

SANDIA REPORT

SAND96-2736 • UC-721

Unlimited Release

Printed June 1997

RECEIVED

JUL 14 1997

OSTI

JUL 2 1997

Automation of ORIGEN2 Calculations for the Transuranic Waste Baseline Inventory Database Using a Pre-Processor and a Post-Processor

Jennifer Liscum-Powell

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-84AL85000.

Approved for public release; distribution is unlimited.



Sandia National Laboratories

SF2900Q(8-81)

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: 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, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. 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, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A08
Microfiche copy: A01

Automation of ORIGEN2 Calculations for the Transuranic Waste Baseline Inventory Database Using a Pre-Processor and a Post-Processor

Jennifer Liscum-Powell
Nuclear Safety and Systems Analysis
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1175

ABSTRACT

The purpose of the work described in this report was to automate ORIGEN2 calculations for the Waste Isolation Pilot Plant (WIPP) Transuranic Waste Baseline Inventory Database (WTWBID); this was done by developing a pre-processor to generate ORIGEN2 input files from WWBID inventory files and a post-processor to remove excess information from the ORIGEN2 output files. The calculations performed with ORIGEN2 estimate the radioactive decay and buildup of various radionuclides in the waste streams identified in the WTWBID. The resulting radionuclide inventories are needed for performance assessment calculations for the WIPP site.

The work resulted in the development of PreORG, which requires interaction with the user to generate ORIGEN2 input files on a site-by-site basis, and PostORG, which processes ORIGEN2 output into more manageable files. Both programs are written in the FORTRAN 77 computer language. After running PreORG, the user will run ORIGEN2 to generate the desired data; upon completion of ORIGEN2 calculations, the user can run PostORG to process the output to make it more manageable. All the programs run on a 386 PC or higher with a math co-processor or a computer platform running under VMS operating system. The pre- and post-processors for ORIGEN2 were generated for use with Rev. 1 data of the WTWBID and can also be used with Rev. 2 and 3 data of the TWBID (Transuranic Waste Baseline Inventory Database).

ACKNOWLEDGMENTS

The author wishes to thank Lawrence Sanchez, Org. 6848, and Leo Rahal, Geocenters Inc., for their technical review and comments regarding this document. The author also wishes to thank Molly Minahan, Tech Reps, Inc., for her assistance as a technical editor and writer, and Debbie Rivard and Jackie Ripple, Tech Reps Inc., for their expert word processing. Carol Crawford, 4415, provided a comprehensive reference check of this document.

CONTENTS

1.0 INTRODUCTION	1
2.0 DESCRIPTION OF ORIGEN2	5
2.1 ORIGEN2 Data Libraries	5
2.2 Characteristics ORIGEN2 Calculates	6
2.3 ORIGEN2 Input Structure	6
2.3.1 Entering Initial Nuclide Concentrations	7
2.4 Calculations Performed by ORIGEN2.....	7
2.5 Method of Solving Differential Equations in ORIGEN2	6
3.0 DESCRIPTION OF PREORG.....	11
3.1 Running PreORG.....	15
4.0 RUNNING ORIGEN2.....	17
5.0 DESCRIPTION OF POSTORG	19
5.1 Running PostORG	20
6.0 USING PREORG, ORIGEN2, AND POSTORG.....	21
6.1 Running PreORG, ORIGEN2, and PostORG on a PC.....	21
6.2 Running PREORG, ORIGEN2, and POSTORG on a VMC computer platform.....	22
6.3 Cautions to Observe when Running PreORG, ORIGEN2, and PostORG	22
7.0 SUMMARY.....	27
8.0 FURTHER READING	29
9.0 REFERENCES	31
APPENDIX A: SAMPLE PREORG.FOR DATA INPUT FILE.....	A-1
APPENDIX B: ORIGEN2.BAT FILE LISTING FOR A PC.....	B-1
APPENDIX C: ORIGEN2 COMMAND DESCRIPTIONS AND SAMPLE ORIGEN2 INPUT FILE CREATED BY PREORG	C-1
APPENDIX D: OR1_PREORG.FOR PROGRAM LISTING.....	D-1
APPENDIX E: PERSONAL COMMUNICATION WITH SAYAN CHAKRABORTI OF SCIENCE APPLICATIONS INTERNATIONAL CORPORATION BY JENNIFER POWELL ON JULY 26, 1995	E-1
APPENDIX F: OR3_POSTORG.FOR PROGRAM LISTING	F-1

APPENDIX G: SAMPLE OUTPUT FILES FROM POSTORG	G-1
--	-----

Figures

Figure 6.2.1 Sample command file used to run PreORG, ORIGEN2, and PostORG.	23
---	----

Tables

Table 1. ORIGEN2 Input Files Created by PreORG	12
--	----

DISCLAIMER

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

1.0 INTRODUCTION

The purpose of this work was to automate ORIGEN2 calculations for the Waste Isolation Pilot Plant (WIPP) Transuranic Waste Baseline Inventory Database (WTWBID). This automation was accomplished by developing a pre-processor that generates complete ORIGEN2 input files from Baseline Inventory Database (BID) data files and a post-processor that processes the ORIGEN2 output files by extracting only the useful information.

There are two printed versions of the radionuclide inventory data at the present time. The first version is the WIPP Transuranic Waste Baseline Inventory Report (WTWBIR), which is the WTWBID (Rev. 1 of radionuclide data) (WTAC, 1995a). The second version is the Transuranic Waste Baseline Inventory Report (TWBIR), which is in TWBID (Rev. 2 of the radionuclide data) (WTAC, 1995b). PREORG and POSTORG can be used with both sets of data.

The calculations performed with ORIGEN2 estimate the radioactive decay and buildup of various radionuclides in the WIPP waste streams identified in the WTWBID. The resulting radionuclide inventories are needed for performance assessment calculations for the WIPP site.

For the purposes of the WIPP Performance Assessment (PA), PreORG and PostORG are used in conjunction with ORIGEN2 to perform decay and radioactive buildup calculations for inventories of "stored" waste (also termed legacy waste). The data for the decayed stored waste is used directly by the TWBIR team. The TWBIR team supplements this data with information on "future" or projected waste to yield WIPP-Scale data, which is used directly in the WIPP PA calculations as part of the source term parameters (i.e., these data are incorporated into the WIPP PA database).

The work resulted in the development of PreORG, which requires interaction with the user to generate ORIGEN2 input files on a site-by-site basis, and PostORG, which processes ORIGEN2 output into more manageable files. Both programs are written in the FORTRAN 77 computer language. After running PreORG, the user will run ORIGEN2

to generate the desired data; upon completion of ORIGEN2 calculations, the user can run PostORG to process the output to make it more manageable. All the programs run on a 386 PC or higher with a math co-processor or a computer platform running under a VMS operating system.

PostORG and PreORG meet the requirements of the WIPP Quality Assurance Plan with extensive testing to ensure accuracy of the calculations performed by this suite of codes (PreORG, ORIGEN2, and PostORG). Documentation of PreORG and PostORG compliance with the requirements of the WIPP Performance Assessment Quality Assurance Plan is located in the WIPP PA Analysis Report for ORIGEN2 Suite (Liscum-Powell et al., 1996). PreORG and PostORG were tested against analytical and hand calculations to verify the accuracy and validity of results obtained from both of these codes (Liscum-Powell et al., 1996).

To utilize this program, data from the WTWBID must be located in data files structured as depicted in Appendix A. Each site (i.e., waste generating site that sends TRU waste to the WIPP facility) has its own data file for each type of waste: contact-handled transuranic (CH-TRU), remotely handled (RH-TRU), and buried TRU. Using the pre-processor system, the user enters the site of interest as well as the type of waste (CH-TRU, RH-TRU, or BURIED TRU) of interest and a base year; the purpose of this process is to decay all inventory in the associated data files to a specific starting point for further decay calculations. The purpose of the base year is to bring all the data for a given site to the same point in time, as site data files list nuclide inventories year by year (e.g., 1970, 1971, 1973, etc.). Following completion of calculations to decay all data to the designated year, PostORG can be used to generate an additional ORIGEN2 input file to decay the total base year inventory for a site into the future. PostORG allows the user to choose how far into the future he/she would like to decay the base year inventory, ranging from 75 years to 10,000 years. (The user should note that ORIGEN2 input files generated by PreORG and PostORG can be used as a template to assist in developing input files for other ORIGEN2 problems, by making the appropriate modifications to the cards. A card

is a line of data in the ORIGEN2 input file that includes an ORIGEN2 command and the information needed to execute the command.)

This report provides documentation on the programs developed in the course of this work as well as a description of the ORIGEN2 code. When using PreORG and PostORG, this report can be used as a manual to aid in performing the calculations desired. This report describes the three codes: ORIGEN2 (Section 2), PreORG (Section 3), and PostORG (Section 5). Section 4 details how to run ORIGEN2. Section 6 provides step-by-step instructions on using PreORG, ORIGEN2, and PostORG. A summary is presented in Section 7 and references are provided in Section 8.

Intentionally Left Blank

2.0 DESCRIPTION OF ORIGEN2

ORIGEN2 is a point-depletion and radioactive-decay computer code developed at Oak Ridge National Laboratory (ORNL) and distributed by the Laboratory's Radiation Shielding Information Center (RSIC). This code was developed to simulate nuclear fuel cycles and calculate nuclide compositions and characteristics of materials. One of its primary uses is to evaluate the radionuclide composition of nuclear waste material following its radioactive decay. The code itself is written entirely in the FORTRAN computer language and runs in a matter of minutes on a 386 PC (see Appendix B) or higher with a math co-processor (Croff, 1980).

2.1 ORIGEN2 Data Libraries

The ORIGEN2 database consists of three libraries: (1) activation products with 720 nuclides, (2) actinides with 130 nuclides, and (3) fission products with 850 nuclides (Croff, 1980). Of the nuclides in the database, 1300 are unique isotopes; 300 of these nuclides are stable. When running ORIGEN2, it is necessary that all nuclides from a given library are grouped together—starting first with the activation products, followed by the actinides and fission products, respectively (see Appendix C). The decay library contains these three libraries in three separate sections with two cards for each nuclide. The information contained for each nuclide includes the nuclide identifier, radioactive half-life, type of decay, and information on build-up of the nuclide. The nuclide identifier used by ORIGEN2 for a given nuclide is determined as follows (Croff, 1980):

$$\text{NUCLID} = 10000 \cdot Z + 10 \cdot A + \text{IS}$$

where NUCLID is the six-digit nuclide identifier used in the code, Z is the nuclide atomic number, A is the atomic mass of nuclide, and IS indicates the isomeric state of the nuclide (0 = ground state or 1 = excited state). For example, the nuclide identifier for Pu239 (Z = 94, A = 239) would be 942390.

2.2 Characteristics ORIGEN2 Calculates

ORIGEN2 can determine numerous characteristics of the nuclear materials following radioactive buildup and decay of a material and can simulate the reactor irradiation of cladding and structural material and burnup of the fuel following simulation of reactor operation. These characteristics include the following: mass, fractional isotopic composition, radioactivity, thermal power, toxicity following ingestion or inhalation, neutron absorption and emission rates, and photon emission rates (Croff, 1980). Internally, ORIGEN2 stores these data or characteristics in tables or vectors indexed by user-assigned numbers. By including certain commands in the input file, the user controls what type of data ORIGEN2 will generate. Appendix C contains a sample ORIGEN2 input file generated by PreORG; Appendix C also provides tables indicating the type of data the code will generate and information regarding how to instruct ORIGEN2 to include particular information in its output files.

2.3 ORIGEN2 Input Structure

The ORIGEN2 input structure is composed of a series of command cards with each line of input representing a single operation and the details of its execution. The commands are executed sequentially by ORIGEN2 as they are read from the input file. Every ORIGEN2 input file must have the extension .U5 to indicate to the code that the input file is the source of ORIGEN2's instructions for a given run. To perform a simple radioactive decay calculation as is done in this work, the following instructions must be given to ORIGEN2 in the input file:

1. Indicate the data libraries to be used for the calculations.
2. Instruct ORIGEN2 to read initial radionuclide concentrations in grams or grams-atom. (*Note: ORIGEN2 does not accept input in Curies although it will output values in Curies. When converting curies into grams for ORIGEN2 input, the user should use the half lives from the ORIGEN2 decay library. The ORIGEN2 decay library is a part of the ORIGEN2 code package and can be found in the directory where the data libraries are located [e.g., /ORIGEN2/LIBS]).*
3. Instruct ORIGEN2 to decay out initial data for a given length of time.
4. Tell ORIGEN2 what output tables you desire.

5. Instruct ORIGEN2 to write desired results to the output file. (*Note: The primary ORIGEN2 output files have the extension .U6*).
6. Terminate execution of program.

The steps above are accomplished using a series of commands as those shown in the sample input file in Appendix C. It should be noted that there are several limitations in ORIGEN2 as far as the number of command cards of a certain type that can be used; the user is directed to the ORIGEN2 manual for further elaboration (Croff, 1980). For instance, the total number of commands that can be used in a given input file is 300.

2.3.1 Entering Initial Nuclide Concentrations

The initial nuclide data is either entered in the ORIGEN2 input file after the STP or END card, or it is placed in an external file with the proper instruction. Each line of initial data cannot exceed 80 bytes (equal to 3 nuclide-identifier and initial-concentration pairs) and must end with two zeros (0 0) to indicate that the end of that card has been reached. The first entry on a card must indicate which of the three libraries the nuclide-decay and accumulation data is contained in (1, 2, or 3); all nuclides from a given library must be grouped together. Each input card reads data until it encounters a line with one zero (0) indicating that the following data is read by a new input card. A particular isotope can be read in a given input string more than once and ORIGEN2 will calculate internally the total initial concentration of the repeated nuclide.

2.4 Calculations Performed by ORIGEN2

ORIGEN2 performs a number of trivial calculations such as simple summations and multiplications, but it is also characterized by two unique features (Croff, 1983): (1) the method it employs to store the equations describing radionuclide decay and accumulation and (2) the method employed to solve these equations for nuclide concentrations. The general form of the equation ORIGEN2 solves is as follows (Croff, 1983):

$$\frac{dN_i}{dt} = \sum_{j=1}^{N_{tot}} \lambda_{ij} \lambda_j N_j + \phi \sum_{k=1}^{N_{tot}} f_{ik} \sigma_k N_k - (\lambda_i + \phi \sigma_i + r_i) N_i + F_i, \quad i = 1, \dots, N_{tot} \quad (1)$$

where

N_i, N_k = Atom density of nuclide i, k

N_{tot} = Number of nuclides

λ_{ij} = Fraction of radioactive disintegration by other nuclides that leads to formation of species i

λ_i = Radioactive decay constant of species i

ϕ = Position and energy averaged neutron flux

σ_k = Spectrum averaged neutron absorption cross-section of nuclide k

f_{ik} = Fraction of neutron absorption by other nuclides that leads to formation of species i

r_i = Continuous removal rate of nuclide i from the system

F_i = Continuous feed rate of nuclide i

For each nuclide of interest, there is one equation of this general form. For decay calculations independent of constant power or flux irradiation, these equations reduce to the Bateman equations (Kaplan, 1964):

$$\frac{dN_i}{dt} = \sum_{j=1}^{N_{tot}} \lambda_{ij} \lambda_j N_j - \lambda_i N_i, \quad i = 1, \dots, N_{tot} \quad (2)$$

These equations make up a system of ordinary differential equations that translate into a sparse matrix. ORIGEN2 stores only the nonzero elements of this matrix by utilizing indexing techniques in what is referred to as the transition matrix (Croff, 1983). Thus, the transition matrix contains all the transformation rates of each parent nuclide through decay and other processes such as neutron absorption.

2.5 Method of Solving Differential Equations in ORIGEN2

The coupled, ordinary differential equations discussed above that represent the accumulation and decay of each nuclide of interest in an ORIGEN2 calculation are solved internally using a compilation of three solution techniques (Croff, 1983). The primary or cornerstone solution technique is the exponential matrix technique; however, problems are encountered with this method when trying to solve a matrix containing widely separated eigenvalues, which are often encountered with the type of problems ORIGEN2 solves (Croff, 1983). To avoid these calculation problems, asymptotic versions of analytical solutions to the nuclide accumulation and decay equations are employed at different points during the solution process (Croff, 1983).

Essentially, ORIGEN2 solves the system of ordinary differential equations by following three steps for each time step or decay interval. These solution steps include the following (Croff, 1983):

1. A set of asymptotic solutions are utilized that are suitable for handling short-lived nuclides whose parent nuclides are not long-lived. ORIGEN2 defines a short-lived nuclide as one whose half-life is less than 14.4% that of the decay interval (Croff, 1983). The concentrations of these nuclides reach an equilibrium or constant value within the time step, and thus, the asymptotic solution is an adequate approximation. These solutions are used to quantify nuclides that reach transient equilibrium during a given decay interval. Transient equilibrium occurs when the parent nuclide is characterized by a half-life that is greater than its daughter's half-life but not by an excessive amount ($T_{1/2}^P \approx 3$ to 10 times $T_{1/2}^D$).
2. The transition matrix is further reduced to a matrix containing only long-lived nuclides and this reduced matrix is solved for the concentrations of these nuclides at the end of the time step using the exponential method. The matrix exponential method utilizes the series representation of the exponential function by translating the summations found in Equations (1) and (2) into exponential functions. The exponential method is described by the equations below (Croff, 1983):

$$N_{dt} = AN \quad (3)$$

where $\mathbf{N}_{dt}$ is a column matrix with the time derivative of the nuclide concentrations, $\mathbf{A}$ is the reduced transition matrix with the long-lived nuclide transformation rates, and $\mathbf{N}$ is a column matrix containing the nuclide concentrations at the end of the time step. Using the exponential method solution technique, this equation reduces to the following relationship (Croff, 1983):

$$\mathbf{N}(t) = e^{\mathbf{A}t} \mathbf{N}(t=0) \quad (4)$$

where $\mathbf{N}(t)$ is the nuclide concentration at time t , $\mathbf{N}(t=0)$ is the initial nuclide concentration at the beginning of the decay interval, and t is the time at the end of a given decay interval.

3. During the third and final calculation step, ORIGEN2 uses another set of asymptotic solutions to analytical equations to calculate concentrations of short-lived nuclides with long-lived parents ($T_{1/2}^P > 10 \text{ times } T_{1/2}^D$). These nuclides reach secular equilibrium in which the daughter and parent activity reach an equilibrium state after approximately seven daughter half-lives have passed. The Gauss-Seidel successive substitution method is employed to solve these equations (Croff, 1983).

3.0 DESCRIPTION OF PREORG

The program PreORG was developed to serve as a pre-processor to ORIGEN2 calculations by automating WIPP TWBID data files into ORIGEN2 input files to perform nuclide decay and accumulation calculations. A listing of PreORG, written entirely in the FORTRAN computer language, is located in Appendix D.

The pre-processor requires interaction with the user and generates ORIGEN2 input files through the following steps:

- STEP 1 Prompts user for input:
- Base year to which all data are initially decayed
 - Filename to write a list of input files generated
 - Site name
 - Enriched Uranium isotopic composition
 - Type of waste
- STEP 2 Reads all data from input file into arrays containing nuclide name (e.g., PU239) and the initial activity in Curies. Counts the number of years in the data file and the number of isotopes in each year.
- STEP 3 Writes basic ORIGEN2 commands to read input for each year.
- STEP 4 Writes ORIGEN2 commands to decay all data to the common base year given by user.
- STEP 5 Converts all nuclide activities to grams using specific activities, assigns appropriate nuclide identifiers (e.g., PU239 $\Rightarrow$ 942390), and determines ORIGEN2 data library where each nuclide decay data is located. (*Note: Half-lives used to determine a nuclide activity are identical to those found in the ORIGEN2 decay libraries.*)
- STEP 6 Writes out initial activities year-by-year, one library at a time (Appendix B).
- STEP 7 Ensures all data in the PreORG input file has been incorporated in ORIGEN2 input file (PreORG output file). Because each ORIGEN2 input file can use a maximum of 15 input cards, if there are more than 15 years of data, return to STEP 3 using STP card and repeat STEPS 3 to 6 until all data are in input file.
- STEP 8 Prompts user to indicate whether or not he/she would like to generate more input files (for a given run, PreORG assumes same base year for all sites):
- If yes, repeat STEPS 1(c) through 1(e) and STEPS 2 through 7.

- If no, terminate execution.

Appendix A contains a list of input files that PreORG will accept from the WIPP WTWBID, plus a listing of what is written to the screen when running PreORG with examples of user input. The user can choose from sites and waste streams that are specifically associated with the WIPP WTWBID or a designated user site as long as the input file structure and name are in the correct format (Appendix A and Table 1). To choose a site or input file that is not associated with the WIPP WTWBID, choose the 'OTHER SITE' category when prompted for a site number and PreORG will prompt you for needed information.

PreORG generates one ORIGEN2 input file for a given year (e.g., 1993) for each site and each waste type and names the input file as indicated in the following table (*Note: The user can specify a base year as needed for the calculation.*). The first four letters of the ORIGEN2 input file indicate the site name, the next two letters describe the type of waste, and the numbers indicate the base year to which all initial inventories are decayed out prior to future decay calculations (Table 1).

Table 1. ORIGEN2 Input Files Created by PreORG

TYPE OF WASTE	PREORG OUTPUT FILENAME	EXAMPLE FOR ORNL SITE
CH TRU	xxxxCH93.U5	ORNLC93.U5
RH TRU	xxxxRH93.U5	ORNLR93.U5
BURIED TRU	xxxxBU93.U5	ORNLBU93.U5

In the WIPP WTWBID data files, activities may be listed that are not broken down into specific isotopes but are grouped together and given the following names: UENR, UDPL, UNAT, UNKN, TBGMAP, and TBGMFP. When PreORG encounters these names, it automatically breaks the activities into the following isotopic breakdowns by multiplying the supplied activity by the activity factor as follows:

UENR - Enriched Uranium

The user enters the atom fractions for U234, U235, and U238 at the beginning of the PreORG run for a given site. The enriched uranium atom fractions can be modified for each site if desired.

UDPL - Depleted Uranium

ISOTOPE	ACTIVITY FACTOR
U238	1.00

UNAT - Natural Uranium (General Electric Nuclear Energy, 1989)

ISOTOPE	ACTIVITY FACTOR
U234	0.000055
U235	0.00720
U238	0.992745

UNKN - Unknown

Ignores these activities for lack of a better breakdown

TBGMAP - Mixed Activation Products (see Appendix E)

ISOTOPE	ACTIVITY FACTOR
Ba137m	9.7000E-03
C14	1.0000E-02
Co58	1.0000E-02
Co60	6.7000E-01
Cs137	1.0300E-02
Eu152	1.0000E-02
Eu154	8.5000E-02
Eu155	8.5000E-02
Fe59	4.0000E-02
Mn54	7.0000E-02

TBGMFP - Mixed Fission Products (see Appendix E)

ISOTOPE	ACTIVITY FACTOR
Ag110	5.0200E-06
Ag110m	2.5070E-06
Ba137m	2.0090E-02
Cd113m	5.0000E-10
Ce144	2.7840E-01
Cs136	5.5570E-04
Cs135	5.7500E-06
Cs137	2.0090E-02
Eu152	3.0000E-07
Eu154	3.0380E-05
Eu155	7.3530E-04
Gd162	9.8620E-06
Nb95	1.0530E-01
Nb95m	1.0590E-03
Pd107	8.5000E-09
Pm147	7.0220E-02
Pr144	2.7840E-01
Rh106	4.0510E-02
Ru106	4.0510E-02
Sb125	7.9850E-04
Sb126	1.0940E-07
Sb126m	1.1050E-07
Se79	4.9900E-08
Sm151	1.9380E-04
Sn119m	4.3720E-05
Sn121m	3.9430E-06
Sn123m	1.8470E-05
Sn126	1.1050E-07
Sr90	1.8810E-02
Tb162	9.8620E-06
Tc99	2.8670E-06
Te125m	2.1760E-04
Te127	1.3540E-03
Te127m	1.3690E-03
Y90	1.8810E-02
Zr93	6.4550E-07
Zr95	4.9880E-02

3.1 Running PreORG

To run the ORIGEN2 pre-processor on a personal computer (PC), the source code must be obtained from either Jennifer Liscum-Powell, Nuclear Facilities Safety and Risk Analysis, Org. 9364, or Lawrence C. Sanchez, WIPP Performance Assessment Computational Support, Org. 6848. If neither of the two personnel listed is available, the user can contact the WIPP Performance Assessment Computational Support Department, Org. 6848. The user installs PREORG.FOR into a specified directory and compiles PREORG.FOR with a FORTRAN77 compiler. Once an executable file has been created, the user types PREORG at the prompt and presses the carriage return. The user then follows the instructions as they appear on the screen. *(Note: All WIPP WTWBID input files must be in the same directory, otherwise PreORG will fail to execute.)*

A user directly related to the WIPP project should obtain an OpenVMS account on the WIPP PA Cluster and run the PreORG from the Code Management System (CMS). PreORG resides in the OR library of CMS. The user can locate the executable file—PREORG_PA96.EXE—in the WP\$PRODROOT:[OR.EXE] library. PreORG should be run using command files; examples of these command files are located in the WP\$TESTROOT:[OR2.TEST] directory.

Intentionally Left Blank

4.0 RUNNING ORIGIN2

To run ORIGIN2 on a PC, the user must follow a few instructions. First, a user ensures that a licensed copy of ORIGIN2 from the Radiation Shielding Information Center (RSIC) of Oak Ridge National Laboratory is installed on the computer. Once ORIGIN2 is installed on a PC, the following directories should be located on the hard drive: C:\ORIGIN2, C:\ORIGIN2\CODE, C:\ORIGIN2\LIBS, C:\ORIGIN2\SAMPLES, and C:\ORIGIN2\INPUTS. To run the code, the file ORIGIN2.BAT must be present in the C:\ORIGIN2 directory and F77I3.EER must be located in the C:\ORIGIN2\CODE directory along with the executable ORIGIN2.EXE. A listing of the ORIGIN2.BAT file for the PC is given in Appendix B. This file is not issued by RSIC with the ORIGIN2 package. To obtain a copy of the source code, contact either Jennifer Liscum-Powell, Org. 9364, or Lawrence C. Sanchez, Org. 6848, or contact the WIPP PA Computational Support Department, Org. 6848. To run ORIGIN2 without this .BAT file, refer to the ORIGIN2 manual (Croff, 1980).

To run ORIGIN2, simply type the following line at the prompt in the C:\ORIGIN2 directory:

C:\ORIGIN2 ORIGIN2 filename [return]

For example, if the ORIGIN2 input file is ORNLCH93.U5, the user would type ORIGIN2 ORNLCH93 to run ORIGIN2. (*Note: Do not include the .U5 extension when ORIGIN2 runs, as it will not execute and will write over the input file!*) Running ORIGIN2 generates a number of output files; however, the only significant file is the one with the .U6 extension. This file contains the existing desired output of ORIGIN2. It will be a sizable file because ORIGIN2 writes out a vast amount of data to its output files, a large portion of which is not needed by the user.

A user directly related to the WIPP project should obtain an OpenVMS account on the WIPP PA Cluster and run ORIGIN2 from the Code Management System (CMS). The user can locate the executable file—ORIGIN2_PA96.EXE—in the

WP\$PRODROOT:[OR.EXE] library. ORIGEN2 should be run using command files; examples of these command files are located in WP\$TESTROOT:[or2.test] directory.

PostORG, described in more detail in the next section, was created to process the output files and remove all excessive and irrelevant data for the purposes of the WIPP WTWBID nuclide decay and accumulation calculations. PostORG also creates a master list of nuclides, sums the radionuclide inventories at the base year and creates an ORIGEN2 input file to decay the accumulated base year radionuclide inventories into the future.

5.0 DESCRIPTION OF POSTORG

PostORG was developed to process ORIGEN2 output files with the extension .U6 to remove excessive data and extract the nuclide decay and accumulation data that are of interest for the scope of these calculations. This post-processor uses the following steps:

- STEP 1 Prompts user for name of ORIGEN2 output file to be processed (e.g., ORNLCH93.U6 or ORNLC93F.U6).
- STEP 2 Prompts user if he/she wishes to generate a ORIGEN2 input file to decay base year (e.g., 1993) data (located in ORNLCH93.SUM file) to a specified time in the future.
- STEP 2a If answer for step 2 is yes, ORIGEN2 input file is created with a name like ORNLC93F.U5 (Must run ORIGEN2 with this input file to generate data and then use PostORG to reduce data to ORNLC93F.SUM file).
- STEP 2b If answer to step 2 is no, no new ORIGEN2 input file is created.
- STEP 3 Reads through output file and separates data by year into individual files (e.g., ORNLCH70.DAT, ORNLCH71.DAT, etc.) and creates a list of these files in an .IND file (e.g., ORNLCH93.IND).
- STEP 4 Calls subroutine to add all activities at same years together. (*Note: The process only considers isotopes of interest to the WIPP that are identical to those listed in Appendix E of the WIPP Baseline Inventory Report, Rev. 1 [WTAC, 1995a].*)
- STEP 5 Writes sums to file .SUM (e.g., ORNLCH93.SUM).
- STEP 6 Prompts user to indicate whether or not he/she wishes to process another output file
- If yes, return to STEP 1.
 - If no, terminate execution.

The resulting output files can be easily imported into a spreadsheet and manipulated as desired by the user. A program listing is provided in Appendix F and a sample output file is listed in Appendix G along with a listing of the output written to the server where POSTORG is executed.

5.1 Running PostORG

To run the ORIGIN2 post-processor on a PC, the user copies PostORG.for into a specified directory and compiles data with a FORTRAN77 compiler. The source code, POSTORG.FOR, may be obtained by contacting Jennifer Liscum-Powell, Org. 9364, or Lawrence C. Sanchez, Org. 6848, or the WIPP PA Computational Support Department, Org. 6848. Once an executable file has been created, the user types PostORG at the prompt and presses the carriage return. The user then follows the instructions as they appear on the screen. (Note: All ORIGIN2 output files that are to be processed must be in the same directory, otherwise PostORG will fail to execute.)

A user directly related to the WIPP project should obtain an OpenVMS account on the WIPP PA Cluster and run PostORG from the Code Management System (CMS). PostORG resides in the OR library of CMS. The user can locate the executable file — POSTORG_PA86.EXE—in the WP\$PRODROOT:[OR.EXE] library. PostORG should be run using command files; examples of these command files are located in the WP\$TESTROOT:[OR2.test] directory.

6.0 USING PREORG, ORIGIN2, AND POSTORG

This section offers step-by-step instructions for running the three codes discussed in this report. By following these instructions, the user should be able to successfully obtain the desired information. There is also a list of warnings for the user to note, particularly when running ORIGIN2 and analyzing the data.

6.1 Running PreORG, ORIGIN2, and PostORG on a PC

To successfully run the three codes discussed in this report, a user should follow the instructions listed below:

PreORG

STEP 1 Change directory to the one containing PREORG.EXE.

STEP 2 At prompt, type PREORG and hit return.

STEP 3 Answer questions as they appear on screen.

ORIGIN2

STEP 1 Copy all input files with .U5 extension into ORIGIN2 directory.

STEP 2 For each ORIGIN2 input file specified to run, type ORIGIN2 filename without extension at the prompt and hit return.

STEP 3 Copy all ORIGIN2 output files with .U6 extension to directory containing POSTORG.EXE.

STEP 4 Delete all unnecessary output files with extensions other than .u6.

PostORG

STEP 1 Change directory to one containing POSTORG.EXE.

STEP 2 At prompt, type POSTORG and hit return.

STEP 3 Answer questions as they appear on screen.

(Note: POSTORG output files may be imported into EXCEL spreadsheet to view and print [optional]).

6.2 Running PREORG, ORIGEN2, and POSTORG on a VMC Computer Platform

To successfully run the three codes discussed in this report on VMS, a user should develop a command file to run the codes sequentially: (1) PREORG, to generate ORIGEN2 input files, (2) ORIGEN2, to generate radionuclide decay and accumulation data, and (3) POSTORG, to extract useful information from ORIGEN2 output files.

A sample command file is shown in Figure 6.2.1 that runs ORIGEN2 for CH-TRU waste from the Hanford and Oak Ridge waste generation sites.

6.3 Cautions to Observe when Running PreORG, ORIGEN2, and PostORG

To successfully run the three codes discussed in this report, please observe the following cautions listed below:

PreORG Cautions

- All input files for a given site and given waste type must have the correct name (see Appendix A).
- All input files must be in the correct format (see Appendix A).

ORIGEN2 Cautions

- Input file must be in correct directory.
- When running the code, do not include .U5 extension on input or ORIGEN2 will fail to execute and write over input file.
- Disable EMM386.EXE memory extender.
- ORIGEN2 does not warn you if errors are present in input. The user is responsible for checking output to make sure data makes sense.
- ORIGEN2 has problems (loses accuracy and yields isotope concentrations that are not feasible) when instructed to decay each initial year of inventory out and then print the years all together in one output table. Thus, PreORG writes commands to decay out each year to the given base year (e.g., 1993) individually and outputs each year's in its own output table (see Appendix C).

```

$!=====
$!
$!  ORIGEN2.com is a sample command file that maybe used to run PreORG,
$!  ORIGEN2, and PostORG.
$!
$!=====
$!
$!  DEFINE LOGICALS and SYMBOLS
$!
$!  Define the PREORG run symbol
$  PREORG ::= "$ wp$prodroot:[OR.EXE]PREORG_PA96"
$  show symbol PREORG
$!
$!  Define the ORIGEN2 run symbol
$  ORIGEN2 ::= "$ wp$prodroot:[OR.EXE]ORIGEN2_PA96"
$  show symbol ORIGEN2
$!
$!  Define the POSTORG run symbol
$  POSTORG ::= "$ wp$prodroot:[OR.EXE]POSTORG_PA96"
$  show symbol POSTORG
$!
$!=====
$!
$!  Get the needed nuclear data library files in appropriate format for
$!  ORIGEN2.  FILES RESIDE IN ORIGEN2 CMS LIBRARY
$!
$libor
$!
$!  Fetch the library files.
$!
$  cms fetch/nohistory or2_decay.lib
$  cms fetch/nohistory or2_pwru.lib
$  cms fetch/nohistory or2_gxuo2brm.lib
$!
$!  Build input files from fetched libraries
$!
$  copy or2_decay.lib + or2_pwru.lib tape9.inp
$  copy or2_gxuo2brm.lib tape10.inp
$!
$!  Delete library files to save space
$!
$  delete or2_decay.lib;*, or2_pwru.lib;*, or2_gxuo2brm.lib;*
$!
$!  Set to ORIGEN2 CMS library
$!
$  libor
$!
$!  FETCH DATA FILES FOR SITE (OR SITES) OF INTEREST
$!
$!  e.g. for Hanford and Oak Ricge CH-TRU:
$!
$  cms fetch/nohistory or2_hanford2.dat
$  rename or2_hanford2.dat hanford2.dat
$!
$  cms fetch/nohistory or2_ornl__2.dat
$  rename or2_ornl__2.dat ornl__2.dat
$!
$!  RUN PREORG.  ANSWER USER INTERACTIVE QUESTIONS IN THIS FILE
$!  Answer 1:  year to decay date out to
$!  Answer 2:  index filename (keeps track of sites)
$!  Answer 3:  waste generation site number
$!  Answer 4:  waste category: CH-TRU(1), RH-TRU(2), or BU-TRU(3)
$!  Answer 5:  indicates if user wishes to change default Uranium enrichment
$!  Answer 6:  enter 1 to run PreORG again or 2 to terminate execution, if enter 1
$!              then answer questions 3 through 6 again, etc.
$!
$  ASSIGN/USER SYSS$INPUT SYSS$COMMAND
$  PREORG
1995
samp01
4
1

```

Figure 6.2.1 Sample command file used to run PreORG, ORIGEN2, and PostORG.

```
n
2
$ PREORG
1995
samp02
13
1
n
2
$!
$! RUN ORIGEN2 for each site
$!
$! HANFORD CH-TRU:
$ rename hanfch95.u5 tape5.inp
$ ORIGEN2
$ rename tape6.out hanfch95.u6
$!
$!
$ rename ornlch95.u5 tape5.inp
$ ORIGEN2
$ rename tape6.out ornlch95.u6
$!
$! RUN POSTORG. ANSWER USER INTERACTIVE QUESTIONS IN THIS FILE
$! Answer 1: ORIGEN2 output filename to process (.u6 extension)
$! Answer 2: base year decayed original data to
$! Answer 3: indicates if user wishes to create ORIGEN2 input file to
$!             decay further into future. If yes, user enters letter to
$!             indicate time in the future to decay to (see manual)
$! Answer 4: enter 1 to run PreORG again or 2 to terminate execution, if enter 1
$!             then answer questions 1 through 4 again, etc.
$ ASSIGN/USER SYS$INPUT SYS$COMMAND
$ POSTORG
hanfch95.u6
1995
n
2
$ POSTORG
ornlch95.u6
1995
n
2
$!
$! Delete extra files.
$!
$ delete *.inp;*
$ delete *.ind;*
$ delete *.u6;*
$ delete *.out;*
$ delete *.dat;*
$!
```

Figure 6.2.1 Sample command file used to run PreORG, ORIGEN2, and PostORG (continued).

- ORIGEN2 includes an "ADD" command that will supposedly add nuclide concentrations stored in one output vector to another one nuclide at a time. However, when this command is used to add final nuclide data together after all initial inventories were decayed out to the base year, inaccuracies are observed along with activities for isotopes that are physically impossible. This problem with ORIGEN2 is avoided by first decaying all initial inventories to a common year by running ORIGEN2 with the input files created by PreORG and then running PostORG to process the output and generate an input file for ORIGEN2 to decay isotopes into the future (Appendix G).
- ORIGEN2 is limited in how many output vectors it will print in its output file. For a given initial nuclide concentrations, ORIGEN2 can output 11 vectors without error. Thus, the initial values can be included in the output along with 10 vectors containing data at 10 different times.
- When using the decay command several times to look at nuclide concentrations at various time intervals in the future, please note that the time interval or step of the decay is determined by subtracting the time given in the previous decay command from the time given in the present one. Please make sure the decay time is greater in the latter command or errors will result (Appendix C).

PostORG Cautions

- Input file names must be 11 characters long and include the extension .u6.
- Input files must be in the same directory as POSTORG.EXE.
- Must enter the same base year as used in PREORG (e.g., 1993).

Intentionally Left Blank

7.0 SUMMARY

PreORG and PostORG were created to automate decay calculations using ORIGEN2 for the WIPP WTWBID. The pre-processor, PreORG, translates WTWBID data files into ORIGEN2 input files by requiring interaction with the user. The user indicates which waste-generation site and waste type is of interest and the base year to which he/she wishes to decay all years of inventory. A user can choose to use a site other than the 20 currently contributing to the WIPP inventory on a site-level basis; however, the user should be cautioned that the naming convention is the same and PreORG will fail to execute if the data file does not exist in the correct format. PostORG processes the ORIGEN2 output files, eliminating excess data and thus creating a more manageable output. PostORG can also generate an ORIGEN2 input file to decay the base year data for a given waste generation site up to 10,000 years into the future. These two codes enable the automation of ORIGEN2 decay and accumulation calculations for the WIPP WTWBID.

Intentionally Left Blank

8.0 FURTHER READING

Further generalized discussion on ORIGEN2 can be found in Croff, 1983. Detailed information on the validation (as applied to analysis of PWR spent fuel) and functional requirements (related to long-term storage of LWR spent fuel) of ORIGEN2 can be found in Hermann et al., 1995, and Broadhead et al., 1995.

Intentionally Left Blank

9.0 REFERENCES

- Broadhead, B.L., M.D. DeHart, J.C. Ryman, J.S. Tang, and C.V. Parks. 1995. *Investigation of Nuclide Importance to Functional Requirements Related to Transport and Long-Term Storage of LWR Spent Fuel*. ORNL/TM-12742. Oak Ridge, TN: Oak Ridge National Laboratory.
- Croff, A.G. 1980. *A User's Manual for ORIGEN2 Computer Code*. ORNL/TM-7175. Oak Ridge, TN: Oak Ridge National Laboratory.
- Croff, A.G. 1983. "ORIGEN2: A Versatile Computer Code for Calculating the Nuclide Compositions and Characteristics of Nuclear Materials," *Nuclear Technology*. Vol. 62, no. 3, 335-352.
- Hermann, O.W., S.M. Bowman, C.V. Parks, and M.C. Brady. 1995. *Validation of the Scale System for PWR Spent Fuel Isotopic Composition Analyses*. ORNL/TM-12667. Oak Ridge, TN: Oak Ridge National Laboratory.
- Kaplan, I. 1964. *Nuclear Physics*. 2nd ed. Reading, MA: Addison-Wesley Publishing Company. pp. 229, 242-243.
- Liscum-Powell, J., R. Carter, and H. Trellue. 1996. "WIPP PA Analysis Report for ORIGEN2 Suite," WBS# 1.1.6.1, SWCF-A:1.4.01.6:SFT:VH511196:QA:ORIGEN2: Analysis Package for Initial Radionuclide Inventory, April 22, 1996. (Copy on file in the Sandia WIPP Central Files, Sandia National Laboratories, Albuquerque, NM as WPO 36974.)
- Tuli, J.K. 1985. *Nuclear Wallet Cards*. Upton, NY: National Nuclear Data Center, Brookhaven National Laboratory. (Copy on file in the Sandia WIPP Central Files, Sandia National Laboratories, Albuquerque, NM as WPO 41010.)
- Turner, J.E. 1986. *Atoms, Radiation, and Radiation Protection*. New York, NY: Pergamon Press. pp. 39, 64.

Walker, F.W., J.R. Parrington, and F. Feiner. 1989. *Nuclides and Isotopes: Chart of the Nuclides*. 14th ed. San Jose, CA: General Electric Company.

WTAC (WIPP Technical Assistance Contractor). 1995a. *Waste Isolation Pilot Plant Transuranic Waste Baseline Inventory Report*. CAO-94-1005, Revision 1. [Carlsbad, NM]: WIPP Technical Assistance Contractor for U.S. Department of Energy. Vols. 1-3.

WTAC (WIPP Technical Assistance Contractor). 1995b. *Transuranic Waste Baseline Inventory Report (Revision 2)*. DOE/CAO-95-1121. Carlsbad, NM: Carlsbad Area Office Technical Assistance Contractor.

APPENDIX A: SAMPLE PREORG.FOR DATA INPUT FILE

Intentionally Left Blank

Appendix A: Sample PREORG.FOR Data Input File

On the following pages is a sample input file for "PREORG.FOR" containing data from the Waste Isolation Pilot Plant (WIPP) Baseline Inventory Database (BID) for the Oak Ridge National Laboratory (ORNL) site for contact-handled transuranic (CH TRU) waste from 1970 to 1977. (Note: This is a portion of the data file for ORNL CH TRU waste).

All input files used with the pre-processor must follow the format presented in this appendix. All input files must also have the proper name structure as indicated in Table A-1 below. The input filename must be eight characters long, excluding the extension—.DAT—that is required. The eighth position of the input file name must indicate the type of waste contained in the file: a 2 for CH TRU waste, a 3 for remotely handled (RH) TRU waste, or a 7 for Buried TRU waste.

Table A-1. PreORG.for Input Filename Structure

TYPE OF WASTE	PREORG.FOR INPUT FILENAME	EXAMPLE FOR ORNL SITE
CH TRU	xxxxxxx2.DAT	ORNL__2.DAT
RH TRU	xxxxxxx3.DAT	ORNL__3.DAT
BURIED TRU	xxxxxxx7.DAT	ORNL__7.DAT

To run the ORIGEN2 pre-processor to generate an ORIGEN2 input file, all input files must be identical in format to the file presented in this appendix. For the purposes of this report, the first two lines are added to indicate the columns in which data are read. (Note: In order for PREORG.FOR to properly read an input file, all data must be located in the designated column because it uses character strings to read and manipulate the data.) If the year is not given in columns 1 to 4, PREORG.FOR will not work. Likewise, it is necessary that the isotope names start in column 3 and that the initial activities are listed in Curies in columns 41 through 50. The end of the data for each year is indicated by the presence of 'ZZZZZ' in columns 3 through 7. The data shown in columns 11 to 16 are not necessary for the preprocessor to run properly, but if

the data are deleted, they must be replaced with spaces to maintain the column integrity in order for the initial activities to be read in correctly.

Following the sample file, Table A-2 lists specific input files for the WIPP Transuranic Waste BID (TWBID).

Appendix A: Sample PREORG.FOR Data Input File

	1	2	3	4	5	5
	1234567890123456789012345678901234567890123456789012345					
1	ORNL STORED	CONTACT	HANDLED	TRU	WASTE	
2	1970	Year of Nuclide Inventories				ORIGEN2 TITLE
3	PU239	20	31		7.0000E+00	
4	AM241	20	36		2.8400E-04	
5	U233	20	23		2.8400E-04	
6	AM243	20	37		1.0100E-04	
7	CF252	20	44		1.0000E-06	
8	CM244	20	45		4.4300E+00	
9	U235	20	25		7.5000E-05	
10	UNKN	0	0		3.6000E+01	
11	ZZZZZ	99	0			
12	1971	Nuclide Name				
13	PU238	20	30		4.2600E+03	
14	PU239	20	31		3.2700E+01	
15	PU240	20	32		1.7600E+00	
16	PU241	20	33		4.0800E+02	
17	AM241	20	36		8.8000E+02	
18	U233	20	23		1.5200E-01	
19	CO60	10	07		1.0000E-06	
20	AM243	20	37		2.4800E+00	
21	BK249	20	46		3.6200E+00	
22	CF252	20	44		1.1400E+00	
23	CM242	20	38		3.3200E+01	
24	CM244	20	45		4.9300E+02	
25	NP237	20	28		3.2300E+00	
26	U235	20	25		4.1000E-04	
27	UNKN	0	0		3.6200E+00	
28	ZZZZZ	99	0			
29	1972	Initial Nuclide Activity in Curies				
30	PU238	20	30		7.8000E+02	
31	PU239	20	31		4.9900E+01	
32	PU240	20	32		4.6900E-01	
33	PU241	20	33		3.0000E+03	
34	AM241	20	36		1.6200E+02	
35	AM243	20	37		3.7000E+00	
36	BK249	20	46		2.0100E-03	
37	CF252	20	44		5.5300E-01	
38	CM244	20	45		4.2000E+02	
39	NP237	20	28		1.0900E-01	
40	TH232	20	18		3.3400E-02	
41	U235	20	25		1.6600E-04	
42	U238	20	27		2.2000E-05	
43	UNKN	0	0		1.0800E+02	
44	ZZZZZ	99	0			
45	1973	Indicates end of inventory for current year has been reached.				
46	PU238	20	30		5.8400E+01	
47	PU239	20	31		3.6600E+00	
48	PU240	20	32		3.5900E-01	
49	PU241	20	33		8.4000E+01	
50	AM241	20	36		1.3900E+01	
51	U233	20	23		2.1900E+00	
52	AM243	20	37		2.2000E+00	
53	BK249	20	46		1.8200E+00	
54	CF252	20	44		5.3500E+01	
55	CM244	20	45		5.5100E+02	
56	NP237	20	28		6.1800E-03	
57	U235	20	25		2.3100E-04	
58	ZZZZZ	99	0			
59	1974					

Appendix A: Sample PREORG.FOR Data Input File

60	PU238	20	30	9.5500E+00
61	AM241	20	36	4.5500E+00
62	U233	20	23	9.7700E+00
63	AM243	20	37	9.2500E-02
64	BK249	20	46	4.5300E+02
65	CF252	20	44	1.8000E-01
66	CM244	20	45	5.0700E+02
67	NP237	20	28	2.7500E-03
68	PM147	10	22	1.8600E+00
69	TH232	20	18	1.0600E-04
70	UNKN	0	0	5.0000E-01
71	ZZZZZ	99	0	
72	1975			
73	PU238	20	30	1.0000E-04
74	PU239	20	31	3.7100E+00
75	AM241	20	36	3.2400E+00
76	U233	20	23	1.9500E+01
77	BK249	20	46	1.0200E-04
78	CF252	20	44	1.7200E-03
79	CM244	20	45	6.2800E-02
80	EU152	10	27	1.0000E-06
81	NP237	20	28	9.2100E-02
82	TH232	20	18	2.1800E-04
83	UNKN	0	0	1.0000E-03
84	ZZZZZ	99	0	
85	1976			
86	PU238	20	30	1.0000E-04
87	PU239	20	31	5.1100E+00
88	AM241	20	36	2.0500E-03
89	U233	20	23	1.8600E+01
90	AM243	20	37	2.0000E-04
91	BK249	20	46	9.0600E-02
92	CF252	20	44	1.0000E-01
93	CM244	20	45	4.1900E-02
94	NP237	20	28	1.2400E-01
95	TH232	20	18	1.0900E-04
96	U235	20	25	1.1400E-04
97	U238	20	27	4.6700E-03
98	ZZZZZ	99	0	
99	1977			
100	PU238	20	30	4.7300E+00
101	PU239	20	31	1.1100E+02
102	PU241	20	33	3.9300E-03
103	PU242	20	34	5.2400E+00
104	AM241	20	36	7.6200E+00
105	CS137	10	18	1.0000E-03
106	AM243	20	37	3.0000E-05
107	CF252	20	44	1.0000E-01
108	CM242	20	38	1.0000E-03
109	CM244	20	45	2.7700E+01
110	NP237	20	28	2.6100E-01
111	TH232	20	18	8.9000E-05
112	U235	20	25	6.9700E-04
113	ZZZZZ	99	0	
114				

Table A-2. Acceptable WIPP TWBID Data Files for PreORG Input

SITE	WIPP TWBID DATA FILENAME		
	CH-TRU	RH-TRU	BURIED TRU
ARCO	arcomp_2.dat	arcomp_3.dat	—
Argonne	argonne2.dat	—	—
ETEC	etec_sl2.dat	—	—
Hanford	hanford2.dat	hanford3.dat	hanford7.dat
INEL	inel__2.dat	inel__3.dat	—
Knolls	—	knolls_3.dat	—
LB Lab	l_b_lab2.dat	—	—
LANL	lanl__2.dat	lanl__3.dat	—
LLNL	llnl__2.dat	—	—
Mound	mound__2.dat	—	—
MURR-DOE	murr__2.dat	—	—
NTS	nts__2.dat	nts__3.dat	—
ORNL	ornl__2.dat	ornl__3.dat	ornl__7.dat
Paducah	paducah2.dat	—	—
Pantex	pantex_2.dat	—	—
Rocky Flats	rocky_f2.dat	—	—
Savannah River	savanna2.dat	savanna3.dat	savanna7.dat
SNL/NM	snl_nm_2.dat	—	snl_nm_7.dat
USARMY	armymc_2.dat	armymc_3.dat	—
West Valley	w_val_d2.dat	w_val_d3.dat	w_val_d7.dat

A typical PreORG run. A listing of what is written to the server follows on the next pages.

C23456789012345678901234567890123456789012345678901234567890123456789012

```

*****
*
*
*      CODE - PREORG. FOR
*
*      JENNIPER LISCUM-POWELL
*
*      CODE-PreORG.for is a program for WIPP TWBID
*
*****
C2345678901234567890123456789012345678901234567890123456789012
  1         2         3         4         5         6         7

```

[illegible][illegible]

```

C      *
C      *          ISSUED BY          *
C      * SANDIA LABORATORIES,       *
C      * A PRIME CONTRACTOR        *
C      ***** TO THE                *
C              UNITED STATES         *
C              DEPARTMENT           *
C              OF                   *
C              ENERGY              *
C      -----NOTICE-----
C
C * THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED *
C * BY THE UNITED STATES GOVERNMENT. NEITHER THE UNITED    *
C * STATES NOR THE UNITED STATES DEPARTMENT OF ENERGY,     *
C * NOR ANY OF THEIR EMPLOYEES,                                *
C * NOR ANY OF THEIR CONTRACTORS, SUBCONTRACTORS, OR THEIR  *
C * EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR   *
C * ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE   *
C *
C * ***** ACCURACY, *****
C *             * COMPLETENESS *
C *             * OR USEFULNESS *
C *             * OF ANY *
C *             * INFORMATION, *
C *             * APPARATUS. *
C * ***** * PRODUCT *****
C *             * OR PROCESS *
C *             * DISCLOSED. *
C *             * OR REPRESENTS *
C *             ** THAT ITS **
C *             * USE WOULD NOT *
C ***** ** INFRINGE ** *****
C             ** PRIVATELY **
C             ** OWNED **
C             ** RIGHTS. **
C             **
C             **
C             **
C             **
C             **

```

C2345678901234567890123456789012345678901234567890123456789012
C 1 2 3 4 5 6 7
CC

THIS PROGRAM TRANSLATES DATA IN THE WIPP BID INTO
ORIGEN2 INPUT FILES TO DECAY INVENTORIES IN THE USER
SPECIFIED SITE DATA FILE TO THE SAME YEAR AS GIVEN
BY THE USER.
ENTER YEAR WISH TO DECAY ALL INVENTORY DATA TO:

1995 ← USER ANSWER

FOR YOUR REFERENCE PREORG.FOR WILL GENERATE A LIST
OF THE ORIGEN2 INPUT FILES CREATED DURING THIS RUN

PLEASE ENTER THE NAME OF THE LIST FILE

Appendix A: Sample PREORG.FOR Data Input File

6 CHARACTERS OR LESS:

CC

samp01 ← USER ANSWER

PLEASE CHOOSE A SITE NAME:

- 1 - ARCO MEDICAL PRODUCTS (ARCOMP)
- 2 - ARGONNE NATIONAL LABORATORY
- 3 - ETEC
- 4 - HANFORD
- 5 - INEL
- 6 - KNOLLS APL
- 7 - LB LABORATORY
- 8 - LOS ALAMOS NATIONAL LABORATORY (LANL)
- 9 - LLNL
- 10 - MOUND PLANT
- 11 - MURR - DOE
- 12 - NEVADA TEST SITE (NTS)
- 13 - OAK RIDGE NATIONAL LABORATORY
- 14 - PADUCAH
- 15 - PANTEX PLANT
- 16 - ROCKY FLATS
- 17 - SAVANNAH RIVER
- 18 - SANDIA NATIONAL LABORATORY - NM
- 19 - US ARMY MATERIAL COMMAND (ARMYMC)
- 20 - WEST VALLEY DEMO
- 21 - OTHER SITE

4 ← USER ANSWER

GENERATING ORIGEN2 INPUT

HANFORD

PLEASE CHOOSE WASTE CATEGORY:

- 1 - CH TRU, 2 - RH TRU, OR 3 - BURIED TRU

1 ← USER ANSWER

PreORG INPUT DATA FILENAME: hanford2.dat
PreORG OUTPUT DATA FILENAME: hanfch95.u5

DEFAULT ENRICHED URANIUM BREAKDOWN IS:

U234 - 0.00
U235 - 0.07
U238 - 0.93

DO YOU WANT A NEW ENRICHED URANIUM BREAKDOWN FOR hanford ?
Enter y (yes) or n (no):

n ← USER ANSWER

CC
CC

INVENTORY DATA FOR THE FOLLOWING YEARS:

1970	1971	1972	1973	1974	1975
1975	1976	1977	1978	1979	1980
1980	1981	1982	1983	1984	1985
1985	1986	1987	1988	1989	1990
1990	1991	1992	1993	0	0

CC

WOULD YOU LIKE TO RUN PROGRAM AGAIN TO GENERATE
MORE ORIGEN2 INPUT FILES? 1-YES 2-NO

CC

1 ← USER ANSWER

PLEASE CHOOSE A SITE NAME:

- 1 - ARCO MEDICAL PRODUCTS (ARCOMP)
- 2 - ARGONNE NATIONAL LABORATORY
- 3 - ETEC
- 4 - HANFORD
- 5 - INEL
- 6 - KNOLLS APL
- 7 - LB LABORATORY
- 8 - LOS ALAMOS NATIONAL LABORATORY (LANL)
- 9 - LLNL
- 10 - MOUND PLANT
- 11 - MURR - DOE
- 12 - NEVADA TEST SITE (NTS)
- 13 - OAK RIDGE NATIONAL LABORATORY
- 14 - PADUCAH
- 15 - PANTEX PLANT
- 16 - ROCKY FLATS
- 17 - SAVANNAH RIVER
- 18 - SANDIA NATIONAL LABORATORY - NM
- 19 - US ARMY MATERIAL COMMAND (ARMYMC)
- 20 - WEST VALLEY DEMO
- 21 - OTHER SITE

8 ← USER ANSWER

GENERATING ORIGEN2 INPUT

LANL

PLEASE CHOOSE WASTE CATEGORY:

- 1 - CH TRU, 2 - RH TRU, OR 3 - BURIED TRU

2 ← USER ANSWER

PreORG INPUT DATA FILENAME: lanl_3.dat
PreORG OUTPUT DATA FILENAME: lanlrx95.u5

DEFAULT ENRICHED URANIUM BREAKDOWN IS:

U234 - 0.00
U235 - 0.07
U238 - 0.93

DO YOU WANT A NEW ENRICHED URANIUM BREAKDOWN FOR lanl_ ?
Enter y (yes) or n (no):

```

y          ← USER ANSWER
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
PLEASE ENTER ENRICHED URANIUM BREAKDOWN AS
ATOM FRACTIONS:
U234 = 0.0          ← USER ANSWER
U235 = 0.30
U238 = 0.70
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

INVENTORY DATA FOR THE FOLLOWING YEARS:
1979    1980    1981    1982    1983    1984
1984    1985    1987    1990    1991    1994
1994      0      0      0      0      0
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

WOULD YOU LIKE TO RUN PROGRAM AGAIN TO GENERATE
MORE ORIGEN2 INPUT FILES? 1-YES 2-NO
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
1          ← USER ANSWER

PLEASE CHOOSE A SITE NAME:
1 - ARCO MEDICAL PRODUCTS (ARCOMP)
2 - ARGONNE NATIONAL LABORATORY
3 - ETEC
4 - HANFORD
5 - INEL
6 - KNOXLS APL
7 - LB LABORATORY
8 - LOS ALAMOS NATIONAL LABORATORY (LANL)
9 - LLNL
10 - MOUND PLANT
11 - MURR - DOE
12 - NEVADA TEST SITE (NTS)
13 - OAK RIDGE NATIONAL LABORATORY
14 - PADUCAH
15 - PANTEX PLANT
16 - ROCKY FLATS
17 - SAVANNAH RIVER
18 - SANDIA NATIONAL LABORATORY - NM
19 - US ARMY MATERIAL COMMAND (ARMYMC)
20 - WEST VALLEY DEMO
21 - OTHER SITE

1          ← USER ANSWER

GENERATING ORIGEN2 INPUT
ARCO MEDICAL PRODUCTS (ARCOMP)
PLEASE CHOOSE WASTE CATEGORY:
1 - CH TRU, 2 - RH TRU, OR 3 - BURIED TRU
1          ← USER ANSWER

PreORG INPUT DATA FILENAME:   arcomp_2.dat

**ERROR**ERROR**ERROR**ERROR**ERROR**ERROR**ERROR**ERROR
      INPUT FILE arcomp_2.dat DOES NOT EXIST
WASTE TYPE: CH FOR THE arcomp_ SITE IS NOT VALID
**ERROR**ERROR**ERROR**ERROR**ERROR**ERROR**ERROR**ERROR

TO CHOOSE ANOTHER SITE ENTER [1] OR
TO EXIT PreORG ENTER [2]:
2          ← USER ANSWER
Stop - Program Terminated

```

APPENDIX B: ORIGEN2.BAT FILE LISTING FOR A PC

Intentionally Left Blank

APPENDIX B: ORIGEN2.BAT FILE LISTING FOR A PC

```

copy %1.u5 tape5.inp >nul
copy %1.u3 tape3.inp >nul
copy %1.u5 dum5.inp
copy %1.u3 dum3.inp
del %1.u*
copy dum5.inp %1.u5
copy dum3.inp %1.u3
echo off
:library
cls
type library.txt
\origen2\code\be rowcol 15,25
\origen2\code\be ask "Which library do you want (1 to 4)? ", 1234 default=1
if errorlevel 4 goto other
if errorlevel 3 goto lmfr
if errorlevel 2 goto bwr
if errorlevel 1 goto pwr
:pwr
cls
type pwr.txt
\origen2\code\be rowcol 17,25
\origen2\code\be ask "Which PWR library do you want (A to J)? ", ABCDEFGHIJ default=1
if errorlevel 10 goto pwr10
if errorlevel 9 goto pwr9
if errorlevel 8 goto pwr8
if errorlevel 7 goto pwr7
if errorlevel 6 goto pwr6
if errorlevel 5 goto pwr5
if errorlevel 4 goto pwr4
if errorlevel 3 goto pwr3
if errorlevel 2 goto pwr2
if errorlevel 1 goto pwr1
:pwr1
copy \origen2\libs\DECAY.lib+\origen2\libs\PWRU.lib tape9.inp >nul
goto photon:
:pwr2
copy \origen2\libs\DECAY.lib+\origen2\libs\PWRPUU.lib tape9.inp >nul
goto photon:
:pwr3
copy \origen2\libs\DECAY.lib+\origen2\libs\PWRPUPU.lib tape9.inp >nul
goto photon:
:pwr4
copy \origen2\libs\DECAY.lib+\origen2\libs\PWRDU3TH.lib tape9.inp >nul
goto photon:
:pwr5
copy \origen2\libs\DECAY.lib+\origen2\libs\PWRPUH.lib tape9.inp >nul
goto photon:
:pwr6
copy \origen2\libs\DECAY.lib+\origen2\libs\PWRU50.lib tape9.inp >nul
goto photon:
:pwr7
copy \origen2\libs\DECAY.lib+\origen2\libs\PWRD5D35.lib tape9.inp >nul
goto photon:
:pwr8
copy \origen2\libs\DECAY.lib+\origen2\libs\PWRD5D33.lib tape9.inp >nul
goto photon:
:pwr9
copy \origen2\libs\DECAY.lib+\origen2\libs\PWRUS.lib tape9.inp >nul
goto photon:
:pwr10
copy \origen2\libs\DECAY.lib+\origen2\libs\PWRUE.lib tape9.inp >nul
goto photon:
:bwr
cls
type bwr.txt
\origen2\code\be rowcol 15,25
\origen2\code\be ask "Which BWR library do you want (A to F)? ", ABCDEF default=1
if errorlevel 6 goto bwr6
if errorlevel 5 goto bwr5
if errorlevel 4 goto bwr4
if errorlevel 3 goto bwr3
if errorlevel 2 goto bwr2
if errorlevel 1 goto bwr1
:bwr1
copy \origen2\libs\DECAY.lib+\origen2\libs\BWRU.lib tape9.inp >nul
goto photon:
:bwr2
copy \origen2\libs\DECAY.lib+\origen2\libs\BWRPUU.lib tape9.inp >nul
goto photon:
:bwr3
copy \origen2\libs\DECAY.lib+\origen2\libs\BWRPUPU.lib tape9.inp >nul
goto photon:
:bwr4
copy \origen2\libs\DECAY.lib+\origen2\libs\BWRUS.lib tape9.inp >nul
goto photon:
:bwr5
copy \origen2\libs\DECAY.lib+\origen2\libs\BWRU50.lib tape9.inp >nul
goto photon:
:bwr6
copy \origen2\libs\DECAY.lib+\origen2\libs\BWRUE.lib tape9.inp >nul
goto photon:
:lmfr
type lmfr.txt
\origen2\code\be rowcol 24,25

```

Appendix B: ORIGEN2.BAT File Listing for PC

```

\origen2\code\be ask "Which LMFBF library do you want (A to X)? ", ABCDEFGHIJKLMNOPQRSTUVWXYZ default=A
if errorlevel 24 goto lmfbf24
if errorlevel 23 goto lmfbf23
if errorlevel 22 goto lmfbf22
if errorlevel 21 goto lmfbf21
if errorlevel 20 goto lmfbf20
if errorlevel 19 goto lmfbf19
if errorlevel 18 goto lmfbf18
if errorlevel 17 goto lmfbf17
if errorlevel 16 goto lmfbf16
if errorlevel 15 goto lmfbf15
if errorlevel 14 goto lmfbf14
if errorlevel 13 goto lmfbf13
if errorlevel 12 goto lmfbf12
if errorlevel 11 goto lmfbf11
if errorlevel 10 goto lmfbf10
if errorlevel 9 goto lmfbf9
if errorlevel 8 goto lmfbf8
if errorlevel 7 goto lmfbf7
if errorlevel 6 goto lmfbf6
if errorlevel 5 goto lmfbf5
if errorlevel 4 goto lmfbf4
if errorlevel 3 goto lmfbf3
if errorlevel 2 goto lmfbf2
if errorlevel 1 goto lmfbf1
:lmfbf1
copy \origen2\libs\DECAY.lib+\origen2\libs\EMOFUUUC.lib tape9.inp >nul
goto photon:
:lmfbf2
copy \origen2\libs\DECAY.lib+\origen2\libs\EMOFUUUA.lib tape9.inp >nul
goto photon:
:lmfbf3
copy \origen2\libs\DECAY.lib+\origen2\libs\EMOFUUUR.lib tape9.inp >nul
goto photon:
:lmfbf4
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOFUUUC.lib tape9.inp >nul
goto photon:
:lmfbf5
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOFUUUA.lib tape9.inp >nul
goto photon:
:lmfbf6
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOFUUUR.lib tape9.inp >nul
goto photon:
:lmfbf7
copy \origen2\libs\DECAY.lib+\origen2\libs\AMORUUUC.lib tape9.inp >nul
goto photon:
:lmfbf8
copy \origen2\libs\DECAY.lib+\origen2\libs\AMORUUUA.lib tape9.inp >nul
goto photon:
:lmfbf9
copy \origen2\libs\DECAY.lib+\origen2\libs\AMORUUUR.lib tape9.inp >nul
goto photon:
:lmfbf10
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOFUUTC.lib tape9.inp >nul
goto photon:
:lmfbf11
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOFUUTA.lib tape9.inp >nul
goto photon:
:lmfbf12
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOFUUTR.lib tape9.inp >nul
goto photon:
:lmfbf13
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOPTTTC.lib tape9.inp >nul
goto photon:
:lmfbf14
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOPTTTA.lib tape9.inp >nul
goto photon:
:lmfbf15
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOPTTTR.lib tape9.inp >nul
goto photon:
:lmfbf16
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOOTTTC.lib tape9.inp >nul
goto photon:
:lmfbf17
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOOTTTA.lib tape9.inp >nul
goto photon:
:lmfbf18
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOOTTTR.lib tape9.inp >nul
goto photon:
:lmfbf19
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOI1TTC.lib tape9.inp >nul
goto photon:
:lmfbf20
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOI1TTA.lib tape9.inp >nul
goto photon:
:lmfbf21
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOI1TTR.lib tape9.inp >nul
goto photon:
:lmfbf22
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOI2TTC.lib tape9.inp >nul
goto photon:
:lmfbf23
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOI2TTA.lib tape9.inp >nul
goto photon:
:lmfbf24
copy \origen2\libs\DECAY.lib+\origen2\libs\AMOI2TTR.lib tape9.inp >nul
goto photon:
:other
cls

```

```

type other.txt
\origen2\code\be rowcol 15,25
\origen2\code\be ask "Which other library do you want (A to H)? ", ABCDEFGH default=1
if errorlevel 8 goto other8
if errorlevel 7 goto other7
if errorlevel 6 goto other6
if errorlevel 5 goto other5
if errorlevel 4 goto other4
if errorlevel 3 goto other3
if errorlevel 2 goto other2
if errorlevel 1 goto other1
:other1
copy \origen2\libs\DECAY.lib+\origen2\libs\FFTFC.lib tape9.inp >nul
goto photon:
:other2
copy \origen2\libs\DECAY.lib+\origen2\libs\THERMAL.lib tape9.inp >nul
goto photon:
:other3
copy \origen2\libs\DECAY.lib+\origen2\libs\CANDUNAU.lib tape9.inp >nul
goto photon:
:other4
copy \origen2\libs\DECAY.lib+\origen2\libs\CANDUSEU.lib tape9.inp >nul
goto photon:
:other5
copy \origen2\libs\DECAY.lib+\origen2\libs\CRBRC.lib tape9.inp >nul
goto photon:
:other6
copy \origen2\libs\DECAY.lib+\origen2\libs\CRBRA.lib tape9.inp >nul
goto photon:
:other7
copy \origen2\libs\DECAY.lib+\origen2\libs\CRBRR.lib tape9.inp >nul
goto photon:
:other8
copy \origen2\libs\DECAY.lib+\origen2\libs\CRBRI.lib tape9.inp >nul
goto photon:
:photon
cls
type photon.txt
\origen2\code\be rowcol 15,25
\origen2\code\be ask "Which photon library do you want (1 to 3)? ", 123 default=1
if errorlevel 3 goto photon3
if errorlevel 2 goto photon2
if errorlevel 1 goto photon1
:photon1
copy \origen2\libs\GXUO2BRM.lib tape10.inp >nul
goto run:
:photon2
copy \origen2\libs\GXHZOBRM.lib tape10.inp >nul
goto run:
:photon3
copy \origen2\libs\GXNOBREM.lib tape10.inp >nul
goto run:
:run
echo on
copy %1.u5 tape5.inp >nul
copy %1.u3 tape3.inp >nul
rem pause
\origen2\code\origen2
\origen2\code\gamma
rem pause
rem combine and save files from run
rem copy tape12.out+tape6.out %1.u6 >nul
rem copy tape13.out+tape11.out %1.u11 >nul
ren tape6.out %1.u6
ren tape7.out %1.u7
ren tape11.out %1.u11
ren tape12.out %1.u12
ren tape13.out %1.u13
ren tape15.out %1.u15
ren tape16.out %1.u16
ren tape20.out %1.u20
ren tape21.out %1.u21
ren tape50.out %1.u50
rem copy %1.u21 \cohort\*.asc
rem cleanup files
del tape*.inp
del tape*.out
echo off
echo *****
echo **
echo ** O R I G E N 2 **
echo ** Oak Ridge Isotope GENERation and Depletion Code **
echo ** Version 2.1 (8-1-91) **
echo **
echo *****
echo **
echo ** Developed by: Oak Ridge National Laboratory **
echo ** Chemical Technology Division **
echo **
echo ** Technical Contact: Scott B. Ludwig **
echo ** (615) 574-7916 FTS 624-7916 **
echo **
echo ** Distributed by: Radiation Shielding Information Center (RSIC) **
echo ** Oak Ridge National Laboratory **
echo ** P.O. Box 2008 **
echo ** Oak Ridge, TN 37831 **
echo ** (615) 574-6176 FTS 624-6176 **
echo *****

```

Appendix B: ORIGEN2.BAT File Listing for PC

```
echo .....
echo ** Execution continuing ... **
echo .....
echo **
echo **      Version 2.1 (8-1-91) for mainframes and 80386 or 80486 PCs      **
echo **
echo .....
echo ***** O R I G E N 2 - Version 2.1 *****
echo ***** Execution Completed *****
echo .....
echo on
```

**APPENDIX C: ORIGEN2 COMMAND DESCRIPTIONS AND SAMPLE ORIGEN2
INPUT FILE CREATED BY PREORG**

Intentionally Left Blank

APPENDIX C: ORIGEN2 Command Descriptions and Sample ORIGEN2 Input File Created by Preorg

C.1 ORIGEN2 Command Description

For the calculations performed by ORIGEN2 with the input files generated by PreORG, the commands described in Table C-1 are utilized. Table C-1 list the commands in the order in which they appear in the sample ORIGEN2 input file listed in this appendix. The descriptions offered in the table are general and come from the ORIGEN2 manual (Croff, 1980). The table also shows the maximum number of times the command can be used in a given input file.

Table C-1. Description of ORIGEN2 Commands Used in PreORG

ORIGEN2 COMMAND	MAXIMUM NUMBER	DESCRIPTION
BAS	10	Supplies case basis printed in ORIGEN2 output file Consists of up to 71 alphanumeric characters
TIT	20	Case title Consists of up to 71 alphanumeric characters
RDA	300	Comment card among input commands added by user to explain input file
LIP	5	Controls the printing of the input data libraries
LIB	5	Reads decay and cross-section libraries
PHO	5	Reads photon production rate per disintegration in 18 energy groups
INP	15	Calls for nuclide composition, continuous nuclide feed rate, or continuous removal rate to be read
DEC	150	Decay nuclide concentration data in one vector for a single decay interval
MOV	99	Moves nuclide composition in one vector to another vector, nuclide by nuclide
HED	50	Assigns headings to a specified vector
OPTA	20	Specifies which output table types (nuclide, element or summary) are to be printed for the actinides
OPTF	20	Specifies which output table types (nuclide, element or summary) are to be printed for the fission products
OPTL	20	Specifies which output table types (nuclide, element or summary) are to be printed for the activation products
OUT	20	Calls for calculated results in some or all output vectors to be printed to a file
ADD	30	Adds the nuclide concentration data of two vectors together, nuclide by nuclide
STP	n/a	Executes the set of commands preceding the STP command, then reads and executes commands that follow.
END	1	Terminates execution of ORIGEN2

C.2 ORIGEN2 Output Tables Description

As described in Section 2.2, ORIGEN2 can calculate several different characteristics of nuclear materials. The following table lists output tables that can be generated by ORIGEN2. The user can control whether or not any of the first 24 tables will be printed; the remaining are always included in the output file. This table can be found in the ORIGEN2 manual (Croff, 1980). For the purposes of the calculations that are desired from the Waste Isolation Pilot Plant (WIPP) Baseline Inventory Database (BID) input files, only Table 7 (see Table C-2) is printed to the output file.

Table C-2. Description of ORIGEN2 Output Tables

TABLE #	DESCRIPTION	UNITS
1	Isotopic composition of each element	atom fraction
2	Isotopic composition of each element	weight fraction
3	Composition	gram-atoms
4	Composition	atom fraction
5	Composition	grams
6	Composition	weight fraction
7	Radioactivity (total)	Ci
8	Radioactivity (total)	fractional
9	Thermal power	watts
10	Thermal power	fractional
11	Radioactivity (total)	Bq
12	Radioactivity (total)	fractional
13	Radioactive inhalation hazard	m ³ air
14	Radioactive inhalation hazard	fractional
15	Radioactive ingestion hazard	m ³ water
16	Radioactive ingestion hazard	fractional
17	Chemical ingestion hazard	m ³ water
18	Chemical ingestion hazard	fractional
19	Neutron absorption rate	neutrons/sec
20	Neutron absorption rate	fractional
21	Neutron-induced fission rate	fissions/sec
22	Neutron-induced fission rate	fractional
23	Radioactivity (alpha)	Ci
24	Radioactivity (alpha)	fractional
25	(alpha,n) neutron production	neutrons/sec
26	Spontaneous fission neutron production	neutrons/sec
27	Photon emission rate	photons/sec
28	Set test parameter ERR	—

To specify the output of the tables listed in Table C-2, a number corresponding to those listed in Table C-3 must be placed on the OPTA, OPTF, and OPTL cards for each of the 24 tables. On each of these cards, 24 NOPTA, NOPTF, or NOPTL values must be entered. ORIGEN2 prints the tables by nuclide, by element, and in a summary table.

Table C-3. Specification of Output Tables on OPTA, OPTF and OPTL Cards

NOPTA(I) NOPTF(I) NOPTL(I)	TABLE TYPE PRINTED		
	NUCLIDE	ELEMENT	SUMMARY
1	Yes	Yes	Yes
2	Yes	Yes	No
3	Yes	No	Yes
4	No	Yes	Yes
5	Yes	No	No
6	No	Yes	No
7	No	No	Yes
8	No	No	No

ORIGEN2 specifies time units for decay intervals and others using the numbers listed in Table C-4 to represent a given unit.

Table C-4. Time Unit Designation

1 = seconds
2 = minutes
3 = hours
4 = days
5 = years
6 = stable
7 = 10^3 years (kY)
8 = 10^6 years (MY)
9 = 10^9 years (GY)

C-3. Sample ORIGEN2 Input File Created by PreORG

The following pages are a sample ORIGEN2 input file created by PreORG for the Oak Ridge National Laboratory (ORNL) site for contact-handled transuranic (CH TRU) waste (ORNLCH93.U5). The spacing in the input file should be observed in all input files for ORIGEN2.

Sample ORIGEN2 Input File Created by PreORG

omlch93.u5

1 -1 > Change reprocessing fractional
2 -1 recoveries
3 -1

4 BAS WIPP BASELINE INVENTORY DATA ← Title Card
5 TIT ORNL STORED CONTACT HANDLED TRU WASTE
6 RDA ORNL ← Comment card with Site Name
7 LIP 0 0 0 — Library Print Card
8 LIB 0 1 2 3 0 0 0 9 3 0 1 1 ← Library Card - Designates
9 PHO 101 102 103 10 Data Libraries Used By
10 RDA INITIAL VALUES FOR: 1970 ORIGEN2. For DECAY
11 INP -1 1 -1 -1 1 1 calculations the 1, 2, 3
12 RDA INITIAL VALUES FOR: 1971 designate the appropriate
13 INP -2 1 -1 -1 1 1 libraries.
14 RDA INITIAL VALUES FOR: 1972
15 INP -3 1 -1 -1 1 1
16 RDA INITIAL VALUES FOR: 1973
17 INP -4 1 -1 -1 1 1
18 RDA INITIAL VALUES FOR: 1974
19 INP -5 1 -1 -1 1 1
20 RDA INITIAL VALUES FOR: 1975
21 INP -6 1 -1 -1 1 1
22 RDA INITIAL VALUES FOR: 1976
23 INP -7 1 -1 -1 1 1
24 RDA INITIAL VALUES FOR: 1977
25 INP -8 1 -1 -1 1 1
26 RDA INITIAL VALUES FOR: 1978
27 INP -9 1 -1 -1 1 1
28 RDA INITIAL VALUES FOR: 1979
29 INP -10 1 -1 -1 1 1
30 RDA INITIAL VALUES FOR: 1980
31 INP -11 1 -1 -1 1 1
32 RDA INITIAL VALUES FOR: 1981
33 INP -12 1 -1 -1 1 1
34 RDA INITIAL VALUES FOR: 1982
35 INP -13 1 -1 -1 1 1
36 RDA INITIAL VALUES FOR: 1983
37 INP -14 1 -1 -1 1 1
38 RDA INITIAL VALUES FOR: 1984
39 INP -15 1 -1 -1 1 1

40 BAS DECAY INITIAL INVENTORIES TO: 1993:ornl_2 ← PreORG input file is
41 RDA DECAY YEAR 1970 TO 23.00 YRS oml_2.dat
42 DEC 23.00 -1 1 5 1 • ORNL site

43 RDA OUTPUT 1970 DECAY TO 1993 DATA
44 MOV 1 2 0 1
45 MOV -1 1 0 1
46 HED 1 1970
47 HED 2 1993
48 RDA PRINT NUCLIDE QUANTITIES IN CURIES
49 OPTA 6*8 5 17*8
50 OPTF 6*8 5 17*8
51 OPTL 6*8 5 17*8
52 RDA OUTPUT COMMANDS
53 OUT 2 1 -1 0
54 RDA DECAY YEAR 1971 TO 22.00 YRS
55 DEC 22.00 -2 2 5 1
56 RDA OUTPUT 1971 DECAY TO 1993 DATA
57 MOV 2 2 0 1
58 MOV -2 1 0 1
59 HED 1 1971
60 HED 2 1993
61 RDA PRINT NUCLIDE QUANTITIES IN CURIES
62 OPTA 6*8 5 17*8

INPUT cards read initial nuclide inventories attached after STP card into vectors to be decayed out to 1993.

INP 1 1 -1 -1 1 1
Store in vector (-1)
Read initial nuclide inventories in grams
Determine continuous nuclide removal and feed rates if needed

DECAY card instructs ORIGEN2 to perform decay and accumulation calculations for nuclide concentrations stored in a given vector over a given interval:

DEC 23.00 -1 1 5 1
Length of decay interval
Vector where initial nuclide concentrations are stored
Time units (Table C-4)
Vector to store nuclide concentrations at end of decay interval
Decay interval length is independent of previous decay intervals used with prior DEC cards

Sample ORIGEN2 Input File Created by PreORG

```

63 OPTF 6*8 5 17*8
64 OPTL 6*8 5 17*8
65 RDA OUTPUT COMMANDS
66 OUT 2 1 -1 0
67 RDA DECAY YEAR 1972 TO 21.00 YRS
68 DEC 21.00 -3 3 5 1
69 RDA OUTPUT 1972 DECAY TO 1993 DATA
70 MOV 3 2 0 1
71 MOV -3 1 0 1
72 HED 1 1972
73 HED 2 1993
74 RDA PRINT NUCLIDE QUANTITIES IN CURIES
75 OPTA 6*8 5 17*8
76 OPTF 6*8 5 17*8
77 OPTL 6*8 5 17*8
78 RDA OUTPUT COMMANDS
79 OUT 2 1 -1 0
80 RDA DECAY YEAR 1973 TO 20.00 YRS
81 DEC 20.00 -4 4 5 1
82 RDA OUTPUT 1973 DECAY TO 1993 DATA
83 MOV 4 2 0 1
84 MOV -4 1 0 1
85 HED 1 1973
86 HED 2 1993
87 RDA PRINT NUCLIDE QUANTITIES IN CURIES
88 OPTA 6*8 5 17*8
89 OPTF 6*8 5 17*8
90 OPTL 6*8 5 17*8
91 RDA OUTPUT COMMANDS
92 OUT 2 1 -1 0
93 RDA DECAY YEAR 1974 TO 19.00 YRS
94 DEC 19.00 -5 5 5 1
95 RDA OUTPUT 1974 DECAY TO 1993 DATA
96 MOV 5 2 0 1
97 MOV -5 1 0 1
98 HED 1 1974
99 HED 2 1993
100 RDA PRINT NUCLIDE QUANTITIES IN CURIES
101 OPTA 6*8 5 17*8
102 OPTF 6*8 5 17*8
103 OPTL 6*8 5 17*8
104 RDA OUTPUT COMMANDS
105 OUT 2 1 -1 0
106 RDA DECAY YEAR 1975 TO 18.00 YRS
107 DEC 18.00 -6 6 5 1
108 RDA OUTPUT 1975 DECAY TO 1993 DATA
109 MOV 6 2 0 1
110 MOV -6 1 0 1
111 HED 1 1975
112 HED 2 1993
113 RDA PRINT NUCLIDE QUANTITIES IN CURIES
114 OPTA 6*8 5 17*8
115 OPTF 6*8 5 17*8
116 OPTL 6*8 5 17*8
117 RDA OUTPUT COMMANDS
118 OUT 2 1 -1 0
119 RDA DECAY YEAR 1976 TO 17.00 YRS
120 DEC 17.00 -7 7 5 1
121 RDA OUTPUT 1976 DECAY TO 1993 DATA
122 MOV 7 2 0 1
123 MOV -7 1 0 1
124 HED 1 1976
125 HED 2 1993
126 RDA PRINT NUCLIDE QUANTITIES IN CURIES
127 OPTA 6*8 5 17*8
128 OPTF 6*8 5 17*8
129 OPTL 6*8 5 17*8
130 RDA OUTPUT COMMANDS
131 OUT 2 1 -1 0
132 RDA DECAY YEAR 1977 TO 16.00 YRS
133 DEC 16.00 -8 8 5 1
134 RDA OUTPUT 1977 DECAY TO 1993 DATA
135 MOV 8 2 0 1
136 MOV -8 1 0 1
137 HED 1 1977
138 HED 2 1993

```

MOV cards move vectors to output easily:
MOV 3 2 0 1

Multiply all data by 1.0

Move nuclide concentrations in vector 3 to vector 2

No external multiplier used

Assign headings to vectors for output:
HED 1 1973
Vector 1 will be printed with 1973 for a heading

Instructs ORIGEN2 which tables to print in output for each of the three libraries

- Actinides
- Fission Products
- Activation Products

See Table C-2 and Table C-3 for more information.

Print 2 vectors in output file .u6

Sample ORIGEN2 Input File Created by PreORG

ornlch93.u5

```

139 RDA PRINT NUCLIDE QUANTITIES IN CURIES
140 OPTA 6*8 5 17*8
141 OPTF 6*8 5 17*8
142 OPTL 6*8 5 17*8
143 RDA OUTPUT COMMANDS
144 OUT 2 1 -1 0
145 RDA DECAY YEAR 1978 TO 15.00 YRS
146 DEC 15.00 -9 9 5 1
147 RDA OUTPUT 1978 DECAY TO 1993 DATA
148 MOV 9 2 0 1
149 MOV -9 1 0 1
150 HED 1 1978
151 HED 2 1993
152 RDA PRINT NUCLIDE QUANTITIES IN CURIES
153 OPTA 6*8 5 17*8
154 OPTF 6*8 5 17*8
155 OPTL 6*8 5 17*8
156 RDA OUTPUT COMMANDS
157 OUT 2 1 -1 0
158 RDA DECAY YEAR 1979 TO 14.00 YRS
159 DEC 14.00 -10 10 5 1
160 RDA OUTPUT 1979 DECAY TO 1993 DATA
161 MOV 10 2 0 1
162 MOV -10 1 0 1
163 HED 1 1979
164 HED 2 1993
165 RDA PRINT NUCLIDE QUANTITIES IN CURIES
166 OPTA 6*8 5 17*8
167 OPTF 6*8 5 17*8
168 OPTL 6*8 5 17*8
169 RDA OUTPUT COMMANDS
170 OUT 2 1 -1 0
171 RDA DECAY YEAR 1980 TO 13.00 YRS
172 DEC 13.00 -11 11 5 1
173 RDA OUTPUT 1980 DECAY TO 1993 DATA
174 MOV 11 2 0 1
175 MOV -11 1 0 1
176 HED 1 1980
177 HED 2 1993
178 RDA PRINT NUCLIDE QUANTITIES IN CURIES
179 OPTA 6*8 5 17*8
180 OPTF 6*8 5 17*8
181 OPTL 6*8 5 17*8
182 RDA OUTPUT COMMANDS
183 OUT 2 1 -1 0
184 RDA DECAY YEAR 1981 TO 12.00 YRS
185 DEC 12.00 -12 12 5 1
186 RDA OUTPUT 1981 DECAY TO 1993 DATA
187 MOV 12 2 0 1
188 MOV -12 1 0 1
189 HED 1 1981
190 HED 2 1993
191 RDA PRINT NUCLIDE QUANTITIES IN CURIES
192 OPTA 6*8 5 17*8
193 OPTF 6*8 5 17*8
194 OPTL 6*8 5 17*8
195 RDA OUTPUT COMMANDS
196 OUT 2 1 -1 0
197 RDA DECAY YEAR 1982 TO 11.00 YRS
198 DEC 11.00 -13 13 5 1
199 RDA OUTPUT 1982 DECAY TO 1993 DATA
200 MOV 13 2 0 1
201 MOV -13 1 0 1
202 HED 1 1982
203 HED 2 1993
204 RDA PRINT NUCLIDE QUANTITIES IN CURIES
205 OPTA 6*8 5 17*8
206 OPTF 6*8 5 17*8
207 OPTL 6*8 5 17*8
208 RDA OUTPUT COMMANDS
209 OUT 2 1 -1 0
210 RDA DECAY YEAR 1983 TO 10.00 YRS
211 DEC 10.00 -14 14 5 1
212 RDA OUTPUT 1983 DECAY TO 1993 DATA
213 MOV 14 2 0 1
214 MOV -14 1 0 1

```

Sample ORIGEN2 Input File Created by PreORG

omlch93.u5

```

215 HED 1 1983
216 HED 2 1993
217 RDA PRINT NUCLIDE QUANTITIES IN CURIES
218 OPTA 6*8 5 17*8
219 OPTF 6*8 5 17*8
220 OPTL 6*8 5 17*8
221 RDA OUTPUT COMMANDS
222 OUT 2 1 -1 0
223 RDA DECAY YEAR 1984 TO 9.00 YRS
224 DEC 9.00 -15 15 5 1
225 RDA OUTPUT 1984 DECAY TO 1993 DATA
226 MOV 15 2 0 1
227 MOV -15 1 0 1
228 HED 1 1984
229 HED 2 1993
230 RDA PRINT NUCLIDE QUANTITIES IN CURIES
231 OPTA 6*8 5 17*8
232 OPTF 6*8 5 17*8
233 OPTL 6*8 5 17*8
234 RDA OUTPUT COMMANDS
235 OUT 2 1 -1 0
236 STP 1
237 2 942390 .1126E+03 952410 .8277E-04 922330 .2934E-01 0 0
238 2 952430 .5068E-03 982520 .1860E-08 962440 .5477E-01 0 0
239 2 922350 .3470E+02 0 0
240 0
241 1 270600 .8835E-09 0 0
242 2 942380 .2489E+03 942390 .5262E+03 942400 .7726E+01 0 0
243 2 942410 .3961E+01 952410 .2565E+03 922330 .1571E+02 0 0
244 2 952430 .1244E+02 972490 .2210E-02 982520 .2121E-02 0 0
245 2 962420 .1004E-01 962440 .6095E+01 932370 .4583E+04 0 0
246 2 922350 .1897E+03 0 0
247 0
248 2 942380 .4557E+02 942390 .8030E+03 942400 .2059E+01 0 0
249 2 942410 .2913E+02 952410 .4721E+02 952430 .1857E+02 0 0
250 2 972490 .1227E-05 982520 .1029E-02 962440 .5193E+01 0 0
251 2 932370 .1547E+03 902320 .3046E+06 922350 .7681E+02 0 0
252 2 922380 .6545E+02 0 0
253 0
254 2 942380 .3412E+01 942390 .5889E+02 942400 .1576E+01 0 0
255 2 942410 .8156E+00 952410 .4051E+01 922330 .2263E+03 0 0
256 2 952430 .1104E+02 972490 .1111E-02 982520 .9952E-01 0 0
257 2 962440 .6812E+01 932370 .8769E+01 922350 .1069E+03 0 0
258 0
259 1 611470 .2005E-02 0 0
260 2 942380 .5580E+00 952410 .1326E+01 922330 .1009E+04 0 0
261 2 952430 .4641E+00 972490 .2765E+00 982520 .3348E-03 0 0
262 2 962440 .6268E+01 932370 .3902E+01 902320 .9667E+03 0 0
263 0
264 2 942380 .5843E-05 942390 .5970E+02 952410 .9443E+00 0 0
265 2 922330 .2015E+04 972490 .6227E-07 982520 .3199E-05 0 0
266 2 962440 .7764E-03 932370 .1307E+03 902320 .1988E+04 0 0
267 3 631520 .5780E-08 0 0
268 0
269 2 942380 .5843E-05 942390 .8223E+02 952410 .5975E-03 0 0
270 2 922330 .1922E+04 952430 .1004E-02 972490 .5531E-04 0 0
271 2 982520 .1860E-03 962440 .5180E-03 932370 .1759E+03 0 0
272 2 902320 .9940E+03 922350 .5275E+02 922380 .1389E+05 0 0
273 0
274 1 551370 .1149E-04 0 0
275 2 942380 .2764E+00 942390 .1786E+04 942410 .3816E-04 0 0
276 2 942420 .1373E+04 952410 .2221E+01 952430 .1505E-03 0 0
277 2 982520 .1860E-03 962420 .3025E-06 962440 .3425E+00 0 0
278 2 932370 .3703E+03 902320 .8117E+03 922350 .3225E+03 0 0

```

Execute all ORIGEN2 commands given above then continue with next set of ORIGEN2 commands which follow initial nuclide concentrations.

First year of inventory in oml_2.dat input file in grams

Indicates end of card of data

Data that follows is read when new input card is encountered.

1970

1971

1972

1973

1974

1975

1976

1977

Appendix C: ORIGEN2 Command Descriptions and Sample ORIGEN2 Input File Created by PreORG

Sample ORIGEN2 Input File Created by PreORG

ornlch93.u5

```

279 0
280 1 430990 .1049E+04 0 0
281 2 942380 .5136E+00 942390 .1157E+03 942410 .9709E-06 0 0
282 2 942420 .3065E+01 952410 .2063E+02 922330 .2872E+02 0 0
283 2 982520 .2865E-03 962420 .5022E-02 962440 .4748E-02 0 0
284 2 992530 .3175E-03 992541 .3474E-04 932370 .2398E+04 0 0
285 2 902320 .2043E+04 922350 .4812E+03 922380 .6248E+05 0 0
286 0
287 1 380900 .1318E+02 551370 .3447E+02 0 0
288 2 942380 .1928E+01 942390 .3717E+02 942410 .8738E-01 0 0
289 2 942420 .5344E+01 952410 .6878E+01 922330 .8080E+03 0 0
290 2 952430 .1054E-02 982490 .2442E-03 982520 .2455E-03 0 0
291 2 962440 .4834E-01 932370 .1518E+03 902320 .4560E+03 0 0
292 2 922350 .1902E+03 922380 .4046E+03 0 0
293 0
294 2 942390 .6565E+02 942400 .1826E+01 942410 .4855E-04 0 0
295 2 942420 .4218E+02 952410 .5858E+01 922330 .1643E+03 0 0
296 2 952430 .8831E+00 982520 .1644E-03 962440 .1208E+01 0 0
297 2 932370 .1169E+02 922350 .2776E+01 922380 .5950E+02 0 0
298 0
299 2 942380 .8998E+02 942390 .8577E+02 942400 .1668E+02 0 0
300 2 952410 .1277E+02 922330 .5424E+03 952430 .1907E+00 0 0
301 2 982520 .8185E-04 962440 .2411E+01 932370 .1788E+02 0 0
302 2 922350 .7311E+02 922380 .5950E+01 0 0
303 0
304 2 942380 .1169E-05 942390 .3218E+03 942410 .4855E-02 0 0
305 2 952410 .4459E+00 922330 .1963E+01 952430 .1024E-01 0 0
306 2 972490 .1404E-03 982520 .3720E-06 962440 .2881E-01 0 0
307 2 932370 .7095E+02 922350 .3239E+01 922380 .5950E+01 0 0
308 0
309 1 551370 .1149E-06 0 0
310 2 942380 .6427E-05 942390 .1200E+03 942400 .5268E+01 0 0
311 2 942420 .2125E+01 952410 .2329E+01 922330 .4898E+01 0 0
312 2 982520 .3720E-05 962440 .6775E+00 882260 .1184E-01 0 0
313 2 902320 .4560E+02 922350 .1758E+03 0 0
314 0
315 2 942380 .4271E+02 942390 .1604E+04 942400 .2239E+04 0 0
316 2 942410 .7457E+03 952410 .2914E-02 922330 .5941E+03 0 0
317 2 982520 .1427E-03 962440 .1114E-01 902300 .3067E-01 0 0
318 2 922350 .6987E+02 0 0
319 0
320 -1
321 -1
322 -1
323 BAS WIPP BASELINE INVENTORY DATA
324 TIT ORNL STORED CONTACT HANDLED TRU WASTE
325 RDA ORNL
326 LIP 0 0 0
327 LIB 0 1 2 3 0 0 0 9 3 0 1 1
328 PHO 101 102 103 10
329 RDA INITIAL VALUES FOR: 1985
330 INP -1 1 -1 -1 1 1
331 RDA INITIAL VALUES FOR: 1986
332 INP -2 1 -1 -1 1 1
333 RDA INITIAL VALUES FOR: 1987
334 INP -3 1 -1 -1 1 1
335 RDA INITIAL VALUES FOR: 1988
336 INP -4 1 -1 -1 1 1
337 RDA INITIAL VALUES FOR: 1989
338 INP -5 1 -1 -1 1 1
339 RDA INITIAL VALUES FOR: 1990
340 INP -6 1 -1 -1 1 1
341 RDA INITIAL VALUES FOR: 1991
342 INP -7 1 -1 -1 1 1
343 RDA INITIAL VALUES FOR: 1992
344 INP -8 1 -1 -1 1 1
345 RDA INITIAL VALUES FOR: 1993
346 INP -9 1 -1 -1 1 1
347 BAS DECAY INITIAL INVENTORIES TO: 1993:ornl___2
348 RDA DECAY YEAR 1985 TO 8.00 YRS
349 DEC 8.00 -1 1 5 1
350 RDA OUTPUT 1985 DECAY TO 1993 DATA
351 MOV 1 2 0 1
352 MOV -1 1 0 1
353 HED 1 1985
354 HED 2 1993

```

Start New Instructions

Nuclide identifier and
concentration pair - limit of
3 per card.

Appendix C: ORIGEN2 Command Descriptions and Sample ORIGEN2 Input File Created by PreORG

Sample ORIGEN2 Input File Created by PreORG

ornlch93.u5

```

355 RDA PRINT NUCLIDE QUANTITIES IN CURIES
356 OPTA 6*8 5 17*8
357 OPTF 6*8 5 17*8
358 OPTL 6*8 5 17*8
359 RDA OUTPUT COMMANDS
360 OUT 2 1 -1 0
361 RDA DECAY YEAR 1986 TO 7.00 YRS
362 DEC 7.00 -2 2 5 1
363 RDA OUTPUT 1986 DECAY TO 1993 DATA
364 MOV 2 2 0 1
365 MOV -2 1 0 1
366 HED 1 1986
367 HED 2 1993
368 RDA PRINT NUCLIDE QUANTITIES IN CURIES
369 OPTA 6*8 5 17*8
370 OPTF 6*8 5 17*8
371 OPTL 6*8 5 17*8
372 RDA OUTPUT COMMANDS
373 OUT 2 1 -1 0
374 RDA DECAY YEAR 1987 TO 6.00 YRS
375 DEC 6.00 -3 3 5 1
376 RDA OUTPUT 1987 DECAY TO 1993 DATA
377 MOV 3 2 0 1
378 MOV -3 1 0 1
379 HED 1 1987
380 HED 2 1993
381 RDA PRINT NUCLIDE QUANTITIES IN CURIES
382 OPTA 6*8 5 17*8
383 OPTF 6*8 5 17*8
384 OPTL 6*8 5 17*8
385 RDA OUTPUT COMMANDS
386 OUT 2 1 -1 0
387 RDA DECAY YEAR 1988 TO 5.00 YRS
388 DEC 5.00 -4 4 5 1
389 RDA OUTPUT 1988 DECAY TO 1993 DATA
390 MOV 4 2 0 1
391 MOV -4 1 0 1
392 HED 1 1988
393 HED 2 1993
394 RDA PRINT NUCLIDE QUANTITIES IN CURIES
395 OPTA 6*8 5 17*8
396 OPTF 6*8 5 17*8
397 OPTL 6*8 5 17*8
398 RDA OUTPUT COMMANDS
399 OUT 2 1 -1 0
400 RDA DECAY YEAR 1989 TO 4.00 YRS
401 DEC 4.00 -5 5 5 1
402 RDA OUTPUT 1989 DECAY TO 1993 DATA
403 MOV 5 2 0 1
404 MOV -5 1 0 1
405 HED 1 1989
406 HED 2 1993
407 RDA PRINT NUCLIDE QUANTITIES IN CURIES
408 OPTA 6*8 5 17*8
409 OPTF 6*8 5 17*8
410 OPTL 6*8 5 17*8
411 RDA OUTPUT COMMANDS
412 OUT 2 1 -1 0
413 RDA DECAY YEAR 1990 TO 3.00 YRS
414 DEC 3.00 -6 6 5 1
415 RDA OUTPUT 1990 DECAY TO 1993 DATA
416 MOV 6 2 0 1
417 MOV -6 1 0 1
418 HED 1 1990
419 HED 2 1993
420 RDA PRINT NUCLIDE QUANTITIES IN CURIES
421 OPTA 6*8 5 17*8
422 OPTF 6*8 5 17*8
423 OPTL 6*8 5 17*8
424 RDA OUTPUT COMMANDS
425 OUT 2 1 -1 0
426 RDA DECAY YEAR 1991 TO 2.00 YRS
427 DEC 2.00 -7 7 5 1
428 RDA OUTPUT 1991 DECAY TO 1993 DATA
429 MOV 7 2 0 1
430 MOV -7 1 0 1

```

Sample ORIGEN2 Input File Created by PreORG

ornich93.u5

```

431 HED 1 1991
432 HED 2 1993
433 RDA PRINT NUCLIDE QUANTITIES IN CURIES
434 OPTA 6*8 5 17*8
435 OPTF 6*8 5 17*8
436 OPTL 6*8 5 17*8
437 RDA OUTPUT COMMANDS
438 OUT 2 1 -1 0
439 RDA DECAY YEAR 1992 TO 1.00 YRS
440 DEC 1.00 -8 8 5 1
441 RDA OUTPUT 1992 DECAY TO 1993 DATA
442 MOV 8 2 0 1
443 MOV -8 1 0 1
444 HED 1 1992
445 HED 2 1993
446 RDA PRINT NUCLIDE QUANTITIES IN CURIES
447 OPTA 6*8 5 17*8
448 OPTF 6*8 5 17*8
449 OPTL 6*8 5 17*8
450 RDA OUTPUT COMMANDS
451 OUT 2 1 -1 0
452 RDA DECAY YEAR 1993 TO .00 YRS
453 DEC .00 -9 9 5 1
454 RDA OUTPUT 1993 DECAY TO 1993 DATA
455 MOV 9 2 0 1
456 MOV -9 1 0 1
457 HED 1 1993
458 HED 2 1993
459 RDA PRINT NUCLIDE QUANTITIES IN CURIES
460 OPTA 6*8 5 17*8
461 OPTF 6*8 5 17*8
462 OPTL 6*8 5 17*8
463 RDA OUTPUT COMMANDS
464 OUT 2 1 -1 0
465 END
466 2 942380 .1209E+02 942390 .1516E+04 952410 .6732E+01 0 0
467 2 922330 .8958E+03 952430 .2785E+01 982520 .9431E-04 0 0
468 2 962420 .3025E-04 962440 .2349E+00 932370 .6697E+02 0 0
469 2 882260 .1012E-03 922350 .5090E+03 922380 .5950E+01 0 0
470 0
471 1 380900 .5859E-03 551370 .9191E-03 0 0
472 2 942380 .1250E-02 942390 .4232E+02 942400 .1677E+02 0 0
473 2 952410 .2434E+01 922330 .2149E+03 952430 .1004E+00 0 0
474 2 982520 .1602E-04 962420 .1513E-06 962440 .1385E-02 0 0
475 2 932370 .9748E+00 922320 .8224E-02 922380 .6456E+03 0 0
476 0
477 1 611470 .1186E-04 0 0
478 2 942380 .2179E+01 942390 .4843E+03 942400 .1734E+03 0 0
479 2 942410 .4816E+01 942420 .5109E+00 952410 .7548E-02 0 0
480 2 922330 .4588E+02 982520 .4595E-04 962440 .5712E-03 0 0
481 2 932370 .4966E+02 922320 .5608E-02 922340 .1601E-02 0 0
482 2 922380 .3600E+03 0 0
483 0
484 2 942380 .5843E-02 942390 .1706E+02 942400 .4390E-02 0 0
485 2 942410 .2719E-04 952410 .7519E-01 922330 .2294E+03 0 0
486 2 952430 .5018E+00 982490 .2442E-02 982520 .8222E-04 0 0
487 2 962440 .6132E-03 962480 .4517E+01 932370 .2838E+02 0 0
488 2 882260 .3035E-02 0 0
489 0
490 2 942380 .3506E-03 942390 .1340E+02 942400 .3516E+01 0 0
491 2 942410 .5340E-03 942420 .6025E-02 952410 .3148E+00 0 0
492 2 922330 .4123E+03 982520 .1860E-04 962440 .3338E-04 0 0
493 2 922320 .5094E-03 922340 .9604E-01 922350 .5044E+03 0 0
494 0
495 2 942380 .1186E-02 942390 .1292E+02 942410 .7078E+01 0 0
496 2 952410 .2049E+00 922330 .5993E+02 952430 .9283E-03 0 0
497 2 982520 .1382E-05 962440 .1199E+01 932370 .1986E+03 0 0
498 2 882260 .1770E-02 922360 .3092E+00 0 0
499 0
500 2 942390 .6453E+01 952410 .2069E-02 922330 .2066E+02 0 0
501 2 982490 .4884E-03 982520 .2046E-04 962420 .2118E-04 0 0
502 2 962440 .1154E-01 882260 .1608E+01 902320 .3739E+03 0 0
503 2 922380 .5058E+02 0 0
504 0
505 1 260590 .8124E-04 270600 .2650E-08 380900 .2197E-07 0 0
506 1 390900 .5508E-11 551370 .9191E-03 611470 .2156E-07 0 0

```

Terminate ORIGEN2 execution

Appendix C: ORIGEN2 Command Descriptions and Sample ORIGEN2 Input File Created by PreORG

Sample ORIGEN2 Input File Created by PreORG

ornlch93.u5

507	2	942380	.4394E+01	942390	.1673E+02	942400	.4829E+00	0	0
508	2	942410	.3262E+01	952410	.9559E+00	922330	.1436E+02	0	0
509	2	952430	.1576E-02	982490	.1465E-04	982520	.5636E-04	0	0
510	2	962440	.3932E-02	932370	.1166E+03	922340	.3954E-01	0	0
511	2	922350	.1851E+02	922360	.1036E+01	922380	.2350E+03	0	0
512	0								
513	1	270600	.1078E-04	380900	.8935E-04	551370	.2504E-03	0	0
514	1	280630	.1798E-05	0	0				
515	2	942380	.2997E+01	942390	.4908E+04	942400	.7023E+03	0	0
516	2	942410	.2767E+02	942420	.4349E+01	952410	.3410E+02	0	0
517	2	922330	.1178E+01	922340	.1365E+04	922350	.2226E+03	0	0
518	2	922360	.1237E+00	922380	.2690E+05	962440	.8148E-02	0	0
519	2	972490	.4273E-08	982490	.1048E-03	982520	.7292E-04	0	0
520	0								

Indicates data library
1 - Activation
Products
2 - Actinides
3 - Fission Products

Sample Index File Created by PreORG: sample.ind

ORIGEN2 INPUT FILE	YEARS OF INVENTORY	SITE NAME	DECAY ALL YEARS TO:
hanfch93.u5	1970 thru 1993	HANFORD	1993
	Enriched U Atom Fractions: U234=	.00 U235=	.93 U238= .01

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
hanfrh93.u5      1970 thru 1988      HANFORD      1993
                  Enriched U Atom Fractions: U234=    .00 U235=    .93 U238=    .06

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      inelch93.u5      . 1970 thru 1992      INEL      1993
                     Enriched U Atom Fractions: U234=   .00 U235=   .07 U238=   .93

```

XX						
inelrh93.u5	1977 thru 1993	INEL		1993		
	Enriched U Atom Fractions: U234=	.00	U235=	.07	U238=	.93

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ornlch93.u5      1970 thru 1993      ORNL      1993
                  Enriched U Atom Fractions: U234= .00 U235= .07 U238= .93

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ornlrh93.u5      1970 thru 1993      ORNL      1993
                  Enriched U Atom Fractions: U234=   .00 U235=   .07 U238=   .93

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      ornibu93.u5      1972 thru 1974      ORNL      1993
                        Enriched U Atom Fractions: U234= .00 U235= .07 U238= .93

```

Intentionally Left Blank

APPENDIX D: OR1_PREORG.FOR PROGRAM LISTING

Intentionally Left Blank

[illegible]

C	*****	**	INFRINGE	**	PRE0076
C		**	PRIVATELY	**	PRE0077
C		**	OWNED	**	PRE0078
C		**	RIGHTS.	**	PRE0079
C		**		**	PRE0080
C		**		**	PRE0081
C		**		**	PRE0082
C		*****			PRE0083
C					PRE0084
C					PRE0085
C	C23456789012345678901234567890123456789012345678901234567890123456789012				PRE0086
C	1 2 3 4 5 6 7				PRE0087
C	CC				PRE0088
C	THIS PROGRAM TRANSLATES DATA FROM THE WIPP BASELINE INVENTORY	C			PRE0089
C	DATABASE INTO AN ORIGEN2 INPUT FILE	C			PRE0090
C	CC				PRE0091
	IMPLICIT real(a-h,o-z), integer(i-n)				PRE0092
	REAL curies(100,500), grams(100,500),decay(50),				PRE0093
	& iu234, iu235, iu238				PRE0094
	INTEGER ycnt, isonum(0:500),var1(500), var2(500), var3(500),				PRE0095
	& lib(100,500), ymax, yr(500), out,				PRE0096
	& cnt, fyr				PRE0097
	CHARACTER input*12, outfil*12, title*133, line*133, answer*1,				PRE0098
	& nucnam(100,500)*6, nucid(100,500)*6, waste*2,				PRE0099
	& list*25, site*15, table*15, year*4				PRE0100
	LOGICAL exists				PRE0101
C					PRE0102
C	PROGRAM HEADER				PRE0103
C					PRE0104
	CALL HEADER				PRE0105
C					PRE0106
	WRITE(6,5000)				PRE0107
5000	FORMAT(10('CCCCCCC'))				PRE0108
	WRITE(6,*)				PRE0109
	WRITE(6,*) ' THIS PROGRAM TRANSLATES DATA IN THE WIPP BID INTO '				PRE0110
	WRITE(6,*) ' ORIGEN2 INPUT FILES TO DECAY INVENTORIES IN THE USER '				PRE0111
	WRITE(6,*) ' SPECIFIED SITE DATA FILE TO THE SAME YEAR AS GIVEN '				PRE0112
	WRITE(6,*) ' BY THE USER. '				PRE0113
	WRITE(6,*) ' ENTER YEAR WISH TO DECAY ALL INVENTORY DATA TO: '				PRE0114
	WRITE(6,*)				PRE0115
	WRITE(6,5000)				PRE0116
	READ 2000, YEAR				PRE0117
2000	FORMAT(A4)				PRE0118
	READ(YEAR,2005) FYR				PRE0119
2005	FORMAT(I4)				PRE0120
	WRITE(6,5000)				PRE0121
C					PRE0122
	IUNIT = 0				PRE0123
	IF(IUNIT.NE.1)THEN				PRE0124
	IUNIT = 1				PRE0125
	WRITE(6,5000)				PRE0126
	WRITE(6,*)				PRE0127
	WRITE(6,*) 'FOR YOUR REFERENCE PREORG.FOR WILL GENERATE A LIST '				PRE0128
	WRITE(6,*) 'OF THE ORIGEN2 INPUT FILES CREATED DURING THIS RUN'				PRE0129
	WRITE(6,*)				PRE0130
	WRITE(6,*) ' PLEASE ENTER THE NAME OF THE LIST FILE '				PRE0131
	WRITE(6,*) ' 6 CHARACTERS OR LESS: '				PRE0132
	WRITE(6,*)				PRE0133
	WRITE(6,5000)				PRE0134
	READ 2010, TABLE				PRE0135
2010	FORMAT(A6)				PRE0136
	LIST = TABLE//'.ind'				PRE0137
	OPEN(UNIT=50,FILE=LIST,STATUS='NEW')				PRE0138
	WRITE(50,5005) LIST				PRE0139
5005	FORMAT(' INDEX FILENAME: ',A10)				PRE0140
	WRITE(50,5010)				PRE0141
5010	FORMAT(T2,'ORIGEN2 INPUT FILE',T24,'YEARS OF INVENTORY',T45				PRE0142
	& 'SITE NAME',3X,' DECAY ALL YEARS TO: ')				PRE0143
	WRITE(50,*)				PRE0144
	ENDIF				PRE0145
C					PRE0146
C	PROMPT USER FOR SITE NAME AND TYPE OF WASTE				PRE0147
C					PRE0148
1000	DO 10 I= 1,100				PRE0149
	YR(I) = 0				PRE0150
	ISONUM(I) = 0				PRE0151

```

10    CONTINUE
C
    NOFILE = 0
    WRITE(6,*) ' PLEASE CHOOSE A SITE NAME: '
1005  WRITE(6,*) ' 1 - ARCO MEDICAL PRODUCTS (ARCOMP) '
    WRITE(6,*) ' 2 - ARGONNE NATIONAL LABORATORY '
    WRITE(6,*) ' 3 - ETEC '
    WRITE(6,*) ' 4 - HANFORD '
    WRITE(6,*) ' 5 - INEL '
    WRITE(6,*) ' 6 - KNOLLS APL '
    WRITE(6,*) ' 7 - LB LABORATORY '
    WRITE(6,*) ' 8 - LOS ALAMOS NATIONAL LABORATORY (LANL) '
    WRITE(6,*) ' 9 - LLNL '
    WRITE(6,*) ' 10 - MOUND PLANT '
    WRITE(6,*) ' 11 - MURR - DOE '
    WRITE(6,*) ' 12 - NEVADA TEST SITE (NTS) '
    WRITE(6,*) ' 13 - OAK RIDGE NATIONAL LABORATORY '
    WRITE(6,*) ' 14 - PADUCAH '
    WRITE(6,*) ' 15 - PANTEX PLANT '
    WRITE(6,*) ' 16 - ROCKY FLATS '
    WRITE(6,*) ' 17 - SAVANNAH RIVER '
    WRITE(6,*) ' 18 - SANDIA NATIONAL LABORATORY - NM '
    WRITE(6,*) ' 19 - US ARMY MATERIAL COMMAND (ARMYMC) '
    WRITE(6,*) ' 20 - WEST VALLEY DEMO '
    WRITE(6,*) ' 21 - OTHER SITE '
    WRITE(6,*)
    READ*, isite
C
C  ASSIGN INPUT AND OUTPUT FILE NAMES
C
    IF(isite.EQ.1)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '
        WRITE(6,*) ' ARCO MEDICAL PRODUCTS (ARCOMP) '
        site = 'arcomp_'
    ELSEIF(isite.EQ.2)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '
        WRITE(6,*) ' ARGONNE NATIONAL LABORATORY '
        site = 'argonne'
    ELSEIF(isite.EQ.3)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '
        WRITE(6,*) ' ETEC '
        site = 'etec_sl'
    ELSEIF(isite.EQ.4)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '
        WRITE(6,*) ' HANFORD '
        site = 'hanford'
    ELSEIF(isite.EQ.5)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '
        WRITE(6,*) ' INEL '
        site = 'inel__'
    ELSEIF(isite.EQ.6)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '
        WRITE(6,*) ' KNOLLS APL '
        site = 'knolls_'
    ELSEIF(isite.EQ.7)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '
        WRITE(6,*) ' LB LABORATORY '
        site = 'l_b_lab'
    ELSEIF(isite.EQ.8)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '
        WRITE(6,*) ' LANL '
        site = 'lanl__'
    ELSEIF(isite.EQ.9)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '
        WRITE(6,*) ' LLNL '
        site = 'llnl__'
    ELSEIF(isite.EQ.10)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '
        WRITE(6,*) ' MOUND PLANT '
        site = 'mound__'
    ELSEIF(isite.EQ.11)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '
        WRITE(6,*) ' MURR-DOE '
        site = 'murr__'
    ELSEIF(isite.EQ.12)THEN
        WRITE(6,*) ' GENERATING ORIGEN2 INPUT '

```

Appendix D: OR1_PreORG.for Program Listing

```

WRITE(6,*)' NEVADA TEST SITE ' PRE0228
site = 'nts' PRE0229
ELSEIF(isite.EQ.13)THEN PRE0230
WRITE(6,*)' GENERATING ORIGEN2 INPUT ' PRE0231
WRITE(6,*)' ORNL ' PRE0232
site = 'ornl' PRE0233
ELSEIF(isite.EQ.14)THEN PRE0234
WRITE(6,*)' GENERATING ORIGEN2 INPUT ' PRE0235
WRITE(6,*)' PADUCAH ' PRE0236
site = 'paducah' PRE0237
ELSEIF(isite.EQ.15)THEN PRE0238
WRITE(6,*)' GENERATING ORIGEN2 INPUT ' PRE0239
WRITE(6,*)' PANTEX ' PRE0240
site = 'pantex_' PRE0241
ELSEIF(isite.EQ.16)THEN PRE0242
write(6,*)' GENERATING ORIGEN2 INPUT ' PRE0243
write(6,*)' ROCKY FLATS ' PRE0244
site = 'rocky_f' PRE0245
ELSEIF(isite.EQ.17)THEN PRE0246
WRITE(6,*)' GENERATING ORIGEN2 INPUT ' PRE0247
WRITE(6,*)' SAVANNAH RIVER SITE ' PRE0248
input = 'savanna' PRE0249
site = 'savanna' PRE0250
ELSEIF(isite.EQ.18)THEN PRE0251
WRITE(6,*)' GENERATING ORIGEN2 INPUT: ' PRE0252
WRITE(6,*)' SNL/NM' PRE0253
site = 'snl_nm' PRE0254
ELSEIF(isite.EQ.19)THEN PRE0255
WRITE(6,*)' GENERATING ORIGEN2 INPUT: ' PRE0256
WRITE(6,*)' US ARMY MATERIAL COMMAND (ARMYMC) ' PRE0257
site = 'armymc_' PRE0258
ELSEIF(isite.EQ.20)THEN PRE0259
WRITE(6,*)' GENERATING ORIGEN2 INPUT : ' PRE0260
WRITE(6,*)' WEST VALLEY ' PRE0261
site = 'w_val_d' PRE0262
ELSEIF(isite.EQ.21)THEN PRE0263
CALL OTHER(site) PRE0264
ELSE PRE0265
WRITE(6,*)'NO SUCH SITE IN WIPP BASELINE INVENTORY DATABASE ' PRE0266
GO TO 9999 PRE0267
ENDIF PRE0268
C PRE0269
WRITE(6,*) PRE0270
CALL TYPE(itype, waste) PRE0271
CALL INNAME(site,itype,input) PRE0272
C PRE0273
C INQUIRE TO SEE IF DATA FILE EXISTS PRE0274
C PRE0275
INQUIRE(FILE=input, EXIST=exists) PRE0276
IF(.NOT.exists)THEN PRE0277
C PRE0278
C ERROR MESSAGE - INPUT FILE NOT FOUND PRE0279
C PRE0280
WRITE(6,5015)input, waste, site PRE0281
5015 FORMAT(/,7('**ERROR'),/,/,t10,' INPUT FILE ',a12, PRE0282
& ' DOES NOT EXIST ',/,/, ' WASTE TYPE: ', a2, ' FOR THE ',a7, PRE0283
& ' SITE IS NOT VALID ',/,/, 7('**ERROR'),/,/) PRE0284
WRITE(6,*)' TO CHOOSE ANOTHER SITE ENTER [1] OR ' PRE0285
WRITE(6,*)' TO EXIT PreORG ENTER [2]: ' PRE0286
WRITE(6,*) PRE0287
READ *, ians PRE0288
IF(ians.EQ.1)THEN PRE0289
WRITE(6,*)' CHOOSE ANOTHER SITE: ' PRE0290
GO TO 1005 PRE0291
ELSEIF(ians.EQ.2)THEN PRE0292
GO TO 9999 PRE0293
ENDIF PRE0294
ENDIF PRE0295
C PRE0296
CALL NUMYRS(input,noyr,nofile) PRE0297
CALL OUTNAM(year,input,outfil) PRE0298
C PRE0299
C ENTER ENRICHED URANIUM BREAKDOWN PRE0300
C PRE0301
iu234 = 0.00 PRE0302
iu235 = 0.07 PRE0303

```

```

        iu238 = 0.93
        WRITE(6,*)
        WRITE(6,*) ' DEFAULT ENRICHED URANIUM BREAKDOWN IS: '
        WRITE(6,*) '          U234 - 0.00 '
        WRITE(6,*) '          U235 - 0.07 '
        WRITE(6,*) '          U238 - 0.93 '
        WRITE(6,5020) site
5020  FORMAT(2x, ' DO YOU WANT A NEW ENRICHED URANIUM BREAKDOWN FOR ',
        &      a15, ' ? ' )
        WRITE(6,*) ' Enter y (yes) or n (no): '
        READ 5025, ANSWER
5025  FORMAT(A1)
        WRITE(6,*)
        WRITE(6,5000)
        IF(answer.EQ.'y'.OR.answer.EQ.'Y')THEN
            WRITE(6,*) ' PLEASE ENTER ENRICHED URANIUM BREAKDOWN AS '
            WRITE(6,*) ' ATOM FRACTIONS: '
            WRITE(6,*) '          U234 = '
            READ*, iu234
            WRITE(6,*) '          U235 = '
            READ*, iu235
            WRITE(6,*) '          U238 = '
            READ*, iu238
            WRITE(6,*)
        ENDIF
        WRITE(6,5000)
        WRITE(6,*)

C
C  OPEN INPUT AND OUTPUT FILES
C
        OPEN(UNIT=10,FILE=INPUT,STATUS='OLD')
        OPEN(UNIT=15,FILE=OUTFIL,STATUS='NEW')
C
C  READ INPUT DATA FROM FILE
C
1015  ycnt = 0
        icnt = 1
C
1020  READ(10,2020,END=8000) line
2020  FORMAT(A133)
        write(6,*) line
C
        IF(icnt.EQ.1)THEN
            title = line
            icnt = 2
            GO TO 1020
        ENDIF
C
        IF(line(1:2).EQ.'19'.OR.line(1:2).EQ.'20')THEN
            ycnt = ycnt + 1
            isonum(ycnt) = 0
            READ(LINE,2025) yr(ycnt)
2025  FORMAT(I4)
            icnt = icnt + 1
            GO TO 1020
        ENDIF
        IF(line(3:7).EQ.'ZZZZZ')GO TO 1020
C
C  READ ISOTOPES AND ACTIVITIES IN INVENTORY FOR PRESENT YEAR AND SITE
C
        isonum(ycnt) = isonum(ycnt) + 1
        READ(LINE,2030) nucnam(ycnt,isonum(ycnt)),
        &      curies(ycnt,isonum(ycnt))
2030  FORMAT(T3,A6,T41,E10.4)
        write(6,2030) nucnam(ycnt,isonum(ycnt)),
        &      curies(ycnt,isonum(ycnt))
C
C  DEPLETED URANIUM - ASSUME 100% U-238
C
        IF(nucnam(ycnt,isonum(ycnt))(1:4).EQ.'UDPL')THEN
            nucnam(ycnt,isonum(ycnt)) = 'U238'
            GO TO 1020
C
C  NATURAL URANIUM - ATOM% FROM CHART OF NUCLIDES
C
        ELSEIF(nucnam(ycnt,isonum(ycnt))(1:4).EQ.'UNAT')THEN

```

Appendix D: OR1_PreORG.for Program Listing

```

nucnam(ycnt,isonum(ycnt)) = 'U234'
tcurie = curies(ycnt,isonum(ycnt))
curies(ycnt,isonum(ycnt)) = 0.000055*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'U235'
curies(ycnt,isonum(ycnt)) = 0.00720*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'U238'
curies(ycnt,isonum(ycnt)) = 0.992745*tcurie
GO TO 1020

C
C ENRICHED URANIUM - 93% U-235 AND 7% U-238
C
ELSEIF(nucnam(ycnt,isonum(ycnt))(1:4).EQ.'UENR') THEN
tcurie = curies(ycnt,isonum(ycnt))
nucnam(ycnt,isonum(ycnt)) = 'U235'
curies(ycnt,isonum(ycnt)) = iu235*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'U238'
curies(ycnt,isonum(ycnt)) = iu238*tcurie
IF(iu234.NE.0) THEN
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'U238'
curies(ycnt,isonum(ycnt)) = iu238*tcurie
ENDIF
GO TO 1020

C
C FOR UNKNOWN NUCLIDES IGNORE ACTIVITIES IN CURIES
C
ELSEIF(nucnam(ycnt,isonum(ycnt))(1:4).EQ.'UNKN') THEN
isonum(ycnt) = isonum(ycnt) - 1
GO TO 1020

C
C FOR PO209 - NO DATA IN ORIGEN2 LIBRARIES
C
ELSEIF(nucnam(ycnt,isonum(ycnt))(1:5).EQ.'PO209') THEN
isonum(ycnt) = isonum(ycnt) - 1
GO TO 1020

C
C 7/26/95 FOR MIXED ACTIVATION PRODUCTS WHOSE ISOTOPIC COMPOSITION
C IS NOT GIVEN BREAKDOWN PROVIDED BY SAYAN CHAKRABORTI, SAIC IS USED
C TO DETERMINE THE ISOTOPIC COMPOSITIONS (FROM HANFORD)
C
ELSEIF(nucnam(ycnt,isonum(ycnt))(1:6).EQ.'TBGMAP') THEN
tcurie = curies(ycnt,isonum(ycnt))
nucnam(ycnt,isonum(ycnt)) = 'BA137M'
curies(ycnt,isonum(ycnt)) = 0.0097*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'C14'
curies(ycnt,isonum(ycnt)) = 0.01*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'CO58'
curies(ycnt,isonum(ycnt)) = 0.01*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'CO60'
curies(ycnt,isonum(ycnt)) = 0.67*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'CS137'
curies(ycnt,isonum(ycnt)) = 0.021241*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'EU152'
curies(ycnt,isonum(ycnt)) = 0.01*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'EU154'
curies(ycnt,isonum(ycnt)) = 0.085*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'EU155'
curies(ycnt,isonum(ycnt)) = 0.085*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'FE59'
curies(ycnt,isonum(ycnt)) = 0.04*tcurie
isonum(ycnt) = isonum(ycnt) + 1
nucnam(ycnt,isonum(ycnt)) = 'MN54'
curies(ycnt,isonum(ycnt)) = 0.07*tcurie
GO TO 1020

C

```

Appendix D: OR1_PreORG.for Program Listing

```

C 7/26/95 FOR MIXED ACTIVATION PRODUCTS WHOSE ISOTOPIC COMPOSITION      PRE0456
C IS NOT GIVEN BREAKDOWN PROVIDED BY SAYAN CHAKRABORTI, SAIC IS USED    PRE0457
C TO DETERMINE THE ISOTOPIC COMPOSITIONS (FROM HANFORD)                PRE0458
C                                                                        PRE0459
C      ELSEIF(nucnam(ycnt,isonum(ycnt))(1:6).EQ.'TBGMFP') THEN          PRE0460
C          tcurie = curies(ycnt,isonum(ycnt))                          PRE0461
C          nucnam(ycnt,isonum(ycnt)) = 'AG110M'                        PRE0462
C          curies(ycnt,isonum(ycnt)) = 5.02E-6*tcurie                  PRE0463
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0464
C          nucnam(ycnt,isonum(ycnt)) = 'AG110'                          PRE0465
C          curies(ycnt,isonum(ycnt)) = 2.507E-8*tcurie                  PRE0466
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0467
C          nucnam(ycnt,isonum(ycnt)) = 'BA137M'                         PRE0468
C          curies(ycnt,isonum(ycnt)) = 0.01848*tcurie                   PRE0469
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0470
C          nucnam(ycnt,isonum(ycnt)) = 'CD113M'                         PRE0471
C          curies(ycnt,isonum(ycnt)) = 5.0E-10*tcurie                   PRE0472
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0473
C          nucnam(ycnt,isonum(ycnt)) = 'CE144'                         PRE0474
C          curies(ycnt,isonum(ycnt)) = 0.2784*tcurie                    PRE0475
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0476
C          nucnam(ycnt,isonum(ycnt)) = 'CS134'                         PRE0477
C          curies(ycnt,isonum(ycnt)) = 5.557E-4*tcurie                  PRE0478
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0479
C          nucnam(ycnt,isonum(ycnt)) = 'CS135'                         PRE0480
C          curies(ycnt,isonum(ycnt)) = 5.750E-8*tcurie                  PRE0481
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0482
C          nucnam(ycnt,isonum(ycnt)) = 'CS137'                         PRE0483
C          curies(ycnt,isonum(ycnt)) = 2.009E-2*tcurie                  PRE0484
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0485
C          nucnam(ycnt,isonum(ycnt)) = 'EU152'                         PRE0486
C          curies(ycnt,isonum(ycnt)) = 3.0E-7*tcurie                    PRE0487
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0488
C          nucnam(ycnt,isonum(ycnt)) = 'EU154'                         PRE0489
C          curies(ycnt,isonum(ycnt)) = 3.038E-5*tcurie                  PRE0490
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0491
C          nucnam(ycnt,isonum(ycnt)) = 'EU155'                         PRE0492
C          curies(ycnt,isonum(ycnt)) = 7.353E-4*tcurie                  PRE0493
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0494
C          nucnam(ycnt,isonum(ycnt)) = 'GD162'                         PRE0495
C          curies(ycnt,isonum(ycnt)) = 9.862E-6*tcurie                  PRE0496
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0497
C          nucnam(ycnt,isonum(ycnt)) = 'NB95'                           PRE0498
C          curies(ycnt,isonum(ycnt)) = 1.053E-1*tcurie                  PRE0499
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0500
C          nucnam(ycnt,isonum(ycnt)) = 'NB95M'                         PRE0501
C          curies(ycnt,isonum(ycnt)) = 1.059E-1*tcurie                  PRE0502
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0503
C          nucnam(ycnt,isonum(ycnt)) = 'PD107'                         PRE0504
C          curies(ycnt,isonum(ycnt)) = 8.5E-9*tcurie                    PRE0505
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0506
C          nucnam(ycnt,isonum(ycnt)) = 'PM147'                         PRE0507
C          curies(ycnt,isonum(ycnt)) = 7.022E-2*tcurie                  PRE0508
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0509
C          nucnam(ycnt,isonum(ycnt)) = 'PR144'                         PRE0510
C          curies(ycnt,isonum(ycnt)) = 2.784E-1*tcurie                  PRE0511
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0512
C          nucnam(ycnt,isonum(ycnt)) = 'RH106'                         PRE0513
C          curies(ycnt,isonum(ycnt)) = 4.051E-2*tcurie                  PRE0514
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0515
C          nucnam(ycnt,isonum(ycnt)) = 'RU106'                         PRE0516
C          curies(ycnt,isonum(ycnt)) = 4.051E-2*tcurie                  PRE0517
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0518
C          nucnam(ycnt,isonum(ycnt)) = 'SB125'                         PRE0519
C          curies(ycnt,isonum(ycnt)) = 7.985E-4*tcurie                  PRE0520
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0521
C          nucnam(ycnt,isonum(ycnt)) = 'SB126'                         PRE0522
C          curies(ycnt,isonum(ycnt)) = 1.094E-7*tcurie                  PRE0523
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0524
C          nucnam(ycnt,isonum(ycnt)) = 'SB126M'                       PRE0525
C          curies(ycnt,isonum(ycnt)) = 1.105E-7*tcurie                  PRE0526
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0527
C          nucnam(ycnt,isonum(ycnt)) = 'SE79'                           PRE0528
C          curies(ycnt,isonum(ycnt)) = 4.99E-8*tcurie                  PRE0529
C          isonum(ycnt) = isonum(ycnt) + 1                             PRE0530
C          nucnam(ycnt,isonum(ycnt)) = 'SM151'                         PRE0531

```

```

curies(ycnt,isonum(ycnt)) = 1.938E-4*tcurie      PRE0532
isonum(ycnt) = isonum(ycnt) + 1                PRE0533
nucnam(ycnt,isonum(ycnt)) = 'SN119M'           PRE0534
curies(ycnt,isonum(ycnt)) = 4.372E-5*tcurie     PRE0535
isonum(ycnt) = isonum(ycnt) + 1                PRE0536
nucnam(ycnt,isonum(ycnt)) = 'SN121M'           PRE0537
curies(ycnt,isonum(ycnt)) = 3.942E-6*tcurie     PRE0538
isonum(ycnt) = isonum(ycnt) + 1                PRE0539
nucnam(ycnt,isonum(ycnt)) = 'SN123M'           PRE0540
curies(ycnt,isonum(ycnt)) = 1.847E-5*tcurie     PRE0541
isonum(ycnt) = isonum(ycnt) + 1                PRE0542
nucnam(ycnt,isonum(ycnt)) = 'SN126'            PRE0543
curies(ycnt,isonum(ycnt)) = 1.105E-7*tcurie     PRE0544
isonum(ycnt) = isonum(ycnt) + 1                PRE0545
nucnam(ycnt,isonum(ycnt)) = 'SR90'              PRE0546
curies(ycnt,isonum(ycnt)) = 1.881E-2*tcurie     PRE0547
isonum(ycnt) = isonum(ycnt) + 1                PRE0548
nucnam(ycnt,isonum(ycnt)) = 'TB162'            PRE0549
curies(ycnt,isonum(ycnt)) = 9.862E-6*tcurie     PRE0550
isonum(ycnt) = isonum(ycnt) + 1                PRE0551
nucnam(ycnt,isonum(ycnt)) = 'TC99'             PRE0552
curies(ycnt,isonum(ycnt)) = 2.867E-6*tcurie     PRE0553
isonum(ycnt) = isonum(ycnt) + 1                PRE0554
nucnam(ycnt,isonum(ycnt)) = 'TE125M'          PRE0555
curies(ycnt,isonum(ycnt)) = 2.178E-4*tcurie     PRE0556
isonum(ycnt) = isonum(ycnt) + 1                PRE0557
nucnam(ycnt,isonum(ycnt)) = 'TE127'            PRE0558
curies(ycnt,isonum(ycnt)) = 1.354E-3*tcurie     PRE0559
isonum(ycnt) = isonum(ycnt) + 1                PRE0560
nucnam(ycnt,isonum(ycnt)) = 'TE127M'          PRE0561
curies(ycnt,isonum(ycnt)) = 1.369E-3*tcurie     PRE0562
isonum(ycnt) = isonum(ycnt) + 1                PRE0563
nucnam(ycnt,isonum(ycnt)) = 'Y90'              PRE0564
curies(ycnt,isonum(ycnt)) = 1.881E-2*tcurie     PRE0565
isonum(ycnt) = isonum(ycnt) + 1                PRE0566
nucnam(ycnt,isonum(ycnt)) = 'ZR93'             PRE0567
curies(ycnt,isonum(ycnt)) = 6.455E-7*tcurie     PRE0568
isonum(ycnt) = isonum(ycnt) + 1                PRE0569
nucnam(ycnt,isonum(ycnt)) = 'ZR95'             PRE0570
curies(ycnt,isonum(ycnt)) = 4.988E-2*tcurie     PRE0571
isonum(ycnt) = isonum(ycnt) + 1                PRE0572
GO TO 1020                                       PRE0573
ENDIF                                           PRE0574
GO TO 1020                                       PRE0575
C                                               PRE0576
C GIVE USER OPTION TO PRINT OUT INTERMEDIATE VALUES PRE0577
C                                               PRE0578
8000 WRITE(6,*) ' INVENTORY DATA FOR FOR THE FOLLOWING YEARS: ' PRE0579
DO 15 i=0,ycnt-1, 5                           PRE0580
    WRITE(6,*) (yr(j), j=i+1,i+5)              PRE0581
15 CONTINUE                                     PRE0582
C                                               PRE0583
    WRITE(50,5030) outfil, yr(1), yr(noyr), site, year PRE0584
5030 FORMAT(T5,A12,T24,I4,' THRU ',I4,T45,A15,10X,A4) PRE0585
    WRITE(50,5035) iu234, iu235, iu238          PRE0586
5035 FORMAT(t24,'Enriched U Atom Fractions: U234= ',f5.2,' U235= ', PRE0587
&      f5.2,' U238= ', f5.2,/)                PRE0588
    WRITE(50,5000)                              PRE0589
    WRITE(50,*)                                  PRE0590
C                                               PRE0591
    out = 15                                     PRE0592
DO 60 cnt=1,nofile                             PRE0593
C                                               PRE0594
C WRITE TO ORIGEN2 PC INPUT FILE                PRE0595
C                                               PRE0596
    WRITE(OUT,5040)                              PRE0597
5040 FORMAT(2(2X,'-1',/),2X,'-1')              PRE0598
    WRITE(OUT,5060)                              PRE0599
5060 FORMAT(2x,'BAS',4x,'WIPP BASELINE INVENTORY DATA') PRE0600
    WRITE(OUT,5065) title                        PRE0601
5065 FORMAT(2x,'TIT',4x,a80)                    PRE0602
    WRITE(OUT,5070) site                        PRE0603
5070 FORMAT(2x,'RDA',4x,a15)                    PRE0604
    WRITE(OUT,5075)                              PRE0605
5075 FORMAT(2X,'LIP',4x,3('0',2x))              PRE0606
    WRITE(OUT,5080)                              PRE0607

```

```

5080  FORMAT(2X,'LIB',3X,'0',2X,'1',2X,'2',2X,'3',2X,3('0',2X),
      +      '9',2X,'3',2X,'0',2X,'1',2X,'1')
      WRITE(OUT,5085)
5085  FORMAT(2X,'PHO',4X,'101',2X,'102',2X,'103',2X,'10')
C
C  WRITE COMMANDS TO READ INPUT IN ORIGEN2 INPUT FILE
C
      IF(nofile.EQ.1)THEN
        ymax = ycnt
        init = 1
      ELSEIF(cnt.NE.nofile)THEN
        init = ymax + 1
        ymax = init + 14
      ELSEIF(cnt.EQ.nofile)THEN
        init = ymax + 1
        ymax = ycnt
      ENDIF
C
      DO 20 i = init, ymax
        WRITE(OUT,5090) yr(i)
5090  FORMAT(2X,'RDA',4X,'INITIAL VALUES FOR: ',i4)
        j = -(i-(cnt-1)*15)
        WRITE(OUT,5095) j
5095  FORMAT(2X,'INP',4X,i4,2X,'1',2X,2('1',2X),2('1',2X))
      20  CONTINUE
C
C  WRITE ORIGEN2 COMMANDS TO DECAY ALL YEARS IN DATABASE
C  FILE OUT TO YEAR GIVEN BY USER, INSTRUCT ORIGEN2 TO OUTPUT
C  EACH YEAR AS CALCULATION IS PERFORMED IF USER SETS IOUT=1
C  AND ADD TO A VECTOR TO HOLD ALL OF YEAR GIVEN BY USER DECAY DATA
C
      WRITE(OUT,5100) fyr, input(1:8)
5100  FORMAT(2X,'BAS',4X,'DECAY INITIAL INVENTORIES TO: ',i4,', ',A8)
C
      DO 25 i = init, ymax
        j = -(i-(cnt-1)*15)
        decay(i) = dble(fyr - yr(i))
        WRITE(OUT,5105) yr(i), decay(i)
5105  &  FORMAT(2X,'RDA',4X,'DECAY YEAR ', i4,2X,'TO ',
      &      f7.2,' YRS')
        ivec = abs(j)
        WRITE(OUT,5110) decay(i), j, ivec
5110  FORMAT(2X,'DEC',4X,f7.2,2X,i3,2X,i3,2X,'5',2X,'1')
        WRITE(OUT,5115) yr(i), fyr
5115  FORMAT(2X,'RDA',4X,'OUTPUT ',i4,' DECAY TO ',i4,' DATA ')
        WRITE(OUT,5120) ivec
5120  FORMAT(2X,'MOV',4X,i3,2X,'2',2X,'0',2X,'1')
        WRITE(OUT,5125) j
5125  FORMAT(2X,'MOV',4X,i3,2X,'1',2X,'0',2X,'1')
        WRITE(OUT,5130) yr(i)
5130  FORMAT(2X,'HED',4X,'1',2X,i4)
        WRITE(OUT,5135) fyr
5135  FORMAT(2X,'HED',4X,'2',2X,i4)
        WRITE(OUT,5140)
5140  FORMAT(2X,'RDA',4X,'PRINT NUCLIDE QUANTITIES IN CURIES')
        WRITE(OUT,5145)
5145  FORMAT(2X,'OPTA',3X,'6*8',2X,'5',2X,'17*8')
        WRITE(OUT,5150)
5150  FORMAT(2X,'OPTF',3X,'6*8',2X,'5',2X,'17*8')
        WRITE(OUT,5155)
5155  FORMAT(2X,'OPTL',3X,'6*8',2X,'5',2X,'17*8')
        WRITE(OUT,5160)
5160  FORMAT(2X,'RDA',4X,'OUTPUT COMMANDS')
        WRITE(OUT,5165)
5165  FORMAT(2X,'OUT',4X,'2',2X,'1',2X,'-1',2X,'0')
C
      25  CONTINUE
C
C
      IF(cnt.EQ.nofile)THEN
        WRITE(OUT,5170)
5170  FORMAT(2X,'END')
      ELSEIF(cnt.NE.nofile)THEN
        WRITE(OUT,5175)
5175  FORMAT(2X,'STP',4X,'1')
      ENDIF

```

Appendix D: OR1_PreORG.for Program Listing

C		PRE0684
C	WRITE INITIAL NUCLIDE INVENTORIES TO ORIGEN2 INPUT FILE	PRE0685
C		PRE0686
	DO 55 i = init, ymax	PRE0687
	num = isonum(i)	PRE0688
	lib1 = 0	PRE0689
	lib2 = 0	PRE0690
	lib3 = 0	PRE0691
	lcnt = 0	PRE0692
C		PRE0693
C	CONVERT INITIAL NUCLIDE CONCENTRATIONS FROM CURIES TO GRAMS AND	PRE0694
C	ASSIGN ORIGEN2 SPECIFIC NUCLIDE IDENTIFIER	PRE0695
C		PRE0696
	DO 30 j = 1, num	PRE0697
	CALL CONV(nucnam(i,j),curies(i,j),nucid(i,j),	PRE0698
	& grams(i,j), lib(i,j))	PRE0699
30	CONTINUE	PRE0700
C		PRE0701
	DO 35 k = 1, num	PRE0702
	IF(lib(i,k).EQ.1)THEN	PRE0703
	lib1 = lib1 + 1	PRE0704
	var1(lib1) = k	PRE0705
	ELSEIF(lib(i,k).EQ.2)THEN	PRE0706
	lib2 = lib2 + 1	PRE0707
	var2(lib2) = k	PRE0708
	ELSEIF(lib(i,k).EQ.3)THEN	PRE0709
	lib3 = lib3 + 1	PRE0710
	var3(lib3) = k	PRE0711
	ENDIF	PRE0712
35	CONTINUE	PRE0713
C		PRE0714
C	FOR PRESENT YEAR, FROM LIBRARY 1 - ACTIVATION PRODUCTS	PRE0715
C		PRE0716
	IF(lib1.LE.3)THEN	PRE0717
	num = lib1	PRE0718
	IF(lib1.EQ.1)THEN	PRE0719
	WRITE(OUT,5185) (nucid(i,var1(L)),grams(i,var1(L)),L=1,lib1)	PRE0720
5185	FORMAT(2x,'1',5x,A7,2x,e10.4,2x,'0',2x,'0')	PRE0721
	ELSEIF(lib1.EQ.2)THEN	PRE0722
	WRITE(OUT,5190) (nucid(i,var1(L)),grams(i,var1(L)), L=1,2)	PRE0723
5190	FORMAT(2x,'1',5x,2(a7,2x,e10.4,2x),'0',2x,'0')	PRE0724
	ELSEIF(lib1.EQ.3)THEN	PRE0725
	WRITE(OUT,5195) (nucid(i,var1(L)),grams(i,var1(L)), L=1,3)	PRE0726
5195	FORMAT(2x,'1',5x,3(a7,2x,e10.4,2x),'0',2x,'0')	PRE0727
	ENDIF	PRE0728
	ELSEIF(lib1.GT.3)THEN	PRE0729
	ldiv1 = int(lib1/3)	PRE0730
	ldiv2 = lib1 - 3*ldiv1	PRE0731
	m = 0	PRE0732
	IF(ldiv2.EQ.0)THEN	PRE0733
1025	m = m + 3	PRE0734
	IF(m.LE.lib1)THEN	PRE0735
	n = m-2	PRE0736
	WRITE(OUT,5200) (nucid(i,var1(L)),grams(i,var1(L)), L=n,m)	PRE0737
5200	FORMAT(2x,'1',5x,3(a7,2x,e10.4,2x),'0',2x,'0')	PRE0738
	GO TO 1025	PRE0739
	ENDIF	PRE0740
	ELSEIF(ldiv2.GT.0)THEN	PRE0741
	IF(ldiv1.EQ.1)THEN	PRE0742
	WRITE(OUT,5200) (nucid(i,var1(L)),grams(i,var1(L)), L=1,3)	PRE0743
	IF(ldiv2.EQ.1)THEN	PRE0744
	WRITE(OUT,5185) (nucid(i,var1(L)),grams(i,var1(L)),L = 4, lib1)	PRE0745
	ELSEIF(ldiv2.EQ.2)THEN	PRE0746
	WRITE(OUT,5190) (nucid(i,var1(L)),grams(i,var1(L)),L=4,lib1)	PRE0747
	ENDIF	PRE0748
	ELSE	PRE0749
	DO 40 k = 1, ldiv1	PRE0750
1030	m = m + 3	PRE0751
	lcnt = lcnt + 1	PRE0752
	IF(lcnt.LE.ldiv1)THEN	PRE0753
	n = m-2	PRE0754
	WRITE(OUT,5200) (nucid(i,var1(L)),grams(i,var1(L)),L=n,m)	PRE0755
	GO TO 1030	PRE0756
	ELSEIF(lcnt.GT.ldiv1)THEN	PRE0757
	m = m-3	PRE0758
	ENDIF	PRE0759

```

40          CONTINUE
           m = m+1
           IF(ldiv2.EQ.1)THEN
WRITE(OUT,5185) (nucid(i,var1(L)),grams(i,var1(L)),L=m,lib1)
           ELSEIF(ldiv2.EQ.2)THEN
WRITE(OUT,5190) (nucid(i,var1(L)),grams(i,var1(L)),L=m,lib1)
           ENDIF
           ENDIF
           ENDIF
           ENDIF
C
C   FOR PRESENT YEAR, FROM LIBRARY 2 - ACTINIDES
C
           lcnt = 0
           IF(lib2.LE.3)THEN
           num = lib2
           IF(lib2.EQ.1)THEN
WRITE(OUT,5205) (nucid(i,var2(L)),grams(i,var2(L)),L=1,lib2)
5205  FORMAT(2x,'2',5x,A7,2x,e10.4,2x,'0',2x,'0')
           ELSEIF(lib2.EQ.2)THEN
WRITE(OUT,5210) (nucid(i,var2(L)),grams(i,var2(L)), L=1,2)
5210  FORMAT(2x,'2',5x,2(a7,2x,e10.4,2x),'0',2x,'0')
           ELSEIF(lib2.EQ.3)THEN
WRITE(OUT,5215) (nucid(i,var2(L)),grams(i,var2(L)), L=1,3)
5215  FORMAT(2x,'2',5x,3(a7,2x,e10.4,2x),'0',2x,'0')
           ENDIF
           ELSEIF(lib2.GT.3)THEN
           ldiv1 = int(lib2/3)
           ldiv2 = lib2 - 3*ldiv1
           m = 0
           IF(ldiv2.EQ.0)THEN
           m = m + 3
           IF(m.LE.lib2)THEN
           n = m-2
WRITE(OUT,5220) (nucid(i,var2(L)),grams(i,var2(L)), L=n,m)
5220  FORMAT(2x,'2',5x,3(a7,2x,e10.4,2x),'0',2x,'0')
           GO TO 1035
           ENDIF
           ELSEIF(ldiv2.GT.0)THEN
           IF(ldiv1.EQ.1)THEN
WRITE(OUT,5220) (nucid(i,var2(L)),grams(i,var2(L)), L=1,3)
           IF(ldiv2.EQ.1)THEN
WRITE(OUT,5205) (nucid(i,var2(L)),grams(i,var2(L)),L=4,lib2)
           ELSEIF(ldiv2.EQ.2)THEN
WRITE(OUT,5210) (nucid(i,var2(L)),grams(i,var2(L)),L=4,lib2)
           ENDIF
           ELSE
           DO 45 k = 1, ldiv1
           m = m + 3
           lcnt = lcnt + 1
           IF(lcnt.LE.ldiv1)THEN
           n = m-2
WRITE(OUT,5220) (nucid(i,var2(L)),grams(i,var2(L)),L=n,m)
           GO TO 1040
           ELSEIF(lcnt.GT.ldiv1)THEN
           m = m-3
           ENDIF
           ENDIF
           CONTINUE
           m = m+1
           IF(ldiv2.EQ.1)THEN
WRITE(OUT,5205) (nucid(i,var2(L)),grams(i,var2(L)),L=m,lib2)
           ELSEIF(ldiv2.EQ.2)THEN
WRITE(OUT,5210) (nucid(i,var2(L)),grams(i,var2(L)),L=m,lib2)
           ENDIF
           ENDIF
           ENDIF
           ENDIF
C
C   FOR PRESENT YEAR, FROM LIBRARY 3 - FISSION PRODUCTS
C
           lcnt = 0
           IF(lib3.LE.3)THEN
           num = lib3
           IF(lib3.EQ.1)THEN
WRITE(OUT,5225) (nucid(i,var3(L)),grams(i,var3(L)),L=1,lib3)
5225  FORMAT(2x,'3',5x,A7,2x,e10.4,2x,'0',2x,'0')

```

	ELSEIF(lib3.EQ.2)THEN	PRE0836
	WRITE(OUT,5230) (nucid(i,var3(L)),grams(i,var3(L)), L=1,2)	PRE0837
5230	FORMAT(2x,'3',5x,2(a7,2x,e10.4,2x),'0',2x,'0')	PRE0838
	ELSEIF(lib3.EQ.3)THEN	PRE0839
	WRITE(OUT,5235) (nucid(i,var3(L)),grams(i,var3(L)), L=1,3)	PRE0840
5235	FORMAT(2x,'3',5x,3(a7,2x,e10.4,2x),'0',2x,'0')	PRE0841
	ENDIF	PRE0842
	ELSEIF(lib3.GT.3)THEN	PRE0843
	ldiv1 = int(lib3/3)	PRE0844
	ldiv2 = lib3 - 3*ldiv1	PRE0845
	m = 0	PRE0846
	IF(ldiv2.EQ.0)THEN	PRE0847
1045	m = m + 3	PRE0848
	IF(m.LE.lib3)THEN	PRE0849
	n = m-2	PRE0850
	WRITE(OUT,5240) (nucid(i,var3(L)),grams(i,var3(L)), L=n,m)	PRE0851
5240	FORMAT(2x,'3',5x,3(a7,2x,e10.4,2x),'0',2x,'0')	PRE0852
	GO TO 1045	PRE0853
	ENDIF	PRE0854
	ELSEIF(ldiv2.GT.0)THEN	PRE0855
	IF(ldiv1.EQ.1)THEN	PRE0856
	WRITE(OUT,5240) (nucid(i,var3(L)),grams(i,var3(L)), L=1,3)	PRE0857
	IF(ldiv2.EQ.1)THEN	PRE0858
	WRITE(OUT,5225) (nucid(i,var3(L)),grams(i,var3(L)),L=4,lib3)	PRE0859
	ELSEIF(ldiv2.EQ.2)THEN	PRE0860
	WRITE(OUT,5230) (nucid(i,var3(L)),grams(i,var3(L)),L=4,lib3)	PRE0861
	ENDIF	PRE0862
	ELSE	PRE0863
	DO 50 k = 1, ldiv1	PRE0864
1050	m = m + 3	PRE0865
	lcnt = lcnt + 1	PRE0866
	IF(lcnt.LE.ldiv1)THEN	PRE0867
	n = m-2	PRE0868
	WRITE(OUT,5240) (nucid(i,var3(L)),grams(i,var3(L)),L=n,m)	PRE0869
	GO TO 1050	PRE0870
	ELSEIF(lcnt.GT.ldiv1)THEN	PRE0871
	m = m-3	PRE0872
	ENDIF	PRE0873
50	CONTINUE	PRE0874
	m = m+1	PRE0875
	IF(ldiv2.EQ.1)THEN	PRE0876
	WRITE(OUT,5225) (nucid(i,var3(L)),grams(i,var3(L)), L=m,lib3)	PRE0877
	ELSEIF(ldiv2.EQ.2)THEN	PRE0878
	WRITE(OUT,5230) (nucid(i,var3(L)),grams(i,var3(L)),L=m,lib3)	PRE0879
	ENDIF	PRE0880
	ENDIF	PRE0881
	ENDIF	PRE0882
	ENDIF	PRE0883
C		PRE0884
	WRITE(OUT,5245)	PRE0885
5245	FORMAT(2X,'0')	PRE0886
C		PRE0887
55	CONTINUE	PRE0888
C		PRE0889
	IF(NOFIL.EQ.1)GO TO 1055	PRE0890
C		PRE0891
60	CONTINUE	PRE0892
C		PRE0893
C	GIVE USER OPTION TO RUN PROGRAM AGAIN	PRE0894
C		PRE0895
1055	WRITE(6,5000)	PRE0896
	WRITE(6,*)	PRE0897
	WRITE(6,*) ' WOULD YOU LIKE TO RUN PROGRAM AGAIN TO GENERATE '	PRE0898
	WRITE(6,*) ' MORE ORIGEN2 INPUT FILES? 1-YES 2-NO '	PRE0899
	WRITE(6,*)	PRE0900
	WRITE(6,5000)	PRE0901
	READ*, irun	PRE0902
	IF(irun.EQ.1)GO TO 1000	PRE0903
C		PRE0904
9999	STOP	PRE0905
C		PRE0906
	END	PRE0907
CC		PRE0908
	SUBROUTINE TYPE(itype,waste)	PRE0909
CC		PRE0910
	INTEGER itype	PRE0911

	CHARACTER waste*2	PRE0912
C	WRITE(6,*)' PLEASE CHOOSE WASTE CATEGORY: '	PRE0913
	WRITE(6,*)' 1 - CH TRU, 2 - RH TRU, OR 3 - BURIED TRU '	PRE0914
	WRITE(6,*)	PRE0915
	READ*, itype	PRE0916
C	IF(itype.EQ.1)THEN	PRE0917
	waste = 'CH'	PRE0918
	ELSEIF(itype.EQ.2)THEN	PRE0919
	waste = 'RH'	PRE0920
	ELSEIF(ITYPE.EQ.3)THEN	PRE0921
	waste = 'BU'	PRE0922
	ENDIF	PRE0923
C	RETURN	PRE0924
	END	PRE0925
CC	SUBROUTINE INNAME(site,itype,input)	PRE0926
CC	CHARACTER site*7, input*12	PRE0927
	INTEGER itype	PRE0928
C	IF(itype.EQ.1)THEN	PRE0929
	input = site//'2.dat'	PRE0930
	ELSEIF(itype.EQ.2)THEN	PRE0931
	input = site//'3.dat'	PRE0932
	ELSEIF(ITYPE.EQ.3)THEN	PRE0933
	input = site//'7.dat'	PRE0934
	ENDIF	PRE0935
C	WRITE(6,*)' PreORG INPUT DATA FILENAME: ', input	PRE0936
C	RETURN	PRE0937
	END	PRE0938
CC	SUBROUTINE NUMYRS(input,noyr,nofile)	PRE0939
CC	C THIS SUBROUTINE WILL COUNT THE NUMBER OF YEARS IN A DATA FILE	PRE0940
C	TO DETERMINE HOW MANY ORIGEN2 INPUT FILES MUST BE USED TO DECAY	PRE0941
C	ALL THE INVENTORIES OUT AS DESIRED. THIS IS DUE TO LIMITATIONS	PRE0942
C	IN THE ORIGEN2 CODE.	PRE0943
CC	IMPLICIT real(a-h,o-z), integer(i-n)	PRE0944
	INTEGER noyr, nofile	PRE0945
	CHARACTER input*12, line*133	PRE0946
C	OPEN(UNIT=10,FILE=input,STATUS='OLD')	PRE0947
C	noyr = 0	PRE0948
	nofile = 0	PRE0949
C	1000 READ(10,5000,END=1100) line	PRE0950
5000	FORMAT(A133)	PRE0951
C	IF(line(1:2).EQ.'19')THEN	PRE0952
	noyr = noyr + 1	PRE0953
	ENDIF	PRE0954
	GO TO 1000	PRE0955
C	IF(noyr.LE.15)THEN	PRE0956
	nofile = 1	PRE0957
	ELSEIF(noyr.GT.15)THEN	PRE0958
	ldiv1 = int(noyr/15)	PRE0959
	ldiv2 = noyr - ldiv1*15	PRE0960
	IF(ldiv2.EQ.0)nofile = ldiv1	PRE0961
	IF(ldiv2.gt.0)nofile=ldiv1+1	PRE0962
	ENDIF	PRE0963
C	CLOSE(UNIT=10)	PRE0964
	RETURN	PRE0965
	END	PRE0966
CC	SUBROUTINE OUTNAM(year,input,outfil)	PRE0967
CC	C THIS SUBROUTINE WILL COUNT THE NUMBER OF YEARS IN A DATA FILE	PRE0968
		PRE0969
		PRE0970
		PRE0971
		PRE0972
		PRE0973
		PRE0974
		PRE0975
		PRE0976
		PRE0977
		PRE0978
		PRE0979
		PRE0980
		PRE0981
		PRE0982
		PRE0983
		PRE0984
		PRE0985
		PRE0986
		PRE0987
		PRE0988

Appendix D: OR1_PreORG.for Program Listing

```

C TO DETERMINE HOW MANY ORIGEN2 INPUT FILES MUST BE USED TO DECAY PRE0988
C ALL THE INVENTORIES OUT AS DESIRED. THIS IS DUE TO LIMITATIONS PRE0989
C IN THE ORIGEN2 CODE. PRE0990
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC PRE0991
IMPLICIT real(a-h,o-z), integer(i-n) PRE0992
CHARACTER input*12, outfil*12, type*2, year*4 PRE0993
C PRE0994
IF(input(8:8).EQ.'2')THEN PRE0995
  type = 'ch' PRE0996
ELSEIF(input(8:8).EQ.'3')THEN PRE0997
  type = 'rh' PRE0998
ELSEIF(input(8:8).EQ.'7')THEN PRE0999
  type = 'bu' PRE1000
ENDIF PRE1001
C PRE1002
outfil = input(1:4)//type//year(3:4)//'.u5' PRE1003
WRITE(6,*)' PreORG OUTPUT FILENAME: ', outfil PRE1004
C PRE1005
RETURN PRE1006
END PRE1007
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC PRE1008
SUBROUTINE OTHER(site) PRE1009
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC PRE1010
CHARACTER site*7, name*12, ch*12, rh*12, bu*12 PRE1011
C PRE1012
WRITE(6,*)' PLEASE ENTER SITE NAME: ' PRE1013
READ 2, name PRE1014
2 FORMAT(A12) PRE1015
WRITE(6,*)' PREORG ACCEPTS FOLLOWING SITE NAME ON ALL FILES: ' PRE1016
C PRE1017
site = name(1:7) PRE1018
WRITE(6,*)' SITE NAME: ', site PRE1019
C PRE1020
ch = site//'2.dat' PRE1021
rh = site//'3.dat' PRE1022
bu = site//'7.dat' PRE1023
C PRE1024
WRITE(6,*)' TO RUN SUCESSFULLY THE FOLLOWING WASTE TYPES MUST ' PRE1025
WRITE(6,*)' HAVE THE FOLLOWING NAMES: ' PRE1026
WRITE(6,*) PRE1027
WRITE(6,*)' CH WASTE ', ch PRE1028
WRITE(6,*) PRE1029
WRITE(6,*)' RH WASTE ', rh PRE1030
WRITE(6,*) PRE1031
WRITE(6,*)' BURIED WASTE ', bu PRE1032
WRITE(6,*) PRE1033
C PRE1034
RETURN PRE1035
END PRE1036
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC PRE1037
SUBROUTINE CONV (name,activ,namy,activy,nuclib) PRE1038
C THIS SUBROUTINE WAS WRITTEN BY HOLLY TRELLUE, 6/8/95 PRE1039
C MODIFIED BY J.L. POWELL 6/29/95 WITH ORIGEN2 DECAY LIBRARY HALF LIVES PRE1040
C NUCLIDES ADDED 10/16/95 PRE1041
CHARACTER name*6, namy*6 PRE1042
INTEGER nuclib PRE1043
REAL activ, activy, m,Thalf,sa,sal PRE1044
C PRE1045
C ALL VALUES FOR ATOMIC MASS NUMBERS AND HALFLIVES IN THIS FILE WERE TAK PRE1046
C FROM THE CHART OF THE NUCLIDES. PRE1047
C PRE1048
IF (name(1:5).EQ.'AC227') THEN PRE1049
C PRE1050
C ATOMIC WEIGHT VALUES(M) WERE CALCULATED BY USING THE MASS EXCESS VALUE PRE1051
C FOR EACH ISOTOPE. MASS EXCESS VALUES WERE TAKEN FROM "NUCLEAR WALLET PRE1052
C CARDS" BY JAGDISH TULI. JANUARY 1985. PRE1053
C PRE1054
m=227.0 + 25.849/931.5016 PRE1055
C PRE1056
C FOR CONFORMITY, THE HALF LIVES ARE CONVERTED TO SECONDS FOR PRE1057
C EVERY ISOTOPE. AC227'S HALF LIFE IS 21.77 YEARS. ALL VALUES FOR PRE1058
C HALFLIVES IN THIS PROGRAM COME FROM THE ORIGEN2 DECAY LIBRARY PRE1059
C PRE1060
Thalf = 6.871e08 PRE1061
nuclib=2 PRE1062
namy(1:6)='892270' PRE1063

```

```

ELSEIF (name(1:5).EQ.'AG110') THEN
  if(name(6:6).EQ.'M') THEN
    m=110.0 - 87.341/931.5016
    Thalf =2.159E07
    namy(1:6)='471101'
    nuclib=3
  ELSE
    m=110.0 - 87.459/931.5016
    Thalf =2.460E01
    namy(1:6)='471100'
    nuclib=3
  ENDIF
ELSEIF (name(1:5).EQ.'AM240') THEN
  m=240.0 + 51.498/931.5016
  Thalf =1.829e05
  namy(1:6)='952400'
  nuclib=2
ELSEIF (name(1:5).EQ.'AM241') THEN
  m=241.0 + 52.931/931.5016
  Thalf = 1.364e10
  namy(1:6)='952410'
  nuclib=2
ELSEIF (name(1:5).EQ.'AM242') THEN
  m=242.0 + 55.463/931.5016
  Thalf = 5.767e04
  namy(1:6)='952420'
  nuclib=2
ELSEIF (name(1:5).EQ.'AM243') THEN
  m=243.0 + 57.169/931.5016
  Thalf=2.329e11
  namy(1:6)='952430'
  nuclib=2
ELSEIF (name(1:5).EQ.'AM244') THEN
  m=244.0 + 59.877/931.5016
  Thalf=3.636e04
  namy(1:6)='952440'
  nuclib=2
ELSEIF (name(1:6).EQ.'BA137M') THEN
  m=137.0 - 87.070/931.5016
  Thalf=1.531e02
  namy(1:6)='561371'
  nuclib=1
ELSEIF (name(1:5).EQ.'BI214') THEN
  m=214.0 - 1.218/931.5016
  Thalf=1.194e03
  namy(1:6)='832140'
  nuclib=2
ELSEIF (name(1:5).EQ.'BK249') THEN
  m=249.0 + 69.843/931.5016
  Thalf=2.765e07
  namy(1:6)='972490'
  nuclib=2
ELSEIF (name(1:3).EQ.'C14') THEN
  m=14.0 + 3.02/931.5016
  Thalf=1.808e11
  namy(1:6)='060140'
  nuclib=1
ELSEIF (name(1:5).EQ.'CD109') THEN
  m=109.0 - 88.507/931.5016
  Thalf=4.009e07
  namy(1:6)='481090'
  nuclib=1
ELSEIF (name(1:6).EQ.'CD113M') THEN
  m=113.0 - 88.786/931.5016
  Thalf=4.604E08
  namy(1:6)='481131'
  nuclib=3
ELSEIF (name(1:5).EQ.'CE144') THEN
  m=144.0 - 80.441/931.5016
  Thalf=2.456e07
  namy(1:6)='581440'
  nuclib=1
ELSEIF (name(1:5).EQ.'CF249') THEN
  m=249.0 + 69.717/931.5016
  Thalf = 1.106e10
  namy(1:6)='982490'

```

PRE1064
 PRE1065
 PRE1066
 PRE1067
 PRE1068
 PRE1069
 PRE1070
 PRE1071
 PRE1072
 PRE1073
 PRE1074
 PRE1075
 PRE1076
 PRE1077
 PRE1078
 PRE1079
 PRE1080
 PRE1081
 PRE1082
 PRE1083
 PRE1084
 PRE1085
 PRE1086
 PRE1087
 PRE1088
 PRE1089
 PRE1090
 PRE1091
 PRE1092
 PRE1093
 PRE1094
 PRE1095
 PRE1096
 PRE1097
 PRE1098
 PRE1099
 PRE1100
 PRE1101
 PRE1102
 PRE1103
 PRE1104
 PRE1105
 PRE1106
 PRE1107
 PRE1108
 PRE1109
 PRE1110
 PRE1111
 PRE1112
 PRE1113
 PRE1114
 PRE1115
 PRE1116
 PRE1117
 PRE1118
 PRE1119
 PRE1120
 PRE1121
 PRE1122
 PRE1123
 PRE1124
 PRE1125
 PRE1126
 PRE1127
 PRE1128
 PRE1129
 PRE1130
 PRE1131
 PRE1132
 PRE1133
 PRE1134
 PRE1135
 PRE1136
 PRE1137
 PRE1138
 PRE1139

nuclib=2	PRE1140
ELSEIF (name(1:5).EQ.'CF250') THEN	PRE1141
m=250.0 + 71.167/931.5016	PRE1142
Thalf=4.128e08	PRE1143
namy(1:6)='982500'	PRE1144
nuclib=2	PRE1145
ELSEIF (name(1:5).EQ.'CF251') THEN	PRE1146
m=251.0 + 74.129/931.5016	PRE1147
Thalf=2.834e10	PRE1148
namy(1:6)='982510'	PRE1149
nuclib=2	PRE1150
ELSEIF (name(1:5).EQ.'CF252') THEN	PRE1151
m=252.0 + 76.03/931.5016	PRE1152
Thalf=8.325e07	PRE1153
namy(1:6)='982520'	PRE1154
nuclib=2	PRE1155
ELSEIF (name(1:5).EQ.'CM242') THEN	PRE1156
m=242.0 + 54.800/931.5016	PRE1157
Thalf=1.410e07	PRE1158
namy(1:6)='962420'	PRE1159
nuclib=2	PRE1160
ELSEIF (name(1:5).EQ.'CM243') THEN	PRE1161
m=243.0 + 57.177/931.5016	PRE1162
Thalf=8.994e08	PRE1163
namy(1:6)='962430'	PRE1164
nuclib=2	PRE1165
ELSEIF (name(1:5).EQ.'CM244') THEN	PRE1166
m=244.0 + 58.449/931.5016	PRE1167
Thalf=5.715e08	PRE1168
namy(1:6)='962440'	PRE1169
nuclib=2	PRE1170
ELSEIF (name(1:5).EQ.'CM245') THEN	PRE1171
m=245.0 + 60.998/931.5016	PRE1172
Thalf=2.682e11	PRE1173
namy(1:6)='962450'	PRE1174
nuclib=2	PRE1175
ELSEIF (name(1:5).EQ.'CM246') THEN	PRE1176
m=246.0 + 62.614/931.5016	PRE1177
Thalf=1.493e11	PRE1178
namy(1:6)='962460'	PRE1179
nuclib=2	PRE1180
ELSEIF (name(1:5).EQ.'CM248') THEN	PRE1181
m=248.0 + 67.388/931.5016	PRE1182
Thalf=1.070e13	PRE1183
namy(1:6)='962480'	PRE1184
nuclib=2	PRE1185
ELSEIF (name(1:4).EQ.'CO58') THEN	PRE1186
m=58.0 - 59.844/931.5016	PRE1187
Thalf=6.115e06	PRE1188
namy(1:6)='270580'	PRE1189
nuclib=1	PRE1190
ELSEIF (name(1:4).EQ.'CO60') THEN	PRE1191
m=60.0 - 61.646/931.5016	PRE1192
Thalf=1.663e08	PRE1193
namy(1:6)='270600'	PRE1194
nuclib=1	PRE1195
ELSEIF (name(1:4).EQ.'CR51') THEN	PRE1196
m=51.0 - 51.447/931.5016	PRE1197
Thalf=2.394e06	PRE1198
namy(1:6)='240510'	PRE1199
nuclib=1	PRE1200
ELSEIF (name(1:5).EQ.'CS134') THEN	PRE1201
m=134.0 - 86.906/931.5016	PRE1202
Thalf=6.507e07	PRE1203
namy(1:6)='551340'	PRE1204
nuclib=1	PRE1205
ELSEIF (name(1:5).EQ.'CS135') THEN	PRE1206
m=135.0 - 87.662/931.5016	PRE1207
Thalf=7.258E13	PRE1208
namy(1:6)='551350'	PRE1209
nuclib=1	PRE1210
ELSEIF (name(1:5).EQ.'CS137') THEN	PRE1211
m=137.0 - 86.556/931.5016	PRE1212
Thalf=9.467e08	PRE1213
namy(1:6)='551370'	PRE1214
nuclib=1	PRE1215

```

ELSEIF (name(1:5).EQ.'ES253') THEN
  m=253.0 + 79.007/931.5016
  Thalf=1.769e06
  namy(1:6)='992530'
  nuclib=2
ELSEIF (name(1:5).EQ.'ES254') THEN
  IF (name(6:6).EQ.'M') THEN
    m=254.0 + 82.072/931.5016
    Thalf=1.415e05
    namy(1:6)='992541'
    nuclib=2
  ELSE
    m=254.0 + 81.994/931.5016
    Thalf=2.382e07
    namy(1:6)='992540'
    nuclib=2
  ENDIF
ELSEIF (name(1:5).EQ.'EU150') THEN
  m=150.0 - 74.800/931.5016
  Thalf=1.136e09
  namy(1:6)='631500'
  nuclib=3
ELSEIF (name(1:5).EQ.'EU152') THEN
  m=152.0 - 72.899/931.5016
  Thalf=4.292e08
  namy(1:6)='631520'
  nuclib=3
ELSEIF (name(1:5).EQ.'EU154') THEN
  m=154.0 - 71.748/931.5016
  Thalf=2.714e08
  namy(1:6)='631540'
  nuclib=3
ELSEIF (name(1:5).EQ.'EU155') THEN
  m=155.0 - 71.829/931.5016
  Thalf=1.565e08
  namy(1:6)='631550'
  nuclib=3
ELSEIF (name(1:4).EQ.'FE55') THEN
  m= 55.0 - 57.476/931.5016
  Thalf=8.2048e07
  namy(1:6)='260550'
  nuclib=1
ELSEIF (name(1:4).EQ.'FE59') THEN
  m=59.0 - 60.661/931.5016
  Thalf=3.888e06
  namy(1:6)='260590'
  nuclib=1
ELSEIF (name(1:5).EQ.'GD162') THEN
  m = 162.0 -64.240/931.5016
  Thalf = 6.000E02
  namy(1:6) = '641620'
  nuclib=1
ELSEIF (name(1:2).EQ.'H3') THEN
  m=3.0 + 14.95/931.5016
  Thalf=3.897e08
  namy(1:6)='010030'
  nuclib=1
ELSEIF (name(1:4).EQ.'I129') THEN
  m=129.0 - 88.507/931.5016
  Thalf=4.954e14
  namy(1:6)='531290'
  nuclib=1
ELSEIF (name(1:4).EQ.'KR85') THEN
  m=85.0 - 81.477/931.5016
  Thalf=3.383e08
  namy(1:6)='360850'
  nuclib=1
ELSEIF (name(1:4).EQ.'MN54') THEN
  m=54.0 - 55.553/931.5016
  Thalf=2.700e07
  namy(1:6)='250540'
  nuclib=1
ELSEIF (name(1:4).EQ.'NB95') THEN
  IF (name(5:5).EQ.'M') THEN
    m=95.0 - 86.547/931.5016
    Thalf=3.118E05

```

```

PRE1216
PRE1217
PRE1218
PRE1219
PRE1220
PRE1221
PRE1222
PRE1223
PRE1224
PRE1225
PRE1226
PRE1227
PRE1228
PRE1229
PRE1230
PRE1231
PRE1232
PRE1233
PRE1234
PRE1235
PRE1236
PRE1237
PRE1238
PRE1239
PRE1240
PRE1241
PRE1242
PRE1243
PRE1244
PRE1245
PRE1246
PRE1247
PRE1248
PRE1249
PRE1250
PRE1251
PRE1252
PRE1253
PRE1254
PRE1255
PRE1256
PRE1257
PRE1258
PRE1259
PRE1260
PRE1261
PRE1262
PRE1263
PRE1264
PRE1265
PRE1266
PRE1267
PRE1268
PRE1269
PRE1270
PRE1271
PRE1272
PRE1273
PRE1274
PRE1275
PRE1276
PRE1277
PRE1278
PRE1279
PRE1280
PRE1281
PRE1282
PRE1283
PRE1284
PRE1285
PRE1286
PRE1287
PRE1288
PRE1289
PRE1290
PRE1291

```

```

        namy(1:6)='410951'
        nuclib=1
ELSE
        m=95.0 - 86.783/931.5016
        Thalf=3.037e06
        namy(1:6)='410950'
        nuclib=1
ENDIF
ELSEIF (name(1:4).EQ.'NI59') THEN
        m=59.0 - 61.153/931.5016
        Thalf=2.5246e12
        namy(1:6)='280590'
        nuclib=1
ELSEIF (name(1:4).EQ.'NI63') THEN
        m=63.0 - 65.512/931.5016
        Thalf=2.9032e9
        namy='280630'
        nuclib=1
ELSEIF (name(1:5).EQ.'NP237') THEN
        m=237.0 + 44.868/931.5016
        Thalf=6.753e13
        namy='932370'
        nuclib=2
ELSEIF (name(1:5).EQ.'NP239') THEN
        m=239.0 + 49.306/931.5016
        Thalf=2.035e05
        namy='932390'
        nuclib=2
ELSEIF (name(1:5).EQ.'PA231') THEN
        m=231.0 + 33.422/931.5016
        Thalf=1.034e12
        namy='912310'
        nuclib=2
ELSEIF (name(1:5).EQ.'PA233') THEN
        m=233.0 + 37.485/931.5016
        Thalf=2.333e06
        namy='912330'
        nuclib=2
ELSEIF (name(1:5).EQ.'PB210') THEN
        m=210.0 - 14.752/931.5016
        Thalf=7.037e08
        namy='822100'
        nuclib=2
ELSEIF (name(1:5).EQ.'PB214') THEN
        m=214.0 - .188/931.5016
        Thalf=1.608e03
        namy='822140'
        nuclib=2
ELSEIF (name(1:5).EQ.'PD107') THEN
        m=107.0 - 88.374/931.5016
        Thalf=2.050E14
        namy='461070'
        nuclib=1
ELSEIF (name(1:5).EQ.'PM147') THEN
        m=147.0 - 79.052/931.5016
        Thalf=8.279e07
        namy='611470'
        nuclib=1
ELSEIF (name(1:5).EQ.'PO210') THEN
        m=210.0 - 15.977/931.5016
        Thalf=1.196e07
        namy='842100'
        nuclib=2
ELSEIF (name(1:5).EQ.'PO214') THEN
        m=214.0 - 4.493/931.5016
        Thalf=1.643e-04
        namy='842140'
        nuclib=2
ELSEIF (name(1:5).EQ.'PO218') THEN
        m=218.0 + 8.351/931.5016
        Thalf=1.830e02
        namy='842180'
        nuclib=2
ELSEIF (name(1:5).EQ.'PR144') THEN
        m=144.0 - 80.76/931.5016
        Thalf=1.037e03

```

PRE1292
 PRE1293
 PRE1294
 PRE1295
 PRE1296
 PRE1297
 PRE1298
 PRE1299
 PRE1300
 PRE1301
 PRE1302
 PRE1303
 PRE1304
 PRE1305
 PRE1306
 PRE1307
 PRE1308
 PRE1309
 PRE1310
 PRE1311
 PRE1312
 PRE1313
 PRE1314
 PRE1315
 PRE1316
 PRE1317
 PRE1318
 PRE1319
 PRE1320
 PRE1321
 PRE1322
 PRE1323
 PRE1324
 PRE1325
 PRE1326
 PRE1327
 PRE1328
 PRE1329
 PRE1330
 PRE1331
 PRE1332
 PRE1333
 PRE1334
 PRE1335
 PRE1336
 PRE1337
 PRE1338
 PRE1339
 PRE1340
 PRE1341
 PRE1342
 PRE1343
 PRE1344
 PRE1345
 PRE1346
 PRE1347
 PRE1348
 PRE1349
 PRE1350
 PRE1351
 PRE1352
 PRE1353
 PRE1354
 PRE1355
 PRE1356
 PRE1357
 PRE1358
 PRE1359
 PRE1360
 PRE1361
 PRE1362
 PRE1363
 PRE1364
 PRE1365
 PRE1366
 PRE1367

namy='591440'	PRE1368
nuclib=1	PRE1369
ELSEIF (name(1:5).EQ.'PU236') THEN	PRE1370
m=236.0 + 42.879/931.5016	PRE1371
Thalf=8.997e07	PRE1372
namy='942360'	PRE1373
nuclib=2	PRE1374
ELSEIF (name(1:5).EQ.'PU238') THEN	PRE1375
m=238.0 + 46.16/931.5016	PRE1376
Thalf=2.769e09	PRE1377
namy='942380'	PRE1378
nuclib=2	PRE1379
ELSEIF (name(1:5).EQ.'PU239') THEN	PRE1380
m=239.0 + 48.584/931.5016	PRE1381
Thalf=7.594e11	PRE1382
namy='942390'	PRE1383
nuclib=2	PRE1384
ELSEIF (name(1:5).EQ.'PU240') THEN	PRE1385
m=240.0 + 50.122/931.5016	PRE1386
Thalf=2.063e11	PRE1387
namy='942400'	PRE1388
nuclib=2	PRE1389
ELSEIF (name(1:5).EQ.'PU241') THEN	PRE1390
m=241.0 + 52.952/931.5016	PRE1391
Thalf=4.544e08	PRE1392
namy='942410'	PRE1393
nuclib=2	PRE1394
ELSEIF (name(1:5).EQ.'PU242') THEN	PRE1395
m=242.0 + 54.713/931.5016	PRE1396
Thalf=1.221e13	PRE1397
namy='942420'	PRE1398
nuclib=2	PRE1399
ELSEIF (name(1:5).EQ.'PU244') THEN	PRE1400
m=244.0 + 59.802/931.5016	PRE1401
Thalf=2.607e15	PRE1402
namy='942440'	PRE1403
nuclib=2	PRE1404
ELSEIF (name(1:5).EQ.'RA223') THEN	PRE1405
m=223.0 + 17.232/931.5016	PRE1406
Thalf=9.879e05	PRE1407
namy='882230'	PRE1408
nuclib=2	PRE1409
ELSEIF (name(1:5).EQ.'RA226') THEN	PRE1410
m=226.0 + 23.662/931.5016	PRE1411
Thalf=5.049e10	PRE1412
namy='882260'	PRE1413
nuclib=2	PRE1414
ELSEIF (name(1:5).EQ.'RA228') THEN	PRE1415
m=228.0 + 28.936/931.5016	PRE1416
Thalf=2.1143e08	PRE1417
namy='882280'	PRE1418
nuclib=2	PRE1419
ELSEIF (name(1:5).EQ.'RH106') THEN	PRE1420
m=106.0 - 86.365/931.5016	PRE1421
Thalf=29.9	PRE1422
namy='451060'	PRE1423
nuclib=1	PRE1424
ELSEIF (name(1:5).EQ.'RN222') THEN	PRE1425
m=222.0 + 16.367/931.5016	PRE1426
Thalf=3.304e05	PRE1427
namy='862220'	PRE1428
nuclib=2	PRE1429
ELSEIF (name(1:5).EQ.'RU106') THEN	PRE1430
m=106.0 - 86.326/931.5016	PRE1431
Thalf=3.181e07	PRE1432
namy='441060'	PRE1433
nuclib=1	PRE1434
ELSEIF (name(1:5).EQ.'SB125') THEN	PRE1435
m=125.0 - 88.258/931.5016	PRE1436
Thalf=8.741e07	PRE1437
namy='511250'	PRE1438
nuclib=1	PRE1439
ELSEIF (name(1:5).EQ.'SB126') THEN	PRE1440
IF (NAME(6:6).EQ.'M') THEN	PRE1441
m=126.0 - 86.382/931.5016	PRE1442
Thalf=1.140E03	PRE1443

namy='511261'	PRE1444
nuclib=1	PRE1445
ELSE	PRE1446
m=126.0 - 86.400/931.5016	PRE1447
Thalf=1.071E06	PRE1448
namy='511260'	PRE1449
nuclib=1	PRE1450
ENDIF	PRE1451
ELSEIF (name(1:4).EQ.'SE79') THEN	PRE1452
m=79.0 - 75.920/931.5016	PRE1453
Thalf=2.050E12	PRE1454
namy='340790'	PRE1455
nuclib=1	PRE1456
ELSEIF (name(1:5).EQ.'SM151') THEN	PRE1457
m=151.0 - 74.587/931.5016	PRE1458
Thalf=2.840e09	PRE1459
namy='621510'	PRE1460
nuclib=1	PRE1461
ELSEIF (name(1:6).EQ.'SN119M') THEN	PRE1462
m=119.0 - 89.978/931.5016	PRE1463
Thalf=2.117E07	PRE1464
namy='501191'	PRE1465
nuclib=1	PRE1466
ELSEIF (name(1:6).EQ.'SN121M') THEN	PRE1467
m=121.0 - 89.197/931.5016	PRE1468
Thalf=1.577E09	PRE1469
namy='501211'	PRE1470
nuclib=1	PRE1471
ELSEIF (name(1:6).EQ.'SN123M') THEN	PRE1472
m=123.0 - 87.795/931.5016	PRE1473
Thalf=2.405E03	PRE1474
namy='501231'	PRE1475
nuclib=1	PRE1476
ELSEIF (name(1:5).EQ.'SN126') THEN	PRE1477
m=126.0 - 86.021/931.5016	PRE1478
Thalf=3.156E12	PRE1479
namy='501260'	PRE1480
nuclib=3	PRE1481
ELSEIF (name(1:4).EQ.'SR89') THEN	PRE1482
m=89.0 - 86.211/931.5016	PRE1483
Thalf=4.363e06	PRE1484
namy='380890'	PRE1485
nuclib=1	PRE1486
ELSEIF (name(1:4).EQ.'SR90') THEN	PRE1487
m=90.0 - 85.942/931.5016	PRE1488
Thalf=9.190e08	PRE1489
namy='380900'	PRE1490
nuclib=1	PRE1491
ELSEIF (name(1:5).EQ.'TA182') THEN	PRE1492
m=182.0 - 46.436/931.5016	PRE1493
Thalf=9.936e06	PRE1494
namy='731820'	PRE1495
nuclib=1	PRE1496
ELSEIF (name(1:5).EQ.'TB162') THEN	PRE1497
m=162.0 - 65.680/931.5016	PRE1498
Thalf=4.482E02	PRE1499
namy='651620'	PRE1500
nuclib=1	PRE1501
ELSEIF (name(1:4).EQ.'TC99') THEN	PRE1502
m=99.0 - 87.324/931.5016	PRE1503
Thalf=6.722e12	PRE1504
namy='430990'	PRE1505
nuclib=1	PRE1506
ELSEIF (name(1:6).EQ.'TE125M') THEN	PRE1507
m=125.0 - 88.879/931.5016	PRE1508
Thalf=5.011e06	PRE1509
namy='521251'	PRE1510
nuclib=1	PRE1511
ELSEIF (name(1:5).EQ.'TE127') THEN	PRE1512
IF (NAME(6:6).EQ.'M') THEN	PRE1513
m=127.0 - 88.198/931.5016	PRE1514
Thalf=9.418E06	PRE1515
namy='521271'	PRE1516
nuclib=1	PRE1517
ELSE	PRE1518
m=127.0 - 88.286/931.5016	PRE1519

```

      Thalf=3.366E04
      namy='521270'
      nuclib=1
    ENDIF
  ELSEIF (name(1:5).EQ.'TH228') THEN
    m=228.0 + 26.749/931.5016
    Thalf=6.037e07
    namy='902280'
    nuclib=2
  ELSEIF (name(1:5).EQ.'TH230') THEN
    m=230.0 + 30.858/931.5016
    Thalf=2.430e12
    namy='902300'
    nuclib=2
  ELSEIF (name(1:5).EQ.'TH232') THEN
    m=232.0 + 35.444/931.5016
    Thalf=4.434e17
    namy='902320'
    nuclib=2
  ELSEIF (name(1:5).EQ.'TH234') THEN
    m=234.0 + 40.607/931.5016
    Thalf=2.082e06
    namy='902340'
    nuclib=2
  ELSEIF (name(1:4).EQ.'U232') THEN
    m=232.0 + 34.587/931.5016
    Thalf=2.272e09
    namy='922320'
    nuclib=2
  ELSEIF (name(1:4).EQ.'U233') THEN
    m=233.0 + 36.915/931.5016
    Thalf=5.002e12
    namy='922330'
    nuclib=2
  ELSEIF (name(1:4).EQ.'U234') THEN
    m=234.0 + 38.141/931.5016
    Thalf=7.716e12
    namy='922340'
    nuclib=2
  ELSEIF (name(1:4).EQ.'U235') THEN
    m=235.0 + 40.915/931.5016
    Thalf=2.221e16
    namy='922350'
    nuclib=2
  ELSEIF (name(1:4).EQ.'U236') THEN
    m=236.0 + 42.441/931.5016
    Thalf=7.389e14
    namy='922360'
    nuclib=2
  ELSEIF (name(1:4).EQ.'U238') THEN
    m=238.0 + 47.305/931.5016
    Thalf=1.410e17
    namy='922380'
    nuclib=2
  ELSEIF (name(1:3).EQ.'Y90') THEN
    m=90.0 - 86.488/931.5016
    Thalf=2.304e05
    namy='390900'
    nuclib=1
  ELSEIF (name(1:4).EQ.'ZN65') THEN
    m=65.0 - 65.91/931.5016
    Thalf=2.107e07
    namy='300650'
    nuclib=1
  ELSEIF (name(1:4).EQ.'ZR93') THEN
    m=93.0 - 87.120/931.5016
    Thalf=4.828E13
    namy='400930'
    nuclib=1
  ELSEIF (name(1:4).EQ.'ZR95') THEN
    m=95.0 - 85.659/931.5016
    Thalf=5.528e06
    namy='400950'
    nuclib=1
  ELSE
    m=.5

```

```

PRE1520
PRE1521
PRE1522
PRE1523
PRE1524
PRE1525
PRE1526
PRE1527
PRE1528
PRE1529
PRE1530
PRE1531
PRE1532
PRE1533
PRE1534
PRE1535
PRE1536
PRE1537
PRE1538
PRE1539
PRE1540
PRE1541
PRE1542
PRE1543
PRE1544
PRE1545
PRE1546
PRE1547
PRE1548
PRE1549
PRE1550
PRE1551
PRE1552
PRE1553
PRE1554
PRE1555
PRE1556
PRE1557
PRE1558
PRE1559
PRE1560
PRE1561
PRE1562
PRE1563
PRE1564
PRE1565
PRE1566
PRE1567
PRE1568
PRE1569
PRE1570
PRE1571
PRE1572
PRE1573
PRE1574
PRE1575
PRE1576
PRE1577
PRE1578
PRE1579
PRE1580
PRE1581
PRE1582
PRE1583
PRE1584
PRE1585
PRE1586
PRE1587
PRE1588
PRE1589
PRE1590
PRE1591
PRE1592
PRE1593
PRE1594
PRE1595

```

[illegible]

PRE1596
PRE1597
PRE1598
PRE1599
PRE1600
PRE1601
PRE1602
PRE1603
PRE1604
PRE1605
PRE1606
PRE1607
PRE1608
PRE1609
PRE1610
PRE1611
PRE1612
PRE1613
PRE1614
PRE1615
PRE1616
PRE1617
PRE1618
PRE1619
PRE1620
PRE1621
PRE1622
PRE1623
PRE1624
PRE1625
PRE1626
PRE1627
PRE1628
PRE1629
PRE1630
PRE1631
PRE1632
PRE1633
PRE1634
PRE1635
PRE1636
PRE1637
PRE1638
PRE1639
PRE1640
PRE1641
PRE1642
PRE1643
PRE1644
PRE1645
PRE1646
PRE1647
PRE1648
PRE1649
PRE1650
PRE1651
PRE1652
PRE1653
PRE1654
PRE1655
PRE1656
PRE1657
PRE1658
PRE1659
PRE1660
PRE1661
PRE1662
PRE1663
PRE1664
PRE1665
PRE1666
PRE1667
PRE1668
PRE1669
PRE1670
PRE1671

```

+ /, ' _/_/_/ _/ _/ _/_/_/_/ _/ _/_/_/ _/ PRE1672
+ /, ' _/ _/_/_/ _/ PRE1673
+ /, ' _/ _/_/_/ _/ PRE1674
+ /, ' _/ _/_/_/ PRE1675
+ /, ' _/ _/_/_/ PRE1676
+ /, ' _/ _/_/_/ PRE1677
+ /, ' _/ _/_/_/ PRE1678
+ /, ' _/ _/_/_/ PRE1679
+ /, ' _/ _/_/_/ PRE1680
+ /, ' _/ _/_/_/ PRE1681
+ /, ' ***** PRE1682
+ /, ' ***** PRE1683
+ /, ' ***** PRE1684
+ /, ' * ISSUED BY * PRE1685
+ /, ' * * PRE1686
+ /, ' * SANDIA LABORATORI* PRE1687
+ /, ' * ES, * PRE1688
+ ) PRE1689
WRITE(6,9804) PRE1690
9804 FORMAT( PRE1691
+ /, ' * A PRIME CONTRACT', PRE1692
+ /, ' OR * PRE1693
+ /, ' ***** TO THE ', PRE1694
+ /, ' * * PRE1695
+ /, ' * UNITED STA', PRE1696
+ /, ' TES * DEPARTMEN', PRE1697
+ /, ' T * OF ', PRE1698
+ /, ' * ENERGY ', PRE1699
+ /, ' * PRE1700
+ /, ' * PRE1701
+ /, ' ***** ---NOTICE-', PRE1702
+ /, ' --- ***** PRE1703
+ /, ' *THIS REPORT WAS PREPARED AS AN A', PRE1704
+ /, ' CCOUNT OF WORK SPONSORED* PRE1705
+ /, ' * BY THE UNITED STATES GOVERNMEN', PRE1706
+ /, ' T. NEITHER THE UNITED * PRE1707
+ /, ' STATES NOR THE UNITED STATES ', PRE1708
+ /, ' DEPARTMENT OF ENERGY, * PRE1709
+ ) PRE1710
WRITE(6,9805) PRE1711
9805 FORMAT( PRE1712
+ /, ' * NOR ANY OF THEIR ', PRE1713
+ /, ' EMPLOYEES, * PRE1714
+ /, ' * NOR ANY OF THEIR CONTRACTORS, S', PRE1715
+ /, ' UBCONTRACTORS, OR THEIR * PRE1716
+ /, ' EMPLOYEES, MAKES ANY WARRANTY, ', PRE1717
+ /, ' EXPRESS OR IMPLIED, OR * PRE1718
+ /, ' ASSUMES ANY LEGAL LIABILITY OR ', PRE1719
+ /, ' RESPONSIBILITY FOR THE * PRE1720
+ /, ' ***** ACCURACY', PRE1721
+ /, ' ***** * PRE1722
+ /, ' * COMPLETENE', * PRE1723
+ /, ' SS * * PRE1724
+ /, ' * OR USEFULN', * PRE1725
+ /, ' ESS * * PRE1726
+ /, ' * OF ANY ', * PRE1727
+ /, ' * * PRE1728
+ /, ' * INFORMATIO', * PRE1729
+ /, ' N, * * PRE1730
+ /, ' * APPARATUS', * PRE1731
+ /, ' * * PRE1732
+ ) PRE1733
WRITE(6,9806) PRE1734
9806 FORMAT( PRE1735
+ /, ' * ***** PRODUCT', * PRE1736
+ /, ' * ***** * PRE1737
+ /, ' * OR PROCES', * PRE1738
+ /, ' S * * PRE1739
+ /, ' * DISCLOSED', * PRE1740
+ /, ' * * PRE1741
+ /, ' * OR REPRESE', * PRE1742
+ /, ' NTS * * PRE1743
+ /, ' * THAT ITS', * PRE1744
+ /, ' * * PRE1745
+ /, ' * * PRE1746
+ /, ' * * PRE1747

```

Appendix D: OR1_PreORG.for Program Listing

```

+ /, ' * * ** USE WOULD ', PRE1748
+ 'NOT ** * PRE1749
+ /, ' ***** ** INFRINGE', PRE1750
+ ' ***** PRE1751
+ /, ' ** PRIVATEL', PRE1752
+ 'Y ** PRE1753
+ /, ' ** OWNED ', PRE1754
+ ' ** PRE1755
+ /, ' ** RIGHTS.', PRE1756
+ ' ** PRE1757
+ ) PRE1758
WRITE(6,9807) PRE1759
9807 FORMAT( PRE1760
+ ' ** PRE1761
+ ' ** PRE1762
+ /, ' ** PRE1763
+ ' ** PRE1764
+ /, ' ** PRE1765
+ ' ** PRE1766
+ /, ' *****', PRE1767
+ '***** PRE1768
+ /, ' ', PRE1769
+ ' ', PRE1770
+ /, ' ', PRE1771
+ ' ', PRE1772
+ /, '1234567890123456789012345678901234567890', PRE1773
+ '12345678901234567890123456789012', PRE1774
+ /, ' 1 2 3 4', PRE1775
+ ' 5 6 7 ' PRE1776
+ /, ' ', PRE1777
+ ' ', PRE1778
+ /, ' ', PRE1779
+ ' ', PRE1780
+ ) PRE1781
WRITE(6,9808) PRE1782
9808 FORMAT( PRE1783
+ ) PRE1784
RETURN PRE1785
END PRE1786

```

**APPENDIX E: PERSONAL COMMUNICATION WITH SAYAN CHAKRABORTI OF
SCIENCE APPLICATIONS INTERNATIONAL CORPORATION BY
JENNIFER POWELL ON JULY 26, 1995**

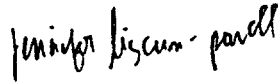
Intentionally Left Blank

Sandia National Laboratories

Albuquerque, New Mexico 87185-1175

date: May 13, 1996

to: FILE


from: Jennifer Liscum-Powell

re: Activity Factors for Mixed Fission Products and Mixed Activation Products in TWBIR Input Data Files for PreORG

The activity factors for TBGMFP (Mixed Fission Products) and TBGMAP (Mixed Activation Products) were obtained from Sayan Chakraborti of SAIC on July 26, 1995. Sayan is the TWBIR Technical Lead and a Senior Engineer with SAIC. These isotopic compositions are based on the Hanford generation site data.

The activity factors are used to translate activities given in generation sites for the isotope name TBGMAP and TBGMFP into actual isotopic compositions. The activity for an isotope is obtained by multiplying the activity of TBGMAP or TBGMFP, respectively, by the appropriate activity factor as given in Tables 1 and 2 below.

The activity factors are attached in the following tables. A copy of the original fax is also attached.

FILE

- 2 -

May 13, 1996

Table 1. Activity Factors for TBGMAP - Mixed Activation Products

ISOTOPE	ACTIVITY FACTOR
Ba137m	9.7000E-03
C14	1.0000E-02
Co58	1.0000E-02
Co60	6.7000E-01
Cs137	1.0300E-02
Eu152	1.0000E-02
Eu154	8.5000E-02
Eu155	8.5000E-02
Fe59	4.0000E-02
Mn54	7.0000E-02

FILE

- 3 -

May 13, 1996

Table 2. Activity Factors for TBGMFP - Mixed Fission Products

ISOTOPE	ACTIVITY FACTOR
Ag110	5.0200E-06
Ag110m	2.5070E-06
Ba137m	2.0090E-02
Cd113m	5.0000E-10
Ce144	2.7840E-01
Cs136	5.5570E-04
Cs135	5.7500E-06
Cs137	2.0090E-02
Eu152	3.0000E-07
Eu154	3.0380E-05
Eu155	7.3530E-04
Gd162	9.8620E-06
Nb95	1.0530E-01
Nb95m	1.0590E-03
Pd107	8.5000E-09
Pm147	7.0220E-02
Pr144	2.7840E-01
Rh106	4.0510E-02
Ru106	4.0510E-02
Sb125	7.9850E-04
Sb126	1.0940E-07
Sb126m	1.1050E-07
Se79	4.9900E-08
Sm151	1.9380E-04
Sn119m	4.3720E-05
Sn121m	3.9430E-06
Sn123m	1.8470E-05
Sn126	1.1050E-07
Sr90	1.8810E-02
Tb162	9.8620E-06
Tc99	2.8670E-06
Te125m	2.1760E-04
Te127	1.3540E-03
Te127m	1.3690E-03
Y90	1.8810E-02
Zr93	6.4550E-07
Zr95	4.9880E-02

TELECOPY TRANSMITTAL SHEET



An Employee-Owned Company

Science Applications International Corporation
2109 Air Park Road SE
Albuquerque, New Mexico 87106

Telecopy #: (505) 842-7798
Verification #: (505) 247-8787

Date: 7/26 Time: 9:30 am 5-Digit Access Code: _____

To: JENNIFER POWELL
(Name/Company)

848-0705
(Telecopy Number)

(Verification Number)

From: SAYAN CHAKRABORTI
(Name/Company)

842-7816
(Phone Number)

Additional Information: The ratios for the activation
products are given on the right side lower
corner of the attached page.

of Pages to Follow: 1

Page 28 of 31

1766 (approx) 13.89

EU-152	3.0000E-07
SE-79	4.9900E-08
2B-93	6.4330E-07
WB-95A	1.8590E-03
WB-95	1.8530E-01
	1.8420E-04

not listed add: n

EU-152	3.0000E-07
SE-79	4.9900E-08
2B-93	6.4330E-07
WB-95A	1.8590E-03
WB-95	1.8530E-01
	1.8420E-04

EU-152	3.000E-07
SE-79	4.9900E-08
2B-93	6.4330E-07
WB-95A	1.8590E-03
WB-95	1.8530E-01
	1.8420E-04

EU-152	3.0000E-07
SE-79	4.9900E-08
2B-93	6.4330E-07
WB-95A	1.8590E-03
WB-95	1.8530E-01
	1.8420E-04

EU-152	3.0000E-07
SE-79	4.9900E-08
2B-93	6.4330E-07
WB-95A	1.8590E-03
WB-95	1.8530E-01
	1.8420E-04

EU-152	3.0000E-07
SE-79	4.9900E-08
2B-93	6.4330E-07
WB-95A	1.8590E-03
WB-95	1.8530E-01
	1.8420E-04

EU-152	3.0000E-07
SE-79	4.9900E-08
2B-93	6.4330E-07
WB-95A	1.8590E-03
WB-95	1.8530E-01
	1.8420E-04

EU-152	3.000E-07
SE-79	4.9900E-08
2B-93	6.4330E-07
WB-95A	1.8590E-03
WB-95	1.8530E-01
	1.8420E-04

EU-152	3.0000E-07
SE-79	4.9900E-08
2B-93	6.4330E-07
WB-95A	1.8590E-03
WB-95	1.8530E-01
	1.8420E-04

EU-152	3.0000E-07
SE-79	4.9900E-08
2B-93	6.4330E-07
WB-95A	1.8590E-03
WB-95	1.8530E-01
	1.8420E-04

Intentionally Left Blank

APPENDIX F: OR3_POSTORG.FOR PROGRAM LISTING

Intentionally Left Blank

[illegible]

```

WRITE(6,5000)
WRITE(6,*)
WRITE(6,*)
WRITE(6,*)'WOULD YOU LIKE TO CREATE AN ORIGEN2 INPUT FILE'
WRITE(6,5005)fyr
5005 FORMAT(' TO DECAY ',a4,' INVENTORY DATA INTO FUTURE? ')
WRITE(6,*)' y(YES) or n(NO) '
WRITE(6,*)
READ 2010, answer
2010 FORMAT(a1)
WRITE(6,5000)
IF(answer.EQ.'y'.OR.answer.EQ.'Y')THEN
CALL FUTURE(time,tstep)
ENDIF
ENDIF
C
table = input(1:8)//'.ind'
C
C INQUIRE TO SEE IF DATA FILE EXISTS
C
INQUIRE(FILE=input, EXIST=exists)
IF(.NOT.exists)THEN
C
C ERROR MESSAGE - INPUT FILE NOT FOUND
C
WRITE(6,5010)input
5010 FORMAT(/,7('**ERROR'),/,/,t10,' INPUT FILE ',a12,
& ' DOES NOT EXIST ',/,/, 7('**ERROR'),/,/)
WRITE(6,*)' TO RE-ENTER FILENAME ENTER [1] OR '
WRITE(6,*)' TO EXIT PostORG ENTER [2]: '
WRITE(6,*)
READ*, ians
IF(ians.EQ.1)THEN
WRITE(6,*)' CHOOSE ANOTHER SITE: '
GO TO 1000
ELSEIF(ians.EQ.2)THEN
GO TO 9999
ENDIF
ENDIF
C
OPEN(UNIT=10,FILE=input,STATUS='OLD')
OPEN(UNIT=15,FILE=table,STATUS='NEW')
C
ccnt = 0
bcnt = 0
acnt = 0
ycnt = 0
lcnt = 0
nofile = 0
WRITE(6,*)
WRITE(6,*)' ELIMINATING EXCESS DATA FROM ORIGEN2 OUTPUT'
IF(sort.NE.'yes')then
WRITE(6,*)' CREATING OUTPUT DATAFILE FOR EACH YEAR '
ENDIF
WRITE(6,*)
1005 READ(10,2015,end=1020) line
2015 FORMAT(a133)
C
C
IF(line(13:14).EQ.'19'.OR.line(13:13).EQ.'2'.OR.
& line(13:14).EQ.'t=')THEN
IF(sort.EQ.'yes'.AND.acnt.EQ.0)THEN
acnt=1
OUT2 = INPUT(1:8)//'.tot'
OPEN(UNIT=20,FILE=OUT2,STATUS='NEW')
WRITE(20,*)' RADIOACTIVITY IN Curies '
WRITE(20,*)
WRITE(20,5015) line
5015 FORMAT(a133)
C
ELSEIF(acnt.NE.1)THEN
acnt=1
type = input(5:6)
close(unit=iout)
C
READ(line,2020) year, base

```

POST0153
 POST0154
 POST0155
 POST0156
 POST0157
 POST0158
 POST0159
 POST0160
 POST0161
 POST0162
 POST0163
 POST0164
 POST0165
 POST0166
 POST0167
 POST0168
 POST0169
 POST0170
 POST0171
 POST0172
 POST0173
 POST0174
 POST0175
 POST0176
 POST0177
 POST0178
 POST0179
 POST0180
 POST0181
 POST0182
 POST0183
 POST0184
 POST0185
 POST0186
 POST0187
 POST0188
 POST0189
 POST0190
 POST0191
 POST0192
 POST0193
 POST0194
 POST0195
 POST0196
 POST0197
 POST0198
 POST0199
 POST0200
 POST0201
 POST0202
 POST0203
 POST0204
 POST0205
 POST0206
 POST0207
 POST0208
 POST0209
 POST0210
 POST0211
 POST0212
 POST0213
 POST0214
 POST0215
 POST0216
 POST0217
 POST0218
 POST0219
 POST0220
 POST0221
 POST0222
 POST0223
 POST0224
 POST0225
 POST0226
 POST0227
 POST0228

Appendix F: OR3_PostORG.for Program Listing

```

2020      FORMAT(t13,a4,t23,a4)                                POST0229
c        WRITE(6,*) year                                     POST0230
C                                                POST0231
C                                                POST0232
C        nofile = nofile + 1                                POST0233
C        k = nofile                                          POST0234
C        icnt(k) = 0                                         POST0235
C        iout = 20 + nofile                                  POST0236
C        title = input(1:6)//year(3:4)//'.DAT'              POST0237
C        outfil(nofile) = title                             POST0238
C        WRITE(15,5020) outfil(nofile), year, fyr           POST0239
5020      FORMAT(t5,a12,t25,a4,' decay out to ',a4)          POST0240
C        OPEN(UNIT=IOUT,FILE=outfil(nofile),STATUS='NEW')   POST0241
C                                                        POST0242
C        WRITE(IOUT,*) ' RADIOACTIVITY IN Curies '          POST0243
C        WRITE(IOUT,*)                                       POST0244
C        WRITE(IOUT,5015) line                               POST0245
C                                                        POST0246
C        GO TO 1005                                           POST0247
C        ENDIF                                                POST0248
C        ENDIF                                                POST0249
C                                                        POST0250
C        IF(line(2:5).EQ.'TIME')GO TO 1005                   POST0251
C        IF(line(2:3).EQ.'AP')GO TO 1005                     POST0252
C        IF(line(2:4).EQ.'ACT')GO TO 1005                    POST0253
C        IF(line(2:6).EQ.'TOTAL')GO TO 1005                 POST0254
C        IF(line(2:6).EQ.'TABLE')GO TO 1005                 POST0255
C        IF(line(7:7).EQ.'E')GO TO 1005                     POST0256
C        IF(line(5:5).EQ.'L')GO TO 1005                     POST0257
C        IF(line(5:5).EQ.'O')GO TO 1005                     POST0258
C        IF(line(5:5).EQ.'F')GO TO 1005                     POST0259
C        IF(line(14:14).EQ.'C')GO TO 1005                   POST0260
C        IF(line(14:14).EQ.'M')GO TO 1005                   POST0261
C        IF(line(2:4).EQ.'THE')GO TO 1005                   POST0262
C        IF(line(1:1).EQ.'0')GO TO 1005                     POST0263
C        IF(line(30:38).EQ.'(ALPHA,N)')THEN                 POST0264
1010      READ(10,2015,end=1020) line2                       POST0265
C        IF(line2(23:37).EQ.'7 NUCLIDE TABLE')THEN        POST0266
C        acnt = 0                                             POST0267
C        GO TO 1005                                           POST0268
C        ENDIF                                                POST0269
C        GO TO 1010                                           POST0270
C        ENDIF                                                POST0271
C                                                        POST0272
C        IF(line(18:18).EQ.'E')THEN                          POST0273
C        IF(sort.EQ.'yes')THEN                                POST0274
C        IF(line(13:13).GT.'0'.OR.line(23:23).GT.'0')THEN   POST0275
C        lcnt = lcnt + 1                                       POST0276
C        READ(line,2025) MSLINE(lcnt)                         POST0277
2025      FORMAT(A133)                                         POST0278
C        ENDIF                                                POST0279
C        ELSEIF(line(23:23).GT.'0')THEN                     POST0280
C        i = nofile                                           POST0281
C        icnt(i) = icnt(i) + 1                                POST0282
C        occur = 0                                             POST0283
C        j = icnt(i)                                          POST0284
C                                                        POST0285
C        isotop(i,j) = line                                    POST0286
C        READ(line,2030) names(i,j),init(i,j),curies(i,j)    POST0287
2030      FORMAT(t2,a6,t12,e10.4,t22,e10.4)                 POST0288
C        WRITE(6,5025) names(i,j),init(i,j),curies(i,j)     POST0289
C                                                        POST0290
C        REMOVE BLANKS FROM ISOTOPE NAMES                   POST0291
C                                                        POST0292
C        CALL BLANKS(names(i,j),tname)                       POST0293
C        names(i,j) = tname                                    POST0294
C        write(iout,5025) names(i,j), init(i,j), curies(i,j) POST0295
5025      FORMAT(T2,A6,T12,E10.4,T23,E10.4)                 POST0296
C        ENDIF                                                POST0297
C        ENDIF                                                POST0298
C                                                        POST0299
C        GO TO 1005                                           POST0300
C                                                        POST0301
1020      IF(sort.EQ.'yes')THEN                                POST0302
C        WRITE(6,*)                                           POST0303
C        WRITE(6,*) '          SORTING DATA FROM ORIGEN2 OUTPUT' POST0304

```

	WRITE(6,*)' CREATING OUTPUT DATAFILE: ', out2	POST0305
	WRITE(6,*)	POST0306
	CALL ABCLIN(lcnt,msline,sorted)	POST0307
	DO 10 I=1,lcnt	POST0308
	WRITE(20,5015) msline(i)	POST0309
10	CONTINUE	POST0310
	CLOSE(UNIT=20)	POST0311
	GO TO 1025	POST0312
	ENDIF	POST0313
C		POST0314
	CLOSE(unit=10)	POST0315
	CLOSE(unit=15)	POST0316
	CLOSE(unit=16)	POST0317
	DO 25 i = 1, nofile	POST0318
	iout = 20 + i	POST0319
	CLOSE(unit=iout)	POST0320
25	CONTINUE	POST0321
C		POST0322
C	CREATE MASTER LIST OF NUCLIDES FOR GIVEN SITE	POST0323
C		POST0324
	WRITE(6,*)	POST0325
	WRITE(6,*)' GENERATING MASTER LIST OF NUCLIDES FOR SITE DATA'	POST0326
	WRITE(6,*)	POST0327
	CALL LIST(nofile,outfil,mnum,master)	POST0328
C		POST0329
C	ALPHABATIZE MASTER LIST OF NUCLIDES	POST0330
C		POST0331
	WRITE(6,*)	POST0332
	WRITE(6,*)' ALPHABETIZING MASTER LIST OF NUCLIDES '	POST0333
	WRITE(6,*)	POST0334
	CALL ABC(mnum,master)	POST0335
C		POST0336
C	TOTAL NUCLIDE INVENTORIES AT BASE YEAR	POST0337
C		POST0338
	WRITE(6,*)	POST0339
	WRITE(6,*)' SUMMING NUCLIDE CONCENTRATIONS '	POST0340
	WRITE(6,*)	POST0341
	CALL ADD(fyr,mnum,master,nofile,outfil,table,total)	POST0342
C		POST0343
C	SORT NUCLIDE INVENTORIES AT BASE YEAR INTO WIPP BIR REV.1 Appendix E	POST0344
C	NUCLIDE LIST	POST0345
C		POST0346
	BIDINP = INPUT(1:8)//'.SUM'	POST0347
	WRITE(6,*)	POST0348
	WRITE(6,*)' SORTING INTO WIPP BIR REV.1 Appendix E NUCLIDE LIST '	POST0349
	WRITE(6,*)' INPUT FILENAME = ', BIDINP	POST0350
	WRITE(6,*)	POST0351
	CALL WIPP(BIDINP)	POST0352
C		POST0353
C	CREATE ORIGEN2 INPUT FILE TO DECAY INVENTORY INTO FUTURE	POST0354
C		POST0355
	IF(answer.EQ.'y'.OR.answer.EQ.'Y')THEN	POST0356
	WRITE(6,*)	POST0357
	WRITE(6,*)' GENERATING ORIGEN2 INPUT FILE TO DECAY BASE YEAR'	POST0358
	WRITE(6,*)' DATA INTO FUTURE'	POST0359
	WRITE(6,*)	POST0360
	ifut = input(1:8)//'.sum'	POST0361
	CALL FINPUT(time,tstep,ifut)	POST0362
	ENDIF	POST0363
C		POST0364
1025	WRITE(6,5000)	POST0365
	WRITE(6,*)' TO RUN PROGRAM AGAIN ENTER 1'	POST0366
	WRITE(6,*)' TO EXIT PROGRAM ENTER 2: '	POST0367
	WRITE(6,5000)	POST0368
	READ*, irun	POST0369
	IF(irun.EQ.1)GO TO 1000	POST0370
C		POST0371
9999	STOP	POST0372
	END	POST0373
CC		POST0374
	SUBROUTINE FUTURE(time,tstep)	POST0375
C		POST0376
C	jennifer liscum-powell - 9364 1/4/96	POST0377
CC		POST0378
C	This subroutine prompts user for time he/she wishes to decay	CPOST0379
C	inventory out into the future. It also determines the ORIGEN2	CPOST0380

Appendix F: OR3_PostORG.for Program Listing

```

C input commands needed in the ORIGEN2 input file to perform the      CPOST0381
C desired calculation                                                  CPOST0382
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC CPOST0383
      IMPLICIT REAL(a-h,o-z), INTEGER(i-n)                            POST0384
      CHARACTER choice*1                                              POST0385
      REAL time, tstep                                                POST0386
C                                                                       POST0387
C PROMPT USER FOR DESIRED TIME TO DECAY INVENTORY INTO FUTURE        POST0388
C                                                                       POST0389
      WRITE(6,*) ' PLEASE CHOOSE HOW LONG INTO FUTURE YOU WISH '    POST0390
      WRITE(6,*) ' TO DECAY INVENTORY OUT TO: '                      POST0391
      WRITE(6,*) ' (a) 75 yrs.      (b) 100 yrs.      (c) 500 yrs.    POST0392
      WRITE(6,*) ' (d) 1000 yrs..   (e) 5000 yrs.    (f) 10,000 yrs.  POST0393
      WRITE(6,*) '                                                         POST0394
      READ 2000, choice                                              POST0395
2000 FORMAT(a1)                                                    POST0396
C                                                                       POST0397
C THE ORIGEN2 INPUT FILE CREATED TO DECAY INVENTORY INTO THE FUTURE  POST0398
C WILL BE NAMED SUCH THAT THE LAST LETTER OF THE FILENAME INDICATES POST0399
C HOW FAR INTO THE FUTURE THE BASE YEAR DATA HAS BEEN DECAYED. FOR POST0400
C EXAMPLE, ornlc95a.u5 IS THE NAME FOR ORNL Contact handled waste   POST0401
C 1995 INVENTORY TO BE DECAYED OUT 75 YEARS.                        POST0402
C                                                                       POST0403
      IF(choice.EQ.'a'.OR.choice.EQ.'A')THEN                          POST0404
        time = 75.0                                                  POST0405
        tstep = 7.50                                                POST0406
      ELSEIF(choice.EQ.'b'.OR.choice.EQ.'B')THEN                     POST0407
        time = 100.0                                                POST0408
        tstep = 10.0                                                POST0409
      ELSEIF(choice.EQ.'c'.OR.choice.EQ.'C')THEN                     POST0410
        time = 500.0                                                POST0411
        tstep = 50.0                                                POST0412
      ELSEIF(choice.EQ.'d'.OR.choice.EQ.'D')THEN                     POST0413
        time = 1000.0                                               POST0414
        tstep = 100.0                                               POST0415
      ELSEIF(choice.EQ.'e'.OR.choice.EQ.'E')THEN                     POST0416
        time = 5000.0                                               POST0417
        tstep = 500.0                                               POST0418
      ELSEIF(choice.EQ.'f'.OR.choice.EQ.'F')THEN                     POST0419
        time = 10000.0                                              POST0420
        tstep = 1000.0                                              POST0421
      ENDIF                                                         POST0422
C                                                                       POST0423
      RETURN                                                         POST0424
      END                                                            POST0425
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC POST0426
      SUBROUTINE BLANKS(names,tname)                                  POST0427
C                                                                       POST0428
C      jennifer liscum-powell - 9364 11/30/95                        POST0429
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC POST0430
C This subroutine removes all blanks from isotope names as          CPOST0431
C written in ORIGEN2 output in order to use subroutine CONV         CPOST0432
C to convert from curies in ORIGEN2 output file to grams and        CPOST0433
C generate a new ORIGEN2 input file.                                CPOST0434
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC POST0435
      IMPLICIT REAL(a-h,o-z), INTEGER(i-n)                            POST0436
      CHARACTER let(6)*1, names*6, tname*6, pos(6)*1, lpos*6        POST0437
      INTEGER blank, cnt, space, sp, k                               POST0438
C                                                                       POST0439
      tname = ' '                                                    POST0440
      do 10 j=1,6                                                    POST0441
        let(j) = names(j:j)                                          POST0442
      continue                                                       POST0443
C                                                                       POST0444
      sp = 0                                                         POST0445
      space = 0                                                       POST0446
      ncnt = 0                                                        POST0447
      cnt = 0                                                         POST0448
      blank = 0                                                       POST0449
      do 15 j =1,6                                                    POST0450
        if(let(j).eq.' ')blank=blank+1                              POST0451
      continue                                                       POST0452
15 C                                                                    POST0453
      do 20 j=1,6                                                    POST0454
        if(let(j).eq.' ')then                                       POST0455
          pos(j)='- '                                               POST0456
        
```

```

else
    pos(j)='+'
endif
continue
C      do 25 j=1,6
        if(pos(j).eq.'+')then
            cnt = cnt + 1
            lpos(cnt:cnt)=let(j)
        endif
25     continue
        space=6-blank
        sp = space+1
        do 30 k=sp,6
            lpos(k:k)' '
30     continue
C      tname=lpos
C
        return
end
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0478
SUBROUTINE LIST(nofile,outfil,mnum,master)          POST0479
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0480
C   jennifer lisum-powell 9364 1/02/96              POST0481
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0482
C This subroutine creates a master list of nuclides for each site CPOST0483
C inventory.                                          CPOST0484
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0485
IMPLICIT real(a-h,o-z), integer(i-n)               POST0486
INTEGER iout,nofile,mnum,icnt,count                POST0487
CHARACTER outf(50)*15,mast(1000)*6,name(1000)*6,line*133 POST0488
C OPEN DATA FILE WITH EARLIEST SITE INVENTORY       POST0490
C                                                     POST0491
OPEN(unit=21,file=outf(1),status='old')             POST0492
OPEN(unit=10,file='LIST',status='new')              POST0493
ICNT = 0                                             POST0494
C CREATE INITIAL LIST OF NUCLIDES                   POST0495
C                                                     POST0496
MSNUM = 0                                           POST0497
1000 READ(21,2000,end=1005) LINE                    POST0498
2000 FORMAT(A133)                                    POST0500
IF(LINE(2:2).NE.' ')THEN                            POST0501
    MSNUM=MSNUM+1                                     POST0502
    READ(LINE,2005)MASTER(msnum)                     POST0503
2005 FORMAT(T2,A6)                                   POST0504
ENDIF                                                 POST0505
GO TO 1000                                            POST0506
C                                                     POST0507
1005 DO 20 I=2,NOFIL                                POST0508
    ICNT = ICNT+1                                      POST0509
    IUNIT = 20+I                                       POST0510
    OPEN(unit=iunit,file=outf(I),status='old')         POST0511
1010 READ(IUNIT,2000,end=1015)LINE                  POST0512
    IF(LINE(2:2).NE.' ')THEN                          POST0513
        ICNT = ICNT+1                                 POST0514
        READ(LINE,2005) NAMES(icnt)                   POST0515
    ENDIF                                              POST0516
    GO TO 1010                                         POST0517
C                                                     POST0518
C CREATE MASTER LIST OF NUCLIDES                      POST0519
C                                                     POST0520
1015 DO 15 J=1,ICNT                                  POST0521
    DO 10 K=1,MNUM                                    POST0522
        IF(NAMES(J).EQ.MASTER(K))THEN                 POST0523
            COUNT=1                                     POST0524
            GO TO 1020                                 POST0525
        ENDIF                                          POST0526
10 CONTINUE                                           POST0527
1020 IF(COUNT.NE.1)THEN                               POST0528
    MNUM = MNUM+1                                       POST0529
    MASTER(MNUM) = NAMES(J)                             POST0530
ENDIF                                                  POST0531
COUNT = 0                                             POST0532
```

```

15      CONTINUE                                POST0533
C                                          POST0534
        ICNT = 0                               POST0535
20     CONTINUE                                POST0536
C                                          POST0537
        DO 22 I=1, MSNUM                       POST0538
            WRITE(10,5000) MASTER(I)           POST0539
5000    FORMAT(T2,A6)                         POST0540
22     CONTINUE                                POST0541
C                                          POST0542
        CLOSE(UNIT=10)                        POST0543
        DO 25 i = 1, nofile                    POST0544
            iout = 20 + i                     POST0545
            CLOSE(unit=iout)                  POST0546
25     CONTINUE                                POST0547
C                                          POST0548
        RETURN                                 POST0549
        END                                     POST0550
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0551
        SUBROUTINE ABC(msnum,master)          POST0552
C                                          POST0553
C       jennifer liscum-powell - 9364 12/18/95 POST0554
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0555
C This subroutine alphabetizes the master list of nuclides prior CPOST0556
C to calculating totals at end of base year.   CPOST0557
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0558
        IMPLICIT real(a-h,o-z), integer(i-n) POST0559
        INTEGER msnum                          POST0560
        CHARACTER master(1000)*6, TEMP*6      POST0561
C                                          POST0562
        do 15 j=1, msnum-1                   POST0563
C                                          POST0564
        do 10 i=1, msnum-1                   POST0565
            if(master(i).ge.master(i+1))then  POST0566
                TEMP = master(i)              POST0567
                master(i) = master(i+1)       POST0568
                master(i+1) = TEMP             POST0569
            endif                             POST0570
10      continue                             POST0571
C                                          POST0572
15      continue                             POST0573
C                                          POST0574
        return                                POST0575
        end                                    POST0576
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0577
        SUBROUTINE ABCLIN(lcnt,msline,sorted) POST0578
C                                          POST0579
C       jennifer liscum-powell - 9364 1/04/96 POST0580
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0581
C This subroutine alphabetizes the master list of nuclides tracking POST0582
C entire lines of data containing inventory decayed into the future. CPOST0583
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0584
        IMPLICIT real(a-h,o-z), integer(i-n) POST0585
        INTEGER lcnt                           POST0586
        CHARACTER mslines(1000)*133, sorted(1000)*133, TEMP*133, POST0587
        & name(1000)*6, tname*6               POST0588
C                                          POST0589
        DO 20 j=1, lcmt-1                    POST0590
C                                          POST0591
        DO 15 i=1, lcmt-1                    POST0592
            READ(mslines(i),2000) name(i)      POST0593
2000    FORMAT(T2,A6)                         POST0594
            CALL BLANKS(name(i),tname)         POST0595
            name(i) = tname                    POST0596
C                                          POST0597
            READ(mslines(i+1),2000) name(i+1) POST0598
            CALL BLANKS(name(i+1),tname)       POST0599
            name(i+1) = tname                 POST0600
            IF(name(i).ge.name(i+1))THEN      POST0601
                write(6,*) name(i), ' & ', name(i+1) POST0602
                TEMP = mslines(i)             POST0603
                mslines(i) = mslines(i+1)     POST0604
                mslines(i+1) = TEMP           POST0605
            endif                             POST0606
15      continue                             POST0607
C                                          POST0608

```

[illegible]

```

        iunit=20+1
CLOSE(UNIT=iunit)
35    CONTINUE
C
RETURN
END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0685
SUBROUTINE WIPP(BIDINP)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0686
jennifer liscum-powell 9364 1/04/96
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0687
This subroutine can be used to sort inventories in *.sum files
C generated by postORG for to include those listed in Appendix E
C of the WIPP Baseline Inventory Report revision 1 (February 1995).
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPOST0688
IMPLICIT REAL(a-h,o-z), INTEGER(i-n)
INTEGER icnt
REAL sumed(500), curies(500)
CHARACTER name(500)*6, iname(500)*6,type*2,input*15,
& output*15, line*133, BIDINP*15
C
C OPEN DATA FILES
input = BIDINP
output= input(1:8)//'.bid'
WRITE(6,*)' WIPP BIR Appendix E Nuclide List File: ', output
C
OPEN(UNIT=10,FILE=input,STATUS='OLD')
OPEN(UNIT=15,FILE=output,STATUS='NEW')
C
WRITE(6,*) INPUT(5:5)
IF(input(5:5).EQ.'c'.OR.input(5:5).eq.'C')THEN
    type='ch'
ELSEIF(input(5:5).EQ.'b'.OR.input(5:5).eq.'B')THEN
    type='bu'
ELSEIF(input(5:5).EQ.'r'.OR.input(5:5).eq.'R')THEN
    type='rh'
ENDIF
WRITE(6,*) TYPE
C
READ NAMES AND CONCENTRATIONS FROM INPUT FILE
iskip = 0
icnt = 0
1000 READ(10,2005,END=1010) line
2005 FORMAT(a80)
iskip = iskip + 1
IF(iskip.GT.1)THEN
    icnt = icnt + 1
    READ(line,2010) iname(icnt), curies(icnt)
2010 FORMAT(T2,A6,T22,E10.4)
C WRITE(6,2010) INAME(ICNT), CURIES(ICNT)
ENDIF
GO TO 1000
C
CALL NAMES(type,name)
C
IF(type.EQ.'ch'.OR.type.EQ.'bu')THEN
DO 10 I = 1, 136
SUMED(I) = 0
10 CONTINUE
C
DO 20 j = 1, icnt
DO 15 k = 1, 136
IF(iname(j).EQ.name(k))THEN
summed(k) = curies(j)
15 CONTINUE
20 CONTINUE
WRITE TO OUTPUT FILE
C
DO 25 j = 1,136
WRITE(15,5005) NAME(j), SUMED(j)

```

```

5005          FORMAT(T2,A6,T22,E10.4)
25      CONTINUE
C
ELSEIF(type.EQ.'rh')THEN
C
DO 30 i = 1, 122
    SUMED(i) = 0
30      CONTINUE
C
DO 40 j = 1, icnt
    DO 35 k = 1, 122
        IF(iname(j).EQ.name(k))THEN
            sumed(k) = curies(j)
        ENDIF
35      CONTINUE
40      CONTINUE
C
WRITE TO OUTPUT FILE
C
DO 45 j = 1,122
    WRITE(15,5005) NAME(j), SUMED(j)
45      CONTINUE
C
ENDIF
C
9999 CLOSE(UNIT=10)
CLOSE(UNIT=15)
C
STOP
C
RETURN
END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE NAMES(type,name)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C jennifer liscum-powell 1/17/96
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C This subroutine list nuclides found in Appendix E of the WIPP Baselin
C Inventory Report Revision 1.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CHARACTER type*2, name(150)*6
C
IF(type.EQ.'ch'.OR.type.EQ.'bu')THEN
NAME(1)   = 'AC225 '
NAME(2)   = 'AC227 '
NAME(3)   = 'AC228 '
NAME(4)   = 'AG109M'
NAME(5)   = 'AG110 '
NAME(6)   = 'AG110M'
NAME(7)   = 'AM241 '
NAME(8)   = 'AM242 '
NAME(9)   = 'AM242M'
NAME(10)  = 'AM243 '
NAME(11)  = 'AM245 '
NAME(12)  = 'AT217 '
NAME(13)  = 'BA137M'
NAME(14)  = 'BI210 '
NAME(15)  = 'BI211 '
NAME(16)  = 'BI212 '
NAME(17)  = 'BI213 '
NAME(18)  = 'BI214 '
NAME(19)  = 'BK249 '
NAME(20)  = 'BK250 '
NAME(21)  = 'C14 '
NAME(22)  = 'CD109 '
NAME(23)  = 'CD113M'
NAME(24)  = 'CE144 '
NAME(25)  = 'CF249 '
NAME(26)  = 'CF250 '
NAME(27)  = 'CF251 '
NAME(28)  = 'CF252 '
NAME(29)  = 'CM242 '
NAME(30)  = 'CM243 '
NAME(31)  = 'CM244 '
NAME(32)  = 'CM245 '
NAME(33)  = 'CM246 '
NAME(34)  = 'CM247 '

```

Appendix F: OR3_PostORG.for Program Listing

NAME(35) = 'CM248'	POST0837
NAME(36) = 'CO58'	POST0838
NAME(37) = 'CO60'	POST0839
NAME(38) = 'CS134'	POST0840
NAME(39) = 'CS135'	POST0841
NAME(40) = 'CS137'	POST0842
NAME(41) = 'ES254'	POST0843
NAME(42) = 'EU150'	POST0844
NAME(43) = 'EU152'	POST0845
NAME(44) = 'EU154'	POST0846
NAME(45) = 'EU155'	POST0847
NAME(46) = 'FE55'	POST0848
NAME(47) = 'FE59'	POST0849
NAME(48) = 'FR221'	POST0850
NAME(49) = 'FR223'	POST0851
NAME(50) = 'H3'	POST0852
NAME(51) = 'I129'	POST0853
NAME(52) = 'KR85'	POST0854
NAME(53) = 'MN54'	POST0855
NAME(54) = 'NB95'	POST0856
NAME(55) = 'NB95M'	POST0857
NAME(56) = 'NI59'	POST0858
NAME(57) = 'NI63'	POST0859
NAME(58) = 'NP237'	POST0860
NAME(59) = 'NP238'	POST0861
NAME(60) = 'NP239'	POST0862
NAME(61) = 'NP240'	POST0863
NAME(62) = 'NP240M'	POST0864
NAME(63) = 'PA231'	POST0865
NAME(64) = 'PA233'	POST0866
NAME(65) = 'PA234'	POST0867
NAME(66) = 'PA234M'	POST0868
NAME(67) = 'PB209'	POST0869
NAME(68) = 'PB210'	POST0870
NAME(69) = 'PB211'	POST0871
NAME(70) = 'PB212'	POST0872
NAME(71) = 'PB214'	POST0873
NAME(72) = 'PD107'	POST0874
NAME(73) = 'PM147'	POST0875
NAME(74) = 'PO210'	POST0876
NAME(75) = 'PO211'	POST0877
NAME(76) = 'PO212'	POST0878
NAME(77) = 'PO213'	POST0879
NAME(78) = 'PO214'	POST0880
NAME(79) = 'PO215'	POST0881
NAME(80) = 'PO216'	POST0882
NAME(81) = 'PO218'	POST0883
NAME(82) = 'PR144'	POST0884
NAME(83) = 'PU236'	POST0885
NAME(84) = 'PU238'	POST0886
NAME(85) = 'PU239'	POST0887
NAME(86) = 'PU240'	POST0888
NAME(87) = 'PU241'	POST0889
NAME(88) = 'PU242'	POST0890
NAME(89) = 'PU243'	POST0891
NAME(90) = 'PU244'	POST0892
NAME(91) = 'RA223'	POST0893
NAME(92) = 'RA224'	POST0894
NAME(93) = 'RA225'	POST0895
NAME(94) = 'RA226'	POST0896
NAME(95) = 'RA228'	POST0897
NAME(96) = 'RH106'	POST0898
NAME(97) = 'RN219'	POST0899
NAME(98) = 'RN220'	POST0900
NAME(99) = 'RN222'	POST0901
NAME(100) = 'RU106'	POST0902
NAME(101) = 'SB125'	POST0903
NAME(102) = 'SB126'	POST0904
NAME(103) = 'SB126M'	POST0905
NAME(104) = 'SE79'	POST0906
NAME(105) = 'SM151'	POST0907
NAME(106) = 'SN119M'	POST0908
NAME(107) = 'SN121M'	POST0909
NAME(108) = 'SN126'	POST0910
NAME(109) = 'SR90'	POST0911
NAME(110) = 'TC99'	POST0912

Appendix F: OR3_PostORG.for Program Listing

```

NAME(111) = 'TE125M'
NAME(112) = 'TE127 '
NAME(113) = 'TE127M'
NAME(114) = 'TH227 '
NAME(115) = 'TH228 '
NAME(116) = 'TH229 '
NAME(117) = 'TH230 '
NAME(118) = 'TH231 '
NAME(119) = 'TH232 '
NAME(120) = 'TH234 '
NAME(121) = 'TL207 '
NAME(122) = 'TL208 '
NAME(123) = 'TL209 '
NAME(124) = 'TL210 '
NAME(125) = 'U232 '
NAME(126) = 'U233 '
NAME(127) = 'U234 '
NAME(128) = 'U235 '
NAME(129) = 'U236 '
NAME(130) = 'U237 '
NAME(131) = 'U238 '
NAME(132) = 'U240 '
NAME(133) = 'Y90 '
NAME(134) = 'ZN95 '
NAME(135) = 'ZR93 '
NAME(136) = 'ZR95 '
ELSEIF (type.EQ.'rh') THEN
NAME(1) = 'AC225 '
NAME(2) = 'AC227 '
NAME(3) = 'AC228 '
NAME(4) = 'AG110 '
NAME(5) = 'AG110M'
NAME(6) = 'AM241 '
NAME(7) = 'AM243 '
NAME(8) = 'AM245 '
NAME(9) = 'AT217 '
NAME(10) = 'BA137M'
NAME(11) = 'BI210 '
NAME(12) = 'BI211 '
NAME(13) = 'BI212 '
NAME(14) = 'BI213 '
NAME(15) = 'BI214 '
NAME(16) = 'BK249 '
NAME(17) = 'C14 '
NAME(18) = 'CD113M'
NAME(19) = 'CE144 '
NAME(20) = 'CF249 '
NAME(21) = 'CF250 '
NAME(22) = 'CF252 '
NAME(23) = 'CM243 '
NAME(24) = 'CM244 '
NAME(25) = 'CM245 '
NAME(26) = 'CM246 '
NAME(27) = 'CM248 '
NAME(28) = 'CO58 '
NAME(29) = 'CO60 '
NAME(30) = 'CR51 '
NAME(31) = 'CS134 '
NAME(32) = 'CS135 '
NAME(33) = 'CS137 '
NAME(34) = 'EU152 '
NAME(35) = 'EU154 '
NAME(36) = 'EU155 '
NAME(37) = 'FE55 '
NAME(38) = 'FE59 '
NAME(39) = 'FR221 '
NAME(40) = 'FR223 '
NAME(41) = 'H3 '
NAME(42) = 'KR85 '
NAME(43) = 'MN54 '
NAME(44) = 'NB95 '
NAME(45) = 'NB95M '
NAME(46) = 'NI63 '
NAME(47) = 'NP237 '
NAME(48) = 'NP239 '
NAME(49) = 'NP240 '
POST0913
POST0914
POST0915
POST0916
POST0917
POST0918
POST0919
POST0920
POST0921
POST0922
POST0923
POST0924
POST0925
POST0926
POST0927
POST0928
POST0929
POST0930
POST0931
POST0932
POST0933
POST0934
POST0935
POST0936
POST0937
POST0938
POST0939
POST0940
POST0941
POST0942
POST0943
POST0944
POST0945
POST0946
POST0947
POST0948
POST0949
POST0950
POST0951
POST0952
POST0953
POST0954
POST0955
POST0956
POST0957
POST0958
POST0959
POST0960
POST0961
POST0962
POST0963
POST0964
POST0965
POST0966
POST0967
POST0968
POST0969
POST0970
POST0971
POST0972
POST0973
POST0974
POST0975
POST0976
POST0977
POST0978
POST0979
POST0980
POST0981
POST0982
POST0983
POST0984
POST0985
POST0986
POST0987
POST0988

```

Appendix F: OR3_PostORG.for Program Listing

```

NAME(50) = 'NP240M' POST0989
NAME(51) = 'PA231' POST0990
NAME(52) = 'PA233' POST0991
NAME(53) = 'PA234' POST0992
NAME(54) = 'PA234M' POST0993
NAME(55) = 'PB209' POST0994
NAME(56) = 'PB210' POST0995
NAME(57) = 'PB211' POST0996
NAME(58) = 'PB212' POST0997
NAME(59) = 'PB214' POST0998
NAME(60) = 'PD107' POST0999
NAME(61) = 'PM147' POST1000
NAME(62) = 'PO210' POST1001
NAME(63) = 'PO211' POST1002
NAME(64) = 'PO212' POST1003
NAME(65) = 'PO213' POST1004
NAME(66) = 'PO214' POST1005
NAME(67) = 'PO215' POST1006
NAME(68) = 'PO216' POST1007
NAME(69) = 'PO218' POST1008
NAME(70) = 'PR144' POST1009
NAME(71) = 'PU238' POST1010
NAME(72) = 'PU239' POST1011
NAME(73) = 'PU240' POST1012
NAME(74) = 'PU241' POST1013
NAME(75) = 'PU242' POST1014
NAME(76) = 'PU244' POST1015
NAME(77) = 'RA223' POST1016
NAME(78) = 'RA224' POST1017
NAME(79) = 'RA225' POST1018
NAME(80) = 'RA226' POST1019
NAME(81) = 'RA228' POST1020
NAME(82) = 'RH106' POST1021
NAME(83) = 'RN219' POST1022
NAME(84) = 'RN220' POST1023
NAME(85) = 'RN222' POST1024
NAME(86) = 'RU106' POST1025
NAME(87) = 'SB125' POST1026
NAME(88) = 'SB126' POST1027
NAME(89) = 'SB126M' POST1028
NAME(90) = 'SE79' POST1029
NAME(91) = 'SM151' POST1030
NAME(92) = 'SN119M' POST1031
NAME(93) = 'SN121M' POST1032
NAME(94) = 'SN126' POST1033
NAME(95) = 'SR90' POST1034
NAME(96) = 'TA182' POST1035
NAME(97) = 'TC99' POST1036
NAME(98) = 'TE125M' POST1037
NAME(99) = 'TE127' POST1038
NAME(100) = 'TE127M' POST1039
NAME(101) = 'TH227' POST1040
NAME(102) = 'TH228' POST1041
NAME(103) = 'TH229' POST1042
NAME(104) = 'TH230' POST1043
NAME(105) = 'TH231' POST1044
NAME(106) = 'TH232' POST1045
NAME(107) = 'TH234' POST1046
NAME(108) = 'TL207' POST1047
NAME(109) = 'TL208' POST1048
NAME(110) = 'TL209' POST1049
NAME(111) = 'TL210' POST1050
NAME(112) = 'U232' POST1051
NAME(113) = 'U233' POST1052
NAME(114) = 'U234' POST1053
NAME(115) = 'U235' POST1054
NAME(116) = 'U236' POST1055
NAME(117) = 'U237' POST1056
NAME(118) = 'U238' POST1057
NAME(119) = 'U240' POST1058
NAME(120) = 'Y90' POST1059
NAME(121) = 'ZR93' POST1060
NAME(122) = 'ZR95' POST1061
ENDIF POST1062
RETURN POST1063
POST1064

```

	END	POST1065
C	CC	POST1066
	SUBROUTINE FINPUT(time,tstep,input)	POST1067
C	CC	POST1068
C	jennifer liscum-powell 9364 1/04/96	POST1069
C	CC	POST1070
C	This subroutine generates an ORIGEN2 input file to decay inventory	POST1071
C	C into the future as specified by the user.	POST1072
C	CC	POST1073
	IMPLICIT REAL(a-h,o-z), INTEGER(i-n)	POST1074
	REAL curies(100,500), grams(1,500), time, tstep	POST1075
	INTEGER ycnt, isonum,varl(500), var2(500), var3(500),	POST1076
&	lib(100,500), yr, out, hed(0:11), dectim	POST1077
	CHARACTER input*15, outfll*15, line*133, choice*1,	POST1078
&	nucnam(100,500)*6, nucid(100,500)*6,fyr*4	POST1079
C	IF(tstep.EQ.7.50)THEN	POST1080
	choice = 'a'	POST1081
	ELSEIF(tstep.EQ.10.0)THEN	POST1082
	choice = 'b'	POST1083
	ELSEIF(tstep.EQ.50.0)THEN	POST1084
	choice = 'c'	POST1085
	ELSEIF(tstep.EQ.100.0)THEN	POST1086
	choice = 'd'	POST1087
	ELSEIF(tstep.EQ.500.0)THEN	POST1088
	choice = 'e'	POST1089
	ELSEIF(tstep.EQ.1000)THEN	POST1090
	choice = 'f'	POST1091
	ENDIF	POST1092
	outfil = input(1:5)//input(7:8)//choice//'.u5'	POST1093
	WRITE(6,*)' ORIGEN2 input filename = ', outfll	POST1094
C	OPEN INPUT AND OUTPUT FILES	POST1095
C	OPEN(UNIT=12,FILE=input,STATUS='OLD')	POST1096
C	OPEN(UNIT=15,FILE=outfil,STATUS='NEW')	POST1097
C	READ INPUT DATA FROM FILE	POST1098
C	yCnt = 0	POST1099
	iCnt = 1	POST1100
	istep = 0	POST1101
C	1000 READ(12,2000,END=1005) line	POST1102
2000	FORMAT(al33)	POST1103
	istep= istep+1	POST1104
C	IF(line(2:3).EQ.'19'.OR.line(2:2).EQ.'2')THEN	POST1105
	ycnt = ycnt + 1	POST1106
	isonum = 0	POST1107
2005	READ(LINE,2005) yr	POST1108
	FORMAT(t2,i4)	POST1109
	GO TO 1000	POST1110
	ENDIF	POST1111
C	READ ISOTOPES AND ACTIVITIES IN INVENTORY FOR PRESENT YEAR AND SITE	POST1112
C	IF(istep.GT.2)THEN	POST1113
	isonum = isonum + 1	POST1114
	READ(line,2010) nucnam(1,isonum), curies(1,isonum)	POST1115
2010	FORMAT(T2,A6,T22,e10.4)	POST1116
	IF(curies(1,isonum).EQ.0)THEN	POST1117
	isonum = isonum - 1	POST1118
	ENDIF	POST1119
	GO TO 1000	POST1120
C	ENDIF	POST1121
C	1005 out = 15	POST1122
C	WRITE TO ORIGIN2 PC INPUT FILE	POST1123
C	WRITE(out,5000)	POST1124
5000	FORMAT(2(2x,'-1',/),2x,'-1')	POST1125
C	WRITE(out,5005) time	POST1126
		POST1127
		POST1128
		POST1129
		POST1130
		POST1131
		POST1132
		POST1133
		POST1134
		POST1135
		POST1136
		POST1137
		POST1138
		POST1139
		POST1140

Appendix F: OR3_PostORG.for Program Listing

```

5005  FORMAT(2x,'BAS',4x,'WIPP BASELINE INVENTORY DATA DECAY OUT ',
      & f7.1,' Yrs')
      WRITE(out,5010) input(1:8)
5010  FORMAT(2x,'TIT',4x,a8)
      WRITE(out,5015)
5015  FORMAT(2x,'LIP',4x,3('0',2x))
      WRITE(out,5020)
5020  FORMAT(2x,'LIB',3x,'0',2x,'1',2x,'2',2x,'3',2x,3('0',2x),
      + '9',2x,'3',2x,'0',2x,'1',2x,'1')
      WRITE(out,5025)
5025  FORMAT(2x,'PHO',4x,'101',2x,'102',2x,'103',2x,'10')
C
      j = 1
      WRITE(out,5030) j
5030  FORMAT(2x,'INP',4x,i4,2x,'1',2x,2('1',2x),2('1',2x))
C
C
      WRITE(out,5035) yr
5035  FORMAT(2x,'RDA',4x,'BASE YEAR IS ',i4,' (T=0)')
      WRITE(out,5040)
5040  FORMAT(2x,'RDA',4x,'ALL OTHER HEADINGS ARE IN YEARS')
      dectim = 0
      hed(0) = 0
      DO 10 i = 1, 10
        j = i + 1
        dectim = dectim + tstep
        hed(i) = dectim
        IF(i.EQ.1)THEN
          WRITE(out,5045) dectim, i, j
          FORMAT(2x,'DEC',4x,i7,2x,i4,2x,i4,1x,
          + '5',1x,'1')
        ELSE
          WRITE(out,5050) dectim, i, j
          FORMAT(2x,'DEC',4x,i7,2x,i4,2x,i4,1x,
          + '5',1x,'0')
        ENDIF
10    CONTINUE
C
C  WRITE TITLES FOR OUTPUT VECTORS
C
      WRITE(out,5055)
5055  FORMAT(2x,'RDA',4x,'ASSIGN TITLES TO OUTPUT VECTORS')
      DO 15 i=1,11
        j = i - 1
        IF(j.EQ.0)THEN
          WRITE(out,5060) i, hed(j)
          FORMAT(2x,'HED',4x,i4,2x,'t= ',i5)
        ELSEIF(j.GT.0)THEN
          WRITE(out,5065) i, hed(j)
          FORMAT(2x,'HED',4x,i4,2x,i5,1x,'y')
        ENDIF
15    CONTINUE
C
C  WRITE OUTPUT COMMANDS
C
      WRITE(out,5070)
      WRITE(out,5075)
      WRITE(out,5080)
      WRITE(out,5085)
      WRITE(out,5090)
      WRITE(out,5095)
5070  FORMAT(2x,'RDA',4x,'PRINT NUCLIDE QUANTITIES IN CURIES')
5075  FORMAT(2x,'OPTA',3x,'6*8',2x,'5',2x,'17*8')
5080  FORMAT(2x,'OPTF',3x,'6*8',2x,'5',2x,'17*8')
5085  FORMAT(2x,'OPTL',3x,'6*8',2x,'5',2x,'17*8')
5090  FORMAT(2x,'RDA',4x,'OUTPUT COMMANDS')
5095  FORMAT(2x,'OUT',4x,'11',2x,'1',2x,'-1',2x,'0')
C
      WRITE(out,5100)
5100  FORMAT(2x,'END')
C
C  WRITE INITIAL NUCLIDE INVENTORIES TO ORIGEN2 INPUT FILE
C
      num = isonum
      lib1 = 0
      lib2 = 0

```

POST1141
 POST1142
 POST1143
 POST1144
 POST1145
 POST1146
 POST1147
 POST1148
 POST1149
 POST1150
 POST1151
 POST1152
 POST1153
 POST1154
 POST1155
 POST1156
 POST1157
 POST1158
 POST1159
 POST1160
 POST1161
 POST1162
 POST1163
 POST1164
 POST1165
 POST1166
 POST1167
 POST1168
 POST1169
 POST1170
 POST1171
 POST1172
 POST1173
 POST1174
 POST1175
 POST1176
 POST1177
 POST1178
 POST1179
 POST1180
 POST1181
 POST1182
 POST1183
 POST1184
 POST1185
 POST1186
 POST1187
 POST1188
 POST1189
 POST1190
 POST1191
 POST1192
 POST1193
 POST1194
 POST1195
 POST1196
 POST1197
 POST1198
 POST1199
 POST1200
 POST1201
 POST1202
 POST1203
 POST1204
 POST1205
 POST1206
 POST1207
 POST1208
 POST1209
 POST1210
 POST1211
 POST1212
 POST1213
 POST1214
 POST1215
 POST1216

Appendix F: OR3_PostORG.for Program Listing

```

lib3 = 0
lcnt = 0
C
C CONVERT INITIAL NUCLIDE CONCENTRATIONS FROM CURIES TO GRAMS AND
C ASSIGN ORIGEN2 SPECIFIC NUCLIDE IDENTIFIER
C
      DO 20 j = 1, num
      CALL CONV(nucnam(1,j),curies(1,j),nucid(1,j),
    &          grams(1,j), lib(1,j))
20  CONTINUE
C
      DO 25 k = 1, num
      IF(lib(1,k).EQ.1)THEN
        lib1 = lib1 + 1
        var1(lib1) = k
      ELSEIF(lib(1,k).EQ.2)THEN
        lib2 = lib2 + 1
        var2(lib2) = k
      ELSEIF(lib(1,k).EQ.3)THEN
        lib3 = lib3 + 1
        var3(lib3) = k
      ENDIF
25  CONTINUE
C
C FOR PRESENT YEAR, FROM LIBRARY 1 - ACTIVATION PRODUCTS
C
      IF(lib1.LE.3)THEN
        num = lib1
        IF(lib1.EQ.1)THEN
          WRITE(out,5105) (nucid(1,var1(L)),grams(1,var1(L)),
    &              L=1,lib1)
5105  FORMAT(2x,'1',5x,A7,2x,e10.4,2x,'0',2x,'0')
          ELSEIF(lib1.EQ.2)THEN
            WRITE(out,5110) (nucid(1,var1(L)),grams(1,var1(L)),
    &              L=1,2)
5110  FORMAT(2x,'1',5x,2(a7,2x,e10.4,2x),'0',2x,'0')
          ELSEIF(lib1.EQ.3)THEN
            WRITE(out,5115) (nucid(1,var1(L)),grams(1,var1(L)),
    &              L=1,3)
5115  FORMAT(2x,'1',5x,3(a7,2x,e10.4,2x),'0',2x,'0')
          ENDIF
        ELSEIF(lib1.GT.3)THEN
          ldiv1 = int(lib1/3)
          ldiv2 = lib1 - 3*ldiv1
          m = 0
          IF(ldiv2.EQ.0)THEN
1010  m = m + 3
            IF(m.LE.lib1)THEN
              n = m-2
              WRITE(out,5120) (nucid(1,var1(L)),grams(1,var1(L)),
    &              L=n,m)
5120  FORMAT(2x,'1',5x,3(a7,2x,e10.4,2x),'0',2x,'0')
              GO TO 1010
            ENDIF
          ELSEIF(ldiv2.GT.0)THEN
            IF(ldiv1.EQ.1)THEN
              WRITE(out,5120) (nucid(1,var1(L)),grams(1,var1(L)),
    &              L=1,3)
            IF(ldiv2.EQ.1)THEN
              WRITE(out,5105) (nucid(1,var1(L)),grams(1,var1(L)),
    &              L=4, lib1)
            ELSEIF(ldiv2.EQ.2)THEN
              WRITE(out,5110) (nucid(1,var1(L)),grams(1,var1(L)),
    &              L=4, lib1)
            ENDIF
          ELSE
1015  DO 30 k = 1, ldiv1
            m = m + 3
            lcnt = lcnt + 1
            IF(lcnt.LE.ldiv1)THEN
              n = m-2
              WRITE(out,5120) (nucid(1,var1(L)),grams(1,var1(L)),
    &              L=n,m)
            GO TO 1015
            ELSEIF(lcnt.GT.ldiv1)THEN
              m = m-3

```

POST1217
 POST1218
 POST1219
 POST1220
 POST1221
 POST1222
 POST1223
 POST1224
 POST1225
 POST1226
 POST1227
 POST1228
 POST1229
 POST1230
 POST1231
 POST1232
 POST1233
 POST1234
 POST1235
 POST1236
 POST1237
 POST1238
 POST1239
 POST1240
 POST1241
 POST1242
 POST1243
 POST1244
 POST1245
 POST1246
 POST1247
 POST1248
 POST1249
 POST1250
 POST1251
 POST1252
 POST1253
 POST1254
 POST1255
 POST1256
 POST1257
 POST1258
 POST1259
 POST1260
 POST1261
 POST1262
 POST1263
 POST1264
 POST1265
 POST1266
 POST1267
 POST1268
 POST1269
 POST1270
 POST1271
 POST1272
 POST1273
 POST1274
 POST1275
 POST1276
 POST1277
 POST1278
 POST1279
 POST1280
 POST1281
 POST1282
 POST1283
 POST1284
 POST1285
 POST1286
 POST1287
 POST1288
 POST1289
 POST1290
 POST1291
 POST1292

Appendix F: OR3_PostORG.for Program Listing

```

30          ENDIF                                POST1293
          CONTINUE                              POST1294
          m = m+1                              POST1295
          IF(ldiv2.EQ.1)THEN                    POST1296
            WRITE(out,5105) (nucid(1,var1(L)),grams(1,var1(L)),
            &                                L=m,lib1) POST1297
          ELSEIF(ldiv2.EQ.2)THEN                POST1298
            WRITE(out,5110) (nucid(1,var1(L)),grams(1,var1(L)),
            &                                L=m,lib1) POST1299
          ENDIF                                POST1300
        ENDIF                                POST1301
      ENDIF                                POST1302
    ENDIF                                POST1303
  ENDIF                                POST1304
C                                POST1305
C FOR PRESENT YEAR, FROM LIBRARY 2 - ACTINIDES POST1306
C                                POST1307
          lcnt = 0                             POST1308
          IF(lib2.LE.3)THEN                     POST1309
            num = lib2                         POST1310
            IF(lib2.EQ.1)THEN                   POST1311
              WRITE(out,5125) (nucid(1,var2(L)),grams(1,var2(L)),
              &                                L=1,lib2) POST1312
            5125  FORMAT(2x,'2',5x,A7,2x,e10.4,2x,'0',2x,'0') POST1313
            ELSEIF(lib2.EQ.2)THEN               POST1314
              WRITE(out,5130) (nucid(1,var2(L)),grams(1,var2(L)),
              &                                L=1,2) POST1315
            5130  FORMAT(2x,'2',5x,2(a7,2x,e10.4,2x),'0',2x,'0') POST1316
            ELSEIF(lib2.EQ.3)THEN               POST1317
              WRITE(out,5135) (nucid(1,var2(L)),grams(1,var2(L)),
              &                                L=1,3) POST1318
            5135  FORMAT(2x,'2',5x,3(a7,2x,e10.4,2x),'0',2x,'0') POST1319
            ENDIF                                POST1320
          ELSEIF(lib2.GT.3)THEN                 POST1321
            ldiv1 = int(lib2/3)                 POST1322
            ldiv2 = lib2 - 3*ldiv1              POST1323
            m = 0                               POST1324
            IF(ldiv2.EQ.0)THEN                   POST1325
              1020  m = m + 3                     POST1326
              IF(m.LE.lib2)THEN                  POST1327
                n = m-2                          POST1328
                WRITE(out,5140) (nucid(1,var2(L)),grams(1,var2(L)),
                &                                L=n,m) POST1329
              5140  FORMAT(2x,'2',5x,3(a7,2x,e10.4,2x),'0',2x,'0') POST1330
              GO TO 1020                         POST1331
            ENDIF                                POST1332
          ELSEIF(ldiv2.GT.0)THEN                 POST1333
            IF(ldiv1.EQ.1)THEN                   POST1334
              WRITE(out,5140) (nucid(1,var2(L)),grams(1,var2(L)),
              &                                L=1,3) POST1335
            ELSEIF(ldiv2.EQ.1)THEN               POST1336
              WRITE(out,5125) (nucid(1,var2(L)),grams(1,var2(L)),
              &                                L=4,lib2) POST1337
            ELSEIF(ldiv2.EQ.2)THEN               POST1338
              WRITE(out,5130) (nucid(1,var2(L)),grams(1,var2(L)),
              &                                L=4,lib2) POST1339
            ENDIF                                POST1340
          ELSE                                   POST1341
            DO 35 k = 1, ldiv1                   POST1342
              1025  m = m + 3                     POST1343
              lcnt = lcnt + 1                     POST1344
              IF(lcnt.LE.ldiv1)THEN                POST1345
                n = m-2                          POST1346
                WRITE(out,5140) (nucid(1,var2(L)),grams(1,var2(L)),
                &                                L=n,m) POST1347
              GO TO 1025                         POST1348
              ELSEIF(lcnt.GT.ldiv1)THEN             POST1349
                m = m-3                          POST1350
              ENDIF                                POST1351
            CONTINUE                              POST1352
            m = m+1                              POST1353
            IF(ldiv2.EQ.1)THEN                    POST1354
              WRITE(out,5125) (nucid(1,var2(L)),grams(1,var2(L)),
              &                                L=m,lib2) POST1355
            ELSEIF(ldiv2.EQ.2)THEN                POST1356
              WRITE(out,5130) (nucid(1,var2(L)),grams(1,var2(L)),
              &                                L=m,lib2) POST1357
            ENDIF                                POST1358
          CONTINUE                              POST1359
          m = m+1                              POST1360
          IF(ldiv2.EQ.1)THEN                    POST1361
            WRITE(out,5125) (nucid(1,var2(L)),grams(1,var2(L)),
            &                                L=m,lib2) POST1362
          ELSEIF(ldiv2.EQ.2)THEN                POST1363
            WRITE(out,5130) (nucid(1,var2(L)),grams(1,var2(L)),
            &                                L=m,lib2) POST1364
          ENDIF                                POST1365
        CONTINUE                              POST1366
      CONTINUE                              POST1367
    CONTINUE                              POST1368
  CONTINUE

```

```

      ENDIF
      ENDIF
      ENDIF
      ENDIF
C
C   FOR PRESENT YEAR, FROM LIBRARY 3 - FISSION PRODUCTS
C
      lcnt = 0
      IF(lib3.LE.3)THEN
        num = lib3
        IF(lib3.EQ.1)THEN
          WRITE(out,5145) (nucid(1,var3(L)),grams(1,var3(L)),
            &               L=1,lib3)
          5145   FORMAT(2x,'3',5x,A7,2x,e10.4,2x,'0',2x,'0')
          ELSEIF(lib3.EQ.2)THEN
            WRITE(out,5150) (nucid(1,var3(L)),grams(1,var3(L)),
            &               L=1,2)
          5150   FORMAT(2x,'3',5x,2(a7,2x,e10.4,2x),'0',2x,'0')
          ELSEIF(lib3.EQ.3)THEN
            WRITE(out,5155) (nucid(1,var3(L)),grams(1,var3(L)),
            &               L=1,3)
          5155   FORMAT(2x,'3',5x,3(a7,2x,e10.4,2x),'0',2x,'0')
          ENDIF
        ELSEIF(lib3.GT.3)THEN
          ldiv1 = int(lib3/3)
          ldiv2 = lib3 - 3*ldiv1
          m = 0
          IF(ldiv2.EQ.0)THEN
            1030   m = m + 3
            IF(m.LE.lib3)THEN
              n = m-2
              WRITE(out,5160) (nucid(1,var3(L)),grams(1,var3(L)),
            &               L=n,m)
            5160   FORMAT(2x,'3',5x,3(a7,2x,e10.4,2x),'0',2x,'0')
            GO TO 1030
            ENDIF
          ELSEIF(ldiv2.GT.0)THEN
            IF(ldiv1.EQ.1)THEN
              WRITE(out,5160) (nucid(1,var3(L)),grams(1,var3(L)),
            &               L=1,3)
            &       IF(ldiv2.EQ.1)THEN
              WRITE(out,5145) (nucid(1,var3(L)),grams(1,var3(L)),
            &               L=4,lib3)
            &       ELSEIF(ldiv2.EQ.2)THEN
              WRITE(out,5150) (nucid(1,var3(L)),grams(1,var3(L)),
            &               L=4,lib3)
            &       ENDIF
            ELSE
              DO 40 k = 1, ldiv1
            1035   m = m + 3
                  lcnt = lcnt + 1
                  IF(lcnt.LE.ldiv1)THEN
                    n = m-2
                    WRITE(out,5160) (nucid(1,var3(L)),grams(1,var3(L)),
            &               L=n,m)
            &       GO TO 1035
                  ELSEIF(lcnt.GT.ldiv1)THEN
                    m = m-3
                  ENDIF
            40   CONTINUE
                  m = m+1
                  IF(ldiv2.EQ.1)THEN
                    WRITE(out,5145) (nucid(1,var3(L)),grams(1,var3(L)),
            &               L=m,lib3)
            &       ELSEIF(ldiv2.EQ.2)THEN
                    WRITE(out,5150) (nucid(1,var3(L)),grams(1,var3(L)),
            &               L=m,lib3)
            &       ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
      ENDIF
      ENDIF
C
      WRITE(out,5165)
      5165   FORMAT(2x,'0')
C
      CLOSE(UNIT=12)

```

POST1369
 POST1370
 POST1371
 POST1372
 POST1373
 POST1374
 POST1375
 POST1376
 POST1377
 POST1378
 POST1379
 POST1380
 POST1381
 POST1382
 POST1383
 POST1384
 POST1385
 POST1386
 POST1387
 POST1388
 POST1389
 POST1390
 POST1391
 POST1392
 POST1393
 POST1394
 POST1395
 POST1396
 POST1397
 POST1398
 POST1399
 POST1400
 POST1401
 POST1402
 POST1403
 POST1404
 POST1405
 POST1406
 POST1407
 POST1408
 POST1409
 POST1410
 POST1411
 POST1412
 POST1413
 POST1414
 POST1415
 POST1416
 POST1417
 POST1418
 POST1419
 POST1420
 POST1421
 POST1422
 POST1423
 POST1424
 POST1425
 POST1426
 POST1427
 POST1428
 POST1429
 POST1430
 POST1431
 POST1432
 POST1433
 POST1434
 POST1435
 POST1436
 POST1437
 POST1438
 POST1439
 POST1440
 POST1441
 POST1442
 POST1443
 POST1444

C	CLOSE(UNIT=15)	POST1445
C	RETURN	POST1446
C	END	POST1447
C	CC	POST1448
C	SUBROUTINE CONV (name,activ,namy,activy,nuclib)	POST1450
C	holly trellue 6741 6/8/95	POST1451
C	CC	POST1452
C	C THIS SUBROUTINE WAS WRITTEN BY HOLLY TRELLUE, 6/8/95	POST1453
C	C MODIFIED BY J.LISCUM-POWELL 6/29/95 WITH ORIGEN2 DECAY LIBRARY HALF	POST1454
C	C LIVES NUCLIDES ADDED 10/16/95, ADDITIONS MADE 1/8/96	POST1455
C	CC	POST1456
C	C This subroutine converts nuclide concentrations in Curies into	POST1457
C	C equivalent grams of the nuclide to be used as ORIGEN2 input.	POST1458
C	CC	POST1459
C	CHARACTER name*6, namy*6	POST1460
C	INTEGER nuclib	POST1461
C	REAL activ, activy, m,Thalf,sa,sal	POST1462
C	C ALL VALUES FOR ATOMIC MASS NUMBERS AND HALFLIVES IN THIS FILE WERE	POST1463
C	C TAKEN FROM THE CHART OF THE NUCLIDES.	POST1464
C	IF (name(1:5).EQ.'AC227') THEN	POST1465
C	C ATOMIC WEIGHT VALUES(M) WERE CALCULATED BY USING THE MASS EXCESS VALUE	POST1466
C	C FOR EACH ISOTOPE. MASS EXCESS VALUES WERE TAKEN FROM "NUCLEAR WALLET	POST1467
C	C CARDS" BY JAGDISH TULI. JANUARY 1985.	POST1468
C	m=227.0 + 25.849/931.5016	POST1469
C	C FOR CONFORMITY, THE HALF LIVES ARE CONVERTED TO SECONDS FOR	POST1470
C	C EVERY ISOTOPE. AC227'S HALF LIFE IS 21.77 YEARS. ALL VALUES FOR	POST1471
C	C HALFLIVES IN THIS PROGRAM COME FROM THE ORIGEN2 DECAY LIBRARY	POST1472
C	Thalf = 6.871e08	POST1473
C	nuclib=2	POST1474
C	namy(1:6)='892270'	POST1475
C	ELSEIF (name(1:5).EQ.'AC225') THEN	POST1476
C	M=225.0 +21.626/931.5016	POST1477
C	Thalf = 8.640e05	POST1478
C	nuclib=2	POST1479
C	namy(1:6)='892250'	POST1480
C	ELSEIF (name(1:5).EQ.'AC228') THEN	POST1481
C	m=228.0 + 28.890/931.5016	POST1482
C	Thalf = 2.207e04	POST1483
C	nuclib=2	POST1484
C	namy(1:6)='892280'	POST1485
C	ELSEIF (name(1:5).EQ.'AG110') THEN	POST1486
C	if(name(6:6).EQ.'M')THEN	POST1487
C	m=110.0 - 87.341/931.5016	POST1488
C	Thalf =2.159E07	POST1489
C	namy(1:6)='471101'	POST1490
C	nuclib=3	POST1491
C	ELSE	POST1492
C	m=110.0 - 87.459/931.5016	POST1493
C	Thalf =2.460E01	POST1494
C	namy(1:6)='471100'	POST1495
C	nuclib=3	POST1496
C	ENDIF	POST1497
C	ELSEIF (name(1:5).EQ.'AM240') THEN	POST1498
C	m=240.0 + 51.498/931.5016	POST1499
C	Thalf =1.829e05	POST1500
C	namy(1:6)='952400'	POST1501
C	nuclib=2	POST1502
C	ELSEIF (name(1:5).EQ.'AM241') THEN	POST1503
C	m=241.0 + 52.931/931.5016	POST1504
C	Thalf = 1.364e10	POST1505
C	namy(1:6)='952410'	POST1506
C	nuclib=2	POST1507
C	ELSEIF (name(1:5).EQ.'AM242') THEN	POST1508
C	m=242.0 + 55.463/931.5016	POST1509
C	Thalf = 5.767e04	POST1510
C	namy(1:6)='952420'	POST1511
C	nuclib=2	POST1512
C	ELSEIF (name(1:5).EQ.'AM243') THEN	POST1513
C		POST1514
C		POST1515
C		POST1516
C		POST1517
C		POST1518
C		POST1519
C		POST1520

m=243.0 + 57.169/931.5016	POST1521
Thalf=2.329e11	POST1522
namy(1:6)='952430'	POST1523
nuclib=2	POST1524
ELSEIF (name(1:5).EQ.'AM244') THEN	POST1525
m=244.0 + 59.877/931.5016	POST1526
Thalf=3.636e04	POST1527
namy(1:6)='952440'	POST1528
nuclib=2	POST1529
ELSEIF (name(1:5).EQ.'AM245') THEN	POST1530
m=245.0 + 61.891/931.5016	POST1531
Thalf=7.452e03	POST1532
namy(1:6)='952450'	POST1533
nuclib=2	POST1534
ELSEIF (name(1:5).EQ.'AM246') THEN	POST1535
m=246.0 + 64.990/931.5016	POST1536
Thalf=1.500e03	POST1537
namy(1:6)='952460'	POST1538
nuclib=2	POST1539
ELSEIF (name(1:5).EQ.'AT217') THEN	POST1540
m=217.0 + 4.383/931.5016	POST1541
Thalf=3.230e-02	POST1542
namy(1:6)='852170'	POST1543
nuclib=2	POST1544
ELSEIF (name(1:6).EQ.'BA137M') THEN	POST1545
m=137.0 - 87.070/931.5016	POST1546
Thalf=1.531e02	POST1547
namy(1:6)='561371'	POST1548
nuclib=1	POST1549
ELSEIF (name(1:5).EQ.'BI210') THEN	POST1550
m=210.0 - 14.815/931.5016	POST1551
Thalf=4.330e05	POST1552
namy(1:6)='832100'	POST1553
nuclib=2	POST1554
ELSEIF (name(1:5).EQ.'BI211') THEN	POST1555
m=211.0 - 11.873/931.5016	POST1556
Thalf=1.278e02	POST1557
namy(1:6)='832110'	POST1558
nuclib=2	POST1559
ELSEIF (name(1:5).EQ.'BI212') THEN	POST1560
m=212.0 - 8.142/931.5016	POST1561
Thalf=3.633e03	POST1562
namy(1:6)='832130'	POST1563
nuclib=2	POST1564
ELSEIF (name(1:5).EQ.'BI213') THEN	POST1565
m=213.0 - 5.244/931.5016	POST1566
Thalf=2.739e03	POST1567
namy(1:6)='832130'	POST1568
nuclib=2	POST1569
ELSEIF (name(1:5).EQ.'BI214') THEN	POST1570
m=214.0 - 1.218/931.5016	POST1571
Thalf=1.194e03	POST1572
namy(1:6)='832140'	POST1573
nuclib=2	POST1574
ELSEIF (name(1:5).EQ.'BK249') THEN	POST1575
m=249.0 + 69.843/931.5016	POST1576
Thalf=2.765e07	POST1577
namy(1:6)='972490'	POST1578
nuclib=2	POST1579
ELSEIF (name(1:5).EQ.'BK250') THEN	POST1580
m=250.0 + 72.951/931.5016	POST1581
Thalf=1.160e04	POST1582
namy(1:6)='972500'	POST1583
nuclib=2	POST1584
ELSEIF (name(1:3).EQ.'C14') THEN	POST1585
m=14.0 + 3.02/931.5016	POST1586
Thalf=1.808e11	POST1587
namy(1:6)='060140'	POST1588
nuclib=1	POST1589
ELSEIF (name(1:5).EQ.'CD109') THEN	POST1590
m=109.0 - 88.507/931.5016	POST1591
Thalf=4.009e07	POST1592
namy(1:6)='481090'	POST1593
nuclib=1	POST1594
ELSEIF (name(1:6).EQ.'CD113M') THEN	POST1595
m=113.0 - 88.786/931.5016	POST1596

Thalf=4.604E08	POST1597
namy(1:6)='481131'	POST1598
nuclib=3	POST1599
ELSEIF (name(1:5).EQ.'CE144') THEN	POST1600
m=144.0 - 80.441/931.5016	POST1601
Thalf=2.456e07	POST1602
namy(1:6)='581440'	POST1603
nuclib=1	POST1604
ELSEIF (name(1:5).EQ.'CF249') THEN	POST1605
m=249.0 + 69.717/931.5016	POST1606
Thalf = 1.106e10	POST1607
namy(1:6)='982490'	POST1608
nuclib=2	POST1609
ELSEIF (name(1:5).EQ.'CF250') THEN	POST1610
m=250.0 + 71.167/931.5016	POST1611
Thalf=4.128e08	POST1612
namy(1:6)='982500'	POST1613
nuclib=2	POST1614
ELSEIF (name(1:5).EQ.'CF251') THEN	POST1615
m=251.0 + 74.129/931.5016	POST1616
Thalf=2.834e10	POST1617
namy(1:6)='982510'	POST1618
nuclib=2	POST1619
ELSEIF (name(1:5).EQ.'CF252') THEN	POST1620
m=252.0 + 76.03/931.5016	POST1621
Thalf=8.325e07	POST1622
namy(1:6)='982520'	POST1623
nuclib=2	POST1624
ELSEIF (name(1:5).EQ.'CM242') THEN	POST1625
m=242.0 + 54.800/931.5016	POST1626
Thalf=1.410e07	POST1627
namy(1:6)='962420'	POST1628
nuclib=2	POST1629
ELSEIF (name(1:5).EQ.'CM243') THEN	POST1630
m=243.0 + 57.177/931.5016	POST1631
Thalf=8.994e08	POST1632
namy(1:6)='962430'	POST1633
nuclib=2	POST1634
ELSEIF (name(1:5).EQ.'CM244') THEN	POST1635
m=244.0 + 58.449/931.5016	POST1636
Thalf=5.715e08	POST1637
namy(1:6)='962440'	POST1638
nuclib=2	POST1639
ELSEIF (name(1:5).EQ.'CM245') THEN	POST1640
m=245.0 + 60.998/931.5016	POST1641
Thalf=2.682e11	POST1642
namy(1:6)='962450'	POST1643
nuclib=2	POST1644
ELSEIF (name(1:5).EQ.'CM246') THEN	POST1645
m=246.0 + 62.614/931.5016	POST1646
Thalf=1.493e11	POST1647
namy(1:6)='962460'	POST1648
nuclib=2	POST1649
ELSEIF (name(1:5).EQ.'CM248') THEN	POST1650
m=248.0 + 67.388/931.5016	POST1651
Thalf=1.070e13	POST1652
namy(1:6)='962480'	POST1653
nuclib=2	POST1654
ELSEIF (name(1:5).EQ.'CM250') THEN	POST1655
m=250.0 + 72.985/931.5016	POST1656
Thalf=5.491e11	POST1657
namy(1:6)='962500'	POST1658
nuclib=2	POST1659
ELSEIF (name(1:4).EQ.'CO58') THEN	POST1660
m=58.0 - 59.844/931.5016	POST1661
Thalf=6.115e06	POST1662
namy(1:6)='270580'	POST1663
nuclib=1	POST1664
ELSEIF (name(1:4).EQ.'CO60') THEN	POST1665
m=60.0 - 61.646/931.5016	POST1666
Thalf=1.663e08	POST1667
namy(1:6)='270600'	POST1668
nuclib=1	POST1669
ELSEIF (name(1:4).EQ.'CR51') THEN	POST1670
m=51.0 - 51.447/931.5016	POST1671
Thalf=2.394e06	POST1672

namy(1:6)='240510'	POST1673
nuclib=1	POST1674
ELSEIF (name(1:5).EQ.'CS134') THEN	POST1675
m=134.0 - 86.906/931.5016	POST1676
Thalf=6.507e07	POST1677
namy(1:6)='551340'	POST1678
nuclib=1	POST1679
ELSEIF (name(1:5).EQ.'CS135') THEN	POST1680
m=135.0 - 87.662/931.5016	POST1681
Thalf=7.258E13	POST1682
namy(1:6)='551350'	POST1683
nuclib=1	POST1684
ELSEIF (name(1:5).EQ.'CS137') THEN	POST1685
m=137.0 - 86.556/931.5016	POST1686
Thalf=9.467e08	POST1687
namy(1:6)='551370'	POST1688
nuclib=1	POST1689
ELSEIF (name(1:5).EQ.'ES253') THEN	POST1690
m=253.0 + 79.007/931.5016	POST1691
Thalf=1.769e06	POST1692
namy(1:6)='992530'	POST1693
nuclib=2	POST1694
ELSEIF (name(1:5).EQ.'ES254') THEN	POST1695
IF (name(6:6).EQ.'M') THEN	POST1696
m=254.0 + 82.072/931.5016	POST1697
Thalf=1.415e05	POST1698
namy(1:6)='992541'	POST1699
nuclib=2	POST1700
ELSE	POST1701
m=254.0 + 81.994/931.5016	POST1702
Thalf=2.382e07	POST1703
namy(1:6)='992540'	POST1704
nuclib=2	POST1705
ENDIF	POST1706
ELSEIF (name(1:5).EQ.'EU150') THEN	POST1707
m=150.0 - 74.800/931.5016	POST1708
Thalf=1.136e09	POST1709
namy(1:6)='631500'	POST1710
nuclib=3	POST1711
ELSEIF (name(1:5).EQ.'EU152') THEN	POST1712
m=152.0 - 72.899/931.5016	POST1713
Thalf=4.292e08	POST1714
namy(1:6)='631520'	POST1715
nuclib=3	POST1716
ELSEIF (name(1:5).EQ.'EU154') THEN	POST1717
m=154.0 - 71.748/931.5016	POST1718
Thalf=2.714e08	POST1719
namy(1:6)='631540'	POST1720
nuclib=3	POST1721
ELSEIF (name(1:5).EQ.'EU155') THEN	POST1722
m=155.0 - 71.829/931.5016	POST1723
Thalf=1.565e08	POST1724
namy(1:6)='631550'	POST1725
nuclib=3	POST1726
ELSEIF (name(1:4).EQ.'FE55') THEN	POST1727
m= 55.0 - 57.476/931.5016	POST1728
Thalf=8.2048e07	POST1729
namy(1:6)='260550'	POST1730
nuclib=1	POST1731
ELSEIF (name(1:4).EQ.'FE59') THEN	POST1732
m=59.0 - 60.661/931.5016	POST1733
Thalf=3.888e06	POST1734
namy(1:6)='260590'	POST1735
nuclib=1	POST1736
ELSEIF (name(1:5).EQ.'GD152') THEN	POST1737
m = 152.0 -74.718/931.5016	POST1738
Thalf = 3.408E21	POST1739
namy(1:6) = '641520'	POST1740
nuclib=1	POST1741
ELSEIF (name(1:5).EQ.'GD162') THEN	POST1742
m = 162.0 -64.240/931.5016	POST1743
Thalf = 6.000E02	POST1744
namy(1:6) = '641620'	POST1745
nuclib=1	POST1746
ELSEIF (name(1:2).EQ.'H3') THEN	POST1747
m=3.0 + 14.95/931.5016	POST1748

```

      Thalf=3.897e08
      namy(1:6)='010030'
      nuclib=1
      ELSEIF (name(1:4).EQ.'I129') THEN
        m=129.0 - 88.507/931.5016
        Thalf=4.954e14
        namy(1:6)='531290'
        nuclib=1
      ELSEIF (name(1:4).EQ.'KR85') THEN
        m=85.0 - 81.477/931.5016
        Thalf=3.383e08
        namy(1:6)='360850'
        nuclib=1
      ELSEIF (name(1:4).EQ.'MN54') THEN
        m=54.0 - 55.553/931.5016
        Thalf=2.700e07
        namy(1:6)='250540'
        nuclib=1
      ELSEIF (name(1:4).EQ.'NB95') THEN
        IF (name(5:5).EQ.'M') THEN
          m=95.0 - 86.547/931.5016
          Thalf=3.118E05
          namy(1:6)='410951'
          nuclib=1
        ELSE
          m=95.0 - 86.783/931.5016
          Thalf=3.037e06
          namy(1:6)='410950'
          nuclib=1
        ENDIF
      ELSEIF (name(1:4).EQ.'NI59') THEN
        m=59.0 - 61.153/931.5016
        Thalf=2.5246e12
        namy(1:6)='280590'
        nuclib=1
      ELSEIF (name(1:4).EQ.'NI63') THEN
        m=63.0 - 65.512/931.5016
        Thalf=2.9032e9
        namy='280630'
        nuclib=1
      ELSEIF (name(1:5).EQ.'NP237') THEN
        m=237.0 + 44.868/931.5016
        Thalf=6.753e13
        namy='932370'
        nuclib=2
      ELSEIF (name(1:5).EQ.'NP239') THEN
        m=239.0 + 49.306/931.5016
        Thalf=2.035e05
        namy='932390'
        nuclib=2
      ELSEIF (name(1:5).EQ.'NP240') THEN
        IF (name(6:6).EQ.'M') THEN
          m=240.0 + 52.321/931.5016
          Thalf=4.440e02
          namy='932401'
          nuclib=2
        ELSE
          m=240.0 + 52.321/931.5016
          Thalf=3.900e03
          namy='932400'
          nuclib=2
        ENDIF
      ELSEIF (name(1:5).EQ.'PA231') THEN
        m=231.0 + 33.422/931.5016
        Thalf=1.034e12
        namy='912310'
        nuclib=2
      ELSEIF (name(1:5).EQ.'PA233') THEN
        m=233.0 + 37.485/931.5016
        Thalf=2.333e06
        namy='912330'
        nuclib=2
      ELSEIF (name(1:5).EQ.'PA234') THEN
        IF (name(6:6).EQ.'M') THEN
          m=234.0 + 40.408/931.5016
          Thalf=7.020e01

```

POST1749
 POST1750
 POST1751
 POST1752
 POST1753
 POST1754
 POST1755
 POST1756
 POST1757
 POST1758
 POST1759
 POST1760
 POST1761
 POST1762
 POST1763
 POST1764
 POST1765
 POST1766
 POST1767
 POST1768
 POST1769
 POST1770
 POST1771
 POST1772
 POST1773
 POST1774
 POST1775
 POST1776
 POST1777
 POST1778
 POST1779
 POST1780
 POST1781
 POST1782
 POST1783
 POST1784
 POST1785
 POST1786
 POST1787
 POST1788
 POST1789
 POST1790
 POST1791
 POST1792
 POST1793
 POST1794
 POST1795
 POST1796
 POST1797
 POST1798
 POST1799
 POST1800
 POST1801
 POST1802
 POST1803
 POST1804
 POST1805
 POST1806
 POST1807
 POST1808
 POST1809
 POST1810
 POST1811
 POST1812
 POST1813
 POST1814
 POST1815
 POST1816
 POST1817
 POST1818
 POST1819
 POST1820
 POST1821
 POST1822
 POST1823
 POST1824

```

        namy='912341'
        nuclib=2
    ELSE
        m=234.0 + 40.334/931.5016
        Thalf=2.412e04
        namy='912340'
        nuclib=2
    ENDIF
ELSEIF (name(1:5).EQ.'PB209') THEN
    m=209.0 - 17.638/931.5016
    Thalf=1.188e04
    namy='822090'
    nuclib=2
ELSEIF (name(1:5).EQ.'PB210') THEN
    m=210.0 - 14.752/931.5016
    Thalf=7.037e08
    namy='822100'
    nuclib=2
ELSEIF (name(1:5).EQ.'PB211') THEN
    m=211.0 - 10.494/931.5016
    Thalf=2.166e03
    namy='822110'
    nuclib=2
ELSEIF (name(1:5).EQ.'PB212') THEN
    m=212.0 - 7.571/931.5016
    Thalf=3.830e04
    namy='822120'
    nuclib=2
ELSEIF (name(1:5).EQ.'PB214') THEN
    m=214.0 - .188/931.5016
    Thalf=1.608e03
    namy='822140'
    nuclib=2
ELSEIF (name(1:5).EQ.'PD107') THEN
    m=107.0 - 88.374/931.5016
    Thalf=2.050E14
    namy='461070'
    nuclib=1
ELSEIF (name(1:5).EQ.'PM147') THEN
    m=147.0 - 79.052/931.5016
    Thalf=8.279e07
    namy='611470'
    nuclib=1
ELSEIF (name(1:5).EQ.'PO210') THEN
    m=210.0 - 15.977/931.5016
    Thalf=1.196e07
    namy='842100'
    nuclib=2
ELSEIF (name(1:5).EQ.'PO211') THEN
    m=211.0 - 12.457/931.5016
    Thalf=5.600e-01
    namy='842110'
    nuclib=2
ELSEIF (name(1:5).EQ.'PO212') THEN
    m=212.0 -10.394/931.5016
    Thalf=3.000e-07
    namy='842120'
    nuclib=2
ELSEIF (name(1:5).EQ.'PO213') THEN
    m=213.0 - 6.676/931.5016
    Thalf=4.200e-06
    namy='842130'
    nuclib=2
ELSEIF (name(1:5).EQ.'PO214') THEN
    m=214.0 - 4.493/931.5016
    Thalf=1.643e-04
    namy='842140'
    nuclib=2
ELSEIF (name(1:5).EQ.'PO215') THEN
    m=215.0 -0.542/931.5016
    Thalf=1.780e-03
    namy='842150'
    nuclib=2
ELSEIF (name(1:5).EQ.'PO216') THEN
    m=216.0 + 1.760 /931.5016
    Thalf=1.500e-01

```

```

POST1825
POST1826
POST1827
POST1828
POST1829
POST1830
POST1831
POST1832
POST1833
POST1834
POST1835
POST1836
POST1837
POST1838
POST1839
POST1840
POST1841
POST1842
POST1843
POST1844
POST1845
POST1846
POST1847
POST1848
POST1849
POST1850
POST1851
POST1852
POST1853
POST1854
POST1855
POST1856
POST1857
POST1858
POST1859
POST1860
POST1861
POST1862
POST1863
POST1864
POST1865
POST1866
POST1867
POST1868
POST1869
POST1870
POST1871
POST1872
POST1873
POST1874
POST1875
POST1876
POST1877
POST1878
POST1879
POST1880
POST1881
POST1882
POST1883
POST1884
POST1885
POST1886
POST1887
POST1888
POST1889
POST1890
POST1891
POST1892
POST1893
POST1894
POST1895
POST1896
POST1897
POST1898
POST1899
POST1900

```

Appendix F: OR3_PostORG.for Program Listing

namy='842160'	POST1901
nuclib=2	POST1902
ELSEIF (name(1:5).EQ.'PO218') THEN	POST1903
m=218.0 + 8.351/931.5016	POST1904
Thalf=1.830e02	POST1905
namy='842180'	POST1906
nuclib=2	POST1907
ELSEIF (name(1:5).EQ.'PR144') THEN	POST1908
m=144.0 - 80.76/931.5016	POST1909
Thalf=1.037e03	POST1910
namy='591440'	POST1911
nuclib=1	POST1912
ELSEIF (name(1:5).EQ.'PU236') THEN	POST1913
m=236.0 + 42.879/931.5016	POST1914
Thalf=8.997e07	POST1915
namy='942360'	POST1916
nuclib=2	POST1917
ELSEIF (name(1:5).EQ.'PU238') THEN	POST1918
m=238.0 + 46.16/931.5016	POST1919
Thalf=2.769e09	POST1920
namy='942380'	POST1921
nuclib=2	POST1922
ELSEIF (name(1:5).EQ.'PU239') THEN	POST1923
m=239.0 + 48.584/931.5016	POST1924
Thalf=7.594e11	POST1925
namy='942390'	POST1926
nuclib=2	POST1927
ELSEIF (name(1:5).EQ.'PU240') THEN	POST1928
m=240.0 + 50.122/931.5016	POST1929
Thalf=2.063e11	POST1930
namy='942400'	POST1931
nuclib=2	POST1932
ELSEIF (name(1:5).EQ.'PU241') THEN	POST1933
m=241.0 + 52.952/931.5016	POST1934
Thalf=4.544e08	POST1935
namy='942410'	POST1936
nuclib=2	POST1937
ELSEIF (name(1:5).EQ.'PU242') THEN	POST1938
m=242.0 + 54.713/931.5016	POST1939
Thalf=1.221e13	POST1940
namy='942420'	POST1941
nuclib=2	POST1942
ELSEIF (name(1:5).EQ.'PU244') THEN	POST1943
m=244.0 + 59.802/931.5016	POST1944
Thalf=2.607e15	POST1945
namy='942440'	POST1946
nuclib=2	POST1947
ELSEIF (name(1:5).EQ.'PU246') THEN	POST1948
m=246.0 + 65.391/931.5016	POST1949
Thalf=9.374e05	POST1950
namy='942460'	POST1951
nuclib=2	POST1952
ELSEIF (name(1:5).EQ.'RA223') THEN	POST1953
m=223.0 + 17.232/931.5016	POST1954
Thalf=9.879e05	POST1955
namy='882230'	POST1956
nuclib=2	POST1957
ELSEIF (name(1:5).EQ.'RA224') THEN	POST1958
m=224.0 + 18.804/931.5016	POST1959
Thalf=3.162e05	POST1960
namy='882240'	POST1961
nuclib=2	POST1962
ELSEIF (name(1:5).EQ.'RA225') THEN	POST1963
m=225.0 + 21.988/931.5016	POST1964
Thalf=1.279e06	POST1965
namy='882250'	POST1966
nuclib=2	POST1967
ELSEIF (name(1:5).EQ.'RA226') THEN	POST1968
m=226.0 + 23.662/931.5016	POST1969
Thalf=5.049e10	POST1970
namy='882260'	POST1971
nuclib=2	POST1972
ELSEIF (name(1:5).EQ.'RA228') THEN	POST1973
m=228.0 + 28.936/931.5016	POST1974
Thalf=2.1143e08	POST1975
namy='882280'	POST1976

```

        nuclib=2
ELSEIF (name(1:5).EQ.'RH106') THEN
    m=106.0 - 86.365/931.5016
    Thalf=29.9
    namy='451060'
    nuclib=1
ELSEIF (name(1:5).EQ.'RN219') THEN
    m=219.0 + 8.828/931.5016
    Thalf=3.960e00
    namy='862190'
    nuclib=2
ELSEIF (name(1:5).EQ.'RN220') THEN
    m=220.0 + 10.590/931.5016
    Thalf=5.560e01
    namy='862200'
    nuclib=2
ELSEIF (name(1:5).EQ.'RN222') THEN
    m=222.0 + 16.367/931.5016
    Thalf=3.304e05
    namy='862220'
    nuclib=2
ELSEIF (name(1:5).EQ.'RU106') THEN
    m=106.0 - 86.326/931.5016
    Thalf=3.181e07
    namy='441060'
    nuclib=1
ELSEIF (name(1:5).EQ.'SB125') THEN
    m=125.0 - 88.258/931.5016
    Thalf=8.741e07
    namy='511250'
    nuclib=1
ELSEIF (name(1:5).EQ.'SB126') THEN
    IF (NAME(6:6).EQ.'M') THEN
        m=126.0 - 86.382/931.5016
        Thalf=1.140E03
        namy='511261'
        nuclib=1
    ELSE
        m=126.0 - 86.400/931.5016
        Thalf=1.071E06
        namy='511260'
        nuclib=1
    ENDIF
ELSEIF (name(1:4).EQ.'SE79') THEN
    m=79.0 - 75.920/931.5016
    Thalf=2.050E12
    namy='340790'
    nuclib=1
ELSEIF (name(1:5).EQ.'SM147') THEN
    m=147.0 - 79.276/931.5016
    Thalf=3.377e18
    namy='621470'
    nuclib=1
ELSEIF (name(1:5).EQ.'SM148') THEN
    m=148.0 - 79.346/931.5016
    Thalf=2.525e23
    namy='621480'
    nuclib=1
ELSEIF (name(1:5).EQ.'SM151') THEN
    m=151.0 - 74.587/931.5016
    Thalf=2.840e09
    namy='621510'
    nuclib=1
ELSEIF (name(1:6).EQ.'SN119M') THEN
    m=119.0 - 89.978/931.5016
    Thalf=2.117E07
    namy='501191'
    nuclib=1
ELSEIF (name(1:6).EQ.'SN121M') THEN
    m=121.0 - 89.197/931.5016
    Thalf=1.577E09
    namy='501211'
    nuclib=1
ELSEIF (name(1:6).EQ.'SN123M') THEN
    m=123.0 - 87.795/931.5016
    Thalf=2.405E03

```

POST1977
 POST1978
 POST1979
 POST1980
 POST1981
 POST1982
 POST1983
 POST1984
 POST1985
 POST1986
 POST1987
 POST1988
 POST1989
 POST1990
 POST1991
 POST1992
 POST1993
 POST1994
 POST1995
 POST1996
 POST1997
 POST1998
 POST1999
 POST2000
 POST2001
 POST2002
 POST2003
 POST2004
 POST2005
 POST2006
 POST2007
 POST2008
 POST2009
 POST2010
 POST2011
 POST2012
 POST2013
 POST2014
 POST2015
 POST2016
 POST2017
 POST2018
 POST2019
 POST2020
 POST2021
 POST2022
 POST2023
 POST2024
 POST2025
 POST2026
 POST2027
 POST2028
 POST2029
 POST2030
 POST2031
 POST2032
 POST2033
 POST2034
 POST2035
 POST2036
 POST2037
 POST2038
 POST2039
 POST2040
 POST2041
 POST2042
 POST2043
 POST2044
 POST2045
 POST2046
 POST2047
 POST2048
 POST2049
 POST2050
 POST2051
 POST2052

Appendix F: OR3_PostORG.for Program Listing

namy='501231'	POST2053
nuclib=1	POST2054
ELSEIF (name(1:5).EQ.'SN126') THEN	POST2055
m=126.0 - 86.021/931.5016	POST2056
Thalf=3.156E12	POST2057
namy='501260'	POST2058
nuclib=3	POST2059
ELSEIF (name(1:4).EQ.'SR89') THEN	POST2060
m=89.0 - 86.211/931.5016	POST2061
Thalf=4.363e06	POST2062
namy='380890'	POST2063
nuclib=1	POST2064
ELSEIF (name(1:4).EQ.'SR90') THEN	POST2065
m=90.0 - 85.942/931.5016	POST2066
Thalf=9.190e08	POST2067
namy='380900'	POST2068
nuclib=1	POST2069
ELSEIF (name(1:5).EQ.'TA182') THEN	POST2070
m=182.0 - 46.436/931.5016	POST2071
Thalf=9.936e06	POST2072
namy='731820'	POST2073
nuclib=1	POST2074
ELSEIF (name(1:5).EQ.'TB162') THEN	POST2075
m=162.0 - 65.680/931.5016	POST2076
Thalf=4.482E02	POST2077
namy='651620'	POST2078
nuclib=1	POST2079
ELSEIF (name(1:4).EQ.'TC99') THEN	POST2080
m=99.0 - 87.324/931.5016	POST2081
Thalf=6.722e12	POST2082
namy='430990'	POST2083
nuclib=1	POST2084
ELSEIF (name(1:6).EQ.'TE125M') THEN	POST2085
m=125.0 - 88.879/931.5016	POST2086
Thalf=5.011e06	POST2087
namy='521251'	POST2088
nuclib=1	POST2089
ELSEIF (name(1:5).EQ.'TE127') THEN	POST2090
IF (NAME(6:6).EQ.'M') THEN	POST2091
m=127.0 - 88.198/931.5016	POST2092
Thalf=9.418E06	POST2093
namy='521271'	POST2094
nuclib=1	POST2095
ELSE	POST2096
m=127.0 - 88.286/931.5016	POST2097
Thalf=3.366E04	POST2098
namy='521270'	POST2099
nuclib=1	POST2100
ENDIF	POST2101
ELSEIF (name(1:5).EQ.'TH227') THEN	POST2102
m=227.0 +25.803/931.5016	POST2103
Thalf=1.617e06	POST2104
namy='902270'	POST2105
nuclib=2	POST2106
ELSEIF (name(1:5).EQ.'TH228') THEN	POST2107
m=228.0 + 26.749/931.5016	POST2108
Thalf=6.037e07	POST2109
namy='902280'	POST2110
nuclib=2	POST2111
ELSEIF (name(1:5).EQ.'TH229') THEN	POST2112
m=229.0 + 29.581/931.5016	POST2113
Thalf=2.316e11	POST2114
namy='902290'	POST2115
nuclib=2	POST2116
ELSEIF (name(1:5).EQ.'TH230') THEN	POST2117
m=230.0 + 30.858/931.5016	POST2118
Thalf=2.430e12	POST2119
namy='902300'	POST2120
nuclib=2	POST2121
ELSEIF (name(1:5).EQ.'TH231') THEN	POST2122
m=231.0 + 33.812/931.5016	POST2123
Thalf=9.187e04	POST2124
namy='902310'	POST2125
nuclib=2	POST2126
ELSEIF (name(1:5).EQ.'TH232') THEN	POST2127
m=232.0 + 35.444/931.5016	POST2128

Thalf=4.434e17	POST2129
namy='902320'	POST2130
nuclib=2	POST2131
ELSEIF (name(1:5).EQ.'TH234') THEN	POST2132
m=234.0 + 40.607/931.5016	POST2133
Thalf=2.082e06	POST2134
namy='902340'	POST2135
nuclib=2	POST2136
ELSEIF (name(1:5).EQ.'TL207') THEN	POST2137
m=207 - 21.049/931.5016	POST2138
Thalf=2.862e02	POST2139
namy(1:6)='812070'	POST2140
nuclib=2	POST2141
ELSEIF (name(1:5).EQ.'TL208') THEN	POST2142
m=208.0 - 16.774/931.5016	POST2143
Thalf=1.842e02	POST2144
namy(1:6)='812080'	POST2145
nuclib=2	POST2146
ELSEIF (name(1:5).EQ.'TL209') THEN	POST2147
m=209 - 13.652/931.5016	POST2148
Thalf=1.320e02	POST2149
namy(1:6)='812090'	POST2150
nuclib=2	POST2151
ELSEIF (name(1:4).EQ.'U232') THEN	POST2152
m=232.0 + 34.587/931.5016	POST2153
Thalf=2.272e09	POST2154
namy='922320'	POST2155
nuclib=2	POST2156
ELSEIF (name(1:4).EQ.'U233') THEN	POST2157
m=233.0 + 36.915/931.5016	POST2158
Thalf=5.002e12	POST2159
namy='922330'	POST2160
nuclib=2	POST2161
ELSEIF (name(1:4).EQ.'U234') THEN	POST2162
m=234.0 + 38.141/931.5016	POST2163
Thalf=7.716e12	POST2164
namy='922340'	POST2165
nuclib=2	POST2166
ELSEIF (name(1:4).EQ.'U235') THEN	POST2167
m=235.0 + 40.915/931.5016	POST2168
Thalf=2.221e16	POST2169
namy='922350'	POST2170
nuclib=2	POST2171
ELSEIF (name(1:4).EQ.'U236') THEN	POST2172
m=236.0 + 42.441/931.5016	POST2173
Thalf=7.389e14	POST2174
namy='922360'	POST2175
nuclib=2	POST2176
ELSEIF (name(1:4).EQ.'U237') THEN	POST2177
m=237.0 + 45.387/931.5016	POST2178
Thalf=5.832e05	POST2179
namy='922370'	POST2180
nuclib=2	POST2181
ELSEIF (name(1:4).EQ.'U238') THEN	POST2182
m=238.0 + 47.305/931.5016	POST2183
Thalf=1.410e17	POST2184
namy='922380'	POST2185
nuclib=2	POST2186
ELSEIF (name(1:4).EQ.'U240') THEN	POST2187
m=240.0 + 52.711/931.5016	POST2188
Thalf=5.076e04	POST2189
namy='922400'	POST2190
nuclib=2	POST2191
ELSEIF (name(1:3).EQ.'Y90') THEN	POST2192
m=90.0 - 86.488/931.5016	POST2193
Thalf=2.304e05	POST2194
namy='390900'	POST2195
nuclib=1	POST2196
ELSEIF (name(1:4).EQ.'ZN65') THEN	POST2197
m=65.0 - 65.91/931.5016	POST2198
Thalf=2.107e07	POST2199
namy='300650'	POST2200
nuclib=1	POST2201
ELSEIF (name(1:4).EQ.'ZR93') THEN	POST2202
m=93.0 - 87.120/931.5016	POST2203
Thalf=4.828E13	POST2204

[illegible]

	+	/,	_/_/_	_/_	_/_	'		POST2281
	+							POST2282
	+	/,	_/_	_/_/_	_/_	SANDIA NATI'	_/_/_	POST2283
	+					'ONAL LABORATORIES	_/_	POST2284
	+	)						POST2285
						WRITE(6,9803)		POST2286
9803						FORMAT(		POST2287
	+		_/_/_	_/_	_/_	Nuclear Wast'		POST2288
	+					'e Management Center _/_/_/_/_/'		POST2289
	+	/,	_/_	_/_/_	_/_			POST2290
	+						_/_/_	POST2291
	+	/,	_/_/_	_/_	_/_/_/_/_			POST2292
	+						_/_/_	POST2293
	+	/,					_/_/_	POST2294
	+							POST2295
	+	/,						POST2296
	+							POST2297
	+	/,						POST2298
	+							POST2299
	+	/,						POST2300
	+							POST2301
	+	/,				*****		POST2302
	+					*****		POST2303
	+	/,			*	ISSUED BY	,	POST2304
	+				*			POST2305
	+	/,			* SANDIA LABORATORI'			POST2306
	+				'ES,	*		POST2307
	+	)						POST2308
						WRITE(6,9804)		POST2309
9804						FORMAT(		POST2310
	+				*	A PRIME CONTRACT'		POST2311
	+				'OR *			POST2312
	+	/,			*****	TO THE '		POST2313
	+				*			POST2314
	+	/,			*	UNITED STA'		POST2315
	+				'TES *			POST2316
	+	/,			*	DEPARTMEN'		POST2317
	+				'T *			POST2318
	+	/,			*	OF '		POST2319
	+				*			POST2320
	+	/,			*	ENERGY '		POST2321
	+				*			POST2322
	+	/,			***** ---NOTICE---			POST2323
	+				-- *****			POST2324
	+	/,			*THIS REPORT WAS PREPARED AS AN A'			POST2325
	+				'CCOUNT OF WORK SPONSORED*			POST2326
	+	/,			* BY THE UNITED STATES GOVERNMENT'			POST2327
	+				'T. NEITHER THE UNITED *			POST2328
	+	/,			* STATES NOR THE UNITED STATES '			POST2329
	+				'DEPARTMENT OF ENERGY,	*		POST2330
	+	)						POST2331
						WRITE(6,9805)		POST2332
9805						FORMAT(		POST2333
	+				*	NOR ANY OF THEIR '		POST2334
	+				'EMPLOYEES,	*		POST2335
	+	/,			* NOR ANY OF THEIR CONTRACTORS, S'			POST2336
	+				'UBCONTRACTORS, OR THEIR *			POST2337
	+	/,			* EMPLOYEES, MAKES ANY WARRANTY, '			POST2338
	+				'EXPRESS OR IMPLIED, OR *			POST2339
	+	/,			* ASSUMES ANY LEGAL LIABILITY OR '			POST2340
	+				'RESPONSIBILITY FOR THE *			POST2341
	+	/,			***** ACCURACY'			POST2342
	+				', ***** *			POST2343
	+	/,			*	COMPLETENESS'		POST2344
	+				'SS * *			POST2345
	+	/,			* OR USEFULN'			POST2346
	+				'ESS * *			POST2347
	+	/,			* OF ANY '			POST2348
	+				*		*	POST2349
	+	/,			* INFORMATIO'			POST2350
	+				'N, *		*	POST2351
	+	/,			* APPARATUS'		*	POST2352
	+				, *		*	POST2353
	+	)						POST2354
						WRITE(6,9806)		POST2355
9806						FORMAT(		POST2356

```

+ * **** PRODUCT', * POST2357
+ * * * OR PROCES', * POST2358
+ /, ' * * * POST2359
+ * * 'S * * * POST2360
+ /, ' * * * DISCLOSED', * POST2361
+ * * * * * POST2362
+ /, ' * * * OR REPRESE', * POST2363
+ * * * * * POST2364
+ /, ' * * * * * POST2365
+ * * * * * POST2366
+ /, ' * * * * * POST2367
+ * * * * * POST2368
+ /, ' * * * * * POST2369
+ * * * * * POST2370
+ /, ' * * * * * POST2371
+ * * * * * POST2372
+ /, ' * * * * * POST2373
+ * * * * * POST2374
+ /, ' * * * * * POST2375
+ * * * * * POST2376
+ * * * * * POST2377
+ ) POST2378
WRITE(6,9807) POST2379
9807 FORMAT( POST2380
+ * * * * * POST2381
+ /, ' * * * * * POST2382
+ * * * * * POST2383
+ /, ' * * * * * POST2384
+ * * * * * POST2385
+ /, ' * * * * * POST2386
+ * * * * * POST2387
+ /, ' * * * * * POST2388
+ * * * * * POST2389
+ /, ' * * * * * POST2390
+ * * * * * POST2391
+ /, '1234567890123456789012345678901234567890', POST2392
+ * * * * * POST2393
+ /, '1 2 3 4', POST2394
+ * * * * * POST2395
+ /, '5 6 7', POST2396
+ * * * * * POST2397
+ /, ' * * * * * POST2398
+ * * * * * POST2399
+ ) POST2400
WRITE(6,9808) POST2401
9808 FORMAT( POST2402
+ ) POST2403
RETURN POST2404
END POST2405
C CMS REPLACEMENT HISTORY, Element OR3_POSTORG.FOR POST2406
C *4 8-APR-1996 11:13:27 JLPWEL "CORRECTION TO FUTURE DECAY OUTPUT POST2407
C *3 3-APR-1996 13:04:29 JLPWEL "CORRECTION TO BID LIST OF ISOTOPES POST2408
C *2 22-MAR-1996 12:16:53 RECARTE "adding in the program statement" POST2409
C *1 20-MAR-1996 14:25:51 CMWILLI "Initial load." POST2410
C CMS REPLACEMENT HISTORY, Element OR3_POSTORG.FOR POST2411

```

APPENDIX G: SAMPLE OUTPUT FILES FROM POSTORG

Intentionally Left Blank

Appendix G: Sample Output Files From Postorg

This appendix contains the three possible output files that can be generated from PostORG. The files include the following: ORNLCH93.SUM, ORNLC93F.U5, and ORNLC93F.SUM. The file ORNLCH93.SUM is a file containing the total 1993 nuclide concentrations after all initial inventories listed in ORNL__2.DAT have been decayed to the base year of 1993. The file ORNLC93F.U5 is an ORIGEN2 input file to decay the nuclide concentrations in ORNLCH93.SUM 75 years beyond the base year of 1993. Finally, ORNLC93F.SUM is the processed and sorted ORIGEN2 output for ORNLC93F.U5.

Appendix G: Sample Output Files From PostORG

ornlch93.sum FILE LISTING

	Base Year	Isotope Name	Activity in Curies at End of Base Year
1	1993		
2	AC225	.1195121E+00	
3	AC227	.3115306E-06	
4	AC228	.3027359E-01	
5	AG109M	.0000000E+00	
6	AG110	.0000000E+00	
7	AG110M	.0000000E+00	
8	AM241	.2327289E+04	
9	AM242	.0000000E+00	
10	AM242M	.0000000E+00	
11	AM243	.9354371E+01	
12	AM245	.2567534E-08	
13	AT217	.1195121E+00	
14	BA137M	.2053139E+04	
15	BI210	.9878327E-01	
16	BI211	.3115306E-06	
17	BI212	.2955630E+00	
18	BI213	.1195121E+00	
19	BI214	.1605505E+01	
20	BK249	.1839830E-03	
21	BK250	.9511000E-12	
22	C 14	.0000000E+00	
23	CD109	.0000000E+00	
24	CD113M	.0000000E+00	
25	CE144	.0000000E+00	
26	CF249	.1122483E+01	
27	CF250	.1660000E-02	
28	CF251	.0000000E+00	
29	CF252	.4050679E+00	
30	CM242	.3147419E-02	
31	CM243	.0000000E+00	
32	CM244	.1246465E+04	
33	CM245	.1758122E-02	
34	CM246	.5562000E-05	
35	CM247	.0000000E+00	
36	CM248	.1963147E-01	
37	CO 58	.0000000E+00	
38	CO 60	.1219268E-01	
39	CS134	.0000000E+00	
40	CS135	.0000000E+00	
41	CS137	.2171169E+04	
42	ES254	.0000000E+00	
43	EU150	.0000000E+00	
44	EU152	.3996000E-06	
45	EU154	.0000000E+00	
46	EU155	.0000000E+00	
47	FE 55	.0000000E+00	
48	FE 59	.1440000E-01	
49	FR221	.1195121E+00	
50	FR223	.4299004E-08	
51	H 3	.0000000E+00	
52	I129	.0000000E+00	
53	KR 85	.0000000E+00	
54	MN 54	.0000000E+00	
55	NB 95	.0000000E+00	
56	NB 95M	.0000000E+00	
57	NI 59	.0000000E+00	

Appendix G: Sample Output Files From PostORG

ornich93.sum FILE LISTING

58	NI 63	.1109000E-03
59	NP237	.6030983E+01
60	NP238	.0000000E+00
61	NP239	.9354371E+01
62	NP240	.0000000E+00
63	NP240M	.7911798E-09
64	PA231	.1560855E-05
65	PA233	.6030983E+01
66	PA234	.3419687E-04
67	PA234M	.2630038E-01
68	PB209	.1195121E+00
69	PB210	.9878327E-01
70	PB211	.3115306E-06
71	PB212	.2955630E+00
72	PB214	.1605505E+01
73	PD107	.0000000E+00
74	PM147	.1454935E-01
75	PO210	.7316528E-01
76	PO211	.8723459E-09
77	PO212	.1893680E+00
78	PO213	.1169344E+00
79	PO214	.1604504E+01
80	PO215	.3115306E-06
81	PO216	.2955630E+00
82	PO218	.1605505E+01
83	PR144	.0000000E+00
84	PU236	.0000000E+00
85	PU238	.6783259E+04
86	PU239	.7955260E+03
87	PU240	.7256365E+03
88	PU241	.5527624E+05
89	PU242	.5463922E+01
90	PU243	.0000000E+00
91	PU244	.7921432E-09
92	RA223	.3115306E-06
93	RA224	.2955630E+00
94	RA225	.1195121E+00
95	RA226	.1605505E+01
96	RA228	.3027359E-01
97	RH106	.0000000E+00
98	RN219	.3115306E-06
99	RN220	.2955630E+00
100	RN222	.1605505E+01
101	RU106	.0000000E+00
102	SB125	.0000000E+00
103	SB126	.0000000E+00
104	SB126M	.0000000E+00
105	SE 79	.0000000E+00
106	SM151	.0000000E+00
107	SN119M	.0000000E+00
108	SN121M	.0000000E+00
109	SN126	.0000000E+00
110	SR 90	.1289080E+04
111	TC 99	.1779000E+02
112	TE125M	.0000000E+00
113	TE127	.0000000E+00
114	TE127M	.0000000E+00

Appendix G: Sample Output Files From PostORG

ornlch93.sum FILE LISTING

115	TH227	.3072296E-06
116	TH228	.2955630E+00
117	TH229	.1195121E+00
118	TH230	.6524139E-03
119	TH231	.6085977E-02
120	TH232	.3426261E-01
121	TH234	.2630038E-01
122	TL207	.3106569E-06
123	TL208	.1061949E+00
124	TL209	.2581469E-02
125	TL210	.0000000E+00
126	U232	.2884000E+00
127	U233	.8934367E+02
128	U234	.8899415E+01
129	U235	.6567377E-02
130	U236	.2434953E-03
131	U237	.1285653E+01
132	U238	.3534738E-01
133	U240	.7911798E-09
134	Y 90	.1289068E+04
135	ZN 65	.0000000E+00
136	ZR 93	.0000000E+00
137	ZR 95	.0000000E+00
138		

Appendix G: Sample Output Files From PostORG

ornlc93f.u5 FILE LISTING

```

1  -1
2  -1
3  -1
4  BAS  WIPP BASELINE INVENTORY DATA DECAY OUT 75  Yrs
5  TIT  ornlch93
6  LIP  0 0 0
7  LIB  0 1 2 3 0 0 0 9 3 0 1 1
8  PHO  101 102 103 10
9  INP  1 1 -1 -1 1 1
10 DEC  5.00 1 2 5 1
11 DEC  10.00 2 3 5 0
12 DEC  15.00 3 4 5 0
13 DEC  20.00 4 5 5 0
14 DEC  25.00 5 6 5 0
15 DEC  35.00 6 7 5 0
16 DEC  45.00 7 8 5 0
17 DEC  55.00 8 9 5 0
18 DEC  65.00 9 10 5 0
19 DEC  75.00 10 11 5 0
20 RDA  ASSIGN TITLES TO OUTPUT VECTORS
21 HED  1 1993
22 HED  2 1998
23 HED  3 2003
24 HED  4 2008
25 HED  5 2013
26 HED  6 2018
27 HED  7 2028
28 HED  8 2038
29 HED  9 2048
30 HED  10 2058
31 HED  11 2068
32 RDA  PRINT NUCLIDE QUANTITIES IN CURIES
33 OPTA 6*8 5 17*8
34 OPTF 6*8 5 17*8
35 OPTL 6*8 5 17*8
36 RDA  OUTPUT COMMANDS
37 OUT  11 1 -1 0
38 END
39 1 561371 .3815E-05 270600 .1077E-04 551370 .2494E+02 0 0
40 1 260590 .2925E-06 280630 .1796E-05 611470 .1569E-04 0 0
41 1 380900 .9441E+01 430990 .1048E+04 390900 .2367E-02 0 0
42 2 892250 .2060E-05 892270 .4308E-08 892280 .1350E-07 0 0
43 2 952410 .6783E+03 952430 .4694E+02 952450 .4156E-15 0 0
44 2 852170 .7425E-13 832100 .7961E-06 832110 .7446E-15 0 0
45 2 832120 .2018E-07 832130 .6180E-08 832140 .3636E-07 0 0
46 2 972490 .1123E-06 972500 .2446E-18 982490 .2741E+00 0 0
47 2 982500 .1519E-04 982520 .7535E-03 962420 .9522E-06 0 0
48 2 962440 .1541E+02 962450 .1024E-01 962460 .1811E-04 0 0
49 2 962480 .4619E+01 872210 .6743E-09 872230 .1112E-15 0 0
50 2 932370 .8558E+04 932390 .4034E-04 932401 .7475E-17 0 0
51 2 912310 .3305E-04 912330 .2906E-03 912340 .1711E-10 0 0
52 2 912341 .3830E-10 822090 .2630E-07 822100 .1294E-02 0 0
53 2 822110 .1262E-13 822120 .2127E-06 822140 .4897E-07 0 0
54 2 842100 .1629E-04 842110 .9136E-20 842120 .1067E-17 0 0
55 2 842130 .9272E-17 842140 .5000E-14 842150 .1057E-19 0 0
56 2 842160 .8488E-12 842180 .5678E-08 942380 .3963E+03 0 0
57 2 942390 .1280E+05 942400 .3185E+04 942410 .5367E+03 0 0
58 2 942420 .1431E+04 942440 .4468E-04 882230 .6084E-11 0 0

```

Appendix G: Sample Output Files From PostORG

ornlc93f.u5 FILE LISTING

59	2	882240	.1856E-05	882250	.3049E-05	882260	.1624E+01	0	0
60	2	882280	.1294E-03	862190	.2395E-16	862200	.3205E-09	0	0
61	2	862220	.1044E-04	902270	.9997E-11	902280	.3607E-03	0	0
62	2	902290	.5619E+00	902300	.3233E-01	902310	.1145E-07	0	0
63	2	902320	.3125E+06	902340	.1136E-05	812070	.1631E-14	0	0
64	2	812080	.3606E-09	812090	.5681E-09	922320	.1348E-01	0	0
65	2	922330	.9231E+04	922340	.1425E+04	922350	.3039E+04	0	0
66	2	922360	.3764E+01	922370	.1575E-04	922380	.1052E+06	0	0
67	2	922400	.8545E-15	0	0				
68	3	631520	.2310E-08	0	0				
69	0								

ornlc93f.u5 FILE LISTING

[illegible]

C
C SANDIA NATIONAL LABORATORIES
Nuclear Waste Management Center
C
C
C

```

C      *
C      *   ISSUED BY
C      * SANDIA LABORATORIES,
C      * A PRIME CONTRACTOR
C      ***** TO THE
C      * UNITED STATES
C      * DEPARTMENT
C      * OF
C      * ENERGY
C      -----NOTICE-----
C *****
C *THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED*
C * BY THE UNITED STATES GOVERNMENT. NEITHER THE UNITED *
C * STATES NOR THE UNITED STATES DEPARTMENT OF ENERGY, *
C      NOR ANY OF THEIR EMPLOYEES,
C * NOR ANY OF THEIR CONTRACTORS, SUBCONTRACTORS, OR THEIR *
C * EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR *
C * ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE *
C      ACCURACY,
C      COMPLETENESS
C      OR USEFULNESS
C      OF ANY
C      INFORMATION,
C      APPARATUS,
C      PRODUCT
C      OR PROCESS
C      DISCLOSED,
C      OR REPRESENTS
C      THAT ITS
C      USE WOULD NOT
C      INFRINGE
C      PRIVATELY
C      OWNED
C      RIGHTS.
C *****

```

C2345678901234567890123456789012345678901234567890123456789012

G 1 2 3 4 5 6 7

```
THIS PROGRAM WILL PROCESS AN ORIGEN2 OUTPUT FILE
GENERATED FROM USING PreORG.FOR - IT WILL SUM ALL
NUCLIDE CONCENTRATIONS TO USE TO CREATE A NEW ORIG
INPUT FILE TO DECAY THE DATA INTO THE FUTURE

PLEASE ENTER ORIGEN2 OUTPUT FILENAME TO BE
PROCESSED WITHOUT EXTENSION (i.e. ornlch95 or
ornlc95a). THE FILENAME MUST BE 8 CHARACTERS
LONG, NOT INCLUDING THE EXTENSION .uf:
```

ornlch95.u6 ← USER ANSWER

ENTER BASE YEAR INITIALLY DECAYED ALL INVENTORY DATA
TO PRIOR TO DECAYING INTO FUTURE (i.e. 1993, 1995)
MAKE SURE THIS IS THE SAME YEAR USED WHILE RUNNING
PreORG for

1995 ← **USER ANSWER**

Appendix G: Sample Output Files From PostORG

ornlc93f.u5 FILE LISTING

```

164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206

WOULD YOU LIKE TO CREATE AN ORIGEN2 INPUT FILE
TO DECAY 1995 INVENTORY DATA INTO FUTURE?
y(YES) or n(NO)

y ←----- USER ANSWER

PLEASE CHOOSE HOW LONG INTO FUTURE YOU WISH
TO DECAY INVENTORY OUT TO:
(a) 75 yrs. (b) 100 yrs. (c) 500 yrs.
(d) 1000 yrs. (e) 5000 yrs. (f) 10,000 yrs.

f ←----- USER ANSWER

ELIMINATING EXCESS DATA FROM ORIGEN2 OUTPUT
CREATING OUTPUT FILES FOR EACH YEAR OF DATA

GENERATING MASTER LIST OF NUCLIDES FOR SITE DATA

ALPHABETIZING MASTER LIST OF NUCLIDES

SUMMING NUCLIDE CONCENTRATIONS

SORTING INTO WIPP BIR REV.1 Appendix E NUCLIDE LIST
INPUT FILENAME = ornlch95.SUM

WIPP BIR Appendix E Nuclide List File: ornlch95.bid

GENERATING ORIGEN2 INPUT FILE TO DECAY BASE YEAR
DATA INTO FUTURE

ORIGEN2 input filename = ornlc95f.u5
*****
TO RUN PROGRAM AGAIN ENTER 1
TO EXIT PROGRAM ENTER 2:
*****

2 ←----- USER ANSWER

```

Appendix G: Sample Output Files From PostORG

ornlc93f.sum FILE LISTING

	1993	1998	2003	2008	2013	2018	2028	2038	2048	2058	2068
1	1.196E-01	1.617E-01	2.039E-01	2.460E-01	2.880E-01	3.301E-01	4.142E-01	4.982E-01	5.821E-01	6.659E-01	7.496E-01
2	3.117E-07	5.483E-07	8.524E-07	1.214E-06	1.625E-06	2.078E-06	3.087E-06	4.201E-06	5.392E-06	6.640E-06	7.929E-06
3	3.028E-02	3.191E-02	3.287E-02	3.344E-02	3.378E-02	3.399E-02	3.418E-02	3.425E-02	3.427E-02	3.428E-02	3.429E-02
4	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
5	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
6	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
7	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
8	2.329E+03	2.703E+03	2.990E+03	3.209E+03	3.374E+03	3.496E+03	3.650E+03	3.722E+03	3.743E+03	3.732E+03	3.704E+03
9	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
10	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
11	9.361E+00	9.357E+00	9.352E+00	9.348E+00	9.344E+00	9.339E+00	9.330E+00	9.322E+00	9.313E+00	9.304E+00	9.295E+00
12	2.569E-09	5.113E-11	9.790E-13	1.875E-14	3.590E-16	6.874E-18	2.520E-21	9.242E-25	3.389E-28	1.242E-31	4.457E-35
13	1.196E-01	1.617E-01	2.039E-01	2.460E-01	2.880E-01	3.301E-01	4.142E-01	4.982E-01	5.821E-01	6.659E-01	7.496E-01
14	2.053E+03	1.829E+03	1.630E+03	1.452E+03	1.293E+03	1.152E+03	9.145E+02	7.259E+02	5.761E+02	4.573E+02	3.629E+02
15	9.881E-02	3.155E-01	5.005E-01	6.584E-01	7.931E-01	9.079E-01	1.089E+00	1.220E+00	1.314E+00	1.381E+00	1.428E+00
16	3.117E-07	5.489E-07	8.535E-07	1.216E-06	1.628E-06	2.082E-06	3.091E-06	4.206E-06	5.399E-06	6.649E-06	7.940E-06
17	2.957E-01	3.090E-01	3.011E-01	2.899E-01	2.784E-01	2.672E-01	2.459E-01	2.265E-01	2.089E-01	1.929E-01	1.783E-01
18	1.196E-01	1.617E-01	2.039E-01	2.460E-01	2.880E-01	3.301E-01	4.142E-01	4.982E-01	5.821E-01	6.659E-01	7.496E-01
19	1.606E+00	1.603E+00	1.599E+00	1.596E+00	1.592E+00	1.589E+00	1.582E+00	1.575E+00	1.568E+00	1.562E+00	1.555E+00
20	1.841E-04	3.525E-06	6.750E-08	1.293E-09	2.475E-11	4.739E-13	1.738E-16	6.372E-20	2.336E-23	8.567E-27	3.141E-30
21	9.520E-13	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
22	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
23	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
24	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
25	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
26	1.123E+00	1.112E+00	1.101E+00	1.091E+00	1.080E+00	1.069E+00	1.048E+00	1.028E+00	1.008E+00	9.878E-01	9.685E-01
27	1.661E-03	1.275E-03	9.779E-04	7.503E-04	5.757E-04	4.417E-04	2.600E-04	1.531E-04	9.010E-05	5.304E-05	3.122E-05
28	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
29	4.034E-01	1.090E-01	2.929E-02	7.874E-03	2.117E-03	5.689E-04	4.111E-05	2.971E-06	2.146E-07	1.551E-08	1.121E-09
30	3.150E-03	1.348E-06	5.765E-10	2.467E-13	1.055E-16	4.515E-20	8.265E-27	1.513E-33	0.000E+00	0.000E+00	0.000E+00
31	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
32	1.247E+03	1.030E+03	8.506E+02	7.025E+02	5.801E+02	4.791E+02	3.267E+02	2.228E+02	1.520E+02	1.036E+02	7.067E+01
33	1.759E-03	2.214E-03	2.664E-03	3.110E-03	3.551E-03	3.988E-03	4.848E-03	5.690E-03	6.515E-03	7.323E-03	8.114E-03
34	5.565E-06	6.629E-06	7.443E-06	8.067E-06	8.543E-06	8.907E-06	9.395E-06	9.677E-06	9.836E-06	9.924E-06	9.970E-06
35	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
36	1.965E-02	1.965E-02	1.965E-02	1.965E-02	1.965E-02	1.965E-02	1.965E-02	1.965E-02	1.965E-02	1.965E-02	1.965E-02
37	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
38	1.218E-02	6.311E-03	3.270E-03	1.694E-03	8.775E-04	4.546E-04	1.220E-04	3.274E-05	8.788E-06	2.359E-06	6.330E-07
39	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
40	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
41	2.170E+03	1.934E+03	1.723E+03	1.535E+03	1.367E+03	1.218E+03	9.667E+02	7.673E+02	6.090E+02	4.834E+02	3.836E+02
42	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
43	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
44	3.996E-07	3.097E-07	2.401E-07	1.861E-07	1.442E-07	1.118E-07	6.714E-08	4.033E-08	2.423E-08	1.455E-08	8.743E-09
45	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
46	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
47	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
48	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
49	1.196E-01	1.617E-01	2.039E-01	2.460E-01	2.880E-01	3.301E-01	4.142E-01	4.982E-01	5.821E-01	6.659E-01	7.496E-01
50	4.303E-09	7.566E-09	1.176E-08	1.676E-08	2.243E-08	2.868E-08	4.260E-08	5.798E-08	7.441E-08	9.163E-08	1.094E-07
51	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
52	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
53	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
54	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00

ornlc93f.sum FILE LISTING

55	NB 95	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
56	NB 95M	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
57	NI 59	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
58	NI 63	1.108E-04	1.067E-04	1.028E-04	9.898E-05	9.532E-05	9.180E-05	8.514E-05	7.896E-05	7.323E-05	6.791E-05	6.298E-05	6.035E+00	6.039E+00	6.044E+00	6.049E+00	6.054E+00	6.060E+00
59	NP237	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
60	NP238	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
61	NP239	9.361E+00	9.357E+00	9.352E+00	9.348E+00	9.344E+00	9.339E+00	9.330E+00	9.322E+00	9.313E+00	9.304E+00	9.295E+00	9.290E+00	9.285E+00	9.280E+00	9.275E+00	9.270E+00	9.265E+00
62	NP240	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
63	NP240M	7.917E-10	1.547E-09	2.302E-09	3.057E-09	3.813E-09	4.568E-09	5.323E-09	6.073E-09	6.818E-09	7.563E-09	8.308E-09	9.053E-09	9.798E-09	1.054E-08	1.299E-08	1.544E-08	1.789E-08
64	PA231	1.562E-06	2.257E-06	2.953E-06	3.650E-06	4.347E-06	5.044E-06	5.741E-06	6.438E-06	7.135E-06	7.832E-06	8.529E-06	9.226E-06	9.923E-06	1.062E-05	1.307E-05	1.552E-05	1.797E-05
65	PA233	6.034E+00	6.039E+00	6.044E+00	6.049E+00	6.054E+00	6.059E+00	6.064E+00	6.069E+00	6.074E+00	6.079E+00	6.084E+00	6.089E+00	6.094E+00	6.099E+00	6.104E+00	6.109E+00	6.114E+00
66	PA234	3.422E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05	4.600E-05
67	PA234M	2.632E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02	3.538E-02
68	PB209	1.196E-01	1.617E-01	2.039E-01	2.460E-01	2.880E-01	3.301E-01	3.722E-01	4.143E-01	4.564E-01	4.985E-01	5.406E-01	5.827E-01	6.248E-01	6.669E-01	7.090E-01	7.511E-01	7.932E-01
69	PB210	9.883E-02	3.154E-01	5.003E-01	6.581E-01	7.927E-01	9.075E-01	1.088E+00	1.219E+00	1.313E+00	1.380E+00	1.427E+00	1.474E+00	1.521E+00	1.568E+00	1.615E+00	1.662E+00	1.709E+00
70	PB211	3.117E-07	5.489E-07	8.535E-07	1.216E-06	1.628E-06	2.082E-06	2.536E-06	3.091E-06	3.646E-06	4.201E-06	4.756E-06	5.311E-06	5.866E-06	6.421E-06	6.976E-06	7.531E-06	8.086E-06
71	PB212	2.957E-01	3.090E-01	3.011E-01	2.899E-01	2.784E-01	2.672E-01	2.459E-01	2.265E-01	2.089E-01	1.929E-01	1.783E-01	1.605E+00	1.602E+00	1.599E+00	1.596E+00	1.592E+00	1.589E+00
72	PB214	1.606E+00	1.603E+00	1.599E+00	1.596E+00	1.592E+00	1.589E+00	1.585E+00	1.582E+00	1.579E+00	1.575E+00	1.568E+00	1.562E+00	1.555E+00	1.548E+00	1.541E+00	1.534E+00	1.527E+00
73	PD107	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
74	PM147	1.455E-02	3.883E-03	1.036E-03	2.765E-04	7.379E-05	1.969E-05	1.108E-05	4.02E-06	9.985E-08	7.110E-09	5.063E-10	3.606E-11	2.531E-12	1.722E-13	1.181E-14	8.01E-16	5.21E-17
75	PO210	7.320E-02	3.155E-01	5.005E-01	6.585E-01	7.932E-01	9.080E-01	1.089E+00	1.220E+00	1.314E+00	1.381E+00	1.428E+00	1.475E+00	1.522E+00	1.569E+00	1.616E+00	1.663E+00	1.710E+00
76	PO211	8.727E-10	1.537E-09	2.390E-09	3.404E-09	4.557E-09	5.828E-09	7.265E-09	8.654E-09	1.012E-08	1.161E-08	1.317E-08	1.478E-08	1.645E-08	1.818E-08	2.000E-08	2.180E-08	2.366E-08
77	PO212	1.894E-01	1.980E-01	1.929E-01	1.857E-01	1.784E-01	1.712E-01	1.575E-01	1.451E-01	1.339E-01	1.236E-01	1.143E-01	1.051E-01	9.64E-02	8.74E-02	7.89E-02	7.09E-02	6.34E-02
78	PO213	1.170E-01	1.582E-01	1.995E-01	2.406E-01	2.818E-01	3.230E-01	4.052E-01	4.874E-01	5.695E-01	6.515E-01	7.334E-01	8.153E-01	8.972E-01	9.791E-01	1.061E-00	1.142E-00	1.223E-00
79	PO214	1.605E+00	1.602E+00	1.599E+00	1.595E+00	1.592E+00	1.588E+00	1.585E+00	1.582E+00	1.579E+00	1.575E+00	1.568E+00	1.561E+00	1.554E+00	1.547E+00	1.540E+00	1.533E+00	1.526E+00
80	PO215	3.117E-07	5.489E-07	8.535E-07	1.216E-06	1.628E-06	2.082E-06	2.536E-06	3.091E-06	3.646E-06	4.201E-06	4.756E-06	5.311E-06	5.866E-06	6.421E-06	6.976E-06	7.531E-06	8.086E-06
81	PO216	2.957E-01	3.090E-01	3.011E-01	2.899E-01	2.784E-01	2.672E-01	2.459E-01	2.265E-01	2.089E-01	1.929E-01	1.783E-01	1.605E+00	1.602E+00	1.599E+00	1.596E+00	1.592E+00	1.589E+00
82	PO218	1.606E+00	1.603E+00	1.599E+00	1.596E+00	1.592E+00	1.589E+00	1.585E+00	1.582E+00	1.579E+00	1.575E+00	1.568E+00	1.562E+00	1.555E+00	1.548E+00	1.541E+00	1.534E+00	1.527E+00
83	PR144	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
84	FU236	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
85	FU238	6.787E+03	6.524E+03	6.278E+03	6.029E+03	5.795E+03	5.571E+03	5.347E+03	5.123E+03	4.900E+03	4.677E+03	4.454E+03	4.231E+03	4.008E+03	3.785E+03	3.562E+03	3.339E+03	3.116E+03
86	FU239	7.960E+02	7.959E+02	7.958E+02	7.957E+02	7.956E+02	7.955E+02	7.954E+02	7.953E+02	7.952E+02	7.951E+02	7.950E+02	7.949E+02	7.948E+02	7.947E+02	7.946E+02	7.945E+02	7.944E+02
87	FU240	7.260E+02	7.263E+02	7.264E+02	7.265E+02	7.266E+02	7.267E+02	7.268E+02	7.269E+02	7.270E+02	7.271E+02	7.272E+02	7.273E+02	7.274E+02	7.275E+02	7.276E+02	7.277E+02	7.278E+02
88	FU241	5.531E+04	4.348E+04	3.418E+04	2.687E+04	2.112E+04	1.660E+04	1.266E+04	9.00E+03	6.340E+03	3.917E+03	2.421E+03	1.496E+03	8.66E+02	5.16E+02	3.05E+02	1.78E+02	1.03E+02
89	FU242	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00	5.466E+00
90	FU243	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
91	FU244	7.928E-10	1.549E-09	2.305E-09	3.061E-09	3.818E-09	4.574E-09	5.329E-09	6.085E-09	6.841E-09	7.597E-09	8.353E-09	9.109E-09	9.865E-09	1.062E-08	1.214E-08	1.366E-08	1.518E-08
92	RA223	3.117E-07	5.489E-07	8.535E-07	1.216E-06	1.628E-06	2.082E-06	2.536E-06	3.091E-06	3.646E-06	4.201E-06	4.756E-06	5.311E-06	5.866E-06	6.421E-06	6.976E-06	7.531E-06	8.086E-06
93	RA224	2.958E-01	3.091E-01	3.011E-01	2.899E-01	2.784E-01	2.672E-01	2.459E-01	2.265E-01	2.089E-01	1.929E-01	1.783E-01	1.605E+00	1.602E+00	1.599E+00	1.596E+00	1.592E+00	1.589E+00
94	RA225	1.196E-01	1.617E-01	2.039E-01	2.460E-01	2.880E-01	3.301E-01	3.722E-01	4.143E-01	4.564E-01	4.985E-01	5.406E-01	5.827E-01	6.248E-01	6.669E-01	7.090E-01	7.511E-01	7.932E-01
95	RA226	1.606E+00	1.603E+00	1.599E+00	1.596E+00	1.592E+00	1.589E+00	1.585E+00	1.582E+00	1.579E+00	1.575E+00	1.568E+00	1.562E+00	1.555E+00	1.548E+00	1.541E+00	1.534E+00	1.527E+00
96	RA228	3.030E-02	3.191E-02	3.287E-02	3.345E-02	3.378E-02	3.399E-02	3.418E-02	3.425E-02	3.425E-02	3.425E-02	3.425E-02	3.425E-02	3.425E-02	3.425E-02	3.425E-02	3.425E-02	3.425E-02
97	RH106	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
98	RN219	3.117E-07	5.489E-07	8.535E-07	1.216E-06	1.628E-06	2.082E-06	2.536E-06	3.091E-06	3.646E-06	4.201E-06	4.756E-06	5.311E-06	5.866E-06	6.421E-06	6.976E-06	7.531E-06	8.086E-06
99	RN220	2.957E-01	3.090E-01	3.011E-01	2.899E-01	2.784E-01	2.672E-01	2.459E-01	2.265E-01	2.089E-01	1.929E-01	1.783E-01	1.605E+00	1.602E+00	1.599E+00	1.596E+00	1.592E+00	1.589E+00
100	RN222	1.606E+00	1.603E+00	1.599E+00	1.596E+00	1.592E+00	1.589E+00	1.585E+00	1.582E+00	1.579E+00	1.575E+00	1.568E+00	1.562E+00	1.555E+00	1.548E+00	1.541E+00	1.534E+00	1.527E+00
101	RU106	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
102	SB125	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
10																		

ornlc93f.sum FILE LISTING

G-13

Intentionally Left Blank

WIPP
DISTRIBUTION
SAND96-2736

S. Chakraborti
SAIC
2109 Air Park Road SE
Albuquerque, NM 87106

P. Drez
8816 Cherry Hills Road NE
Albuquerque, NM 87111

Tech Reps Inc. (2)
Attn: M. Minahan
L. Robledo
5000 Marble NE, Suite 222
Albuquerque, NM 87110

Internal

<u>MS</u>	<u>Org.</u>	
1175	9364	J. Liscum-Powell
0571	5914	T. W. Laub
1335	6801	M. S. Y. Chu
1395	6800	L.E. Shephard
1328	6848	K. Cochrane (4)
1328	6848	H-N. Jow
1328	6848	L. C. Sanchez (8)
1328	6848	B. Schlapper
1328	6849	J. S. Rath
1399	6850	M. Brady
1326	6851	R. Aguilar
1326	6851	G. E. Barr
1328	6849	F. J. Davis
0734	6472	R. D. Waters
1145	9363	P. J. Griffin
1145	9363	G. Harms
1175	9364	C. A. Ottinger
1330	6811	K. Hart (2)
1330	4415	NWM Library (2)
9018	8940-2	Central Technical Files
0899	4916	Technical Library (5)
0619	12690	Review and Approval Desk (2), For DOE/OSTI

Intentionally Left Blank