZS), N, NTOT, L)
C
      CALL CHAIN6 ( N, L, NR, D(LS), D(LS), D(LIND), D(LBF1), D(LBF2),
      * D(LIN1), D(LIN2), D(LDLA), D(LNPO) )
C
40 CONTINUE
      REWIND NT
      REWIND NR
      WRITE (KOUT, 53) NR, NTOT, NCHTOT
53 FORMAT (1H1,45H CONVERTED DECAY LIBRARY WAS WRITTEN ON UNIT
      * , I3/1H0, 22H NUMBER OF NUCLIDES = , I4/1H0,
      * 21H NUMBER OF CHAINS = , I4)
      RETURN
      END
      SUBROUTINE CHAIN2 (ANC, BNC, DA1, DA2, NPOSCH, NCHAIN,
      * INDAU1, INDAU2, NP, IAZS, NTOT, NT, KOUT)
C
C -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C
C
C
C ** CALLED FROM CHAIN1 **
C
      DIMENSION NP(1), INDAU1(1), INDAU2(1), NPOSCH(1), NCHAIN(1)
      * , IAZS(1)
      DOUBLE PRECISION ANC(1), BNC(1), DA1(1), DA2(1), FIND, BLANK
C
      DATA BLANK / 8H          /

```

```

REWIND NT
N = 1
4 READ (NT, 8, END=10) ANC(N), DA1(N), DA2(N)
  N = N + 1
  GO TO 4
8 FORMAT ( A8, 2X, 2( A8, 12X) )
10 NTOT = N - 1
  REWIND NT
  DO 12 I = 1, NTOT
    NPOSCH(I) = 0
    NCHAIN(I) = 0
    INDAU1(I) = 0
    INDAU2(I) = 0
    NP(I) = I
    L = ID(ANC(I))
    IZ = L / 10000
    L = L - IZ * 10000
    IA = L / 10
    IS = L - IA * 10
    IAZS(I) = IA * 10000 + IZ * 10 + IS
12 BNC(I) = ANC(I)
  CALL SORTP (BNC, NP, NTOT, KOUT)
  DO 40 M = 1, NTOT
    GO TO 16
14 FIND = DA2(M)
  IF (FIND .EQ. BLANK) GO TO 40
  GO TO 20
16 FIND = DA1(M)
  IF (FIND .EQ. BLANK) GO TO 14
20 I = 1
  N = NTOT
22 J = (I + N) / 2
  IF (FIND - BNC(J) ) 24, 36, 34
24 IF (J .EQ. N) GO TO 50
C   AT 50 IT MEANS NOT FOUND — ERROR
C   BELOW, FIND .LT. BNC -- LOOK IN LOWER HALF OF TABLE...
  N = J
  GO TO 22
34 IF ( J .EQ. I) GO TO 60
C   BELOW, FIND .GT. BNC -- LOOK IN UPPER HALF OF TABLE...
  I = J + 1
  GO TO 22
36 IF (FIND .EQ. DA1(M) ) INDAU1(M) = NP(J)
  IF (FIND .EQ. DA2(M) ) INDAU2(M) = NP(J)
  IF (FIND .EQ. DA1(M) ) GO TO 14
40 CONTINUE
  DO 46 I = 1, NTOT
    J = INDAU1(I)
    K = INDAU2(I)
    IF (J .GT. 0 .AND. DA1(I) .NE. ANC(J) ) GO TO 41
    IF (K .GT. 0 .AND. DA2(I) .NE. ANC(K) ) GO TO 42
    GO TO 46

```



```

41 WRITE(KOUT,44) I, ANC(I), DA1(I), ANC(J)
   GO TO 46
42 WRITE(KOUT,44) I, ANC(I), DA2(I), ANC(K)
44 FORMAT(1H0,50H NO.  NUCLIDE DAUGHTER  POINTER (INCORRECT)
   * /  1X, 13,1X, A8, 2X, A8, 4X, A8)
46 CONTINUE
   RETURN
50 WRITE (KOUT,55)
55 FORMAT (1H0,  40H**ERROR - NUCLIDE NOT FOUND 1      )
   STOP 3
60 WRITE (KOUT,65)
65 FORMAT (1H0,  40H**ERROR - NUCLIDE NOT FOUND 2      )
   RETURN
   END
   SUBROUTINE CHAIN3 ( MX, INDAU1, INDAU2, NCHAIN, ANC, NPOSCH,
   * NTOT, NCHTOT, MAXMX)

C
C   ** CALLED FROM CHAIN1 **
C
C   -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C
   DIMENSION MX(1), INDAU1(1), INDAU2(1), NCHAIN(1), NPOSCH(1)
   DOUBLE PRECISION ANC(1), TL209
   COMMON /IONUMB/ NR, NREAD, KOUT, NTEMP
   DATA TL209 /8HTL-209    /
   LOGICAL DONE

C
C SET ALL IN CHAIN TO POSITION NO. AND SAME CHAIN NO.
C START 1ST LOOP AT TL-209 IF IT IS THERE..
C
   N = 0
   I1 = 1
   DO 48 I = 1, NTOT
   IF (ANC(I) .NE. TL209) GO TO 48
   I1 = I
   GO TO 52
48 CONTINUE
   GO TO 52
50 IF (DONE) GO TO 60
52 DONE = .TRUE.
   DO 58 I = I1, NTOT
   IF ( NCHAIN(I) .GT. 0 ) GO TO 58
   M = INDAU1(I)
   IF (M .EQ. 0) GO TO 58
   IF (INDAU1(M) .NE. 0) GO TO 54
C AT 54, MEANS DAUGHTER HAS DAUGHTER..
   L = INDAU2(I)
   IF (L .GT. 0 .AND. INDAU1(L) .NE. 0) GO TO 54
   IF (NCHAIN(M) .NE. 0) GO TO 53
C AT 53, MEANS DAUGHTER WAS GIVEN CHAIN NO. IN ANOTHER BRANCH...
   IF (L .GT. 0) NCHAIN(M) = NCHAIN(L)
   IF (NCHAIN(M) .NE. 0) GO TO 53

```

```

C
C BELOW DAUGHTER HAS NO DAUGHTER OR CHAIN NO.
C SET BOTH TO THE NEXT CHAIN NO..
    N = N + 1
    NPOSCH(M) = 1
    NPOSCH(I) = 2
    NCHAIN(M) = N
    NCHAIN(I) = N
    MX(N) = 2
C     CHECK FOR 2ND DAUGHTER...
C
    L = INDAU2(I)
    IF (L .EQ. 0) GO TO 58
    NCHAIN(L) = N
    MX(N) = 3
    NPOSCH(L) = 2
    NPOSCH(I) = 3
    GO TO 58
53 NUM = NCHAIN(M)
    NCHAIN(I) = NUM
    MX(NUM) = MX(NUM) + 1
    NPOSCH(I) = MX(NUM)
    K = INDAU2(I)
    IF (K .EQ. 0) GO TO 58
    IF (NPOSCH(M) .EQ. 0) K = M
    IF (NPOSCH(K) .GT. 0) GO TO 58
    NCHAIN(K) = NUM
    MX(NUM) = MX(NUM) + 1
    NPOSCH(K) = MX(NUM)
    GO TO 58
C
C BELOW, DAUGHTER HAS DAUGHTER.
C
54 K = INDAU2(I)
    IF (K .GT. 0) K = NCHAIN(K)
    IF (M .GT. 0) M = NCHAIN(M)
    NUM = MAX0 (K, M)
    IF (NUM .EQ. 0) GO TO 57
    M = INDAU1(I)
    NCHAIN(M) = NUM
    GO TO 53
C     MX(NUM) = MX(NUM) + 1
C     NPOSCH(I) = MX(NUM)
C     NCHAIN(I) = NUM
C     GO TO 58
57 DONE = .FALSE.
58 CONTINUE
    I1 = 1
    GO TO 50
60 NCHTOT = N
    MAXMX = 0
    DO 80 K = 1, N

```

```

80 MAXMX = MAX0 ( MX(K), MAXMX )
   KX = 0
   DO 90 I = 1, NTOT
     J = NCHAIN(I)
     M = INDAU1(I)
     K = INDAU2(I)
     IF (K.GT.0) K1 = NCHAIN(K)
     IF (M.GT.0) M1 = NCHAIN(M)
     IF (M.GT.0 .AND. M1.NE.J) WRITE(KOUT,99)ANC(I),J,ANC(M),M1
     IF (M.GT.0 .AND. M1.NE.J) KX = 1
     IF (K.GT.0 .AND. K1.NE.J) WRITE(KOUT,99)ANC(I),J,ANC(K),K1
     IF (K.GT.0 .AND. K1.NE.J) KX = 1
90 CONTINUE
99 FORMAT (1H0,'ERROR IN CHAIN'/1H0,'CHAIN NO. OF ',A8,' IS',I4
> /1H0, 'CHAIN NO. OF ',A8, ' IS', I4
> /1H0, 'HOWEVER, THESE NUCLIDES ARE IN SAME CHAIN.' )
   IF (KX.NE. 0) CALL ERROR
   RETURN
   END
   SUBROUTINE CHAIN4 (ANC, NP, NCHAIN, DLAM, BF1, BF2, BNC,
* INDAU1, INDAU2, NT, NR,
* NTOT, NCHTOT, MAXMX)

```

```

C
C ** CALLED FROM CHAIN1 **
C
C -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C
   DIMENSION NP(1), NCHAIN(1), DLAM(1), BF1(1), BF2(1)
* , INDAU1(1), INDAU2(1)
   DOUBLE PRECISION ANC(1), BNC(1)
C
   DATA SE, TM, HR, DA, YR/4ES ,4HM ,4HH ,4HD ,4HY /
   DATA SMI, SHR, SDA, SYR /60.0, 3600.0, 8.64E04, 3.15576E07/
C
   REWIND NT
   REWIND NR
   DO 8 I = 1, NTOT
     READ (NT,4) BF1(I), BF2(I), DLAM(I), TF
4 FORMAT (20X, F10.8, 10X, F10.8, 20X, F9.6, A1)
     ALN2 = ALOG(2.0)
     IF (TF .EQ. SE) DLAM(I) =ALN2 / DLAM(I)
     IF (TF .EQ. TM) DLAM(I) =ALN2 / (DLAM(I) * SMI)
     IF (TF .EQ. HR) DLAM(I) =ALN2 / (DLAM(I) * SHR)
     IF (TF .EQ. DA) DLAM(I) =ALN2 / (DLAM(I) * SDA)
     IF (TF .EQ. YR) DLAM(I) =ALN2 / (DLAM(I) * SYR)
8 CONTINUE
   WRITE (NR) NTOT, NCHTOT, MAXMX
   WRITE (NR) ( BNC(I), I = 1, NTOT)
   WRITE (NR) ( NP(I), I = 1, NTOT)
   WRITE (NR) (NCHAIN(I), I = 1, NTOT)
   WRITE (NR) ( DLAM(I), I = 1, NTOT)
   WRITE (NR) ( ANC(I), I = 1, NTOT)

```

```

WRITE (NR) ( BF1(I), I = 1, NTOT)
WRITE (NR) ( BF2(I), I = 1, NTOT)
WRITE (NR) (INDAU1(I), I = 1, NTOT)
WRITE (NR) (INDAU2(I), I = 1, NTOT)

```

```

RETURN

```

```

END

```

```

SUBROUTINE CHAIN5 (MX, NPOSCH, IND, INDAU1, INDAU2, NCHAIN,
* IAZS, N, NTOT, L)

```

```

C
C ** CALLED FROM CHAIN1 **
C
C -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C

```

```

    DIMENSION MX(1), NPOSCH(1), IND(1)
    * , INDAU1(1), INDAU2(1), NCHAIN(1), IAZS(1)
    LOGICAL DONE

```

```

C
C MAKE MATRICES, GET INDICES BY REVERSE POSITIONS..
C
C NEED TO REPOSITION NUCLIDES TO MAKE MATRIX TRIANGULAR....
C

```

```

    L = 0
    DO 20 I = 1, NTOT
    IF (NCHAIN(I) .NE. N) GO TO 20
    NPOSCH(I) = MX(N) - NPOSCH(I) + 1
    K = NPOSCH(I)
    IND(K) = I
    L = L + 1
    IF (MX(N) .EQ. L) GO TO 24
20 CONTINUE
    CALL ERROR

```

```

C
C NOW, IF I DECAYS TO J WHERE IND(K2) = J AND IND(K1) = I AND K2 < K1,
C SET IND(K2) = I AND IND(K) = IND(K-1), FOR K = K2, K1-1. THIS MAKES
C EACH NUC. DECAY TO NUC. OF HIGHER INDEX IN MATRIX, MAKING IT TRIANG..
C

```

```

24 MAX = MX(N)
30 DONE = .TRUE.
    DO 44 K1 = 2, MAX
    I = IND(K1)
    J = INDAU1(I)
    IF (J .EQ. 0) GO TO 44
    K2 = NPOSCH(J)
    IF (K2 .LT. K1) GO TO 34
    J = INDAU2(I)
    IF (J .EQ. 0) GO TO 44
    K2 = NPOSCH(J)
    IF (K2 .GT. K1) GO TO 44
34 KK = K1 - K2
    DO 40 K = 1, KK
    II = IND(K1 - K)
    NPOSCH(II) = K1 - K + 1

```

```

40 IND (K1 - K + 1) = II
   DONE = .FALSE.
   IND(K2) = I
   NPOSCH(I) = K2
44 CONTINUE
   IF (.NOT. DONE) GO TO 30

C
C BELOW, REORDER, FIRST KEEPING DAUGHTERS AT HIGHER POSITIONS
C AND, SECOND, MAKING DECENDING AZS NOS. (OR MOVING NUCLIDE TO
C FOLLOW LAST POSITION OF HIGHER AZS NOS., WHILE ALWAYS
C PRECEDING DAUGHTERS).
C
60 DONE = .TRUE.
C   ITERATE UNTIL DONE (NO CHANGE)
   DO 90 K2 = 1, MAX
   I = IND(K2)
   L1 = INDAU1(I)
   J2 = MAX
   IF (L1 .EQ. 0) GO TO 64
   L2 = INDAU2(I)
   IF (L2 .EQ. 0) L2 = L1
   J2 = MIN0(NPOSCH(L1), NPOSCH(L2)) - 1
   IF (K2 .GT. J2) CALL ERROR
   IF (K2 .EQ. J2) GO TO 90
64 J1 = K2 + 1
   IF (J1 .GT. J2) GO TO 90

C
C NOW, COMPARE NUC AT K2 WITH ALL AF THOSE FROM J1 TO J2.
C PUT NUC AT K2 AFTER LAST J, IF IAZS(I) < IAZS(K) WHERE,
C I = IND(K2) β K = IND(K1=J). FOR NONE, K1 = 0.
C
   K1 = 0
   DO 70 J = J1, J2
   K = IND(J)
   IF (IAZS(I) .LT. IAZS(K)) K1 = J
70 CONTINUE
   IF (K1 .EQ. 0) GO TO 90
   KK = K1 - K2
   DO 80 K = 1, KK
   II = IND(K2 + K)
   NPOSCH(II) = K2 + K - 1
80 IND (K2 + K - 1) = II
   DONE = .FALSE.
   IND(K1) = I
   NPOSCH(I) = K1
90 CONTINUE
   IF (.NOT. DONE) GO TO 60
   RETURN
   END
   SUBROUTINE CHAIN6 (N, L, NR, S, C, IND, BF1, BF2, INDAU1, INDAU2,
   * DLAM, NPOSCH)

```

C

```

C ** CALLED FROM CHAIN1 **
C
C -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C
  DIMENSION S(1), C(L,L), IND(1), BF1(1), BF2(1), INDAU1(1),
  * INDAU2(1), DLAM(1), NPOSCH(1)
  LSQ = L * L
  DO 4 I = 1, LSQ
4 S(I) = 0.0
  DO 8 J = 1, L
    I = IND(J)
    C(J,J) = -DLAM(I)
    M = INDAU1(I)
    IF (M .EQ. 0) GO TO 8
    K = NPOSCH(M)
C    NEED TRANS, AT J-TH COLUMN AND K-TH ROW....
C
    C(J, K) = DLAM(M) * BF1(I)
    M = INDAU2(I)
    IF (M .EQ. 0) GO TO 8
    K = NPOSCH(M)
    C(J, K) = DLAM(M) * BF2(I)
8 CONTINUE
  WRITE (NR) LSQ, L, N, (IND(I), I = 1, L), (S(J), J = 1, LSQ)
  RETURN
  END
  SUBROUTINE EDIT (IEDIT, MXE, NEXP, ANC, DLAM, BF1, BF2, INDAU1,
  * INDAU2, S, JINL, NUCID, IND, CONC, EVCT, RVEC, IREF)
C
C ** CALLED FROM PRIMUS CODE MAIN **
C
C <<<<EDITS PROBLEM WORKING LIBRARY>>>>
C ---MOD. FROM AZS TO ZAS NO., PER R. D. SHARP / OWH, 11/19/82
C
  COMMON /IONUMB/ NR, NREAD, KOUT, NTEMP
  COMMON /IEXT/ IEXTRA
  REAL*8 ANC(1)
  DIMENSION DLAM(1), BF1(1), BF2(1), INDAU1(1), INDAU2(1), S(1),
  * IREF(1), JINL(1), NUCID(1), IND(1), CONC(NEXP,1), EVCT(1), RVEC(1)
C
  REWIND NTEMP
  REWIND NREAD
  REWIND NR
  READ (NREAD) NUMCHA, KTRUNC
  ISK = KTRUNC + 3
  DO 10 I = 1, ISK
10 READ (NREAD)
  READ (NR) NTOT
  DO 14 I = 1, 3
14 READ (NR)
  READ (NR) ( DLAM(L), L = 1, NTOT)
  READ (NR) ( ANC(L), L = 1, NTOT)

```

```

READ(NR) ( BF1(L), L = 1, NTOT)
READ(NR) ( BF2(L), L = 1, NTOT)
READ(NR) (INDAU1(L), L = 1, NTOT)
READ(NR) (INDAU2(L), L = 1, NTOT)
JJ = 0
LN = 0
DO 100 N1 = 1, NUNCHA
  READ(NTMP) L1, (DUM5,M=1,5), (IND(I), I = 1, L1), RIND, JIND
  READ(NREAD) L, LSQ, NCE, NCR, KS, JS, (S(I), I = 1, LSQ),
  * (JINL(J), J = 1, NCR), (NUCID(K), K = 1, L)
  IF (L1 .NE. L) CALL ERROR
  IF (KTRUNC .EQ. 1) READ (NREAD)
  DO 70 J = 1, NCR
    JJ = JJ + 1
    IF (LN + NCE - JINL(J) .GT. 48) LN = 0
    IF (LN .GT. 0) WRITE(KOUT,34)
    IF (LN .GT. 0) GO TO 30
    WRITE (KOUT,26)
26  FORMAT(1H1)
30  LN = LN + 9 + NCE - JINL(J)
    J1 = JINL(J)
    IF (IEXTRA .EQ. 0) WRITE(KOUT,36) N1, ANC(IND(J1))
    IF (IEXTRA .GT. 0) WRITE(KOUT,37) N1, ANC(IND(J1)), RVEC(JJ)
34  FORMAT(1H0)
36  FORMAT (25X,'DECAY CHAIN (NO.', I3,') FOR RELEASED NUCLIDE ',A8/1H0,
  *3X,'NUCLIDE ID NUMBER ',
  *'DECAY CONSTANT ', 10(1L.),'RADIOACTIVE DAUGHTERS OF NUCLIDE',
  * 10(1H.)/6X,'NAME (Z-A-S) ',
  *'(PER SECOND)', 2(' NAME BRANCHING FRACTION') /1H )
37  FORMAT (23X,'DECAY CHAIN (NO.' I3,
  * ') AND ASSOCIATED DATA FOR RELEASED NUCLIDE ',
  * A8/ 47X,'(RELEASE VECTOR =',1PE11.4,')' /1H0
  *3X,'NUCLIDE ID NUMBER ARRAY EXPOSURE ',
  *'DECAY CONSTANT ', 10(1H.),'RADIOACTIVE DAUGHTERS OF NUCLIDE',
  * 10(1H.)/6X,'NAME (Z-A-S) ',
  * ' DATA VECTOR ',
  *'(PER SECOND)', 2(' NAME BRANCHING FRACTION') /1H )
  K = 0
  IT = IREF(JJ) - 1
C
C  IREF WAS SET IN LIBMAK FOR START OF OF NEXT EXP. NUC. FOR KTH REL.
C
DO 70 M = J1, L
  I = IND(M)
  IDI = NUCID(M)
44  K = K + 1
  II = K + IT
  IF (INDAU1(I) .GT. 0) GO TO 46
  IF (IEXTRA.EQ.0)WRITE(KOUT,50) K, ANC(I), IDI, DLAM(I)
  IF (IEXTRA.GT.0)WRITE(KOUT,52) K, ANC(I), IDI, CONC(II,JJ),
  * EVEC(II), DLAM(I)
  GO TO 60

```

```

46 IF (INDAU2(I) .GT. 0) GO TO 48
   IF (IEXTRA.EQ.0)WRITE(KOUT,54) K, ANC(I), IDI, DLAM(I),
   * ANC(INDAU1(I)), BF1(I)
   IF (IEXTRA.GT.0)WRITE(KOUT,56) K, ANC(I), IDI, CONC(II,JJ),
   * EVECT(II), DLAM(I), ANC(INDAU1(I)), BF1(I)
   GO TO 60
48 IF (IEXTRA.EQ.0)WRITE(KOUT,54) K, ANC(I), IDI, DLAM(I),
   * ANC(INDAU1(I)), BF1(I),          ANC(INDAU2(I)), BF2(I)
   IF (IEXTRA.GT.0)WRITE(KOUT,56) K, ANC(I), IDI, CONC(II,JJ),
   * EVECT(II), DLAM(I), ANC(INDAU1(I)), BF1(I),
   * ANC(INDAU2(I)), BF2(I)
50 FORMAT (1X, I2, '. ', A8, I8, 2X,
   * 1PE15.4, 4X, '(DAUGHTER IS STABLE)')
52 FORMAT (1X, I2, '. ', A8, I8, 2X, 1P2E12.4 ,
   * 1PE14.4, 4X, '(DAUGHTER IS STABLE)')
54 FORMAT (1X, I2, '. ', A8, I8, 2X,
   * 1PE15.4, 4X, A8, 0PF12.6, 8X, A8, F12.6)
56 FORMAT (1X, I2, '. ', A8, I8, 2X, 1P2E12.4 ,
   * 1PE14.4, 4X, A8, 0PF12.6, 8X, A8, F12.6)
60 IF (IND(M) .NE. I) GO TO 70
   IF (M .NE. KS) GO TO 66
   I = KIND
   IDI = ID (ANC(I))
   GO TO 44
66 IF (M .NE. JS) GO TO 70
   I = JIND
   IDI = ID (ANC(I))
   GO TO 44
70 CONTINUE
   IT = IT + NCR
   IF (IEDIT .GT. 1) CALL WRMAT (IEDIT, N1, L, S)
100 CONTINUE
   REWIND NREAD
   REWIND NTEMP
   REWIND NR
   RETURN
END
FUNCTION UNPACK(ZAS)
C   THIS ROUTINE UNPACKS THE ZAS NUMBER TO FIND Z,A, AND S, WHERE
C   A=MASS NUMBER, Z=ATOMIC NUMBER, S=THE EXCITATION STATE
C   ZAS = 10000*Z + 10*A + S
C       WHERE S = 0 IF THE NUCLIDE IS IN THE NORMAL STATE
C       S >= 1 WHEN IN THE EXCITATION STATE
DOUBLE PRECISION NAME, NAMEIN, BLANK, UNPACK
INTEGER A, ZAS, Z, EXCITE
INTEGER*2 NUCNAM(103), ELEMNT
LOGICAL*1 LNAME(4,103), HYPHEN, M, LBLANK, LONAME(8), DIGIT(10)
EQUIVALENCE (LNAME(1), NUCNAM(1)), (LONAME(1), NAME),
- (ELEMNT, NAME)
DATA HYPHEN/'-'/, M/'M'/, LBLANK/' '/, DIGIT/'0','1',
- '2','3','4','5','6','7','8','9'/
DATA BLANK/8H /

```



```

DATA NUCNAM/2HH-, 2HHE, 2HLI, 2HBE, 2HB-, 2HC-, 2HN-,
- 2HO-, 2HF-, 2HNE, 2HNA, 2HMG, 2HAL, 2HSI, 2HP-,
- 2HS-, 2HCL, 2HAR, 2HK-, 2HCA, 2HSC, 2HTI, 2HV-, 2HCR,
- 2HDN, 2HFF, 2HCO, 2HNI, 2HCU, 2HZN, 2HGA, 2HGE, 2HAS,
- 2HSE, 2HBR, 2HKB, 2HRB, 2HSR, 2HY-, 2HZR, 2HNB, 2HMO,
- 2HTC, 2HRU, 2HRH, 2HPD, 2HAG, 2HCD, 2HIN, 2HSN, 2HSB,
- 2HTE, 2HI-, 2HXE, 2HCS, 2HBA, 2HLA, 2HCE, 2HPR, 2HND,
- 2HPM, 2HSM, 2HEU, 2HGD, 2HTB, 2HDY, 2HKO, 2HER, 2HTM,
- 2HYB, 2HLU, 2HHF, 2HTA, 2HW-, 2HRE, 2HOS, 2HIR, 2HPT,
- 2HAU, 2HBG, 2HTL, 2HPB, 2HBI, 2HPO, 2HAT, 2HRN, 2HFR,
- 2HRA, 2HAC, 2HTH, 2HPA, 2HU-, 2HNP, 2HPU, 2HAM, 2HCM,
- 2HBK, 2HCF, 2HES, 2HFM, 2HMD, 2HNO, 2HLW/
NAME=BLANK
Z=ZAS/10000
ITEMP=ZAS-Z*10000
A=ITEMP/10
EXCITE=ITEMP-A*10
IF(Z.GE.1.AND.Z.LE.103) GO TO 10
WRITE(6,10100) Z,ZAS
10100 FORMAT(' Z NUMBER NOT IN LIST = ',I5/' ZAS = ',I10)
RETURN
10 ELEMENT=NUCNAM(Z)
I=0
J=0
IP=3
IF(LONAME(2).EQ.HYPHEN) GO TO 20
LONAME(IP)=HYPHEN
IP=IP+1
20 IF(A.LT.100) GO TO 30
I=A/100
LONAME(IP)=DIGIT(I+1)
IP=IP+1
30 IF(A.LT.10) GO TO 40
J=A-100*I
J=J/10
LONAME(IP)=DIGIT(J+1)
IP=IP+1
40 K=A-100*I - 10*J
LONAME(IP)=DIGIT(K+1)
IP=IP+1
IF(EXCITE.NE.0) LONAME(IP)=M
IP=IP+1
IF(EXCITE.GT.1) LONAME(IP)=DIGIT(EXCITE+1)
UNPACK=NAME
RETURN
C
C THIS ENTRY PERFORMS THE REVERSE TRANSFORMATION,
C FROM THE ALPHANUMERIC NUCLIDE NAME TO THE ZAS NUMBER
C
ENTRY ID(NAMEIN)
NAME=NAMEIN
DO 100 I=1,103

```

```

      IF(ELEMNT.EQ.NUCNAM(I)) GO TO 110
100   CONTINUE
      WRITE(6,10200)ELEMNT
10200 FORMAT(' ELEMENT NAME: ',A2,' NOT FOUND')
      ID=0
      RETURN
110   IS=0
      IA=0
      IZ=1
      IP=4
      IF(LONAME(2).EQ.HYPHEN) IP=3
120   DO 140 I=1,10
      IF(LONAME(IP).EQ.DIGIT(I)) GO TO 150
140   CONTINUE
150   IA=IA*10+I-1
      IP=IP+1
      IF(LONAME(IP).EQ.LBLANK) GO TO 210
      IF(LONAME(IP).EQ.M) GO TO 180
      GO TO 120
180   IS=1
      IP=IP+1
      IF(LONAME(IP).EQ.LBLANK) GO TO 210
      DO 190 I=1,10
      IF(LONAME(IP).EQ.DIGIT(I)) GO TO 200
190   CONTINUE
200   IS=I-1
210   ID=IZ*10000+IA*10+IS
      RETURN
      END
      FUNCTION ISERCH (X, B, K, M)

```

```

C
C -----WRITTEN FOR ANEMOS CODE BY O. W. HERMANN, 9/4/80.
C
C -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 9/4/80.
C
C
C      ** CALLED FROM SUBR. SETDEC **
C
C APPLIES NORMAL TABLE SPLITTING METHOD, RETURNING POINTER...
C
C      DOUBLE PRECISION X, B(1)
C      DIMENSION K(1)
C
C      ISERCH = M + 1
C      SET FOR RETURN VALUE WHEN X NOT FOUND...
C
C      I = 1
C      N = M
22  J = (I + N) / 2
      IF (X .GT. B(J) ) GO TO 34
      IF (X .LT. B(J) ) GO TO 24
C      IF X = B(J), ISERCH = K(J) SINCE IT IS FOUND

```

```

      ISERCH = K(J)
      GO TO 40
24 IF (J .EQ. N) GO TO 40
C   J=N MEANS IT IS NOT FOUND IN ENTIRE TABLE - RETURN...
      N = J
C   LOOKS IN LOWER HALF OF SPLIT TABLE IN REPEAT AT STATEMENT 22....
      GO TO 22
34 IF (J .EQ. I) GO TO 40
C   NOT FOUND - RETURN
      I = J + 1
C   LOOKS IN UPPER HALF OF SPLIT TABLE IN REPEAT AT STATEMENT 22....
      GO TO 22
40 RETURN
      END
C   PROGRAM BLOCK DATA
      BLOCK DATA
C
C   -----WRITTEN FOR ANEMOS CODE BY O. W. HERMANN, 12/30/80.
C
C   -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C
      INTEGER*2 ELE(105)
      COMMON/LABEL/ ELE
      DATA ELE/' H','HE','LI','BE',' B',' C',' N',' O',' F','NE','NA','M
1G','AL','SI',' P',' S','CL','AR',' K','CA','SC','TI',' V','CR','MN
2','FE','CO','NI','CU','ZN','GA','GE','AS','SE','BE','KR','RB','SR'
3,' Y','ZR','NB','MO','TC','RU','RH','PD','AG','CD','IN','CN','SB',
4'TE',' I','XE','CS','BA','LA','CE','PR','ND','PM','SM','EU','GD',
5TB','DY','HO','ER','TM','YB','LU','HF','TA',' W','RE','OS','IR','P
6T','AU','HG','TL','PB','BI','PO','AT','RN','FR','RA','AC','TH','PA
7',' U','NP','PU','AM','CM','BK','CF','ES','FM','MD','NO','LR',
8 '04','05' /
      END
      SUBROUTINE LIBMAX (NRELIN, NEXPSZ, MXE, IDEXP, CONC, VELD, SOUR,
* DEPVEL, IDEXPT, IDREL, IDRELT, QSHORT, SYMBOL, JNUM, NUME,
* R, ITRAN, QIN, WET, DRY, NUCID, JINL,
* RVEC, RVECT, IC, JC, CPAK, IREF, NPACK, NUCOUT, NUMX, NAME)
C
C   -----WRITTEN FOR ANEMOS CODE BY O. W. HERMANN, 12/30/80.
C
C   -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C
      ** CALLED FROM PRIMUS CODE MAIN **
C
      MAKES IDREL, CONC, VELD FOR 2ND/3RD RECORDS.
      MAKES JINL FOR CHAIN RECORDS. WRITES WORKING LIBRARY.
C
C MOD. TO MAKE OUTPUT ORDER SAME AS RELEASED NUCLIDE INPUT ORDER.
C   BY OWH, 11/19/82
C
      DIMENSION JNUM(1), IDEXP(1), CONC(NEXPSZ,1), VELD(1),
* R(1), ITRAN(2,1),QIN(1), WET(1), DRY(NUMX,1),

```

```

* NUCID(1), JINL(1), NUNE(1), IREF(1),
* IC(1), JC(1), CPAK(1), RVECT(1), RVEC(1),
* SOUR(MXE,1), DEPVEL(MXE,1), IDEXPT(1), IDREL(1), IDRELT(1)
DOUBLE PRECISION QSHORT(1), SYMBOL(MXE,1), NAME, SYMK, SYMJ
COMMON /IONUMB/ NR, NREAD, KOUT, NTEMP
COMMON/NUCDAT/ NUNCHA, NEXP, NEXPT, KTRUNC, LMX, NSHORT, NREL

```

C

```

CALL RELESE(NRELIN, NREL, NEXP, MXE, QSHORT, IDEXP, SYMBOL, JNUM,
* NSHORT)

```

C

```

RELESE SETS UP JNUM...

```

C

```

DO 8 J = 1, NREL
RVEC(J) = 0.0
DO 8 I = 1, NEXP
8 CONC(I, J) = 0.0
DO 9 I = 1, NEXP
9 VELD(I) = 0.0

```

C

```

DO 14 K = 1, NREL
J = JNUM(K)
RVEC(K) = RVECT(J)
NUM = NUNE(J)
IDI = ID(SYMBOL(1, J) )
IDREL(K) = IDI
DO 14 N = 1, NUM
IF (N .NE. 1) IDI = ID(SYMBOL(N, J) )
DO 12 I = 1, NEXP
IF (IDEXP(I) .NE. IDI) GO TO 12
IF (N .EQ. 1) IREF(K) = I

```

C

C

```

FOUND ITH POSITION OF NTH READ IN SOURCE IN NUCLIDE IDRELK).

```

C

```

NOTE, MODEL PERMITS MISTAKEN NUC NOT IN CHAIN TO BE INPUT, SKIPS.

```

C

```

CONC(I, K) = SOUR(N, J)
IF (VELD(I) .NE. 0.0) GO TO 14
IF (DEPVEL(N,J) .EQ. 0.0) GO TO 14
VELD(I) = DEPVEL (N, J)
GC TO 14

```

```

12 CONTINUE

```

```

14 CONTINUE

```

```

REWIND NTEMP

```

```

REWIND NREAD

```

```

WRITE (NREAD)NUNCHA, KTRUNC, NPACK, NAME, NUCOUT

```

```

WRITE (NREAD) (IC(N), N = 1, NPACK), (JC(I), I = 1, NPACK)

```

```

DO 21 N = 1, NPACK

```

```

21 CPAK(N) = CONC( IC(N), JC(N) )

```

```

WRITE (NREAD) (CPAK(N), N = 1, NPACK), (VELD(I), I = 1, NEXP),

```

```

* (RVEC(J), J = 1, NREL)

```

```

WRITE (NREAD) NREL, NEXP, (IDREL(K), K = 1, NREL),

```

```

* (IDEXP(N), N = 1, NEXP)

```

```

IF (KTRUNC .LT. 1) GO TO 34

```

```

CALL RELESE (NRELIN, NRELT, NEXPT, MXE, QSHORT, IDEXPT,

```

```

* SYMBOL, JNUM, NSHORT)
DO 22 I = 1, NEXPT
  VELD (I) = 0.0
DO 22 J = 1, NRELT
22 CONC(I,J) = 0.0
DO 30 K = 1, NRELT
  J = JNUM(K)
  IDI = ID(SYMBOL(1, J) )
  IDRELT (K) = IDI
30 NUM = NUME(J)
  IF (KTRUNC .GT. 0)
    *WRITE (NREAD) NRELT,NEXPT,(IDRELT(K), K = 1, NRELT),
    * (IDEXPT(N),N = 1, NEXPT)
34 DO 100 NNN = 1, NUMCHA
  READ (NTEMP) L, LSQ, NCEXP, KSHORT, JSHORT, LOUT,
  * (DUM, M=1,L), KIND, JIND,
  * (NUCID(N), N = 1, L), (R(L1), L1 = 1, LSQ),
  * (QIN(L2), L2 = 1, L), (ITRAN(1,I),ITRAN(2,I), I = 1, LOUT),
  * (WET(K), K = 1, L), NUMX, ((DRY(JX,M), JX=1,NUMX), M=1,L)
  * , KID, JID, DLAMK, DLAMJ, SYMK, SYMJ
  K = 0
DO 40 I = 1, NREL
DO 38 J = 1, L
  IF (NUCID(J) .NE. IDREL(I) ) GO TO 38
  K = K + 1
  JINL(K) = J
GO TO 40
38 CONTINUE
40 CONTINUE
  LX = L
  NCREL = K
  QTOT=0.

```

C

```

  WRITE (NREAD) L, LSQ, NCEXP, NCREL, KSHORT, JSHORT,
  * (R(L1), L1=1,LSQ), (JINL(K1), K1=1,NCREL), (NUCID(N1), N1=1,L),
  * LOUT, QTOT, (QIN(L2), L2=1,L), (ITRAN(1,I),ITRAN(2,I),I=1,LOUT),
  * (WET(K), K = 1, LX), NUMX, ((DRY(JX, M), JX=1,NUMX), M=1,LX)
  * , KID, JID, DLAMK, DLAMJ, SYMK, SYMJ
  IF (KTRUNC .LT. 1) GO TO 100
  READ (NTEMP) L, LSQ, (R(I), I = 1, LSQ)
  K = 0
DO 70 I = 1, NRELT
DO 58 J = 1, L
  IF (NUCID(J) .NE. IDRELT(I) ) GO TO 58
  K = K + 1
  JINL(K) = J
GO TO 70
58 CONTINUE
70 CONTINUE
  NCREL = K
  NCEXP = L
  LX = L

```

```

      IF (KSHORT .GT. L) KSHORT = 0
      IF (JSHORT .GT. L) JSHORT = 0
      IF (KSHORT .GT. 0) NCEXP = NCEXP + 1
      IF (JSHORT .GT. 0) NCEXP = NCEXP + 1
      WRITE (NREAD) L, LSQ, NCEXP, NCREL, KSHORT, JSHORT,
      * (R(L1), L1=1,LSQ), (JINL(K1), K1=1,NCREL), (NUCID(N1), N1=1,L),
      * LOUT, QTOT, (QIN(L2), L2=1,L), (ITRAN(1,I),ITRAN(2,I),I=1,LOUT),
      * (WET(K), K = 1, LX), NUMX, ((DRY(JX, M), JX=1,NUMX), M=1,LX)
      * , KID, JID, DLANK, DLANK, SYMK, SYMJ
100 CONTINUE
      RETURN
      END
      SUBROUTINE MOVEIN (NR, MXE, NE, SYMBOL, XDAT, EV2, RELSYM,
      * EXPSYM, XINPUT, ETECT, RTECT, IEDIT)

```

C
C
C
C
C
C
C
C

** CALLED FROM PRIMUS CODE MAIN **

-----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.

MOVES EXTRA DATA INTO 2-D ARRAYS FOR LIBMAX...

```

      DOUBLE PRECISION SYMBOL(MXE,1), EXPSYM(1), RELSYM(1)
      DIMENSION XDAT(MXE,1), EV2(MXE,1), NE(1),
      * XINPUT(1), ETECT(1), RTECT(1)
      I = 0
      DO 20 N = 1, NR
      JR = NE(N)
      IF (RELSYM(N) .NE. EXPSYM(I+1) )
      * CALL XCHNG (JR, RELSYM(N), EXPSYM(I+1), XINPUT(I+1), ETECT(I+1))
      DO 10 J = 1, JR
      I = I + 1
      SYMBOL(J,N) = EXPSYM(I)
      XDAT(J,N) = XINPUT(I)
10 EV2(J,N) = ETECT(I)
      IF(IEDIT.NE.0)
      * WRITE(6,666) (N,J,SYMBOL(J,N),XDAT(J,N),EV2(J,N),J=1,JR)
666 FORMAT (1H ,2I4,2X,A8,1P2E17.4)
      20 CONTINUE
      RETURN
      END
      SUBROUTINE REDUCE (M, II, C, QIN, IND, NUCID, W, L, KSHORT,
      * JSHORT, KIND, JIND)

```

C
C
C
C
C
C
C
C

-----WRITTEN FOR ANEMOS CODE BY O. W. HERMANN, 12/30/80.

-----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.

** CALLED FROM SUBR. SETDEC **

DIMENSION C(M,M), QIN(L), IND(L), NUCID(L), W(1)

```

IJ = 0
I1 = 0
DO 8 I = II, L
I1 = I1 + 1
IND(I1) = IND(I)
QIN(I1) = QIN(I)
R = QIN(I1)
J1 = 0
DO 6 J = II, L
J1 = J1 + 1
IF (I1 .NE. J1) R = R + C(J,I)
6 C(J1, I1) = C(J, I)
IF (R .EQ. 0.0) IJ = I1 - 1
8 CONTINUE

C
C   DELETE NUCLIDES WITH NO PARENTS OR QIN-VALUE.
C
L = L - II + 1
IF (IJ .EQ. 0) GO TO 40
IF (IJ .EQ. L-1) GO TO 12
GO TO 14
C   REVISED FROM GO TO 22, 12/14/82 OWH
12 L = L - 1
IF (IJ .EQ. L) GO TO 40
14 DO 20 I = 2, L
R = QIN(I)
IJ = I - 1
DO 18 J = 1, IJ
18 R = R + C(J, I)
IF (R .EQ. 0.0) GO TO 22
C   FOR R=0 THERE ARE NO PARENTS OR OMITTED NUCLIDE...
20 CONTINUE
GO TO 40
C   DONE AT 40
C   NOW, NEED TO DELETE ELEMENT AT IJ + 1..
C
C   SCOOT COLUMNS ONLY, THRU IJ ROWS...
C
22 IK = IJ + 2
IF (IJ .EQ. L-1) GO TO 12
IF (IK .GT. L) CALL ERROR
DO 26 I = 1, IJ
C   SCOOT COLUMNS ONLY, THRU IJ ROWS...
DO 26 J = IK, L
26 C(J-1, I) = C(J, I)
DO 32 I = IK, L
C   NOW, SCOOT ROWS BACK, NOT COLUMNS FOR IJ COLUMNS...
DO 30 J = 1, IJ
30 C(J, I-1) = C(J, I)
C   NOW, SCOOT BOTH FOR REMAINING ELEMENTS....
DO 32 J = IK, L
32 C(J-1, I-1) = C(J, I)

```

```

DO 36 I = IK, L
  IF (KSHORT .EQ. 0) NUCID(I-1) = NUCID(I)
  IND(I-1) = IND(I)
36 QIN(I-1) = QIN(I)
  IF (KSHORT .EQ. 0) GO TO 12
  L = L - 1
C   ALWAYS DECREASE AFTER USED AT TOP OF DO LOOP
C
40 CONTINUE
  DCUT = 17.329
  LS = MIN0 (KSHORT, JSHORT)
  IF (LS .GT. 0) GO TO 70
  IF (L .EQ. 1) GO TO 70
  IJ = 0
  DO 60 I = 2, L
    ABSC = ABS( C(I,I) )
    IF ( ABSC .LT. DCUT) GO TO 60
    JJ = 0
    KK = I - 1
    DO 50 J = 1, KK
      IF (C(J, I) .EQ. 0.0) GO TO 50
      JJ = JJ + 1
      IF (C(I, I) / C(J, I) .LT. 2000.0) GO TO 60
      IF (JJ .GT. 1) IJ = J
      IF (JJ .GT. 1) GO TO 60
      BFJTOI = -C(J,I) / C(I,I)
      LJ = J
      IJ = I - 1
50 CONTINUE
    IF (IJ .EQ. 0) GO TO 60
C
C HALF-LIFE < 0.04 SEC, LAMDA RATIO > 2000 AND NO BRANCHING IS THE
C CRITERIA THAT HAS BEEN MET - DELETE NUC FROM MATRIX, BUT NEED TO
C PRESERVE TRANSITION TO DAUGHTER.....
C
    IST = I + 1
    DO 56 K = IST, L
      IF (C(I, K) .EQ. 0.0) GO TO 56
C
C WHEN ELEMENT IN SHORT-LIVED NUC COLUMN IS FOUND, ADD IT TO THE
C ELEMENT IN LONG-LIVED PARENT COLUMN OF SAME ROW TO MAKE TRANSITION
C FROM LONG-LIVED NUC DIRECTLY TO DAUGHTER OF THE SHORT-LIVED DAUGHTER
C BEING REMOVED.....
C
      C(LJ, K) = C(LJ, K) + C(I, K) * BFJTOI
C
56 CONTINUE
    GO TO 66
60 CONTINUE
    GO TO 70
C   WHEN IT FALLS THRU 60, NO MORE SHORT-LIVED NUCS TO BE REMOVED.
C

```



```

66 IF (KSHORT .GT. 0) JSHORT = KK
   IF (KSHORT .GT. 0) JIND = IND(KK+1)
   IF (KSHORT .EQ. 0) KIND = IND(KK+1)
   IF (KSHORT .EQ. 0) KSHORT = KK
   GO TO 22
70 CONTINUE
C  NOW, MOVE C-ARRAY INTO A COMPACT W-ARRAY OF SIZE L*L...
   K = 0
   DO 80 I = 1, L
   DO 80 J = 1, L
   K = K + 1
80 W(K) = C(J,I)
   RETURN
   END
   SUBROUTINE RELESE (NRELIN, NREL, NE, MXE, QSHORT, IDE, SYMBOL,
* JNUM, NSHORT)
C
C  ** CALLED FROM LIBMAK **
C
C  -----WRITTEN FOR ANEMOS CODE BY O. W. HERMANN, 12/30/80.
C
C  -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C
C
C  PURPOSE TO SETUP JNUM POINTERS TO RELEASED NUCS READ IN, SET NREL.
C
   INTEGER IDSH(20), IDE(1), JNUM(1)
   DOUBLE PRECISION SYMBOL(MXE, 1), QSHORT(1)
   COMMON /IONUMB/ NR, NREAD, KOUT, NTEMP
   IF (NSHORT .EQ. 0) GO TO 4
   DO 2 N = 1, NSHORT
2  IDSH(N) = ID (QSHORT(N) )
4  K = 0
   DO 40 I = 1, NE
   DO 24 J = 1, NRELIN
   IDN = ID (SYMBOL(1, J) )
C
C  CAN IDN BE RELEASED,  $\beta$  IS IT NEXT IN ORDER...
C
   IF (IDE(I) .NE. IDN) GO TO 24
   IF (NSHORT .EQ. 0) GO TO 22
   DO 20 N = 1, NSHORT
   IF (IDSH(N) .EQ. IDN) GO TO 28
20 CONTINUE
22 K = K + 1
   JNUM(K) = J
   GO TO 40
24 CONTINUE
   GO TO 40
28 WRITE(KOUT, 32) QSHORT(N)

```

```

32 FORMAT(1H0,A8,' IS A VERY SHORT-LIVED NUCLIDE.  MODEL EXCLUDES IT
*FROM DECAY MATRIX AND SETS EQUAL TO PARENT ACTIVITY.'/1H ,
*'IT IS INCLUDED AS AN EXPOSURE NUCLIDE, BUT WAS DELETED AS A RELEA
*SED NUCLIDE. ' )
40 CONTINUE
  NREL = K
  RETURN
  END
  SUBROUTINE SETDEC(NTOT, NTOT1, MAXMX, MAX2, NUCDRY, NNUC, NUMX,
* NUCOUT, BNC,
* ANC, NN, CARRAY, QNAM, INNP, NDONE, DLAM, ITRAN, JTRAN, QIN, IND,
* NUCID, WET, DRY, WTR, DRR, S, ICHAIN, R, IDEXP, IDEXPT, QSHORT,
* IC, JC, NNN)
C
C -----WRITTEN FOR ANEMOS CODE BY O. W. HERMANN, 12/30/80.
C
C -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C
C ** CALLED FROM PRIMUS CODE MAIN **
C
C ***NOTE***
C   THIS SUBROUTINE WAS WRITTEN FIRST FOR THE ANEMOS CODE.  THEN, IN
C   ORDER TO EFFICIENTLY ADAPT IT TO THE PRIMUS CODE, IT WAS COMPLETELY
C   COPIED AND MADE A PART OF PRIMUS.  THIS ENHANCED ITS RELIABILITY
C   WITHOUT DUPLICATE DEBUGGING PROCEDURES AND SAVED CONSIDERABLE
C   PROGRAMMER TIME.  THUS, CERTAIN VARIABLES NEEDED FOR ANEMOS HAVE BEEN
C   RETAINED, ALTHOUGH THEY ARE MEANINGLESS TO PRIMUS.  THESE VARIABLES
C   INCLUDE: WET, DRY, WTR, DRR, ITRAN AND JTRAN.  PLEASE IGNORE THEM
C   WHEN READING OR USING THE SUBROUTINE FOR PRIMUS.
C   ..... OWH
C
C MOD. TO MAKE OUTPUT ORDER SAME AS RELEASED NUCLIDE INPUT ORDER.
C   (USES ORDER OF 1ST NUC., IF 2 OR MORE NUCS IN SAME CHAIN.)
C   BY OWH, 11/19/82
C
  DOUBLE PRECISION QNAM(NNUC), BNC(NTOT), ANC(NTOT)
  * , QSHORT(1), BLANK, SYMK, SYMJ
  DIMENSION NN(NTOT1), CARRAY(NNUC), INNP(NUCDRY), DLAM(NTOT1),
  * ITRAN(2,1), JTRAN(2,1), QIN(MAXMX), IND(MAXMX), NUCID(MAXMX),
  * IC(1), JC(1), ICHAIN(NNUC), R(1), IDEXP(1), IDEXPT(1),
  * WET(MAXMX), DRY(NUMX,MAXMX), S(MAX2),WTR(NUCDRY),DRR(NUMX,NUCDRY)
  LOGICAL*1 NDONE(NNUC)
  LOGICAL CALDRY
  COMMON /IONUMB/ NR, NREAD, KOUT, NTEMP
  COMMON/NUCDAT/ NUMCHA, NEXP, NEXPT, KTRUNC, LMX, NSHORT, NREL
  DATA BLANK /' ' /
  NUCOUT = 0
  NUMCHA = 0
  NUCTOT = 0
  NNN = 0
  III = 0
  JJJ = 0

```

```

NU = 0
LZ = 0
KK = 0
NSHORT = 0
READ (NR) (BNC(I), I = 1, NTOT)
READ (NR) ( NN(I), I = 1, NTOT)
NN(NTOT1) = NTOT1
DO 20 K = 1, NUCDRY
20 INNP(K) = ISERCH (QNAM(K), BNC, NN, NTOT)
C
C   ISERCH RETURNS POINTER OF INDEX TO NEXT READ OF NN, WHICH GIVES
C   THE CHAIN NO. OF NUCLIDE...
C
C
READ (NR) ( NN(I), I = 1, NTOT)
READ (NR) (DLAM(I), I = 1, NTOT)
READ (NR) (ANC(I), I = 1, NTOT)
DLAM(NTOT+1) = 0.0
NN (NTOT+1) = 0
C   THE FIRST NNUC OF NUCDRY ARE THE RELEASED NUCS.
DO 30 K = 1, NNUC
NDONE(K) = .FALSE.
I = INNP(K)
30 ICHAIN(K) = NN(I)
C
REWIND NTEMP
NUC = 0
DO 32 K = 1, 4
32 READ(NR)
NCH1 = 0
C
C   NOW, MULTI-LINK CHAINS ARE PROCESSED
C
C   B I G L O O P.....S T A R T S   H E R E....
C
DO 100 KNUM = 1, NNUC
IF (NDONE(KNUM) ) GO TO 100
C
IF (ICHAIN(KNUM) .NE. 0) GO TO 40
K = KNUM
NUC = NUC + 1
NU = NU + 1
L = 1
I = INNP(K)
NID = ID(ANC(I) )
IDEXP(NU) = NID
C   FUNCTION ID CONVERTS BCD TO Z-A-S ID NO.
NDONE(K) = .TRUE.
DLAMDI = -DLAM(I)
NNN = NNN + 1
III = III + 1
JJJ = JJJ + 1

```

```

      IF (III .GT. 4000) CALL ERROR
      IF (JJJ .GT. 4000) CALL ERROR
      IC(NNN) = III
      JC(NNN) = JJJ
      KID = 0
      JID = 0
      DLAMK = 0.0
      DLAMJ = 0.0
      SYMK = BLANK
      SYMJ = BLANK
C
      WRITE (NTEMP)L,L,L,LZ,LZ,L,I,L,L, NID, DLAMDI, CARRAY(K), L, L,
* WTR(K), NUMX, (DDR(JX, K), JX = 1, NUMX)
* , KID, JID, DLAMK, DLAMJ, SYMK, SYMJ
      NUMCHA = NUMCHA + 1
      GO TO 100
40 NCH = ICHAIN(KNUM)
      NSKIP = NCH - NCH1 - 1
      NCH1 = NCH
      IF (NSKIP .EQ. 0) GO TO 46
      IF (NSKIP .GT. 0) GO TO 42
      REWIND NR
      NSKIP = NCH + 9
42 DO 44 NSK = 1, NSKIP
44 READ(NR)
46 NUMCHA = NUMCHA + 1
C
C READ RECORD OF CHAIN INFO.
C
C
      READ (NR) LSQ, L, N1, (IND(I),I=1,L), (S(M), M=1,LSQ)
      IF (N1 .NE. NCH) CALL ERROR
      DO 54 I = 1, L
54 QIN(I) = 0.0
C SET VALUE OF QIN  $\beta$  REDUCE MATRIX...
C
      II = L
      DO 60 M = 1, NNUC
      IF (ICHAIN(M) .NE. NCH) GO TO 60
      DO 56 I = 1, L
      IF (IND(I) .NE. INNP(M) ) GO TO 56
      NDONE(M) = .TRUE.
      QIN(I) = CARRAY(M)
      II = MIN0 (I, II)
      GO TO 60
56 CONTINUE
      CALL ERROR
60 CONTINUE
C
C NOW, SET NUCID FROM FUNCTION ID..
C

```

```

      J = 0
      DO 61 I = II, L
      J = J + 1
      K = IND(I)
61  NUCID(J) = ID ( ANC(K) )
      M = L
      KSHORT = 0
      JSHORT = 0
      CALL REDUCE (M, II, S, QIN, IND, NUCID, R, L, KSHORT, JSHORT,
* KIND, JIND)
      LSQ = L * L
C
C   SET WET AND DRY ARRAYS FOR WRITE...
      KST = NNUC + 1
      IF (KST .GT. NUCDRY) GO TO 72
      DO 64 J = 1, L
      WET(J) = 0.0
      DO 62 JX = 1, NUMX
62  DRY(JX,J) = 0.0
      DO 64 K = KST, NUCDRY
      IF (IND(J) .NE. INNP(K) ) GO TO 64
      WET(J) = WTR(K)
      DO 63 JX = 1, NUMX
63  DRY(JX, J) = DRR(JX, K)
64  CONTINUE
C
C   SET ITRAN, JTRAN & NUCID
C
72  LOUT = 0
      DO 80 J = 1, L
      IF (QIN(J) .EQ. 0.0) GO TO 80
      DO 78 I = J, L
      LOUT = LOUT + 1
      IF (KSHORT .NE. I) GO TO 75
      ITRAN(2,LOUT) = J
      ITRAN(1,LOUT) = I
      LOUT = LOUT + 1
      GO TO 76
75  IF (JSORT .NE. I) GO TO 76
      ITRAN(2,LOUT) = J
      ITRAN(1,LOUT) = I
      LOUT = LOUT + 1
76  ITRAN(2,LOUT) = J
78  ITRAN(1,LOUT) = I
80  CONTINUE
      JJ = 0
      LL = 0
      NCEXP = L
      IF (KSHORT .GT. 0) NCEXP = NCEXP + 1
      IF (JSORT .GT. 0) NCEXP = NCEXP + 1
      KS = KSHORT
      JS = JSORT

```

```

      IF (KS .EQ. 0) KS = 200
      IF (JS .EQ. 0) JS = 200
      JOUT = 0
      IF (III + NCEXP .GT. 4000) CALL ERROR
      IF (JJJ + L .GT. 4000) CALL ERROR
      DO 86 J = 1, L
      IF (QIN(J) .EQ. 0.0) GO TO 86
      JJJ = JJJ + 1
C      JJJ IS INCREASED BY ONE, FOR EACH RELEASED NUCLIDE.
C
      J1 = J
      IF (J .GT. KS) J1 = J1 + 1
      IF (J .GT. JS) J1 = J1 + 1
      JOUT = JOUT + 1
      DO 84 I = J1, NCEXP
      NNN = NNN + 1
      IC(NNN) = III + I
84 JC(NNN) = JJJ
C      JJJ WAS INCREASED BY ONE, FOR EACH RELEASED NUCLIDE.
C
86 CONTINUE
      III = III + NCEXP
      DO 90 K = 1, NCEXP
      NU = NU + 1
      IDEXP(NU) = NUCID(K)
      LL = LL + 1
      IF (KTRUNC .GT. 0 .AND. LL .LE. LMX) IDEXPT(KK + LL) = NUCID(K)
      IF (K .EQ. 1) GO TO 88
      IF (K-1 .EQ. KSHORT) GO TO 90
      IF (K-1 .EQ. JSHORT) GO TO 90
88 JJ = JJ + 1
      NUCID(JJ) = NUCID(K)
90 CONTINUE
C
      DO 94 M = 1, NNUC
      IF (ICHAIN(M) .NE. NCH) GO TO 94
      DO 92 I = 1, L
      IF (IND(I) .EQ. INMP(M) ) GO TO 94
92 CONTINUE
      NSHORT = NSHORT + 1
      IF (NSHORT .GT. 20) CALL ERROR
      QSHORT ( NSHORT ) = QNAM(M)
      QNAM(M) = BLANK
94 CONTINUE
      KK = KK + LL
      NEXPT = KK
      IF (KTRUNC .LE. 0) GO TO 98
      LN = MIN0 (L, LMX)
      DO 96 J = 1, LN
      DO 96 I = 1, LN
      K = L * (J - 1) + I
      N = LN * (J - 1) + I

```

```

96 S(N) = R(K)
C  WRITE CHAIN RECORD TO BE READ BY SUBRS. DPLETE & DRYDEP...
C
98 KID = KSHORT
   JID = JSHORT
   DLAMK = 0.0
   DLAMJ = 0.0
   SYMK = BLANK
   SYMJ = BLANK
   IF (KID .EQ. 0) GO TO 99
   DLAMK = DLAM(KIND)
   SYMK = ANC(KIND)
   KID = ID(SYMK)
   IF (JID .EQ. 0) GO TO 99
   DLAMJ = DLAM(JIND)
   SYMJ = ANC(JIND)
   JID = ID(SYMJ)
99 WRITE (NTEMP) L, LSQ, NCEXP, KSHORT, JSHORT, LOUT,
   * (IND(L3), L3 = 1, L), KIND, JIND,
   * (NUCID(N), N = 1, L), (R(L1), L1 = 1, LSQ),
   * (QIN(L2), L2 = 1, L), (ITRAN(1,I), ITRAN(2,I), I = 1, LOUT),
   * (WET(K), K = 1, L), NUMX, ((DRY(JX,M), JX=1,NUMX), M=1,L)
   * , KID, JID, DLAMK, DLAMJ, SYMK, SYMJ
C
   NUCTOT = NUCTOT + NCEXP
   NUCOUT = NUCOUT + LOUT
   IF (KTRUNC .LE. 0) GO TO 100
   LNSQ = LN * LN
   WRITE (NTEMP) LN, LNSQ, (S(I), I = 1, LNSQ)
100 CONTINUE
   NUCOUT = NUCOUT + NUC
   NUCTOT = NUCTOT + NUC
   NEXP = NUCTOT
   REWIND NTEMP
   RETURN
   END
   SUBROUTINE SORTP(AE, L, N, NOUT)
C   ** AE-ARRAY AND L-POINTER-ARRAY IS SORTED, ASCENDINGLY **
   DIMENSION AE(1), L(1)
C   N IS NO. OF ITEMS IN AE-ARRAY
C   L IS AN ARRAY ASSOCIATED WITH AE (MAY BE POINTERS) ***
C
   DOUBLE PRECISION AE, AT
   M=N
2  M=M/2
   IF (M.LE.0) GO TO 8
   K=N - M
   DO 4 J=1,K
     I = J
3  IM = I + M
   IF (AE(I).LE.AE(IM) ) GO TO 4
   AT = AE(I)

```

```

      AE(I) = AE(IM)
      AE(IM) = AT
      LP = L(I)
      L(I) = L(IM)
      L(IM) = LP
      I = I - M
      IF (I.GT.0) GO TO 3
4    CONTINUE
      GO TO 2
8    KF = 0
10   JF = KF
      DO 20 JJ=1,N
      IF (JF .GT. 0) WRITE(NOUT,18) L(JJ),AE(JJ)
      IF ( JJ .EQ. N) GO TO 20
      IF (AE(JJ+1) .LT. AE(JJ) ) KF = 1
      IF (JF .EQ. 0) GO TO 20
      IF (AE(JJ+1) .LT. AE(JJ) ) WRITE(NOUT,19)
18  FORMAT (1X, I10,1X,A8)
19  FORMAT ( 20X, 6HERROR  )
20  CONTINUE
      IF (JF .EQ. 1) CALL ERROR
      IF (KF .EQ. 1) GO TO 10
      RETURN
      END
      SUBROUTINE WRMAT(IEDIT, N1, L, S)
C
C    ** CALLED FROM EDIT **
C    -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C
C    ..WRITES DECAY MATRIX OF EACH CHAIN ON UNIT IEDIT, IF IEDIT > 1..
C
      REAL S(L,L)
C
      WRITE (IEDIT, 2) N1, L, L
2    FORMAT (1H0, ' DECAY MATRIX APPLIED FOR CHAIN NUMBER', I3,
      * ' FOR SIZE', I3, ' X', I3 /1X)
C
      WRITE (IEDIT, 8) ((I, J, S(J,I), J=1,L), I = 1, L)
C
C    J POINTS TO COLUMN ELEMENTS OF ROW I - OR PRINT ROWS ARE FOR ROWS.
C
8    FORMAT ( T5, I2,   T3, 'I=', T7, ' ,   J=', I2, 1PE12.4 ,
      *          T30, I2,   T28, 'I=', T32, ' ,   J=', I2, 1PE12.4 ,
      *          T55, I2,   T53, 'I=', T57, ' ,   J=', I2, 1PE12.4 ,
      *          T80, I2,   T78, 'I=', T82, ' ,   J=', I2, 1PE12.4 ,
      *          T105, I2,  T103, 'I=', T107, ' ,   J=', I2, 1PE12.4 )
      RETURN
      END
      SUBROUTINE WRTLIB(NR, IO)
      REAL*8 ANC(500), BNC(500), D1, D2
      DIMENSION NP(500), NCHAIN(500), DLAM(500), BF1(500),
      - BF2(500), INDAU1(500), INDAU2(500), IND(40),

```



```

- S(40,40),FORM(9),OFFSET(6)
  INTEGER OUT(6)
  DATA FORM/'(1X','I2,1','X,A8','1PE',
- '10.3','41X','1P7','E11.','3)'/,
- OFFSET/'41X','52X','63X','74X','85X','96X'/
  READ(NR)NTOT,NCHTOT,MAXMX
  READ(NR)(BNC(I),I=1,NTOT)
  READ(NR)(NP(I),I=1,NTOT)
  READ(NR)(NCHAIN(I),I=1,NTOT)
  READ(NR)(DLAM(I),I=1,NTOT)
  READ(NR)(ANC(I),I=1,NTOT)
  READ(NR)(BF1(I),I=1,NTOT)
  READ(NR)(BF2(I),I=1,NTOT)
  READ(NR)(INDAU1(I),I=1,NTOT)
  READ(NR)(INDAU2(I),I=1,NTOT)
  WRITE(6,1000)NTOT,NCHTOT,MAXMX
1000 FORMAT('1THE DECAY LIBRARY WRITTEN BY PRIMUS CONTAINS: '/
- ' ',I3,' NUCLIDES'/' ',I3,' CHAINS'/'
- ' ',I3,' MAXIMUM CHAIN LENGTH')
  IF(I0.LT.2) GO TO 999
  IPAGE=2
  WRITE(6,1100)
1100 FORMAT('1 I BNC NP NCHAIN DLAM',6X,
- 'ANC',6X,'BF1',6X,'BF2 INDAU1 INDAU2'/' )
  DO 5 I=1,NTOT
    IPAGE=IPAGE+1
    IF(IPAGE.LT.60) GO TO 5
    WRITE(6,1100)
    IPAGE=2
  5 WRITE(6,1200)I,BNC(I),NP(I),NCHAIN(I),DLAM(I),
- ANC(I),BF1(I),BF2(I),INDAU1(I),INDAU2(I)
1200 FORMAT(' ',I3,1X,A8,2I4,2X,1PE10.3,2X,A8,OP2F9.6,I4,I7)
999 IPAGE=2
  WRITE(6,2000)
2000 FORMAT('1','SINGLE NUCLIDES'/' ' ',6X,'DECAY CONSTANT'/'
- ' NAME PER SECOND'/' )
  DO 20 I=1,NTOT
    IF(NCHAIN(I).NE.0) GO TO 20
    WRITE(6,3000)ANC(I),DLAM(I)
3000 FORMAT(' ',A8,1PE9.2)
    IPAGE=IPAGE+1
    IF(IPAGE.LT.60) GO TO 20
    WRITE(6,2000)
    IPAGE=2
  20 CONTINUE
  WRITE(6,3500)
3500 FORMAT('1NUCLIDE CHAINS. THE DECAY MATRIX IS PRINTED WITH',
- ' THE DIAGONAL IN THE COLUMN TO THE FAR RIGHT. MOST ZERO',
- ' ENTRIES ARE OMITTED.')
  WRITE(6,4500)
  IPAGE=4
  DO 40 I=1,NCHTOT

```

```

      READ(10)LSQ,L,N,(IND(I1),I1=1,L),((S(J1,J2),J2=1,L),J1=1,L)
      IPAGE=IPAGE+4+L
      IF(IPAGE.LT.60) GO TO 25
      WRITE(6,3500)
      WRITE(6,4500)
      IPAGE=L+8
25     WRITE(6,4000)I
4000    FORMAT(/'OCHAIN NUMBER',I4)
4500    FORMAT('O',9X,'DECAY CONSTANT ',15('-'),
-       'DAUGHTERS',15('-'),10X,'DECAY MATRIX'/
-       '      NAME   PER SECOND ',2(' NAME  BRANCH. FRAC. '))
      NTITLE=MINO(L,6)
      DO 27 K=2,NTITLE
        K1=1-K
27      OUT(NTITLE+K1)=K1
      NT=NTITLE-1
      WRITE(6,4700)(OUT(K1),K1=1,NT)
4700    FORMAT(' ',67X,'D',5(I2,8X,'D'))
      DO 40 II=1,L
        J=IND(II)
        DO 30 K=1,II
          IF(S(II,K).NE.0.) GO TO 35
30      CONTINUE
        K=II
35      NUMSP=MINO(L,6)+K-II
      IF(NUMSP.LT.1.OR.NUMSP.GT.6) WRITE(6,7000)NUMSP
7000    FORMAT(' NUMSP=',I3,'μ MUST BE 1-6')
      IF(NUMSP.LT.1.OR.NUMSP.GT.6) NUMSP=1
      FORM(6)=OFFSET(NUMSP)
      IF(BF1(J).NE.0.) D1=ANC(INDAU1(J))
      IF(BF2(J).NE.0.) D2=ANC(INDAU2(J))
      WRITE(6,FORM) II,ANC(J),DLAM(J),(S(II,M),M=K,II)
      IF(BF1(J).NE.0.) WRITE(6,5000)D1,BF1(J)
5000    FORMAT('+',25X,A8,F9.6)
      IF(BF2(J).NE.0.) WRITE(6,6000)D2,BF2(J)
6000    FORMAT('+',45X,A8,F9.6)
40      CONTINUE
      RETURN
      END
      SUBROUTINE XCHNG(JR, REL, SYM, X, E)
C
C -----WRITTEN FOR PRIMUS CODE BY O. W. HERMANN, 12/30/80.
C
C ** CALLED FROM MOVEIN **
C
      COMMON /IONUMB/ NR, NREAD, KOUT, NTEMP
      REAL*8 REL, SYM(1)
      REAL X(1), E(1)
C
      IF (JR .LE. 1) GO TO 20
      XA = X(1)
      EA = E(1)

```

```

DO 8 I = 2, JR
IF (REL .NE. SYM(I) ) GO TO 8
X(1) = X(I)
E(1) = E(I)
X(I) = XA
E(I) = EA
SYM(I) = SYM(1)
SYM(1) = REL
GO TO 30
8 CONTINUE
20 WRITE (KOUT, 22) REL
22 FORMAT( 1H1, 20(1H*) , ' INPUT ERROR' /1H0, A8,
* 'IS NOT IN THE PROPER POSITION IN EXPSYM-ARRAY. CASE STOPS.')
STOP 101
30 RETURN
END

```

.(11) DESTINATION INOPERATIVE #2

Appendix D

APPENDIX D: Method of Extending Use of the Code

The matrix operator method for computing radiologic decay products which is described in this document could be applied to other problems without using the full methodology of the CRRIS. This would require three parts of the CRRIS: the PRIMUS code, the DRALIST-CRRIS/1 Decay Library, and the decay package from TERRA or ANDROS. While the decay package subroutines in ANEMOS could be used, those in TERRA and ANDROS are more appropriate. The following list includes the specific subroutines that make up the decay package:

DECAYC	- control routine called
DECAYI	- called prior to DECAYC if concentration array is packed, as in ANEMOS data (ISTYP=2)
BACK	
CONVRT	
DECAIT	
DECDC	
DECEX	
DECP	
EXTRAC	
PACKIT	
PACKY	
UNPAKY	

The nature of the coding permits the conversion to other applications with a relatively small amount of effort. The comments at the beginning of subroutine DECAYC provide definitions of the parameters passed to the decay package.

The parameters passed in the call to DECAYC or DECAYI are:

NEXP	number of exposure nuclides,
CONC(NEXP)	initial concentrations (see ICTYP and ISTYP below),
REMLAM(NEXP)	additional rate constants (see IDIAG below)
CRATE(NEXP)	scaling factors for CONC (see IMULT below).

The parameters returned are:

VELD(NEXP) deposition velocities,
 IC(NEXP) positions of nuclides in chains,
 JC(NEXP) chain numbers of nuclides,
 CPAK(NEXP) concentrations read from unit NCONC.

The calculations in the decay package are controlled by the flags in COMMON/TIMECM/. The following flags in that common must be set prior to calling the decay package:

TIME the time (in seconds) over which decay occurs

ICTYP =0, decay computed with initial source term, no deposition;
 CONC should be in units of activity (Ci or Bq)
 =1, decay computed with no source term, constant deposition;
 CONC should be in units of activity/time (Ci/s or Bq/s)

ISTYP =0, source read from unit NCONC (possibly ANEMOS)
 =1, source read from unit NPRIME, written by PRIMUS
 =3, source passed in array CONC via subroutine call

IMULT =0, no multiplication of CONC
 =1, CONC multiplied by VELD, read from unit NCONC
 =2, CONC multiplied by CRATE, passed via subroutine call

IDIAG =0, decay matrix diagonal unmodified
 =1, array REMLAM, passed via subroutine call, added to
 the diagonal

NPRIME unit number from which PRIMUS file is read

NCONC unit number from which concentrations (from ANEMOS, TERRA,
 etc.) are read

KOUT unit number for printed output from decay package

NREL number of released nuclides

NEXP number of exposure nuclides

KTRUNC =0, chains are not truncated

=1, chains are truncated.

Some additional reading of the program may be required to incorporate the decay package into another system.

INTERNAL DISTRIBUTION

1-5. C. F. Baes III	32. T. W. Oakes
6-10. C. L. Begovich	33. W. F. Ohnesorge
11. M. A. Bell	34. S. Ohr
12. R. O. Chester	35. D. C. Parzyck
13. S. J. Cotter	36. M. R. Patterson
14. C. C. Coutant	37. C. R. Richmond
15. K. L. Daniels	38. P. S. Rohwer
16. K. F. Eckerman	39. R. D. Sharp
17. C. S. Fore	40-44. A. L. Sjoreen
18. H. Gerstner	43. ORNL Central Research Library
19-23. O. W. Hermann	44. ORNL, Y-12 Technical Library
24. S. G. Hildebrand	45. ORNL Patent Section
25. S. V. Kaye	46-50. Laboratory Records Department
26. D. C. Kocher	51. Laboratory Records, ORNL-RC
26-30. C. W. Miller	52. RSIC Library
31. B. D. Murphy	

EXTERNAL DISTRIBUTION

- 53. Office of Assistant Manager, Energy Research and Development, DOE-ORO, Oak Ridge, Tennessee 37831
- 54. L. R. Anspaugh, Lawrence Livermore Laboratory, P. O. Box 808, Livermore, California 94550
- 55. D. A. Baker, Battelle-Pacific Northwest Laboratories, P. O. Box 999, Richland, Washington 99352
- 56. Hugh W. Church, Sandia Laboratory, Division 5333, Albuquerque, New Mexico 87115
- 57. Frank J. Congel, U.S. Nuclear Regulatory Commission, Washington, D.C. 20555
- 58. T. V. Crawford, Savannah River Laboratory, Aiken, South Carolina 29801
- 59. R. G. Cuddihy, Inhalation Toxicology Research Institute, 5200 Gibson Blvd., S.E., Albuquerque, New Mexico 87108
- 60. F. L. Culler, Electric Power Research Institute, P. O. Box 10412, Palo Alto, California 94303
- 61. Ray Dickson, National Oceanic and Atmospheric Administration, 550 Second Street, Idaho Falls, Idaho 93401
- 62. G. G. Eichholz, School of Nuclear Engineering, Georgia Institute of Technology, Atlanta, Georgia 20332

63. Carl V. Gogolak, U.S. Department of Energy, Environmental Measurements Laboratory, 376 Hudson Street, New York, New York 10014
64. J. W. Healy, Health Physics Division, Los Alamos Scientific Laboratory, P. O. Box 1663, Los Alamos, New Mexico 97544
65. Bruce Hicks, Director, Air Resources Atmospheric Turbulence and Diffusion Laboratory, National Oceanic and Atmospheric Administration, Oak Ridge, Tennessee 37830
66. Wayne Lowder, U.S. Department of Energy, Washington, D.C. 20545
67. W. A. Mills, U.S. Nuclear Regulatory Commission, Washington, D.C. 20555
68. Dade Moeller, School of Public Health, Harvard University, 677 Huntington Ave., Boston, Mass. 02115
- 69-73. Christopher B. Nelson, Office of Radiation Programs, U.S. Environmental Protection Agency, Washington, D.C. 20460
74. Yook C. Ng, Biomedical and Environmental Research Division, Lawrence Livermore Laboratory, P. O. Box 808, Livermore, California 94550
75. J. K. Soldat, Pacific Northwest Laboratories, Battelle Memorial Institute, Richland, Washington 99352
76. W. L. Templeton, Ecosystems Department, Battelle-Pacific Northwest Laboratories, Richland, Washington 99352
77. E. C. Watson, Battelle-Pacific Northwest Laboratories, P. O. Box 999, Richland, Washington 99352
78. F. W. Whicker, Department of Radiology and Radiation Biology, Colorado State University, Fort Collins, Colorado 80521
79. W. H. Wilkie, Tennessee Valley Authority, River Oaks Building, Muscle Shoals, Alabama 35660
- 80-106. Technical Information Center, P. O. Box 62, Oak Ridge, Tennessee 37831