**RECON: A Computer Program for
Analyzing Repository Economics
Documentation and User's Manual
Revision 1**

**L.L. Clark
M.E. Schutz
A.T. Luksic**

July 1985

**Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830**

**Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute**

DISCLAIMER

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

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161

NTIS Price Codes
Microfiche A01

Printed Copy

Pages	Price Codes
001-025	A02
026-050	A03
051-075	A04
076-100	A05
101-125	A06
126-150	A07
151-175	A08
176-200	A09
201-225	A010
226-250	A011
251-275	A012
276-300	A013

RECON: A COMPUTER PROGRAM FOR ANALYZING
REPOSITORY ECONOMICS

DOCUMENTATION AND USER'S MANUAL

REVISION 1

L. L. Clark
M. E. Schutz
A. T. Luksic

July 1985

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

PROGRAM DESCRIPTION	6.1
INSTRUCTIONS FOR USING THE PREPROCESSOR	6.3
7.0 REFERENCES	7.1
APPENDIX A - MODEL VARIABLE DESCRIPTIONS	A.1
APPENDIX B - RECON MODEL LISTING	B.1

1.0 INTRODUCTION

In 1981, the Office of Nuclear Waste Isolation commissioned the Pacific Northwest Laboratory to prepare a consistent set of cost estimates for constructing and operating nuclear waste repositories in different media. To aid in accomplishing this task, a computer model named RECON was created to parametrically calculate repository costs. During 1982, the model was further developed under the direction of the Office of National Waste Terminal Storage (NWTs) Integration.

In May of 1983, initial documentation and a user's manual for the RECON program was issued. Since that time additional model revisions and enhancements have been made under the direction of the Geologic Repository Deployment Division of the Office of Civilian Radioactive Waste Management. This document is a revision of the earlier 1983 document and includes updates and enhancements incorporated in the model since that time.

In the following sections the model is described in general terms, the flow of calculations in the model is shown, the structure and equations used in the various subroutines are defined, and a detailed set of input instructions is provided in a user's manual. The appendices list all the model variable descriptions, list the complete program, and present output for a sample case.

2.0 GENERAL RECON MODEL DESCRIPTION

The RECON Model is a computer program designed to estimate costs of mined geologic repositories based on parametric data input. Given a reference design and database, RECON can readily estimate costs of design parameter variations. Although the model generates estimates with less precision and detail than those generated by an architect engineer, it can generate first-order estimates much more quickly and cheaply and can be used to focus attention on areas where more detailed design efforts are cost-effective and necessary.

It should be emphasized that the model described in this document has a very complex and extensive set of data requirements and makes a number of implicit assumptions. Attempts to use RECON without a thorough understanding of repository design limitations could lead to erroneous results and conclusions. The authors recommend, therefore, that they be consulted by anyone desiring to implement the model.

As previously mentioned, RECON uses parametric data input to describe the repository. Basic model input parameters describe facilities, construction times, shafts, mine design, emplacement limitations, waste flows available for disposal, waste processing parameters (labor, materials, utility, and equipment requirements), facility construction costs and unit labor, materials, and utility and equipment costs. Using this information, the model determines the number of facility modules for receiving, packaging, transporting, and emplacing the wastes. Based on the facility requirements, the model calculates labor, material, and equipment requirements. Labor requirements are calculated in shifts based on available working days. From the shifts and days worked, utility requirements are calculated. Equipment replacement requirements are calculated based on equipment life (stated in years or units processed) and processing rates. All of the above requirements are calculated year by year for each waste type thereby simulating actual repository operations.

Excavation of the mine is modeled assuming initial shaft, shaft pillar, and main corridor excavations followed by room excavations needed to emplace waste volumes received the following year. Hoisting, surface storage, and offsite disposal tonnages are also calculated. The user has the option of

allowing the model to calculate requirements and costs for mining labor materials and equipment for the above operations or the user can input unit costs (\$/ton) for the above rock operations. Mine excavation requirements are based on input data describing room, panel, and corridor dimensions and spacings. Input density of waste emplacement in rooms is based on minimum waste spacing (calculated externally using very-near-field thermal criteria or rock stress criteria). Any required expansion of this spacing due to thermal limits is calculated internally in the model using waste package thermal densities, and far-field and near-field limiting thermal densities. Annual excavation requirements are then calculated for each waste type using the emplacement density, annual waste units requiring disposal, and room and corridor dimensions.

Backfilling operations are modeled using input data that determine startup of these operations and annual backfilling rates. Based on excavated volumes, the model determines the annual rock mass backfilled and the duration of backfilling operations if these extend beyond waste emplacement operations. Cost calculation methodology is specified by the same option governing the mining cost calculations discussed above.

Startup of decommissioning and sealing operations is dynamically determined by either the end of waste operations or the end of backfilling operations. The model also has provisions to allow for an interim caretaker period. Waste operations cease either at the end of an input time period or when the total underground area requirement exceeds a maximum size. Backfilling operations end when all excavated rooms are backfilled. During decommissioning operations, the remaining corridors and shaft pillar areas are backfilled, the shafts are sealed, and the surface facilities are decommissioned. Backfilling costs are determined as explained above. The model uses unit sealing costs together with shaft diameters to calculate total sealing costs. Decommissioning costs are calculated based on an input percentage of total surface facility costs.

Also included in the costs, but not specified above are land, licensing, site preparation, and startup costs. Land, licensing, and site preparation costs are input determined, while startup costs are based on an input percentage of total construction costs.

Contingencies account for unidentifiable costs of labor, materials, and equipment and other unspecified items associated with uncertainties in the design. These are calculated by multiplying input percentages by construction costs, waste operations costs, rock operations costs, and decommissioning costs.

Output tables show summaries of total construction, waste operations, rock operations, and decommissioning costs by waste type. Further breakdowns of costs within these categories are shown in backup tables as a function of time. Levelized unit costs are calculated per unit mass, unit volume, and per package using an input-specified discount rate. Supporting tables show summaries of emplacement descriptions, rock logistics, waste operations logistics, and backfilling logistics.

3.0 FLOWCHART DESCRIPTION

This section outlines the sequence of operations discussed in the general model descriptions in Section 2.0. This sequence is shown graphically in Figure 3.1. The controlling calculational routine is called HUB. This routine calls subroutines that model operations in the various repository components and calculate costs. The figure shows the sequence of operations necessary to provide the data needed for each subroutine. Table 3.1 lists the subroutines and describes the main operations in each. A more detailed explanation of each subroutine is given in Section 4.0.

FIGURE 3.1. Calculational Flow for the RECON Model

TABLE 3.1. General Subroutine Descriptions for the RECON Model

<u>Subroutine</u>	<u>Main Operations</u>
INPUT	A. Read all Input Data
WASTOP	A. Calculate Modules for: Remote and Contact Handled Packaging Remote and Contact Handled Transport Remote and Contact Handled Emplacement Remote Handled Emplacement Excavations Remote and Contact Handled Waste Shafts B. Calculate Shift Requirements for RH Waste Operations CH Waste Operations C. Calculate Labor Requirements for Waste Operations and Material Requirements including package materials for Waste Operations D. Calculate Emplacement Configurations and Area Requirements E. Determine Year When Repository is Full F. Determine Equipment Requirements for Waste Operations G. Calculate and Sum Labor, Materials, and Equipment Costs for Each Waste Operation
EMPLACE PPUT, SPUT, TPUT, SSTOR	A. Determine if Coemplacement is Required B. If Coemplacement is Required, Determine Waste Classification C. Determine Number of Emplacements, Rooms Required, and Rooms Remaining Available for Emplacement D. Determine any onsite storage requirements
MINDEV	A. Calculate Rock Volume and Mass Mined B. Calculate Labor, Materials and Equipment Requirements for Mining C. Calculate Backfilling Done During Waste Operations D. Calculate Other Rock Operations Logistics During Waste Operations

TABLE 3.1. (Contd)

<u>Subroutine</u>	<u>Main Operations</u>
RECEVE	<ul style="list-style-type: none"> A. Calculate Rail and Truck Shipments Received B. Calculate Number of Receiving Modules Required C. Calculate Receiving and Packaging Facility Costs
SHAFT	<ul style="list-style-type: none"> A. Calculate Shaft Usage for Waste and Mined Rock Hoisting Shafts B. Calculate Number of Rock Hoisting Shafts Required
ROCKOP	<ul style="list-style-type: none"> A. Calculate Backfilling Masses and Resource Requirements B. Calculate Rock Masses Hoisted and Lowered and Associated Resource Requirements C. Calculate Surface Rock Storage and Handling Logistics and Associated Resource Requirements
UTILITY	<ul style="list-style-type: none"> A. Calculate Electricity, Steam, Water, and Diesel Fuel Requirements B. Calculate Electricity, Steam, Water, and Diesel Fuel Costs
SURFST	<ul style="list-style-type: none"> A. Distribute Capital Costs over Time and Waste Type for: <ul style="list-style-type: none"> Support Facilities Underground Facilities Transportation Capital Rock Operations Capital Emplacement Capital B. Calculate Support Facility Equipment Replacement Costs C. Calculate Site Preparation Costs D. Calculate Labor and Materials Requirements for Support Facilities
PREOP	<ul style="list-style-type: none"> A. Calculate Costs for: <ul style="list-style-type: none"> Land Acquisition Licensing Architect-Engineer Services Startup Operations
POSTOP	<ul style="list-style-type: none"> A. Calculate: <ul style="list-style-type: none"> Shaft Sealing Cost Facility Decommissioning Costs Surveillance Costs

TABLE 3.1. (Contd)

WAFAM	A. Calculate Cost Allocation Factors
REPLAS	A. Calculate Equipment Replacement Requirements
OUTPUT	A. Calculate Appropriate Units for Tables B. Develop Output Tables C. Calculate Totals for Output Tables
OUT22	A. Calculate Costs for Alternate Set of Cost Categories B. Develop Output Table 22
FUNCTIONS:	
HOLDC	A. Functional Form for Calculating Hole Drilling Costs
HOLMC	A. Functional Form for Calculating Costs for Engineered Barriers in Boreholes
SKFUNC	A. Functional Form for Estimating Shaft Construction and Sealing Costs

4.0 DETAILED SUBROUTINE DESCRIPTIONS AND EQUATION DEFINITIONS

This section describes in detail the operations performed in each subroutine and function and defines the general equations used to perform the calculations.

THE INPUT SUBROUTINE

This subroutine reads in all of the data required to perform the calculations. Because the user's manual in Section 5.0 describes each of the input variables and the data options, this routine is not discussed further in this section.

THE WASTOP SUBROUTINE

Definitions - The waste operations subroutine (WASTOP) calculates the parameters and costs of waste operations at the repository. These operations are defined to include the following components: waste receiving, waste packaging, waste transport within the repository, waste emplacement, and excavations in the rooms for waste emplacement.

Calculations performed in WASTOP use the concept of a modular unit of operation. The capital cost, capacity, equipment parameters, labor parameters, and materials parameters for each modular unit of operation are read in as input variables. Every module of a particular type (e.g. packaging module) has the same capacity. The following module definitions are used for each waste operations component:

- Receiving - This is a facility module including receiving bays, shipping container decontamination stations, unloading facilities and equipment, and waste transfer and surge storage facilities and equipment. Remote and contact-handled modules are specified independently in the model.
- Packaging - This is a remote-operations facility module and includes a hot cell, transfer ports, and equipment to perform desired operations such as decanning, removal and insertion of waste, welding,

leak testing, decontamination and movement of packaged waste to the waste shaft. This facility is only used for remote-handled waste. If a facility is required, facility costs are assumed to be distributed over all remote-handled wastes.

- Transportation - This is an equipment module consisting of a single transporter that moves the packaged waste from the shaft stations underground to the emplacement area. Remote- and contact-handled modules are specified separately.
- Emplacement - This is an equipment module consisting of a single piece of equipment to place the waste package in its final emplacement location. Some designs incorporate transportation and emplacement in a single piece of equipment. Remote- and contact-handled modules are specified separately.
- Emplacement Excavation - This is an equipment module consisting of a single piece of excavation equipment with associated equipment to transfer excavated media to holding areas. This equipment is usually borehole drilling and rock-mucking equipment. Excavation modules are not needed if all wastes are simply placed in rooms.

The data input associated with each module is described in Section 5.0 and includes the following:

- labor - requirement and wage level
- materials - requirement and unit price
- equipment - capacity per shift, lifetime, and unit cost
- capital cost.

Calculations - The WASTOP subroutine first calculates annual module operating times required for remote- and contact-handled waste receiving. Receiving module operating time is calculated as follows:

$$\begin{aligned} \text{Receiving module operating} &= \sum_{K=1}^{NWT} TTTR(K) * \frac{RECUNT(K,I)}{RCCAP(K)} * CRRAIL(K,I) \\ \text{time in shifts} &+ TTTK(K) * \frac{RECUNT(K,I)}{TCCAP(K)} * (1 - CRRAIL(K,I)) \end{aligned} \quad (1)$$

where

RECUNT = annual waste receipts

RCCAP = rail cask capacity (in waste units)

TCCAP = truck cask capacity (in waste units)

CRRAIL = percentage of cask shipments made by rail

TTTR = the turnaround time in eight-hour shifts for rail casks for the receiving module assuming 100% rail cask usage.

TTTK = the turnaround time in eight-hour shifts for truck casks for the receiving module assuming 100% truck cask usage

K = waste type index (NWT = total number of waste types)

I = year index.

Packaging module operating time is independent of receiving module capacity and is calculated from input module capacity as follows:

$$\text{Packaging module operating time} = \sum_{K=1}^{NWT} \frac{\text{RECUNT}(K,I)}{\text{WUPP}(K) * \text{PKGCAP}(K)} \quad (2)$$

where

PKGCAP(K) = packaging module capacity in packages readied for emplacement per day.

WUPP(K) = waste units consolidated per package

After determining the module operating times for the above modules WASTOP calculates the number of modules needed to satisfy the waste input requirements according to the following equation:

$$\text{Number of modules (M,I)} = \frac{\text{Operating time requirements(M,I)}}{\text{Maximum DPY} * \text{NALSH} * \text{NBPM}} \quad (3)$$

where

DPY = days per year

NALSH = maximum number of shifts per day allowed

NBPM = number of parallel receiving stations per module

M = either remote- or contact-handled wastes.

The model compares the number of modules calculated with the maximum number of modules allowed (input). If the maximum is exceeded, the number of waste units available for disposal is proportionately reduced, the module calculations are redone, and a flag is printed on the appropriate output table. This operation is performed sequentially for receiving and packaging.

Shift requirements are calculated separately for remote- and contact-handled wastes based on the following:

$$\text{Shift requirement (M,I)} = \frac{\text{total operating time requirement (M,I)}}{\text{number of modules (M,I)} * \text{DPY} * \text{NBPM}} \quad (4)$$

where

M = either remote- or contact-handled wastes

Shift requirements for the packaging facility are assumed to be the same as for remote-handled receiving.

WASTOP next calculates the number of shifts required to transfer the wastes underground in the waste shafts according to the following equation:

$$\text{Shaft shift requirement (M,I)} = \frac{\text{number of packages (M,I)}}{\text{NPKPT(M)} * \text{MTRPSD(M)} * \text{DPY}} \quad (5)$$

where

NPKPT = number of packages per trip

MTRPSD = maximum trips per shift

DPY = days per year

M = remote- or contact-handled waste

I = year index

The model logic allows the user to specify the number of waste shafts to be used (in which the waste processed per year is reduced if the shaft capacity is exceeded) or the program to calculate the number of shafts used (in which case additional waste shafts are installed if shaft capacity is exceeded). The user can also choose which type of waste shafts should be used (remote or contact or both). If a remote waste shaft is specified but no contact-handled waste shaft is specified, the program uses the largest shaft shift value calculated above to represent the combined waste shaft. If the number of shifts required for shaft operation is within the maximum, but greater than shift requirements for receiving and packaging, the largest shift requirement is used for all components. The model assumes that all subsequent underground operations are governed by this shift requirement. If the shift requirements calculated for shaft operations exceed the maximum allowable shift requirements, the number of waste units available for disposal is proportionately reduced (if the waste shafts are user-specific), the number of modules and shafts is recalculated and a flag is printed on the appropriate output table.

The module requirements for transportation, emplacement, and emplacement excavation are based on the following equation:

$$\text{Modules (K,I)} = \frac{\text{units processed (K,I)}}{\text{equipment capacity (M)} * \text{shifts operated} * \text{days per year}} \quad (6)$$

Manpower requirements are developed from the shifts operated for each module component based on the following:

$$\text{Labor requirement in man-years (I,L)} = \text{WOLBRR} * \text{SFTR(L)} * \text{NRMR (I)} \quad (7)$$

where

WOLBRR = labor requirement per module per shift

SFTR = fraction of shift operated

NRMR = number of modules required

L = shift operated

I = year index

K = waste type index

Materials requirements for each component are calculated from the following:

$$\begin{aligned} \text{Materials requirements (K,I)} &= \text{materials required per process unit} \\ &\quad * \text{number of process units} \end{aligned} \quad (8)$$

All of the above calculations have been performed for just one year of operation. Next, the model determines the waste emplacement density and the resulting area requirements. The total area requirements are summed year by year and compared with a maximum emplacement area allowable. The allowable emplacement area is the total underground area less the central pillar, main corridor, and exclusion areas. The exclusion area is an unminable fraction of the total underground area. When the maximum area allowable has been exceeded, the model recalculates the waste emplacements and area requirements for that year and stops emplacement operations. The waste emplacement density is calculated from the number of rows per room (input) and proposed pitch (input) as follows:

$$\begin{aligned} \text{Number of packages} \\ \text{per room (K,I)} \end{aligned} = \left[\frac{\text{RMLN(K)} - 2 * \text{CLSEMP(K)}}{\text{pitch(K,I)}} + 1 \right] * \text{rows per room(K)} \quad (9)$$

where

K = waste type

I = year

RMLN = room length

CLSEMP = distance from cross corridor to first emplacement
pitch = centerline distance between emplacements.

The near-field and far-field heat loads are calculated using the following relationships:

$$\text{Near-field heat load} = \sum_{K=1}^{NWT} NPKGPR(K,I) * TDWI(K) * \frac{XRASHO(K) * M2ACRE}{RMLN(K) * RMWTH(K)} \quad (10)$$

where

NPKGPR = number of packages per room

TDWI = thermal density per package in kilowatts

XRASHO = extraction ratio (i.e., room width/centerline distance
between rooms)

M2ACRE = square meters per acre constant

RMLN = room length

RMWTH = room width

K = waste type index (NWT = total number of waste types)

I = year index

$$\text{Far-field heat load} = \sum_{K=1}^{NWT} \frac{XRASHO(K) * M2ACRE}{RMWTH(K) * PANWTH} NPKGPR(K) * TDPP(K) \quad (11)$$

where

PANWTH = panel width

If the near-field or far-field heat loads exceed the near- or far-field thermal limits (input), the program recalculates the pitch between canisters and the resulting emplacement density such that the thermal limit is met. The model then determines if coemplacement options have been selected, and if so, calculates the emplacement densities of the wastes emplaced with the primary wastes.

The WASTOP routine then calculates equipment requirements for the waste operations components. These calculations employ a subroutine named REPLAS. The number of equipment modules needed each year are calculated (see Equation (5)) and fed, along with the units processed, the equipment capacity and the equipment life, to REPLAS. The REPLAS subroutine keeps track of the usage of each piece of equipment during the period of waste operations and determines, based on equipment life, when replacements are needed. These requirements are then fed back to the subroutine calling REPLAS.

In calculating labor, equipment, and materials requirements for emplacement excavation operations, one of four emplacement methods can be employed: 1) hole drilling in room floors, 2) simple placement in rooms 3) trenching in room floors, and 4) boreholes in rock pillars between rooms. The method of emplacing each waste type is specified by the user. For the hole drilling methods, two cost calculation options are available. Option 1 is similar to previous calculations in that the user must specify the unit equipment, labor, and material requirements and costs for the hole drilling operations. (Emplacement methods 2 and 3 must use this type of input.) In Option 2, the user specifies that a unit hole drilling cost be calculated. This unit cost is then multiplied by the units requiring hole emplacement to develop the total hole drilling cost. The unit cost calculated in Option 2 is based on functional relationships shown below which were developed from previous unpublished hole drilling cost estimates for vertical boreholes. A separate functional form is shown for salt, since a better fit to the data was found using this form.

Basalt and Granite Media

$$\text{Unit hole drilling cost in dollars per meter of depth} = A * d^B \quad (12)$$

where

A and B = parameter constants:

d = hole diameter

Salt Medium

$$\text{Unit hole drilling cost in dollars per meter of depth} = C * e^{kd} \quad (13)$$

where

C and k = parameter constants

e = natural logarithm = 2.71828

d = hole diameter.

The above unit costs are multiplied by hole depth in meters to develop the final unit hole drilling cost. The hole diameters used above are calculated in the model using the relationship below:

$$\text{Hole diameter} = \text{package diameter} + 2 * (\text{water barrier thickness} + \text{sleeve thickness} + \text{backfill thickness}) \quad (14)$$

The water barrier, sleeve, and backfill thicknesses are specified by the user.

The rock masses excavated for hole drilling or trenching are calculated and later added to the rock masses mined in excavating shafts, rooms, and corridors to develop total rock masses hoisted and stored on the surface.

The final calculations in WASTOP determine the labor, materials, and equipment costs for each waste operations component using the requirements developed above, in the equations shown below:

$$\text{Labor cost}_N(K,I) = \sum_{L=1}^{IS} (\text{labor requirement}_N(I,L) * \text{labor cost}_N * \text{shift cost differential (L)} * \text{overhead factor} * \text{LCAF}(K,I)) \quad (15)$$

where

N = waste operations component (e.g., receiving, packaging, etc.)

K = waste type index

I = year index

L = shift

IS = number of shifts

LCAF = labor cost allocation factor = units processed by waste
type/total units processed by year

$$\begin{aligned} \text{Materials cost}_N(K,I) &= \text{materials requirements}_N(K,I) \\ &\quad * \text{materials costs}_N(K,I) \end{aligned} \tag{16}$$

$$\begin{aligned} \text{Equipment cost}_N(K,I) &= \text{new equipment requirements}_N(K,I) \\ &\quad * \text{unit equipment cost}_N \end{aligned} \tag{17}$$

The above cost calculations for waste operations are transferred to the HUB subroutine, which performs the various summations.

THE COEMPLACEMENT SUBROUTINES

The RECON model allows the user to emplace individual waste types in separate rooms or to coemplace or co-mingle different waste types in the same room. If only one type of waste is emplaced per room, the model attributes all of the excavation, hauling and rock storage costs associated with that room to that waste type.

When different waste types are coemplaced in the same room, one of these waste types is generally a heat-producing remote-handled waste type. The spacing of this waste type is limited by thermal considerations. This heat-generating waste type is referred to as the primary waste type. The coemplacement logic developed in RECON allows minimal-heat-generating, remote handled waste types and contact-handled waste types to be emplaced in the same room as

the heat generating primary waste types. In the coemplacement scenarios, all of the rock excavation, transport, and storage costs are attributed to the primary waste type. However, costs for borehole drilling and emplacement in these rooms are attributed to each waste type. Figure 4.1 shows a flow diagram for the coemplacement logic. Starting in the upper left-hand corner of the diagram the logic determines if coemplacement is needed. This is done by checking the emplacement method index shown in Table 4.1. The emplacement method index numbers identify the various waste emplacement options available. An index number is specified for each waste type. If the index number is 1 through 4, then the waste is placed by itself in a room and coemplacement is not used. If the emplacement method index number is 5 through 15, then the waste type is coemplaced according to the option specified. For example, if the emplacement method index number is 6, then primary wastes would be placed in vertical boreholes, secondary wastes would also be emplaced in vertical boreholes between the primary waste boreholes, and tertiary wastes would be placed in the rooms after the boreholes have been filled with the primary and secondary wastes. The waste classification index for classifying wastes into the primary, secondary, and tertiary categories is shown in Table 4.2. High-level and spent fuel wastes are remote-handled, high heat-generating wastes and are classed as primary wastes whether or not they are coemplaced with other wastes. Remote-handled, low heat-generating TRU waste that is not coemplaced with other wastes is classified as a primary waste. If the remote-handled TRU waste is coemplaced with other waste, it is classified as a secondary waste. Contact-handled TRU waste that is not coemplaced with other wastes is classified as a primary waste. Contact handled TRU waste which is coemplaced with other wastes is classified either as a secondary or a tertiary waste depending on whether remote-handled TRU waste is present. Referring again to Figure 4.1, if the program determines that all of the waste types are classified as primary wastes, then coemplacement is not done and the program treats each waste type as being emplaced in a separate room. If an emplacement index number greater than 4 is encountered, however, then wastes are coemplaced and the program proceeds to the emplace subroutine. The emplace subroutine determines whether the waste type is a primary, secondary, or tertiary waste, and whether there is space available in rooms to coemplace the

FIGURE 4.1. Flow Diagram for the Complacment Logic

TABLE 4.1. Emplacement Method Index Key

Emplacement Method Index	Waste Emplacement Method		
	Primary	Secondary	Tertiary
1	VBH	None	None
2	RE	None	None
3	T	None	None
4	HBH	None	None
5	VBH	TRU in VBH	None
6	VBH	TRU in VBH	CH in RE
7	VBH	CH in RE	None
8	HBH	TRU in HBH	None
9	HBH	TRU in HBH	CH in RE
10	HBH	CH in RE	None
11	HBH	TRU in T	None
12	HBH	TRU in T	CH in RE
13	RE	TRU in RE	None
14	RE	CH in RE	None
15	RE	TRU in RE	CH in RE

VBH = Vertical Borehole.
HBH = Horizontal Borehole.
RE = Room Emplacement.
T = Trench in Room Floor

TABLE 4.2. Waste Classification Index

Type	Handling	Heat Gen.	Coemplaced with Heat Generating Waste	Class
1. High-level waste	Remote-handled	Yes	No	Primary
2. Spent Fuel	Remote-handled	Yes	No	Primary
3. RH TRU waste	Remote-handled	No	No	Primary
4. RH TRU Waste	Remote-handled	No	Yes	Secondary
5. CH TRU waste	Contact-handled	No	No	Primary
6. CH TRU waste	Contact-handled	No	Yes	Tertiary

secondary and tertiary wastes. If no space is available in that year, then the wastes are stored in temporary storage until space in the rooms becomes available. If wastes are stored in temporary storage beyond a cutoff year which is specified in input, then individual rooms for each stored waste type are excavated and the waste is emplaced in these rooms. Once the program determines whether the waste is primary, secondary or tertiary waste and determines that space is available for emplacement, control is passed to one of three subroutines, depending on the waste class. These three subroutines (PPUT, SPUT, TPUT) keep track of the amount of waste emplaced and the remaining space available in each room. If a room is filled with secondary or tertiary waste in a given year and additional waste remains to be emplaced, then the program checks to see if additional primary waste rooms are available that year. If not, the remaining secondary and tertiary wastes are placed in temporary storage as explained previously. The amount of secondary and tertiary wastes emplaced in the primary waste room is specified in input as the number of secondary or tertiary waste containers per primary waste container. Ultimately, if there is more secondary or tertiary waste than can be emplaced in primary waste rooms, it will be placed in its own room.

THE MINDEV SUBROUTINE

Definitions - The mine development subroutine (MINDEV) calculates the mine excavation requirements prior to and during operations. Excavations prior to operations include mining of the shafts, shaft pillar corridors, and all main corridors. Excavations during operations assume that side drifts and rooms are excavated such that sufficient emplacement area is available to meet waste emplacement requirements after a specified lag period in years (input). MINDEV calculates the rock logistics during waste operations for mining, hoisting, surface storage, offsite disposal, and backfilling. The following definitions are used for these rock operations' components:

- Mining - This operation includes mining the rock face, mucking, haulage of mined rock to storage bins, rock bolting, water control, and any other operations necessary to prepare the corridors and rooms for usage.

- Hoisting - This operation includes loading the rock skips from the storage bin and hoisting the rock to the surface. Lowering rock to finish backfilling the repository is also included in this operation.
- Surface Storage - This operation includes moving the mined rock from the hoisting shaft to the surface storage site, placing the rock at the site, and covering the storage pile (if needed). Recovery of rock and movement to the shaft for backfilling purposes is included in this operation.
- Offsite Disposal - This operation includes loading the railcars from the storage pile, transporting the rock, and disposing of the rock offsite.
- Backfilling - This operation includes transporting the rock from the shaft to the backfilling area, and backfilling the required volume. If backfilling is done during waste operations, only the filled rooms and associated drift corridors are assumed to be backfilled.

MINDEV includes calculation of labor, materials, and equipment requirements for the mining component. Calculations of requirements for other components is done in the ROCKOP subroutine since these operations may extend over a longer time period than waste emplacement operations.

Calculations - The MINDEV subroutine first calculates rock masses mined during waste emplacement operations using the following equations:

$$\text{Rock mass mined (K,I)} = (\text{room volume (K,II)} + \text{corridor volume (K,I)}) \quad (18)$$

$$\quad \quad \quad * \text{rock density}$$

where

K = waste type index

I = year index

II = year waste emplaced

and

$$\text{Room volume (K,II)} = \text{rooms filled with waste (K,II)} * \text{room length} * \text{room width} * \text{room height} \quad (19)$$

$$\text{Corridor volume} = \sum_{M=1}^{\text{NCOR}} (\text{corridor length}^{(M)} * \text{corridor width}^{(M)} * \text{corridor height}^{(M)} * \text{NCORPP(M)} * \text{number of panels filled with waste}) \quad (20)$$

where

NCOR = number of different corridor types in each panel

NCORPP = number of corridors per panel of each type

M = corridor type

Next, the MINDEV subroutine calculates the rock masses mined during shaft and main corridor excavation as follows:

$$\text{Rock excavation for shafts (K,I)} = \sum_{L=1}^{\text{NST}} \left[\text{SFTXR(L)} + \left[\frac{\text{SFTDD(L)}}{2} \right]^2 * \pi * \text{shaft depth(L)} * \text{ALL(K)} * \text{SCSKD(I)} * \text{rock-density} \right] \quad (21)$$

where

SFTXR = incremental shaft radius excavated for creep allowance
or lining to finished diameter

SFTDD = shaft design diameter

ALL = waste type allocation factor = rock mined in waste operation
(K,I)/total rock mined in waste operations(I)

SCSKD = shaft construction schedule

L = type of shaft

K = waste type (NST = number of shaft types)

I = year index

π = pi

$$\text{Rock excavation for corridors (K,I)} = \text{shaft pillar corridor rock mass (K,I)} + \text{main corridor rock mass (K,I)} \quad (22)$$

and

$$\begin{aligned} \text{shaft pillar corridor rock mass (K,I)} = & \text{shaft pillar corridor length} \\ & * \text{width} * \text{height} * \text{rock density} \\ & * \text{ALL(K)} * \text{PORMDF1(I)} \end{aligned} \quad (23)$$

$$\begin{aligned} \text{Main corridor rock mass (K,I)} = & \text{main corridor length} * \text{width} * \text{height} \\ & * \text{rock density} * \text{ALL(K)} * \text{PORMDF2(I)} \end{aligned} \quad (24)$$

where

PORMDF1 and 2 = Pre-operational mining time distribution factors for shaft pillar and main corridor mining, respectively

Mining resource requirements are calculated as follows:

$$\text{Mining machines (I)} = \frac{\text{total rock mined (I)}}{\text{equipment capacity} * \text{number of shifts} * \text{days per year}} \quad (25)$$

where

$$\text{total rock mined (I)} = (\text{Eq. 18} + \text{Eq. 22} + \text{Eq. 23} + \text{Eq. 24})$$

$$\begin{aligned} \text{Mining labor (K,I)} = & \text{mining machines (I)} * \text{labor requirement per machine} \\ & \text{in man-years} * \text{shifts} * \frac{\text{Total Rock Mined (K,I)}}{\text{Total Rock Mined (I)}} \end{aligned} \quad (26)$$

$$\text{Mining materials(K,I)} = \text{material units per tonne} * \text{rock mass mined (K,I)} \quad (27)$$

where

K = waste type index

I = year index

The number of shifts operated is specified in input for both pre-operational mining and for mining during waste operations. Replacement of mining machines is calculated similarly to equipment replacements described in waste operations.

The MINDEV subroutine next calculates backfilling requirements during operations by keeping track of the rooms and corridors mined and the associated volumes. Each year the program checks to see if backfilling operations can start. If backfilling operations are not allowed to commence during waste operations, then all backfilling is done after the specified waiting period in the ROCKOP subroutine. If backfilling operations commence during waste operations a backfilling efficiency is calculated as follows:

$$\text{Backfilling efficiency} = \text{percentage of room filled/rock expansion factor} \quad (28)$$

Using the efficiency, the volume of rock mined in waste operations that year, and the percentage of supplementary backfill material needed (if any), the program calculates the number of rooms and associated corridors that can be backfilled. (Rock stored on the surface is not used for backfill during this period.) The total number of room-corridor units backfilled is summed every year and compared to the total rooms filled with waste. When the two are equal, backfilling of rooms in subsequent years is limited to rooms filled with waste in that year. If the rooms backfilled do not "catch up" with rooms filled with waste during the waste operations period, the number of unbackfilled rooms is sent to ROCKOP, where backfilling after waste operations is calculated.

Finally, the MINDEV subroutine calculates the rock masses associated with hoisting, surface storage, and offsite disposal during waste operations. Hoisting requirements are the difference between mining and backfilling requirements each year. Surface storage requirements are the sum of the hoisting requirements. If the input variable defining surface rock storage area is zero, the program simply calculates the area requirement and stores all rock on the surface. If a limited rock storage area is specified, the program

calculates and sums the total storage area required until it is equal to the limiting area. Excess rock sent to the surface is sent to offsite transportation and disposal.

THE RECEVE SUBROUTINE

The waste receiving subroutine (RECEVE) calculates the number of remote-handled, contact-handled and total waste shipments received at the repository using the following equation:

$$\text{Waste shipments (K,I)} = \frac{\text{waste units received (K,I)}}{\text{rail cask capacity (K) * \% by rail(K,I)} + \text{truck cask capacity(K) * \% by truck(K,I)}} \quad (29)$$

where

K = waste type index

I = year index

The RECEVE subroutine also uses the receiving and packaging module requirements calculated in WASTOP to calculate the total number of modules needed and the total receiving and packaging facility costs, using the relationship:

$$\text{Facility cost} = \text{unit module cost} * \text{maximum number of modules} \quad (30)$$

Distribution of these costs over time and waste type is calculated in HUB.

THE SHAFT SUBROUTINE

The shaft subroutine determines the number of shafts needed to hoist excavated rock from the mine. Rock is assumed to be hoisted in a section of the men and materials shaft. If additional capacity is needed, mine production (rock hoisting) shafts are excavated. The number of rock hoisting shafts is determined as follows:

$$\text{Number of shafts (integer)} = \frac{\text{sum of all rock hoisted (I)} - \text{MMSCAP}}{\text{MPSCAP}} + 0.999 \quad (31)$$

where

MMSCAP = men and materials shaft rock hoisting capacity per year

MPSCAP = mine production shaft rock hoisting capacity per year

I = year index

Rock hoisting capacities for the shafts are calculated from the number of tonnes per trip, the number of trips per shift-year and the number of shifts operated per day (3 maximum).

Shaft costs for sinking, lining, and water control are calculated as functions of shaft diameter, depth and medium using relationships derived from previously published data (Parsons 1978). These relationships are linear functions of shaft diameter and are shown below.

$$\text{Cost in millions of 1978 \$ per meter of depth} = A + B * (\text{shaft diameter} + 2 * \text{INCRAD}) \quad (32)$$

where

	<u>sinking</u>	<u>lining</u>	<u>water control</u>
Salt A =	15,320	-4,264	-2,800,000
B =	352	2,474	1,660,000
Basalt A =	14,000	616	150,000
B =	93	0	19,000
Granite A =	21,800	1,180	1,050,000
B =	90	0	27,000

INCRAD = incremental shaft radius excavated due to creep problems.

These relationships should be used with caution since the data were derived based on generic formation descriptions and may give erroneous results when applied to specific site locations or rock formations.

After the sinking, lining, and water control costs are calculated they are summed together with shaft equipment costs (input) to develop total shaft costs for each shaft. This information is then fed to HUB.

THE ROCKOP SUBROUTINE

The ROCKOP subroutine calculates rock mass logistics for backfilling, hoisting, and surface storage operations that occur between the end of waste operations and the start of decommissioning. In addition this routine calculates the labor, materials, and equipment resource requirements for the above operations over the entire operating period.

ROCKOP begins by determining the backfilling rate to use after waste operations have ended. This rate is specified by the user or as a default; if the rate specified is zero, the program uses the combined rock lowering rates of the men and materials shafts and any mine production shafts. Next, the program determines if backfilling operations have commenced during waste operations. If so, the remaining unbackfilled room volume is divided by the backfilling rate to determine the time period for continuing backfill operations. Using this time period the annual volumes of rock and any supplementary material used for backfill are calculated. If backfilling operations have not commenced during operations, the program calculates the starting year for backfilling operations using the following relationship:

$$\text{Backfilling starting year index} = \text{IBYR} + \text{IYRPRO} - \text{IYRCON} + 1 \quad (33)$$

where

IBYR = the number of years after waste operations startup that backfilling begins

IYRPRO = the calendar year of waste operations startup

IYRCON = the calendar year in which repository construction begins.

The program then uses the total rooms excavated, the room unit volume, and the backfill efficiency to calculate total backfill requirements. These requirements are divided by the backfilling rate previously calculated to determine the time period (from the start of backfilling) required to backfill the rooms and corridors.

The ROCKOP subroutine then calculates backfilling requirements during the decommissioning period (which immediately follows the backfilling period

calculated above). These requirements include backfilling of the main corridors and shaft pillar area. Filling of the shafts is assumed to be included in the shaft sealing costs and requirements. The backfilling data are then used to generate the rock lowering and surface storage logistics using the same logic as explained in MINDEV.

Based on the above rock logistics, the model then calculates equipment, labor, and materials requirements for the backfilling, hoisting, and surface storage operations using the REPLAS subroutine and equations similar to Equations (7) and (8) in WASTOP.

THE UTILITY SUBROUTINE

The UTILITY subroutine generates approximate utility requirements and costs for electricity, steam, water, and diesel fuel used in repository operations. This is accomplished by defining the following:

- average electrical requirements for facilities with significant electrical loads
- water usage requirements for the repository as a whole
- diesel fuel requirements for rail operations, other surface usage and underground usage.
- unit costs for electricity, steam, and diesel fuel.

Electricity usage in surface facilities is calculated as a function of operating time. Equation (34) below shows a typical calculation.

$$\begin{aligned} \text{Receiving facility electricity requirements} &= \text{facility load} * \text{number of modules} \\ &\quad * \text{number of shifts/day} * 8 \text{ hr/shift} \\ &\quad * \text{days per year} \end{aligned} \quad (34)$$

Other electricity requirements such as for mine ventilation and air conditioning assume constant usage throughout the year. Electricity for mining operations is a function of the rock mass mined.

Steam requirements are functions of underground area and process capacity for waste processing. The equation for the latter usage is shown below:

$$\begin{aligned} \text{Waste processing steam requirement} &= \text{requirement per module per year} \\ &\quad * \text{number of modules} \\ &\quad * \text{waste type allocation factor} \end{aligned} \quad (35)$$

Diesel fuel requirements are calculated as a function of working days per year for surface operations, as a function of packages processed for underground transportation, and as a function of shipments received for rail operations. Water requirements are assumed to be constant from year to year.

The above requirements are summed by type of resource each year and multiplied by the unit resource costs (input). Costs are summed for all resources and sent to HUB.

THE SURFST SUBROUTINE

This subroutine calculates construction costs for surface and underground support facilities. Site preparation costs and support facility equipment replacement costs are also calculated in this routine. Equation (36) shows a typical facility cost calculation.

$$\text{Facility construction cost (K,I)} = \sum_{L=1}^{\text{NSC}} \text{facility cost (L)} \quad (36)$$

$$* \text{WAFMV}^{(K)} * \text{SFCS (I)}$$

where

WAFMV = waste type allocation factor based on mined rock volumes

SFCS = surface facility construction schedule (input)

NSC = number of surface facilities described

K = waste type index

I = year index

L = facility index

The support facility cost calculation sums the costs according to the breakdown supplied by the user. This allows the user to use substantial cost detail, if available, or simple aggregate numbers if this is all that is known. Capital costs for facilities to support specific operations underground such as emplacement, transportation and rock operations are calculated separately by equations similar to Equation (36). Support facility equipment cost replacement is assumed to be a function of time rather than facility throughput. Because of this, a separate piece of logic is used in SURFST to calculate these replacements. This logic compares the equipment life (or multiples of this life) specified for each support facility with the current operation year. When a match is found, costs are incurred for new equipment purchases. Support facility labor and materials costs are calculated using input requirements per facility and equations similar to equations 15 and 16.

Site preparation costs are a simple multiple of costs per acre (input) and the surface operations area. All of the above cost calculations are then fed to the HUB subroutine.

THE PREOP SUBROUTINE

The preoperations subroutine (PREOP) calculates costs (other than construction costs) that are incurred prior to the start of waste operations. These include costs for land purchase, licensing, architect-engineer (A-E) services, and facility startup. Land and licensing costs are specified as input variables and are simply multiplied by waste allocation and timing factors in the program to develop costs as functions of time and waste type. Startup and A-E services are calculated as functions of total field costs. Equation (37) below show the calculation of A-E services.

$$\text{A-E cost}(K,I) = \text{AECON1} * \frac{\text{FLDCST}(K,I)}{\text{TOT FLD}(I)} * (\text{FIELD COST})^{\text{AECON2}} \quad (37)$$

where

AECON1 and AECON2 are constants derived from empirical data

$$\text{FIELD COST} = \text{surface facility costs} + \text{underground facility costs} \\ + \text{shaft costs} + \text{site preparation costs}$$

The values of the constants used in the equation were derived from plots of A-E costs as functions of field costs in the Commercial Waste Management GEIS (DOE 1978).

THE POSTOP SUBROUTINE

The post-operations subroutine (POSTOP) calculates costs associated with decommissioning of surface facilities, sealing of shafts, and surveillance of the sealed repository. Facility decommissioning costs are calculated as an input percentage of total surface facility construction costs. Shaft sealing costs are calculated using the following relationship:

$$\text{Shaft sealing costs} = \text{SSCON1} + \text{SSCON2} * (\text{shaft diameter}) \quad (38)$$

where

SSCON1 = shaft sealing constant 1

SSCON2 = shaft sealing constant 2.

The above functional relationship was assumed since it seems intuitive that sealing costs will vary as some function of the shaft diameter. Sufficient data are not yet available, however, to substantiate this function.

Surveillance costs are based on an input constant surveillance cost each year and an input surveillance period. After calculation these costs are all sent to HUB.

THE HUB PROGRAM

The main program (HUB) calls each of the previous subroutines in proper sequence and performs cost calculations and summations from each subroutine's

output. HUB then calculates contingency costs and sums the previous cost calculations to develop total costs and unit costs. Finally, HUB calls the OUTPUT subroutine to print the results of the calculations. The following subsections describe the operations performed in HUB on data from the foregoing subroutines followed by an explanation of the total and unit cost calculations.

INPUT - The only operation performed following this subroutine is the calculation of the present worth factor (PWF) for use in leveling unit costs. This factor is calculated as follows:

$$\text{Present worth factor (I)} = \frac{1}{(1 + \text{real discount rate})^{I+IDL}} \quad (39)$$

where

I = year index

IDL = base year index for discounting (input)

WASTOP - Following WASTOP, the HUB routine calculates total waste operations labor requirements, sums the labor and equipment requirements over waste type, and sums the equipment and labor costs for all waste operations. HUB then calculates total operations costs for each operation as shown below:

$$\begin{aligned} \text{Total operations cost}_N (K,I) = & \text{receiving cost}_N (K,I) \\ & + \text{materials cost}_N (K,I) \\ & + \text{equipment cost}_N (K,I) \end{aligned} \quad (40)$$

where

N = waste operation index

K = waste type index

I = time index

MINDEV - No additional operations are performed after this subroutine.

RECEVE - After the RECEVE subroutine, HUB calculates remote-handling receiving, contact-handled receiving, and packaging facility costs as functions of time and waste type. The program logic identifies when each facility module is

needed and then distributes the construction cost over the construction period immediately prior to module startup. The general equation describing the cost calculation is as follows:

$$\begin{aligned} \text{Facility cost}_N (K,I) = & \text{number of new modules needed}_N (I) \\ & * \text{module cost}_N * \text{construction schedule (I)} \\ & * \text{WAF (K)} \end{aligned} \quad (41)$$

where

N = module type index

K = waste type index

I = year index

WAF = waste type allocation factor based on waste units processed

SHAFT - Using the shaft costs developed in the SHAFT subroutine, the HUB program develops total shaft costs as functions of time and waste type according to the relationship below.

$$\begin{aligned} \text{Total shaft cost (K,I)} = & \sum_{L=1}^{NST} (\text{shaft cost (L)} * \text{number of shafts (L)} \\ & * \text{shaft construction schedule (I)} * \text{WAF (K)}) \end{aligned} \quad (42)$$

where

NST = number of shaft types

K = waste type index

I = year index

WAF = waste type allocation factor (varies with type of shaft).

ROCKOP - The HUB subroutine uses the rock mass logistics and resource requirements developed in MINDEV and ROCKOP to calculate the costs associated with each type of rock operation. Costs may be calculated by one of two options defined by the value of the variable MSWTCH. If MSWTCH equals zero, costs are calculated by multiplying the rock masses associated with each operation by a cost per unit mass (input). If MSWTCH equals one or more, total costs and a

unit cost are calculated in the program using the resource requirements and unit resource costs as shown below.

$$\begin{aligned} \text{Total rock operations cost}_N = & \sum_{I=1}^{LE} \sum_{K=1}^{NWT} [\text{labor}_N(K,I) * \text{labor cost}_N \\ & + \text{materials}_N(K,I) * \text{materials cost}_N \\ & + \text{equipment}_N(I) * \text{equipment cost}_N \\ & * \text{WAFMV}(K)] + \text{capital costs}_N \end{aligned} \quad (43)$$

$$\text{Unit rock operations costs}_N = \frac{\text{total rock operations cost}_N}{\sum_{I=1}^{LE} \sum_{K=1}^{NWT} \text{rock mass}_N(K,I)} \quad (44)$$

where

N = rock operation index

LE = time period of rock operation

NWT = number of waste types

I = year index

K = waste type index

WAFMV = waste type allocation factor using mined volume

The unit rock operations cost is then multiplied by the rock operations mass to develop the rock operations cost as functions of waste type and time. These cost arrays are then added over waste type to develop rock operations costs as a function of time.

UTILITY - No additional operations are performed after this subroutine.

SURFST - After the SURFST subroutine, HUB sums the facility costs into two categories: surface and underground. Surface facility costs include receiving, packaging, support facilities, and support facilities equipment replacement. Underground facilities include transportation support facilities, emplacement support facilities, and emplacement excavation support facilities (if any). Calculations of these summations are shown below:

$$\begin{aligned}
\text{Surface facility costs (K,I)} &= \text{support facility costs (K,II)} \\
&\quad + \text{support facility equipment replacement (K,I)} \\
&\quad + \text{receiving facility cost (K,II)} \\
&\quad + \text{packaging facility cost (K,II)} \qquad (45)
\end{aligned}$$

$$\begin{aligned}
\text{Underground facility costs (K,I)} &= \text{transport facility cost (K,II)} \\
&\quad + \text{emplacement facility cost (K,II)} \\
&\quad + \text{emplacement excavation facility} \\
&\quad \quad \text{cost (K,II)} \qquad (46)
\end{aligned}$$

where

K = waste type

I = waste operation and construction year

II = construction year.

PREOP - No additional operations are performed after this subroutine.

POSTOP - No additional operations are performed after this subroutine.

After completion of the above calculations, HUB sums the equipment costs over waste type and time for the output tables. Following this, the routine calculates contingency costs for preoperations (i.e., construction, licensing, etc.), rock operations, waste operations, and backfilling and decommissioning operations. This is accomplished by multiplying the cost summations in the categories by the relevant contingency factor, as shown below:

$$\begin{aligned}
\text{Contingency cost for preoperations} &= \text{CONFAC(1)} * [\text{landcost} \\
&\quad + \text{licensing cost} \\
&\quad + \text{site preparation cost} \\
&\quad + \text{shaft cost} \\
&\quad + \text{surface facilities cost} \\
&\quad + \text{underground facilities cost} \\
&\quad + \text{A-E costs} + \text{startup costs}] \qquad (47)
\end{aligned}$$

$$\begin{aligned} \text{Contingency cost for rock operations} = & \text{CONFAC}(2) * [\text{mining cost} \\ & + \text{hoisting cost} \\ & + \text{surface storage cost} \\ & + \text{offsite disposal cost}] \end{aligned} \quad (48)$$

$$\begin{aligned} \text{Contingency cost for waste operations} = & \text{CONFAC}(3) * [\text{receiving cost} \\ & + \text{packaging cost} \\ & + \text{transportation cost} \\ & + \text{emplacement cost} \\ & + \text{emplacement excavation cost} \\ & + \text{utilities cost}] \end{aligned} \quad (49)$$

$$\begin{aligned} \text{Contingency cost for decommissioning operations} = & \text{CONFAC}(4) \\ & * [\text{backfilling cost} \\ & + \text{sealing cost} \\ & + \text{decommissioning cost} \\ & + \text{surveillance cost}] \end{aligned} \quad (50)$$

where

CONFAC = the contingency factor for each operation class above.

Note: All of the above variables except CONFAC are dimensioned by waste type and time.

HUB next calculates total costs by summing each cost component over waste type. The components are then summed into the four general categories shown in the contingency calculations above, and the general categories are summed to develop the total costs.

The levelized cost calculations are developed using the cost components as functions of time and waste type, the discount factor and waste masses, volumes, and package units. The general equation used for these calculations is shown below:

$$\text{Levelized unit cost}_N(K) = \frac{\sum_{I=1}^{LE} [\text{cost component } (K,I) * \text{PWF}(I)]}{\sum_{I=1}^{LE} [\text{throughput}_N(K,I) * \text{PWF}(I)]} \quad (51)$$

where

N = type of throughput (i.e., waste mass, waste volume, package)

K = waste type index

I = year index (LE = entire calculation period)

PWF = present worth factor

The levelized cost components are then summed to develop the subtotals and totals, and the final subroutine OUTPUT is called.

THE OUTPUT SUBROUTINE

This subroutine prints the tables showing the repository requirements and costs. The only calculations made in this subroutine are a few table summations and an adjustment of dollar units to correspond to the dollar units shown on the table (i.e., all costs calculated in the body of the model are in dollars. Many tables show results in millions or billions of dollars necessitating adjustment of the values calculated).

A complete listing of input data is printed in front of each case; however, the variables are not named. A complete input data listing, including variable names and descriptions, is available using the preprocessor (see Section 6).

The output tables desired can be selected by the user by specifying the table numbers in input (see input data listing, p. 5.19). If no table numbers are specified, all output tables are printed.

THE OUT22 SUBROUTINE

This subroutine restates the cost summary shown in Table 21 into different cost categories. This was done in order to develop a cost matrix that is more similar to a reference cost matrix developed in 1983 by the Office of NWTIS Integration. Table 4.3 shows the cost format for Table 21. Opposite this

format, is shown the corresponding Table 22 cost matrix locations. Table 4.4 shows the index of these cost matrix locations in the Table 22 format. Tables 4.3 and 4.4 are shown so that the user can determine how the costs shown in Table 22 can be combined to develop the costs shown in Table 21. For example, total surface facilities costs shown in Table 21, under construction and preoperation, can be developed by adding the matrix locations numbers 5, 6, and 20 in Table 4.4. Because Table 22 shows considerable additional cost detail, additional input data are required for the rock operations. This additional detail is requested when the value of the variable MSWITCH is set equal to one (see page 4.22 for explanation). If the value of the variable MSWITCH is not equal to one, Table 22 will not be printed.

OTHER SUBROUTINES AND FUNCTIONS

Another subroutine named WAFAM and another function named HOLMC are used in the model in addition to those previously mentioned. The WAFAM subroutine is a central calculating place for all of the waste type allocation fractions used in the programs. The HOLMC function is used to calculate material costs for emplacement when package or hole barrier dimensions are varied without varying the package cost. The function calculates the incremental volume of material required (+ or -) and ratios the reference package or barrier cost (input) by this volume to develop the new cost.

EMTYPE is a subroutine that determines whether emplacement is in boreholes, trenches, or simple room emplacement by examining the value of the variable IMET (see the Coemplacement Subroutine section).

TABLE 4.3. Table 21 Cost Summary Format

Table 21 Cost Category	Corresponding Table 22 Cost Matrix Locations
Construction and Preoperation	
Land	1
License	2
Site Preparation	3
Shaft	7+12+(13+30) ^(a) +17+18+19+24+29+34+35+36
Surface Facilities	5+6+20
Underground Facilities	8+10+11
Architect Engineer	4
Startup	4
Mining and Rock Removal Operations	
Mining	9+26+42+53+64+46+57+68+14+15+31+32
Hoisting	(13+30) ^(b) +45+56+67
Surface Handle and Store	16+33+47 ^(c) +58 ^(c) +69 ^(c)
Offsite Disposal	47 ^(d) +58 ^(d) +69 ^(d)
Waste Operations	
Receiving	23+40+51+62
Packaging	22+39+50+61
Transportation	25+41+52+63
Emplacement Excavation	27+43+54+65
Emplacement Operations	28+44+55+66
Other	48+59+70+37+72+73+74+75+76+77+78+79
Backfilling and Decommissioning	
Backfilling	81+82+83+84+85+86+87+88+89
Sealing	81+82+83+84+85+86+87+88+89
Decommissioning	91+92+93
Surveillance	91+92+93
Contingency	21+38+49+60+71+80+90+94

(a) Shaft portion of category.

(b) Rock hoisting portion.

(c) Surface handling and storage portion.

(d) Offsite disposal portion.

TABLE 4.4. Table 22 Category Numbers for Determining Corresponding Categories

System	Cost in Millions of Dollars							
	Capital Construction		Operations				Decommissioning	
	Facility Constr.	Capital Equip.	Labor	Material	Equip Replace.	Utility	Backfill & Seal	Decomm.
Site								
Land	1							
Licensing	2							
Site Preparation	3							
Other Prestartup Costs	4							
Waste Package Preparation								
Subtotal	5	22	39	50	61	72		91
Waste Handling and Emplacement								
Receiving	6	23	40	51	62	73		92
Waste Shaft	7						81	
Waste Transfer	8	25	41	52	63	74	82	
Room Excavation	9	26	42	53	64	75		
Waste Emplacement Excavation	10	27	43	54	65	76		
Waste Emplacement	11	28	44	55	66			
Other Rock Excav. & Handling								
Men & Materials Shafts	12	29					83	
Rock Hoisting & R.H. Shafts	13	30	45	56	67		84	
Shaft Pillar Excavation	14	31					85	
All Corridor Excavation	15	32	46	57	68	77	86	
Surface Rock Handl. Stor & Di	16	33	47	58	69			
Ventilation								
Supply Shaft	17	34				78	87	
Waste Empl. Exhaust Shaft	18	35					88	
Mining Exhaust Shaft	19	36					89	
Support	20	37	48	59	70	79		93
Contingency	21	38	49	60	71	80	90	94

5.0 RECON INPUT VARIABLES AND FORMATS

RECON is designed to run on the UNIVAC 1110 or CDC CYBER Series computers. The program reads an input data file created either by an interactive preprocessor (described in Section 6.0) or by reading batch card images. Only two format types are used for reading the data file: four-character alpha-numeric, and free-field (real or integer). Data for input into the free-field real or free-field integer format must exist (i.e., if a value of zero is required, it must be input as zero and not blank). Blanks are used as data field separators and at least one must exist between each data item. If card images are used, a one-to-one correspondence must always be maintained between the data on the cards and the data requirements for input. Table 5.1 lists the input variable names, descriptions and formats. The card images and sequences are also shown to aid the user in setting up a card deck if this input method is chosen. A complete listing of a sample case is shown in Appendix B.

TABLE 5.1. Input Instructions

Card	Variable	Function/Variable Description	Format
GENERAL INFORMATION			
1	DATYPE	Data Category Description	20A4
2	TITLE	Case Description	20A4
3	RPTYPE	Repository Type, Description	4A4
	PLACE	Repository Location, Description	4A4
	TEKNEK	Mining Technique, Description	4A4
4	IYRCON	Calendar Year Construction Begins	Free-Integer
	IYRPRO	Calendar Year Production Begins	Free-Integer
	LASTYR	Calendar Last Year of Waste Receipts	Free-Integer
	IYDOL	Calendar Year for Discounting	Free-Integer
	IDDYR	Number of Years of Decommissioning	Free-Integer
	NWT	Number of Waste Types in this Case	Free-Integer
	DRATE	Discount Rate, Fraction	Free-Real
5	DEPTH	Repository Depth, meters	Free-Real
	DENSTY	Repository Rock Density, g/cm ³	Free-Real
	SPUC	Site Preparation Cost, \$/acre	Free-Real
	CONFAC(1)	Contingency Factor-Site Preparation, fraction	Free-Real
	CONFAC(2)	Contingency Factor-Rock Operations, fraction	Free-Real
	CONFAC(3)	Contingency Factor-Waste Operations, fraction	Free-Real
	CONFAC(4)	Contingency Factor-Backfilling + Decommissioning, fraction	Free-Real
6	SFTNUM(1)	Number of Men and Materials Shafts	Free-Integer
	SFTNUM(2)	Number of Mining Ventilation Shafts	Free-Integer
	SFTNUM(3)	Number of Remote-Handled-Waste Shafts (if = -1 code will calculate)	Free-Integer
	SFTNUM(4)	Number of Contact-Handled Waste Shafts (if = 1 code will calculate)	Free-Integer
	SFTNUM(5)	Number of Mining-Exhaust Shafts	Free-Integer
	SFTNUM(6)	Number of Waste-Exhaust Shafts	Free-Integer
7*	WSTYP	Waste Type Description	3A4
	PKGDS1	Package Description, Output Line One	4A4
	PKGDS2	Package Description, Output Line Two	4A4
	PKGDS3	Package Description, Output Line Three	4A4
	*	Card 7 repeated for each Waste Type (NWT)	
8	KIND(K)	Waste Type Definition (where K = 1, NWT) 1 = Remote-Handled Waste 2 = Contact-Handled Waste 3 = TRU Waste	Free-Integer
	UNTMAS(K)	Waste Unit Mass, kgU (where K = 1, NWT)	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
<u>WASTE OPERATIONS (Page 4.1)</u>			
1	DATYPE	Data Category Description	20A4
2	SIZE	Undeveloped area, acres	Free-Real
	CPA	Central Pillar Area, acres	Free-Real
	GXEF	Geometric Exclusion Factor, Fraction	Free-Real
	NFTL(K)	Near-Field Thermal Limit, kW/acre (where K = 1, NWT)	Free-Real
	FFTL	Far-Field Thermal Limit, kW/acre	Free-Real
	NALSH	Maximum Number of Shifts Allowed to Work	Free-Integer
	NRHRMA	Maximum Number of Remote-Handled-Waste Receiving Modules Allowed	Free-Integer
	NCHRMA	Maximum Number of Contact-Handled-Waste Receiving Modules Allowed	Free-Integer
	WRKYR	Number of Work Days per year	Free-Real
	MTRPSD	Maximum Trips per Shift for Remote-Handled- Waste Shaft	Free-Integer
	MTCPSD	Maximum Trips per Shift for Contact-Handled- Waste Shaft	Free-Integer
	SHCFAC(1)	Shift Differential-Shift 1, (1.0)	Free-Real
	SHCFAC(2)	Shift Differential-Shift 2, 1+ fraction of shift 1	Free-Real
	SHCFAC(3)	Shift Differential-Shift 3, 1+ fraction of shift 1	Free-Real
	OVRHDF	Overhead Factor, 1+ fraction of salaries	Free-Real Free-Real
3	PANLEN(K)	Panel Length, meters (where K = 1, NWT)	Free-Real
	PANWTH(K)	Panel Width, meters (where K = 1, NWT)	Free-Real
4	WUPP(K)	Number of Waste Units per Package (where K = 1, NWT)	Free-Real
	WVOLPU(K)	Waste Volume per unit, m ³ (where K = 1, NWT)	Free-Real
5	TTTR(K)	Average Turnaround Time-Rail Cask Receiving, days (where K = 1, NWT)	Free-Real
	TTTK(K)	Average Turnaround Time-Truck Cask Receiving, days (where K = 1, NWT)	Free-Real
6	RCCAP(K)	Rail Container Capacity, waste units/Rail Car (where K = 1, NWT)	Free-Real
	TCCAP(K)	Truck Container Capacity, waste units/Truck (where K = 1, NWT)	Free-Real
7	PKGCAP(K)	Packaging Module Capacity, Package/Shift (where K = 1, NWT)	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
8	RMLN(K)	Room Length, meters (where K = 1, NWT)	Free-Real
	RMWTH(K)	Room Width, meters (where K = 1, NWT)	Free-Real
9	CLSEMP(K)	Closest Emplacement to Corridor, meters (where K = 1, NWT)	Free-Real
	XRASHO(K)	Extraction Ratio (room/room + pillar), fraction (where K = 1, NWT)	Free-Real
10	HPMIN(K)	Minimum Hole Pitch, meters (where K = 1, NWT)	Free-Real
	PKGPUP(K)	Emplacement Units per Unit Pitch per row (where K = 1, NWT)	Free-Real
11	NBSPPA(K)	Number of Boresections per Panel	Free-Integer
12*	WSTSUP(I)	Waste Quantity Available for Processing, units/year (where I = 1, LASTYR-IYRCON +1)	Free-Real
	*	Card 12 Repeated for Each Waste Type (NWT)	
13*	TDWI(I)	Thermal Density of Remote-Handled-Waste Received, kW/waste unit (where I = 1, LASTYR-IYRCON +1)	Free-Real
	*	Card 13 Repeated for Each Waste Type (NWT)	
14*	CRRAIL(I)	Percent of Waste Units shipped by Rail, fraction (where I = 1, LASTYR - IYRCON + 1)	Free-Real
	*	Card 14 Repeated for Each Waste Type (NWT)	
<u>Remote-Handled-Waste Receiving Module Data</u>			
15	WOCST(1)	Capital Cost, \$/module	Free-Real
	WOLBRR(1)	Labor Requirement, man-years/shift/module	Free-Real
	WOLBRC(1)	Labor Cost, \$/man-year	Free-Real
	WOEQPL(1)	Equipment Life, waste units	Free-Real
	WOEQPC(1)	Equipment Capacity, waste units/equipment unit/shift	Free-Real
	WOECST(1)	Equipment Cost, \$/equipment unit	Free-Real
16	WOMATU(1,K)	Materials Requirement, material unit/waste unit (where K = 1, NWT)	Free-Real
	WOMATC(1,K)	Materials Cost, \$/material unit (where K = 1, NWT)	Free-Real
<u>Contact-Handled-Waste Receiving Module Data</u>			
17	WOCST(2)	Capital Cost, \$/module	Free-Real
	WOLBRR(2)	Labor Requirement, man-years/shift/module	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
<u>Contact-Handled-Waste Receiving Module Data (contd)</u>			
17	(contd)		
	WOLBRC(2)	Labor Cost, \$/man-year	Free-Real
	WOEQPL(2)	Equipment Life, receiving units	Free-Real
	WOEQPC(2)	Equipment Capacity, receiving units/equipment unit/shift	Free-Real
	WOECST(2)	Equipment Cost, \$/equipment unit	Free-Real
18	WOMATU(2,K)	Materials Requirement, material unit/receiving unit (where K = 1, NWT)	Free-Real
	WOMATC(2,K)	Materials Cost, \$/material unit (where K = 1, NWT)	Free-Real
<u>Waste Packaging Module Data</u>			
19	WOCST(3)	Capital Cost, \$/module	Free-Real
	WOLBRRR(3)	Labor Requirement, man-years/shift/module	Free-Real
	WOLBRC(3)	Labor Cost, \$/man-year	Free-Real
	WOEQPL(3)	Equipment Life, packaging units	Free-Real
	WOEQPC(3)	Equipment Capacity, packaging units/equipment unit/shift	Free-Real
	WOECST(3)	Equipment Cost, \$/equipment unit	
20	WOMATU(3,K)	Materials Requirement, material unit/package (where K = 1, NWT)	Free-Real
	WOMATC(3,K)	Materials Cost, \$/Material Unit (where K = 1, NWT)	Free-Real
<u>MINE DEVELOPMENT (Page 4.9)</u>			
1	DATYPE	Data Category Description	20A4
2	LEAD	Time Lag Between Room Excavation and Waste Emplacement, years	Free-Integer
	NCOR	Number of Access and Ventilation Corridor Types	Free-Integer
	TCPCLN	Total Central Pillar Corridor Length, meters	Free-Real
	PCTFIL	Portion of Each Room Backfilled, fraction	Free-Real
	XPANFC	Expansion of Rock Volume After Mining, 1+ fraction	Free-Real
	MSA	Mined Rock Storage Area, acres (if MSA = 0, code will calculate)	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
		MINE DEVELOPMENT (Page 4.9)	(contd)
2	(contd)		
	TONPAR	Rock Stored per Unit Surface Area, tons/acre	Free-Real
	OFTADC	Offsite Transportation and Disposal Cost, \$/tonne	Free-Real
	NBYR	Years after construction start up that backfilling begins, years	Free-Integer
	OTHBFL	Portion of Backfill Volume Not of Original Mined Rock, fraction	Free-Real
	OTBDEN	Density of Other Backfill Material, g/cm ³	Free-Real
	BKRATE	Backfill Rate, tonnes/year (if = "0", uses shaft capacity)	Free-Real
	MSHIFT(1)	Number of shift/day worked (250 day year) for mining during repository construction	Free-Integer
	MSHIFT(2)	Number of shift/day worked (250 day year) for mining during repository operation	Free-Integer
3	RMHMIN(K)	Minimum Room Height, meters (where K = 1, NWT)	Free-Real
4	CORDSC(1)	Corridor Description "MAIN CORRIDOR"	12A4
5	CORWTH(1)	Main Corridor Width, meters	Free-Real
	CORHT(1)	Main Corridor Height, meters	Free-Real
	DUMMY	"0"	Free-Integer
6	CORLEN(1)	Main Corridor Length, meters	Free-Real
7	CORDSC	Miscellaneous Corridor Description (access, ventilation, etc.)	12A4
8	CORWTH	Miscellaneous Corridor Width, meters	Free-Real
	CORHT	Miscellaneous Corridor Height, meters	Free-Real
	NCORPP	Number of Corridors of This Type per Panel	Free-Integer
9	CORLEN(K)	Miscellaneous Corridor Length, meters (where K = 1, NWT)	Free-Real
	*	Repeat cards 7, 8, and 9 for each Corridor type (NCOR)	
10	DEL RHT(K)	Delta Room Height, meters (where K = 1, NWT)	Free-Real
	PKGLN(K)	Package Length, meters (where K = 1, NWT)	Free-Real
	NRWSPR(K)	Number of Rows per Room (where K = 1, NWT)	Free-Integer

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
11	PORMDF(1,I)	Central Pillar Area Mining Cost Distribution Schedule, fraction/year (where I = 1, IYRPRO - IYRCON)	Free-Real
	PORMDF(2,I)	Main Corridor Area Mining Cost Distribution Schedule, fraction/year (where I = 1, IYRPRO-IYRCON)	Free-Real
	SCSKD(I)	Shafts Excavation Cost Distribution Schedule, fraction/year (where I = 1, IYRPRO-IYRCON)	Free-Real
12	SFTDD(1)	Men and Materials Shaft Diameter, meters	Free-Real
	SFTXR(1)	Men and Materials Shaft Incremental Excavated Radius, meters	Free-Real
	SFTXD(1)	Men and Materials Shaft Excavated Depth, meters	Free-Real
	SFTDD(2)	Mining Ventilation Shaft Diameter, meters	Free-Real
	SFTXR(2)	Mining Ventilation Shaft Incremental Excavated Radius, meters	Free-Real
	SFTXD(2)	Mining Ventilation Shaft Excavated Depth, meters	Free-Real
	SFTDD(3)	Remote-Handled-Waste Shaft Diameter, meters	Free-Real
	SFTXR(3)	Remote-Handled-Waste Shaft Incremental Excavated Radius, meters	Free-Real
	SFTXD(3)	Remote-Handled-Waste Shaft-Excavated Depth, meters	Free-Real
	SFTDD(4)	Contact-Handled-Waste Shaft Diameter, meters	Free-Real
	SFTXR(4)	Contact-Handled-Waste Shaft Incremental Excavated Radius, meters	Free-Real
	SFTXD(4)	Contact-Handled-Waste Shaft Excavated Depth, meters	Free-Real
	SFTDD(5)	Mining Exhaust Shaft Diameter, meters	Free-Real
	SFTXR(5)	Mining Exhaust Shaft Incremental Excavated Radius, meters	Free-Real
	SFTXD(5)	Mining Exhaust Shaft Excavated Depth, meters	Free-Real
	SFTDD(6)	Waste Exhaust Shaft Diameter, meters	Free-Real
	SFTXR(6)	Waste Exhaust Shaft Incremental Excavated Radius, meters	Free-Real
	SFTXD(6)	Waste Exhaust Shaft Excavated Depth, meters	Free-Real
	SFTDD(7)	Rock Hoisting Shaft Diameters, meters	Free-Real
	SFTXR(7)	Rock Hoisting Shaft Incremental Excavated Radius, meters	Free-Real
	SFTXD(7)	Rock Hoisting Shaft Excavated Depth, meters	Free-Real
13	MSWTCH	Rock Operations Cost Input Data Selection Switch "0" = Input is Unit Price Data "1" = Input is Detailed Cost Data	Free-Integer

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
Where MSWTCH = 0			
14	VARMC	Variable Mining Cost, \$/tonne	Free-Real
	RCKHC	Rock Hoisting Cost, \$/tonne	Free-Real
	SURFDC	Surface Disposal Cost, \$/tonne	Free-Real
	BKFLC	Backfilling Cost, \$/tonne	Free-Real
Where MSWTCH = 1			
14	ROCST(1)	Excavation and Haulage Capital Cost, \$	Free-Real
	ROLBRR(1)	Excavation and Haulage Labor Requirement, man-years/equipment unit/shift	Free-Real
	ROLBRC(1)	Excavation and Haulage Labor Cost, \$/man-year	Free-Real
	ROEQPL(1)	Excavation and Haulage Equipment Life, tonnes/unit	Free-Real
	ROEQPC(1)	Excavation and Haulage Equipment Capacity tonnes/unit/shift-year	Free-Real
	ROECST(1)	Excavation and Haulage Equipment Cost, \$/unit	Free-Real
	ROMATU(1)	Excavation and Haulage Material Requirement material unit/tonne	Free-Real
	ROMATC(1)	Excavation and Haulage Material Cost, \$/material unit	Free-Real
Where MSWTCH = 1			
15	ROCST(2)	Hoisting Capital Cost, \$	Free-Real
	ROLBRR(2)	Hoisting Labor Requirement, man-years/unit/ shift	Free-Real
	ROLBRC(2)	Hoisting Labor Cost, \$/man-year	Free-Real
	ROEQPL(2)	Hoisting Equipment Life, tonnes/unit	Free-Real
	ROEQPC(2)	Hoisting Equipment Capacity, tonnes/unit/ shift	Free-Real
	ROECST(2)	Hoisting Equipment Costs, \$/unit	Free-Real
	ROMATU(2)	Hoisting Material Requirement, tonnes/unit/ shift	Free-Real
	ROMATC(2)	Hoisting Material Cost, \$/material unit	Free-Real
Where MSWTCH = 1			
16	ROCST(3)	Rock Stockpile Capital Cost, \$	Free-Real
	ROLBRR(3)	Rock Stockpile labor Requirement, man-years/ unit/shift	Free-Real
	ROLBRC(3)	Rock Stockpile Labor Cost, man-years/unit/ shift	Free-Real
	ROEQPL(3)	Rock Stockpile Equipment Life, tonnes/unit	Free-Real
	ROEQPC(3)	Rock Stockpile Equipment Capacity, tonnes/ unit/shift	Free-Real
	ROECST(3)	Rock Stockpile Equipment Costs, \$/unit	Free-Real
	ROMATU(3)	Rock Stockpile Material Requirement, material unit/tonne	Free-Real
	ROMATC(3)	Rock Stockpile Material Cost, \$/material unit	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
Where MSWTCH = 1			
17	ROCST(4)	Offsite Disposal Capital Cost, \$	Free-Real
	ROLBRR(4)	Offsite Disposal Labor Requirement, man-years/unit/shift	Free-Real
	ROLBRC(4)	Offsite Disposal Labor Cost, \$/man-year	Free-Real
	ROEQPL(4)	Offsite Disposal Equipment Life, tonnes/unit	Free-Real
	ROEQPC(4)	Offsite Disposal Equipment Capacity, tonnes/unit/shift/year	Free-Real
	ROECST(4)	Offsite Disposal Equipment Costs, \$/unit	Free-Real
	ROMATU(4)	Offsite Disposal Material Requirement, material unit/tonne	Free-Real
	ROMATC(4)	Offsite Disposal Material Cost, material unit/tonne	Free-Real
Where MSWTCH = 1			
18	ROCST(5)	Backfilling and Haulage Capital Cost, \$	Free-Real
	ROLBRR(5)	Backfilling and Haulage Labor Requirement, man-year/unit/shift	Free-Real
	ROLBRC(5)	Backfilling and Haulage Labor Cost, \$/man-year	Free-Real
	ROEQPL(5)	Backfilling and Haulage Equipment Life, tonnes/unit	Free-Real
	ROEQPC(5)	Backfilling and Haulage Equipment Capacity, tonnes/unit/shift	Free-Real
	ROECST(5)	Backfilling and Haulage Equipment Cost, \$/unit	Free-Real
	ROMATU(5)	Backfilling and Haulage Material Requirement, material unit/tonne	Free-Real
	ROMATC(5)	Backfilling and Haulage Material Cost, \$/material unit	Free-Real
RECEIVING OPERATIONS (Page 4.14)			
1	DATYPE	Data Category Description	20A4
2	NRBPRM	Number of Receiving Bays per Module R-H	Free-Integer
	NCBPRM	Number of Receiving Bays per Module C-H	Free-Integer
3	RMCWS1(L)	Remote-Handled-Waste Receiving Module Construction Schedule Prior to Operations, fraction/year (where L = 1, IYRPRO - IYRCON)	Free-Real
	RMCWS2(L)	Remote-Handled-Waste Receiving Module Construction Schedule During Operations, fraction/year (where L = 1, IYRPRO - IYRCON)	Free-Real
4	RMNWS1(L)	Contact-Handled-Waste Receiving Module Construction Schedule Prior to Operations, fraction/year (where L = 1, IYRPRO - IYRCON)	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
		RECEIVING OPERATIONS (Page 4.14)	(contd)
4	(contd)		
	RMNWS2(L)	Contact-Handled-Waste Receiving Module Construction Schedule During Operations, fraction/year (where L = 1, IYRPRO - IYRCON)	Free-Real
		UNDERGROUND OPERATIONS	
1	DATYPE	Data Category Description	20A4
2	TOCST(1)	Remote-Handled-Waste Transportation, Capital Cost, \$	Free-Real
	TOLBRR(1)	Remote-Handled-Waste Transportation Unit Labor Requirement, man-years/equipment unit	Free-Real
	TOLBRC(1)	Remote-Handled-Waste Transportation Unit Labor Cost, \$/man-year	Free-Real
	TOEQPL(1)	Remote-Handled-Waste Transportation Unit Equipment Life, packages	Free-Real
	TOEQPC(1)	Remote-Handled-Waste Transportation Unit Equipment Capacity, packages/equipment-unit/shift	Free-Real
	TOECST(1)	Remote-Handled-Waste Transportation Unit Equipment Cost, \$/equipment unit	Free-Real
3	TOMATU(1,K)	Remote-Handled-Waste Transportation Unit Material Requirement, material units/package (where K = 1, NWT)	Free-Real
	TOMATC(1,K)	Remote-Handled-Waste Transportation Unit Materials Cost, \$ (where K = 1, NWT)	Free-Real
4	TOCST(2)	Contact-Handled-Waste Transportation Capital Cost, \$	Free-Real
	TOLBRR(2)	Contact-Handled-Waste Transportation Unit Labor Requirement, man-years/equipment unit	Free-Real
	TOLBRC(2)	Contact-Handled-Waste Transportation Unit Labor Cost, \$/man-year	Free-Real
	TOEQPL(2)	Contact-Handled-Waste Transportation Unit Equipment Life, packages	Free-Real
	TOEQPC(2)	Contact-Handled-Waste Transportation Unit Equipment Capacity, packages/equipment unit/shift	Free-Real
	TOECST(2)	Contact-Handled-Waste Transportation Unit Equipment Cost, \$/equipment unit	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format																																																																
5	TOMATU(2,K)	Contact-Handled-Waste Transportation Unit Materials Cost, \$/material unit (where K = 1, NWT)	Free-Real																																																																
	TOMATC(2,K)	Contact-Handled-Waste Transportation Unit Materials Cost, \$/material unit (where K = 1, NWT)	Free-Real																																																																
6	IDET	Materials Cost Input Data Selection Switch "0" = Input is Costs "1" = Input is Reference Cost Factors	Free-Integer																																																																
	IUHDCM	Hole Drilling Cost Calculation Method Switch "0" = Input is Cost "1" = Input is Unit Cost Functional Relations	Free-Integer																																																																
7	TRCHWD(K)	Trench Width, meters ("0" if IMET ≠ 3)	Free-Real																																																																
8	NPKPEU	Number of Packages per Emplacement Relations	Free-Integer																																																																
	PKG DIA	Package Diameter, meters	Free-Real																																																																
	REF DIA	Reference Diameter Used to Describe Package Costs, meters	Free-Real																																																																
	HOLDEL	Hole Depth Delta, meters	Free-Integer																																																																
	IMET	Emplacement Method																																																																	
		<table> <tr> <th></th><th>PRIMARY</th><th>SECONDARY</th><th>TERTIARY</th></tr> <tr><td>1 =</td><td>Vertical Borehole</td><td>None</td><td>None</td></tr> <tr><td>2 =</td><td>Room Emplacement</td><td>None</td><td>None</td></tr> <tr><td>3 =</td><td>Trenching</td><td>None</td><td>None</td></tr> <tr><td>4 =</td><td>Horizontal Borehole</td><td>None</td><td>None</td></tr> <tr><td>5 =</td><td>Vertical Borehole</td><td>TRU in VBH</td><td>None</td></tr> <tr><td>6 =</td><td>Vertical Borehole</td><td>TRU in VBH</td><td>CH in RE</td></tr> <tr><td>7 =</td><td>Vertical Borehole</td><td>CH in RE</td><td>None</td></tr> <tr><td>8 =</td><td>Horizontal Borehole</td><td>TRU in HBH</td><td>None</td></tr> <tr><td>9 =</td><td>Horizontal Borehole</td><td>TRU in HBH</td><td>CH in RE</td></tr> <tr><td>10 =</td><td>Horizontal Borehole</td><td>CH in RE</td><td>None</td></tr> <tr><td>11 =</td><td>Horizontal Borehole</td><td>TRU in T</td><td>None</td></tr> <tr><td>12 =</td><td>Horizontal Borehole</td><td>TRU in T</td><td>CH in RE</td></tr> <tr><td>13 =</td><td>Room Emplacement</td><td>TRU in RE</td><td>None</td></tr> <tr><td>14 =</td><td>Room Emplacement</td><td>CH in RE</td><td>None</td></tr> <tr><td>15 =</td><td>Room Emplacement</td><td>TRU in RE</td><td>CH in RE</td></tr> </table>		PRIMARY	SECONDARY	TERTIARY	1 =	Vertical Borehole	None	None	2 =	Room Emplacement	None	None	3 =	Trenching	None	None	4 =	Horizontal Borehole	None	None	5 =	Vertical Borehole	TRU in VBH	None	6 =	Vertical Borehole	TRU in VBH	CH in RE	7 =	Vertical Borehole	CH in RE	None	8 =	Horizontal Borehole	TRU in HBH	None	9 =	Horizontal Borehole	TRU in HBH	CH in RE	10 =	Horizontal Borehole	CH in RE	None	11 =	Horizontal Borehole	TRU in T	None	12 =	Horizontal Borehole	TRU in T	CH in RE	13 =	Room Emplacement	TRU in RE	None	14 =	Room Emplacement	CH in RE	None	15 =	Room Emplacement	TRU in RE	CH in RE	
	PRIMARY	SECONDARY	TERTIARY																																																																
1 =	Vertical Borehole	None	None																																																																
2 =	Room Emplacement	None	None																																																																
3 =	Trenching	None	None																																																																
4 =	Horizontal Borehole	None	None																																																																
5 =	Vertical Borehole	TRU in VBH	None																																																																
6 =	Vertical Borehole	TRU in VBH	CH in RE																																																																
7 =	Vertical Borehole	CH in RE	None																																																																
8 =	Horizontal Borehole	TRU in HBH	None																																																																
9 =	Horizontal Borehole	TRU in HBH	CH in RE																																																																
10 =	Horizontal Borehole	CH in RE	None																																																																
11 =	Horizontal Borehole	TRU in T	None																																																																
12 =	Horizontal Borehole	TRU in T	CH in RE																																																																
13 =	Room Emplacement	TRU in RE	None																																																																
14 =	Room Emplacement	CH in RE	None																																																																
15 =	Room Emplacement	TRU in RE	CH in RE																																																																
	* Repeat card 8 for each waste type (NWT)																																																																		
9	MPRIME	Number of primary heat generating waste types	Free-Integer																																																																
	NNHG	Number of non-heat generating waste types	Free-Integer																																																																
	ICLR	Year inventories of stored non-heat generating waste can be cleared by solo emplacement	Free-Integer																																																																

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
	Skip cards 10 through 13 if MPRIME is equal to NWT		
10	STOMAX(K)	Storage capacity limit for secondary waste (packages) (where K = 1, NWT)	Free-Real
11*	COCON	Number of non-heat generating waste packages that can be coemplaced per primary waste package	Free-Real
	PCTCON	Fraction secondary waste emplacement capacity available for this secondary waste	Free-Real
	*	Repeat card 11 for each secondary waste by primary waste	
12	SWPPR	Secondary waste packages per room solo emplacement	Free-Real
13	CONTRU	Number of CH packages that can be emplaced per TRU waste package for each primary heat-generating waste type	Free-Real
	*	(Skip card 13 if no contact-handled waste available)	
14	UOCST(1)	Remote-Handled-Waste Emplacement Capital Cost, \$	Free-Real
	UOLBRR(1)	Remote-Handled-Waste Emplacement Labor Requirement, man-years/equipment unit	Free-Real
	UOLBRC(1)	Remote-Handled-Waste Emplacement Labor Cost, \$/man-year	Free-Real
	UOEQPL(1)	Remote-Handled-Waste Emplacement Unit Equipment Life, packages	Free-Real
	UOEQPC(1)	Remote-Handled-Waste Emplacement Unit Equipment Capacity, packages/equipment unit/shift	Free-Real
	UOECST(1)	Remote-Handled-Waste Emplacement Unit Equipment Cost, \$/equipment unit	Free-Real
Where IDET = "0"			
15*	UOMATU(1)	Remote-Handled-Waste Barrier Material Unit Requirement, material unit/emplacement unit	Free-Real
	UOMATC(1)	Remote-Handled-Waste Barrier Unit Material Cost, \$/Emplacement unit	Free-Real
	*	Card 15 Repeated for Each Remote-Handled-Waste Type	

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
16	UOCST(2)	Contact-Handled-Waste Emplacement Capital Cost \$	Free-Real
	UOLBRR(2)	Contact-Handled-Waste Emplacement Labor Requirement, man-years/equipment unit	Free-Real
	UOLBRC(2)	Contact-Handled-Waste Emplacement Labor Cost, \$/man-year	Free-Real
	UOEQPL(2)	Contact-Handled-Waste Emplacement Unit Equipment Life, packages	Free-Real
	UOEQPC(2)	Contact-Handled-Waste Emplacement Unit Equipment Capacity, packages/equipment unit/shift	Free-Real
	UOECST(2)	Contact-Handled-Waste Emplacement Unit Equipment Cost, \$/equipment unit	Free-Real
Where IDET = "0"			
17*	UOMATU(2)	Contact-Handled-Waste Emplacement Material Unit Requirement, material unit/emplacement unit	Free-Real
	UOMATC(2)	Contact-Handled-Waste Emplacement Unit Material Cost, \$/material unit	Free-Real
	*	Card 17 Repeated for Each Contact-Handled-Waste Type	
18*	WABATH	Water Barrier Material Thickness, meters	Free-Real
	SLEETH	Sleeve Thickness, meters	Free-Real
	BCKFTH	Backfill Material Thickness, meters	Free-Real
	CAPLEN	Shield Plug Length, meters	Free-Real
	*	Repeat card 18 for each waste type	
Where IDET = "1"			
19*	REFMAT	Reference Hole Liner Material Cost, \$/package	Free-Real
	REFSLV	Reference Sleeve Cost, \$/sleeve	Free-Real
	REFBKF	Reference Backfill Material Cost, \$/hole	Free-Real
	REFDEP	Reference Depth for Reference Costs, meters	Free-Real
20*	CHMTEM	Contact-Handled-Waste Materials Cost, \$/Unit Pitch	Free-Real
	PKGPRK	Packages Emplaced per Rack	Free-Real
	RCKCST	Rack Cost, \$/rack	Free-Real
	*	Card 19 and 20 are Repeated for Each Waste Type (NWT)	
Where IUHDCM = "0"			
21	EECST(1)	Hole Drilling Capital Cost, \$	Free-Real
	EELBRR(1)	Hole Drilling, Labor Requirement, man-years/equipment unit/shift/day	Free-Real
	EELBRC(1)	Hole Drilling Labor Cost, \$/man-year	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
21	(contd)		
	EEEQPL(1)	Hole Drilling, Emplacement Units/equipment unit	Free-Real
	EEEQPC(1)	Hole Drilling Unit Equipment Capacity, emplacement units/equipment unit/shift/day	Free-Real
	EEECST(1)	Hole Drilling Unit Equipment Cost, \$/equipment unit	Free-Real
	EEMATU(1)	Hole Drilling Material Requirement material units/emplacement unit	Free-Real
	EEMATC(1)	Hole Drilling Material Cost, \$/material unit	Free-Real
Where IUHDCM = "1"			
22	EELBRC(1)	Hole Drilling Unit Labor Cost, \$/man-year	Free-Real
	EEECST(1)	Hole Drilling Unit Equipment Cost, \$/equipment unit	Free-Real
	EEMATC(1)	Hole Drilling Materials Cost, \$/material unit	Free-Real
	HDLFCF	Portion of Hole Drilling Unit Cost Attributed to Labor, fraction	Free-Real
	HDECFC	Portion of Hole Drilling Unit Cost Attributed to Equipment, fraction	Free-Real
	HDMCFC	Portion of Hole Drilling Unit Cost Attributed to materials, fraction	Free-Real
	HDCON1	Hole Drilling Cost Function Constant A (see Page 4.8)	Free-Real
	HDCON2	Hole Drilling Cost Function Constant B (see Page 4.8)	
	NTYP	Hole Drilling Cost Functional Form (0 = Power, 1 = Exponential)	Free-Integer
23	EECST(2)	Room Emplacement Capital Cost, \$	Free-Real
	EELBRR(2)	Room Emplacement Labor Requirement, man-years/emplacement unit/shift	Free-Real
	EELBRC(2)	Room Emplacement Labor Cost, \$/man-year	Free-Real
	EEEQPL(2)	Room Emplacement Unit Equipment Life, packages/equipment unit	Free-Real
	EEEQPC(2)	Room Emplacement Unit Equipment Capacity packages/equipment unit/shift	Free-Real
	EEECST(2)	Room Emplacement Unit Equipment Cost, \$/equipment unit	Free-Real
	EEMATU(2)	Room Emplacement Material Requirement, material units/package	
	EEMATC(2)	Room Emplacement material Cost, \$/material unit	Free-Real
24	EECST(3)	Trenching Capital Cost, \$	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
24	(contd)		
	EELBRR(3)	Trenching Labor Requirement, man-years/ emplacement unit/shift	Free-Real
	EELBRC(3)	Trenching Labor Cost, \$/man-year	Free-Real
	EEEQPL(3)	Trenching Unit Equipment Life, packages/ equipment unit	Free-Real
	EEEQPC(3)	Trenching Unit Equipment Capacity, packages/ equipment unit/shift	Free-Real
	EEECST(3)	Trenching Unit Equipment Cost, \$/equipment unit	Free-Real
	EEMATU(3)	Trenching Material Requirement, material units/package	Free-Real
	EEMATC(3)	Trenching Material Cost, \$/material unit	Free-Real
SHAFT OPERATIONS (Page 4.14)			
1	DATYPE	Data Category Description	20A4
2	MXMPRH	Maximum Trips per Shift-Mine Production Shaft	Free-Integer
	MXMMRH	Maximum Trips per Shift-Men and Materials Shaft	Free-Integer
	TNPT1	Rock Hoisting Capacity-Mine Production Shaft, tonnes/trip	Free-Real
	TNPT2	Rock Hoisting Capacity-men and Materials Shaft, tonnes/trip	Free-Real
	SCCON1	Shaft Sinking Cost Function Constant A (see page 4.15)	Free-Real
	SCCON2	Shaft Sinking Cost Function Constant B (see Page 4.15)	Free-Real
	IEQTP1	Sinking Cost Functional Form (1 = linear, 2 = exponential)	Free-Integer
	CCCON1	Shaft Lining Cost Function Constant A (see Page 4.15)	Free-Real
	CCCON2	Shaft Lining Cost Function Constant B (see Page 4.15)	Free-Real
	IEQTP2	Lining Cost Functional Form (1 = linear, 2 = exponential)	Free-Real
	WCCON1	Water Control Cost Function Constant (see Page 4.15)	Free-Real
	WCCON2	Water Control Cost Function Constant (see Page 4.15)	Free-Real
	IEQTP3	Water Control Cost Functional Form (1 = linear, 2 = exponential)	Free-Integer
3	SFTLD(1)	Men and Materials Shaft Lined Depth, meters	Free-Real
	SFTLD(2)	Mining Ventilation Shaft Lined Depth, meters	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
3	(contd)		
	SFTLD(3)	Remote-Handled-Waste Shaft Lined Depth, meters	Free-Real
	SFTLD(4)	Contact-Handled-Waste Shaft Lined Depth, meters	Free-Real
	SFTLD(5)	Mining Exhaust Shaft Lined Depth, meters	Free-Real
	SFTLD(6)	Waste Exhaust Shaft Lined Depth, meters	Free-Real
	SFTLD(7)	Rock-Hoisting Shaft Lined Depth, meters	Free-Real
4	SFTHE(1)	Men and Materials Shaft Hoisting Equipment Cost, \$	Free-Real
	SFTHE(2)	Mining Ventilation Shaft Hoisting Equipment Cost, \$	Free-Real
	SFTHE(3)	Remote-Handled-Waste Shaft Hoisting Equipment Cost, \$	Free-Real
	SFTHE(4)	Contact-Handled-Waste Shaft Hoisting Equipment Cost, \$	Free-Real
	SFTHE(5)	Mining Exhaust Shaft Hoisting Equipment Cost, \$	Free-Real
	SFTHE(6)	Waste Exhaust Shaft Hoisting Equipment Cost, \$	Free-Real
	SFTHE(7)	Rock Hoisting Shaft Hoisting Equipment Cost, \$	Free-Real
5	SFTSS(1)	Men and Materials Shaft Station Cost, \$	Free-Real
	SFTSS(2)	Mining Ventilation Shaft Station Cost, \$	Free-Real
	SFTSS(3)	Remote-Handled Waste Shaft Station Cost, \$	Free-Real
	SFTSS(4)	Contact-Handled Waste Shaft Station Cost, \$	Free-Real
	SFTSS(5)	Mining Exhaust Shaft Station Cost, \$	Free-Real
	SFTSS(6)	Waste Exhaust Shaft Station Cost, \$	Free-Real
	SFTSS(7)	Rock Hoisting Shaft Station Cost, \$	Free-Real
6	NPKPTR(K)	Remote-Handled-Waste Shaft Transport Rate, packages/trip (where K = 1, NWT)	Free-Integer
	NPKPTC(K)	Contact-Handled-Waste Shaft Transport Rate, packages/trip (where K = 1, NWT)	Free-Integer

SURFACE FACILITY REQUIREMENTS (Page 4.18)

1	DATYPE	Data Category Description	20A4
2	SFCS(L)	Other Surface Facility Construction Schedule, fraction/year (where L = 1, IYRPRO - IYRCON)	Free-Real
3	NSC	Number of Support Facility Descriptions	Free-Integer
	SOA	Surface Site Operations Area, Acres	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
4*	SFTITL	Support Facility Name	12A4
5*	SFCST	Support Facility Capital Cost, \$	Free-Real
	SFLBRR	Support Facility Labor Requirement, man-years/facility/shift	Free-Real
	SLBR	Number of Shifts per Facility (if = -1, code uses Remote-Handled-Waste Operations Shift Requirements) (if = -2, code uses Contact-Handled-Waste Operations Shift Requirement)	Free-Real
	SFLBRC	Support Facility Labor Cost, \$/man-year	Free-Real
	SFEQPL	Support Facility Equipment Life, years	Free-Real
	SFECST	Support Facility Equipment Cost, \$/facility	Free-Real
	SFMATU	Support Facility Material Requirements, Annual material units/facility	Free-Real
	SFMATC	Support Facility Material Cost, \$/material unit	Free-Real
	*	Card Set 4 and 5 Repeated for Each Facility Description (NSC)	
UTILITY DATA (Page 4.17)			
1	DATYPE	Data Category Description	20A4
2	UPC	Electricity Unit Cost, mills/kWh	Free-Real
	STMC	Steam Unit Cost, \$/10 ³ lb	Free-Real
	DWTRC	Demineralized Water Unit Cost, \$/10 ⁶ liters	Free-Real
	WTRC	Potable Water Unit Cost, \$/10 ⁶ liters	Free-Real
	DSLC	Diesel Fuel Unit Cost, \$/liter	Free-Real
3	RAPP	Electrical Load-Receiving, and Packaging, kW/module	Free-Real
	SFSP	Electrical Load-Lag Storage Pool, kW	Free-Real
	OWHP	Electrical Load-Waste Handling Building, kW	Free-Real
	SFP	Electrical Load-Support Facilities, kW	Free-Real
	MROP	Electrical Load-Rock Operations, kW/tonne/year	Free-Real
	RVP	Electrical Load-Mine Ventilation, kW/acre	Free-Real
	ACP	Electrical Load-Air Conditioning, kW	Free-Real
	HEATS	Steam Requirement-Underground Heating, 10 ³ lb/acre	Free-Real
	RAPS	Steam Requirement-Receiving Facility Heating, 10 ³ lb/module	Free-Real
	ORS	Steam Requirement-Other Facilities, 10 ³ lb	Free-Real
4	VOLW	Potable Water Requirement, 10 ⁶ liters	Free-Real
	VOLDW	Demineralized Water Requirement, 10 ⁶ liters	Free-Real

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
4	(contd)		
	SUD	Diesel Fuel Requirement-Surface, liters/day	Free-Real
	UGD	Diesel Fuel Requirement-Underground, liters/ waste unit	Free-Real
	RED	Diesel Fuel Requirement-Rail Use, liters/ shipping container	Free-Real
<hr/> PRE-OPERATIONS (Page 4.19) <hr/>			
1	DATYPE	Data Category Description	20A4
2	SCDF(L)	Startup Cost Distribution Schedule fraction/ year (where L = 1, IYRPRO - IYRCON)	Free-Real
	CLNDF(L)	Land Cost Distribution Schedule, fraction/ year (where L = 1, IYRPRO - IYRCON)	Free-Real
	CLCDF(L)	Licensing Cost Distribution Schedule, fraction/year (where L = 1, IYRPRO-IYRCON)	Free-Real
3	TCLAND	Land Cost, \$	Free-Real
	TCLICE	Licensing Cost, \$	Free-Real
	STPCT	Start-Up Cost, fraction of field cost	Free-Real
	AECON1	A-E Cost Function Constant (see Page 4.19)	Free-Real
	AECON2	A-E Cost Function Constant (see Page 4.19)	Free-Real
	IEQTP4	A-E Cost Functional Form (1 = linear, 2 = exponential)	Free-Real
<hr/> POST OPERATIONS (Page 4.19) <hr/>			
1	DATYPE	Data Category Description	20A4
2	DBDF(I)	Shaft Pillar & Corridor Backfilling Cost Distribution Schedule, fraction/year (where I = 1, IDDYR)	Free-Real
3	SSDF(I)	Shaft Sealing Distribution Schedule, fraction/year (where I = 1, IDDYR)	Free-Real
4	FDDF(I)	Facility Decommissioning Schedule Factor, fraction (where I = 1, IDDYR)	Free-Real
5	SSCON1	Shaft Sealing Cost Function Coefficient (see Page 4.20)	Free-Real
	SSCON2	Shaft Sealing Cost Function Coefficient (see Page 4.20)	Free-Real
	IEQTP5	Shaft Sealing Cost Functional Form (1 = linear, 2 - exponential)	Free-Integer

TABLE 5.1. (contd)

Card	Variable	Function/Variable Description	Format
5	(contd)		
	DCOMCF	Support Facility Decommissioning Cost, fraction of construction cost	Free-Real
	SURECST	Annual Surveillance Cost, \$	Free-Real
	ITSURV	Surveillance Time Period, years	Free-Integer
6*	NDDC	Enter "0" (zero)	Free-Integer
	DDFRAC	Enter "0" (zero)	Free-Real
7*	SCCST	Enter "0" (zero)	Free-Real
	SCLBRR	Enter "0" (zero)	Free-Real
	SCLBRC	Enter "0" (zero)	Free-Real
	SCEQPC	Enter "0" (zero)	Free-Real
	SCMATU	Enter "0" (zero)	Free-Real
	SCMATC	Enter "0" (zero)	Free-Real
	*	These variables reserved for future development. Input zeros for proper run termination.	
8	TABLES	Desired tables to be printed. Omit card if all tables are required.	Free-Integer

6.0 THE INTERACTIVE PREPROCESSOR

Because of the large amount of input data required for RECON, an interactive program named the PREPROCESSOR was written to aid the user in developing data input. This section describes the operation and use of the PREPROCESSOR.

The PREPROCESSOR is a menu-driven program that creates an input data file written in the correct format for execution by the RECON model. The data file may be created using data from either an existing case or from all new data entered via the terminal. Following data entry the user has the option of verifying (and changing if necessary) data in a newly developed case or exiting and writing the file to disc storage prior to execution by RECON.

PROGRAM DESCRIPTION

The PREPROCESSOR is written in standard ASCII Fortran on the UNIVAC 1170 computer. Figure 6.1 shows the main program segments and the flow of logic. The PREPROCESSOR user is first asked to choose one of three menu options. These options define 1) whether a whole new database is to be created, 2) if an existing case is to be updated, or 3) whether the PREPROCESSOR is to be exited. The user then selects a series of options that define how many and which data categories will be changed. If menu Option 1 is chosen at the beginning, all category options are automatically added. The data descriptions and values on the input file will be listed for each data section chosen. After making the appropriate changes in a data category, the user is asked if additional categories need to be changed if menu Option 2 has been selected. If yes, the user proceeds until all changes are made. When all changes are made, the user is returned to the main menu to either verify the data or exit the PREPROCESSOR. The PREPROCESSOR then writes the appropriate files, indicates that the user can now execute RECON, and the program terminates.

The main advantages of using the PREPROCESSOR for data input instead of card files are that 1) like data is grouped in one place making input variables easier to find, and 2) all of the necessary data items in the variable input arrays are called for. This avoids the problem of forgetting some data inputs when changing the number of waste types or the number of years of operation,

FIGURE 6.1. Program Logic and Flowsheet for the PREPROCESSOR

for example. Another benefit from using the PREPROCESSOR is that a hard copy of the complete data base identified by variable name and description can be obtained for reference purposes.

When writing the output data files, the PREPROCESSOR uses the same array names as used in the RECON read file. This facilitates tracing of data from the PREPROCESSOR input into RECON, should this be necessary.

INSTRUCTIONS FOR USING THE PREPROCESSOR

This section contains instructions for executing the PREPROCESSOR and describes in more detail the input options and screen messages.

Before the PREPROCESSOR is executed, certain files must be assigned for proper execution. These are discussed below.

File 7 is an output file generated by the PREPROCESSOR to give the user a list of input data values described by name. This file can be printed or destroyed depending upon the user's choice after the PREPROCESSOR has completed execution. This file, however, represents the only complete input data listing as RECON does not list all of the input data.

File 8 is an output file designed to receive the newly generated case. This file can either be a temporary or a permanent file depending on the user's requirements. The data sequence on this file is an exact image of the data sequence read by RECON. This is the file submitted for execution as an input data file for RECON2.

The options available within the PREPROCESSOR are as follows:

1. Develop data for a new case

Only files 7 and 8 need be assigned with this suboption. The program automatically requires that all data categories be reviewed.

2. Update an existing case

In addition to files 7 and 8, file 11 must also be assigned. File 11 must contain a previously created case that the user wishes to modify and it must be in the proper data format. The user may choose to enter all new data or specific categories.

3. Exit from the PREPROCESSOR

When the user activates this option, the PREPROCESSOR calls the subroutines where that write files 7 and 8. The PREPROCESSOR then asks the user whether all the tables are to be printed or only specific ones. If the answer is all tables the PREPROCESSOR will return to the main program and exit. If the response is specific tables the PREPROCESSOR will query the user whether table headings are to be listed. If the answer is no table headings, the user will input the total number of tables followed by the appropriate table number separated by a space or a comma. If the response is to print the table headings, the PREPROCESSOR will list the table headings in 3 groups. After the first group (Tables 1 through 9), the user is prompted to enter the table numbers separated by a space or comma. If no tables are requested from this first group a carriage return is entered. The PREPROCESSOR will then proceed to the second group of Tables 10 through 22. The same logic is used as in the first group of tables. The third group contains Tables 23 through 27.

When all entries have been completed the PREPROCESSOR will write this information to file 8.

The user may wish to assign only the necessary files or he may assign all the files listed above, which will give the user the freedom of executing more than one variation during an execution of the PREPROCESSOR.

After an option is chosen, the PREPROCESSOR gives the user the choice of entering all data for a new case or only data pertaining to specific areas for Options 1a, 2 and 3. If the option to enter data for only specific areas is chosen, the PREPROCESSOR lists all categories and requests the user to choose one. If the category chosen contains a further subdivision by section within that category, the PREPROCESSOR will list all sections pertaining to that category and request the user to enter the section codes he wishes to update or

verify. The PREPROCESSOR will only update or verify one category at a time. However, multiple sections can be entered for updating within each category. If the user has requested a category or section that deals with data that differs by waste type, the PREPROCESSOR will prompt the user to input an individual waste type index number(s) or will allow the user to default to the waste type index numbers that reside in the reference data case. More than one waste type can be input by simply entering the index numbers separated by a space or a comma. The outline below shows the organization of the data into categories and sections in the order used in the PREPROCESSOR:

Categories:

1. General Systems data
 - a. Case title
 - b. Site Parameters
 - c. Constraints
 - d. Repository Timing and Waste Source Term Schedules
 - e. Construction and Cost Expenditure Schedules
 - f. Miscellaneous Cost Parameters
2. Waste Handling Systems
 - a. General Waste Descriptions
 - b. Receiving Data
 - c. Packaging Data
 - d. Transfer Data
 - e. Emplacement Operations Data
3. Repository Systems
 - a. Panel Parameters
 - b. Room Parameters
 - c. Corridor Parameters
4. Shaft Description Data
 - a. Men and Material Shaft Data
 - b. Remote-Handled Waste Shaft Data
 - c. Contact-Handled Waste Shaft Data
 - d. Mine Ventilation Shaft Data
 - e. Mining Exhaust Shaft Data
 - f. Waste Exhaust Shaft data

- g. Rock Hoisting Shaft Data
- h. Shaft Functions Shaft Data
- 5. Rock Handling and Disposal Descriptions and Costs
 - a. General Information
 - b. Rock Excavation
 - c. Rock Hoisting
 - d. Surface Storage
 - e. Offsite Disposal
 - f. Backfilling
- 6. Support Facilities
- 7. Utilities
 - a. Electrical
 - b. Steam
 - c. Potable Water
 - d. Demineralized water
 - e. Diesel fuel
- 8. Pre-Operation Support
- 9. Decommissioning and Surveillance

After receiving the requested category and sectional code options, the PREPROCESSOR stores this information in a category option matrix. As the PREPROCESSOR continues execution, the category option matrix is queried when a new section is reached in the programming. If the matrix does not contain a value in the position reserved for the current section, the coding will bypass that section and jump to the next section where the process is repeated. Only when the option matrix contains a value for the current section will the PREPROCESSOR begin to list the descriptions of the variables with the default values from the reference data base or an old case. Care should be taken when viewing values on the screen, because the values are written with a G15.5 format. This format was chosen because of the wide range of values for data. If the user wishes to change the value, he simply enters the correct value. The input is read as character data, so the user is not required to follow any restrictive formats. After reading the value, the PREPROCESSOR will make the correct conversion to either real or integer, using the DECODE

function, depending on the variable type. If the user wishes to retain the referenced data base value as his current value, he simply depresses the return key on the keyboard.

When the PREPROCESSOR completes the category chosen, control is passed back to the menu where the user may wish to select another category. If no further categories need updating, control is passed back to the main option menu. When the user has completed all data entry and selects the option to exit from the PREPROCESSOR, only then are the new data listings and the new case files (files 7 and 8) created with values that are current. The new case file that is created is in the required format for execution with the RECON code. After completing creation of the necessary files, the PREPROCESSOR indicates that RECON may now be executed and ends.

7.0 REFERENCES

Parsons, et al. April 1978. Technical Support for GEIS: Radioactive Waste Isolation in Geologic Formations, Volume 18. Y/OWI/TM-36/18, Parsons Brickerhoff, Quade and Douglas, Inc., New York, New York.

U.S. Department of Energy (DOE). Final Environmental Impact Statement - Management of Commercially Generated Radioactive Waste - Volume I of 3. National Technical Information Service, Springfield, Virginia.

APPENDIX A

MODEL VARIABLE DESCRIPTIONS

APPENDIX A

MODEL VARIABLE DESCRIPTIONS

The following tables list descriptions of all model variables with the variable names and units and show where the variables are used in the model. Where no units are shown, the variable is either unitless or is an alphanumeric. All variables are shown in alphabetic order. The variable listing for the RECON model itself is presented first, followed by the variable listing for the PREPROCESSOR.

RECON VARIABLE LISTING

Variable Name	Description	Units	Location
A	Dummy variable	--	HOLDC, HOLMC, SKFUNC, OUTPUT, SPUT, TPUT
ABDCST	Annual backfilling and decommissioning cost	\$/year	OUTPUT, PROCED
AC	Annual capacity (dummy variable)	waste units/year	REPLAS
ACBFC	All corridor excavation backfilling and sealing costs	\$	OUT22
ACBKFL	Accumulated backfill	tonnes	OUT22
ACONTG	Annual contingency cost	\$/year	HUB, OUTPUT, PROCED
ACP	Air conditioning building load	kW	INPUT, UTILITY, PROCED
ACPOW	Air conditioning power requirement	kWh	UTILITY
ACPOWC	Cumulated air conditioning power requirement	kWh	OUTPUT, PROCED, UTILITY
AECON	Constant AECON1 allocated by waste type	--	PREOP
AECON1	Constant used to estimate A-E costs	--	INPUT, PREOP, PROCED
AECON2	Constant used to estimate A-E costs	--	INPUT, P#EOP, PROCED
AESERV	A-E costs	dollars	PREOP, HUB, OUT22, PROCED
AMLBCT	Annual emplacement labor cost	\$/year	WASTOP, HUB, OUTPUT, PROCED
ANBCT	Annual backfilling cost	\$/year	HUB, OUTPUT, PROCED
ANCPTY	Annual mining capacity	tonnes/year	MINDEV, PROCED
ANDDCT	Annual decommissioning cost	\$/year	OUTPUT, POSTOP, PROCED
ANEQCP	Annual equipment capacity (dummy variable)	units/year	ROCKOP
ANFILD	Annual rooms filled with waste	--	MINDEV, OUTPUT, PROCED
ANNEX	Annual rooms excavated	--	MINDEV, OUTPUT, PROCED
ANPAR	Annual panel area required	acres/year	WASTOP, OUTPUT, PROCED

A.1

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
ANSLCT	Annual shaft and tunnel sealing cost	\$/year	OUTPUT, POSTOP, PROCED
ANUTC	Annual utilities cost	\$/year	HUB, UTILITY, WAFAM, OUTPUT, PROCED
APLBCT	Annual packaging labor cost	\$/year	WASTOP, HUB, OUTPUT, PROCED
AREAPP	Area per panel	acres/panel	WASTOP, MINDEV, PPUT, PROCED
ARLBCT	Annual receiving labor cost	\$/year	WASTOP, HUB, OUTPUT, PROCED
ASHIFT	Annual shifts worked at repository	--	OUTPUT
ASPC	Annual site preparation cost	\$/year	HUB, SURFST, PROCED
ATLBCT	Annual transportation labor cost	\$/year	WASTOP, HUB, OUTPUT, PROCED
AVLRPA	Available repository area	acres	WASTOP, OUTPUT, PROCED
AXLBCT	Annual labor cost for emplacement excavation	\$	HUB, OUTPUT, PROCED, WASTOP
B	Dummy variable	--	OUTPUT, HOLDC, HOLMC, SKFUNC
BCKFTH	Backfill material thickness	meters	INPUT, PROCED, WASTOP
BEGPA	Beginning panel area available	acres	WASTOP
BFLEFF	Backfilling efficiency (percentage of theoretical room volume filled)	--	MINDEV, ROCKOP
BFLMCS	Backfill materials cost	\$	HOLMC
BKFLBR	Backfilling labor required	man-year	HUB, PROCED, ROCKOP, WAFAM
BKFLC	Backfilling cost	\$m ³ (\$/MT)	HUB, INPUT, PROCED, OUTPUT
BKFLYR	Last year of backfilling	integer	ROCKOP
BKFMAT	Starting year of backfilling	integer	ROCKOP, HUB, PROCED
BKFYRS	Backfilling materials required	tonnes	ROCKOP

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
BKRATE	Backfilling rate	tonnes/year	INPUT, PROCED, ROCKOP
BKSHFT	Work shifts for backfilling after waste operations are ended	--	PROCED, ROCKOP
BLANK	Character constant	--	INPUT, OUTPUT, PROCED
BSFILD	Bore sections filled with emplaced waste	--	WASTOP, PPUT, PROCED
C	Dummy variable	--	HOLMC, OUTPUT, SSTOR, WASTOP
CAPCTY	Equipment capacity (dummy variable)	--	WASTOP
CAPLEN	Cap length	meters	INPUT, PROCED, WASTOP
CAREW	Cumulative emplaced area by waste type	acres	MINDEV, OUTPUT, PROCED
CCCON1	Lining cost function constant 1	--	INPUT, PROCED, SHAFT
CCCON2	Lining cost function constant 2	--	INPUT, PROCED, SHAFT
CEOC	Subtotal all corridor excavation during years of operation	\$	OUT22
CEQPT	Equipment replacement costs associated with waste handling and emplacement	\$	HUB, OUTPUT, OUT22, PROCED
CFSPCT	Contact handling floor space percentage	fraction	PROCED
CHMTEM	CH waste materials cost	\$/unit pitch	INPUT, PROCED, WASTOP
CHRECU	CH receiving units	waste units	WAFAM
CHRECW	CH receiving units by waste type	waste units	PROCED, WAFAM
CHRLBR	CH receiving labor requirement	man-year	PROCED, WASTOP
CHTRU	CH Tru waste storage	waste units	PROCED, SSTOR
CLABOR	Labor costs associated with waste handling and emplacement	\$	HUB, OUTPUT, OUT22, PROCED

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
CLCDF	Time distribution factors for licensing costs	--	INPUT, PREOP, PROCED
CLNDF	Time distribution factors for land costs	--	INPUT, PREOP, PROCED
CLSEMP	Closest emplacement to corridor	meters	INPUT, PROCED, WASTOP
CMATL	Material costs associated with waste handling and emplacement	\$	HUB, OUTPUT, OUT22, PROCED
CMCAP	CH waste receiving module capacity	waste units	OUTPUT, PROCED, WASTOP
CMMAS	Cumulative corridor rock mass mined	tonnes	HUB, OUT22, PROCED
CMMAX	Contact-handled waste receiving module capacity	waste units/year	WASTOP
CNTBKF	Backfilling and sealing contingency costs	\$	OUT22
CNTCSB	Total capital construction contingency costs	\$	OUT22
CNTDD	Decommission contingency costs	\$	OUT22
CNTEQC	Capital equipment contingency costs	\$	OUT22
CNTEQR	Replacement equipment contingency cost	\$	OUT22
CNTFCC	Facility construction contingency cost	\$	OUT22
CNTLBR	Labor (operations) contingency cost	\$	OUT22
CNTMAT	Material (operations) contingency cost	\$	OUT22
CNTOSB	Total contingency for operations	\$	OUT22
CNTUTL	Utility contingency cost	\$	OUT22
COCON	Number of non-heat-generating waste packages that can be coemplaced per primary waste package	--	INPUT, PROCED, SPUT, TPUT, WASTOP
CONFAC	Contingency factor	fraction	HUB, INPUT, PROCED
CONTGY	Contingency costs	\$	HUB, PROCED

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
CONTRU	Number of CH packages that can be emplaced per TRU waste packages for each primary waste type	--	INPUT, PROCED, SSTOR
CORARA	Main corridor area	acres	OUTPUT, MINDEV, PROCED, ROCKOP, UTILITY, WASTOP
CORDSC	Corridor descriptions	--	INPUT, OUTPUT, PROCED
CORHT	Corridor height	meters	INPUT, MINDEV, OUTPUT, PROCED, ROCKOP
CORLEN	Corridor length per room	meters/room	INPUT, OUTPUT, PROCED, MINDEV, WASTOP
CORMAS	Corridor rock mass mined	tonnes	MINDEV, PROCED, HUB, PREOP
CORVOL	Corridor rock volume mined	m ³	MINDEV, PROCED, HUB, OUT22
CORVR	Corridor volume per room	m ³	MINDEV
CORWTH	Corridor width	meters	INPUT, MINDEV, OUTPUT, PROCED, ROCKOP, UTILITY, WASTOP
COSTLC	Annual licensing costs by waste type	\$/year	HUB, PREOP, PROCED
COSTLD	Annual land costs by waste type	\$/year	HUB, PREOP, PROCED
COWAST	Waste coemplacement	tonnes	OUTPUT, PROCED, SPUT, TPUT
CPA	Central pillar area	acres	INPUT, OUTPUT, PROCED, WASTOP
CPCMM	Central pillar corridor rock mass mined	tonnes	MINDEV, PROCED, HUB, PREOP
CPEQ	Cost of equipment replacement for waste package preparation during operations by waste type	\$	HUB, OUT22, PROCED
CPLBR	Cost of labor for waste package preparation during operations by waste type	\$	HUB, OUT22, PROCED
CPMAT	Cost of materials for waste package preparation during operations by waste type	\$	HUB, OUT22, PROCED

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
CRRAIL	Percent of waste units shipped by rail	fraction	INPUT, OUTPUT, PROCED, RECEVE, WAFAM, WASTOP
CSTAE	Cost of A-E services by year and waste type	\$	OUTPUT, PREOP, PROCED
CSTBKF	Annual backfilling costs by waste type	\$/year	HUB, PROCED
CSTHST	Annual rock hoisting costs by waste type	\$/year	HUB, PROCED
CSTLIC	Cost of licensing by year and waste type	\$	OUTPUT, PREOP, PROCED
CSTLND	Cost of land by year and waste type	\$	OUTPUT, PREOP, PROCED
CSTOSD	Cost of offsite disposal	\$	HUB, PROCED
CSTOT	Annual total construction costs	\$	OUTPUT, PROCED
CSTSF	Annual surface facility construction costs	\$	HUB, OUTPUT, PROCED
CSTSHS	Annual rock surface handling costs by waste type	\$	HUB, PROCED
CSTSHT	Cost of shafts by year and waste type	\$	HUB, PROCED, OUTPUT
CSTSP	Annual site preparation costs	\$	OUTPUT, PREOP, PROCED, SURFST
CSTSU	Annual startup costs	\$	OUTPUT, PREOP, PROCED
CSTUGF	Cost of underground facilities by year and waste type	\$	HUB, OUTPUT, PROCED
CUMBKF	Cumulative rock backfilled by year	tonnes	MINDEV
CUMFLD	Cumulative rooms filled	--	MINDEV, PROCED
CUMPAR	Cumulative panel area	acres	OUTPUT, PROCED, WASTOP
D	Diameter	meters	HOLDC, HOLMC, OUTPUT, POSTOP, SHAFT, SKFUNC
DA	Today's date	--	HUB, OUTPUT, PROCED
DASH	Character constant	--	OUTPUT

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
DATE	Internal date function in computer	--	HUB, OUTPUT
DATYPE	Subroutine data name	--	INPUT
DBDF	Decommissioning backfill distribution factor	--	INPUT, PROCED, ROCKOP
DCOMCF	Facility decommissioning cost factor	fraction	INPUT, POSTOP, PROCED
DDCST	Variable defined for future usage (not currently used)	--	INPUT, PROCED
DDEQPC	Variable defined for future usage (not currently used)	--	INPUT, PROCED
DDFRAC	Variable defined for future usage (not currently used)	--	INPUT, PROCED
DDLBRC	Variable defined for future usage (not currently used)	--	INPUT, PROCED
DDLBRR	Variable defined for future usage (not currently used)	--	INPUT, PROCED
DDMATC	Variable defined for future usage (not currently used)	--	INPUT, PROCED
DDTITL	Variable defined for future usage (not currently used)	--	INPUT, PROCED
DECOMC	Decommissioning cost by year and waste type	\$	HUB, POSTOP, PROCED
DELRHT	Delta room height	meters	INPUT, MINDEV, PROCED
DENSTY	Rock density	tonnes/m ³	INPUT, MINDEV, PROCED, ROCKOP, WASTOP, HUB, OUT22
DEPTH	Mine depth	meters	INPUT, OUTPUT, PROCED
DIESEL	Diesel fuel requirements by waste type and time	liters	OUTPUT, PROCED, UTILITY
DNEMPU	Number of emplacement units (dummy variable)	--	WASTOP
DPY	Days per standard work-year	--	MINDEV, PROCED, ROCKOP, SHAFT, WASTOP
DRATE	Discount rate	--	HUB, INPUT, OUTPUT, POSTOP, PROCED

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
DRECNT	Dummy variable for receiving and packaging waste quantities	--	WASTOP
DSLC	Diesel fuel unit cost	\$/liter	INPUT, PROCED, UTILITY
DSLCST	Diesel fuel cost	\$/year	OUTPUT, PROCED, UTILITY
DSMINE	Unit diesel fuel use in mining	liter/tonne	UTILITY
DSRAIL	Unit diesel fuel use in rail operations	liter/arrival	UTILITY
DSRKHT	Dummy variable for annual rock hoisted	tonnes/year	ROCKOP
DSURF	Unit diesel fuel use in waste operations	liter/shipment	UTILITY
DUMPKG	Number of packages (dummy variable)	--	WASTOP
DWTRC	Unit Demineralized water cost	\$/10 ⁶ liters	INPUT, PROCED, UTILITY
E	Dummy variable	--	HOLMC, OUTPUT
ECOST	Annual emplacement excavation costs by waste type	\$/year	HUB, OUTPUT, PROCED
EEC	Total capital construction costs for waste emplacement excavation	\$	OUT22
EECAPC	Emplacement excavation capital cost by waste type and time	\$	HUB, PROCED, SURFST
EECST	Capital cost of emplacement excavation	\$	INPUT, PROCED, SURFST
EEECST	Unit equipment cost for emplacement excavation	\$/equip. unit	HUB, INPUT, PROCED, WASTOP
EEEQPC	Emplacement excavation equipment capacity	empl. units/equip./shift	INPUT, PROCED, WASTOP
EEEQPL	Emplacement excavation equipment life	empl. units/equip. unit	INPUT, OUTPUT, PROCED, WASTOP
EELBRC	Unit emplacement excavation labor cost	\$/man-year	INPUT, PROCED, WASTOP
EELBRR	Unit emplacement excavation labor requirement	man-year/equip./shift	INPUT, PROCED, WASTOP

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
EEMATC	Unit emplacement excavation materials cost	\$/material unit	INPUT, PROCED, WASTOP
EEMATR	Unit emplacement excavation materials by year and waste type	material units	PROCED, WASTOP
EEMATU	Unit emplacement excavation materials	material units/empl. unit	INPUT, PROCED, WASTOP
EEO	Total operations costs for waste emplacement excavation	\$	OUT22
EEXLBR	Emplacement excavation labor required by year and waste type	man-year	HUB, PROCED, WASTOP, WAFAM
EEXRKM	Rock mass excavated for emplacement excavations	tonnes	HUB, PROCED, WASTOP, MINDEV, OUTPUT
EL	Equipment life (dummy variable)	--	REPLAS
EMCST	Waste emplacement cost by year and waste type	\$	HUB, OUTPUT, PROCED
EMEQCT	Annual emplacement equipment costs	\$/year	HUB, OUTPUT, PROCED, WASTOP, OUT22
EMEQPC	Annual emplacement equipment costs by waste type	\$/year	HUB, PROCED, WASTOP
EMLBCT	Emplacement labor cost by year and waste type	\$	HUB, PROCED, WASTOP
EMMATC	Emplacement materials cost by year and waste type	\$	HUB, PROCED, WASTOP
EMMTCT	Annual emplacement material costs	\$/year	OUTPUT, PROCED, WASTOP
EMPLBR	Emplacement labor requirements by year and waste type	man-year	HUB, PROCED, WASTOP, WAFAM
EMPMCH	Number of emplacement machines	--	HUB, OUTPUT, PROCED, WASTOP
EQLIFE	Equipment life (dummy variable)	--	WASTOP
EQREEX	New equipment replacements for emplacement excavation	equipment units	HUB, PROCED, WASTOP
EQREMP	New equipment replacements for emplacement	equipment units	PROCED, WASTOP
EQRPKG	New equipment replacements for packaging	equipment units	PROCED, WASTOP

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
EQREC	New equipment replacements for receiving	equipment units	PROCED, WASTOP
EQTRN	New equipment replacements for transportation	equipment units	PROCED, WASTOP
ERC	Capital construction costs for room excavation	\$	OUT22
ERCLBR	Emplacement labor for CH wastes by year and shift	man-year	PROCED, WASTOP
ERO	Total operations costs for room excavation	\$	OUT22
ERRLBR	Emplacement labor for RH wastes by year and shift	man-year	PROCED, WASTOP
EXEQCT	Annual emplacement excavation equipment costs	\$/year	HUB, OUTPUT, PROCED, OUT22, WASTOP
EXEQPC	Annual emplacement excavation equipment costs by waste type	\$/year	HUB, PROCED, WASTOP
EXLBCT	Annual emplacement excavation labor costs by waste type	\$/year	HUB, PROCED, WASTOP
EXLBR	Emplacement excavation labor by year and shift	man-year	PROCED, WASTOP
EXMAC	Excavation machines	--	WASTOP
EXMACH	Emplacement excavation machine requirements	equipment units	HUB, OUTPUT, PROCED, WASTOP
EXMATC	Emplacement excavation materials costs by year and waste type	\$	HUB, PROCED, WASTOP
EXMTCT	Annual emplacement excavation materials costs	\$/year	OUTPUT, PROCED, WASTOP
EXPLIF	Expended machine life	life units	REPLAS
F	Borehole diameter	meters	HOLMC
FDDF	Facility decommissioning distribution factor	--	INPUT, POSTOP, PROCED
FFHTLD	Far-field heat load	kilowatts	PROCED, WASTOP
FFTL	Far-field thermal limit	kW/acre	INPUT, PROCED, WASTOP
FLDCST	Field cost	\$	PREOP, PROCED

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
FMOD	Fractional number of component modules	--	WASTOP
FULL	Maximum # CHTRU capacity in existing rooms	waste units	SPUT, SSTOR, TPUT
G	Dummy variable	--	HOLMC
GXEF	Geometric exclusion factor	acres	INPUT, OUTPUT, PROCED, WASTOP
H	Dummy variable	--	HOLMC
HASFEC	Capital equipment cost for rock hoisting and RH shaft	\$	OUT22
HDCON1	Hole drilling constant 1	--	INPUT, PROCED, WASTOP
HDCON2	Hole drilling constant 2	--	INPUT, PROCED, WASTOP
HDECFC	Hole drilling equipment unit cost fraction	--	INPUT, PROCED, WASTOP
HDLFCFC	Hole drilling labor unit cost fraction	--	INPUT, PROCED, WASTOP
HDMASS	Cumulative rock mass excavated for emplacement excavation	tonnes	HUB, OUTPUT, OUT22, PROCED
HDMAST	Cumulative rock mass excavated for emplacement excavation divided by 1000	tonnes	OUTPUT
HDMCFC	Hole drilling materials unit cost fraction	--	INPUT, PROCED, WASTOP
HEAT	Total steam requirement for heating	lbs	UTILITY
HEATS	Unit steam heating requirement underground	lbs/acre of repository	INPUT, PROCED, UTILITY
HOLD	Hole depth	meters	PROCED, WASTOP
HOLDC	Hole drilling cost function	--	HOLDC, WASTOP
HOLDEL	Hole depth delta	meters	INPUT, PROCED, WASTOP
HOLDIA	Hole diameter	meters	PROCED, WASTOP
HOLMC	Hole material cost function	--	HOLMC, WASTOP

A.11

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
HOLXC	Hole excavation cost	\$/hole	PROCED, WASTOP
HPMIN	Minimum hole pitch	meters	INPUT, PROCED, WASTOP
HPY	Hours per year	--	UTILITY
HSTCAP	Rock holsting shaft capacity	tonnes	SHAFT
HTLDNF	Near-field heat load	kilowatts	PROCED, WASTOP
I	Index (usually refers to year)	--	HUB, INPUT, MINDEV, OUTPUT, POSTOP, PREOP, RECEVE, ROCKOP, SHAFT, SURFST, UTILITY, WAFAM, WASTOP
IB	Index	--	HUB, OUTPUT, SURFST
ICH	Logic flag	--	INPUT
ICLR	Year inventories of stored non-heat generating waste can be cleared by solo emplacement	year	INPUT, PROCED, SSTOR, WASTOP
ICNT	Logic flag	--	WASTOP
IDD	Decommissioning year index	--	HUB, INPUT, OUTPUT, POSTOP, PROCED, ROCKOP, SURFST, WAFAM, OUT22
IDDYR	Year decommissioning starts	calendar year	INPUT
IDET	Detailed cost input switch	--	INPUT, PROCED, WASTOP
IDL	Time from base discounting year to construction startup	years	HUB
IE	Index	--	HUB
IEL	Integer value of equipment life	--	REPLAS
TEMP	Emplacement method	--	EMTYP, MINDEV, PROCED, WAFAM, WASTOP
IEQPL	Integer value of expired equipment life	--	REPLAS

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
IEQTP1	Linear or exponential equation selection switch	--	INPUT, PROCED, SHAFT
IEQTP2	Linear or exponential equation selection switch	--	INPUT, PROCED, SHAFT
IEQTP3	Linear or exponential equation selection switch	--	INPUT, PROCED, SHAFT
IEQTP4	Linear or exponential equation selection switch	--	INPUT, PROCED, SHAFT
IEQTP5	Linear or exponential equation selection switch	--	INPUT, PROCED, SHAFT
IFLAG	Logic flag	--	MINDEV, OUTPUT
IFLG	Logic flag	--	MINDEV, OUTPUT
IFY	First year of new equipment operation	--	REPLAS
IHSTFL	Logic flag	--	OUTPUT, PROCED, SHAFT
II	Index	--	HUB, POSTOP, ROCKOP, SURFST, WASTOP
IICNT	Logic flag	--	WASTOP
III	Index	--	HUB
IMET	Emplacement method switch	--	HUB, INPUT, PROCED, WAFAM, WASTOP
IPMCW	Incremental packaging modules	--	HUB, PROCED
IRMCW	Incremental receiving modules for canistered waste	--	HUB, PROCED
IRMNW	Incremental receiving modules for non-canistered waste	--	HUB, PROCED
IRP	Index	--	REPLAS
IS	Number of Integral shifts	--	WASTOP
ISC	Number of shifts operated for CH waste	--	PROCED, WASTOP
ISR	Number of shifts operated for RH waste	--	PROCED, WASTOP

A.13

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
ISTOP	Logic flag	--	WASTOP
IT	Index marking beginning of surveillance period	--	POSTOP
ITE	Index marking end of surveillance period	--	POSTOP
ITSURV	Surveillance period	years	INPUT, POSTOP, PROCED
IUHDCM	Hole drilling cost calculation method switch	--	INPUT, PROCED, WASTOP
IYDOL	Discount year	calendar year	HUB, INPUT, PROCED
IYR	Year variable used to print calendar year in OUTPUT	--	OUTPUT
IYRCON	Year in which construction begins	calendar year	HUB, INPUT, MINDEV, OUTPUT
IYRDIS	Year to which costs are discounted	calendar year	INPUT, OUTPUT, PROCED
IYRPRO	Year in which production begins	calendar year	INPUT, MINDEV, OUTPUT, PROCED
IZ1	Variable to zero out common blocks	--	HUB, PROCED
IZ2	Variable to zero out common blocks	--	HUB, PROCED
IZ3	Variable to zero out common blocks	--	HUB, PROCED
IZ4	Variable to zero out common blocks	--	HUB, PROCED
IZ5	Variable to zero out common blocks	--	HUB, PROCED
IZ6	Variable to zero out common blocks	--	HUB, PROCED
IZ7	Variable to zero out common blocks	--	HUB, PROCED
IZ8	Variable to zero out common blocks	--	HUB, PROCED
IZ9	Variable to zero out common blocks	--	HUB, PROCED
IZ10	Variable to zero out common blocks	--	HUB, PROCED
IZ11	Variable to zero out common blocks	--	HUB, PROCED

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
IZ12	Variable to zero out common blocks	--	HUB, PROCED
IZ13	Variable to zero out common blocks	--	HUB, PROCED
J	Index	--	INPUT, OUTPUT, OUT22
JNWT	Index	--	OUTPUT
K	Index (usually refers to waste type)	--	HUB, INPUT, MINDEV, OUTPUT, OUT22, POSTOP, PREOP, RECEVE, ROCKOP, SURFST, UTILITY, EMPLAC, EMTYP, PPUT, SPUT, SSTOR, TPUT, WAFAM, WASTOP
KB	Index	--	OUTPUT
KD	Index	--	REPLAS, WASTOP
KE	Index	--	OUTPUT
KIND	Waste type definition: 1 = remote handled 2 = contact handled 3 = TRU	--	HUB, INPUT, OUTPUT, PROCED, RECEVE, SURFST, WAFAM, WASTOP, EMPLAC, EMTYP, MINDEV, SSTOR, TPUT
KK	Waste type Index	--	WASTOP, SSTOR, TPUT
KKN	Index	--	TPUT
KN	Index	--	SPUT, SSTOR, TPUT
KNWT	Index	--	OUTPUT
L	Index	--	HUB, INPUT, MINDEV, OUTPUT, POSTOP, REPLAS, SHAFT, SURFST, WAFAM, WASTOP
LASTYR	Last year of any operation (determines array sizes)	calendar year	INPUT
LB	Index	--	HUB, MINDEV, OUTPUT, POSTOP, ROCKOP, SURFST, UTILITY, WASTOP

A.15

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
LBB	Year Index	--	ROCKOP
LE	Index (usually refers to end of backfilling operations)	--	HUB, MINDEV, OUTPUT, POSTOP, REPLAS, ROCKOP, SURFST, UTILITY, WAFAM, WASTOP
LEAD	Lead time for room excavation (input)	years	INPUT, MINDEV, OUTPUT, PROCED
LED	Lead time for room excavation	years	MINDEV, OUTPUT
LEE	Total years required for construction, operation, back- filling and decommissioning	years	OUTPUT
LL	Index (usually refers to end of construction)	--	HUB, INPUT, MINDEV, OUTPUT, PREOP, PROCED, SURFST, UTILITY, WASTOP
LLL	Index for first year of repository operations	--	WASTOP
LLP1	Index for first year of repository operations	--	HUB
LOOP	Index	--	OUTPUT
LX	Index	--	OUTPUT
LY	Index (usually refers to end of waste operations)	--	HUB, INPUT, MINDEV, OUTPUT, PROCED, RECEVE, ROCKOP, SHAFT, SURFST, WAFAM, WASTOP
LYY	Index referring to last year of program operation	--	INPUT, UTILITY
M	Index	--	INPUT, MINDEV, OUTPUT, REPLAS
MEOP	Maximum number of equipment replacements	equipment units	REPLAS
MINEC	Rock operations costs	\$	OUT22
MINEQC	Rock operations replacement equipment cost	\$	OUT22
MODFLG	Logic flag	--	OUTPUT, PROCED, WASTOP

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
MPRIME	Number of primary heat-generating waste types	--	EMPLAC, EMTYPE, INPUT, MINDEV, OUTPUT, PROCED, SPUT, Sstor, TPUT, WAFAM, WASTOP
MRFLG	Logic flag	--	MINDEV, OUTPUT, PROCED
MROP	Mined rock operations power	kW/tonne	INPUT, PROCED, UTILITY
MSA	Mined rock storage area	acres	INPUT, MINDEV, PROCED
MSFLG	Logic flag	--	MINDEV
MSHIFT	Number of shift per day worked (250-day year)	shifts/day	INPUT, MINDEV, PROCED, ROCKOP
MSWCH	Mining cost input type selector	--	HUB, INPUT, MINDEV, PROCED, ROCKOP, OUTPUT
MTCPSD	Maximum shaft trips per shift for CH waste	--	INPUT, OUTPUT, PROCED, WASTOP
MTRPSD	Maximum shaft trips per shift for RH waste	--	INPUT, OUTPUT, PROCED, WASTOP
MXMMRH	Maximum trips per shift-year for rock hoisted in men and materials shaft	tonnes/year	INPUT, OUTPUT, PROCED, ROCKOP SHAFT
MXMPRH	Maximum trips per shift-year for rock hoisted in mine production shaft	tonnes/year	INPUT, OUTPUT, PROCED, ROCKOP SHAFT
M2ACRE	Square meters per acre constant	--	MINDEV, PROCED, ROCKOP, UTILITY, WASTOP
N	Index	--	HOLDC, HOLMC, HUB, INPUT, OUTPUT, REPLAS
NA	Number of equipment units required for a given year	equipment units	REPLAS
NAA	Index	--	REPLAS
NALSH	Number of shifts allowed	--	INPUT, PROCED, WASTOP
NB	Index	--	REPLAS
NBEQP	Equipment requirements in current year (dummy variable)	--	ROCKOP

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
NBKF	Total years required for construction, operation and backfilling	years	HUB, OUTPUT, POSTOP, PROCED, ROCKOP, MINDEV, OUT22
NBKYRS	Integer number of years required for backfilling after waste operations	years	ROCKOP
NBSPPA	Number of bore sections per panel	--	INPUT, PROCED, WASTOP
NBYR	Time from waste operations startup to backfilling startup	years	INPUT, MINDEV, PROCED, ROCKOP
NC	Maximum number of construction years	--	PROCED
NCBPRM	Number of receiving bays per module - CH		INPUT, PROCED, WASTOP
NCHRNA	Maximum CH waste-receiving modules	--	INPUT, PROCED, WASTOP
NCOR	Number of corridor types in a panel	--	INPUT, MINDEV, OUTPUT, PROCED
NCORPP	Number of corridors per panel	--	INPUT, MINDEV, OUTPUT, PROCED
NCR	Maximum number of corridor types defined	--	PROCED
NDC	Number of decommissioning cost categories	--	PROCED
NDDC	Variable set aside for future use (not currently used)	--	INPUT, PROCED
NDSMP	Number of mine production shafts	--	PROCED, ROCKOP, SHAFT
NDUM	Dummy variable	--	MINDEV
NEE	Number of emplacement excavation methods	--	INPUT, OUTPUT, PROCED, SURFST, WAFAM
NEMPU	Number of emplacement units	--	PROCED, WASTOP
NEO	Maximum emplacement types	--	PROCED
NEQC	Number of equipment units needed in current year (dummy)	--	WASTOP

RECON VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
NEQLY	Number of equipment units needed in previous year (dummy)	--	WASTOP
NEQRPL	Equipment replacements for rock operations	--	HUB, MINDEV, PROCED, ROCKOP, OUT22
NEWEQP	Equipment replacements (dummy variable)	equipment units	WASTOP
NFAC	Maximum number of surface facility categories	--	PROCED
NFLG1	Logic flag	--	WASTOP
NFTL	Near-field thermal limit	kilowatts/acre	INPUT, PROCED, WASTOP
NHLPPA	Number of heat-limited packages per unit area	--	WASTOP
NLAST	Dummy variable indexing the previous year	--	MINDEV, ROCKOP
NM	Index	--	REPLAS, WASTOP
NMMACH	Number of mining machines required each year	--	MINDEV, PROCED
NMODCH	Total number of CH waste-receiving modules constructed	--	PROCED, RECEVE
NMODPH	Total number of packaging modules constructed	--	RECEVE
NMODRH	Total number of RH waste-receiving modules constructed	--	PROCED, RECEVE
NN	Number of waste types plus 1	--	OUTPUT
NNHG	Number of non-heat-generating waste types	--	INPUT, OUTPUT, PROCED
NPKGPR	Number of packages per room	--	OUTPUT, PROCED, WASTOP
NPKGTT	Total number of waste packages for each waste type	--	OUTPUT
NPKPEU	Number of packages per emplacement unit	--	INPUT, PROCED, WASTOP
NPKPTC	Number of packages per shaft trip - CH	--	INPUT, OUTPUT, PROCED, WASTOP
NPKPTR	Number of packages per shaft trip - RH	--	INPUT, OUTPUT, PROCED, WASTOP

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
NPKPTT	Total remote and contact packages per shaft trip	--	OUTPUT
NPKSTM	Total packages processed each year	package units	PROCED, WAFAM
NPKSUM	Total packages processed in entire repository	package units	WAFAM
NPMRH	Number of packaging modules required in any year	--	HUB, OUTPUT, PROCED, RECEVE, WASTOP
NPPRW	Number of packages per row	--	OUTPUT, PROCED, WASTOP
NPSMWT	Total number of packages processed for each waste type	--	PROCED, WAFAM
NPBPRM	Number of receiving bays per module - RH	--	INPUT, PROCED, WASTOP
NRHRMA	Maximum RH waste-receiving modules	--	INPUT, PROCED, WASTOP
NRMCH	Number of CH waste-receiving modules in any year	--	HUB, OUTPUT, PROCED, RECEVE, WASTOP
NRMRH	Number of RH waste-receiving modules in any year	--	HUB, OUTPUT, PROCED, RECEVE, UTILITY, WASTOP
NRO	Number of rock operations	--	INPUT, PROCED, SURFST
NRON	Rock operations number index	--	HUB, MINDEV, ROCKOP
NRWSPR	Number of rows per room	--	INPUT, PROCED, WASTOP
NS	Number of labor cost differentials for shift work	--	INPUT, PROCED
NSC	Number of support facilities	--	INPUT, OUTPUT, PROCED, SURFST, OUT22
NSFELF	Surface facility equipment life (in integer years)	years	SURFST
NSFTT	Total number of repository shafts	--	OUTPUT
NSHFLG	Logic flag	--	WASTOP
NST	Number of shaft types	--	HUB, INPUT, MINDEV, OUTPUT, POSTOP, PROCED, SHAFT, OUT22

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
NSTM1	Number of repository shafts minus 1	--	INPUT
NTA	Output table numbering variable	--	OUTPUT
NTB	Total number of waste packages emplaced	--	OUTPUT
NTO	Number of transportation operations	--	INPUT, PROCED
NTPS	Total number of trips per shift in contact and remote waste shafts	--	OUTPUT
NTPSMP	Number of trips per shift in mined rock hoisting shaft	--	OUTPUT
NTYP	Linear or exponential function switch for hole drilling	--	WASTOP, INPUT, PROCED
NU	New equipment replacements (dummy variable)	--	REPLAS
NUMPKG	Number of waste packages	--	HUB, OUTPUT, PROCED, UTILITY, WAFAM, WASTOP, PPUT
NUO	Number of underground operations	--	INPUT, PROCED, SURFST
NW	Number of waste types (for variable dimensions)	--	HUB, MINDEV, OUTPUT, PREOP, PROCED, REPLAS, ROCKOP, WASTOP, OUT22, WAFAM
NWO	Number of surface waste operations	--	INPUT, PROCED
NWP1	Summation Index	--	HUB, INPUT, OUTPUT, PROCED, UTILITY, OUT22
NWT	Number of waste types	--	HUB, INPUT, MINDEV, OUTPUT, POSTOP, PREOP, PROCED, RECEVE, ROCKOP, SURFST, UTILITY, WAFAM, WASTOP, OUT22, SSTOP, TPUT
NWTT	Output Index	--	OUTPUT
NX	Number of mine production shafts (dummy variable)	--	SHAFT

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
NY	Number of years (for variable dimensions)	--	HUB, MINDEV, PROCED, RECEVE, WASTOP, OUTPUT, PREOP
NYRO	Maximum time allowed for all repository activities	years	HUB, PROCED, ROCKOP
OFTADC	Offsite transportation and disposal unit cost	\$/tonne	HUB, INPUT, PROCED
ONEMIL	Constant equal to 1,000,000	--	OUTPUT, PROCED, HUB, OUT22, POSTOP, RECEVE, UTILITY
ONETHO	Constant equal to 1000	--	OUTPUT, PROCED, OUT22
OPMC	Mining cost - annual	\$/year	HUB, PROCED
ORMASS	Cumulative rock mass mined for rooms	tonnes	OUT22
ORS	Steam requirements - other facilities	10 ³ lbs/year	INPUT, PROCED, UTILITY
OSCMAS	Cumulative rock mass mined for corridor	tonnes	OUT22
OSFC	Other surface facility costs	millions \$	HUB, PREOP, PROCED
OSTMC	Cumulative steam cost	\$	OUT22, PROCED, UTILITY
OTBDEN	Other backfill density	g/cm ³	INPUT, MINDEV, PROCED, ROCKOP
OTBKF	Mass of other backfill required	tonnes	HUB, MINDEV, PROCED, ROCKOP, OUT22
OTHBFL	Percentage of other backfill per m ³ backfilled	percent	INPUT, MINDEV, PROCED, ROCKOP
OTMATC	Unit cost for other materials	\$/man-year	PROCED, SURFST
OTMTCT	Other materials costs	\$	OUTPUT, PROCED, SURFST
OVRHDF	Overhead factor	--	HUB, INPUT, PROCED, SURFST, WASTOP, OUTPUT, OUT22
OWHP	Other waste-handling building electrical load	kW	INPUT, PROCED, UTILITY
OWHPOW	Other waste-handling building power requirements	kWh	UTILITY

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
OWLBR	Support labor costs	\$	HUB, OUTPUT, PROCED, SURFST, OUT22
OWOPC	Support material costs	\$	HUB, OUTPUT, PROCED, SURFST
P	Package diameter (dummy variable)	--	HOLMC, PROCED
PACKS	Number of secondary waste packages received	waste units	SPUT, TPUT
PACST	Total packaging costs by waste type and time	\$	HUB, OUTPUT, PROCED
PALBCT	Packaging labor cost	\$	HUB, PROCED, WASTOP
PAMTCT	Packaging material cost	\$	OUTPUT, PROCED, WASTOP
PANLEN	Panel length	meters	INPUT, OUTPUT, PROCED, WASTOP
PANWTH	Panel width	meters	INPUT, OUTPUT, PROCED, WASTOP
PARQCT	Annual packaging equipment cost	\$/year	HUB, OUTPUT, PROCED, WASTOP
PCTCON	Fraction secondary waste emplacement capacity available for secondary waste	--	INPUT, PROCED, SPUT, TPUT
PCTFIL	Percentage of room filled w/expanded backfill	--	INPUT, MINDEV, PROCED
PFCST	Packaging facility cost	\$/year	HUB, PROCED, OUT22
PFDECC	Decommissioning cost for waste package preparation during construction by waste type	\$	OUT22
PFPOWC	Utilities cost for waste package preparation during operations by waste type	\$	OUT22, PROCED, UTILITY
PI	Constant = 3.14157	--	MINDEV, PROCED, WASTOP
PITCH	Distance between emplacement units	meters	OUTPUT, PROCED, WASTOP
PKEQPC	Packaging equipment capacity	packages/equip. unit/ shift	HUB, PROCED, WASTOP, OUT22

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
PKGCAP	Packaging facility capacity per shift	waste units/shift- year	INPUT, PROCED, WASTOP
PKGDI A	Package diameter	meters	INPUT, PROCED, WASTOP
PKGDS1	Package narrative description	--	INPUT, PROCED
PKGDS2	Package narrative description	--	INPUT, PROCED
PKGDS3	Package narrative description	--	INPUT, PROCED
PKGLBR	Packaging labor requirement	man-years	HUB, PROCED, WASTOP, WAFAM
PKGLN	Package length	meters	INPUT, MINDEV, PROCED, WASTOP
PKGPRK	Packages per rack unit	integer	INPUT, PROCED, WASTOP
PKGUP	Packages per unit pitch	integer	INPUT, PROCED, WASTOP, OUTPUT
PKGTD	Package thermal density	kW/package	WASTOP
PKMATC	Packaging materials cost	\$	HUB, PROCED, WASTOP
PKMATR	Unit packaging materials requirement	material unit/package unit	PROCED, WASTOP
PLACE	Location of repository	--	INPUT, OUTPUT, PROCED
PMCAP	Packaging module capacity	waste units/year	OUTPUT, PROCED, WASTOP
PMMAX	Maximum packaging facility capacity	waste units/year	WASTOP
PNLREQ	Area requirement for emplacement	acres	OUTPUT, PROCED, WASTOP, MINDEV, PPUT
PORMDF	Pre-operation mining cost time distribution	fraction	INPUT, MINDEV, PROCED
POWCST	Electric power cost	\$	OUTPUT, PROCED, UTILITY
POWER	Total electric power requirement	kW/year	OUTPUT, PROCED, UTILITY

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
PRECST	Sum of other capital facility construction other startup costs	\$	OUT22
PRMMAS	Cumulative room mass mined	tonnes	HUB, OUT22, PROCED
PRODC	Pre-production costs by waste type and year	\$/year	FREOP, PROCED
PSCMAS	Cumulative capital facility construction of all corridor excavation	--	HUB, OUT22
PWF	Present worth factor	--	HUB, PROCED
R	Dummy variable	--	HUB, PROCED
RAPP	Receiving and packaging power	kW/module	INPUT, PROCED, UTILITY
RAPS	Steam requirement for receiving and packaging	10 ³ lbs/year	INPUT, PROCED, UTILITY
RATE	Discount rate (x 100.)	--	OUTPUT
RCC	Total capital costs for receiving	\$	OUT22
RCCAP	Rail container capacity	waste units	INPUT, OUTPUT, PROCED, RECEVE, WASTOP
RCEQPC	Receiving equipment capacity	waste units/shift/ module	HUB, PROCED, WASTOP
RCKCST	Rack cost for trench emplacement	\$/rack	INPUT, PROCED, WASTOP
RCKHC	Unit rock hoisting cost	\$/tonne	HUB, INPUT, PROCED
RCKHST	Rock hoisted, annual	tonnes/year	HUB, MINDEV, OUTPUT, PROCED, ROCKOP
RCKREM	Rock remaining to be backfilled	tonnes/year	ROCKOP
RCMATC	Receiving materials cost	\$	HUB, PROCED, WASTOP
RCMATR	Receiving materials requirement	material units	PROCED, WASTOP

RECON VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
RCO	Subtotal receiving costs associated with waste handling and emplacement during years of operation	\$	OUT22
RDSLC	Cumulative diesel fuel cost used in all operations	\$	OUT22, PROCED, UTILITY
REBFC	Backfilling and sealing costs for room excavation	\$	OUT22, PROCED
RECAPC	Capital facility construction for room excavation	\$	OUT22
RECLBR	Receiving labor requirement	man-years	HUB, PROCED, WASTOP, WAFAM
RECST	Total receiving costs by waste type and time	\$	HUB, OUTPUT, PROCED
RECUC	Utility costs for receiving costs associated with waste handling and emplacement during years of operation	\$	OUT22, PROCED
RECUNT	Receiving units by year and waste type	waste units	HUB, OUTPUT, PROCED, RECEVE, WAFAM, WASTOP, SPUT, TPUT
RECUWT	Total receiving units processed by waste type	waste units	PROCED, WAFAM
RED	Diesel fuel requirement for rail engines	10 ³ liters/container	INPUT, PROCED, UTILITY
REEQCT	Receiving equipment cost	\$	HUB, OUTPUT, PROCED, WASTOP, OUT22
REFBKF	Reference backfill material cost for boreholes	\$/hole	INPUT, PROCED, WASTOP
REFDEP	Reference depth for boreholes	meters	INPUT, PROCED, WASTOP
REFDIA	Reference diameter for boreholes	meters	INPUT, PROCED, WASTOP
REFMAT	Reference barrier materials costs for boreholes	\$/hole	INPUT, PROCED, WASTOP
REFSLV	Reference sleeve costs for boreholes	\$/hole	INPUT, PROCED, WASTOP
RELBCT	Receiving labor cost	\$/year	HUB, PROCED, WASTOP
REMBKF	Remaining backfill volume	m ³	PROCED, ROCKOP
REMTCT	Receiving material costs	\$/year	OUTPUT, PROCED, WASTOP

RECON VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
REPC	Utility cost for room excavation	\$	OUT22, PROCED
RESUB	Total costs for other rock excavation and handling	\$	OUT22
RFCST	Total receiving facility construction cost	\$/year	HUB, PROCED
RFDECC	Decommissioning costs associated with receiving during years of operation	\$	OUT22
RFPOWC	Cumulative cost for receiving, packaging and other waste handling power requirement	\$	OUT22, PROCED, UTILITY
RHCC	Cumulative cost for rock handling and material	\$	OUTPUT, PROCED
RHOC	Total operations cost for rock hoisting and RH shafts	\$	OUT22
RHPKG	RH packages processed by waste type	package units	PROCED, WAFAM
RHPLBR	RH packaging labor by shift and year	man-years	PROCED, WASTOP
RHRECU	RH waste units processed	waste units	WAFAM
RHRECW	RH waste units processed by waste type	waste units	PROCED, WAFAM
RHRLBR	RH receiving labor by shift and year	man-years/shift	PROCED, WASTOP
RHSC	Total capital construction costs for rock hoisting and RH shafts	\$	OUT22
RKBKF	Rock backfilled	tonnes/year	HUB, MINDEV, OUTPUT, PROCED, ROCKOP, OUT22
RKHERC	Replacement equipment costs for rock hoisting and RH shafts	\$	OUT22
RKHLBR	Rock hoisting labor	man-years	HUB, PROCED, ROCKOP, OUTPUT, OUT22, WAFAM
RKHLC	Labor costs for rock hoisting and RH shaft	\$	OUT22
RKHMAT	Rock hoisting materials	material units	HUB, OUT22, PROCED, ROCKOP

RECON VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
RKPMC	Material costs for rock hoisting and RH shaft	\$	OUT22
RKHSEC	Capital equipment cost for surface rock handling and storage	\$	OUT22
RKHSFC	Capital facility construction costs for surface rock handling and storage	\$	OUT22
RKMSM	Annual mined rock mass	tonnes/year	HUB, MINDEV, OUTPUT, PROCED, UTILITY, WAFAM
RKM1	Sum of rock mass mined	tonnes	OUT22
RKM2	Sum of rock mined for emplacement, rooms and corridor	tonnes	OUT22
RKOFFS	Rock mass transported offsite	tonnes/year	HUB, MINDEV, OUTPUT, PROCED
RKSURF	Rock mass stored on surface	tonnes/year	MINDEV, OUTPUT, PROCED, ROCKOP
RKVOL	Unit room volume excavated by waste type	m ³	MINDEV, PROCED, HUB, OUT22
RMCAP	Receiving module capacity	waste units/year	OUTPUT, PROCED, WASTOP
RMCWS1	RH waste-receiving module construction schedule - initial	--	HUB, INPUT, PROCED
RMCWS2	RH waste-receiving module construction schedule - during operations	--	HUB, INPUT, PROCED
RMFILD	Rooms filled with waste	--	HUB, MINDEV, OUTPUT, PROCED, WASTOP, OUT22, PPUT, SPUT, SSTR
RMHMIN	Minimum room height	meters	INPUT, MINDEV, PROCED
RMHT	Room height	meters	MINDEV, OUTPUT, PROCED
RMHTT	Minimum room height plus delta room height	meters	MINDEV
RMLN	Room length	meters	INPUT, MINDEV, PROCED, UTILITY, WASTOP

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
RMMATR	Materials used in rock excavation	--	HUB, MINDEV, PROCED, OUT22
RMMAX	Maximum remote-handled waste receiving facility capacity	waste units/year	WASTOP
RMACE	Capital room equipment cost for room excavation	\$	OUT22
RMNLBR	Labor required for rock excavation	man-years	HUB, MINDEV, PROCED, OUT22, WAFAM
RMNWS1	CH waste-receiving module construction schedule - Initial	--	HUB, INPUT, PROCED
RMNWS2	CH waste-receiving module construction schedule - during operations	--	HUB, INPUT, PROCED
RMPILU	Room-pillar units	Integer	OUTPUT, PROCED, WASTOP, MINDEV, PPUT
RMSPBS	Number of room-pillar units per panel per bore sections	--	PPUT, PROCED, WASTOP
RMUVOL	Room and corridor unit volume	m ³	MINDEV, PROCED
RMWTH	Room width	meters	INPUT, MINDEV, OUTPUT, PROCED, UTILITY, WASTOP
ROCAPC	Rock operations capital costs per year	\$/year	PROCED, SURFST, PREOP
ROCAST	Rock operations capital costs	\$	HUB, INPUT, PROCED, SURFST, OUTPUT, OUT22
ROECST	Rock operations unit equipment costs	\$/equipment unit	HUB, INPUT, PROCED, OUT22
ROEQPC	Rock operations equipment capacities	tonnes/equipment/ shift	INPUT, MINDEV, PROCED, ROCKOP
ROEQPL	Rock operations equipment life	tonnes/unit	INPUT, MINDEV, OUTPUT, PROCED, ROCKOP
ROLBRC	Rock operations unit labor costs	\$/man-year	HUB, INPUT, PROCED, OUTPUT, OUT22

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
ROLBRR	Rock operations unit labor requirements	man-years/equipment unit/shift	INPUT, MINDEV, PROCED, ROCKOP
ROMATC	Rock operations unit material costs	\$/material unit	HUB, INPUT, PROCED, OUTPUT, OUT22
ROMATU	Rock operations unit material requirements	material units/tonne	INPUT, MINDEV, PROCED, ROCKOP
ROOMFC	Fraction of the rock backfilled in rooms and access corridors which is backfilled into just the rooms	tonnes	OUT22
ROPOW	Rock operations power requirement	kW/year	UTILITY
RPPOW	Receiving and packaging facility power requirement	kW/year	UTILITY
RPTYP	Repository description	--	INPUT, OUTPUT, PROCED
RRCHAS	Annual costs for surface storage of mined rock	\$	HUB, OUTPUT, PROCED
RRCHST	Annual costs for hoisting of mined rock	\$	HUB, OUTPUT, PROCED
RRCMIN	Annual costs for mining of mined rock	\$	HUB, OUTPUT, PROCED
RRCOSD	Annual costs for offsite disposal of mined rock	\$	HUB, OUTPUT, PROCED
RRCTOT	Total annual rock operations costs	\$	OUTPUT, PROCED
RSTEAM	Receiving and packaging facility steam requirements	10 ³ lbs/year	UTILITY
RSTMC	Cumulative steam power cost	\$	OUT22, PROCED, UTILITY
RUBKFL	Room units backfilled	--	MINDEV, PROCED
RVP	Ventilation power	kW/acre of repository	INPUT, PROCED, UTILITY
S	Dummy variable	--	HOLMC, SURFST
SCCON1	Shaft sinking function constant 1	\$	INPUT, PROCED, SHAFT
SCCON2	Shaft sinking function constant 2	\$/meter	INPUT, PROCED, SHAFT
SCCST	Variable for future use (not currently used)	--	INPUT, PROCED

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
SCDF	Startup cost distribution factor	--	INPUT, PROCED
SCEC	Utility costs for all corridor excavation	\$	OUT22, PROCED
SCEERC	Replacement equipment costs for all corridor excavation	\$	OUT22
SCELBR	Labor costs for all corridor excavation	\$	OUT22
SCEMTC	Material costs for all corridor excavation	\$	OUT22
SCEQPC	Variable for future use (not currently used)	--	INPUT, PROCED
SCEXC	Capital construction costs for all corridor excavation	\$	OUT22
SCHSHP	Annual CH waste shipments	--	PROCED, RECEVE
SCLBRC	Variable for future use	--	INPUT, PROCED
SCLBRR	Variable for future use	--	INPUT, PROCED
SCMATC	Variable for future use	--	INPUT, PROCED
SCMATU	Variable for future use	--	INPUT, PROCED
SCMCAP	CH waste-receiving module capacity per year	waste units/year	WASTOP
SCSKD	Shaft construction schedule	--	HUB, INPUT, MINDEV, PROCED
SDSLC	Cumulative diesel fuel costs	\$	OUT22, PROCED, UTILITY
SEALCT	Shaft sealing cost by waste type and time	\$	HUB, POSTOP, PROCED
SFCAPC	Surface facility capital cost	\$/year	HUB, PROCED, SURFST, OUT22
SFCCST	Total surface facility cost allocated by waste type	\$	HUB, POSTOP, PROCED
SFCS	Surface facility construction schedule	--	INPUT, PROCED, SURFST
SFCST	Support facility capital cost	\$	INPUT, OUTPUT, PROCED, SURFST, OUT22
SFCTOT	Total shaft cost by component	\$	OUTPUT, PROCED

RECON VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
SFDECC	Decommissioning costs for support facilities	\$	OUT22
SFECST	Facility unit equipment cost	\$/facility	INPUT, PROCED, SURFST
SFEQCT	Annual surface facility equipment cost	\$/year	HUB, OUTPUT, PROCED, SURFST, OUT22
SFEQPC	Support facility equipment cost	\$/year	HUB, PROCED, SURFST
SFEQPL	Support facility equipment life	years	INPUT, OUTPUT, PROCED, SURFST
SFEXC	Capital facilities costs for waste emplacement exhaust shaft	\$	OUTPUT, OUT22, PROCED
SFLBRC	Support facility unit labor cost	\$/man-year	INPUT, PROCED, SURFST
SFLBRR	Support facility unit labor requirement	man-years/facility/ shift	INPUT, PROCED, SURFST
SFMASS	Cumulative rock excavated in shaft sinking	tonnes	HUB, OUTPUT, PROCED
SFMAST	Cumulative rock excavated in shaft sinking divided by 1000	tonnes	OUTPUT
SFMATC	Support facility unit material cost	\$/material unit	INPUT, PROCED, SURFST, OUT22
SFMATU	Support facility unit material requirements	material units/ facility	INPUT, PROCED, SURFST, OUT22
SFMMC	Capital facility construction costs for men and materials shafts	\$	OUTPUT, OUT22, PROCED
SFMXS	Capital facility construction costs for mining exhaust shaft	\$	OUTPUT, OUT22, PROCED
SFP	Support facility electrical load	kW	INPUT, PROCED, UTILITY
SFPQWC	Cumulative support facility power requirements costs	\$	OUTPUT, PROCED, UTILITY
SFPPOW	Support facility power requirements	kWh	UTILITY

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
SFRHC	Capital facility construction costs for rock hoisting and RH shaft	\$	OUTPUT, OUT22, PROCED
SFSP	Spent fuel storage electrical load	kW	INPUT, PROCED, UTILITY, OUT22
SFSPOW	Spent fuel storage power requirement	kWh	UTILITY
SFTC	Fraction of shift operated - CH waste	--	PROCED, WASTOP
SFTDD	Shaft design diameter	meters	INPUT, MINDEV, OUTPUT, POSTOP, PROCED, SHAFT
SFTHE	Hoisting equipment cost	\$	INPUT, OUTPUT, PROCED, SHAFT
SFTITL	Support facility name	--	INPUT, OUTPUT, PROCED
SFTLC	Shaft lining cost	\$/meter	OUTPUT, PROCED, SHAFT
SFTLD	Shaft lined depth	meters	INPUT, OUTPUT, PROCED, SHAFT
SFTNUM	Shaft number	--	HUB, INPUT, OUTPUT, POSTOP, PROCED, SHAFT, WASTOP, MINDEV, ROCKOP
SFTR	Fraction of shift operated - RH wastes	--	PROCED, WASTOP
SFTREQ	Preliminary shift requirements	no. of shifts	WASTOP
SFTSC	Shaft sinking cost	\$/meter	OUTPUT, PROCED, SHAFT
SFTSLC	Backfilling and sealing costs for other rock excavation and handling	\$	OUT22, POSTOP, PROCED
SFTSS	Underground shaft station construction cost	\$	INPUT, OUTPUT, PROCED, SHAFT
SFTWC	Shaft water control cost	\$	OUTPUT, PROCED, SHAFT
SFTXD	Shaft excavation depth	meters	INPUT, MINDEV, OUTPUT, PROCED, SHAFT, POSTOP
SFTXR	Shaft incremental excavation radius	meters	INPUT, MINDEV, OUTPUT, POSTOP, PROCED, SHAFT

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
SFVSC	Capital facility construction costs for supply shaft	\$	OUTPUT, OUT22, PROCED
SFWSC	Capital facility construction costs for waste shaft	\$	OUTPUT, OUT22, PROCED
SFWSE	Capital equipment costs for waste shaft	\$	OUTPUT, OUT22, PROCED
SHAFTX	Rock excavated in shaft sinking	tonnes	MINDEV, PROCED
SHCFAC	Shift cost factor	1 + fraction	INPUT, PROCED, WASTOP
SHCPY	Shaft cost per year	\$/year	HUB, PREOP, PROCED
SHDFC	Total cost per shaft	\$/shaft	HUB, OUTPUT, PROCED, SHAFT
SHRKMD	Rock excavated in shaft sinking allocated over construction years	tonnes	HUB, MINDEV, OUTPUT, PROCED
SITSUB	Subtotal facility construction site costs	\$	OUT22
SIZE	Repository size	acres	INPUT, OUTPUT, PROCED, UTILITY, WASTOP
SKED	Surface facility construction schedule (dummy variable)	--	HUB
SKFUNC	Shaft sinking cost function	\$/ft of depth	POSTOP, PREOP, SHAFT, SKFUNC
SLBR	Number of shifts per day worked by support personnel	--	INPUT, PROCED, SURFST
SLCU	Unit repository costs	\$/kgHM, \$/m ³ , \$/package	HUB, PROCED
SLEETH	Sleeve thickness	meters	INPUT, PROCED, WASTOP
SLEVCS	Sleeve cost	\$	HOLMC
SLEVOR	Sleeve outside radius	meters	PROCED, WASTOP
SMMSC	Capital facility construction costs for men and materials shafts	\$	OUT22
SMNDWT	Total mined rock mass by waste type	tonnes	PROCED, WAFAM

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
SOA	Surface operations area	acres	INPUT, OUTPUT, PROCED, SURFST
SOTBKF	Annual supplemental backfill required	tonnes	MINDEV, OUTPUT, PROCED, ROCKUP
SPBFC	Backfilling and sealing costs for shaft pillar excavation	\$	OUT22
SPEC	Total capital construction costs for shaft pillar excavation	\$	OUT22
SPECAC	Capital facility construction for shaft pillar excavation	\$	OUT22
SPMASS	Cumulative central pillar corridor rock mass mined	tonnes	HUB, OUT22, PROCED
SPMCAP	Shifts required to package RH waste	shifts	WASTOP
SPMEQC	Capital equipment for shaft pillar excavation	\$	OUT22
SPUC	Site preparations cost	\$ thousands/acre	INPUT, OUTPUT, PROCED, SURFST
SRHC	Total capital construction costs for surface rock handling and storage	\$	OUT22
SRHERC	Replacement equipment costs for surface rock handling and storage	\$	OUT22
SRHLBR	Storage and rock handling labor	man-years/year	HUB, PROCED, ROCKOP, OUT22
SRHLC	Labor costs for surface rock handling and storage	\$	OUT22
SRHMAT	Storage and rock handling materials	material units/year	HUB, PROCED, ROCKOP
SRHMC	Material costs for surface rock handling and storage	\$	OUT22
SRHOC	Total operations cost for surface rock handling and storage	\$	OUT22
SRHPKG	Total RH packages	package units	WAFAM
SRHSHP	Annual RH waste shipments	shipments/year	PROCED, RECEVE

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
SRKBKF	Annual rock mass backfilled	tonnes/year	MINDEV, OUTPUT, PROCED, ROCKOP
SRKHST	Annual rock mass hoisted	tonnes/year	MINDEV, OUTPUT, PROCED, ROCKOP, SHAFT
SRKMND	Annual rock mass mined	tonnes/year	MINDEV, OUTPUT, PROCED, WAFAM, HUB, OUT22
SRKVOL	Cumulative unit room excavated	m ³	OUT22, PROCED
SRMCAP	RH waste-receiving module capacity	waste units/year	WASTOP
SRMF	Total rooms filled with waste in last operation year	rooms	WASTOP
SRMFLD	Annual rooms filled	rooms	PROCED, WASTOP
SRMM	Annual rock mass mined for all waste types	tonnes	MINDEV, PROCED, ROCKOP
SRMUVL	Cumulative room and corridor unit volume	m ³	OUT22, PROCED
SRMWT	Total rooms filled with waste by waste type	rooms	PROCED, WASTOP
SRRMC	Shift requirements for CH receiving module	shifts	OUTPUT, PROCED, SURFST, UTILITY, WASTOP
SRRMP	Shift requirements for packaging module	shift	WASTOP
SRRMR	Shift requirements for RH receiving module	shifts	WASTOP, OUTPUT, PROCED, ROCKOP, SURFST, UTILITY, WASTOP, SHAFT
SRSFTC	Shift requirements for CH waste shaft	shifts	WASTOP
SRSFTR	Shift requirements for RH waste shaft	shifts	WASTOP
SS	Surface storage mined rock capacity	tonnes	MINDEV
SSCON1	Shaft sealing function constant 1	--	INPUT, POSTOP, PROCED
SSCON2	Shaft sealing function constant 2	--	INPUT, POSTOP, PROCED
SSDF	Shaft sealing cost distribution over time	\$	INPUT, POSTOP, PROCED

RECON VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
SSFCST	Total support facility capital costs	\$	OUTPUT
STARTC	Startup costs	\$/year	HUB, PREOP, PROCED, OUT22
STEAM	Annual steam requirement allocated by waste type	10 ³ lbs	OUTPUT, PROCED, UTILITY
STMC	Unit steam cost	10 ³ lbs/year	INPUT, PROCED, UTILITY
STMCST	Annual steam cost	\$/year	OUTPUT, PROCED, UTILITY
STOMAX	Storage capacity limit for secondary waste for coemplacement per primary waste package	--	INPUT, PROCED, SSTOR
STOR	Storage capacity for coemplacement	--	PROCED, SPUT, SSTOR, TPUT
STPCT	Percentage of field costs used for startup costs	--	PROCED, SPUT, SSTOR, TPUT
SUD	Diesel fuel requirement - surface use	10 ³ liters/day	INPUT, PROCED, UTILITY
SUMBKF	Sum of rock mass backfilled	tonnes	HUB, OUTPUT, PROCED
SUMCPK	Annual waste packages processed for all waste types	waste packages	PROCED, WASTOP
SUMCRC	Total annual waste units received	--	PROCED, WASTOP
SUMEPU	Sum of number of like emplacement units	--	PROCED, WASTOP
SUMET	Sum of number of waste packages	--	PROCED, WAFAM
SUMIND	Total rock mass mined	tonnes	PROCED, WAFAM
SUMLBR	Sum of labor requirements	man-year	WAFAM
SUMMET	Total packages emplaced by emplacement method and waste type	--	PROCED, WAFAM
SUMREC	Total waste units processed	waste units	WAFAM
SUMRPK	Annual RH waste packages processed	package units	PROCED, WASTOP
SUMRRC	Annual RH receiving units processed	waste units	PROCED, WASTOP

RECON VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
SUMRSH	Total RH waste shipments	shipments	WAFAM
SUMSHC	Total waste shipments	shipments	WAFAM
SUMUTL	Total utility costs	\$	WAFAM
SUPC	Annual CH waste packages processed	package units	WASTOP
SUPCC	Support costs for capital construction	\$	OUT22
SUPEQC	Support costs for replacement equipment	\$	OUT22
SUPFCC	Support costs for capital facility construction	\$	OUT22
SUPFEC	Support costs for capital equipment	\$	OUT22
SUPLBC	Support costs for labor	\$	OUT22
SUPMTC	Support costs material	\$	OUT22
SUPR	Yearly RH waste packages processed	package units	WASTOP
SUPSUB	Total support costs for operations	\$	OUT22
SUPUTC	Support costs for utilities	\$	OUT22
SURC	Yearly CH receiving units processed	waste units	WASTOP
SURCST	Annual surveillance cost	\$/year	INPUT, POSTOP, PROCED
SURF	Rock mass stored on surface (dummy variable)	tonnes	MINDEV
SURFDC	Unit surface handling and storage cost	\$/tonne	HUB, INPUT, PROCED
SURR	Yearly RH receiving units processed	waste units	WASTOP
SURVC	Total discounted surveillance cost	\$	POSTOP
SURVCT	Surveillance cost by year and waste type	\$	HUB, POSTOP, PROCED
SVSS	Total costs for supply shaft during operations	\$	OUT22

RECON VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
SWPPR	Secondary waste package per room solo emplacement	--	INPUT, PROCED, SSTOR
T	Dummy variable	--	HOLMC
TA	Temporary storage variable for printing various output variables	--	OUTPUT
TABLES	Array for defining which tables to output	--	PROCED, HUB, INPUT, OUTPUT
TACCST	Total annual repository cost	\$	OUTPUT, PROCED
TAECST	Total architect-engineer costs allocated by waste type	\$	HUB, OUTPUT, PROCED
TALBCT	Total labor cost	\$	HUB, OUTPUT, PROCED
TANF	Near-field total area	acres	PROCED, WASTOP
TB	Temporary storage variable for printing various output variables	--	OUTPUT
TBDCST	Backfilling and decommission cost category subtotal by waste type	--	HUB, OUTPUT, PROCED
TBFCST	Total backfilling cost allocated by waste type	\$	HUB, OUTPUT, PROCED
TBKCT	Total backfilling cost	\$	HUB
TC	Temporary storage variable for printing various output variables	--	OUTPUT
TCCAP	Truck container capacity	waste units/cask	INPUT, OUTPUT, PROCED, RECEVE, WASTOP
TCCHRM	Total cost of CH receiving modules	\$	OUTPUT, PROCED, RECEVE, OUT22
TCLAND	Land purchase cost	\$	INPUT, PREOP, PROCED
TCLICE	Licensing cost	\$	INPUT, PREOP, PROCED
TCNCST	Total contingency costs by waste type	\$	HUB, OUTPUT, PROCED
TCONT	Total contingency costs	\$	OUT22

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
TCPCLN	Total central pillar corridor length	meters	INPUT, MINDEV, PROCED, ROCKOP, UTILITY
TCRHPM	Total cost of RH packaging modules	\$	OUTPUT, PROCED, RECEVE
TCRHRM	Total cost of RH receiving modules	\$	OUTPUT, PROCED, RECEVE, OUT22
TD	Temporary storage variable for printing various output variables	--	OUTPUT
TDCST	Total decommissioning costs allocated by waste type	\$	HUB, OUTPUT, PROCED
TDWI	Heat density per waste unit	kW/waste unit	INPUT, OUTPUT, PROCED, WASTOP
TE	Temporary storage variable for printing various output variables	--	OUTPUT
TECAPC	Total capital costs associated with emplacement excavation	\$	HUB, OUTPUT, PROCED
TEEC	Total costs for waste emplacement excavation	\$	OUT22
TEEEC	Total emplacement excavation equipment costs by emplacement method	\$	HUB, OUTPUT, PROCED
TEKNEK	Mining method	--	INPUT, OUTPUT, PROCED
TEMCST	Total emplacement cost allocated by waste type	\$	HUB, OUTPUT, PROCED
TEQPT	Total emplacement excavation equipment cost	\$	HUB, OUTPUT, PROCED
TEWC	Total costs for waste emplacement	\$	OUT22
TEXCST	Total emplacement excavation cost by waste type	\$	HUB, OUTPUT, PROCED
TEXLBR	Total emplacement excavation labor	man-year	WASTOP
TF	Temporary storage variable for printing various output variables	--	OUTPUT
TG	Temporary storage variable for printing various output variables	--	OUTPUT

RECON VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
TH	Temporary storage variable for printing various output variables	--	OUTPUT
THDMAS	Total annual rock mass mined in thousand tonnes/year	tonnes	OUTPUT
THEET	Total repository heat load	kilowatts	WASTOP
THSCST	Total rock hoisting cost allocated by waste type	\$	HUB, OUTPUT, PROCED
TI	Today's time	--	HUB, OUTPUT, PROCED
TITLE	Title	--	INPUT, OUTPUT, PROCED
TLBRWT	Total labor by waste type	--	INPUT, OUTPUT, PROCED
TLCCST	Total licensing cost allocated by waste type	\$	HUB, OUTPUT, PROCED, OUT22
TLDCST	Total land cost allocated by waste type	\$	HUB, OUTPUT, PROCED, OUT22
TMATCT	Annual materials costs for waste operations	\$/year	OUTPUT, PROCED, WASTOP, SURFST
TMINCT	Total rock excavation costs	\$	HUB
TMMSC	Total cost for men and materials shafts	\$	OUT22
TNFTA	Total near-field thermal area	acres	WASTOP
TNPT1	Rock hoisting capacity for rock hoisting shaft	tonnes/trip	INPUT, OUTPUT, PROCED, ROCKOP, SHAFT
TNPT2	Rock hoisting capacity for men and materials shaft	tonnes/trip	INPUT, OUTPUT, PROCED, ROCKOP, SHAFT
TOCST	Transportation capital costs	\$	INPUT, PROCED, SURFST
TOECST	Transportation operations unit equipment cost	\$/equipment unit	INPUT, PROCED, WASTOP
TOEQPC	Transportation operations equipment capacity	packages/equipment unit/shift	INPUT, PROCED, WASTOP
TOEQPL	Transportation operations equipment life	packages	INPUT, OUTPUT, PROCED, WASTOP

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
TOLBRC	Transportation operations unit labor cost	\$/man-year	INPUT, PROCED, WASTOP
TOLBRR	Transportation operations unit labor requirement	man-years/equipment unit	INPUT, PROCED, WASTOP
TOMATC	Transportation operations unit material cost	\$/material unit	INPUT, PROCED, WASTOP
TOMATU	Transportation operations unit material requirements	material unit/package	INPUT, PROCED, WASTOP
TOMCST	Total rock excavation cost allocated by waste type	\$	HUB, OUTPUT, PROCED
TONPAR	Surface rock storage density	tonnes/acre	INPUT, MINDEV, PROCED
TONPT	Dummy variable for rock hoisting shaft capacity	--	OUTPUT
TOPCST	Total waste operations cost allocated by waste type	\$	HUB, OUTPUT, PROCED
TORMAS	Total rock mass mined for rooms and corridors	tonnes	OUT22
TOT	Output character constant	--	OUTPUT
TOTFLD	Total field cost	\$	PREOP
TOTSFC	Total facility capital cost	\$	OUTPUT, PROCED
TP	Total rock mass backfilled	tonnes	REPLAS, ROCKOP
TPKCST	Total packaging operations costs allocated by waste type	\$	HUB, OUTPUT, PROCED
TPRMAS	Total rock mass mined	tonnes	OUT22
TPYMM	Annual rock hoisting capacity for men and materials shaft	tonnes	SHAFFT
TPYMP	Annual rock hoisting capacity for rock hoisting shaft	tonnes	SHAFT
TRBKF	Total material backfilled	tonnes	HUB
TRCAPC	Transportation capital cost	--	HUB, PROCED, SURFST

RECON VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
TRCC	Total rock excavation and handling costs for rock operation	\$	OUTPUT, PROCED
TRCCST	Total receiving operations costs allocated by waste type	\$	HUB, OUTPUT, PROCED
TRCHWD	Trench width	meters	INPUT, PROCED, WASTOP
TRCLBR	Transportation labor requirement	man-year	PROCED, WASTOP
TRCST	Annual transport costs allocated by waste type	\$	HUB, OUTPUT, PROCED
TREC	Replacement equipment costs by rock operation	\$	HUB, OUTPUT, PROCED
TREFC	Total capital facility construction for receiving	\$	OUT22
TREM	Total rock mass remaining to be backfilled	tonnes	ROCKOP
TREQCT	Annual transportation equipment cost	\$/year	HUB, OUTPUT, PROCED, WASTOP, OUT22
TREQPC	Transportation equipment capacity (dummy)	packages/year	HUB, PROCED, WASTOP
TRFILD	Total rooms filled with waste	--	MINDEV
TRHCT	Total rock hoisting cost	\$	HUB
TRHSC	Total rock hoisting and RH shafts costs	\$	OUT22
TRHSTD	Total rock hoisted	tonnes	HUB
TRKMND	Total rock mined	tonnes	HUB
TRLBCT	Transportation labor cost	\$	HUB, PROCED, WASTOP
TRM	Variable for storing total rock mined over various time periods	tonnes	MINDEV
TRMATC	Transportation materials cost	\$	HUB, PROCED, WASTOP
TRMATR	Transportation materials requirement	material units	PROCED, WASTOP
TRMTCT	Annual transportation materials cost	\$/year	OUTPUT, PROCED, WASTOP

RECON VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
TRNLBR	Transport labor required by waste type and time	man-year	HUB, PROCED, WASTOP, WAFAM
TRNSPT	Number of transporters	--	HUB, OUTPUT, PROCED, WASTOP
TRRLBR	RH waste transport labor by waste type and time	man-year	PROCED, WASTOP
TRSHCT	Total rock surface handling cost	\$	HUB
TRSHF	Total remote-handled waste shipments received	--	PROCED, RECEVE
TSCEC	Total corridor excavation costs	\$	OUT22
TSFCST	Total surface facility capital costs	\$	HUB, OUTPUT, PROCED
TSFEC	Support facility equipment replacement cost by facility type	\$	OUTPUT, PROCED, SURFST
TSFMAS	Total rock excavated in shaft sinking	tonnes	OUTPUT
TSH	Total waste shipments received	--	OUTPUT
TSHCST	Total shaft construction cost by waste type	\$	HUB, OUTPUT, PROCED
TSHPCN	Total cask shipments	shipments	OUTPUT, PROCED, RECEVE, UTILITY, WAFAM
TSLOCST	Total shaft sealing cost allocated by waste type	\$	HUB, OUTPUT, PROCED
TSPCST	Total site preparation cost allocated by waste type	\$	HUB, OUTPUT, PROCED, OUT22
TSPEC	Total costs for shaft and pillar excavation	\$	OUT22
TSRHC	Total costs for rock hoisting and RH shafts	\$	OUT22
TSRRMR	Shift requirement for RH receiving module	shifts	SHAFT
TSSCST	Total surface handling cost allocated by waste type	\$	HUB, OUTPUT, PROCED
TSUBCN	Total facility construction cost allocated by waste type	\$	HUB, OUTPUT, PROCED
TSUBRO	Total rock operation cost allocated by waste type	\$	HUB, OUTPUT, PROCED

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
TSUCST	Total startup cost allocated by waste type	\$	HUB, OUTPUT, PROCED
TSUPP	Total support costs	\$	OUT22
TSURVC	Annual surveillance cost	\$	OUTPUT, POSTOP, PROCED
TSVCST	Total surveillance cost allocated by waste type	\$	HUB, OUTPUT, PROCED
TTANF	Limiting emplacement area per horizontal borehole defined by the near field thermal limit	--	WASTOP
TTCAPC	Total transport capital cost	\$	HUB, OUTPUT, PROCED, OUT22
TTEC	Transport equipment cost by waste handling category	\$	HUB, OUTPUT, PROCED
TTI	Temporary storage variable for printing various cost Totals	--	OUTPUT
TTRCC	Total receiving costs	\$	OUT22
TTRCST	Total transport cost allocated by waste type	\$	HUB, OUTPUT, PROCED
TTREC	Total receiving costs	\$	OUT22
TTRHPM	Total facility construction capital waste package preparation costs by waste type	\$	OUT22
TTSH	Total waste shipments by waste type	--	OUTPUT
TTTK	Average receiving facility turnaround time per truck cask	days	INPUT, PROCED, WASTOP
TTTR	Average receiving facility turnaround time per rail cask	days	INPUT, PROCED, WASTOP
TUCAPC	Total underground facility capital cost	\$	HUB, OUTPUT, PROCED, OUT22
TUEC	Emplacement equipment cost by waste handling category	\$	HUB, OUTPUT, PROCED
TUFCST	Total underground facility construction cost by waste type	\$	HUB, OUTPUT, PROCED
TUTCST	Total support operations costs allocated by waste type	\$	HUB, OUTPUT, PROCED

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
TVEX	Total waste emplacement exhaust shaft and costs	\$	OUT22
TVMX	Total mining exhaust shaft costs	\$	OUT22
TVSS	Total supply shaft costs	\$	OUT22
TWEC	Receiving facility equipment cost by waste handling category	\$	HUB, OUTPUT, PROCED
TWOEQC	Annual waste operations equipment cost	\$	HUB, OUTPUT, PROCED
TWOPCT	Annual support operations costs allocated by waste type	\$	HUB, OUTPUT, PROCED
TWPC	Total costs for waste package preparations by waste type	\$	OUT22
TWSC	Total waste shaft costs	\$	OUT22
TWTC	Total waste transfer costs	\$	OUT22
TWTCST	Total repository cost allocated by waste type	\$	HUB, OUTPUT, PROCED
UAECST	Unit cost of A-E services	\$/unit	HUB, OUTPUT, PROCED
UBDCST	Unit cost for all backfilling and decommissioning operations	\$/unit	HUB, OUTPUT, PROCED
UBFCST	Unit cost of backfilling	\$/unit	HUB, OUTPUT, PROCED
UBFRMS	Unbackfilled rooms	--	MINDEV, PROCED
UBFRPY	Unbackfilled rooms per year	--	MINDEV, PROCED, UTILITY
UBK VOL	Unbackfilled volume	m ³	MINDEV, PROCED, ROCKOP
UCNCST	Unit cost of contingencies	\$/unit	HUB, OUTPUT, PROCED
UDCCST	Unit cost of decommissioning	\$/unit	HUB, OUTPUT, PROCED
UDSLC	Total utility costs for waste transfer	\$	OUT22, PROCED, UTILITY
UEMCST	Unit cost of emplacement operations	\$/unit	HUB, OUTPUT, PROCED

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
UEXCST	Unit cost of emplacement excavation operations	\$/unit	HUB, OUTPUT, PROCED
UFCAPC	Underground facility capital costs	\$/year	HUB, PROCED, SURFST
UGD	Diesel fuel requirement - underground	10 ³ liters/package	INPUT, PROCED, UTILITY
UGFC	Total capital costs of underground facilities	\$	HUB, PREOP, PROCED
UHSCST	Unit cost of rock hoisting operations	\$/unit	HUB, OUTPUT, PROCED
ULCCST	Unit cost of repository licensing	\$/unit	HUB, OUTPUT, PROCED
ULDCST	Unit cost of land acquisition	\$/unit	HUB, OUTPUT, PROCED
UNTMAS	Mass per waste unit	kilograms	HUB, INPUT, PROCED
UOCST	Emplacement operations capital cost	\$ millions	INPUT, PROCED, SURFST
UODCST	Unit cost of offsite rock disposal	\$/unit	HUB, OUTPUT, PROCED
UOECST	Emplacement operations unit equipment cost	\$/equipment unit	INPUT, PROCED, WASTOP
UOEQPC	Emplacement operations equipment capacity	emplacements/equipment/shift	INPUT, PROCED, WASTOP
UOEQPL	Emplacement operations equipment life	emplacements	INPUT, OUTPUT, PROCED, WASTOP
UOLBRC	Emplacement operations unit labor cost	\$/man-year	INPUT, PROCED, WASTOP
UOLBRR	Emplacement operations unit labor requirement	man-years/equipment unit	INPUT, PROCED, WASTOP
UOMATC	Emplacement operations unit material cost	\$/material unit	INPUT, PROCED, WASTOP
UOMATU	Emplacement operations material unit requirement	material unit/ emplacement unit	INPUT, PROCED, WASTOP
UOMCST	Unit cost of mining operations	\$/unit	HUB, OUTPUT, PROCED
UOPCST	Unit cost for all waste operations	\$/unit	HUB, OUTPUT, PROCED
UPC	Power cost	cents/kWh	INPUT, PROCED, UTILITY

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
UPKCST	Unit cost of waste packaging	\$/unit	HUB, OUTPUT, PROCED
URCCST	Unit cost of waste receiving	\$/unit	HUB, OUTPUT, PROCED
USDLIF	Used equipment life	life units	REPLAS
USFCST	Unit cost of support facilities	\$/unit	HUB, OUTPUT, PROCED
USHCST	Unit cost of shaft construction	\$/unit	HUB, OUTPUT, PROCED
USLCST	Unit cost of shaft sealing	\$/unit	HUB, OUTPUT, PROCED
USPCST	Unit cost of site preparation	\$/unit	HUB, OUTPUT, PROCED
USSCST	Unit cost of surface rock handling and storage	\$/unit	HUB, OUTPUT, PROCED
USUBCN	Unit cost of all capital construction	\$/unit	HUB, OUTPUT, PROCED
USUBRO	Unit cost of all rock operations	\$/unit	HUB, OUTPUT, PROCED
USUCST	Unit cost of startup operations	\$/unit	HUB, OUTPUT, PROCED
USVCST	Unit cost of post-operational surveillance	\$/unit	HUB, OUTPUT, PROCED
UTRCST	Unit cost of transport operations in repository	\$/unit	HUB, OUTPUT, PROCED
UUFCST	Unit cost of underground facility construction	\$/unit	HUB, OUTPUT, PROCED
UUTCST	Unit cost of repository utilities	\$/unit	HUB, OUTPUT, PROCED
UWTCST	Total unit cost for repository	\$/unit	HUB, OUTPUT, PROCED
VARMC	Variable mining cost	\$/tonne	HUB, INPUT, PROCED, PREOP
VENCHT	Ventilation corridor height	meters	PROCED
VENCLN	Ventilation corridor length	meters	PROCED
VENCWT	Ventilation corridor width	meters	PROCED
VENTPC	Total power requirement for ventilation	kW	OUT22, PROCED, UTILITY

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
VENTUC	Total utility costs for supply shaft	\$	OUT22
VEXS	Total capital costs for waste emplacement exhaust shaft	\$	OUT22
VMXS	Total capital costs for mining exhaust shaft	\$	OUT22
VNTPOW	Power requirement for ventilation	kW	UTILITY
VOLDW	Demineralized water volume	10 ³ liters/day	INPUT, PROCED, UTILITY
VOLMND	Rock volume mined	m ³	MINDEV, PROCED
VOLW	Water volume	10 ³ liters/day	INPUT, PROCED, UTILITY
VSSC	Total capital costs for supply shaft	\$	OUT22
VSUB	Total ventilation costs	\$	OUT22
WABATH	Water barrier thickness	meters	INPUT, PROCED, WASTOP
WABOR	Diameter of hole plus water barrier	meters	PROCED, WASTOP
WAF	Dummy waste allocation factor	--	HUB, POSTOP, SURFST
WAFAL	Waste allocation factor based on all wastes	--	PROCED, UTILITY, WAFAM
WAFCN	Waste allocation factor based on canistered waste	--	HUB, PROCED, SURFST, UTILITY, WAFAM
WAFLBR	Labor factor by waste type	--	POSTOP, PROCED, SURFST, UTILITY, WAFAM
WAFMET	Allocation factor based on packages used in each emplacement method	--	PROCED, SURFST, WAFAM
WAFMV	Waste allocation factor based on mined volume	--	HUB, MINDEV, POSTOP, PREOP, ROCKOP, SURFST, WAFAM, OUT22, PROCED
WAFMYY	Waste allocation factor based on mined volume by year	--	PROCED, UTILITY, WAFAM

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
WAFNC	Waste allocation factor based on non-canistered waste	--	HUB, POSTOP, PROCED, SURFST, WAFAM
WAFRHP	Waste allocation factor based on RH packages	--	HUB, POSTOP, PROCED, WAFAM, RECEVE
WAFRSH	Waste allocation factor based on RH waste shipments	--	PROCED, WAFAM
WAFSHC	Waste allocation factor based on all waste shipments	--	PROCED, WAFAM
WAFUGW	Waste allocation factor based on all waste packages processed	--	HUB, POSTOP, PROCED, WAFAM
WAFUTL	Waste allocation factor based on utility costs	--	PROCED, WAFAM
WATC	Total water cost	\$	OUT22
WATER	Repository water usage requirement	10 ⁶ liters	OUTPUT, PROCED, UTILITY
WATERC	Total water cost	\$	OUTPUT, PROCED, UTILITY, OUT22
WBCSTF	Incremental water barrier cost factor	--	HOLMC
WCCON1	Shaft sinking water control cost constant 1	--	INPUT, PROCED, SHAFT
WCCON2	Shaft sinking water control cost constant 2	--	INPUT, PROCED, SHAFT
WEC	Total capital costs for waste emplacement	\$	OUT22
WEEPC	Total utility costs for waste emplacement excavation	\$	OUT22, PROCED
WEO	Total operations costs for waste emplacement	\$	OUT22
WHSUB	Total costs for waste handling and emplacement	\$	OUT22
WOCST	Waste operations capital cost	\$	HUB, INPUT, PROCED, RECEVE
WOECST	Waste operations unit equipment cost	\$/equipment unit	INPUT, PROCED, WASTOP
WOEQPC	Waste operations unit equipment capacity	waste unit/equipment unit/shift	INPUT, PROCED, WASTOP

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
WOEQPL	Waste operations unit equipment life	waste units	INPUT, OUTPUT, PROCED, WASTOP
WOLABR	Waste operations labor requirement	man-years	PROCED, WASTOP
WOLBRC	Waste operations unit labor cost	\$/man-year	INPUT, PROCED, WASTOP
WOLBRR	Waste operations unit labor requirement	man-years/shift/ module	INPUT, PROCED, WASTOP
WQMATC	Waste operations unit materials cost	\$/material unit	INPUT, PROCED, WASTOP
WOMATU	Waste operations unit materials requirement	material unit/module	INPUT, PROCED, WASTOP
WOPLBR	Labor requirement for all repository waste operations	man-years	HUB, OUTPUT, PROCED
WPC	Subtotal capital construction waste package preparation costs by waste type	\$	OUT22
WPO	Subtotal costs for waste package preparation during operations by waste type	\$	OUT22
WPSUB	Subtotal of all costs associated with waste package preparation	\$	OUT22
WRBKFL	Total rock backfilled into waste rooms	tonnes	MINDEV, OUT22, PROCED
WRKYR	Days in work year - constant (250)	--	INPUT, MINDEV, PROCED, UTILITY, WASTOP
WSC	Total capital costs for waste shaft	\$	OUT22
WSFTSC	Total backfill and sealing costs for waste shaft	\$	OUT22
WSTSUP	Annual waste units requiring disposal	waste units	INPUT, PROCED, WASTOP
WSTVOL	Waste volume	m ³	HUB, PROCED, WASTOP
WSTYP	Waste type descriptions	--	INPUT, OUTPUT, PROCED, OUT22
WTC	Total capital costs for waste transfer	\$	OUT22
WTHEET	Total repository heat by waste type	kilowatts	PROCED, WASTOP

RECON VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
WTO	Total operations costs for waste transfer	\$	OUT22
WTRC	Unit water cost	cents/10 ⁶ liters	INPUT, PROCED, UTILITY
WUPP	Maximum waste units per package	--	HUB, INPUT, PROCED, WASTOP, SPUT, TPUT
WVOLPU	Waste volume per waste unit	m ³	INPUT, PROCED, WASTOP
X	Dummy variable	--	HOLMC
XPANFC	Rock expansion factor after mining	fraction	INPUT, MINDEV, PROCED, ROCKOP
XRASHO	Extraction ratio (room/room + pillar)	--	INPUT, OUTPUT, PROCED, WASTOP
ZER01	Variable used to zero out common blocks	--	HUB, PROCED
ZER02	Variable used to zero out common blocks	--	HUB, PROCED
ZER03	Variable used to zero out common blocks	--	HUB, PROCED
ZER04	Variable used to zero out common blocks	--	HUB, PROCED
ZER05	Variable used to zero out common blocks	--	HUB, PROCED
ZER06	Variable used to zero out common blocks	--	HUB, PROCED
ZER07	Variable used to zero out common blocks	--	HUB, PROCED
ZER08	Variable used to zero out common blocks	--	HUB, PROCED
ZER09	Variable used to zero out common blocks	--	HUB, PROCED
ZER010	Variable used to zero out common blocks	--	HUB, PROCED
ZER011	Variable used to zero out common blocks	--	HUB, PROCED
ZER012	Variable used to zero out common blocks	--	HUB, PROCED
ZER013	Variable used to zero out common blocks	--	HUB, PROCED

PREPROCESSOR VARIABLE LISTING

Variable Name	Description	Units	Location
A	Dummy variable	--	DECODE1, DECODER
ACP	Air conditioning building load	kWh	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
AECON1	Constant used to estimate A-E costs	--	OLDCASE, OTPREP, PROCED, RDPREP, WRITE
AECON2	Constant used to estimate A-E costs	--	OLDCASE, OTPREP, PROCED, RDPREP, WRITE, OTRUCK
B	Dummy variable	--	DECODER
BCKFTH	Backfill material thickness	meters	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
BKFLC	Backfilling cost	$\text{\$/m}^3$ ($\text{\$/MT}$)	OLDCASE, OTRUCK, PROCED, RDROCK, WRITE
BKRATE	Backfilling rate	tonnes/year	OLDCASE, OTRUCK, PROCED, RDROCK, WRITE
BLANK	Character constant	--	DECODE1, DECODER, OTDECM, OTGEN, OTPREP, OTRUCK, OTRPOS, OTSHFT, OTSUPT, OTUTLT, OTWAST, PROCED, RDDECM, RDGEN, RDPREP, RDROCK, RDRPOS, RDSHFT, RDSUPT, RDUTLT, RDWAST, DECODE1, DECODER
C	Dummy variable	--	DECODE1, DECODER
CAPLEN	Cap length	meters	OLDCASE, OTWASTE, PROCED, RDWAST, WRITE
CCCON1	Lining cost function constant 1	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
CCCON2	Lining cost function constant 2	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
CHMTEM	CH waste material costs	\$/unit pitch	OLDCASE, OTWAST, PROCED, RDWAST, WRITE, CHRECW, PROCED
CLCDF	Time distribution factors for licensing costs	--	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
CLNDF	Time distribution factors for land cost	--	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
CLSEMP	Closest emplacement to corridor	meters	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
COCON	Number of non-heat generating waste packages that can be coemplaced per primary waste package	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
CONFAC	Contingency factor	--	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
CONTRU	Number of CH packages that can be emplaced per TRU waste packages for each primary waste type	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
CORDSC	Corridor descriptions	--	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
CORHT	Corridor height	meters	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
CORLEN	Corridor length per room	meters	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
CORWTH	Corridor width	meters	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
CPA	Central pillar area	acres	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
CRRAIL	Percent of waste units shipped by rail	fraction	OLDCASE, OTDECM, OTWAST, PROCED, RDDECM, RDWAST, WRITE
D	Dummy variable	--	DECODER
DATYPE	Subroutine data name	--	OLDCASE, PROCED, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
DBDF	Decommissioning backfill distribution factor	--	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
DCOMCF	Facility decommissioning cost factor	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
DDCST	Variable defined for future usage (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
DDEQPC	Variable defined for future usage (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
DDFRAC	Variable defined for future usage (not currently used)	--	OLDCASE, PROCED, WRITE
DDLBR	Variable defined for future usage (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
DDLBR	Variable defined for future usage (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
DDMATC	Variable defined for future usage (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
DDTITL	Variable defined for future usage (not currently used)	--	OLDCASE, PROCED, WRITE
DELRHT	Delta room height	meters	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
DENSTY	Rock density	tonnes/m ³	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
DEPTH	Mine depth	meters	OLDCASE, OTGEN, OTSHFT, OTWAST, PROCED, RDGEN, WRITE
DRATE	Discount rate	--	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
DSLC	Diesel fuel cost	\$/liters	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
DWTRC	Unit demineralized water cost	\$/10 ⁶ liters	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
EECST	Capital cost of emplacement excavation	\$	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
EEECST	Unit equipment cost for emplacement	\$/equip. unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
EEEQPC	Emplacement excavation equipment capacity	empl. unit/ equip./shift	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
EEEQPL	Emplacement excavation equipment life	empl. units/ equip. unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
EELBRC	Unit emplacement excavation labor cost	\$/man-year	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
EELBRR	Unit emplacement excavation labor requirement	man-year/ equip./shift	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
EEMATC	Unit emplacement excavation materials cost	\$/material unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
EEMATU	Unit emplacement excavation materials by year and waste type	material units	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
FDDF	Facility decommissioning distribution factor	--	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
FFTL	Far-field thermal limit	kW/acre	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
GXEF	Geometric exclusion factor	acres	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
HDCON1	Hole drilling constant	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
HDCON2	Hole drilling constant	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
HDECFC	Hole drilling equipment unit cost fraction	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
HDLFC	Hole drilling labor unit cost fraction	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
HDMCFC	Hole drilling materials unit cost fraction	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
HEATS	Unit steam heating requirement - underground	lbs/acre of repository	OLDCASE, OTUTLT, RDUTLT, WRITE
HOLDEL	Hole depth delta	meters	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
HPMIN	Minimum hole pitch	meters	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
I	Index	--	MAIN, OLDCASE, OTDECM, OTGEN, OTWAST, RDDECM, RDGEN, RDROCK, RDRPOS, RDSUPT, RDWAST, WRITE
IA	Dummy variable	--	DECODEI
IANS	Input answer variable	--	MAIN, TABLE
IC	Output device designator	--	WRITE, TABLE
ICARD	Character variable used for reading in data	--	PROCED, RDDECM, RDGEN, RDPREP, RDROCK, RDRPOS, RDSHFT, RDSUPT, RDUTLT, RDWAST
ICFLG	Logic flag	--	OTWAST, RDWAST
ICH	Logic flag	--	OLDCASE, WRITE
ICIN	Logic flag	--	OLDCASE
ICLR	Year inventories of stored non-heat generating waste can be cleared by solo emplacement	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
ICNT	Dummy variable	--	MAIN

A.57

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
IDD	Decommissioning year index	--	OLDCASE, PROCED, WRITE
IDDYR	Year decommissioning starts	calendar year	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
IDET	Detailed cost input switch	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
IEND		--	DECODE1, DECODER
IEQTP1	Linear or exponential equation selection switch	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
IEQTP2	Linear or exponential equation selection switch	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
IEQTP3	Linear or exponential equation selection switch	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
IEQTP4	Linear or exponential equation selection switch	--	OLDCASE, OTPREP, PROCED, RDPREP, WRITE
IEQTP5	Linear or exponential equation selection switch	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
IFLG	Linear or exponential equation selection switch	--	MAIN, OTSHFT, RDGEN, RDROCK, RDRPOS, RDSHFT, RDSUPT, RDWAST
II	Index	--	OTWAST, RDWAST, WRITE
IMET	Emplacement method switch	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
IMETF1	Logic flag	--	RDWAST
IMETF2	Logic flag	--	RDWAST
IMETF3	Logic flag	--	RDWAST
IN	Unit device designator	--	MAIN, OTROCK, OTSHFT, OTWAST, PROCED, RDDECM, RDGEN, RDPREP, RDROCK, RDRPOS, RDSHFT, RDSUPT, ROUTLT, RDWAST

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
INDEX	Category Index array	--	MAIN, READ
INUM	Index	--	MAIN, PROCED, RDGEN, RDRPOS, RDSHFT, RDWAST
INWT	Index	--	MAIN, PROCED, RDGEN, RDRPOS, RDSHFT, RDWAST
IO	Unit device designator	--	MAIN, PROCED, RDDECM, RDGEN, RDPREP, RDROCK, RDRPOS, RDSHFT, RDSUPT, RDUTLT, RDWAST
IOPT	Option indicator	--	MAIN
IOPTA	Option indicator	--	MAIN, PROCED, RDGEN, RDPREP, RDROCK, RDRPOS, RDSHFT, RDSUPT, RDUTLT, RDWAST
IOPTB	Option indicator	--	MAIN, PROCED, RDGEN, RDROCK, RDRPOS, RDSHFT, RDUTLT, RDWAST
IOPTC	Option indicator	--	MAIN, PROCED
IOUT	Unit device designator	--	OTDECM, OTGEN, OTPREP, OTROCK, OTRPOS, OTSHFT, OTSUPT, OTUTLT, OTWAST, PROCED
IRFLG	Logic flag	--	OTWAST, RDWAST
ISEC	Section category flag	--	MAIN
ISECF	Section category flag	--	MAIN
ISFLG	Logic flag	--	MAIN
ISTR	Logic flag	--	DECODE I, DECODER
IT	Index	--	DECODE I, DECODER
ITSURV	Surveillance period	year	OLD CASE, OTGEN, PROCED, RDGEN, WRITE

A.59

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
IUHDCM	Hole drilling cost calculation method switch	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
IWNUM	Index	--	MAIN, PROCED, RDGEN, RDRPOS, RDSHFT, RDWAST
IYDOL	Discount year	calendar year	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
IYR	Year variable used to print calendar year	--	OTWAST, RDWAST
IYRCON	Year in which construction begins	calendar year	OLDCASE, OTDECM, OTGEN, OTWAST, PROCED, RDDECM, RDGEN, RDWAST, WRITE
IYRDIS	Year to which costs are discounted	calendar year	OLDCASE, PROCED, WRITE
IYRPRO	Year in which production begins	calendar year	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
IZERO	Constant value	--	TABLE
J	Index	--	DECODEI, DECODER, MAIN, OLDCASE, OTDECM, OTGEN, OTRPOS, OTWAST, RDDECM, RDGEN, RDRPOS, RDWAST, WRITE
JFLG	Logic flag	--	TABLE
JJ	Index	--	OTRPOS, RDRPOS, WRITE
J1	Index	--	OTGEN, RDGEN
K	Index	--	DECODEI, DECODER, MAIN, OLDCASE, OTDECM, OTGEN, OTRPOS, OTSHFT, OTWAST, RDDECM, RDGEN, RDRPOS, RDSHFT, RDWAST, WRITE
KIND	Waste type definition: 1 = Remote handled 2 = Contact handled 3 = Tru	--	OLDCASE, OTSHFT, OTWAST, PROCED, RDSHFT, RDWAST, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
KNUM	Index	--	RDGEN, RDRPOS, RDSHFT, RDWAST
L	Index	--	DECODEI, DECODER, MAIN, OLDCASE, OTWAST, RDWAST, WRITE
LASTYR	Last year of operation	--	OLDCASE, OTDECM, OTGEN, OTWAST, PROCED, RDDECM, RDGEN, RDWAST, WRITE
LEAD	Lead time for room excavation	years	OLDCASE, OTROCK, PROCED, RDROCK, WRITE WRITE
LL	Index	--	OLDCASE, OTDECM, OTGEN, RDDECM, RDGEN, WRITE
LY	Index (usually refers to end of waste operations)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
LYY	Index	--	OLDCASE, PROCED, WRITE
M	Index	--	OLDCASE, OTWAST, RDSUPT, RDWAST, WRITE
MCNT	Logic flag	--	OTWAST, RDWAST
MFLG	Logic flag	--	RDROCK
MM	Index	--	OTGEN, OTWAST, RDGEN, RDSUPT, RDWAST
MPRIME	Number of primary heat-generating waste types	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
MROP	Mined rock operations power	kW/tonne	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
MSA	Mined rock storage area	acres	OLDCASE, OTROCK, PROCED, RDROCK, WRITE
MSHIFT	Number of shift per day worked (250-day year)	shift/day	OLDCASE, OTROCK, PROCED, RDROCK, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
MSWTCH	Mining cost input type selector	--	OLDCASE, OTROCK, PROCED, RDROCK, WRITE
MTCPSD	Maximum shaft trips per shift for CH waste	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
MTRPSD	Maximum shaft trips per shift for RH waste	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
MXMMRH	Maximum trips per shift-year for rock hoisted in men and materials shaft	tonnes/year	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
MXMPRH	Maximum trips per shift-year for rock hoisted in mine production shaft	tonnes/year	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
N	Index	--	DECODE1, OTSUPT, RDROCK, RDSUPT
NACOR	Number of additional corridor types	--	OTRPOS, RDRPOS
NALSH	Number of shifts allowed	--	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
NBSPPA	Number of bore sections per panel	--	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
NBYR	Time from waste operations startup to backfilling startup	years	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
NC	Maximum number of construction years	--	OTDECM, PROCED, RDDECM
NCBPRM	Number of receiving bays per module - CH	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
NCHRMA	Maximum CH waste-receiving modules	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
NCOR	Number of corridor types in a panel	--	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
NCORPP	Number of corridors per panel	--	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
NCR	Maximum number of corridor types defined	--	PROCED
NDC	Number of decommissioning cost categories	--	PROCED
NDDC	Variable set aside for future use (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
NEE	Number of emplacement excavation methods	--	OLDCASE, PROCED, WRITE
NEO	Number of emplacement methods	--	PROCED
NFAC	Number of surface facility categories	--	PROCED
NFCLT	Answer response	--	RDSUPT
NFTL	Near-field thermal limit	kW/acre	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
NNHG	Number of non-heat-generating waste types	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
NPKEU	Number of packages per emplacement unit	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
NPKPTC	Number of packages per shaft trip - CH	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
NPKPTR	Number of packages per shaft trip - RH	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
NRBPRM	Number of receiving bays per module - RH	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
NRHRMA	Maximum RH waste-receiving modules	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
NRO	Number of rock operations	--	OLDCASE, PROCED, WRITE
NRWSPR	Number of rows per room	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
NS	Number of labor cost differentials for shift work	--	OLDCASE, PROCED, WRITE

A.63

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
NSC	Number of support facilities	--	OLDCASE, OTSUPT, PROCED, RDSUPT, WRITE
NST	Number of shaft types	--	OLDCASE, OTDECM, OTSHFT, PROCED, RDDECM, RDSHFT, WRITE
NSTM1	Number of repository shafts minus 1	--	OLDCASE, WRITE
NT0	Number of transportation operations	--	OLDCASE, PROCED, WRITE
NTYP	Hole drilling linear or exponential function switch	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
NUM	Index	--	MAIN, RDGEN, RDWAST
NUO	Number of underground operations	--	OLDCASE, PROCED, WRITE
NW	Maximum number of waste types	--	MAIN, PROCED
NWO	Number of surface waste operations	--	OLDCASE, PROCED, WRITE
NWP1	Summation Index	--	PROCED
NWT	Number of waste types	--	OLDCASE, OTDECM, OTGEN, OTPREP, OTRPOS, OTSHFT, OTWAST, PROCED, RDDECM, RDGEN, RDPREP, RDRPOS, RDSHFT, RDWAST, WRITE
NY	Number of years (for variable dimension)	--	PROCED
NYRO	Maximum time allowed for all repository activities	--	PROCED
OFTADC	Offsite transportation and disposal unit cost	\$/tonne	OLDCASE, OTROCK, PROCED, RDROCK, WRITE
ORS	Steam requirements - other facilities	10 lbs/year	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
OTBDEN	Other backfill density	g/cm ³	OLDCASE, OTROCK, PROCED,
OTHBFL	Percentage of other backfill per m ³ backfilled	percent	OLDCASE, OTROCK, PROCED, RDROCK, WRITE

A.64

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
OVRHDF	Overhead factor	--	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
OWHP	Other waste-handling building electrical load	kW	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
PANLEN	Panel length	meters	OLDCASE, PROCED, RDRPOS, WRITE
PANWTH	Panel width	meters	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
PCTCON	Fraction secondary waste emplacement capacity available for secondary waste	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
PCTFIL	Percentage of room filled w/expanded backfill	--	OLDCASE, OTROCK, PROCED, RDROCK, WRITE
PKGCAP	Packaging facility capacity per shift	waste units/ shift-year	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
PKGDIA	Package diameter	meters	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
PKGDS1	Package narrative description	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
PKGDS2	Package narrative description	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
PKGDS3	Package narrative description	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
PKGLN	Package length	meters	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
PKGPRK	Packages per rack unit	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
PKGUP	Packages per unit pitch	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
PLACE	Location of repository	--	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
PORMDF	Pre-operation mining cost time distribution	fraction	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
RAPP	Electrical load - receiving and packaging power	kW/module	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
RAPS	Steam requirement for receiving and packaging	10 ³ lbs/year	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
RCCAP	Rail container capacity	waste units	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
RCKCST	Rack cost for trench emplacement	\$/rack	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
RCKHC	Unit rock hoisting cost	\$/tonne	OLDCASE, OTROCK, PROCED, RDROCK, WRITE
RED	Diesel fuel requirement for rail engines	10 ³ liters/ container	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
REFBKF	Reference backfill material cost for boreholes	\$/hole	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
REFDEP	Reference depth for boreholes	meters	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
REFDIA	Reference diameter for boreholes	meters	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
REFMAT	Reference barrier material cost for boreholes	\$/hole	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
REFSLV	Reference sleeve costs for boreholes	\$/hole	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
RMCWS1	RH waste-receiving module construction schedule - Initial	--	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
RMWS2	RH waste-receiving module construction schedule - during operations	--	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
RMHMIN	Minimum room height	meters	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
RMLN	Room length	meters	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
RMNWS1	CH waste-receiving module construction schedule - initial	--	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
RMNWS2	CH waste-receiving module construction schedule - during operations	--	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
RMWTH	Room width	meters	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
ROGST	Rock operations capital costs	\$	OLDCASE, OTRUCK, PROCED, RDROCK, WRITE
ROECST	Rock operations unit equipment costs unit	\$/equipment	OLDCASE, OTRUCK, PROCED, RDROCK, WRITE
ROEQPC	Rock operations equipment capacities	tonnes/equip	OLDCASE, OTRUCK, PROCED, RDROCK, WRITE
ROEQPL	Rock operations equipment lifetimes	tonnes	OLDCASE, OTRUCK, PROCED, RDROCK, WRITE
ROLBRC	Rock operations unit labor costs	\$/man-year	OLDCASE, OTRUCK, PROCED, RDROCK, WRITE
ROLBRR	Rock operations unit labor requirements	man-years/ equipment unit shift	OLDCASE, OTRUCK, PROCED, RDROCK, WRITE
ROMATC	Rock operations unit material costs	\$/material unit	OLDCASE, OTRUCK, PROCED, RDROCK, WRITE
ROMATU	Rock operations unit material requirements	material unit/ tonne	OLDCASE, OTRUCK, PROCED, RDROCK, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
RPTYP	Repository description	--	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
RVP	Ventilation power	kW/acre of repository	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
SCCON1	Shaft sinking function constant 1	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
SCCON2	Shaft sinking function constant 2	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
SCCST	Variable for future use (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
SCDF	Startup cost distribution factor	--	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
SCEQPC	Variable for future use (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
SCLBRC	Variable for future use (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
SCLBRR	Variable for future use (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
SCMATC	Variable for future use (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
SCMATU	Variable for future use (not currently used)	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
SCSKD	Shaft construction schedule	--	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
SFCS	Surface facility construction schedule	--	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
SFCST	Support facility capital cost	\$	OLDCASE, OTSUPT, PROCED, RDSUPT, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
SFECST	Support facility equipment cost	\$/facility	OLDCASE, OTSUPT, PROCED, RDSUPT, WRITE
SFEQPL	Support facility equipment life	years	OLDCASE, OTSUPT, PROCED, RDSUPT, WRITE
SFLBRC	Support facility labor cost	\$/man-year	OLDCASE, OTSUPT, PROCED, RDSUPT, WRITE
SFLBRR	Support facility labor requirement	man-year/ facility/shift	OLDCASE, OTSUPT, PROCED, RDSUPT, WRITE
SFMATC	Support facility material cost	\$/material unit	OLDCASE, OTSUPT, PROCED, RDSUPT, WRITE
SFMATU	Support facility material requirements	material units/ facility	OLDCASE, OTSUPT, PROCED, RDSUPT, WRITE
SFP	Support facility electrical load	kW	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
SFSP	Spent fuel storage electrical load	kW	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
SFTDD	Shaft design diameter	meters	OLDCASE, OTDECM, OTSHFT, PROCED, RDDECM, RDSHFT, WRITE
SFTHE	Holsting equipment cost	\$	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
SFTITL	Support facility name	--	OLDCASE, OTSUPT, PROCED, RDSUPT, WRITE
SFTLD	Shaft lined depth	meters	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
SFTNUM	Shaft number	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
SFTSS	Underground shaft station construction cost	\$	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
SFTXD	Shaft excavation depth	meters	OLDCASE, OTSHT, PROCED, RDSHT, WRITE
SFTXR	Shaft incremental excavation radius	meters	OLDCASE, OTDECM, OTSHT, PROCED, RDDECM, RDSHT, WRITE
SHCFAC	Shift cost factor	1 + fraction	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
SIZE	Repository size	acres	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
SLBR	Number of shifts per day worked by support personnel	--	OLDCASE, OTSUPT, PROCED, RDSUPT, WRITE
SLEETH	Sleeve thickness	meters	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
SOA	Surface operations area	acres	OLDCASE, OTSUPT, PROCED, RDSUPT, WRITE
SPUC	Site preparation cost	\$ thousand/acre	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
SSCON1	Shaft sealing function constant 1	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE,
SSCON2	Shaft sealing function constant 2	--	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
SSDF	Shaft sealing cost distribution over time	--	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
STMC	Unit steam cost	10 ³ lbs/year	OLDCASE, OTULT, PROCED, RDUULT, WRITE
STQMAX	Storage capacity limit for secondary waste for coemplacement per primary waste package	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
STPCT	Percentage of field costs used for startup costs	--	OLDCASE, OTPREP, PROCED, RDPREP, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
SUD	Diesel fuel requirement - surface use	10 ³ liters/day	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
SURCST	Annual surveillance cost	\$/year	OLDCASE, OTDECM, PROCED, RDDECM, WRITE
SURFDC	Unit surface handling and storage costs	\$/tonne	OLDCASE, OTROCK, PROCED, RDROCK, WRITE
SWPPR	Secondary waste package per room solo emplacement	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TA	Dummy variable	--	TABLE
TABLES	Output tables requested to be printed	--	OLDCASE, PROCED, TABLES, WRITE
TCCAP	Truck container capacity	waste units/ cask	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TCLAND	Land purchase cost	\$	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
TCLICE	Licensing cost	\$	OLDCASE, OTPREP, PROCED, RDPREP, WRITE
TCPCLN	Total central pillar corridor length	meters	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE
TDWI	Heat density per waste unit	kW/waste unit	OLDCASE, OTDECM, OTWAST, PROCED, RDDECM, RDWAST, WRITE
TEKNEK	Mining method	--	OLDCASE, OTROCK, PROCED, RDROCK, WRITE
TITLE	Title	--	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
TNPT1	Rock hoisting capacity for rock hoisting shaft	tonnes/trip	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
TNPT2	Rock hoisting capacity for men and materials shaft	tonnes/trip	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
TOCST	Total waste operations costs	\$	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TOECST	Transportation operations unit equipment cost	\$/equipment unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TOEQPC	Transportation operations equipment capacity	packages/equip. unit/shift	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TOEQPL	Transportation operations equipment life	packages	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TOLBRC	Transportation operations unit labor cost	\$/man-year	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TOLBRR	Transportation operations unit labor requirement	man-years/ equipment unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TOMATC	Transportation operations unit material cost	\$/material unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TOMATU	Transportation operations unit material requirement	material unit/ package	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TONPAR	Surface rock storage density	tonnes/acre	OLDCASE, OTROCK, PROCED, RDROCK, WRITE
TRCHWD	Trench width	meters	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TTTK	Average receiving facility turnaround time per truck	days	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
TTTR	Average receiving facility turnaround time per rail car	days	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
UGD	Diesel fuel requirement - underground	10 ³ liters/ package	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
UNTMAS	Mass per waste unit	kilograms	OLDCASE, OTWAST, PROCED, RDWAST, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
UOCST	Emplacement operations capital cost	\$ millions	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
UOECST	Emplacement operations unit equipment cost	\$/equipment unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
UOEQPC	Emplacement operations equipment capacity	emplacements/ equipment/shift	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
UOEQPL	Emplacement operations equipment life	emplacements	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
UOLBRC	Emplacement operations unit labor cost	\$/man-year	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
UOLBRR	Emplacement operations unit labor requirement	man-year/equip. unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
UOMATC	Emplacement operations unit material cost	\$/material unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
UOMATU	Emplacement operations material unit	material unit/ emplacement unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
UPC	Power cost	cents/kWh	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
VARMC	Variable mining cost	\$/tonne	OLDCASE, OTROCK, PROCED, RDROCK, WRITE
VOLDW	Demineralized water volume	10 ³ liters/day	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
VOLW	Water volume	10 ³ liters/day	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
WABATH	Water barrier thickness	meters	OLDCASE, OTWAST, PROCED, RDGEN, RDWAST, WRITE
WCCON1	Shaft sinking water control cost constant 1	--	OLDCASE, OTSHFT, PROCED, RDShft, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

Variable Name	Description	Units	Location
WCCON2	Shaft sinking water control cost constant 2	--	OLDCASE, OTSHFT, PROCED, RDSHFT, WRITE
WOCST	Waste operations capital cost	\$	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
WOECST	Waste operations unit equipment cost	\$/equipment unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
WOEQPC	Waste operations unit equipment capacity	waste unit/equip- ment/unit/shift	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
WOEQPL	Waste operations unit equipment life	waste units	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
WOLBRC	Waste operations unit labor cost	\$/man-year	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
WOLBRR	Waste operations unit labor requirement	man-years/ shift/module	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
WOMATC	Waste operations unit material cost	\$/material unit	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
WOMATU	Waste operations unit material requirement	material unit/ module	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
WRKYR	Days in work year - constant (250)	--	OLDCASE, OTGEN, PROCED, RDGEN, WRITE
WSTSUP	Annual waste units requiring disposal	waste units	OLDCASE, OTDECM, OTGEN, PROCED, RDDECM, RDGEN, WRITE
WSTYP	Waste type description	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
WTRC	Unit water cost	cents/10 ⁶ liters	OLDCASE, OTUTLT, PROCED, RDUTLT, WRITE
WUPP	Maximum waste units per package	--	OLDCASE, OTWAST, PROCED, RDWAST, WRITE

PREPROCESSOR VARIABLE LISTING (contd)

<u>Variable Name</u>	<u>Description</u>	<u>Units</u>	<u>Location</u>
WVOLPU	Waste volume per waste unit	m ³	OLDCASE, OTWAST, PROCED, RDWAST, WRITE
XPANFC	Rock expansion factor after mining	fraction	OLDCASE, OTROCK, PROCED, RDROCK, WRITE
XRASHO	Extraction ratio (room/room + pillar)	--	OLDCASE, OTRPOS, PROCED, RDRPOS, WRITE

APPENDIX B

RECON MODEL LISTING

APPENDIX B

RECON MODEL LISTING

Included in this appendix is a listing of the RECON computer program, together with a set of data tables showing the model output. The data in the tables are shown simply for illustration and do not correspond to any actual case.


```

C
PARAM PROC
C
  REAL NFTL
  REAL MSA
  REAL MROP
  INTEGER RMPILU
  INTEGER SFTNUM
  CHARACTER*8,DA,TI
C
  PARAMETER NY = 100
  PARAMETER IO = 6
  PARAMETER IN = 5
  PARAMETER NW = 10
  PARAMETER NWP1 = NW +1
  PARAMETER NC = 10
  PARAMETER NCR = 6
  PARAMETER NS = 3
  PARAMETER NST= 7
  PARAMETER NDC = 2
  PARAMETER NFAC = 50
  PARAMETER NRO = 5
  PARAMETER NTO =2
  PARAMETER NUO = 2
  PARAMETER NWO = 3
  PARAMETER NEE = 3
  PARAMETER NEO = 5
  PARAMETER NYRO = NY+40
  PARAMETER PI = 3.1415927
  PARAMETER M2ACRE = 4047
  PARAMETER ONEMIL = 1000000.
  PARAMETER ONETHO = 1000.
  PARAMETER NEQP = 24
  PARAMETER IZ1 = 3882
  PARAMETER IZ2 = 100
  PARAMETER IZ3 = 9576
  PARAMETER IZ4 = 525
  PARAMETER IZ5 = 14004
  PARAMETER IZ6 = 2785
  PARAMETER IZ7 = 4345
  PARAMETER IZ8 = 372
  PARAMETER IZ9 = 134
  PARAMETER IZ10 = 925
  PARAMETER IZ11 = 6232
  PARAMETER IZ12 = 2720
  PARAMETER IZ13 = 159
C
  DIMENSION ZERO1 (IZ1)
  COMMON/PARAM/ AREAPP(NW)

```

```

COMMON/PARAM/ CORARA
COMMON/PARAM/ CORMAS(NW,NC)
COMMON/PARAM/ CORWTH(NCR)
COMMON/PARAM/ CPCMM(NW,NC)
COMMON/PARAM/ CSTOT(NY)
COMMON/PARAM/ DPY
COMMON/PARAM/ DENSTY
COMMON/PARAM/ IDD
COMMON/PARAM/ IEMP(NW)
COMMON/PARAM/ IMET(NW)
COMMON/PARAM/ ISC
COMMON/PARAM/ ISR
COMMON/PARAM/ IYDOL
COMMON/PARAM/ IYRCON
COMMON/PARAM/ IYRDIS
COMMON/PARAM/ IYRPRO
COMMON/PARAM/KIND(NW)
COMMON/PARAM/ LEAD
COMMON/PARAM/ LL
COMMON/PARAM/ LY
COMMON/PARAM/ LYY
COMMON/PARAM/ MSWTCH
COMMON/PARAM/ NPSMWT(NW)
COMMON/PARAM/ NSC
COMMON/PARAM/ NUMPKG(NWP1,NY)
COMMON/PARAM/ NWT
COMMON/PARAM/ PKGLN(NW)
COMMON/PARAM/ PKGPUP(NW)
COMMON/PARAM/ PNLREQ(NW,NY)
COMMON/PARAM/ RECUNT(NWP1,NY)
COMMON/PARAM/ RECUWT(NW)
COMMON/PARAM/ RCKHST(NW,NYRO)
COMMON/PARAM/ RKBKF (NW,NYRO)
COMMON/PARAM/ RKMSM(NW,NY)
COMMON/PARAM/ RKVOL(NW)
COMMON /PARAM/ SRKVOL
COMMON/PARAM/ RMFILD(NWP1,NY+10)
COMMON/PARAM/ PRMMAS
COMMON/PARAM/ MRFLG(NW)
COMMON/PARAM/ RMLN(NW)
COMMON/PARAM/ RMWTH(NW)
COMMON/PARAM/ RPTYP(4)
COMMON/PARAM/ RRCTOT(NY)
COMMON/PARAM/ SFTDD (NST)
COMMON/PARAM/ SFTNUM (NST)
COMMON/PARAM/ SHAFTX(NW)
COMMON/PARAM/ SFTXD (NST)
COMMON/PARAM/ SFTXR (NST)
COMMON/PARAM/ SIZE

```

```

COMMON/PARAM/  SRKMND(NY)
COMMON/PARAM/  SRMFLD
COMMON/PARAM/  SRRMC(NY)
COMMON/PARAM/  SRRMR(NY)
COMMON/PARAM/  SUMIND
COMMON/PARAM/  TWOPCT(NWP1,NY)
COMMON/PARAM/  UBFRMS
COMMON/PARAM/  VENCLN(NW)
COMMON/PARAM/  VENCHT(NW)
COMMON/PARAM/  VENCWT(NW)
COMMON/PARAM/  VOLMND(NW,NY)
COMMON/PARAM/  WRKYR
COMMON/PARAM/  WSTYP(3,NWP1)
COMMON/PARAM/  WSTVOL(NW,NY)
COMMON/PARAM/  WUPP(NW)
EQUIVALENCE(ZERO1  (1),AREAPP)

```

```

C
END

```

```

C
C

```

```

COCOM  PROC
COMMON/COCOM/  BSFILD(NW,NY)
COMMON/COCOM/  CHTRU(NY)
COMMON/COCOM/  COCON(NW,NW)
COMMON/COCOM/  CONTRU(NW)
COMMON/COCOM/  COWAST(NW,NW,NY+10)
COMMON/COCOM/  ICLR
COMMON/COCOM/  MPRIME
COMMON/COCOM/  NNHG
COMMON/COCOM/  PCTCON(NW,NW)
COMMON/COCOM/  RMSPBS(NW)
COMMON/COCOM/  STOMAX(NWP1)
COMMON/COCOM/  STOR(NWP1)
COMMON/COCOM/  SUPT
COMMON/COCOM/  SURT
COMMON/COCOM/  SWPPR(NW)

```

```

END

```

```

C
C

```

```

INCOM  PROC
C

```

```

DIMENSION  ZERO2  (IZ2)
COMMON/INCOM/  BKFLC
COMMON/INCOM/  BLANK
COMMON/INCOM/  CONFAC(4)
COMMON/INCOM/  DRATE
COMMON/INCOM/  HDECFC
COMMON/INCOM/  HDLCFC
COMMON/INCOM/  HDMCFC

```

```

COMMON/INCOM/ IDET
COMMON/INCOM/ IUHDCM
COMMON/INCOM/ P (NW)
COMMON/INCOM/ PKGDS1(4,NW)
COMMON/INCOM/ PKGDS2(4,NW)
COMMON/INCOM/ PKGDS3(4,NW)
COMMON/INCOM/ NPKPEU(NW)
COMMON/INCOM/ PLACE (4)
COMMON/INCOM/ RCKHC
COMMON/INCOM/ SURFDC
COMMON/INCOM/ SPUC
COMMON/INCOM/ TEKNEK(4)
COMMON/INCOM/ TOMATC(NT0,NW)
COMMON/INCOM/ TOMATU(NT0,NW)
COMMON/INCOM/ UNTMAS(NW)
COMMON/INCOM/ VARMC
COMMON/INCOM/ NBSPPA(NW)
COMMON/INCOM/ EQUIVALENCE(ZERO2 (1),BKFLC)

```

19JAN

END

C

OUTCOM PROC

```

COMMON/OUTCOM/ ACPOWC
COMMON/OUTCOM/ CEQPT(5)
COMMON/OUTCOM/ CLABOR(5)
COMMON/OUTCOM/ CMATL(5)
COMMON/OUTCOM/ CPMAT(NW)
COMMON/OUTCOM/ CMMASS
COMMON/OUTCOM/ CPEQ(NW)
COMMON/OUTCOM/ CPLBR(NW)
COMMON/OUTCOM/ HDMASS
COMMON/OUTCOM/ OSTMC
COMMON/OUTCOM/ PFPOWC(NW)
COMMON/OUTCOM/ REBFC
COMMON/OUTCOM/ RECUC
COMMON/OUTCOM/ REPC
COMMON/OUTCOM/ RDSLC
COMMON/OUTCOM/ RHCC
COMMON/OUTCOM/ RSTMC
COMMON/OUTCOM/ SDSLC
COMMON/OUTCOM/ SFEXC
COMMON/OUTCOM/ SFMASS
COMMON/OUTCOM/ SFMMC
COMMON/OUTCOM/ SFMXC
COMMON/OUTCOM/ SFPOWC
COMMON/OUTCOM/ SFRHC
COMMON/OUTCOM/ SFTSLC(NST)
COMMON/OUTCOM/ SFVSC
COMMON/OUTCOM/ SPMASS
COMMON/OUTCOM/ UDSLC

```

```

COMMON/OUTCOM/ WRBKFL
COMMON/OUTCOM/ ROPOWC
COMMON/OUTCOM/ VENTPC
COMMON/OUTCOM/ RFPOWC
COMMON/OUTCOM/ WEEPC
COMMON/OUTCOM/ SCEC
COMMON/OUTCOM/ SFWSC
COMMON/OUTCOM/ SFWSE

```

END

C

C

HUBCOM PROC

C

```

DIMENSION ZERO3 (IZ3)
COMMON/HUBCOM/ ABDCT(NYRO)
COMMON/HUBCOM/ ANBFCT(NYRO)
COMMON/HUBCOM/CONTGY(NW,NYRO,4)
COMMON/HUBCOM/ACONTG(NYRO)
COMMON/HUBCOM/ CSTBKF(NW,NYRO)
COMMON/HUBCOM/ Csthst(NW,NYRO)
COMMON/HUBCOM/ CSTOSD(NW,NYRO)
COMMON/HUBCOM/ CSTSHS(NW,NYRO)
COMMON/HUBCOM/ DA
COMMON/HUBCOM/ DEPTH
COMMON/HUBCOM/ ECOST(NWP1,NY)
COMMON/HUBCOM/ EMCST(NWP1,NY)
COMMON/HUBCOM/ IPMCW(NY)
COMMON/HUBCOM/ IRMCW (NY)
COMMON/HUBCOM/ IRMNW (NY)
COMMON/HUBCOM/ OPMC (NW,NYRO)
COMMON/HUBCOM/ PACST(NWP1,NY)
COMMON/HUBCOM/ PFCST (NW,NY)
COMMON/HUBCOM/ PWF (NYRO)
COMMON/HUBCOM/ RECST(NWP1,NY)
COMMON/HUBCOM/ RFCST (NW,NY)
COMMON/HUBCOM/ SHCPY(NW,NY)
COMMON/HUBCOM/ SLCU (3,NWP1)
COMMON/HUBCOM/ TACCST(NYRO)
COMMON/HUBCOM/ TAECST(NWP1)
COMMON/HUBCOM/ TBDCST(NWP1)
COMMON/HUBCOM/ TBFCST(NWP1)
COMMON/HUBCOM/ TCNCST(NWP1)
COMMON/HUBCOM/ TDCCST(NWP1)
COMMON/HUBCOM/ TECAPC
COMMON/HUBCOM/ TEQPT
COMMON/HUBCOM/ TEMCST(NWP1)
COMMON/HUBCOM/ TEXCST(NWP1)
COMMON/HUBCOM/ THSCST(NWP1)
COMMON/HUBCOM/ TI

```

COMMON/HUBCOM/	TITLE (20)
COMMON/HUBCOM/	TOTSFC
COMMON/HUBCOM/	TLCCST(NWP1)
COMMON/HUBCOM/	TLDCST(NWP1)
COMMON/HUBCOM/	TODCST(NWP1)
COMMON/HUBCOM/	TOMCST(NWP1)
COMMON/HUBCOM/	TOPCST(NWP1)
COMMON/HUBCOM/	TPKCST(NWP1)
COMMON/HUBCOM/	TRCCST(NWP1)
COMMON/HUBCOM/	TRCST(NWP1,NY)
COMMON/HUBCOM/	TSFCST(NWP1)
COMMON/HUBCOM/	TSHCST(NWP1)
COMMON/HUBCOM/	TSLCST(NWP1)
COMMON/HUBCOM/	TSPCST(NWP1)
COMMON/HUBCOM/	TSSCST(NWP1)
COMMON/HUBCOM/	TSUBCN(NWP1)
COMMON/HUBCOM/	TSUBRO(NWP1)
COMMON/HUBCOM/	TSUCST(NWP1)
COMMON/HUBCOM/	TSVCST(NWP1)
COMMON/HUBCOM/	TTCAPC
COMMON/HUBCOM/	TTEC(NT0)
COMMON/HUBCOM/	TTRCST(NWP1)
COMMON/HUBCOM/	TUCAPC
COMMON/HUBCOM/	TUFCST(NWP1)
COMMON/HUBCOM/	TUTCST(NWP1)
COMMON/HUBCOM/	TWTCST(NWP1)
COMMON/HUBCOM/	TWOEQC(NY)
COMMON/HUBCOM/	UAECST(NWP1,3)
COMMON/HUBCOM/	UBDCST(NWP1,3)
COMMON/HUBCOM/	UBFCST(NWP1,3)
COMMON/HUBCOM/	UCNCST(NWP1,3)
COMMON/HUBCOM/	UDCCST(NWP1,3)
COMMON/HUBCOM/	UEMCST(NWP1,3)
COMMON/HUBCOM/	UEXCST(NWP1,3)
COMMON/HUBCOM/	UGFC(NW,NC)
COMMON/HUBCOM/	UHSCST(NWP1,3)
COMMON/HUBCOM/	ULCCST(NWP1,3)
COMMON/HUBCOM/	ULDCST(NWP1,3)
COMMON/HUBCOM/	UODCST(NWP1,3)
COMMON/HUBCOM/	UOMCST(NWP1,3)
COMMON/HUBCOM/	UOPCST(NWP1,3)
COMMON/HUBCOM/	UPKCST(NWP1,3)
COMMON/HUBCOM/	URCCST(NWP1,3)
COMMON/HUBCOM/	USFCST(NWP1,3)
COMMON/HUBCOM/	USHCST(NWP1,3)
COMMON/HUBCOM/	USLCST(NWP1,3)
COMMON/HUBCOM/	USPCST(NWP1,3)
COMMON/HUBCOM/	USSCST(NWP1,3)
COMMON/HUBCOM/	USUBCN(NWP1,3)

```

COMMON/HUBCOM/  USUBRO(NWP1,3)
COMMON/HUBCOM/  USUCST(NWP1,3)
COMMON/HUBCOM/  USVCST(NWP1,3)
COMMON/HUBCOM/  UTRCST(NWP1,3)
COMMON/HUBCOM/  UUFCST(NWP1,3)
COMMON/HUBCOM/  UUTCST(NWP1,3)
COMMON/HUBCOM/  UWTCST(NWP1,3)
COMMON/HUBCOM/  WOPLBR(NWP1,NY)
EQUIVALENCE(ZERO3  (1),ABDCST(1))

```

END

C

C

RECCOM PROC

C

```

DIMENSION ZERO4  (IZ4)
COMMON/RECCOM/  NMODCH
COMMON/RECCOM/  NMODRH
COMMON/RECCOM/  RMCWS1(NC)
COMMON/RECCOM/  RMCWS2(NC)
COMMON/RECCOM/  RMNWS1(NC)
COMMON/RECCOM/  RMNWS2(NC)
COMMON/RECCOM/  SRHSHP(NY)
COMMON/RECCOM/  SCHSHP(NY)
COMMON/RECCOM/  TCRHRM
COMMON/RECCOM/  TCCHRM
COMMON/RECCOM/  TCRHPM
COMMON/RECCOM/  TTRHPM(NW)
COMMON/RECCOM/  TRSHP(NY)
COMMON/RECCOM/  TSHPCN(NW,NY)
EQUIVALENCE(ZERO4  (1),NMODCH)

```

C

END

C

C

SOPCOM PROC

C

```

DIMENSION ZERO5  (IZ5)
COMMON/SOPCOM/  AMLBCT(NY)
COMMON/SOPCOM/  ANPAR(NY)
COMMON/SOPCOM/  APLBCT(NY)
COMMON/SOPCOM/  ARLBCT(NY)
COMMON/SOPCOM/  OWLBR(NY)
COMMON/SOPCOM/  ATLBCT(NY)
COMMON/SOPCOM/  AXLBCT(NY)
COMMON/SOPCOM/  EEMATR(NW,NY)
COMMON/SOPCOM/  EEXLBR(NW,NY)
COMMON/SOPCOM/  EEXRKM(NW,NY)
COMMON/SOPCOM/  EMEQCT(NY)
COMMON/SOPCOM/  EMEQPC(NW,NY)

```


COMMON/SOPCOM/ EMLBCT(NW,NY)
COMMON/SOPCOM/AVLRPA(NY)
COMMON/SOPCOM/ CHRLBR(NY,NS)
COMMON/SOPCOM/CLSEMP(NW)
COMMON/SOPCOM/ CMCAP
COMMON/SOPCOM/CPA
COMMON/SOPCOM/ CRRAIL(NW,NY)
COMMON/SOPCOM/CUMPAR(NY)
COMMON/SOPCOM/ EMMATC(NW,NY)
COMMON/SOPCOM/ EMMTCT(NY)
COMMON/SOPCOM/ EMPLBR(NW,NY)
COMMON/SOPCOM/ EMPMCH(NWP1,NY)
COMMON/SOPCOM/ ERCLBR(NY,NS)
COMMON/SOPCOM/ ERRLBR(NY,NS)
COMMON/SOPCOM/ EQREEX(NW,NY)
COMMON/SOPCOM/ EQREMP(NW,NY)
COMMON/SOPCOM/ EQRPKG(NW,NY)
COMMON/SOPCOM/ EQRREC(NW,NY)
COMMON/SOPCOM/ EQRTRN(NW,NY)
COMMON/SOPCOM/ EXEQCT(NY)
COMMON/SOPCOM/ EXEQPC(NW,NY)
COMMON/SOPCOM/ EXLBR (NY,NS)
COMMON/SOPCOM/ EXLBCT(NW,NY)
COMMON/SOPCOM/ EXMACH(NWP1,NY)
COMMON/SOPCOM/ EXMATC(NW,NY)
COMMON/SOPCOM/ EXMTCT(NY)
COMMON/SOPCOM/GXEF
COMMON/SOPCOM/FFHTLD(NW,NY)
COMMON/SOPCOM/FFTL
COMMON/SOPCOM/HPMIN(NW)
COMMON/SOPCOM/HTLDNF(NW,NY)
COMMON/SOPCOM/ MTCPSD
COMMON/SOPCOM/ MTRPSD
COMMON/SOPCOM/ NALSH
COMMON/SOPCOM/ NEMPU(NW,NY)
COMMON/SOPCOM/ NCHRNA
COMMON/SOPCOM/NFTL (NW)
COMMON/SOPCOM/NPKGPR(NW,NY)
COMMON/SOPCOM/ NPMRH (NY)
COMMON/SOPCOM/ NRHRMA
COMMON/SOPCOM/ NRMCH (NY)
COMMON/SOPCOM/ NRMRH (NY)
COMMON/SOPCOM/NPPRW(NW,NY)
COMMON/SOPCOM/ NRWSPR(NW)
COMMON/SOPCOM/ OTMATC(NW,NY)
COMMON/SOPCOM/ OTMTCT(NY)
COMMON/SOPCOM/ OVRHDF
COMMON/SOPCOM/ PALBCT(NW,NY)
COMMON/SOPCOM/ PAMTCT(NY)

COMMON/SOPCOM/ PANLEN(NW)
COMMON/SOPCOM/ PANWTH(NW)
COMMON/SOPCOM/ PARQCT(NY)
COMMON/SOPCOM/ PITCH(NW,NY)
COMMON/SOPCOM/ PKEQPC(NW,NY)
COMMON/SOPCOM/ PKGCAP(NW)
COMMON/SOPCOM/ PKGLBR(NW,NY)
COMMON/SOPCOM/ PKMATC(NW,NY)
COMMON/SOPCOM/ PKMATR(NW,NY)
COMMON/SOPCOM/ PMCAP
COMMON/SOPCOM/ RCCAP (NW)
COMMON/SOPCOM/ RCEQPC(NW,NY)
COMMON/SOPCOM/ RCMATC(NW,NY)
COMMON/SOPCOM/ RCMATR(NW,NY)
COMMON/SOPCOM/ REEQCT(NY)
COMMON/SOPCOM/ RECLBR(NW,NY)
COMMON/SOPCOM/ RELBCT(NW,NY)
COMMON/SOPCOM/ REMTCT(NY)
COMMON/SOPCOM/ RHPLBR(NY,NS)
COMMON/SOPCOM/ RMCAP
COMMON/SOPCOM/ RHRLBR(NY,NS)
COMMON/SOPCOM/RMPILU(NW)
COMMON/SOPCOM/ SFTC (NS)
COMMON/SOPCOM/ SFTR (NS)
COMMON/SOPCOM/ SHCFAC(NS)
COMMON/SOPCOM/SRMWT(NW)
COMMON/SOPCOM/ SUMCPK(NY)
COMMON/SOPCOM/ SUMCRC(NY)
COMMON/SOPCOM/ SUMEPU(NY)
COMMON/SOPCOM/ SUMRPK(NY)
COMMON/SOPCOM/ SUMRRC(NY)
COMMON/SOPCOM/ TALBCT(NY)
COMMON/SOPCOM/TANF(NW)
COMMON/SOPCOM/ TCCAP (NW)
COMMON/SOPCOM/TDWI(NY,NW)
COMMON/SOPCOM/TMATCT(NY)
COMMON/SOPCOM/ TOCST (NTO)
COMMON/SOPCOM/ TOECST(NTO)
COMMON/SOPCOM/ TOEQPC(NTO)
COMMON/SOPCOM/ TOEQPL(NTO)
COMMON/SOPCOM/ TOLBRC(NTO)
COMMON/SOPCOM/ TOLBRR(NTO)
COMMON/SOPCOM/ TRCHWD(NW)
COMMON/SOPCOM/ TRCLBR(NY,NS)
COMMON/SOPCOM/ TREQCT(NY)
COMMON/SOPCOM/ TREQPC(NW,NY)
COMMON/SOPCOM/ TRLBCT(NW,NY)
COMMON/SOPCOM/ TRMATC(NW,NY)
COMMON/SOPCOM/ TRMATR(NW,NY)

8DEC
8DEC

```

COMMON/SOPCOM/ TRMTCT(NY)
COMMON/SOPCOM/ TRNLBR(NW,NY)
COMMON/SOPCOM/ TRNSPT(NWP1,NY)
COMMON/SOPCOM/ TRRLBR(NY,NS)
COMMON/SOPCOM/ TTTK (NW)
COMMON/SOPCOM/ TTTR (NW)
COMMON/SOPCOM/ TWEC (NWO)
COMMON/SOPCOM/ UGRMOD(NY)
COMMON/SOPCOM/ WOLABR(NW,NY)
COMMON/SOPCOM/ WOLBRR(NWO)
COMMON/SOPCOM/ WOCST (NWO)
COMMON/SOPCOM/ WOLBRC(NWO)
COMMON/SOPCOM/ WOECST(NWO)
COMMON/SOPCOM/ WOEQPL(NWO)
COMMON/SOPCOM/ WOEQPC(NWO)
COMMON/SOPCOM/ WOMATC(NWO,NW)
COMMON/SOPCOM/ WOMATU(NWO,NW)
COMMON/SOPCOM/WSTSUP(NW,NY)
COMMON/SOPCOM/WTHEET(NW)
COMMON/SOPCOM/WVOLPU(NW)
COMMON/SOPCOM/XRASHO(NW)
EQUIVALENCE(ZERO5 (1),AMLBCT(1))

```

C
END

C
C
SSTCOM PROC
C

```

DIMENSION ZERO6 (IZ6)
COMMON/SSTCOM/ CSTSP(NC)
COMMON/SSTCOM/ CSTSF(NY)
COMMON/SSTCOM/ASPC(NW,NY)
COMMON/SSTCOM/ EECAPC(NW,NY)
COMMON/SSTCOM/ OSFC(NW,NY)
COMMON/SSTCOM/ OWOPC(NWP1,NY)
COMMON/SSTCOM/ ROCAPC(NW,NY)
COMMON/SSTCOM/ SFCAPC(NW,NY)
COMMON/SSTCOM/ SFCCST(NW)
COMMON/SSTCOM/ SFCST (NFAC)
COMMON/SSTCOM/ SFEQPC(NW,NY)
COMMON/SSTCOM/ SFEQCT(NY)
COMMON/SSTCOM/ SFLBRC(NFAC)
COMMON/SSTCOM/ SFLBRR(NFAC)
COMMON/SSTCOM/ SFECST(NFAC)
COMMON/SSTCOM/ SFEQPL(NFAC)
COMMON/SSTCOM/ SFMATC(NFAC)
COMMON/SSTCOM/ SFMATU(NFAC)
COMMON/SSTCOM/ SFTITL(12,NFAC)
COMMON/SSTCOM/ SFCS (NC)

```

```

COMMON/SSTCOM/ SLBR(NFAC)
COMMON/SSTCOM/ SOA
COMMON/SSTCOM/ TRCAPC(NW,NY)
COMMON/SSTCOM/ TSFEC (NFAC)
COMMON/SSTCOM/ UFCAPC(NW,NY)
EQUIVALENCE(ZERO6(1),CSTSP(1))

```

C
END

C
C
MNDCOM PROC
C

```

DIMENSION ZERO7 (IZ7)
COMMON/MNDCOM/ ANFILE(NY)
COMMON/MNDCOM/ ANCPTY(NY)
COMMON/MNDCOM/ ANNEX(NY)
COMMON/MNDCOM/ BKRATE
COMMON/MNDCOM/ CAREW(NW,NY)
COMMON/MNDCOM/ CORDSC(12,NCR)
COMMON/MNDCOM/CORHT(NCR)
COMMON/MNDCOM/CORLEN(NCR,NW)
COMMON/MNDCOM/ CORVOL(NW,NY)
COMMON/MNDCOM/ CUMFLD(NY)
COMMON/MNDCOM/ DELRHT(NW)
COMMON/MNDCOM/ IHSTFL(NYRO)
COMMON/MNDCOM/NBYR
COMMON/MNDCOM/NCOR
COMMON/MNDCOM/NCORPP(NCR)
COMMON/MNDCOM/ NMMACH(NY)
COMMON/MNDCOM/ MSA
COMMON/MNDCOM/ MSHIFT(2)
COMMON/MNDCOM/ OFTADC
COMMON/MNDCOM/ OTBKF(NW,NYRO)
COMMON/MNDCOM/OTHBFL
COMMON/MNDCOM/OTBDEN
COMMON/MNDCOM/ PCTFIL
COMMON/MNDCOM/ PORMDF(2,NC)
COMMON/MNDCOM/ NEQRPL(NCR,NYRO)
COMMON/MNDCOM/ RKOFFS(NY)
COMMON/MNDCOM/ RKSURF(NYRO)
COMMON/MNDCOM/ RMHMIN(NW)
COMMON/MNDCOM/ RMHT(NW)
COMMON/MNDCOM/ RMMATR(NW,NY)
COMMON/MNDCOM/ RMNLBR(NW,NY)
COMMON/MNDCOM/ RMUVOL(NW)
COMMON/MNDCOM/ SRMUVL
COMMON/MNDCOM/ ROCST (NRO)
COMMON/MNDCOM/ ROLBRC(NRO)
COMMON/MNDCOM/ ROLBRR(NRO)

```

6JAN

```

COMMON/MNDCOM/ ROECST(NRO)
COMMON/MNDCOM/ ROEQPL(NRO)
COMMON/MNDCOM/ ROEQPC(NRO)
COMMON/MNDCOM/ ROMATC(NRO)
COMMON/MNDCOM/ ROMATU(NRO)
COMMON/MNDCOM/ RRCHAS(NYRO)
COMMON/MNDCOM/ RRCHST(NYRO)
COMMON/MNDCOM/ RRCMIN(NY)
COMMON/MNDCOM/ RRCOSD(NY)
COMMON/MNDCOM/ RUBKFL(NW)
COMMON/MNDCOM/ SCSKD(NC)
COMMON/MNDCOM/ SHRKMD(NW,NY)
COMMON/MNDCOM/ SOTBKF(NYRO)
COMMON/MNDCOM/ SRKBKF(NYRO)
COMMON/MNDCOM/ SRKHST(NYRO)
COMMON/MNDCOM/ SRM
COMMON/MNDCOM/ SRMM(NW)
COMMON/MNDCOM/ TCPCLN
COMMON/MNDCOM/ TONPAR
COMMON/MNDCOM/ TRCC(6)
COMMON/MNDCOM/ TREC (NRO)
COMMON/MNDCOM/ UBFRPY(NY)
COMMON/MNDCOM/ UBKVOL(NW)
COMMON/MNDCOM/ XPANFC
EQUIVALENCE(ZERO7 (1),ANFILD(1))

```

C
END

C
C
WAFCOM PROC
C

```

DIMENSION ZERO8 (IZ8)
COMMON/WAFCOM/ CFSPCT
COMMON/WAFCOM/ MODFLG(4)
COMMON/WAFCOM/ NPKSTM(NY)
COMMON/WAFCOM/ CHRECW(NW)
COMMON/WAFCOM/ RHPKG(NW)
COMMON/WAFCOM/ RHRECW(NW)
COMMON/WAFCOM/ SMNDWT(NW)
COMMON/WAFCOM/ WAFAL (NW)
COMMON/WAFCOM/ WAFCN (NW)
COMMON/WAFCOM/ WAFMET(NW,NEE)
COMMON/WAFCOM/ SUMMET(NW,NEE)
COMMON/WAFCOM/ SUMET(NEE)
COMMON/WAFCOM/ WAFMV (NW)
COMMON/WAFCOM/ WAFMVY(NW,NY)
COMMON/WAFCOM/ WAFNC (NW)
COMMON/WAFCOM/ WAFRHP(NW)
COMMON/WAFCOM/ WAFRSH(NW)

```

```

COMMON/WAFCOM/  WAFSHC(NW)
COMMON/WAFCOM/  WAFUGW(NW)
COMMON/WAFCOM/  WAFUTL(NW)
COMMON/WAFCOM/  WAFLBR(NW)
EQUIVALENCE(ZERO8 (1),CFSPCT)

```

END

C

C

SFTCOM PROC

C

```

DIMENSION ZERO9 (IZ9)
COMMON/SFTCOM/  CCON1
COMMON/SFTCOM/  CCON2
COMMON/SFTCOM/  IEQTP2
COMMON/SFTCOM/  CSTSHT(NC)
COMMON/SFTCOM/  MXMMRH
COMMON/SFTCOM/  MXMPRH
COMMON/SFTCOM/  NDSMP
COMMON/SFTCOM/  NRBPRM
COMMON/SFTCOM/  NCBPRM
COMMON/SFTCOM/  NPKPTR(NW)
COMMON/SFTCOM/  NPKPTC(NW)
COMMON/SFTCOM/  TNPT1
COMMON/SFTCOM/  TNPT2
COMMON/SFTCOM/  SCON1
COMMON/SFTCOM/  SCON2
COMMON/SFTCOM/  IEQTP1
COMMON/SFTCOM/  SFCTOT(6)
COMMON/SFTCOM/  SFTHE (NST)
COMMON/SFTCOM/  SFTLC (NST)
COMMON/SFTCOM/  SFTLD(NST)
COMMON/SFTCOM/  SFTSC (NST)
COMMON/SFTCOM/  SFTSS (NST)
COMMON/SFTCOM/  SFTWC(NST)
COMMON/SFTCOM/  SHDFC(NST)
COMMON/SFTCOM/  WCCON1
COMMON/SFTCOM/  WCCON2
COMMON/SFTCOM/  IEQTP3
EQUIVALENCE(ZERO9 (1),CCON1)

```

10DEC

10DEC

10DEC

C

END

C

C

PRECOM PROC

C

```

DIMENSION ZERO10 (IZ10)
COMMON/PRECOM/  AECON1
COMMON/PRECOM/  AECON2
COMMON/PRECOM/  IEQTP4

```

10DEC

```

COMMON/PRECOM/ AESERV(NW,NY)
COMMON/PRECOM/ CLCDF (NC)
COMMON/PRECOM/ CLNDF (NC)
COMMON/PRECOM/ COSTLD(NW,NC)
COMMON/PRECOM/ COSTLC(NW,NC)
COMMON/PRECOM/ CSTAE (NC)
COMMON/PRECOM/ CSTLND(NC)
COMMON/PRECOM/ CSTLIC(NC)
COMMON/PRECOM/ CSTUGF(NC)
COMMON/PRECOM/ CSTSU (NC)
COMMON/PRECOM/ FLDCST(NW,NC)
COMMON/PRECOM/ PRODC(NW,NY)
COMMON/PRECOM/ SCDF (NC)
COMMON/PRECOM/ STARTC(NW,NY)
COMMON/PRECOM/ STPCT
COMMON/PRECOM/ TCLAND
COMMON/PRECOM/ TCLICE
EQUIVALENCE(ZERO10 (1),AECON1)
C
END
C
PSTCOM PROC
C
DIMENSION ZERO11 (IZ11)
COMMON/PSTCOM/ DBDF(NC)
COMMON/PSTCOM/ DCOMCF
COMMON/PSTCOM/ ANDDCT(NYRO)
COMMON/PSTCOM/ ANSLCT(NYRO)
COMMON/PSTCOM/ BKFLBR(NW,NYRO)
COMMON/PSTCOM/ BKFMAT(NW,NYRO)
COMMON/PSTCOM/ BKSHFT(NYRO)
COMMON/PSTCOM/ NDDC
COMMON/PSTCOM/ DDFRAC
COMMON/PSTCOM/ DDCST (NDC)
COMMON/PSTCOM/ DDLBRC(NDC)
COMMON/PSTCOM/ DDLBRR(NDC)
COMMON/PSTCOM/ DDEQPC(NDC)
COMMON/PSTCOM/ DDMATC(NDC)
COMMON/PSTCOM/ DDTITL(12,NDC)
COMMON/PSTCOM/ DECOMC(NW,NYRO)
COMMON/PSTCOM/ FDDF(NC)
COMMON/PSTCOM/ ITSURV
COMMON/PSTCOM/ NBKF
COMMON/PSTCOM/ REMBKF(NW)
COMMON/PSTCOM/ RKHLBR(NW,NYRO)
COMMON/PSTCOM/ RKHMAT(NW,NYRO)
COMMON/PSTCOM/ SEALCT(NW,NYRO)
COMMON/PSTCOM/ SCCST
COMMON/PSTCOM/ SCLBRC

```

```

COMMON/PSTCOM/ SCLBRR
COMMON/PSTCOM/ SCEQPC
COMMON/PSTCOM/ SCMATC
COMMON/PSTCOM/ SCMATU
COMMON/PSTCOM/ SRHLBR(NW,NYRO)
COMMON/PSTCOM/ SRHMAT(NW,NYRO)
COMMON/PSTCOM/ SSDF(NC)
COMMON/PSTCOM/ SSSCON1
COMMON/PSTCOM/ SSSCON2
COMMON/PSTCOM/ IEQTP5
COMMON/PSTCOM/ SUMBK(NYRO)
COMMON/PSTCOM/ SURCST
COMMON/PSTCOM/ TSURVC(NYRO)
COMMON/PSTCOM/ SURVCT(NW,NYRO)
EQUIVALENCE(ZERO11 (1),DBDF(1))

```

10DEC

C
END

C
C
UTLCOM PROC
C

```

DIMENSION ZERO12 (IZ12)
COMMON/UTLCOM/ ACP
COMMON/UTLCOM/ ANUTC(NWP1,NY)
COMMON/UTLCOM/ DIESEL(NWP1,NY)
COMMON/UTLCOM/ DSLC
COMMON/UTLCOM/ DSLCST(NWP1,NY)
COMMON/UTLCOM/ DWTRC
COMMON/UTLCOM/ HEATS
COMMON/UTLCOM/ MROP
COMMON/UTLCOM/ ORS
COMMON/UTLCOM/ OWHP
COMMON/UTLCOM/ POWCST(NWP1,NY)
COMMON/UTLCOM/ POWER (NWP1,NY)
COMMON/UTLCOM/ RAPP
COMMON/UTLCOM/ RAPS
COMMON/UTLCOM/ RED
COMMON/UTLCOM/ RVP
COMMON/UTLCOM/ SFP
COMMON/UTLCOM/ SFSP
COMMON/UTLCOM/ STEAM (NWP1,NY)
COMMON/UTLCOM/ STMC
COMMON/UTLCOM/ STMCST(NWP1,NY)
COMMON/UTLCOM/ SUD
COMMON/UTLCOM/ UGD
COMMON/UTLCOM/ UPC
COMMON/UTLCOM/ VOLDW
COMMON/UTLCOM/ VOLW
COMMON/UTLCOM/ WATER (NWP1,NY)

```



```

COMMON/UTLCOM/  WATERC(NWP1,NY)
COMMON/UTLCOM/  WTRC
EQUIVALENCE(ZERO12 (1),ACP)

C
END
C
C
UOPCOM  PROC
C
  DIMENSION  ZERO13 (IZ13)
COMMON/UOPCOM/  BCKFTH(NW)
COMMON/UOPCOM/  CAPLEN(NW)
COMMON/UOPCOM/  CHMTEM(NW)
COMMON/UOPCOM/  EECST (NEO)
COMMON/UOPCOM/  EELBRC(NEO)
COMMON/UOPCOM/  EELBRR(NEO)
COMMON/UOPCOM/  EEEEST(NEO)
COMMON/UOPCOM/  EEEQPC(NEO)
COMMON/UOPCOM/  EEEQPL(NEO)
COMMON/UOPCOM/  EEMATC(NEO)
COMMON/UOPCOM/  EEMATU(NEO)
COMMON/UOPCOM/  HDCON1
COMMON/UOPCOM/  HDCON2
COMMON/UOPCOM/  NTYP
COMMON/UOPCOM/  HOLD(NW)
COMMON/UOPCOM/  HOLDEL(NW)
COMMON/UOPCOM/  HOLDIA(NW)
COMMON/UOPCOM/  HOLXC  (NW)
COMMON/UOPCOM/  PKGDIA(NW)
COMMON/UOPCOM/  PKGPRK(NW)
COMMON/UOPCOM/  RCKCST(NW)
COMMON/UOPCOM/  REFBKF(NW)
COMMON/UOPCOM/  REFDEP(NW)
COMMON/UOPCOM/  REFDIA(NW)
COMMON/UOPCOM/  REFMAT(NW)
COMMON/UOPCOM/  REFSLV(NW)
COMMON/UOPCOM/  SLEETH(NW)
COMMON/UOPCOM/  SLEVOR(NW)
COMMON/UOPCOM/  TEEEC (NEO)
COMMON/UOPCOM/  TUEC  (NUO)
COMMON/UOPCOM/  UOCST (NUO)
COMMON/UOPCOM/  UOLBRC(NUO)
COMMON/UOPCOM/  UOLBRR(NUO)
COMMON/UOPCOM/  UOECST(NUO)
COMMON/UOPCOM/  UOEQPL(NUO)
COMMON/UOPCOM/  UOEQPC(NUO)
COMMON/UOPCOM/  UOMATC(NUO,NW)
COMMON/UOPCOM/  UOMATU(NUO,NW)
COMMON/UOPCOM/  WABATH(NW)

```

```

COMMON/UOPCOM/ WABOR(NW)
EQUIVALENCE(ZERO13 (1),BCKFTH(1))
C
END
C
TABL PROC
    INTEGER TABLES
    COMMON /TABL/ TABLES (27)
END
C
C
COMENT PROC
C
C ABCC      = COST-ADMINISTRATION BUILDING, 10**6 $
C ACCC      = COST-AIR-CONDITIONING BUILDING, 10**6 $
C ACP       = AIR CONDITIONING POWER, KW
C ALC       = ANNUAL LABOR COST, $/YEAR
C AMMC      = ANNUAL MAINTENANCE MATERIALS COST, $/YEAR
C APC       = ANNUAL PACKING COST, $/YEAR
C BLDMNT    = LABOR+COST-MAINTENANCE BUILDING
C BKFLC     = BACKFILLING COST, $/CUBIC METER
C CFCON     = CASH FLOW - CONSTRUCTION
C CFOPR     = CASH FLOW - OPERATING
C CLSEMP    = CLOSEST EMPLACEMENT TO CORRIDOR, METERS
C CORLEN    = CORRIDOR LENGTH/ROOM, METERS/ROOM
C CPA       = CENTRAL PILLAR AREA, AREA,ACRES
C CRPPM     = RECEIPTS, WASTE UNITS PER PACKING MODULE PER SHIFT YEAR
C CRPRM     = RECEIPTS, CONTAINER UNITS PER CANISTERED WASTE RECEIVING MODULE
C           = PER SHIFT YEAR
C CRRAIL    = CONTAINER RECEIPTS BY RAIL, PERCENT
C CTCC      = COST-COOLING TOWER, 10**6 $
C DA        = TODAYS DATE
C DDWS      = CANISTERED WASTE SHAFT, DIAMETER METERS
C DELCHT    = DELTA CORRIDOR HEIGHT, METERS
C DELCWT    = DELTA CORRIDOR WIDTH, METERS
C DELRHT    = DELTA ROOM, METERS
C DENSTY    = ROCK DENSITY, TONNES/CUBIC METER
C DEPTH     = MINE DEPTH, METERS
C DSLC      = DIESEL-COST, $/L
C DWTRC     = DEMINERALIZED WATER-COST, CENTS/10**6 L
C EBCC      = COST-STORAGE AREA EXHAUST BUILDING, 10**6 $
C ECRMHR    = EQUIPMENT COST, RECEIVING MODULE, REMOTE HANDLED WASTE
C ECRMCH    = EQUIPMENT COST, RECEIVING MODULE, CONTACT HANDLED WASTE
C ECPM      = EQUIPMENT COST, PACKING MODULE
C ECMMS     = EQUIPMENT COST, MINED ROCK STORAGE
C ECWDB     = EQUIPMENT COST, WASTE OPERATIONS BUILDING
C ECADB     = EQUIPMENT COST, ADMINISTRATION BUILDING
C ECMVB     = EQUIPMENT COST, MINE VENTILATION BUILDING
C ECSAE     = EQUIPMENT COST, STORAGE AREAEXHAUST

```

C ECSP = EQUIPMENT COST, STEAM PLANT
 C ECAC = EQUIPMENT COST, AIR CONDITIONING BUILDING
 C ECFH = EQUIPMENT COST, FIREHOUSE
 C ECMB = EQUIPMENT COST, MAINTENANCE BUILDING
 C ECWH = EQUIPMENT COST, WAREHOUSE
 C ECST = EQUIPMENT COST, SEWAGE TREATMENT
 C ECROB = EQUIPMENT COST, RAIL OPERATIONS BUILDING
 C ECPAS = EQUIPMENT COST, POWER AND SUBSTATION
 C ECUGO = EQUIPMENT COST, UNDERGROUND OPERATIONS
 C EMMC = IMMOBILIZATION MATERIALS, \$/ PACKAGE
 C EMPCON = EMPLACEMENT CONFIGURATION, PACK/ROW
 C EPSUB = LABOR+COST-ELECTRIC POWER SUBSTATION
 C FILRL = FILLER MATERIAL COST, \$/PACKAGE
 C FHCC = COST-FIRE HOUSE, 10**6 \$
 C FIRHSE = LABOR+COST-FIRE HOUSE
 C FLUMM = FRACTION OF LABOR USED FOR MAINTENANCE MATERIALS
 C FXCSKD = FIXED MINING COST CONSTRUCTION SCHEDULE
 C FXMNC = FIXED MINING COST
 C HEATS = STEAM-HEATING, LBS/ACRE OF REPOSITORY
 C HOLMNT = HOLE LINER MATERIAL
 C HOLXC = HOLE EXCAVATION COST, \$/HOLE
 C HPMIN = MINIMUM HOLE PITCH, METERS
 C IRMCW = INCREMENTAL RECEIVING MODULE FOR CANISTERED WASTE
 C IRMNW = INCREMENTAL RECEIVING MODULE FOR NON-CANISTERED WASTE
 C IYDOL = DISCOUNT YEAR
 C IYRCON = YEAR IN WHICH CONSTRUCTION BEGINS
 C IYRPRO = YEAR IN WHICH PRODUCTION BEGINS
 C KIND = WASTE TYPE DEFINITION, 1 = CANISTERED, 2 = NON-CANISTERED
 C LY = LAST YEAR OF OPERATION (INDEX)
 C MBCC = COST-MAINTENANCE BUILDING, 10**6 \$
 C MMP = MINING MACHINERY POWER, KW/TONNE MINED
 C MROP = MINED ROCK OPERATIONS POWER, KW/ TONNE
 C MAS = MINED ROCK STORAGE AREA, ACRES
 C NFTL = NEARFIELD THERMAL LOAD, KW/ACRE
 C NMODCW = NUMBER OF RECEIVING MODULES FOR CANISTERED WASTE
 C NMODNW = NUMBER OF RECEIVING MODULES FOR NON-CANISTERED WASTE
 C NPAKM = NUMBER OF PACKAGING MODULES, CANISTERED WASTE
 C NWTP = NUMBER OF WASTE TYPES
 C OPCMIN = MINING COST TOTAL
 C OPMC = MINING COST ANNUAL
 C ORS = STEAM-OTHER USE, 10**3 LBS/YEAR
 C OTHRC = OTHER WASTE PACKAGE MATERIAL COST, \$/PACKAGE
 C OVRHDF = OVERHEAD FACTOR
 C OVRPC = OVERPACK COST, \$/PACKAGE
 C OVRTMF = OVERTIME FACTOR
 C OWHP = OTHER WASTE HANDLING BUILDING POWER, KW
 C P = WASTE INPUT CONDITION, 1 = PRE-PACKAGED, 2 = NON-PACKAGED
 C PFCST = PACKAGING FACILITY COST, \$/YEAR
 C PKGDS1 = PACKAGE NARRATIVE DESCRIPTION

C PKGDS2 = PACKAGE NARRATIVE DESCRIPTION
 C PKGDS3 = PACKAGE NARRATIVE DESCRIPTION
 C PKGLN = PACKAGE LENGTH, METERS
 C PLACE = LOCATION OF REPOSITORY
 C PMCC = COST-RECEIVING MODULE-PACKAGING 10**6 \$/MODULE
 C PRICC = PRIMARY CONTAINER COST, \$/PACKAGE
 C PWF = PRESENT WORTH FACTOR
 C RAPP = RECEIVING AND PACKAGING POWER, KW/MODULE
 C RAPS = STEAM-RECEIVING-PACKAGING, 10**3 LBS/YEAR
 C RCCAP = RAIL CONTAINER CAPACITY, WASTE UNITS
 C RCKHC = ROCK HOISTING COST
 C RCKHST = ROCK HOISTED, ANNUAL
 C RED = DIESEL-RAIL ENGINE, 10**3 L/CONTAINER
 C RHCC = COST-MINED MATERIAL HANDLING, 10**6 \$
 C RKBKF = ROCK BACKFILLED, TONNES
 C RKMSM = ANNUAL MINED ROCK MASS
 C RMCCC = COST-RECEIVING MODULE- CANISTERED 10**6 \$/MODULE
 C RMCSW1 = CANISTERED WASTE RECEIVING MODULE CONSTRUCTION SCHEDULE PRE-OP
 C RMCSW2 = CANISTERED WASTE RECEIVING MODULE CONSTRUCTION SCHEDULE POST-OP
 C RMLN = ROOM LENGTH, METERS
 C RMNCC = COST-RECEIVING MODULE-NON-CANISTERD 10**6 \$/MODULE
 C RMNSW1 = NON-CANISTERED WASTE RECEIVING MODULE CONSTRUCTION SCHEDULE PRE-OP
 C RMNSW2 = NON-CANISTERED WASTE RECEIVING MODULE CONSTRUCTION SCHEDULE PRE-OP
 C RMNSW2 = NON-CANISTERED WASTE RECEIVING MODULE CONSTRUCTION SCHEDULE POST-OP
 C RMWTH = ROOM WIDTH METERS
 C ROBCC = COST-RAIL OPERATION BUILDING, 10**6 \$
 C RPTYP = REPOSITORY DESCRIPTION
 C RSMIN = MINIMUM ROW SPACING
 C RVP = VENTILATION POWER, KW/ACRE OF REPOSITORY
 C RWSPRM = ROWS PER ROOM
 C SACC = COST-MINED MATERIAL STOCKPILE-PREPARATION 10**3 \$/ACRE
 C SCDAS = SUMMARY COST - DECOMMISSIONING AND SEALING
 C SCOSF = SUMMARY COST - OTHER SURFACE FACILITIES
 C SCPOP = SUMMARY COST - PRE-OPERATIONAL
 C SCSHF = SUMMARY COST - SHAFTS AND HEADFRAMES
 C SCSKD = SHAFT CONSTRUCTION SCHEDULE
 C SCSOP = SUMMARY COST - SURFACE OPERATING
 C SCTOT = SUMMARY COST - TOTALS
 C SCUGD = SUMMARY COST - UNDERGROUND MINE DEVELOPMENT
 C SCUOP = SUMMARY COST - UNDERGROUND OPERATING
 C SCUTL = SUMMARY COST - UTILITIES
 C SCWPF = SUMMARY COST - WASTE PACKAGING FACILITIES
 C SCWRF = SUMMARY COST - WASTE RECEIVING FACILITIES
 C
 C SFTDD(ST,RP) = SHAFT DESIGN DIAMETER, METERS
 C SFTXR(ST,RP) = INCREMENTAL EXCAVATION RADIUS, METERS
 C SFTXD(ST,RP) = EXCAVATION DEPTH, METERS
 C SFTLD(ST,RP) = LINED DEPTH, METERS
 C SFTSC(ST,RP) = SINKING COST, \$/METER

C SFTLC(ST,RP) = LINING COST, \$/METER
 C SFTHE(ST,RP) = HOISTING EQUIPMENT COST, \$
 C SFTNUM (ST,RP)
 C WHERE ST= 1 - MEN AND MATERIALS SHAFT
 C WHERE ST= 2 - MINING VENTILATION SHAFT
 C WHERE ST= 3 - CANISTERED WASTE SHAFT
 C WHERE ST= 4 - LOW LEVEL WASTE SHAFT
 C WHERE ST= 5 - DEVELOPMENT EXHAUST SHAFT
 C WHERE ST= 6 - STORAGE EXHAUST SHAFT
 C WHERE RP= REPOSITORY TYPE INDEX
 C
 C SFCS = SURFACE FACILITY COMPLETION SCHEDULE
 C SFP = SUPPORT FACILITY POWER, KW
 C SFSP = SPENT FUEL STORAGE POWER, KW
 C SIZE = REPOSITORY SIZE, ACRES
 C SOA = SITE OPERATIONS AREA, ACRES
 C SHCPY = SHAFT COST PER YEAR
 C SHDFC = SHAFT AND HEAD FRAME UNIT COST
 C SPCC = COST-STEAM PLANT, 10**6 \$
 C SPSCC = COST-STAND BY POWER SUBSTATION 10**6 \$
 C SPUC = COST-SITE PREPARATION, 10**3 \$/ACRE
 C STCC = COST-SEWAGE TREATMENT, 10**6 \$
 C STMC = STEAM-COST, CENTS/10**3 LBS/YEAR
 C SUD = DIESEL-SURFACE USE, 10**3 L/DAY
 C SURFOC = SURFACE FACILITY DISPOSAL COST
 C TCCAP = TRUCK CONTAINER CAPACITY, UNITS
 C TDWI = RECEIVING FACILITY, KW/WASTE UNIT
 C TEKNEK = MINING METHOD
 C TI = TODAYS TIME
 C TITLE = TITLE
 C TRMND = TOTAL ROCK MINED, TONNES
 C TRNHT = TRANSPORTER HEIGHT, METERS
 C TRNWTW = TRANSPORTER WIDTH, METERS
 C UGD = DIESEL-UNDERGROUND, 10**3 L/PACKAGE
 C UPC = POWER COST, CENTS/KWH
 C VARMC = VARIABLE MINING COST
 C VBCC = COST-MINED VENTILATION BUILDING, 10**6 \$
 C VOLDW = DEMINERALIZED WATER-VOLUME, 10**3 L/DAY
 C VOLW = WATER-VOLUME, 10**3 L/DAY
 C WAFAL = WASTE ALLOCATION FACTOR FOR ALL WASTES
 C WAFCN = WASTE ALLOCATION FACTOR FOR CANISTERED WASTE
 C WAFNC = WASTE ALLOCATION FACTOR FOR NON-CANISTERED WASTE
 C WAFMV = WASTE ALLOCATION FACTOR FOR MINED VOLUME
 C WCSC = COST-WATER CONTROL, DURING SINKING 10**6 \$/DIAMETER/SHAFT
 C WHCC = COST-WAREHOUSE, 10**6 \$
 C WOBCC = COST-WASTE OPERATION BUILDING, 10**6 \$
 C WSTYP = FACILITY PARAMETER DESCRIPTIONS
 C WTCC = COST-WATER TANKS, 10**6 \$
 C WTRC = WATER-COST, CENTS/10**6 L

```
C WUPP      = MAXIMUM WASTE UNITS/PACKAGE
C XRASHO    = ROOM /(ROOM + PILLAR)
```

```
C
END
```

```
C
INCLUDE PARAM
INCLUDE OUTCOM
INCLUDE INCOM
INCLUDE RECCOM
INCLUDE SOPCOM
INCLUDE SSTCOM
INCLUDE MNDCOM
INCLUDE SFTCOM
INCLUDE HUBCOM
INCLUDE PRECOM
INCLUDE PSTCOM
INCLUDE UTLCOM
INCLUDE UOPCOM
INCLUDE WAFCOM
```

```
C
1  CONTINUE
   CALL DIVSET(1000)
C  DO 1  I=1,IZ1
C  ZERO1 (I) = 0.0
C  1  CONTINUE
   DO 2  I=1,IZ2
   ZERO2 (I) = 0.0
   2  CONTINUE
   DO 3  I=1,IZ3
   ZERO3 (I) = 0.0
   3  CONTINUE
   DO 4  I=1,IZ4
   ZERO4 (I) = 0.0
   4  CONTINUE
   DO 5  I=1,IZ5
   ZERO5 (I) = 0.0
   5  CONTINUE
   DO 6  I=1,IZ6
   ZERO6 (I) = 0.0
   6  CONTINUE
   DO 7  I=1,IZ7
   ZERO7 (I) = 0.0
   7  CONTINUE
   DO 8  I=1,IZ8
   ZERO8 (I) = 0.0
   8  CONTINUE
   DO 9  I=1,IZ9
   ZERO9 (I) = 0.0
   9  CONTINUE
```

```

DO 10 I=1,IZ10
ZERO10 (I) = 0.0
10 CONTINUE
DO 11 I=1,IZ11
ZERO11 (I) = 0.0
11 CONTINUE
DO 12 I=1,IZ12
ZERO12 (I) = 0.0
12 CONTINUE
DO 13 I=1,IZ13
ZERO13 (I) = 0.0
13 CONTINUE
C
C
C
CALL DATE(DA)
CALL TIME(TI)
C
C
CALL INPUT
IDL=IYRCON-IYDOL
DO 25 I=1,NYRO
II = IDL + I
PWF(I)=1./((1.+DRATE)**II)
25 CONTINUE
C
C
CALL WASTOP
C WASTE OPERATIONS
C
C
DO 3000 I=1,LY
DO 3010 K=1,NWT
WOPLBR(K,I) = RECLBR(K,I) + PKGLBR(K,I) + TRNLBR(K,I) +EMPLBR(K,I)
1 + EEXLBR(K,I)
WOPLBR(NWP1,I) =WOPLBR(NWP1,I) + WOPLBR(K,I)
TRNSPT(NWP1,I) = TRNSPT(1,I) + TRNSPT(2,I)
EMPMCH(NWP1,I) = EMPMCH(NWP1,I) + EMPMCH(K,I)
EXMACH(NWP1,I) = EXMACH(1,I) + EXMACH(2,I)
3010 CONTINUE
3000 CONTINUE
DO 3200 I=1,LY
DO 3200 K=1,NWT
RECST(K,I) = RELBCT(K,I) + RCMATC(K,I) + RCEQPC(K,I)
PACST(K,I) = PALBCT(K,I) + PKMATC(K,I) + PKEQPC(K,I)
TRCST(K,I) = TRLBCT(K,I) + TRMATC(K,I) + TREQPC(K,I)
EMCST(K,I) = EMLBCT(K,I) + EMMATC(K,I) + EMEQPC(K,I)
ECOST(K,I) = EXLBCT(K,I) + EXMATC(K,I) + EXEQPC(K,I)
TWOPCT(K,I) = RECST(K,I) + PACST(K,I) + TRCST(K,I) + EMCST(K,I)

```

```

1 + ECOST(K,I)
RECST(NWP1,I) = RECST(NWP1,I) + RECST(K,I)
PACST(NWP1,I) = PACST(NWP1,I) + PACST(K,I)
TRCST(NWP1,I) = TRCST(NWP1,I) + TRCST(K,I)
EMCST(NWP1,I) = EMCST(NWP1,I) + EMCST(K,I)
ECOST(NWP1,I) = ECOST(NWP1,I) + ECOST(K,I)
3200 CONTINUE
C
C MINE DEVELOPMENT
C
C CALL MINDEV
C
C CALL RECEVE
C
C RECEIVING
C
DO 405 I=1,NY
IRMCW(I) = 0
IRMNW(I) = 0
DO 405 K=1,NWT
PFCST(K,I) = 0.0
RFCST(K,I) = 0.0
405 CONTINUE
C
C
LLP1 =LL + 1
DO 100 K = 1,NWT
DO 75 I=LLP1,LY
IF(KIND(K).EQ.2) GO TO 45
IRMCW(I)=NMRH(I)- NMRH(I-1)
IF(IRMCW(I).LE.0) GO TO 40
III = 0
IB = I - LL
IE = I - 1
DO 30 II = IB,IE
III = III + 1
SKED = RMCWS1(III)
IF(I.GT.LL) SKED = RMCWS2(III)
RFCST(K,II) = IRMCW(I) * WOCST(1) * SKED * WAFCN(K)
+ + RFCST(K,II)
30 CONTINUE
40 CONTINUE
IPMCW(I) = NPMRH(I) - NPMRH(I-1)
IF(IPMCW(I).LE.0) GO TO 75
III= 0
IB = I - LL
IE = I - 1
DO 42 II = IB,IE

```



```

      III = III + 1
      SKED = RMCWS1(III)
      IF(I.GT.LL) SKED = RMCWS2(III)
      PFCST(K,II) = IPMCW(I) * WOCST(3) * SKED * WAFCN(K)
+ + PFCST(K,II)
42 CONTINUE
   GO TO 75
45 CONTINUE
   IRMNW(I)=NRMCH(I)- NRMCH(I-1)
   IF(IRMNW(I).LE.0) GO TO 75
   III = 0
   IB = I - LL
   IE = I - 1
   DO 50 II= IB,IE
   III = III + 1
   SKED = RMNWS1 (III)
   IF(I.GT.LL) SKED = RMNWS2(III)
   RFCST(K,II) = RFCST(K,II) + IRMNW(I) * WOCST(2)*SKED*WAFNC(K)
50 CONTINUE
75 CONTINUE
100 CONTINUE
C
C
C
C SHAFT CONSTRUCTION
C
C   CALL SHAFT
C
C   DO 750 I=1,LL
C   CSTSHT(I) =0.
C   DO 750 K=1,NW
C   SHCPY(K,I) = 0.0
750 CONTINUE
C   DO 770 K=1,NWT
C   DO 770 L=1,NST
C   DO 770 I=1,LL
C   WAF = WAFMV(K)
C   IF(L.EQ.1) WAF = WAFMV(K)
C   IF(L.EQ.3) WAF = WAFRHP(K)
C   IF(L.EQ.3.AND.SFTNUM(4).EQ.0) WAF = WAFUGW(K)
C   IF(L.EQ.4) WAF = WAFNC(K)
C   SHCPY(K,I)=SHCPY(K,I)+SFTNUM(L)*SHDFC(L)*SCSKD(I)*WAF
770 CONTINUE
C   DO 780 K=1,NWT
C   DO 780 I=1,LL
C   CSTSHT(I) = SHCPY(K,I) + CSTSHT(I)
780 CONTINUE
C
C   CALL ROCKOP

```

```

C
C ROCK OPERATIONS COST CALCULATIONS
C
    LE = NBKF + IDD
C
    DO 705 K = 1,NWT
    DO 705 I = 1,LE
    SUMBKF(I) = SUMBKF(I)+RKBKF(K,I)+OTBKF(K,I)
705  CONTINUE
    IF(MSWTCH.EQ.0) GO TO 710
    NRON = 1
    DO 690 I=1,LY
    DO 690 K=1,NWT
C
C
C
    TMINCT = TMINCT + RMNLBR(K,I)*OVRHDF * ROLBRC(NRON) + RMMATR(K,I)
    1 * ROMATC(NRON) + NEQRPL(NRON,I) * ROECST(NRON)
    1 * WAFMV(K)
    TRKMND = TRKMND + RKMSM(K,I)
690  CONTINUE
C
    TMINCT = TMINCT + ROCST(NRON)
C ROCK HOISTING
C
    NRON = 2
    DO 695 I=1,LE
    DO 695 K=1,NWT
    TRHCT = TRHCT + RKHLBR(K,I) *OVRHDF* ROLBRC(NRON) + RKHMAT(K,I)
    1 * ROMATC(NRON) + NEQRPL(NRON,I) * ROECST(NRON) * WAFMV(K)
    TRHSTD = TRHSTD + ABS(RCKHST(K,I) - SHRKMD(K,I))
    SHROCK = SHROCK + SHRKMD(K,I)
695  CONTINUE
    TRHCT = TRHCT + ROCST(NRON)
C
C SURFACE ROCK HANDLING
C
    NRON = 3
    DO 700 I=1,LE
    DO 700 K=1,NWT
    TRSHCT = TRSHCT + SRHLBR(K,I)*OVRHDF * ROLBRC(NRON) + SRHMAT(K,I)
    1 * ROMATC(NRON) + NEQRPL(NRON,I) * ROECST(NRON) * WAFMV(K)
700  CONTINUE
    TRSHCT = TRSHCT + ROCST(NRON)
    NRON = 5
    DO 702 I=1,LE
    DO 701 K=1,NWT
C
C BACKFILLING

```

```

C      TBKCT = TBKCT + BKFLBR(K,I) * OVRHDF * ROLBRC(NRON) + BKFMAT(K,I)
1      * ROMATC(NRON) + NEQRPL(NRON,I) * ROECST(NRON) * WAFMV(K)
701  CONTINUE
      TRBKF = TRBKF + SUMBKF(I)
702  CONTINUE
      TBKCT = TBKCT + ROCST(NRON)
      VARMC = TMINCT / TRKMND
      RCKHC = TRHCT / TRHSTD
      SURFDC = TRSHCT / (TRHSTD + SHROCK)
      BKFLC = TBKCT / TRBKF
710  CONTINUE
      DO 730 K=1,NWT
      DO 720 I=1,LY
      OPMC(K,I) = VARMC * RKMSM(K,I)
      CSTHST(K,I) = RCKHC * ABS(RCKHST(K,I)-SHRKMD(K,I))
      CSTSHS(K,I) = SURFDC * ABS(RCKHST(K,I))
      CSTBKF(K,I) = BKFLC * (RKBKF(K,I) + OTBKF(K,I))
      CSTOSD(K,I) = OFTADC * RKOFFS(I) * WAFMV(K)
      RRCMIN(I) = RRCMIN(I) + OPMC(K,I)
      RRCHST(I) = RRCHST(I) + CSTHST(K,I)
      RRCHAS(I) = RRCHAS(I) + CSTSHS(K,I)
      ANBFCT(I) = ANBFCT(I) + CSTBKF(K,I)
      RRCOSD(I) = RRCOSD(I) + CSTOSD(K,I)
      RKMSM(K,I) = RKMSM(K,I) + SHRKMD(K,I) + EEXRKM(K,I)
      SRKMND(I) = SRKMND(I) + EEXRKM(K,I) + SHRKMD(K,I)
720  CONTINUE
      LB = LY + 1
      DO 725 I=LB,LE
      CSTHST(K,I) = RCKHC * ABS(RCKHST(K,I))
      CSTSHS(K,I) = SURFDC * ABS(RCKHST(K,I))
      CSTBKF(K,I) = BKFLC * (RKBKF(K,I) + OTBKF(K,I))
      RRCHST(I) = RRCHST(I) + CSTHST(K,I)
      RRCHAS(I) = RRCHAS(I) + CSTSHS(K,I)
      ANBFCT(I) = ANBFCT(I) + CSTBKF(K,I)
725  CONTINUE
730  CONTINUE

C
C      CALL SURFST
C
C      UTILITY COSTS
C
C      CALL UTILITY
C
C
C
C      SURFACE SITE

```

```

C
DO 3100 I=1,LY
  TWOEQC(I) = REEQCT(I) + PARQCT(I) + TREQCT(I) + EMEQCT(I) +
1 EXEQCT(I) + SFEQCT(I)
  TALBCT(I) = ARLBCT(I) + APLBCT(I) + ATLBCT(I) + AMLBCT(I) + AXLBCT(I)
+ + OWLBR(I)
3100 CONTINUE
C
C
C SUM SURFACE OPERATIONS
C
DO 652 I=1,LL
DO 652 K=1,NWT
OSFC(K,I) = SFCAPC(K,I)
UGFC(K,I) = TRCAPC(K,I) + UFCAPC(K,I) + EECAPC(K,I)
652 CONTINUE
DO 655 I=1,LY
DO 655 K=1,NWT
OSFC(K,I) = RFCST(K,I) + PFCST(K,I) + OSFC(K,I)
TTCAPC = TTCAPC + TRCAPC(K,I)
TECAPC = TECAPC + EECAPC(K,I)
TUCAPC = TUCAPC + UFCAPC(K,I)
SFCCST(K) = SFCCST(K) + OSFC(K,I)
CSTS(I) = CSTS(I) + OSFC(K,I)
OWOPC(K,I) = OWOPC(K,I) + ANUTC(K,I) + SFEQPC(K,I)
OWOPC(NWP1,I) = OWOPC(NWP1,I) + OWOPC(K,I)
TWOPCT(K,I) = TWOPCT(K,I) + OWOPC(K,I)
TWOPCT(NWP1,I) = TWOPCT(NWP1,I) + TWOPCT(K,I)
655 CONTINUE
DO 660 K=1,NWT
DO 660 I=1,LL
CSTUGF(I) = CSTUGF(I) + UGFC(K,I)
660 CONTINUE
C
C
C
C PRE-PRODUCTION
C
CALL PREOP
C
C
C POST-PRODUCTION
C
CALL POSTOP
C
C CALCULATE EQUIPMENT COST SUMMARIES
C
DO 670 K=1,NWT
DO 670 I=1,LY

```

```

        IF(KIND(K).EQ.2) GO TO 665
        TWEC(1) = TWEC(1) + RCEQPC(K,I)
        TTEC(1) = TTEC(1) + TREQPC(K,I)
        TUEC(1) = TUEC(1) + EMEQPC(K,I)
        GO TO 666
665 CONTINUE
        TWEC(2) = TWEC(2) + RCEQPC(K,I)
        TTEC(2) = TTEC(2) + TREQPC(K,I)
        TUEC(2) = TUEC(2) + EMEQPC(K,I)
666 CONTINUE
        TWEC(3) = TWEC(3) + PKEQPC(K,I)
        TREC(1) = TREC(1) + NEQRPL(1,I) * ROECST(1) * WAFMV(K)
        TEQPT = TEQPT+RCEQPC(K,I)+TREQPC(K,I)+EMEQPC(K,I)+PKEQPC(K,I) +
1 NEQRPL(1,I) * ROECST(1) * WAFMV(K)
670 CONTINUE
        DO 680 K=1,NWT
        DO 680 I=1,LE
C
        DO 680 NRON=2,NRO
        TREC(NRON) = TREC(NRON) + NEQRPL(NRON,I) * ROECST(NRON) * WAFMV(K)
        TEQPT = TEQPT + NEQRPL(NRON,I) * ROECST(NRON) * WAFMV(K)
680 CONTINUE
        DO 685 K=1,NWT
        DO 685 I=1,LY
        L = IMET(K)
        IF(IMET(K).GE.4) L=1
        TEEEC(L) =TEEEEC(L) + EQREEX(K,I) * EEEECST(L)
        TEQPT = TEQPT + EQREEX(K,I) * EEEECST(L)
685 CONTINUE
C
C   CONTINGENCY
C
C   CALCULATE CONTINGENCY COSTS FOR PREOPERATIONS
C
        N = 1
        DO 2000 I=1,LL
        DO 2000 K=1,NWT
        CONTGY(K,I,N) = CONFAC(N) * (COSTLD(K,I) + COSTLC(K,I)
1 + STARTC(K,I) + AESERV(K,I) + ASPC(K,I) + SHCPY(K,I)
2 + OSFC(K,I) + UGFC(K,I))
2000 CONTINUE
C
        DO 2005 I=LL+1,LY
        DO 2005 K=1,NWT
        CONTGY(K,I,N) = CONTGY(K,I,N) + OSFC(K,I) * CONFAC(N)
2005 CONTINUE
C
C   CALCULATE CONTINGENCY COSTS FOR ROCK OPERATIONS

```

```

C
DO 2010 I=1,LY
DO 2010 K =1,NWT
N = 2
CONTGY(K,I,N) = CONFAC(N) * (OPMC(K,I) + CSTHST(K,I)
1 + CSTSHS(K,I) + CSTOSD(K,I))
C
C
C    CALCULATE CONTINGENCY FOR WASTE OPERATIONS
C
N=3
CONTGY(K,I,N) = CONFAC(N) * (RECST(K,I) + PACST(K,I) +TRCST(K,I)
1 + EMCST(K,I) + ECOST(K,I) + OWOPC(K,I))
C
C
C    CALCULATE CONTINGENCY FOR BACKFILLING AND DECOMMISSIONING OPERATIO
C
N = 4
CONTGY(K,I,N)=CONFAC(N) * CSTBKF(K,I)
DO 2010 N=1,4
ACONTG(I) = ACONTG(I) + CONTGY(K,I,N)
2010 CONTINUE
LB= LY+1
DO 2015 I=LB,LE
DO 2015 K=1,NWT
N=2
CONTGY(K,I,N) = CONFAC(N) *(CSTHST(K,I) +CSTSHS(K,I))
N=4
CONTGY(K,I,N) = CONFAC(N) * (CSTBKF(K,I) + SEALCT(K,I)
1 + DECOMC(K,I) + SURVCT(K,I))
ACONTG(I) = CONTGY(K,I,2) + CONTGY(K,I,4)
1 + ACONTG(I)
2015 CONTINUE
C
C
C    CALCULATE COST SUMMARIES BY WASTE TYPE
C    PRE-OPERATIONS
C
DO 2110 K=1,NWT
DO 2100 I=1,LL
TLDCST(K) = TLDCST(K) + COSTLD(K,I)
TLCCST(K) = TLCCST(K) + COSTLC(K,I)
TSPCST(K) = TSPCST(K) + ASPC(K,I)
TSHCST(K) = TSHCST(K) + SHCPY(K,I)
TSFCST(K) = TSFCST(K) + OSFC(K,I)
TUFCST(K) = TUFCST(K) + UGFC(K,I)
TAECST(K) = TAECST(K) + AESERV(K,I)
TSUCST(K) = TSUCST(K) + STARTC(K,I)
2100 CONTINUE
DO 2105 I = LL+1,LY
TSFCST(K) = TSFCST(K)+OSFC(K,I)
2105 CONTINUE

```

```

    TSUBCN(K) = TLDCST(K) + TLCCST(K) + TSPCST(K) + TSHCST(K)
1 + TSFCST(K) + TUFCST(K) + TAECST(K) + TSUCST(K)
    TLDCST(NWP1) = TLDCST(NWP1) + TLDCST(K)
    TLCCST(NWP1) = TLCCST(NWP1) + TLCCST(K)
    TSPCST(NWP1) = TSPCST(NWP1) + TSPCST(K)
    TSHCST(NWP1) = TSHCST(NWP1) + TSHCST(K)
    TSFCST(NWP1) = TSFCST(NWP1) + TSFCST(K)
    TUFCST(NWP1) = TUFCST(NWP1) + TUFCST(K)
    TAECST(NWP1) = TAECST(NWP1) + TAECST(K)
    TSUCST(NWP1) = TSUCST(NWP1) + TSUCST(K)
    TSUBCN(NWP1) = TSUBCN(NWP1) + TSUBCN(K)
2110 CONTINUE
C
C    ROCK OPERATIONS
C
    DO 2130 K=1,NWT
    DO 2120 I=1,LE
    TOMCST(K) = TOMCST(K) + OPMC(K,I)
    THSCST(K) = THSCST(K) + CSTHST(K,I)
    TSSCST(K) = TSSCST(K) + CSTSHS(K,I)
    TODCST(K) = TODCST(K) + CSTOSD(K,I)
2120 CONTINUE
    TSUBRO(K) = TOMCST(K) + THSCST(K)+TSSCST(K)+TODCST(K)
    TOMCST(NWP1) = TOMCST(NWP1) + TOMCST(K)
    THSCST(NWP1) = THSCST(NWP1) + THSCST(K)
    TSSCST(NWP1) = TSSCST(NWP1) + TSSCST(K)
    TODCST(NWP1) = TODCST(NWP1) + TODCST(K)
    TSUBRO(NWP1) = TSUBRO(NWP1) + TSUBRO(K)
2130 CONTINUE
C
C    WASTE OPERATIONS
C
    DO 2150 K=1,NWT
    DO 2140 I=1,LY
    TRCCST(K) = TRCCST(K) + RECST(K,I)
    TPKCST(K) = TPKCST(K) + PACST(K,I)
    TTRCST(K) = TTRCST(K) + TRCST(K,I)
    TEMCST(K) = TEMCST(K) + EMCST(K,I)
    TEXTCST(K) = TEXTCST(K) + ECOST(K,I)
    TUTCST(K) = TUTCST(K) + OWOPC(K,I)
2140 CONTINUE
    TOPCST(K) = TRCCST(K) + TPKCST(K) + TTRCST(K) + TEMCST(K)
1 + TEXTCST(K) + TUTCST(K)
    TRCCST(NWP1) = TRCCST(NWP1) + TRCCST(K)
    TPKCST(NWP1) = TPKCST(NWP1) + TPKCST(K)
    TTRCST(NWP1) = TTRCST(NWP1) + TTRCST(K)
    TEMCST(NWP1) = TEMCST(NWP1) + TEMCST(K)
    TEXTCST(NWP1) = TEXTCST(NWP1) + TEXTCST(K)
    TUTCST(NWP1) = TUTCST(NWP1) + TUTCST(K)

```

```

TOPCST(NWP1) = TOPCST(NWP1) + TOPCST(K)
2150 CONTINUE
C
C POST OPERATIONS
C
DO 2170 K=1,NWT
DO 2160 I=1,LE
TBFCST(K) = TBFCST(K) + CSTBKF(K,I)
TSLCST(K) = TSLCST(K) + SEALCT(K,I)
TDCCST(K) = TDCCST(K) + DECOMC(K,I)
TSVCST(K) = TSVCST(K) + SURVCT(K,I)
2160 CONTINUE
TBDCST(K) = TBFCST(K)+TSLCST(K)+TDCCST(K)+TSVCST(K)
TBFCST(NWP1) = TBFCST(NWP1) + TBFCST(K)
TSLCST(NWP1) = TSLCST(NWP1) + TSLCST(K)
TDCCST(NWP1) = TDCCST(NWP1) + TDCCST(K)
TSVCST(NWP1) = TSVCST(NWP1) + TSVCST(K)
TBDCST(NWP1) = TBDCST(NWP1) + TBDCST(K)
2170 CONTINUE
DO 2200 K=1,NWT
DO 2200 I= 1,LE
DO 2200 N= 1,4
TCNCST(K) = TCNCST(K) + CONTGY(K,I,N)
2200 CONTINUE
DO 2205 K=1,NWT
TCNCST(NWP1) = TCNCST(NWP1) + TCNCST(K)
2205 CONTINUE
DO 2210 K=1,NWT
TWTCTST(K) = TSUBCN(K) + TSUBRO(K) + TOPCST(K) + TBDCST(K)
1 + TCNCST(K)
TWTCTST(NWP1) = TWTCTST(NWP1) + TWTCTST(K)
2210 CONTINUE
C
DO 2215 K=1,NWP1
DO 2215 L=1,3
SLCU(L,K) = 0.0
2215 CONTINUE
C
LE=NBKF+IDD
DO 2220 K=1,NWT
DO 2220 I=1,LY
SLCU(1,K)=SLCU(1,K)+PWF(I)*RECUNT(K,I)*UNTMAS(K)
SLCU(2,K)=SLCU(2,K)+PWF(I)* WSTVOL(K,I)
SLCU(3,K)=SLCU(3,K)+PWF(I)*RECUNT(K,I)/WUPP(K)
SLCU(1,NWP1)=SLCU(1,NWP1)+PWF(I)*RECUNT(K,I)*UNTMAS(K)
SLCU(2,NWP1)=SLCU(2,NWP1)+PWF(I)*WSTVOL(K,I)
SLCU(3,NWP1)=SLCU(3,NWP1)+PWF(I)*RECUNT(K,I)/WUPP(K)
2220 CONTINUE
DO 2250 L=1,3

```



```

DO 2250 K=1,NWT
DO 2230 I=1,LL
  ULDCST(K,L) = ULDCST(K,L) + COSTLD(K,I) * PWF(I)
  ULCCST(K,L) = ULCCST(K,L) + COSTLC(K,I) * PWF(I)
  USPCST(K,L) = USPCST(K,L) + ASPC(K,I) * PWF(I)
  USHCST(K,L) = USHCST(K,L) + SHCPY(K,I) * PWF(I)
  USFCST(K,L) = USFCST(K,L) + OSFC(K,I) * PWF(I)
  UUFCST(K,L) = UUFCST(K,L) + UGFC(K,I) * PWF(I)
  UAECST(K,L) = UAECST(K,L) + AESERV(K,I) * PWF(I)
  USUCST(K,L) = USUCST(K,L) + STARTC(K,I) * PWF(I)
2230 CONTINUE
DO 2235 I=LL+1,LY
  USFCST(K,L) = USFCST(K,L) + OSFC(K,I) * PWF(I)
2235 CONTINUE
DO 2240 I=1,LY
  URCCST(K,L) = URCCST(K,L) + RECST(K,I) * PWF(I)
  UPKCST(K,L) = UPKCST(K,L) + PACST(K,I) * PWF(I)
  UTRCST(K,L) = UTRCST(K,L) + TRCST(K,I) * PWF(I)
  UEMCST(K,L) = UEMCST(K,L) + EMCST(K,I) * PWF(I)
  UEXCST(K,L) = UEXCST(K,L) + ECOST(K,I) * PWF(I)
  UUTCST(K,L) = UUTCST(K,L) + OWOPC(K,I) * PWF(I)
2240 CONTINUE
DO 2250 I= 1,LE
  UOMCST(K,L) = UOMCST(K,L) + OPMC(K,I) * PWF(I)
  UHSCST(K,L) = UHSCST(K,L) + Csthst(K,I) * PWF(I)
  USSCST(K,L) = USSCST(K,L) + CSTSHS(K,I) * PWF(I)
  UODCST(K,L) = UODCST(K,L) + CSTOSD(K,I) * PWF(I)
  UBFCT(K,L) = UBFCT(K,L) + CSTBKF(K,I) * PWF(I)
  USLCST(K,L) = USLCST(K,L) + SEALCT(K,I) * PWF(I)
  UDCCST(K,L) = UDCCST(K,L) + DECOMC(K,I) * PWF(I)
  USVCST(K,L) = USVCST(K,L) + SURVCT(K,I) * PWF(I)
DO 2250 N=1,4
  UCNCST(K,L) = UCNCST(K,L) + CONTGY(K,I,N) * PWF(I)
2250 CONTINUE
DO 2260 L=1,3
DO 2260 K=1,NWT
  ULDCST(NWP1,L) = ULDCST(NWP1,L) + ULDCST(K,L)
  ULCCST(NWP1,L) = ULCCST(NWP1,L) + ULCCST(K,L)
  USPCST(NWP1,L) = USPCST(NWP1,L) + USPCST(K,L)
  USHCST(NWP1,L) = USHCST(NWP1,L) + USHCST(K,L)
  USFCST(NWP1,L) = USFCST(NWP1,L) + USFCST(K,L)
  UUFCST(NWP1,L) = UUFCST(NWP1,L) + UUFCST(K,L)
  UAECST(NWP1,L) = UAECST(NWP1,L) + UAECST(K,L)
  USUCST(NWP1,L) = USUCST(NWP1,L) + USUCST(K,L)
  UOMCST(NWP1,L) = UOMCST(NWP1,L) + UOMCST(K,L)
  UHSCST(NWP1,L) = UHSCST(NWP1,L) + UHSCST(K,L)
  USSCST(NWP1,L) = USSCST(NWP1,L) + USSCST(K,L)
  UODCST(NWP1,L) = UODCST(NWP1,L) + UODCST(K,L)
  URCCST(NWP1,L) = URCCST(NWP1,L) + URCCST(K,L)

```

```

UPKCST(NWP1,L) = UPKCST(NWP1,L) + UPKCST(K,L)
UTRCST(NWP1,L) = UTRCST(NWP1,L) + UTRCST(K,L)
UEMCST(NWP1,L) = UEMCST(NWP1,L) + UEMCST(K,L)
UEXCST(NWP1,L) = UEXCST(NWP1,L) + UEXCST(K,L)
UUTCST(NWP1,L) = UUTCST(NWP1,L) + UUTCST(K,L)
UBFCST(NWP1,L) = UBFCST(NWP1,L) + UBFCST(K,L)
USLCST(NWP1,L) = USLCST(NWP1,L) + USLCST(K,L)
UDCCST(NWP1,L) = UDCCST(NWP1,L) + UDCCST(K,L)
USVCST(NWP1,L) = USVCST(NWP1,L) + USVCST(K,L)
UCNCST(NWP1,L) = UCNCST(NWP1,L) + UCNCST(K,L)

```

2260 CONTINUE

```
DO 2270 L=1,3
```

```
DO 2270 K=1,NWP1
```

```
IF(SLCU(L,K).EQ.0.) GO TO 2350
```

```

ULDCST(K,L) = ULDCST(K,L) / SLCU(L,K)
ULCCST(K,L) = ULCCST(K,L) / SLCU(L,K)
USPCST(K,L) = USPCST(K,L) / SLCU(L,K)
USHCST(K,L) = USHCST(K,L) / SLCU(L,K)
USFCST(K,L) = USFCST(K,L) / SLCU(L,K)
UUF CST(K,L) = UUF CST(K,L) / SLCU(L,K)
UAECST(K,L) = UAECST(K,L) / SLCU(L,K)
USUCST(K,L) = USUCST(K,L) / SLCU(L,K)
UOMCST(K,L) = UOMCST(K,L) / SLCU(L,K)
UHSCST(K,L) = UHSCST(K,L) / SLCU(L,K)
USSCST(K,L) = USSCST(K,L) / SLCU(L,K)
UODCST(K,L) = UODCST(K,L) / SLCU(L,K)
URCCST(K,L) = URCCST(K,L) / SLCU(L,K)
UPKCST(K,L) = UPKCST(K,L) / SLCU(L,K)
UTRCST(K,L) = UTRCST(K,L) / SLCU(L,K)
UEMCST(K,L) = UEMCST(K,L) / SLCU(L,K)
UEXCST(K,L) = UEXCST(K,L) / SLCU(L,K)
UUTCST(K,L) = UUTCST(K,L) / SLCU(L,K)
UBFCST(K,L) = UBFCST(K,L) / SLCU(L,K)
USLCST(K,L) = USLCST(K,L) / SLCU(L,K)
UDCCST(K,L) = UDCCST(K,L) / SLCU(L,K)
USVCST(K,L) = USVCST(K,L) / SLCU(L,K)
UCNCST(K,L) = UCNCST(K,L) / SLCU(L,K)

```

```
GO TO 2270
```

2350 CONTINUE

```

ULDCST(K,L) = 0.
ULCCST(K,L) = 0.
USPCST(K,L) = 0.
USHCST(K,L) = 0.
USFCST(K,L) = 0.
UUF CST(K,L) = 0.
UAECST(K,L) = 0.
USUCST(K,L) = 0.
UOMCST(K,L) = 0.
UHSCST(K,L) = 0.

```

```

USSCST(K,L) = 0.
UODCST(K,L) = 0.
URCCST(K,L) = 0.
UPKCST(K,L) = 0.
UTRCST(K,L) = 0.
UEMCST(K,L) = 0.
UEXCST(K,L) = 0.
UUTCST(K,L) = 0.
UBFCST(K,L) = 0.
USLCST(K,L) = 0.
UDCCST(K,L) = 0.
USVCST(K,L) = 0.
UCNCST(K,L) = 0.
2270 CONTINUE
DO 2280 L=1,3
DO 2280 K=1,NWP1
USUBCN(K,L) = ULDCST(K,L) + ULCCST(K,L) +USPCST(K,L) +USHCST(K,L)
1 + USFCST(K,L) + UUFCST(K,L) + UAECST(K,L) + USUCST(K,L)
USUBRO(K,L) = UOMCST(K,L) + UHSCST(K,L) +USSCST(K,L)+UODCST(K,L)
UOPCST(K,L) = URCCST(K,L) + UPKCST(K,L) + UTRCST(K,L) +UEMCST(K,L)
1 + UEXCST(K,L) + UUTCST(K,L)
UBDCST(K,L) = UBFCST(K,L) + USLCST(K,L)+ UDCCST(K,L) + USVCST(K,L)
UWTCST(K,L) = USUBCN(K,L) + USUBRO(K,L) + UOPCST(K,L)+ UBDCST(K,L)
1 + UCNCST(K,L)
2280 CONTINUE
C
C
DO 2450 K=1,NWT
DO 2450 I=1,NY
RECUNT(NWP1,I) = RECUNT(NWP1,I) + RECUNT(K,I)
NUMPKG(NWP1,I) = NUMPKG(NWP1,I) + NUMPKG(K,I)
RMFILD(NWP1,I) = RMFILD(NWP1,I) + RMFILD(K,I)
2450 CONTINUE
C
DO 2460 K=1,NWT
DO 2460 I=LL+1,LY
CLABOR(3) = CLABOR(3) + RMNLBR(K,I) * OVRHDF * ROLBRC(1)
CMATL(3) = CMATL(3) + RMMATR(K,I) * ROMATC(1)
2460 CONTINUE
C
C
DO 2400 K=1,NWT
DO 2410 I=1,LL
SPMASS = SPMASS + CPCMM(K,I)
CMASS = CMASS + CORMAS(K,I)
PSCMAS = PSCMAS + CORVOL(K,I) * DENSTY
PRMMAS = PRMMAS + RMFILD(K,I+LEAD) * RKVOL(K) * DENSTY
2410 CONTINUE
DO 2420 I=1,LY

```

```

      HDMASS = HDMASS + EEXRKM(K,I)
2420  CONTINUE
2400  CONTINUE
      DO 2430 K=1,NWT
      SFMASS = SFMASS + SHAFTX(K)
2430  CONTINUE
      DO 2471 K=1,NWT
      DO 2470 I=1,LY
      CPMAT(K) = CPMAT(K) + PKMATC(K,I) / ONEMIL
      CPEQ(K) = CPEQ(K) + PKEQPC(K,I)/ ONEMIL
      CPLBR(K) = CPLBR(K) + PALBCT(K,I) / ONEMIL
2470  CONTINUE
2471  CONTINUE
C
      DO 2480 I=LL+1,LY
      CEQPT(3) = CEQPT(3) + (NEQRPL(1,I) * ROECST(1))
2480  CONTINUE
      DO 2490 K=1,NWT
      CPEQ(K) = CPEQ(K) - PKEQPC(K,LL+1)/ONEMIL
2490  CONTINUE
C
      CALL OUTPUT
C
C
C
1100  FORMAT(20A4)
C
      END
      SUBROUTINE EMPLAC(K,I)
C
      INCLUDE PARAM
      INCLUDE SFTCOM
      INCLUDE SOPCOM
      INCLUDE UOPCOM
      INCLUDE INCOM
      INCLUDE COCOM
C
9999  CONTINUE
      IF(IMET(K).LE.15) GOTO 100
      WRITE(IO,99) K
99    FORMAT(' IMET OF WASTE TYPE',I5,' IS OUT OF RANGE')
      STOP
C
100  CONTINUE
C
      GOTO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15) IMET(K)
C
1    CONTINUE
2    CONTINUE

```

```

3  CONTINUE
4  CONTINUE
C
C  SOLO EMPLACEMENT OPTIONS 1,2,3, AND 4
C
C  CALL PPUT(K,I)
C
C  GOTO 200
C
C
5  CONTINUE
6  CONTINUE
7  CONTINUE
C
C  COEMPLACEMENT OPTIONS WITH VBH THE PRIMARY EMPLACEMENT METHOD
C
C  IF(KIND(K).NE.1) GOTO 150
C
C  PRIMARY VBH EMPLACEMENT
C
C  CALL PPUT(K,I)
C
C  GOTO 200
C
150 CONTINUE
    IF(IMET(K).EQ.6.AND.KIND(K).EQ.2) GOTO 20
    CALL SPUT(K,I)
    GOTO 200
20  CONTINUE
C
C  TERTIARY EMPLACEMENT
C
C  CALL TPUT(K,I)
C  GOTO 200
C
8  CONTINUE
9  CONTINUE
10 CONTINUE
11 CONTINUE
12 CONTINUE
C
C  COEMPLACEMENT OPTIONS WITH HBH THE PRIMARY EMPLACEMENT METHOD
C
C  IF(KIND(K).NE.1) GOTO 30
C  CALL PPUT(K,I)
C  GOTO 200
C
30 CONTINUE
    IF(IMET(K).EQ.9.OR.IMET(K).EQ.12.AND.KIND(K).EQ.2) GOTO 40

```

```

C
C SECONDARY EMPLACEMENT
C
C     CALL SPUT(K,I)
C     GOTO 200
C
C 40 CONTINUE
C
C TERTIARY EMPLACEMENT
C
C     CALL TPUT(K,I)
C     GOTO 200
C
C 13 CONTINUE
C 14 CONTINUE
C 15 CONTINUE
C
C COEMPLACEMENT OPTIONS WITH ROOM EMPLACEMENT THE PRIMARY E. METHOD
C
C     IF(K.GT.MPRIME) GOTO 50
C     CALL PPUT(K,I)
C     GOTO 200
C 50 CONTINUE
C     IF(IMET(K).EQ.15.AND.KIND(K).EQ.2) GOTO 60
C
C SECONDARY EMPLACEMENT
C
C     CALL SPUT(K,I)
C     GOTO 200
C 60 CONTINUE
C
C TERTIARY EMPLACEMENT
C
C     CALL TPUT(K,I)
C
C 200 CONTINUE
C
C STORE UNEMPLACED SECONDARY WASTE
C
C     IF(K.LE.MPRIME) RETURN
C     CALL SSTOR(K,I)
C
C RETURN
C DEBUG SUBCHK
C AT 9999
C END
C SUBROUTINE EMTYPE(K,IMET,KIND,IEMP,MPRIME)

```

```

C      FINDS EMPLACEMENT METHOD FROM VALUES OF K,IMET,KIND
C      BY RETURNING THE FOLLOWING VALUES OF IEMP(K)
C      IEMP = 1  BOREHOLE
C      IEMP = 2  ROOM EMPLACEMENT
C      IEMP = 3  TRENCH
C
1  CONTINUE
    L = IMET
    IF (K.GT.MPRIME) GOTO 10
C
    IF(L.EQ.1.OR.L.GE.4.AND.L.LE.12) GOTO 100
    IF(L.EQ.3) GOTO 300
    GOTO 200
10  CONTINUE
    IF(KIND.EQ.3) GOTO 20
    GOTO 200
20  CONTINUE
    IF(L.EQ.5.OR.L.EQ.6.OR.L.EQ.8.OR.L.EQ.9) GOTO 100
    IF(L.EQ.11.OR.L.EQ.12) GOTO 300
    GOTO 200
C
C      BOREHOLE
C
100  CONTINUE
    IEMP = 1
    RETURN
C
C      ROOM EMPLACEMENT
C
200  CONTINUE
    IEMP = 2
    RETURN
C
C      TRENCH
C
300  CONTINUE
    IEMP = 3
    RETURN
    DEBUG SUBCHK
    AT 1
    END
    FUNCTION HOLDC(A,B,D,N)
C
    IF(N.NE.1) GO TO 100
    HOLDC = A * EXP(B*D)
    RETURN
100 CONTINUE
C

```

```

        IF(N.GT.4) GO TO 200
        HOLDC = A * D **B
        RETURN
200 CONTINUE
C
        HOLDC = 0.0
        RETURN
        END
        FUNCTION HOLMC(A,B,C,D,E,F,G,H,P,R,S,T,N)
C
        X = (P-G)
        IF(X.EQ.0.) GO TO 100
        WBCSTF = X / (G-A) + 1
        SLEVCS = X / (2* B-C) + 1
        BFLMCS = X / (2* D-E) + 1
        GO TO 200
100 CONTINUE
        WBCSTF = 1.
        SLEVCS = 1.
        BFLMCS = 1.
200 CONTINUE
C
        HOLMC = (H * WBCSTF + R * SLEVCS + S * BFLMCS) / T
C
        RETURN
        END
        SUBROUTINE INPUT
C
        INCLUDE PARAM
        INCLUDE INCOM
        INCLUDE HUBCOM
        INCLUDE SOPCOM
        INCLUDE MNDCOM
        INCLUDE WAFCOM
        INCLUDE RECCOM
        INCLUDE UOPCOM
        INCLUDE SFTCOM
        INCLUDE UTLCOM
        INCLUDE SSTCOM
        INCLUDE PRECOM
        INCLUDE PSTCOM
        INCLUDE COCOM
        INCLUDE TABL
C
C
        DATA BLANK/'      '/
        DATA (WSTYP(L,NWP1),L=1,3)/'REPO','SITO','RY  '/
C
1 CONTINUE

```



```

C      READ(IN,1100) DATYPE
      READ(IN,1100) TITLE
      READ(IN,1100) (RPTYP(L),L=1,4),(PLACE(L),L=1,4),(TEKNEK(L),
1L=1,4)
      READ(IN,*) IYRCON,IYRPRO,LASTYR,IYDOL,IDDYR,NWT
1  , DRATE
      IYRDIS = IYDOL
      READ(IN,*) DEPTH,DENSTY,SPUC
1,CONFAC
      NSTM1 = NST-1
      READ(IN,*) (SFTNUM(L),L=1,NSTM1)
      DO 75 K=1,NWT
      READ(IN,1100) (WSTYP(L,K),L=1,3),
1 (PKGDS1(L,K),L=1,4),(PKGDS2(L,K),L=1,4),(PKGDS3(L,K),L=1,4)
75 CONTINUE
      READ(IN,*) (KIND(K),K=1,NWT),(UNTMAS(K),K=1,NWT)

C
C
      LL=IYRPRO-IYRCON
      LY=LASTYR-IYRCON+1
      IDD=IDDYR
      LYY=LY

C
C
      WASTOP

C
      READ(IN,1100) DATYPE
      READ(IN,*) SIZE, CPA,
1 GXEF, (NFTL(K),K=1,NWT), FFTL
2,NALSH,NRHRMA,NCHRNA,WRKYR,MTRPSD,MTCPSD
3,(SHCFAC(L),L=1,NS),OVRHDF
      READ(IN,*) (PANLEN(K),K=1,NWT), (PANWTH(K),K=1,NWT)
      READ(IN,*) (WUPP (K),K=1,NWT), (WVOLPU(K),K=1,NWT)
      READ(IN,*) (TTTR (K),K=1,NWT), (TTTK (K),K=1,NWT)
      READ(IN,*) (RCCAP (K),K=1,NWT), (TCCAP (K),K=1,NWT)
      READ(IN,*) (PKGCAP(K),K=1,NWT)
      READ(IN,*) (RMLN (K),K=1,NWT), (RMWTH (K),K=1,NWT)
      READ(IN,*) (CLSEMP(K),K=1,NWT), (XRASHO(K),K=1,NWT)
      READ(IN,*) (HPMIN (K),K=1,NWT), (PKGUP(K),K=1,NWT)
      READ(IN,*) (NBSPPA(K),K=1,NWT)
      DO 50 K=1,NWT
      READ(IN,*) (WSTSUP(K,I),I=1,LY)
50 CONTINUE
      DO 60 K=1,NWT
      READ(IN,*) (TDWI(I,K),I=1,LY)
60 CONTINUE
      DO 70 K=1,NWT
      READ(IN,*) (CRRAIL(K,I),I=1,LY)
70 CONTINUE

```

```

DO 120 L = 1,NWO
  READ(IN,*) WOCST(L),WOLBRR(L),WOLBRC(L),WOEQPL(L),
1 WOEQPC(L),WOECST(L)
  READ(IN,*) (WOMATU(L,K),K=1,NWT),(WOMATC(L,K),K=1,NWT)
120 CONTINUE
C
C MINDEV
C
  READ(IN,1100) DATYPE
  READ(IN,*) LEAD, NCOR, TCPCLN, PCTFIL, XPANFC, MSA, TONPAR,
1 OFTADC, NBYR, OTHBFL, OTBDEN, BKRATE, MSHIFT(1), MSHIFT(2)
  READ(IN,*) (RMHMIN(K),K=1,NWT)
  NCOR = NCOR + 1
  DO 145 M= 1, NCOR
    READ (IN,1100) (CORDSC(N,M),N=1,12)
    READ (IN,*) CORWTH(M), CORHT(M), NCORPP(M)
    READ(IN,*) (CORLEN(M,K),K=1,NWT)
145 CONTINUE
    READ(IN,*) (DELRHT(K),K=1,NWT), (PKGLN(K),K=1,NWT), (NRWSPR(K),
1 K=1,NWT)
    READ(IN,*) (PORMDF(1,I),I=1,LL), (PORMDF(2,I),I=1,LL)
1 , (SCSKD(I),I=1,LL)
    READ(IN,*) (SFTDD(L),SFTXR(L),SFTXD(L),L=1,NST)
    READ(IN,*)MSWTCH
    IF(MSWTCH.EQ.0) GO TO 142
    DO 140 L=1,NRO
      READ(IN,*) ROCST(L),ROLBRR(L),ROLBRC(L),ROEQPL(L),
1 ROEQPC(L),ROECST(L),ROMATU(L),ROMATC(L)
140 CONTINUE
      GO TO 144
142 CONTINUE
      READ(IN,*) VARMC,RCKHC,SURFDC,BKFLC
144 CONTINUE
C
C RECEIVE
C
  READ(IN,1100) DATYPE
  READ(IN,*) NRBPRM, NCBPRM
  READ(IN,*) (RMCWS1(I),I=1,LL) , (RMCWS2(I),I=1,LL)
  READ(IN,*) (RMNWS1(I),I=1,LL) , (RMNWS2(I),I=1,LL)
C
C UGOPR
C
  READ(IN,1100) DATYPE
  DO 130 L =1,NTO
    READ(IN,*) TOCST(L),TOLBRR(L),TOLBRC(L),TOEQPL(L),
1 TOEQPC(L),TOECST(L)
    READ(IN,*) (TOMATU(L,K),K=1,NWT),(TOMATC(L,K),K=1,NWT)
130 CONTINUE

```

```

      READ(IN,*) IDET,IUHDCM
      READ(IN,*) (TRCHWD(K),K=1,NWT)
      DO 200 K=1,NWT
      READ(IN,*) NPKPEU(K),PKGDI A(K),REFDIA(K),HOLDEL(K),IMET(K)
200  CONTINUE
      READ(IN,*) MPRIME,NNHG,ICLR
      IF(MPRIME.EQ.NWT) GOTO 235
      READ(IN,*) (STOMAX(K),K=1,NWT)
      IF(NNHG.NE.NWT-MPRIME) NNHG = NWT - MPRIME
      IF(NNHG.EQ.0) GOTO 235
      DO 230 K=1,MPRIME
      DO 230 J=1,NNHG
      READ(IN,*) COCON(J,K),PCTCON(J,K)
230  CONTINUE
      READ(IN,*) (SWPPR(K),K=1,NWT)
      ICH = 0
      DO 275 K=1,NWT
      IF(KIND(K).NE.2) GOTO 275
      IF(IMET(K).LE.4.OR.IMET(K).EQ.7.OR.IMET(K).EQ.10.OR.
+      IMET(K).GE.13) GOTO 275
      ICH = 1
275  CONTINUE
      IF(ICH.EQ.0) GOTO 4470
      DO 4460 J=1,NWT,5
      JJ=J+4
      READ(IN,*) (CONTRU(K),K=J,JJ)
4460 CONTINUE
4470 CONTINUE
235  CONTINUE
      DO 250 L=1,NUO
      READ(IN,*) UOCST(L),UOLBRR(L), UOLBRC(L),UOEQPL(L),UOEQPC(L),
1UOECST(L)
      IF(IDET.EQ.1) GO TO 250
      DO 240 K=1,NWT
      IF(KIND(K).EQ.1) M=1
      IF(KIND(K).EQ.2) M=2
      IF(KIND(K).EQ.3) M=1
      IF(M.NE.L) GO TO 240
      READ(IN,*) UOMATU(L,K),UOMATC(L,K)
240 CONTINUE
250 CONTINUE
      DO 280 K=1,NWT
      READ(IN,*) WABATH(K),SLEETH(K),BCKFTH(K),CAPLEN(K),
280  CONTINUE
      IF(IDET.NE.1) GO TO 300
      DO 270 K=1,NWT
      READ(IN,*) REFMAT(K), REFSLV(K),REFBKF(K)
1,REFDEP(K)
      READ(IN,*) CHMTEM(K),PKGPRK(K),RCKCST(K)

```

```

270 CONTINUE
300 CONTINUE
    DO 155 L=1,NEE
    IF(IUHDCM.EQ.1.AND.L.EQ.1) GO TO 350
    READ(IN,*) EECST(L),EELBRR(L),EELBRC(L),EEEQPL(L),EEEQPC(L),
1 EEEECST(L),EEMATU(L),EEMATC(L)
    GO TO 155
350 CONTINUE
    READ(IN,*) EELBRC(L),EEECST(L),EEMATC(L),
1 HDLCFC,HDECFC,HDMCFC,HDCON1,HDCON2,NTYP
155 CONTINUE
C
C SHAFT
C
    READ(IN,1100) DATYPE
    READ(IN,*) MXMPRH,MXMMRH, TNPT1, TNPT2,
1 SCCON1, SCCON2,IEQTP1, CCON1, CCON2, IEQTP2
2 ,WCCON1,WCCON2,IEQTP3
    READ(IN,*) (SFTLD(L),L=1,NST)
    READ(IN,*) (SFTHE(L),L=1,NST)
    READ(IN,*) (SFTSS(L),L=1,NST)
    READ(IN,*) (NPKPTR(K),K=1,NWT),(NPKPTC(K),K=1,NWT)
C
C SURFST
C
    READ(IN,1100) DATYPE
    READ(IN,*) (SFCS(L),L=1,LL)
    READ(IN,*) NSC,SOA
    DO 150 L=1,NSC
    READ(IN,1100) (SFTITL(N,L),N=1,12)
    READ(IN,*) SFCST(L),SFLBRR(L),SLBR(L),SFLBRC(L),SFEQPL(L),
1 SFECST(L),SFMATU(L),SFMATC(L)
150 CONTINUE
C
C UTILITY
C
    READ(IN,1100) DATYPE
    READ(IN,*) UPC ,STMC, DWTRC, WTRC, DSLC
    READ(IN,*) RAPP, SFSP, OWHP, SFP,MROP,RVP,
1 ACP, HEATS, RAPS, ORS
    READ(IN,*) VOLW, VOLDW, SUD, UGD, RED
C
C PREOP
C
    READ(IN,1100) DATYPE
    READ(IN,*) (SCDF(I),I=1,LL)
1, (CLNDF(I),I=1,LL)
2, (CLCDF(I),I=1,LL)
    READ(IN,*) TCLAND,TCLICE,STPCT,AECON1,AECON2,IEQTP4

```

```

C
C POSTOP
C
    READ(IN,1100) DATYPE
    IF (IDD.EQ.0) GOTO 170
    READ(IN,*) (DBDF(I),I=1,IDD)
    READ(IN,*) (SSDF(I),I=1,IDD)
    READ(IN,*) (FDDF(I),I=1,IDD)
170 CONTINUE
    READ(IN,*) SSCON1, SSCON2, IEQTP5, DCOMCF, SURCST, ITSURV
    READ(IN,*) NDDC,DDFRAC
    IF(NDDC.EQ.0) GO TO 165
    DO 160 L=1,NDDC
    READ(IN,1100) (DDTITL(N,L),N=1,12)
    READ(IN,*) DDCST(L),DDLBR(L),DDLBR(L),DDEQPC(L),DDMATC(L)
160 CONTINUE
165 CONTINUE
    READ(IN,*) SCCST,SCLBRR,SCLBRC,SCEQPC,SCMATU,SCMATC
C
C
1100 FORMAT(20A4)
C
C
C TABLE TO BE PRINTED
C
    DO 400 I=1,27
    TABLES(I) = 0
400 CONTINUE
    READ (IN,*,END=410,ERR=410) TABLES
    RETURN
410 CONTINUE
    IF (TABLES(1).NE.0) RETURN
    DO 420 I=1,27
    TABLES(I) = I
420 CONTINUE
    RETURN
    DEBUG SUBCHK
    AT 1
    END
    SUBROUTINE MINDEV
C
    INCLUDE PARAM
    INCLUDE MNDCOM
    INCLUDE OUTCOM
    INCLUDE PSTCOM
    INCLUDE WAFCOM
    INCLUDE SFTCOM
    INCLUDE SOPCOM
    INCLUDE COCOM

```

```

C      DIMENSION CORVR(NW)
C
1      CONTINUE
C
      LE=LY-LEAD
      DPY = WRKYR
      LED = LEAD
C
      DO 150 I=1,NY
      SRKMND(I) = 0.0
      SRKHST(I) = 0.0
      SRKBKF(I) = 0.0
      RKSURF(I) = 0.0
150    CONTINUE
C      MINING DURING OPERATION
      DO 200 K=1,NWT
      DO 100 M=2,NCOR
      CORVR(K) = CORVR(K) + CORLEN(M,K) * CORHT(M) * CORWTH(M)
1      * NCORPP(M) / RMPILU(K)
100    CONTINUE
      RMHT(K) = AMAX1(PKGLN(K),RMHMIN(K))+DELRHT(K)
      RMHTT = RMHMIN(K) + DELRHT(K)
      CALL EMTYPE (K,IMET(K),KIND(K),IEMP(K),MPRIME)
      IF(IEMP(K).EQ.2) RMHT(K) = AMAX1(PKGLN(K)*PKGUP(K),RMHMIN(K))
      *      + DELRHT(K)
      IF (IEMP(K).EQ.2.AND.RMHTT.NE.RMHT(K)) MRFLG(K) = 1
      IF(IMET(K).GE.4) RMHT(K) = RMHMIN(K) + DELRHT(K)
      RKVOL(K) = RMLN(K)*RMWTH(K)*RMHT(K)
      DO 200 I=1,LE
      DO 170 M=2,NCOR
      CORVOL(K,I)=CORVOL(K,I)+CORLEN(M,K)*CORWTH(M)*CORHT(M)*NCORPP(M)
1      * PNLREQ(K,I+LED) / AREAPP(K)
170    CONTINUE
      VOLMND(K,I) = RMFILD(K,I+LED)*RKVOL(K) + CORVOL(K,I)
      RKMSM(K,I) = VOLMND(K,I) * DENSTY + RKMSM(K,I)
200    CONTINUE
C
C      PRE-OPERATIONAL MINING
C
      SRM = 0.0
      DO 220 K=1,NWT
      SRMM(K) = 0.0
220    CONTINUE
C
      DO 250 K=1,NWT
      DO 250 I=1,LY
      SRMM(K) = SRMM(K) + RKMSM(K,I)

```

```

    SRM = SRM + RKMSM(K,I)
250 CONTINUE
C
    DO 260 K=1,NWT
    SHAFTX (K) = 0.
    DO 260 L=1,NST
    SHAFTX(K) = SHAFTX(K) + (((SFTXR(L) + SFTDD(L) /2.) **2)*PI)
1 * SFTXD(L)) * (SRMM(K)/SRM) * DENSTY * SFTNUM(L)
260 CONTINUE
    DO 265 K=1,NWT
    DO 265 I=1,LL
    SHRKMD(K,I) = SHAFTX(K) * SCSKD(I)
265 CONTINUE
C
    DO 270 K=1,NWT
    DO 270 I=1,LL
    CPCMM(K,I) = TCPCLN * CORWTH(1)*CORHT(1)*DENSTY*SRMM(K)/
1SRM * PORMDF(1,I)
    CORMAS(K,I) = CORARA*CORHT(1)*DENSTY*SRMM(K)/SRM*
1 PORMDF(2,I) * M2ACRE
270 CONTINUE
C
C   MINING LOGISTICS
C
    DO 300 K=1,NWT
    DO 300 I=1,LY
    ANFILD(I) = ANFILD(I) + RMFILD(K,I)
300 CONTINUE
    DO 310 I=1,LE
    ANNEX(I) = ANNEX(I) + RMFILD(K,I+LED)
310 CONTINUE
    DO 330 I=2,LY
    CAREW(K,I)=CAREW(K,I-1) + PNLREQ(K,I)
330 CONTINUE
    CUMFLD(1) = ANFILD(1)
    DO 350 I=2,LY
    CUMFLD(I) = CUMFLD(I-1) + ANFILD(I)
350 CONTINUE
    DO 360 K=1,NWT
    DO 360 I=1,LL
    RKMSM(K,I) = RKMSM(K,I) + CORMAS(K,I) + CPCMM(K,I)
360 CONTINUE
    DO 520 I=1,LY
    DO 520 K=1,NWT
    SRKMND(I) = SRKMND(I) + RKMSM(K,I)
520 CONTINUE
C   MINING          RESOURCE REQUIREMENTS
C   LABOR
C   PRE-OPERATIONAL MINING

```

```

C      ASSUMES 3 SHIFT/DAY OPERATION
      NRON = 1
      IF(MSWTCH.EQ.0) GO TO 371
      DO 365 I=1,LL
      IF(SRKMND(I).EQ.0.) GO TO 365
      TRM = 0.0
      DO 361 K=1,NWT
      TRM = TRM + RKMSM(K,I)
361 CONTINUE
      ANCPTY(I) = ROEQPC(NRON) * MSHIFT(1) * DPY
C      NUMBER OF MINING MACHINES NEEDED
      IF(ANCPTY(I).EQ.0.) GO TO 365
      NMMACH(I) = TRM / ANCPTY(I) + .999
C      LABOR CALCULATION
      DO 363 K=1,NWT
      RMNLBR(K,I) = NMMACH(I) * ROLBRR(NRON) * 3.*RKMSM(K,I)/SRKMND(I)
363 CONTINUE
365 CONTINUE
C
C      MINING DURING OPERATIONS
C
      LB = LL + 1
      DO 368 I=LB,LY
      IF(SRKMND(I).EQ.0.) GO TO 368
      TRM = 0.0
      DO 362 K=1,NWT
      TRM = TRM + RKMSM(K,I)
362 CONTINUE
      ANCPTY(I) = ROEQPC(NRON) * MSHIFT(2) * DPY
      IF(ANCPTY(I).EQ.0.) GO TO 368
      NMMACH(I) = TRM / ANCPTY(I) + .999
      DO 364 K=1,NWT
      RMNLBR(K,I) = NMMACH(I) * ROLBRR(NRON) * MSHIFT(2)
      1 * RKMSM(K,I) / SRKMND(I)
C      MATERIALS
      RMMATR(K,I) = ROMATU(NRON) * RKMSM(K,I)
364 CONTINUE
368 CONTINUE
      DO 369 I=1,LY
      TRM = SRKMND(I)
      IF(I.EQ.1) NLAST = 0
      NDUM = NMMACH(I)
      CALL REPLAS(NDUM,NLAST,K,NEQRPL(NRON,I),TRM,ANCPTY(I),
      1ROEQPL(NRON),NW,I)
      NLAST = NDUM
369 CONTINUE
371 CONTINUE
C
C      BACKFILLING LOGISTICS

```



```

C
DO 370 K=1,NWT
  RMUVOL(K) = RKVOL(K) + CORVR(K)
370 CONTINUE
  CUMBKFL = 0.0
C
  IFLG = 0
  IFLAG = 0
  LB= NBYR +IYRPRO-IYRCON+1
  BFLEFF = PCTFIL/XPANFC
  IF(LB.GT.LY) UBFRMS = CUMFLD(LY)
  IF(LB.GT.LY) GO TO 530
  DO 500 I=LB,LY
    IF(IFLAG.GT.0) GO TO 450
    DO 380 K=1,NWT
      RUBKFL(K) = (VOLMND(K,I)+OTHBFL / (1.-OTHBFL)*VOLMND(K,I))
1 / (RMUVOL(K) * BFLEFF)
      CUMBKFL = CUMBKFL + RUBKFL(K)
380 CONTINUE
      IF(IFLG.EQ.0) UBFRMS = CUMFLD(I) - CUMBKFL
      IF(IFLG.EQ.0) GOTO 381
      DO 382 K=1,NWT
        UBFRMS = UBFRMS + RMFILD(K,I) - RUBKFL(K)
382 CONTINUE
381 CONTINUE
      IF(UBFRMS.LE.0.) GO TO 400
      DO 390 K=1,NWT
        RKBKFL(K,I) = VOLMND(K,I) * DENSTY
        OTBKFL(K,I)=VOLMND(K,I)*OTHBFL*OTBDEN / (1.-OTHBFL)
        IF(VOLMND(K,I).EQ.0.) GO TO 385
        GO TO 390
385 CONTINUE
        RKBKFL(K,I) = RMFILD(K,I) * RMUVOL(K) *BFLEFF * DENSTY
1 * (1. - OTHBFL)
        OTBKFL(K,I)=OTHBFL*(RKBKFL(K,I)/DENSTY) *OTBDEN / (1.-OTHBFL)
        UBFRMS = UBFRMS - RMFILD(K,I)
        IFLG = I
390 CONTINUE
        IF(UBFRMS.LE.0.) GO TO 400
        GO TO 455
400 CONTINUE
        IFLAG = I
        TRFILD = 0.0
        DO 405 K=1,NWT
          TRFILD = TRFILD + RMFILD(K,I)
405 CONTINUE
          DO 410 K=1,NWT
            RKBKFL(K,I) = (RUBKFL(K) + UBFRMS * RMFILD(K,I) / TRFILD)
1 * RMUVOL(K) * BFLEFF * DENSTY * (1.-OTHBFL)

```

```

    OTBKF(K,I)=OTHBFL*RKBKF(K,I)*OTBDEN/DENSTY / (1.-OTHBFL)
410 CONTINUE
    UBFMS = 0.0
    GO TO 455
450 CONTINUE
    DO 420 K=1,NWT
        RKBKF(K,I) = RMFILD(K,I)*RMUVOL(K)*BFLEFF*DENSTY
        1 * (1-OTHBFL)
        OTBKF(K,I)=OTHBFL*RKBKF(K,I)*OTBDEN/DENSTY / (1.-OTHBFL)
420 CONTINUE
455 CONTINUE
    UBFRPY(I) = UBFMS
500 CONTINUE
530 CONTINUE
C
    CALL WAFAM
C
    SURF = 0.
    MSFLG = 1
    IF (MSA.EQ.0.0) MSFLG = 0
    DO 600 I=1,LY
        DO 550 K=1,NWT
            RCKHST(K,I)=(RKMSM(K,I)+SHRKMD(K,I)+EEXRKM(K,I))-RKBKF(K,I)
            IF(MSFLG.EQ.0) MSA = MSA + RCKHST(K,I) / TONPAR
            SURF=SURF+RCKHST(K,I)
            SRKHST(I) = SRKHST(I) + RCKHST(K,I)
            SRKBKF (I) = SRKBKF(I) + RKBKF(K,I)
            SOTBKF(I) = SOTBKF(I) + OTBKF(K,I)
            RCKHST(K,I) = ABS(RCKHST(K,I))
550 CONTINUE
            SS = MSA * TONPAR
            RKOFFS(I)=AMAX1(0.,(SURF-SS))
            RKSURF(I) = SURF-RKOFFS(I)
            SURF = SURF - RKOFFS(I)
            SRKHST(I) = ABS(SRKHST(I))
600 CONTINUE
C
    DO 620 K=1,NWT
        UBKVOL(K) = UBFMS * RMUVOL(K) * WAFMV(K)
620 CONTINUE
C
    DO 630 K=1,NWT
        DO 630 I=LB,NBKF
            WRBKFL = WRBKFL + RKBKF(K,I) + OTBKF(K,I)
630 CONTINUE
C
    RETURN
    DEBUG SUBCHK
    AT 1

```

```

END
SUBROUTINE  OUTPUT
C
  INCLUDE PARAM
  INCLUDE OUTCOM
  INCLUDE INCOM
  INCLUDE RECCOM
  INCLUDE SOPCOM
  INCLUDE SSTCOM
  INCLUDE MNDCOM
  INCLUDE SFTCOM
  INCLUDE HUBCOM
  INCLUDE PRECOM
  INCLUDE PSTCOM
  INCLUDE UTLCOM
  INCLUDE UOPCOM
  INCLUDE WAFCOM
  INCLUDE COCOM
  INCLUDE TABL
C
  CHARACTER *1 NTA
  DIMENSION TOT(2), NTA(10), NPKGT(NW), TTSH(NW+1), TSH(NY),
* TSFNAS(NW), THDMAS(NW), DHPPKG(NW,NY)
  DATA DASH/'-'/
  DATA BLANK/' '/
  DATA TOT/'TOTA','L'/
  DATA NTA /'A','B','C','D','E','F','G','H','I','J'/
C
1  CONTINUE
  IT = 1
C
  LED = LEAD
  LE=LY+IDD
  LEE = NBKF + IDD
C
C TABLE 1
C
  IF (TABLES(IT).NE.1) GOTO 8010
  WRITE(IO,1000)
  WRITE(IO,1055)  TITLE
  WRITE(IO,1001)  DA
  WRITE(IO,1002)  TI
  WRITE(IO,1003)  (RPTYP(L),L=1,4)
  WRITE(IO,1004)
  WRITE(IO,1005)  (PLACE(L),L=1,4)
  WRITE(IO,1006)  IYRCON
  WRITE(IO,1007)  IYRPRO
  WRITE(IO,1008)  DEPTH
  WRITE(IO,1026)  SOA

```

```

WRITE(IO,1027) SIZE
WRITE(IO,1028) (TEKNEK(L),L=1,4)
WRITE(IO,1029) CPA
DO 50 M=1,NCOR
WRITE(IO,1030) (CORDSC(N,M),N=1,12)
IF (M.GT.1) WRITE(IO,1047) NCORPP(M)
WRITE(IO,1031) CORWTH(M)
WRITE(IO,1033) CORHT(M)
IF(M.GT.1) GOTO 51
WRITE(IO,1670) CORLEN(1,1)
GOTO 50
51 CONTINUE
WRITE(IO,1032)
DO 55 K=1,NWT
WRITE(IO,1650) (WSTYP(L,K),L=1,3),CORLEN(M,K)
55 CONTINUE
50 CONTINUE
WRITE(IO,1036) GXEF
WRITE(IO,1037) SPUC
RATE = DRATE * 100.
WRITE(IO,1038) RATE
WRITE(IO,1000)
KNWT=1
NWTT=NWT
IF(NWT.GT.5) NWTT=5
JNWT = NWTT-KNWT+1
9010 CONTINUE
WRITE(IO,1039)
LX = JNWT * 13-2
WRITE(IO,1040) (DASH,L=1,LX)
WRITE(IO,1041) ((WSTYP(L,K),L=1,3),K=KNWT,NWTT)
LX = JNWT*11
WRITE(IO,1042) (DASH,L=1,LX)
WRITE(IO,1043) (XRASHO(K),K=KNWT,NWTT)
WRITE(IO,1034) (PANWTH(K),K=KNWT,NWTT)
WRITE(IO,1035) (PANLEN(K),K=KNWT,NWTT)
WRITE(IO,1044) (RMWTH (K),K=KNWT,NWTT)
WRITE(IO,1045) (RMHT (K),K=KNWT,NWTT)
WRITE(IO,1046) (RMPILU(K),K=KNWT,NWTT)
IF(NWTT.EQ.NWT) GOTO 9000
KNWT=NWTT+1
NWTT=KNWT+4
IF(NWTT.GT.NWT) NWTT=NWT
JNWT=NWTT-KNWT+1
GOTO 9010
9000 CONTINUE
DO 9005 K=1,NWT
IF(MRFLG(K).EQ.1) WRITE(IO,1640) PKGPUP(K),(WSTYP(L,K),L=1,3)
9005 CONTINUE

```

```

        WRITE(IO,1052)
        WRITE(IO,1053)
        WRITE(IO,1054)
8010  IF (TABLES(IT).EQ.1) IT = IT+1
C
C      TABLE 2
C
        IF (TABLES(IT).NE.2) GOTO 8020
        KNWT=1
        NWTT=NWT
        IF(NWT.GT.5) NWTT=5
        JNWT = NWTT-KNWT+1
        WRITE(IO,1000)
        WRITE(IO,1055) TITLE
        WRITE(IO,1001) DA
        WRITE(IO,1002) TI
        WRITE(IO,1078) (RPTYP(L),L=1,4)
        WRITE(IO,1079)
        WRITE(IO,1072)
        WRITE(IO,1073)
        WRITE(IO,1080) RMCAP,PMCAP
        WRITE(IO,1081)
        WRITE(IO,1082) CMCAP
        WRITE(IO,1083)
220   CONTINUE
        LX = JNWT * 13-2
        WRITE(IO,1223)
        WRITE(IO,1224) (DASH,L=1,LX)
        LX = JNWT * 11
        WRITE(IO,1084) ((WSTYP(L,K),L=1,3), K=KNWT,NWTT)
        WRITE(IO,1085) (DASH,L=1,LX)
        WRITE(IO,1010)
        WRITE(IO,1086) (RCCAP(K),K=KNWT,NWTT)
        WRITE(IO,1087) (TCCAP(K),K=KNWT,NWTT)
        IF(NWTT.EQ.NWT) GOTO 225
        KNWT = NWTT+1
        NWTT = NWTT+4
        IF(NWTT.GT.NWT) NWTT=NWT
        JNWT = NWTT-KNWT+1
        GOTO 220
225   CONTINUE
8020  IF (TABLES(IT).EQ.2) IT = IT+1
C
C      TABLE 3
C
        IF (TABLES(IT).NE.3) GOTO 8030
        KNWT=1
        NWTT=NWT
        IF(NWT.GT.10) NWT=10

```

```

JNWT = NWTT-KNWT+1
9030 CONTINUE
WRITE(IO,1000)
WRITE(IO,1055) TITLE
WRITE(IO,1001) DA
WRITE(IO,1002) TI
WRITE(IO,1160) (RPTYP(L),L=1,4)
WRITE(IO,1161)
WRITE(IO,1162)
LX = JNWT * 13-(NWT+1)
WRITE(IO,1163) (DASH,L=1,LX)
WRITE(IO,1164) ((WSTYP(L,K),L=1,3), K=KNWT,NWTT)
LX = JNWT * 11
WRITE(IO,1165) (DASH,L=1,LX)
WRITE(IO,1010)
DO 350 K=KNWT,NWTT
DO 350 I=1,LY
CRRAIL(K,I) = CRRAIL(K,I) * 100.
350 CONTINUE
IYR = IYRCON - 1
DO 400 I=1,LY
IYR = IYR + 1
WRITE(IO,1166) IYR,(CRRAIL(K,I),K=KNWT,NWTT)
400 CONTINUE
IF (NWTT.EQ.NWT) GOTO 9035
KNWT=NWTT+1
NWTT=NWTT+9
IF(NWTT.GT.NWT) NWTT=NWT
GOTO 9030
9035 CONTINUE
8030 IF (TABLES(IT).EQ.3) IT = IT+1
C
C TABLE 4 A
C
IF (TABLES(IT).NE.4) GOTO 8040
WRITE(IO,1000)
WRITE(IO,1055) TITLE
WRITE(IO,1001) DA
WRITE(IO,1002) TI
WRITE(IO,1215) (RPTYP(L),L=1,4)
WRITE(IO,1216)
WRITE(IO,1217)
WRITE(IO,1218)
WRITE(IO,1219)
WRITE(IO,1220)
WRITE(IO,1221)
WRITE(IO,1010)
IYR = IYRCON - 1
DO 340 I=1,LY

```

```

        IYR = IYR + 1
        WRITE(IO,1222) IYR,  NRMRH(I), NPMRH(I), NRMCH(I)
340  CONTINUE
C
C      TABLE 4B
C
        KNWT=1
        NWTT=NWT
        IF(NWT.GT.5) NWTT=5
        JNWT = NWTT-KNWT+1
        IF(NWT.LE.5) JNWT=JNWT+1
9040  CONTINUE
        WRITE(IO,1000)
        WRITE(IO,1055) TITLE
        WRITE(IO,1001) DA
        WRITE(IO,1002) TI
        WRITE(IO,1460) (RPTYP(L),L=1,4)
        WRITE(IO,1216)
        WRITE(IO,1410)
        LX = JNWT * 13-2
        WRITE(IO,1420) (DASH,L=1,LX)
        LX = JNWT * 11
        IF(NWT.LE.5) WRITE(IO,1430) ((WSTYP(L,K),L=1,3), K=KNWT,NWTT),TOT
        IF(NWT.GT.5.AND.NWTT.EQ.5)
+   WRITE(IO,1430) ((WSTYP(L,K),L=1,3),K=KNWT,NWTT)
        IF(NWT.GT.5.AND.NWTT.GT.5) WRITE(IO,1430)
+   ((WSTYP(L,K),L=1,3),K=KNWT,NWTT),TOT
        WRITE(IO,1440) (DASH,L=1,LX)
        WRITE(IO,1010)
        IYR = IYRCON - 1
        DO 320 I=1,NWP1
        TTSH(I) = 0
320  CONTINUE
        DO 341 I=1,LY
        IYR = IYR + 1
        DO 325 K = KNWT,NWTT
        TSH(I) = TSHPCN(K,I) + TSH(I)
        TTSH(K) = TTSH(K)+TSHPCN(K,I)
325  CONTINUE
        TTSH(NWT+1) = TTSH(NWT+1)+TSH(I)
        IF(NWT.LE.5) WRITE(IO,1445) IYR, (TSHPCN(K,I),K=KNWT,NWTT),TSH(I)
        IF(NWT.GT.5.AND.NWTT.EQ.5)
+   WRITE(IO,1445) IYR,(TSHPCN(K,I),K=KNWT,NWTT)
        IF(NWT.GT.5.AND.NWTT.GT.5)
+   WRITE(IO,1445) IYR,(TSHPCN(K,I),K=KNWT,NWTT),TSH(I)
341  CONTINUE
        LX = (JNWT)*11
        WRITE (IO,1440) (DASH,L=1,LX)
        IF(NWT.LE.5) WRITE(IO,1447) (TTSH(K),K=KNWT,NWTT+1)

```

```

      IF(NWT.GT.5.AND.NWTT.EQ.5)
+ WRITE(IO,1447) (TTSH(K),K=KNWT,NWTT)
      IF(NWT.GT.5.AND.NWTT.GT.5) WRITE(IO,1447) (TTSH(K),K=KNWT,NWTT+1)
      IF(NWTT-1.EQ.NWT) GOTO 9050
      IF(NWTT.EQ.NWT) GOTO 9050
      KNWT=NWTT+1
      NWTT=KNWT+4
      IF(NWTT.GT.NWT) NWTT=NWT
      JNWT=NWTT-KNWT+1
      IF(NWTT.EQ.NWT) JNWT=JNWT+1
      GOTO 9040
9050 CONTINUE
8040 IF (TABLES(IT).EQ.4) IT = IT+1
C
C   TABLE 5
C
      IF (TABLES(IT).NE.5) GOTO 8050
      KB = 1
      KE = NWT
      IFLAG = 0
      M = 0
825 CONTINUE
      DO 901 K =KB,KE
      M = M+1
      WRITE(IO,1000)
      WRITE(IO,1055) TITLE
      WRITE(IO,1001) DA
      WRITE(IO,1002) TI
      IF(K.NE.NWP1) WRITE(IO,1186) NTA(M),(WSTYP(L,K),L=1,3)
      IF(K.NE.NWP1) WRITE(IO,1187) (RPTYP(L),L=1,4)
      IF(K.EQ.NWP1) WRITE(IO,1225) NTA(M),(RPTYP(L),L=1,4)
      IF(K.EQ.NWP1) WRITE(IO,1336)
      WRITE(IO,1188)
      WRITE(IO,1189)
      WRITE(IO,1190)
      WRITE(IO,1191)
      WRITE(IO,1010)
      IYR = IYRCON - 1
      TA = 0.0
      NTB = 0
      TC = 0.0
      DO 900 I=1,LY
      ASHIFT = SRRMR(I)
      IF(K.EQ.NWP1) GO TO 830
      IF(KIND(K).EQ.2) ASHIFT = SRRMC(I)
      GO TO 840
830 CONTINUE
      ASHIFT = AMAX1(SRRMR(I), SRRMC(I))
840 CONTINUE

```



```

      TA = TA+RECUNT(K,I)
      NTB = NTB+NUMPKG(K,I)
      TC = TC+RMFILD(K,I)
      IYR = IYR + 1
      IF (K.NE.NWP1)
+ WRITE(IO,1350) IYR,ASHIFT,RECUNT(K,I),NUMPKG(K,I),WOPLBR(K,I),
1 RMFILD(K,I)
      IF (K.EQ.NWP1)
1 WRITE(IO,1192) IYR,ASHIFT,RECUNT(K,I),NUMPKG(K,I),WOPLBR(K,I),
1 TRNSPT(K,I),EMPMCH(K,I),EXMACH(K,I),RMFILD(K,I)
900 CONTINUE
      WRITE(IO,1307) TA,NTB,TC
      IF (K.EQ.NWP1) GOTO 901
      IF(MODFLG(1).EQ.1.AND.KIND(K).EQ.1.OR.KIND(K).EQ.3.AND.
* MODFLG(1).EQ.1) WRITE(IO,1560)
      IF(MODFLG(3).EQ.1.AND.KIND(K).EQ.1.OR.KIND(K).EQ.3.AND.
* MODFLG(3).EQ.1) WRITE(IO,1565)
      IF(MODFLG(2).EQ.1.AND.KIND(K).EQ.2) WRITE(IO,1570)
      IF(MODFLG(4).EQ.1.AND.KIND(K).EQ.2) WRITE(IO,1575)
901 CONTINUE
      IF(IFLAG.EQ.1) GO TO 850
      KB = NWP1
      KE = NWP1
      IFLAG = 1
      GO TO 825
850 CONTINUE
8050 IF (TABLES(IT).EQ.5) IT = IT+1
C
C TABLE 6
C
      IF (TABLES(IT).NE.6) GOTO 8060
      KNWT=1
      NWTT=NWT
      IF(NWT.GT.5) NWTT=5
      JNWT = NWTT-KNWT+1
9070 CONTINUE
      DO 501 I=KNWT,NWTT
      NPKG(I) = 0
501 CONTINUE
      WRITE(IO,1000)
      WRITE(IO,1055) TITLE
      WRITE(IO,1001) DA
      WRITE(IO,1002) TI
      WRITE(IO,1167) (RPTYP(L),L=1,4)
      WRITE(IO,1168)
      WRITE(IO,1169) ((WSTYP(L,K),L=1,3), K=KNWT,NWTT)
      LX = JNWT * 20
      WRITE(IO,1170) (DASH,L=1,LX)
      WRITE(IO,1171) (BLANK,L=KNWT,NWTT)

```

```

WRITE(IO,1172) (BLANK,L=KNWT,NWTT)
WRITE(IO,1173) (BLANK,L=KNWT,NWTT)
LX = JNWT * 19
DO 505 K=1,NWT
DO 505 I=1,LY
DHPPKG(K,I) = TDWI(I,K) * WUPP(K)
505 CONTINUE
WRITE(IO,1174) (DASH,L=1,LX)
WRITE(IO,1010)
IYR = IYRCON - 1
DO 500 I=1,LY
DO 510 J =KNWT,NWTT
NPKGT(J) = NPKGT(J)+NUPPKG(J,I)
510 CONTINUE
IYR = IYR + 1
WRITE(IO,1175) IYR,(NUPPKG(K,I),DHPPKG(K,I),K=KNWT,NWTT)
500 CONTINUE
LX = JNWT*9
WRITE (IO,1360) (DASH,L=1,LX)
WRITE(IO,1308) (NPKGT(J),J=KNWT,NWTT)
IF(NWTT.EQ.NWT) GOTO 9080
KNWT=NWTT+1
NWTT=KNWT+4
IF(NWTT.GT.NWT) NWTT=NWT
JNWT=NWTT-KNWT+1
GOTO 9070
9080 CONTINUE
8060 IF (TABLES(IT).EQ.6) IT = IT+1
C
C TABLE 7
C
IF (TABLES(IT).NE.7) GOTO 8070
KNWT=1
NWTT=NWT
IF(NWT.GT.4) NWTT=4
JNWT = NWTT-KNWT+1
9090 CONTINUE
WRITE(IO,1000)
WRITE(IO,1055) TITLE
WRITE(IO,1001) DA
WRITE(IO,1002) TI
WRITE(IO,1176) (RPTYP(L),L=1,4)
WRITE(IO,1177)
WRITE(IO,1180) ((WSTYP(L,K),L=1,3),K=KNWT,NWTT)
LX = JNWT * 24
WRITE(IO,1181) (DASH,L=1,LX)
WRITE(IO,1182) (BLANK,L=KNWT,NWTT)
WRITE(IO,1183) (BLANK,L=KNWT,NWTT)
LX = JNWT * 22

```

```

WRITE(IO,1184) (DASH,L=1,LX)
WRITE(IO,1010)
IYR = IYRCON - 1
DO 800 I=1,LY
IYR = IYR + 1
WRITE(IO,1185) IYR,(PITCH(K,I),NPPRW(K,I),NPKGPR(K,I),K=KNWT,NWTT
+)
800 CONTINUE
IF(NWTT.EQ.NWT) GOTO 9100
KNWT=NWTT+1
NWTT=KNWT+3
IF(NWTT.GT.NWT) NWTT=NWT
JNWT=NWTT-KNWT+1
GOTO 9090
9100 CONTINUE
8070 IF (TABLES(IT).EQ.7) IT = IT+1
C
C
C TABLE 8
C
DO 620 K=1,NWT
TSFMAS(K) = 0.0
THDMAS(K) = 0.0
DO 605 I = 1,LY
RKMSM(K,I) = RKMSM(K,I) / ONETHO
RKBKF(K,I) = RKBKF(K,I) / ONETHO
RCKHST(K,I) = RCKHST(K,I) / ONETHO
THDMAS(K) = THDMAS(K) + EEXRKM(K,I)/ONETHO
TSFMAS(K) = TSFMAS(K) + SHRKMD(K,I)/ONETHO
605 CONTINUE
LB = LY+1
DO 606 I = LB,LEE
RKBKF(K,I) = RKBKF(K,I) / ONETHO
RCKHST(K,I) = RCKHST(K,I) / ONETHO
606 CONTINUE
IF (TABLES(IT).NE.8) GOTO 8082
WRITE(IO,1000)
WRITE(IO,1055) TITLE
WRITE(IO,1001) DA
WRITE(IO,1002) TI
WRITE(IO,1011) NTA(K),(WSTYP(L,K),L=1,3)
WRITE(IO,1018) (RPTYP(L),L=1,4)
WRITE(IO,1012)
WRITE(IO,1013)
WRITE(IO,1014)
WRITE(IO,1015)
WRITE(IO,1010)
8082 IYR = IYRCON-1
TA = 0.0

```

```

TB = 0.0
TC = 0.0
TD = 0.0
TE = 0.0
TF = 0.0
DO 600 I=1,LY
  IYR = IYR + 1
  TA = TA+RKMSM(K,I)
  TB = TB+RKBKF(K,I)
  TC = TC+RCKHST(K,I)
  TD = TD+RMFILD(K,I+LED)
  TE = TE+RMFILD(K,I)
  IF (TABLES(IT).EQ.8)
+ WRITE(IO,1016) IYR,RKMSM(K,I),RKBKF(K,I),RCKHST(K,I),
1RMFILD(K,I+LED),RMFILD(K,I),PNLREQ(K,I),CAREW(K,I)
600 CONTINUE
  LB = LY + 1
  DO 610 I=LB,LEE
    IYR = IYR + 1
    TB = TB+RKBKF(K,I)
    TC = TC+RCKHST(K,I)
    IF (TABLES(IT).EQ.8)
+ WRITE(IO,1303) IYR,RKBKF(K,I),RCKHST(K,I)
610 CONTINUE
  IF (TABLES(IT).NE.8) GOTO 620
  WRITE(IO,1309) TA,TB,TC,TD,TE
  WRITE(IO,1600)
  WRITE(IO,1620) TSFMAS(K),THDMAS(K)
620 CONTINUE
  IF (TABLES(IT).EQ.8) IT = IT+1
C
C
C   TABLE 9
C
  IF (TABLES(IT).NE.9) GOTO 8090
  IFLG = 0
  WRITE(IO,1000)
  WRITE(IO,1055)  TITLE
  WRITE(IO,1001)  DA
  WRITE(IO,1002)  TI
  WRITE(IO,1017)  (RPTYP(L),L=1,4)
  WRITE(IO,1009)
  WRITE(IO,1019)
  WRITE(IO,1020)
  WRITE(IO,1021)
  WRITE(IO,1022)
  WRITE(IO,1023)
  WRITE(IO,1010)

```

C

```

8090 DO 720 I = 1,LY
      SRKMND(I) = SRKMND(I) / ONETHO
      SRKBKF(I) = SRKBKF(I) / ONETHO
      SRKHST(I) = SRKHST(I) / ONETHO
      RKSURF(I) = RKSURF(I) / ONETHO
      RKOFFS(I) = RKOFFS(I) / ONETHO
720  CONTINUE
      LB = LY+1
      DO 721 I = LB,LEE
        SRKBKF(I) = SRKBKF(I) / ONETHO
        SRKHST(I) = SRKHST(I) / ONETHO
        RKSURF(I) = RKSURF(I) / ONETHO
721  CONTINUE
      IYR = IYRCON - 1
      TA = 0.0
      TB = 0.0
      TC = 0.0
      TD = 0.0
      TE = 0.0
      TF = 0.0
      DO 700 I=1,LY
        IYR = IYR + 1
        TA = TA + SRKMND(I)
        TB = TB + SRKBKF(I)
        TC = TC + SRKHST(I)
        TD = TD + RKOFFS(I)
        TE = TE + ANNEX(I)
        TF = TF + ANFILD(I)
        IF (IHSTFL(I).EQ.1) GOTO 725
        IF (TABLES(IT).EQ.9)
          + WRITE(IO,1024) IYR,SRKMND(I),SRKBKF(I),SRKHST(I),RKSURF(I),
          1 RKOFFS(I),ANNEX(I),ANFILD(I),ANPAR(I),CUMPAR(I),
          2 AVLRPA(I)
        GO TO 700
725  CONTINUE
      IFLG = 1
      IF (TABLES(IT).EQ.9)
        + WRITE(IO,1390) IYR,SRKMND(I),SRKBKF(I),SRKHST(I),RKSURF(I),
        1 RKOFFS(I),ANNEX(I),ANFILD(I),ANPAR(I),CUMPAR(I),
        2 AVLRPA(I)
700  CONTINUE
      DO 710 I=LB,LEE
        IYR = IYR + 1
        TB = TB + SRKBKF(I)
        TC = TC + SRKHST(I)
        IF (TABLES(IT).EQ.9)
          + WRITE(IO,1304) IYR,SRKBKF(I),SRKHST(I),RKSURF(I)
710  CONTINUE
      IF (TABLES(IT).NE.9) GOTO 8100

```

```

WRITE(IO,1025) TA,TB,TC,TD,TE,TF
IF(IFLG.EQ.1) WRITE(IO,1400)
WRITE(IO,1600)
HDMAS = HDMAS/ONETHO
SFMAS = SFMAS/ONETHO
WRITE(IO,1620) SFMAS,HDMAS
IT = IT+1

C
C   TABLE 10
C
8100 IF (TABLES(IT).EQ.10) WRITE (IO,186)
186  FORMAT('+')
      WRITE(IO,186)
      RHCC = ROCST(2)/ONEMIL
      DO 185 K=1,NWT
      DO 185 I=1,LL
      RHCC = RHCC + (RKHLBR(K,I) * OVRHDF * ROLBRC(2) + RKHMAT(K,I)
+ * ROMATC(2) ) / ONEMIL
185  CONTINUE
      DO 180 L = 1,NST
      SFTSC(L) = SFTSC(L) / ONEMIL
1 * SFTNUM(L)
      SFTWC(L) = SFTWC(L) / ONEMIL
1 * SFTNUM(L)
      SFTLC(L) = SFTLC(L) / ONEMIL
1 * SFTNUM(L)
      SFTHC(L) = SFTHC(L) / ONEMIL
1 * SFTNUM(L)
      SFTSS(L) = SFTSS(L) / ONEMIL
1 * SFTNUM(L)
      SHDFC(L) = SHDFC(L) / ONEMIL
1 * SFTNUM(L)
      SFCTOT(1) = SFCTOT(1) + SFTSC(L)
      SFCTOT(2) = SFCTOT(2) + SFTWC(L)
      SFCTOT(3) = SFCTOT(3) + SFTLC(L)
      SFCTOT(4) = SFCTOT(4) + SFTHC(L)
      SFCTOT(5) = SFCTOT(5) + SFTSS(L)
      SFCTOT(6) = SFCTOT(6) + SHDFC(L)
180 CONTINUE
C
      SFWSC = SFTSC(3) + SFTWC(3) + SFTLC(3) + SFTSC(4) + SFTWC(4)
+ + SFTLC(4) + SFTSS(3) + SFTSS(4)
      SFWSE = SFTHC(3) + SFTHC(4)
      SFMMC = SFTSC(1) + SFTWC(1) + SFTLC(1) + SFTSS(1)
      SFRHC = SFTSC(7) + SFTWC(7) + SFTLC(7) + SFTSS(7) + RHCC
      SFVSC = SFTSC(2) + SFTWC(2) + SFTLC(2) + SFTSS(2)
      SFEXC = SFTSC(6) + SFTWC(6) + SFTLC(6) + SFTSS(6)
      SFMXC = SFTSC(5) + SFTWC(5) + SFTLC(5) + SFTSS(5)
      IF (TABLES(IT).NE.10) GOTO 8104

```

```

WRITE(IO,1000)
WRITE(IO,1055)  TITLE
WRITE(IO,1001)  DA
WRITE(IO,1002)  TI
WRITE(IO,1123)
WRITE(IO,1147)
WRITE(IO,1124)
WRITE(IO,1125)
WRITE(IO,1126)
WRITE(IO,1127)
WRITE(IO,1128)  (SFTDD(L),L=1,7)
WRITE(IO,1129)  (SFTXR(L),L=1,7)
WRITE(IO,1130)  (SFTXD(L),L=1,7)
WRITE(IO,1131)  (SFTLD(L),L=1,7)
WRITE(IO,1132)
8104  NTPS = MTRPSD + MTCPSD
      IF (TABLES(IT).NE.10) GOTO 8105
      WRITE(IO,1133) MTRPSD,MTCPSD,NTPS
      WRITE(IO,1134)
8105  NPKPTT = 0.0
      IF (TABLES(IT).NE.10) GOTO 8106
      DO 200 K=1,NWT
        IF (SFTNUM(4).EQ.0)
+ WRITE(IO,1148)  (WSTYP(L,K),L=1,3),NPKPTR(K)
        IF (SFTNUM(4).GT.0)
+ WRITE(IO,1380)  (WSTYP(L,K),L=1,3),NPKPTC(K)
200  CONTINUE
8106  IF(SFTNUM(7).GT.0) NTPSMP = MXMPRH
      IF(SFTNUM(7).GT.0) TONPT = TNPT1
      IF (TABLES(IT).NE.10) GOTO 8108
      WRITE(IO,1135)
      WRITE(IO,1149) MXMMRH,NTPSMP
      WRITE(IO,1136) TNPT2,TONPT
8108  DO 210 L=1,7
      NSFTT = NSFTT + SFTNUM(L)
210  CONTINUE
      IF (TABLES(IT).NE.10) GOTO 8110
      WRITE(IO,1138)  (SFTNUM(L),L=1,7),NSFTT
      WRITE(IO,1139)  IYRDIS
      WRITE(IO,1140)  (SFTSC(L),L=1,7), SFCTOT(1)
      WRITE(IO,1141)  (SFTWC(L),L=1,7), SFCTOT(2)
      WRITE(IO,1142)  (SFTLC(L),L=1,7), SFCTOT(3)
      WRITE(IO,1143)  (SFTHE(L),L=1,7), SFCTOT(4)
      WRITE(IO,1144)  (SFTSS(L),L=1,7), SFCTOT(5)
      WRITE(IO,1146)
      WRITE(IO,1145)  (SHDFC(L),L=1,7), SFCTOT(6)
      IT = IT +1
C
C  TABLE 11

```

```

C
8110 TCRHRM = TCRHRM / ONEMIL
      TCCHRM = TCCHRM / ONEMIL
      TCRHPM = TCRHPM / ONEMIL
      TTCAPC = TTCAPC / ONEMIL
      TECAPC = TECAPC / ONEMIL
      TUCAPC = TUCAPC / ONEMIL
      IF (TABLES(IT).NE.11) GOTO 8112
      WRITE(IO,1000)
      WRITE(IO,1055)  TITLE
      WRITE(IO,1001)  DA
      WRITE(IO,1002)  TI
      WRITE(IO,1065)  (RPTYP(L),L=1,4)
      WRITE(IO,1066)
      WRITE(IO,1067)  IYRDIS
      WRITE(IO,1068)
      WRITE(IO,1069)  TCRHRM
      WRITE(IO,1070)  TCCHRM
      WRITE(IO,1071)  TCRHPM
      WRITE(IO,1301)  TTCAPC
      WRITE(IO,1302)  TECAPC
      WRITE(IO,1305)  TUCAPC
8112 DO 150 L=1,NSC
      SFCST(L) = SFCST(L) / ONEMIL
      IF (TABLES(IT).EQ.11) WRITE(IO,1076) (SFTITL(N,L),N=1,12),SFCST(L)
      SSFCST = SSFCST + SFCST(L)
150  CONTINUE
      IF (TABLES(IT).EQ.11) WRITE(IO,1090)
      TOTSFC = TOTSFC + TCRHRM + TCCHRM + TCRHPM + TTCAPC + TECAPC +
1      TUCAPC + SSFCST
      IF (TABLES(IT).NE.11) GOTO 8120
      WRITE(IO,1089) TOTSFC
      IT = IT+1

C
C      TABLE 12
C
8120 DO 160 L = 1,3
      TWEC(L) = TWEC(L) / ONEMIL
160  CONTINUE
      DO 164 L = 1,2
      TTEC(L) = TTEC(L) / ONEMIL
      TUEC(L) = TUEC(L) / ONEMIL
164  CONTINUE
      DO 166 L = 1,5
      TREC(L) = TREC(L) / ONEMIL
166  CONTINUE
      IF (TABLES(IT).NE.12) GOTO 8122
      TEQPT = TEQPT / ONEMIL
      WRITE(IO,1000)

```



```

WRITE(IO,1055) TITLE
WRITE(IO,1001) DA
WRITE(IO,1002) TI
WRITE(IO,1091) (RPTYP(L),L=1,4)
WRITE(IO,1092)
WRITE(IO,1093)
WRITE(IO,1094)
WRITE(IO,1095)
WRITE(IO,1096)
WRITE(IO,1097) WOEQPL(1), TWEC(1)
WRITE(IO,1098) WOEQPL(2), TWEC(2)
WRITE(IO,1099) WOEQPL(3), TWEC(3)
WRITE(IO,1100)
WRITE(IO,1101) TOEQPL(1),TTEC(1)
WRITE(IO,1102) TOEQPL(2),TTEC(2)
WRITE(IO,1103) UOEQPL(1),TUEC(1)
WRITE(IO,1104) UOEQPL(2),TUEC(2)
WRITE(IO,1105)
WRITE(IO,1106) ROEQPL(1), TREC(1)
WRITE(IO,1108) ROEQPL(2), TREC(2)
WRITE(IO,1109) ROEQPL(3), TREC(3)
WRITE(IO,1110) ROEQPL(4), TREC(4)
WRITE(IO,1111) ROEQPL(5), TREC(5)
WRITE(IO,1112)
8122 DO 170 L=1,NSC
      TSFEC(L) = TSFEC(L) / ONEMIL
      TEQPT = TEQPT+TSFEC(L)
      IF (TABLES(IT).EQ.12)
+ WRITE(IO,1113) (SFTITL(N,L),N=1,12), SFEQPL(L), TSFEC(L)
170 CONTINUE
      IF (TABLES(IT).EQ.12) WRITE(IO,1077)
      DO 175 L=1,NEE
      TEEEC(L) = TEEEC(L) / ONEMIL
175 CONTINUE
      IF (TABLES(IT).NE.12) GOTO 8130
      WRITE(IO,1107) EEEQPL(1),TEEEEC(1)
      WRITE(IO,1114) EEEQPL(2),TEEEEC(2)
      WRITE(IO,1115) EEEQPL(3),TEEEEC(3)
      WRITE(IO,1330)
      WRITE(IO,1122) TEQPT
      IT = IT+1
C
C      TABLE 13
C
8130 TA = 0.0
      TB = 0.0
      TC = 0.0
      TD = 0.0
      TE = 0.0

```

```

      TF = 0.0
      TG = 0.0
      TH = 0.0
      TTI = 0.0
      IF (TABLES(IT).NE.13) GOTO 8132
      WRITE(IO,1000)
      WRITE(IO,1055)  TITLE
      WRITE(IO,1001)  DA
      WRITE(IO,1002)  TI
      WRITE(IO,1056)  (RPTYP(L),L=1,4)
      WRITE(IO,1057)
      WRITE(IO,1058)  IYRDIS
      WRITE(IO,1059)
      WRITE(IO,1060)
      WRITE(IO,1061)
      WRITE(IO,1062)
      WRITE(IO,1063)
8132  IYR = IYRCON - 1
      DO 100 I=1,LL
      CSTLND(I) = CSTLND(I) / ONEMIL
      CSTLIC(I) = CSTLIC(I) / ONEMIL
      CSTSHT(I) = CSTSHT(I) / ONEMIL
      CSTUGF(I) = CSTUGF(I) / ONEMIL
      CSTSP (I) = CSTSP (I) / ONEMIL
      CSTSF (I) = CSTSF (I) / ONEMIL
      CSTAE (I) = CSTAE (I) / ONEMIL
      CSTSU (I) = CSTSU (I) / ONEMIL
      CSTOT(I) = CSTLND(I) + CSTLIC(I) + CSTSP(I) + CSTSHT(I)+CSTSF(I)
1+CSTUGF(I)+CSTAE(I)+CSTSU(I)
      TA = TA+CSTLND(I)
      TB = TB+CSTLIC(I)
      TC = TC+CSTSP(I)
      TD = TD+CSTSHT(I)
      TE = TE+CSTSF(I)
      TF = TF+CSTUGF(I)
      TG = TG+CSTAE(I)
      TH = TH+CSTSU(I)
      TTI = TTI+CSTOT(I)
      IYR = IYR + 1
      IF (TABLES(IT).EQ.13)
+ WRITE(IO,1064) IYR,CSTLND(I), CSTLIC(I), CSTSP(I), CSTSHT(I),
1  CSTSF(I),CSTUGF(I),CSTAE(I),CSTSU(I),CSTOT(I)
100 CONTINUE
      LB = LL + 1
      DO 110 I=LB,LY
      IYR = IYR +1
      CSTSF(I) = CSTSF(I) / ONEMIL
      CSTOT(I) = CSTSF(I)
      TTI = TTI+CSTSF(I)

```

```

      TE = TE+CSTSF(I)
      IF (TABLES(IT).EQ.13)
+ WRITE(IO,1300) IYR,CSTSF(I),CSTSF(I)
110 CONTINUE
      IF (TABLES(IT).NE.13) GOTO 8140
      WRITE(IO,1310) TA,TB,TC,TD,TE,TF,TG,TH,TTI
      IT = IT+1
C
C      TABLE 14
C
8140 DO 280 I=1,LEE
      RRCMIN(I) = RRCMIN(I) / ONEMIL
      RRCHST(I) = RRCHST(I) / ONEMIL
      RRCHAS(I) = RRCHAS(I) / ONEMIL
      RRCOSD(I) = RRCOSD(I) / ONEMIL
      RRCTOT(I) = RRCTOT(I) / ONEMIL
280 CONTINUE
      IF (TABLES(IT).NE.14) GOTO 8142
      WRITE(IO,1000)
      WRITE(IO,1055) TITLE
      WRITE(IO,1001) DA
      WRITE(IO,1002) TI
      WRITE(IO,1150) (RPTYP(L),L=1,4)
      WRITE(IO,1151) IYRDIS
      WRITE(IO,1153)
      WRITE(IO,1154)
      WRITE(IO,1155)
      WRITE(IO,1156)
      WRITE(IO,1010)
8142 IYR = IYRCON - 1
      DO 300 I=1,LEE
      RRCTOT(I) = RRCMIN(I)+RRCHST(I)+RRCHAS(I)+ RRCOSD(I)
      IYR = IYR + 1
      IF (TABLES(IT).EQ.14)
+ WRITE(IO,1157) IYR,RRCMIN(I),RRCHST(I),RRCHAS(I),RRCOSD(I),
1 RRCTOT(I)
      TRCC(1) = TRCC(1) + RRCMIN(I)
      TRCC(2) = TRCC(2) + RRCHST(I)
      TRCC(3) = TRCC(3) + RRCHAS(I)
      TRCC(4) = TRCC(4) + RRCOSD(I)
      TRCC(5) = TRCC(5) + RRCTOT(I)
300 CONTINUE
      IF (TABLES(IT).NE.14) GOTO 8150
      WRITE(IO,1121)
      WRITE(IO,1158) (TRCC(I),I=1,5)
      WRITE(IO,1610)
      IT = IT+1
C
C      TABLE 15

```

```

C
8150 IF (TABLES(IT).NE.15) GOTO 8152
      WRITE(IO,1000)
      WRITE(IO,1055)  TITLE
      WRITE(IO,1001)  DA
      WRITE(IO,1002)  TI
      WRITE(IO,1208)  (RPTYP(L),L=1,4)
      WRITE(IO,1209)
      WRITE(IO,1196)  IYRDIS
      WRITE(IO,1197)
      WRITE(IO,1226)
      WRITE(IO,1199)
      WRITE(IO,1010)
8152 IYR = IYRCON - 1
      TA = 0.0
      TB = 0.0
      TC = 0.0
      TD = 0.0
      TE = 0.0
      TF = 0.0
      TG = 0.0
      DO 1205 I = 1,LY
        ARLBCT(I) = ARLBCT(I)/ONEMIL
        APLBCT(I) = APLBCT(I)/ONEMIL
        ATLBCT(I) = ATLBCT(I)/ONEMIL
        AXLBCT(I) = AXLBCT(I)/ONEMIL
        AMLBCT(I) = AMLBCT(I)/ONEMIL
        OWLBR(I) = OWLBR(I)/ONEMIL
        TALBCT(I) = TALBCT(I)/ONEMIL
        TA = TA+ARLBCT(I)
        TB = TB+APLBCT(I)
        TC = TC+ATLBCT(I)
        TD = TD+AXLBCT(I)
        TE = TE+AMLBCT(I)
        TF = TF+OWLBR(I)
        TG = TG+TALBCT(I)
        IYR = IYR + 1
        IF (TABLES(IT).EQ.15)
+ WRITE(IO,1200) IYR,ARLBCT(I),APLBCT(I),ATLBCT(I),AXLBCT(I),
1 AMLBCT(I),OWLBR(I),TALBCT(I)
1205 CONTINUE
      IF (TABLES(IT).NE.15) GOTO 8154
      WRITE(IO,1314)
      WRITE(IO,1315) TA,TB,TC,TD,TE,TF,TG
      IT = IT+1
8154 CLABOR(1) = TA
      CLABOR(2) = TC
      CLABOR(4) = TD
      CLABOR(5) = TE

```

```

C
C   TABLE 16
C
      IF (TABLES(IT).NE.16) GOTO 8160
      WRITE(IO,1000)
      WRITE(IO,1055)  TITLE
      WRITE(IO,1001)  DA
      WRITE(IO,1002)  TI
      WRITE(IO,1212)  (RPTYP(L),L=1,4)
      WRITE(IO,1209)
      WRITE(IO,1196)  IYRDIS
      WRITE(IO,1197)
      WRITE(IO,1226)
      WRITE(IO,1199)
      WRITE(IO,1010)
8160  IYR = IYRCON - 1
      TA = 0.0
      TB = 0.0
      TC = 0.0
      TD = 0.0
      TE = 0.0
      TE = 0.0
      TF = 0.0
      TG = 0.0
      DO 1206 I = 1,LY
      REMTCT(I) = REMTCT(I)/ONEMIL
      PAMTCT(I) = PAMTCT(I)/ONEMIL
      TRMTCT(I) = TRMTCT(I)/ONEMIL
      EXMTCT(I) = EXMTCT(I)/ONEMIL
      EMMTCT(I) = EMMTCT(I) / ONEMIL
      OTMTCT(I) = OTMTCT(I)/ONEMIL
      TMTCT(I) = TMTCT(I)/ONEMIL
      TA = TA+REMTCT(I)
      TB = TB+PAMTCT(I)
      TC = TC+TRMTCT(I)
      TD = TD+EXMTCT(I)
      TE = TE+EMMTCT(I)
      TF = TF+OTMTCT(I)
      TG = TG+TMTCT(I)
      IYR = IYR + 1
      IE (TABLES(IT).EQ.16)
      + WRITE(IO,1200) IYR,REMTCT(I),PAMTCT(I),TRMTCT(I),EXMTCT(I),
      1 EMMTCT(I),OTMTCT(I), TMTCT(I)
1206  CONTINUE
      IF (TABLES(IT).NE.16) GOTO 8162
      WRITE(IO,1314)
      WRITE(IO,1315) TA,TB,TC,TD,TE,TF,TG
      IT = IT+1
8162  CMATL(1) = TA

```

```

      CMATL(2) = TC
      CMATL(4) = TD
      CMATL(5) = TE
C
C      TABLE 17
C
      IF (TABLES(IT).NE.17) GOTO 8170
      WRITE(IO,1000)
      WRITE(IO,1055)  TITLE
      WRITE(IO,1001)  DA
      WRITE(IO,1002)  TI
      WRITE(IO,1213)  (RPTYP(L),L=1,4)
      WRITE(IO,1209)
      WRITE(IO,1196)  IYRDIS
      WRITE(IO,1197)
      WRITE(IO,1228)
      WRITE(IO,1199)
      WRITE(IO,1010)
8170  IYR = IYRCON - 1
      TA = 0.0
      TB = 0.0
      TC = 0.0
      TD = 0.0
      TE = 0.0
      TF = 0.0
      TG = 0.0
      DO 1207 I = 1,LY
      REEQCT(I) = REEQCT(I)/ONEMIL
      PARQCT(I) = PARQCT(I)/ONEMIL
      TREQCT(I) = TREQCT(I)/ONEMIL
      EXEQCT(I) = EXEQCT(I)/ONEMIL
      EMEQCT(I) = EMEQCT(I)/ONEMIL
      SFEQCT(I) = SFEQCT(I)/ONEMIL
      TWOEQC(I) = TWOEQC(I)/ONEMIL
      TA = TA+REEQCT(I)
      TB = TB+PARQCT(I)
      TC = TC+TREQCT(I)
      TD = TD+EXEQCT(I)
      TE = TE+EMEQCT(I)
      TF = TF+SFEQCT(I)
      TG = TG+TWOEQC(I)
      IYR = IYR + 1
      IF (TABLES(IT).EQ.17)
+ WRITE(IO,1200) IYR,REEQCT(I),PARQCT(I),TREQCT(I),EXEQCT(I),
1 EMEQCT(I),SFEQCT(I),TWOEQC(I)
1207 CONTINUE
      IF (TABLES(IT).NE.17) GOTO 8172
      WRITE(IO,1063)
      WRITE(IO,1314)

```

```

WRITE(IO,1315) TA,TB,TC,TD,TE,TF,TG
IT = IT+1
8172 CEQPT(1) = TA
      CEQPT(2) = TC
      CEQPT(4) = TD
      CEQPT(5) = TE
C
C   TABLE 18
C
      IF (TABLES(IT).NE.18) GOTO 8180
      WRITE(IO,1000)
      WRITE(IO,1055)  TITLE
      WRITE(IO,1001)  DA
      WRITE(IO,1002)  TI
      WRITE(IO,1279)  (RPTYP(L),L=1,4)
      WRITE(IO,1296)
      WRITE(IO,1280)
      WRITE(IO,1281)
      WRITE(IO,1282) IYRDIS
      WRITE(IO,1283)
      WRITE(IO,1284)
      WRITE(IO,1297)
8180 IYR = IYRCON -1
      NN = NWP1
      TA = 0.0
      TB = 0.0
      TC = 0.0
      TD = 0.0
      TE = 0.0
      TF = 0.0
      TG = 0.0
      TH = 0.0
      TTI = 0.0
      DO 750 I=1,LY
      IYR = IYR + 1
      POWER (NN,I) = POWER (NN,I) / ONETHO
      STEAM (NN,I) = STEAM (NN,I) / ONETHO
      DIESEL(NN,I) = DIESEL(NN,I) / ONETHO
      POWCST(NN,I) = POWCST(NN,I) / ONEMIL
      STMCST(NN,I) = STMCST(NN,I) / ONEMIL
      WATERC(NN,I) = WATERC(NN,I) / ONEMIL
      DSLCST(NN,I) = DSLCST(NN,I) / ONEMIL
      ANUTC (NN,I) = ANUTC (NN,I) / ONEMIL
      TA = TA+POWER(NN,I)
      TB = TB+STEAM(NN,I)
      TC = TC+WATER(NN,I)
      TD = TD+DIESEL(NN,I)
      TE = TE+POWCST(NN,I)
      TF = TF+STMCST(NN,I)

```

```

      TG = TG+WATERC(NN,I)
      TH = TH+DSLCST(NN,I)
      TTI = TTI+ANUTC(NN,I)
      IF (TABLES(IT).EQ.18)
+ WRITE(IO,1285) IYR,POWER(NN,I),STEAM(NN,I),WATER(NN,I),DIESEL(NN
1,I),POWCST(NN,I),STMCST(NN,I),WATERC(NN,I),DSLCST(NN,I),ANUTC(NN,I
+)
750 CONTINUE
      WRITE(IO,1063)
      IF (TABLES(IT).NE.18) GOTO 8190
      WRITE(IO,1317)
      WRITE(IO,1318) TA,TB,TC,TD,TE,TF,TG,TH,TTI
      WRITE(IO,1286)
      IT = IT+1
C
C
C
C
C      TABLE 19
8190 IFLAG = 0
      KB = 1
      KE = NWT
      M = 0
1201 CONTINUE
      DO 1204 K=KB,KE
      M = M+1
      IF (TABLES(IT).NE.19) GOTO 8192
      WRITE(IO,1000)
      WRITE(IO,1055) TITLE
      WRITE(IO,1001) DA
      WRITE(IO,1002) TI
      IF(K.NE.NWP1) WRITE(IO,1193) NTA(M),(WSTYP(L,K),L=1,3)
      IF(K.EQ.NWP1) WRITE(IO,1194) NTA(M),(RPTYP(L),L=1,4)
      IF(K.NE.NWP1) WRITE(IO,1178) (RPTYP(L),L=1,4)
      IF(K.EQ.NWP1) WRITE(IO,1335)
      WRITE(IO,1196) IYRDIS
      WRITE(IO,1197)
      WRITE(IO,1226)
      WRITE(IO,1199)
      WRITE(IO,1010)
8192 IYR = IYRCON - 1
      TA = 0.0
      TB = 0.0
      TC = 0.0
      TD = 0.0
      TE = 0.0
      TF = 0.0
      TG = 0.0
      DO 1202 I=1,LY

```



```

RECST(K,I) = RECST(K,I)/ONEMIL
PACST(K,I) = PACST(K,I)/ONEMIL
TRCST(K,I) = TRCST(K,I)/ONEMIL
ECOST(K,I) = ECOST(K,I)/ONEMIL
EMCST(K,I) = EMCST(K,I)/ONEMIL
OWOPC(K,I) = OWOPC(K,I) / ONEMIL
TWOPCT(K,I) = TWOPCT(K,I)/ONEMIL
TA = TA+RECST(K,I)
TB = TB+PACST(K,I)
TC = TC+TRCST(K,I)
TD = TD+ECOST(K,I)
TE = TE+EMCST(K,I)
TF = TF+OWOPC(K,I)
TG = TG+TWOPCT(K,I)
IYR = IYR + 1
IF (TABLES(IT).EQ.19)
+ WRITE(IO,1200) IYR,RECST(K,I),PACST(K,I),TRCST(K,I),ECOST(K,I),
1 EMCST(K,I), OWOPC(K,I), TWOPCT(K,I)
1202 CONTINUE
IF (TABLES(IT).NE.19) GOTO 1204
WRITE(IO,1063)
WRITE(IO,1319)
WRITE(IO,1320) TA,TB,TC,TD,TE,TF,TG
1204 CONTINUE
IF(IFLAG.EQ.1) GO TO 1203
KB = NWP1
KE = NWP1
IFLAG = 1
GO TO 1201
1203 CONTINUE
IF (TABLES(IT).EQ.19) IT = IT+1
C
C
C
TABLE 20
DO 520 I=1,LEE
ANBFCT(I) = ANBFCT(I) / ONEMIL
ANSLCT(I) = ANSLCT(I) / ONEMIL
ANDDCT(I) = ANDDCT(I) / ONEMIL
TSURVC(I) = TSURVC(I) / ONEMIL
ABDCST(I) = ANBFCT(I) + ANSLCT(I) + ANDDCT(I) + TSURVC(I)
520 CONTINUE
DO 530 I=1,LEE
SOTBKF(I) = SOTBKF(I) / ONETHO
SUMBKF(I) = SUMBKF(I) / ONETHO
530 CONTINUE
IF (TABLES(IT).NE.20) GOTO 8200
WRITE(IO,1000)
WRITE(IO,1055) TITLE
WRITE(IO,1001) DA

```

```

WRITE(IO,1002) TI
WRITE(IO,1230) (RPTYP(L),L=1,4)
WRITE(IO,1231)
WRITE(IO,1232) IYRDIS
WRITE(IO,1233)
WRITE(IO,1234)
WRITE(IO,1235)
WRITE(IO,1010)
8200 IYR = IYRCON - 1
TA = 0.0
TB = 0.0
TC = 0.0
TD = 0.0
TE = 0.0
TF = 0.0
TG = 0.0
TH = 0.0
DO 550 I=1,LEE
IYR = IYR + 1
TA = TA+SRKBKF(I)
TB = TB+SOTBKF(I)
TC = TC+SUMBKF(I)
TD = TD+ANBFCT(I)
TE = TE+ANSLCT(I)
TF = TF+ANDDCT(I)
TG = TG+TSURVC(I)
TH = TH+ABDCST(I)
IF (TABLES(IT).EQ.20)
+ WRITE(IO,1236) IYR,SRKBKF(I),SOTBKF(I),SUMBKF(I),ANBFCT(I),
1ANSLCT(I),ANDDCT(I), TSURVC(I),ABDCST(I)
550 CONTINUE
IF (TABLES(IT).NE.20) GOTO 8210
WRITE(IO,1063)
WRITE(IO,1322)
WRITE(IO,1324) TA,TB,TC,TD,TE,TF,TG,TH
IT = IT+1

C
C
C
8210 DO 580 K=1,NWP1
TLDCST(K) = TLDCST(K) / ONEMIL
TLCCST(K) = TLCCST(K) / ONEMIL
TSPCST(K) = TSPCST(K) / ONEMIL
TSHCST(K) = TSHCST(K) / ONEMIL
TSFCST(K) = TSFCST(K) / ONEMIL
TUFCST(K) = TUFCST(K) / ONEMIL
TAECST(K) = TAECST(K) / ONEMIL
TSUCST(K) = TSUCST(K) / ONEMIL
TSUBCN(K) = TSUBCN(K) / ONEMIL

```

```

TOMCST(K) = TOMCST(K) / ONEMIL
THSCST(K) = THSCST(K) / ONEMIL
TSSCST(K) = TSSCST(K) / ONEMIL
TODCST(K) = TODCST(K) / ONEMIL
TSUBRO(K) = TSUBRO(K) / ONEMIL
TRCCST(K) = TRCCST(K) / ONEMIL
TPKCST(K) = TPKCST(K) / ONEMIL
TTRCST(K) = TTRCST(K) / ONEMIL
TEXCST(K) = TENCST(K) / ONEMIL
TEMCST(K) = TEMCST(K) / ONEMIL
TUTCST(K) = TUTCST(K) / ONEMIL
TOPCST(K) = TOPCST(K) / ONEMIL
TBFCST(K) = TBFCST(K) / ONEMIL
TSLCST(K) = TSLCST(K) / ONEMIL
TDCCST(K) = TDCCST(K) / ONEMIL
TSVCST(K) = TSVCST(K) / ONEMIL
TBDCST(K) = TBDCST(K) / ONEMIL
TCNCST(K) = TCNCST(K) / ONEMIL
TWTCT(K) = TWTCT(K) / ONEMIL
580  CONTINUE
      KNWT=1
      NWTT=NWT
      IF(NWT.GT.5) NWTT=5
      JNWT = NWTT-KNWT+1
      LOOP=NWP1
      IF(NWT.GT.5) LOOP=NWTT+1
      IF (TABLES(IT).NE.21) GOTO 9120
9110  CONTINUE
      WRITE(IO,1000)
      WRITE(IO,1055)  TITLE
      WRITE(IO,1001)  DA
      WRITE(IO,1002)  TI
      WRITE(IO,1237)  (RPTYP(L),L=1,4)
      WRITE(IO,1238)  IYRDIS
      WRITE(IO,1239)
      LX = (JNWT+1) * 13 -3
      WRITE(IO,1240)  (DASH,L=1,LX)
      IF (NWTT.EQ.NWT) WRITE (IO,1241) ((WSTYP(L,K),L=1,3),K=KNWT,NWTT)
      + ,TOT
      IF (NWTT.NE.NWT) WRITE(IO,1241) ((WSTYP(L,K),L=1,3),K=KNWT,NWTT+1)
      LX = (JNWT+1) * 10
      WRITE(IO,1242)  (DASH,L=1,LX)
      WRITE(IO,1243)
      WRITE(IO,1244)  (TLDCST(K),K=KNWT,NWTT) , TLDCST(LOOP)
      WRITE(IO,1245)  (TLCCST(K),K=KNWT,NWTT), TLCCST(LOOP)
      WRITE(IO,1246)  (TSPCST(K),K=KNWT,NWTT) ,TSPCST(LOOP)
      WRITE(IO,1247)  (TSHCST(K),K=KNWT,NWTT), TSHCST(LOOP)
      WRITE(IO,1249)  (TSFCST(K),K=KNWT,NWTT),TSFCST(LOOP)
      WRITE(IO,1214)  (TUFCT(K),K=KNWT,NWTT), TUFCT(LOOP)

```

```

WRITE(IO,1250) (TAECS(T),K=KNWT,NWTT) ,TAECS(LOOP)
WRITE(IO,1251) (TSUCS(T),K=KNWT,NWTT) ,TSUCS(LOOP)
WRITE(IO,1288) (TSUBCN(K),K=KNWT,NWTT), TSUBCN(LOOP)
WRITE(IO,1292)
WRITE(IO,1252) (TOMCS(T),K=KNWT,NWTT) ,TOMCS(LOOP)
WRITE(IO,1253) (THSCS(T),K=KNWT,NWTT) ,THSCS(LOOP)
WRITE(IO,1254) (TSSCS(T),K=KNWT,NWTT) ,TSSCS(LOOP)
WRITE(IO,1255) (TODCS(T),K=KNWT,NWTT) ,TODCS(LOOP)
WRITE(IO,1288) (TSUBRO(K),K=KNWT,NWTT) ,TSUBRO(LOOP)
WRITE(IO,1293)
WRITE(IO,1256) (TRCCS(T),K=KNWT,NWTT) ,TRCCS(LOOP)
WRITE(IO,1257) (TPKCS(T),K=KNWT,NWTT) ,TPKCS(LOOP)
WRITE(IO,1258) (TTRCS(T),K=KNWT,NWTT) ,TTRCS(LOOP)
WRITE(IO,1291) (TEXCS(T),K=KNWT,NWTT) ,TEXCS(LOOP)
WRITE(IO,1259) (TEMC(S),K=KNWT,NWTT) ,TEMC(S)(LOOP)
WRITE(IO,1260) (TUTC(S),K=KNWT,NWTT) ,TUTC(S)(LOOP)
WRITE(IO,1288) (TOPCS(T),K=KNWT,NWTT) ,TOPCS(LOOP)
WRITE(IO,1294)
WRITE(IO,1261) (TBFC(S),K=KNWT,NWTT) ,TBFC(S)(LOOP)
WRITE(IO,1262) (TSLCS(T),K=KNWT,NWTT) ,TSLCS(LOOP)
WRITE(IO,1263) (TDCCS(T),K=KNWT,NWTT) ,TDCCS(LOOP)
WRITE(IO,1287) (TSVCS(T),K=KNWT,NWTT) ,TSVCS(LOOP)
WRITE(IO,1288) (TBDC(S),K=KNWT,NWTT) ,TBDC(S)(LOOP)
WRITE(IO,1289) (TCNC(S),K=KNWT,NWTT) ,TCNC(S)(LOOP)
WRITE(IO,1290) (TWTCS(T),K=KNWT,NWTT) ,TWTCS(LOOP)
IF(NWTT.EQ.NWT) GOTO 9120
IF(NWT.GT.5) NWTT=NWTT+1
KNWT=NWTT+1
NWTT=KNWT+4
IF(NWTT.GT.NWT) NWTT=NWT
LOOP=NWP1
IF(NWTT.GT.NWT) LOOP=NWTT+1
JNWT=NWTT-KNWT+1
GOTO 9110
9120 CONTINUE
IF (TABLES(IT).EQ.21) IT = IT+1
C
C
C
C
TABLE 22
IF (TABLES(IT).NE.22) GOTO 8220
WRITE(IO,1000)
IF(MSWTCH.EQ.1) GOTO 645
WRITE(IO,640)
640 FORMAT(' ROCK HOISTING COST CALCULATION OPTION MUST',
+ ' BE SET TO 1 (CALCULATE USING RESOURCE REQUIREMENTS COST)'/
+ ' TO OBTAIN CORRECT FIGURES FOR TABLE 22.',
+ ' TABLE 22 IS NOT PRINTED IF THIS OPTION IS NOT 1.')
IF (TABLE(IT).EQ.22) IT=IT+1

```

```

        GOTO 647
645    CONTINUE
        WRITE(IO,1055) TITLE
        WRITE(IO,1001) DA
        WRITE(IO,1002) TI
C
        CALL OUT22
8220   IF (TABLES(IT).EQ.22) IT = IT+1
C
647    CONTINUE
C      TABLE 23
C
        IF (TABLES(IT).NE.23) GOTO 8230
        WRITE(IO,1000)
        WRITE(IO,1055) TITLE
        WRITE(IO,1001) DA
        WRITE(IO,1002) TI
        WRITE(IO,1264) (RPTYP(L),L=1,4)
        WRITE(IO,1265) IYRDIS
        WRITE(IO,1267)
        WRITE(IO,1268)
        WRITE(IO,1269)
        WRITE(IO,1270)
        WRITE(IO,1271)
        WRITE(IO,1010)
8230   TA = 0.0
        TB = 0.0
        TC = 0.0
        TD = 0.0
        TE = 0.0
        TF = 0.0
        IYR = IYRCON - 1
        DO 650 I=1,LY
        IYR = IYR + 1
        ACONTG(I) = ACONTG(I) / ONEMIL
        TACCST(I)=CSTOT(I) + RRCTOT(I) + TWOPCT(NWP1,I) + ABDCST(I)
1 + ACONTG(I)
        TA = TA+CSTOT(I)
        TB = TB+RRCTOT(I)
        TC = TC+TWOPCT(NWP1,I)
        TD = TD+ABDCST(I)
        TE = TE+ACONTG(I)
        TF = TF +TACCST(I)
        IF (TABLES(IT).EQ.23)
+ WRITE(IO,1272) IYR,CSTOT(I),RRCTOT(I),TWOPCT(NWP1,I),ABDCST(I),
1ACONTG(I),TACCST(I)
650    CONTINUE
        IB = LY + 1
        LE = NBKF + IDD

```

```

DO 660 I=IB,LE
IYR = IYR + 1
ACONTG(I) = ACONTG(I) / ONEMIL
TACCST(I) = RRCTOT(I) + ABDCST(I)
1 + ACONTG(I)
TB = TB+RRCTOT(I)
TD = TD+ABDCST(I)
TE = TE+ACONTG(I)
TF = TF+TACCST(I)
IF (TABLES(IT).EQ.23)
+ WRITE(IO,1298) IYR,RRCTOT(I),ABDCST(I),ACONTG(I),TACCST(I)
660 CONTINUE
IF (TABLES(IT).NE.23) GOTO 8240
WRITE(IO,1339)
WRITE(IO,1340) TA,TB,TC,TD,TE,TF
IT = IT+1
C
C
C
TABLE 24
8240 IF (TABLES(IT).NE.24) GOTO 8242
KNWT=1
NWTT=NWT
IF(NWT.GT.5) NWTT=5
JNWT = NWTT-KNWT+1
LOOP=NWP1
IF(NWT.GT.5) LOOP=NWTT+1
9140 CONTINUE
WRITE(IO,1000)
WRITE(IO,1055) TITLE
WRITE(IO,1001) DA
WRITE(IO,1002) TI
WRITE(IO,1273) (RPTYP(L),L=1,4)
WRITE(IO,1274) IYRDIS
WRITE(IO,1239)
LX = JNWT * 13 - 3
WRITE(IO,1240) (DASH,L=1,LX)
WRITE(IO,1241) ((WSTYP(L,K),L=1,3),K=KNWT,NWTT)
LX = JNWT * 10
WRITE(IO,1242) (DASH,L=1,LX)
WRITE(IO,1243)
WRITE(IO,1244) (ULDCST(K,1),K=KNWT,NWTT)
WRITE(IO,1245) (ULCCST(K,1),K=KNWT,NWTT)
WRITE(IO,1246) (USPCST(K,1),K=KNWT,NWTT)
WRITE(IO,1247) (USHCST(K,1),K=KNWT,NWTT)
WRITE(IO,1249) (USFCST(K,1),K=KNWT,NWTT)
WRITE(IO,1214) (UUFCSST(K,1),K=KNWT,NWTT)
WRITE(IO,1250) (UAECST(K,1),K=KNWT,NWTT)
WRITE(IO,1251) (USUCST(K,1),K=KNWT,NWTT)
WRITE(IO,1288) (USUBCN(K,1),K=KNWT,NWTT)

```

```

WRITE(IO,1292)
WRITE(IO,1252) (UOMCST(K,1),K=KNWT,NWTT)
WRITE(IO,1253) (UHSCST(K,1),K=KNWT,NWTT)
WRITE(IO,1254) (USSCST(K,1),K=KNWT,NWTT)
WRITE(IO,1255) (UODCST(K,1),K=KNWT,NWTT)
WRITE(IO,1288) (USUBRO(K,1),K=KNWT,NWTT)
WRITE(IO,1293)
WRITE(IO,1256) (URCCST(K,1),K=KNWT,NWTT)
WRITE(IO,1257) (UPKCST(K,1),K=KNWT,NWTT)
WRITE(IO,1258) (UTRCST(K,1),K=KNWT,NWTT)
WRITE(IO,1291) (UEXCST(K,1),K=KNWT,NWTT)
WRITE(IO,1259) (UEMCST(K,1),K=KNWT,NWTT)
WRITE(IO,1260) (UUTCST(K,1),K=KNWT,NWTT)
WRITE(IO,1288) (UOPCST(K,1),K=KNWT,NWTT)
WRITE(IO,1294)
WRITE(IO,1261) (UBFCST(K,1),K=KNWT,NWTT)
WRITE(IO,1262) (USLCST(K,1),K=KNWT,NWTT)
WRITE(IO,1263) (UDCCST(K,1),K=KNWT,NWTT)
WRITE(IO,1287) (USVCST(K,1),K=KNWT,NWTT)
WRITE(IO,1288) (UBDCST(K,1),K=KNWT,NWTT)
WRITE(IO,1289) (UCNCST(K,1),K=KNWT,NWTT)
WRITE(IO,1290) (UWTCST(K,1),K=KNWT,NWTT)
IF(NWTT.EQ.NWT) GOTO 9150
KNWT=NWTT+1
NWTT=KNWT+4
IF(NWTT.GT.NWT) NWTT=NWT
JNWT=NWTT-KNWT+1
LOOP=NWP1
IF(NWTT.GT.NWT) LOOP=NWTT+1
GOTO 9140
9150 CONTINUE
8242 IF (TABLES(IT).EQ.24) IT = IT+1
C
C TABLE 25
C
IF (TABLES(IT).NE.25) GOTO 8250
KNWT=1
NWTT=NWT
IF(NWT.GT.5) NWTT=5
JNWT = NWTT-KNWT+1
9160 CONTINUE
WRITE(IO,1000)
WRITE(IO,1055) TITLE
WRITE(IO,1001) DA
WRITE(IO,1002) TI
WRITE(IO,1275) (RPTYP(L), L=1,4)
WRITE(IO,1276) IYRDIS
WRITE(IO,1239)
LX = JNWT * 13 -3

```

```

WRITE(IO,1240) (DASH,L=1,LX)
WRITE(IO,1241) ((WSTYP(L,K),L=1,3),K=KNWT,NWTT)
LX = JNWT * 10
WRITE(IO,1242) (DASH,L=1,LX)
WRITE(IO,1243)
WRITE(IO,1244) (ULDCST(K,2),K=KNWT,NWTT)
WRITE(IO,1245) (ULCCST(K,2),K=KNWT,NWTT)
WRITE(IO,1246) (USPCST(K,2),K=KNWT,NWTT)
WRITE(IO,1247) (USHCST(K,2),K=KNWT,NWTT)
WRITE(IO,1249) (USFCST(K,2),K=KNWT,NWTT)
WRITE(IO,1214) (UUF CST(K,2),K=KNWT,NWTT)
WRITE(IO,1250) (UAECST(K,2),K=KNWT,NWTT)
WRITE(IO,1251) (USUCST(K,2),K=KNWT,NWTT)
WRITE(IO,1288) (USUBCN(K,2),K=KNWT,NWTT)
WRITE(IO,1292)
WRITE(IO,1252) (UOMCST(K,2),K=KNWT,NWTT)
WRITE(IO,1253) (UHSCST(K,2),K=KNWT,NWTT)
WRITE(IO,1254) (USSCST(K,2),K=KNWT,NWTT)
WRITE(IO,1255) (UODCST(K,2),K=KNWT,NWTT)
WRITE(IO,1288) (USUBRO(K,2),K=KNWT,NWTT)
WRITE(IO,1293)
WRITE(IO,1256) (URCCST(K,2),K=KNWT,NWTT)
WRITE(IO,1257) (UPKCST(K,2),K=KNWT,NWTT)
WRITE(IO,1258) (UTRCST(K,2),K=KNWT,NWTT)
WRITE(IO,1291) (UEXCST(K,2),K=KNWT,NWTT)
WRITE(IO,1259) (UEMCST(K,2),K=KNWT,NWTT)
WRITE(IO,1260) (UUTCST(K,2),K=KNWT,NWTT)
WRITE(IO,1288) (UOPCST(K,2),K=KNWT,NWTT)
WRITE(IO,1294)
WRITE(IO,1261) (UBFCST(K,2),K=KNWT,NWTT)
WRITE(IO,1262) (USLCST(K,2),K=KNWT,NWTT)
WRITE(IO,1263) (UDCCST(K,2),K=KNWT,NWTT)
WRITE(IO,1287) (USVCST(K,2),K=KNWT,NWTT)
WRITE(IO,1288) (UBDCST(K,2),K=KNWT,NWTT)
WRITE(IO,1289) (UCNCST(K,2),K=KNWT,NWTT)
WRITE(IO,1290) (UWTCST(K,2),K=KNWT,NWTT)
IF(NWTT.EQ.NWT) GOTO 9170
KNWT=NWTT+1
NWTT=KNWT+4
IF(NWTT.GT.NWT) NWTT=NWT
JNWT=NWTT-KNWT+1
GOTO 9160
9170 CONTINUE
8250 IF (TABLES(IT).EQ.25) IT = IT+1
C
C TABLE 26
C
IF (TABLES(IT).NE.26) GOTO 8260
KNWT=1

```



```

NWTT=NWT
IF(NWT.GT.5) NWTT=5
JNWT = NWTT-KNWT+1
9180 CONTINUE
WRITE(IO,1000)
WRITE(IO,1055)  TITLE
WRITE(IO,1001)  DA
WRITE(IO,1002)  TI
WRITE(IO,1277) (RPTYP(L), L=1,4)
WRITE(IO,1278) IYRDIS
WRITE(IO,1239)
LX = JNWT * 13 -3
WRITE(IO,1240) (DASH,L=1,LX)
WRITE(IO,1241) ((WSTYP(L,K),L=1,3),K=KNWT,NWTT)
LX = JNWT * 10
WRITE(IO,1242) (DASH,L=1,LX)
WRITE(IO,1243)
WRITE(IO,1244) (ULDCST(K,3),K=KNWT,NWTT)
WRITE(IO,1245) (ULCCST(K,3),K=KNWT,NWTT)
WRITE(IO,1246) (USPCST(K,3),K=KNWT,NWTT)
WRITE(IO,1247) (USHCST(K,3),K=KNWT,NWTT)
WRITE(IO,1249) (USFCST(K,3),K=KNWT,NWTT)
WRITE(IO,1214) (UUF CST(K,3),K=KNWT,NWTT)
WRITE(IO,1250) (UAECST(K,3),K=KNWT,NWTT)
WRITE(IO,1251) (USUCST(K,3),K=KNWT,NWTT)
WRITE(IO,1288) (USUBCN(K,3),K=KNWT,NWTT)
WRITE(IO,1292)
WRITE(IO,1252) (UOMCST(K,3),K=KNWT,NWTT)
WRITE(IO,1253) (UHSCST(K,3),K=KNWT,NWTT)
WRITE(IO,1254) (USSCST(K,3),K=KNWT,NWTT)
WRITE(IO,1255) (UODCST(K,3),K=KNWT,NWTT)
WRITE(IO,1288) (USUBRO(K,3),K=KNWT,NWTT)
WRITE(IO,1293)
WRITE(IO,1256) (URCCST(K,3),K=KNWT,NWTT)
WRITE(IO,1257) (UPKCST(K,3),K=KNWT,NWTT)
WRITE(IO,1258) (UTRCST(K,3),K=KNWT,NWTT)
WRITE(IO,1291) (UEXCST(K,3),K=KNWT,NWTT)
WRITE(IO,1259) (UEMCST(K,3),K=KNWT,NWTT)
WRITE(IO,1260) (UUTCST(K,3),K=KNWT,NWTT)
WRITE(IO,1288) (UOPCST(K,3),K=KNWT,NWTT)
WRITE(IO,1294)
WRITE(IO,1261) (UBFCST(K,3),K=KNWT,NWTT)
WRITE(IO,1262) (USLCST(K,3),K=KNWT,NWTT)
WRITE(IO,1263) (UDCCST(K,3),K=KNWT,NWTT)
WRITE(IO,1287) (USVCST(K,3),K=KNWT,NWTT)
WRITE(IO,1288) (UBDCST(K,3),K=KNWT,NWTT)
WRITE(IO,1289) (UCNCST(K,3),K=KNWT,NWTT)
WRITE(IO,1290) (UWTCST(K,3),K=KNWT,NWTT)
IF(NWTT.EQ.NWT) GOTO 9190

```

```

      KNWT=NWTT+1
      NWTT=KNWT+4
      IF(NWTT.GT.NWT) NWTT=NWT
      JNWT=NWTT-KNWT+1
      GOTO 9180
9190  CONTINUE
8260  IF (TABLES(IT).EQ.26) IT = IT+1
C
C
C      TABLE 27 COEMPLACEMENT
C
C      IF (TABLES(IT).NE.27) GOTO 8999
      DO 1510 K=1,MPRIME
      WRITE(IO,1500) K
1500  FORMAT('1      COEMPLACEMENT ARRAY COWAST '
+ // ' PRIMARY WASTE TYPE ',I5)
      WRITE(IO,1505) (I,I=1,NNHG)
1505  FORMAT('0','IYR',9(9X,I3))
      IYR = IYRCON-1
      DO 1530 L=1,LY+10
      IYR = IYR + 1
      WRITE(IO,1540) IYR,(COWAST(J,K,L),J=1,NNHG)
1530  CONTINUE
1510  CONTINUE
8999  CONTINUE
1540  FORMAT(' ',I5,5X,9(G12.5))
1000  FORMAT('1')
1001  FORMAT(' ',T111,'DATE; ',A8)
1002  FORMAT(' ',T111,'TIME; ',A8)
1003  FORMAT('0',T34,'TABLE 1 MINE DEVELOPMENT PARAMETERS FOR A REPOSIT
      1ORY IN ',4A4)
1004  FORMAT(' ',T34,7('-'))
1005  FORMAT('LOCATION',T60,4A4)
1006  FORMAT('OYEAR CONSTRUCTION BEGINS',T62,I4)
1007  FORMAT('OYEAR OPERATION BEGINS',T62,I4)
1008  FORMAT('ODEPTH, METERS (A)',T59,F8.1)
1009  FORMAT(' ',T20,7('-'))
1010  FORMAT(' ')
1011  FORMAT('0',T20,'TABLE 8',A1,'. REPOSITORY DEVELOPMENT SCHEDULE FOR
      1 ',3A4)
1012  FORMAT('0',T17,'ROCK, THOUSAND OF TONNES (1)',/
      1 ,T12,40('-'),T67,'ROOMS',T87,'AREA REQUIRED, ACRES')
1013  FORMAT(' ',T42,'TRANSPORT',T57,25('-'),T87,25('-'))
1014  FORMAT(' YEAR',T12,'MINED',T27,'BACKFILLED',T42,'TO SURFACE',
      1T57,'EXCAVATED',T72,'FILLED',T87,'ANNUAL',T102,'CUMULATIVE')
1015  FORMAT(' ----',6X ,7(10('-'),5X))
1016  FORMAT(' ',I4,T11,3(F10.0,5X),4(F10.2,5X))
1017  FORMAT('0',T20,'TABLE 9 SUMMARY DEVELOPMENT SCHEDULE FOR A REPOSI

```

```

1TORY IN ',4A4)
1018 FORMAT(' ',T20,9('-'),2X,'IN ',4A4)
1019 FORMAT('0',T25,'ROCK, THOUSAND OF TONNES (1)',T97,'WASTE STORAGE A
+ REA ACRES')
1020 FORMAT(' ',T11,58('-'),T79,'ROOMS',T95,34('-'))
1021 FORMAT(' ',T35,'TRANSPORT',T47,'SURFACE',T59,'TRANSPORT'
1,T71,22('-'),T95,'ANNUAL',T107,'CUMULATIVE REMAINING')
1022 FORMAT(' YEAR',T11,'MINED',T23,'BACKFILLED TO SURFACE STORAGE',
1T59,'OFFSITE',T71,'EXCAVATED',T83,'FILLED',T95,2('REQUIRED',4X),
2T119,'AVAILABLE')
1023 FORMAT(' ----',T11,10(10('-'),2X))
1024 FORMAT(' ',I4,5X,5(F10.0,2X),5(F10.1,2X))
1025 FORMAT('0',T11,3(10('-'),2X),12X,3(10('-'),2X)//
+ ' TOTAL',T11,3(F10.0,2X),12X,F10.0,2(2X,F10.1))
1026 FORMAT('0TOTAL SURFACE AREA, HECTARES (B)',T59,F8.1)
1027 FORMAT('0TOTAL UNDERGROUND AREA, HECTARES (B)',T59,F8.1)
1028 FORMAT('0MINING METHOD',T51,4A4)
1029 FORMAT('0CENTRAL PILLAR AREA, HECTARES (B)',T59,F8.1)
1030 FORMAT('0',12A4)
1031 FORMAT(' ',T4,'WIDTH, METERS (A)',T59,F8.1)
1032 FORMAT(' ',T4,'LENGTH BY WASTE, METERS (A)')
1033 FORMAT(' ',T4,'HEIGHT, METERS (A)',T59,F8.1)
1034 FORMAT('0PANEL WIDTH, METERS (A)',T38,7(5X,F8.1))
1035 FORMAT(' PANEL LENGTH, METERS (A)',T38,7(5X,F8.1))
1036 FORMAT(' GEOMETRIC EXCLUSION FACTOR',T59,F8.1)
1037 FORMAT('0SITE PREPARATION COST, $/HECTARE (B)',T59,F8.1)
1038 FORMAT('0DISCOUNT RATE, PERCENT',T59,F8.1)
1039 FORMAT('0',T41,'WASTE TYPE')
1040 FORMAT(' ',T41,89A1)
1041 FORMAT(' ',T41,7(3A4,1X))
1042 FORMAT(' ',T41,7(11A1,2X))
1043 FORMAT('0EXTRACTION RATIO',T38,7(5X,F8.3))
1044 FORMAT('0ROOM WIDTH, METERS (A)',T38,7(5X,F8.1))
1045 FORMAT(' ROOM HEIGHT, METERS (A)',T38,7(5X,F8.1))
1046 FORMAT('0ROOMS PER PANEL',T38,7(5X,I8))
1047 FORMAT(' ',T4,'CORRIDORS/PANEL',T63,I3)
1052 FORMAT('0',35('-'))
1053 FORMAT(' (A) 1 METER = 3.281 FEET')
1054 FORMAT(' (B) 1 HECTARE = 2.471 ACRES')
1055 FORMAT(' CASE DESCRIPTION - ',20A4)
1056 FORMAT('0',T30,'TABLE 13 CONSTRUCTION AND PREPARATION COSTS FOR A
1REPOSITORY IN ',4A4)
1057 FORMAT(' ',T30,8('-'))
1058 FORMAT(' ',T50,'(MILLIONS OF ',I4,' DOLLARS)')
1059 FORMAT('0',T61,'FACILITY CONSTRUCTION')
1060 FORMAT(' ',T34,'SITE',T47,'SHAFT',T60,23('-'),T86,'ARCHITECT')
1061 FORMAT(' YEAR LAND',T21,'LICENSE',T34,'PREPARE',T47,'CONSTRUCT',
1 T60,'SURFACE',T73,'SUBSURFACE',T86,'ENGINEER',T99,'STARTUP',
2 T112,'TOTAL')

```

```

1062 FORMAT(' ----',2X,9(10('-'),3X))
1063 FORMAT(' ')
1064 FORMAT(' ',I4,2X,8(F8.2,5X),F8.2)
1065 FORMAT('OTABLE 11 WASTE FACILITY CAPITAL COSTS FOR A REPOSITORY IN
1 ' ',4A4)
1066 FORMAT(' -----')
1067 FORMAT('O',T51,'MILLIONS OF ',I4,' $')
1068 FORMAT(' ',T51,18('-'))
1069 FORMAT('OWASTE RECEIVING MODULE - REMOTE HANDLED' ,T51,F12.2)-
1070 FORMAT('OWASTE RECEIVING MODULE - CONTACT HANDLED' ,T51,F12.2)
1071 FORMAT('OWASTE PACKAGING MODULE',T51,F12.2)
1072 FORMAT('O',T41,'RECEIVING',T51,'PACKAGING')
1073 FORMAT(' ',T41,9('-'),T51,9('-'))
1076 FORMAT('O',12A4,T51,F12.2)
1077 FORMAT('OEMPLACEMENT EXCAVATION METHODS')
1078 FORMAT('O',T17,'TABLE 2 RECEIVING AND PROCESSING FOR A REPOSITORY
1 IN ',4A4)
1079 FORMAT(' ',T17,7('-'))
1080 FORMAT('OR.H. WASTE PROCESSING RATE',T41,F8.1,T51,F8.1)
1081 FORMAT(' PER MODULE, UNITS/SHIFT')
1082 FORMAT('OC.H. WASTE PROCESSING RATE',T41,F8.1)
1083 FORMAT(' PER MODULE, UNITS/SHIFT')
1084 FORMAT(' ',T41,7(3A4,1X))
1085 FORMAT(' ',T41,7(11A1,2X))
1086 FORMAT(' RAIL CONTAINER CAPACITY, UNITS',T41,7(F8.1,5X))
1087 FORMAT('OTRUCK CONTAINER CAPACITY, UNITS',T41,7(F8.1,5X))
1089 FORMAT(' TOTAL' ,T51,F12.2)
1090 FORMAT('O-----')
1091 FORMAT('O',T15,'TABLE 12 EQUIPMENT LIFE AND COST FOR A REPOSITORY
1IN ',4A4)
1092 FORMAT(' ',T15,8('-'))
1093 FORMAT('O',T83,'COSTS,')
1094 FORMAT(' ',T51,'LIFE',T81,'MILLIONS $')
1095 FORMAT(' ',T51,25('-'),T81,10('-'))
1096 FORMAT('OSURFACE WASTE OPERATIONS')
1097 FORMAT(' RECEIVING MODULE - REMOTE HANDLED WASTE',T49,F10.0,
1T61,'RECEIVING UNITS',T81,F10.2)
1098 FORMAT(' RECEIVING MODULE - CONTACT HANDLED WASTE',T49,F10.0,
1T61,'RECEIVING UNITS',T81,F10.2)
1099 FORMAT(' PACKAGING MODULE',T49,F10.0,T61,'RECEIVING UNITS',
1T81,F10.2)
1100 FORMAT('OUNDERGROUND WASTE OPERATIONS')
1101 FORMAT(' TRANSPORTER - REMOTE HANDLED WASTE',T49,F10.0,T61,
1'PACKAGES',T81,F10.2)
1102 FORMAT(' TRANSPORTER - CONTACT HANDLED WASTE',T49,F10.0,T61,
1'PACKAGES',T81,F10.2)
1103 FORMAT(' EMLACEMENT - REMOTE HANDLED WASTE',T49,F10.0,T61,
1'PACKAGES',T81,F10.2)
1104 FORMAT(' EMLACEMENT - CONTACT HANDLED WASTE',T49,F10.0,T61,

```

```

1 'PACKAGES',T81,F10.2)
1105 FORMAT('ROCK OPERATIONS')
1106 FORMAT('    EXCAVATION AND HAULAGE',T49,F10.0,T61,'TONNES ROCK',
1T81,F10.2)
1107 FORMAT('    ',T4,'HOLE DRILLING',T49,F10.0,T61,'PACKAGES',T81,F10.2)
1108 FORMAT('    HOISTING',T49,F10.0,T61,'TONNES ROCK',T81,F10.2)
1109 FORMAT('    ROCK STOCKPILE',T49,F10.0,T61,'TONNES ROCK',T81,F10.2)
1110 FORMAT('    OFFSITE DISPOSAL',T49,F10.0,T61,'TONNES ROCK',T81,
1F10.2)
1111 FORMAT('    BACKFILLING AND HAULAGE',T49,F10.0,T61,'TONNES ROCK',
1T81,F10.2)
1112 FORMAT('OSUPPORT OPERATIONS')
1113 FORMAT('    ',T4,12A4,T49,F10.0,T61,'YEARS',T81,F10.2)
1114 FORMAT('    ',T4,'ROOM EMPLACEMENT',T49,F10.0,T61,'PACKAGES',T81,F10.
12)
1115 FORMAT('    ',T4,'TRENCHING',T49,F10.0,T61,'PACKAGES',T81,F10.2)
1121 FORMAT('O',T11,5(10('-'),5X))
1122 FORMAT(' TOTAL',T81,F10.2)
1123 FORMAT('O',T47,'TABLE 10 SHAFT PARAMETERS AND COSTS')
1124 FORMAT('O',T63,'WASTE TRANSPORT',T85,'EXHAUST')
1125 FORMAT('    ',T41,'MEN AND',T52,'MINING',T63,2(20('-'),2X),'ROCK')
1126 FORMAT('    ',T41,'MATERIALS',T52,'VENT',T63,'REMOTE',T74,'CONTACT',
1T85,'MINING',T96,'WASTE',T107,'HOISTING',T118,'TOTAL')
1127 FORMAT('    ',T41,8(9('-'),2X))
1128 FORMAT('ODESIGN DIAMETER, METERS' ,T41,7(F7.1,4X))
1129 FORMAT('OINCREMENTAL EXCAVATED RADIUS, METERS' ,T41,7(F7.1,4X))
1130 FORMAT('OEXCAVATED DEPTH, METERS' ,T41,7(F7.1,4X))
1131 FORMAT('OLINED DEPTH, METERS' ,T41,7(F7.1,4X))
1132 FORMAT('OWASTE TRANSPORT CAPACITY')
1133 FORMAT('    TRIPS PER SHIFT',T63,2(17,4X),T118,17)
1134 FORMAT('    PACKAGES PER TRIP')
1135 FORMAT('OROCK HOISTING CAPACITY')
1136 FORMAT('    TONNES PER TRIP',T41,F7.0,T107,F7.0)
1138 FORMAT('ONUMBER OF SHAFTS' ,T41,8(4X,I2,5X))
1139 FORMAT('OSHAFT COSTS, MILLIONS OF ',I4,' DOLLARS')
1140 FORMAT('    SINKING' ,T41,8(F7.1,4X))
1141 FORMAT('    WATER CONTROL' ,T41,8(F7.1,4X))
1142 FORMAT('    LINING' ,T41,8(F7.1,4X))
1143 FORMAT('    EQUIPMENT' ,T41,8(F7.1,4X))
1144 FORMAT('    SHAFT STATIONS' ,T41,8(F7.1,4X))
1145 FORMAT('    TOTAL' ,T41,8(F7.1,4X))
1146 FORMAT('    -----')
1147 FORMAT('    ',T47, 8('-'))
1148 FORMAT('    ',T6,3A4,T66,I3)
1149 FORMAT('    TRIPS PER SHIFT',T41,I7,T107,I7)
1150 FORMAT('O',T8,'TABLE 14 MINING AND ROCK REMOVAL OPERATIONS COSTS S
1UMMARY FOR A REPOSITORY IN ',4A4)
1151 FORMAT('    ',T8,8('-'),2X,'COSTS IN MILLIONS OF ',I4,' DOLLARS')
1153 FORMAT('    ',T41,'SURFACE')

```

```

1154 FORMAT(' ',T41,'HANDLE AND',T56,'OFFSITE')
1155 FORMAT(' YEAR',T11,'MINING (1)',T26,'HOISTING (1)',T41,'STORE (2)'
1,T56,'DISPOSAL (2)',T71,'TOTAL')
1156 FORMAT(' ----',T11,5(10('-'),5X))
1157 FORMAT(' ',I4,T11,5(F8.2,7X))
1158 FORMAT(' TOTAL',T11,5(F8.2,7X))
1160 FORMAT('0', T6,'TABLE 3 WASTE SHIPPING DISTRIBUTION FOR A REPOSI
1TORY IN ',4A4)
1161 FORMAT(' ',T7,7('-'))
1162 FORMAT('0',T7,'PERCENTAGE OF UNITS SHIPPED BY RAIL FOR:')
1163 FORMAT(' ',T7,102A1)
1164 FORMAT(' YEAR',T7,10(3A4))
1165 FORMAT(' ----',T7,10(11A1,1X))
1166 FORMAT(' ',I4,T7,10(F 8.1,4X))
1167 FORMAT('0',T26,'TABLE 6 EMLACEMENT PACKAGE THERMAL OUTPUT FOR A
1 REPOSITORY IN ',4A4)
1168 FORMAT(' ',T26,7('-'))
1169 FORMAT('0',T8,5(3A4,10X))
1170 FORMAT(' ',T8,5(20A1,2X))
1171 FORMAT(' ',T7,5(10X,A1,'AVERAGE',4X))
1172 FORMAT(' ',T7,5(A1,'NUMBER OF HEAT RATE',1X))
1173 FORMAT(' YEAR',T7,5(A1,'PACKAGES KW/PACKAGE',1X))
1174 FORMAT(' ----',T8,5(9A1,1X,10A1,2X))
1175 FORMAT(' ',I4,T8,5(I9,1X,F10.2,2X))
1176 FORMAT('0',T35,'TABLE 7 WASTE LOADING PARAMETERS FOR A REPOSITOR
1Y IN ',4A4)
1177 FORMAT(' ',T35,7('-'))
1178 FORMAT(' ',T27,10('-'),1X,'IN ',4A4)
1180 FORMAT('0',T8,5(3A4,14X))
1181 FORMAT(' ',T8,5(24A1,2X))
1182 FORMAT(' ',T7,5(A1,'PITCH, PACKAGES PACKAGES '))
1183 FORMAT(' YEAR',T7,4(A1,'METERS PER ROW PER ROOM '))
1184 FORMAT(' ----',T8,4(6A1,1X,8A1,1X,8A1,2X))
1185 FORMAT(' ',I4,T8,4(F6.1,1X,I8,1X,I8,1X))
1186 FORMAT('0',T27,'TABLE 5',A1,'. WASTE OPERATIONS PARAMETERS FOR '
+ ,3A4)
1187 FORMAT(' ',T27,9('-'),2X,'IN ',4A4)
1188 FORMAT('0',T61,'NUMBER OF',T77,'NUMBER OF',T92,'NUMBER OF',
1 T105,'NUMBER OF')
1189 FORMAT(' ',T10,'NUMBER OF',T23,'UNITS',T35,'NUMBER OF',T48,
1 'NUMBER OF',T61,'UNDERGROUND',T77,'EMPLACEMENT',T92,'DRILLING',
2 T105,'ROOMS')
1190 FORMAT(' YEAR',T10,'SHIFTS',T23,'RECEIVED',T35,'PACKAGES',T48,
1 'PERSONNEL',T61,'TRANSPORTERS',T77,'MACHINES',T92,'MACHINES',T105,
2 'FILLED')
1191 FORMAT(' ----',T10,9('-'),T23,8('-'),T35,9('-'),T48,9('-'),
1T61,12('-'),T77,11('-'),T92,9('-'),T105,9('-'))
1192 FORMAT(' ',I4,T10,F9.2,4X,F8.1,4X,I9,4X,F9.1,4X,F12.1,4X,F11.1,
1 4X,F9.1,4X,F9.1)

```

```

1193 FORMAT('0',T27,'TABLE 19',A1,'. WASTE OPERATIONS COST SUMMARY FOR
+ ,3A4)
1194 FORMAT('0',T27,'TABLE 19',A1,'. WASTE OPERATIONS COST SUMMARY FOR
1A REPOSITORY IN ',4A4)
1195 FORMAT(' ',T27,8('-'))
1196 FORMAT(' ',T54,' MILLIONS OF ',I4,' DOLLARS ',/,T55,24('-'))
1197 FORMAT('0',T60,'EMPLACEMENT',T77,'EMPLACEMENT')
1199 FORMAT(' ----',T9,7(14('-'),3X))
1200 FORMAT(' ',I4,T9,7(F9.2,8X))
1208 FORMAT('0',T24,'TABLE 15 WASTE OPERATIONS LABOR COST SUMMARY FOR
1A REPOSITORY IN ',4A4)
1209 FORMAT(' ',T24,8('-'))
1210 FORMAT(' YEAR RECEIVING',T26,'PACKAGING',T43,'TRANSPORTATION',
1T60,'EXCAVATION',T77,'OPERATIONS',T94,'TOTAL')
1211 FORMAT(' ----',T9,6(14('-'),3X))
1212 FORMAT('0',T24,'TABLE 16 WASTE OPERATIONS MATERIALS COST SUMMARY
1FOR A REPOSITORY IN ',4A4)
1213 FORMAT('0',T24,'TABLE 17 WASTE OPERATIONS EQUIPMENT COST SUMMARY
1FOR A REPOSITORY IN ',4A4)
1214 FORMAT(' ',T11,'UNDERGROUND FACILITIES',T47,6(F10.2,3X))
1215 FORMAT('0',T10,'TABLE 4A RECEIVING AND PROCESSING FOR A REPOSITOR
1Y IN ',4A4)
1216 FORMAT(' ',T10,8('-'))
1217 FORMAT('0',T18,'NUMBER OF MODULES REQUIRED')
1218 FORMAT(' ',T8,47('-'),T57,70A1)
1219 FORMAT(' ',T8,2('REMOTE-HANDLED',2X),T40,'CONTACT-HANDLED')
1220 FORMAT(' YEAR',T8,'RECEIVING',T24,'PACKAGING',T40,'RECEIVING')
1221 FORMAT(' ----',T8,3(14('-'),2X),T40,15('-'),T57,5(11A1,2X),1X,
1 A4,A1)
1222 FORMAT(' ',I4,T8,3(I10,6X))
1223 FORMAT('0',T41,'WASTE TYPE')
1224 FORMAT(' ',T41,89A1)
1225 FORMAT('0',T27,'TABLE 5',A1,'. WASTE OPERATIONS PARAMETERS SUMMARY
1 FOR A REPOSITORY IN ',4A4)
1226 FORMAT(' YEAR RECEIVING',T26,'PACKAGING',T43,'TRANSPORTATION',
1T60,'EXCAVATION',T77,'OPERATIONS',T94,'OTHER',T111,'TOTAL')
1228 FORMAT(' YEAR RECEIVING',T26,'PACKAGING',T43,'TRANSPORTATION',
1T60,'EXCAVATION',T77,'OPERATIONS',T94,'OTHER ',T111,'TOTAL')
1230 FORMAT('0',T21,'TABLE 20 BACKFILLING AND DECOMMISSIONING SUMMARY F
1OR A REPOSITORY IN ',4A4)
1231 FORMAT(' ',T21,8('-'))
1232 FORMAT('0',T8,'BACKFILL MATERIALS, THOUSAND TONNES',T53,'COSTS IN
1 MILLIONS OF ',I4,' DOLLARS')
1233 FORMAT(' ',T8,42('-'),T53,72('-'))
1234 FORMAT(' YEAR ROCK',T23,'OTHER',T38,'TOTAL',T53,'BACKFILLING',
1T68,'SEALING',T83,'DECOMMISSION',T98,'SURVEILLANCE',T113,'TOTAL')
1235 FORMAT(' ----',T8,8(12('-'),3X))
1236 FORMAT(' ',I4,T8,8(F11.2,4X))
1237 FORMAT('0',T31,'TABLE 21 COST SUMMARY FOR A REPOSITORY IN ',4A4)

```

```

1238 FORMAT(' ',T31,8('-'),2X,'MILLIONS OF ',I4,' DOLLARS')
1239 FORMAT('0',T47,'WASTE TYPE')
1240 FORMAT(' FROM',T47,86A1)
1241 FORMAT(' TABLE COST CATEGORY',T47,6(3A4,1X))
1242 FORMAT(' -----',T9,35('-'), T47, 6(10A1,3X),A4,A1)
1243 FORMAT('0 13',T9,'CONSTRUCTION AND PREOPERATION')
1244 FORMAT(' ',T11,'LAND',T47,6(F10.2,3X))
1245 FORMAT(' ',T11,'LICENSE',T47,6(F10.2,3X))
1246 FORMAT(' ',T11,'SITE PREPARATION',T47,6(F10.2,3X))
1247 FORMAT(' ',T11,'SHAFT',T47,6(F10.2,3X))
1249 FORMAT(' ',T11,'SURFACE FACILITIES',T47,6(F10.2,3X))
1250 FORMAT(' ',T11,'ARCHITECT ENGINEER',T47,6(F10.2,3X))
1251 FORMAT(' ',T11,'STARTUP',T47,6(F10.2,3X))
1252 FORMAT(' ',T11,'MINING',T47,6(F10.2,3X))
1253 FORMAT(' ',T11,'HOISTING',T47,6(F10.2,3X))
1254 FORMAT(' ',T11,'SURFACE HANDLE AND STORE',T47,6(F10.2,3X))
1255 FORMAT(' ',T11,'OFFSITE DISPOSAL',T47,6(F10.2,3X))
1256 FORMAT(' ',T11,'RECEIVING',T47,6(F10.2,3X))
1257 FORMAT(' ',T11,'PACKAGING',T47,6(F10.2,3X))
1258 FORMAT(' ',T11,'TRANSPORTATION',T47,6(F10.2,3X))
1259 FORMAT(' ',T11,'EMPLACEMENT OPERATIONS',T47,6(F10.2,3X))
1260 FORMAT(' ',T11,'OTHER',T47,6(F10.2,3X))
1261 FORMAT(' ',T11,'BACKFILLING',T47,6(F10.2,3X))
1262 FORMAT(' ',T11,'SEALING',T47,6(F10.2,3X))
1263 FORMAT(' ',T11,'DECOMMISSIONING',T47,6(F10.2,3X))
1264 FORMAT('0',T29,'TABLE 23 CASH FLOW SUMMARY FOR A REPOSITORY IN',
14A4)
1265 FORMAT(' ',T29, 8('-'),T40,'MILLIONS OF ',I4,' DOLLARS')
1267 FORMAT('0',T29,'OPERATING COSTS')
1268 FORMAT(' ',T29,52('-'))
1269 FORMAT(' ',T9,'CONSTRUCTION',T29,'OPERATIONS',
1T47,'WASTE',T65,'BACKFILLING AND')
1270 FORMAT(' YEAR',T9,'STARTUP COSTS',T29,'MINING',T47,
1'OPERATIONS',T65,'DECOMMISSIONING CONTINGENCIES TOTAL')
1271 FORMAT(' ----',T9,15('-'),T29,5(15('-'),3X))
1272 FORMAT(' ',I4,T9,F10.2,T29,5(F12.2,6X))
1273 FORMAT('0',T31,'TABLE 24 UNIT COST SUMMARY FOR A REPOSITORY IN '
1,4A4)
1274 FORMAT(' ',T31,8('-'),T42,I4,' DOLLARS PER KILOGRAM HEAVY METAL')
1275 FORMAT('0',T31,'TABLE 25 UNIT COST SUMMARY FOR A REPOSITORY IN '
1,4A4)
1276 FORMAT(' ',T31,8('-'),T42,I4,' DOLLARS PER CUBIC METER OF WASTE')
1277 FORMAT('0',T31,'TABLE 26 UNIT COST SUMMARY FOR A REPOSITORY IN '
1,4A4)
1278 FORMAT(' ',T31,8('-'),T42,I4,' DOLLARS PER PACKAGE ')
1279 FORMAT('0',T29,'TABLE 18 UTILITIES REQUIREMENTS AND COST SUMMARY
1 FOR A REPOSITORY IN ',4A4)
1280 FORMAT('0',T9,'REQUIREMENTS')
1281 FORMAT(' ',T9,49('-'))

```



```

1282 FORMAT(' ',T9,'ELECTRICITY',T22,'STEAM',T35,'WATER',T48,'DIESEL',
1 T61,'COSTS, MILLIONS OF ',I4,' DOLLARS')
1283 FORMAT(' ',T9,'THOUSANDS',T22,'THOUSANDS',T35,'MILLIONS',T48,'THOU
1SANDS',T61, 62('-'))
1284 FORMAT(' ',T9,'OF KWH',T22,'OF MJ(A)',T35,'OF LITERS',T48,'OF LITE
1RS',T61,'ELECTRICITY',T74,'STEAM',T87,'WATER',T100,'DIESEL',T113,'
2TOTAL')
1285 FORMAT(' ',I4,T9,9(F9.2,4X))
1286 FORMAT('O (A) MJ(MEGA JOULE)= .977 POUNDS OF STEAM AT 100C.')
1287 FORMAT(' ',T11,'SURVEILLANCE',T47,6(F10.2,3X))
1288 FORMAT('O',T9,'SUBTOTAL',T47,6(F10.2,3X))
1289 FORMAT('O',T9,'CONTINGENCY',T47,6(F10.2,3X))
1290 FORMAT('O',T3,'TOTAL',T47,6(F10.2,3X))
1291 FORMAT(' ',T11,'EMPLACEMENT EXCAVATION',T47,6(F10.2,3X))
1292 FORMAT('O 14',T9,'MINING AND ROCK REMOVAL OPERATIONS')
1293 FORMAT('O 19',T9,'WASTE OPERATIONS')
1294 FORMAT('O 20',T9,'BACKFILLING AND DECOMMISSIONING')
1296 FORMAT(' ',T29,8('-'))
1297 FORMAT(' ----',T9,9(10('-'),3X))
1298 FORMAT(' ',I4,T29,F12.2,T65,3(F12.2,6X))
1299 FORMAT('OSUPPORT FACILITY EQUIPMENT COST',T51,F12.2)
1300 FORMAT(' ',I4,T60,F8.2,T112,F8.2)
1301 FORMAT('OUNDERGROUND TRANSPORTATION FACILITY',T51,F12.2)
1302 FORMAT('OEMPLACEMENT EXCAVATION FACILITY',T51,F12.2)
1303 FORMAT(' ',I4,T26,2(F10.0,5X))
1304 FORMAT(' ',I4,17X,3(F10.0,2X))
1305 FORMAT('OEMPLACEMENT SUPPORT FACILITY',T51,F12.2)
C
1306 FORMAT('O', ' TOTAL',T54,5(F10.1,3X))
1307 FORMAT('O',T23,9('-'),4X,8('-'),T105,9('-'))//
+ ' TOTAL',T22,F9.1,4X,I9,T105,F9.1)
1308 FORMAT('O', ' TOTAL',T8,5(I9,13X))
1309 FORMAT('O',T11,3(10('-'),5X),2(10('-'),5X))//
+ ' TOTAL',T11,3(F10.0,5X),2(F10.2,5X))
1310 FORMAT('O',T8,9(10('-'),3X))// ' TOTAL',1X,8(F8.2,5X),F8.2)
1311 FORMAT(' -----',T11,5(10('-'),5X))
1312 FORMAT('O',T9,6(14('-'),3X))
1313 FORMAT('O', 'TOTAL',T9,7(F9.2,8X))
1314 FORMAT('O',T9,7(14('-'),3X))
1315 FORMAT('O', 'TOTAL',T9,7(F9.2,8X))
1317 FORMAT('O',T9,9(10('-'),3X))
1318 FORMAT(' TOTAL',T8,9(F10.2,3X))
1319 FORMAT(' ',T9,7(14('-'),3X))
1320 FORMAT(' ', 'TOTAL',T9,7(F9.2,8X))
1322 FORMAT('O',T8,8(12('-'),3X))
1324 FORMAT(' ', 'TOTAL',T8,8(F11.2,4X))
1330 FORMAT('O-----')
1335 FORMAT(' ',T27,10('-'))
1336 FORMAT(' ',T27,9('-'))

```

```

1339 FORMAT('0',T9,15('-'),T29,5(15('-'),3X))
1340 FORMAT('0TOTAL',T9,F10.2,T29,5(F12.2,6X))
1350 FORMAT(' ',I4,T10,F9.2,4X,F8.1,4X,I9,4X,F9.1,48X,F9.1)
      RETURN
1360 FORMAT(' ',T8,5(9A1,13X))
1370 FORMAT(' ',T57,5(10A1,2X))
1380 FORMAT(' ',T6,3A4,T76,I3)
1390 FORMAT(' ',I4,5X,2(F10.0,2X),F10.0,'* ',2(F10.0,2X),5(F10.1,2X))
1400 FORMAT('0          * ROCK HOISTING CAPACITY WAS EXCEEDED')
1420 FORMAT(' ',T8,124A1)
1410 FORMAT('0',T18,'NUMBER OF CONTAINER SHIPMENTS FOR:')
1430 FORMAT(' YEAR',T8,5(3A4,1X),2X,A4,A1)
1440 FORMAT(' ----',T8,6(11A1,2X))
1445 FORMAT(' ',I4,T8,5(F8.1,5X),F8.1)
1447 FORMAT(' ',T8,5(F8.1,5X),F8.1)
1446 FORMAT(' ',T8,15('-'),T17,5(11A1,2X),1X,A4,A1)
1460 FORMAT('0',T10,'TABLE 4B RECEIVING AND PROCESSING FOR A REPOSITOR
      1Y IN ',4A4)
1560 FORMAT('0    ALLOWABLE NUMBER OF REMOTE-HANDLED RECEIVING'
      + ' MODULES IS CONSTRAINING REMOTE WASTE RECEIPT RATES BELOW'
      + ' INPUT VALUES')
1565 FORMAT('0    CAPACITY OF REMOTE-HANDLED WASTE SHAFT '
      + ' IS CONSTRAINING REMOTE HANDLED WASTE RECEIPT RATES '
      + ' BELOW INPUT VALUES')
1570 FORMAT('0    ALLOWABLE NUMBER OF CONTRACT-HANDLED RECEIVING'
      + ' MODULES IS CONSTRAINING CONTACT WASTE RECEIPT RATES '
      + ' BELOW INPUT VALUES')
1575 FORMAT('0    CAPACITY OF CONTACT HANDLED WASTE SHAFT IS '
      + ' CONSTRAINING CONTACT HANDLED WASTE RECEIPT RATES '
      + ' BELOW INPUT VALUES')
1600 FORMAT('0    (1) QUANTITIES INCLUDE ALL ROCK EXCAVATED'
      * ' (PANELS,CORRIDORS, SHAFTS, BOREHOLES, PILLARS, ETC.))')
1610 FORMAT('0    (1) QUANTITIES INCLUDE ALL ROCK EXCAVATED '
      * ' EXCEPT SHAFT '
      * ' (PANELS, CORRIDORS, BOREHOLES, PILLARS,ETC.)'//
      * ' (2) INCLUDES ALL ROCK EXCAVATED (PANELS, CORRIDORS, '
      * ' SHAFTS, BOREHOLES, PILLARS, ETC.))')
1620 FORMAT('0',T7,'THOUSAND OF TONNES EXCAVATED FOR SHAFT ARE ',F10.0/
      * /T7,'THOUSAND OF TONNES EXCAVATED FOR HOLE DRILLING ARE ',F10.0)
1640 FORMAT('0 MINIMUM ROOM HEIGHT HAS BEEN INCREASED FROM INPUT'
      * ' VALUES IN ORDER TO STACK PACKAGES ',F10.2,' HIGH FOR '
      * ' 3A4,' WASTE')
1650 FORMAT(' ',T7,3A4,T59,F8.1)
1670 FORMAT(' ',T4,'LENGTH, METERS (A)',T59,F8.1)
      DEBUG SUBCHK
      AT 1
      END
      SUBROUTINE OUT22
      INCLUDE PARAM

```

```

INCLUDE OUTCOM
INCLUDE INCOM
INCLUDE HUBCOM
INCLUDE RECCOM
INCLUDE SOPCOM
INCLUDE SSTCOM
INCLUDE MNDCOM
INCLUDE WAFCOM
INCLUDE SFTCOM
INCLUDE PRECOM
INCLUDE PSTCOM
INCLUDE UOPCOM
INCLUDE UTLCOM
DIMENSION WPC(NW), WPO(NW), TWPC(NW), WHSUB(11), PFDECC(NW)
DIMENSION WPSUB(10), VSUB(7), TOTAL(11), RESUB(10)
REAL MINEC,MINEQC

C
18  CONTINUE
C
    LED = LEAD
    LE = LY - LEAD
    LEE = NBKF + IDD
    LB = LL + 1
    DO 100 K=1,NWT
    DO 110 I=1,LL
    PRECST = PRECST + (AESERV(K,I) + STARTC(K,I)) / ONEMIL
110  CONTINUE
100  CONTINUE
    SITSUB = TLDCST(NWP1) + TLCCST(NWP1) + TSPCST(NWP1) + PRECST
    DO 130 K=1,NWT
    PKEQPC(K,LL+1) = PKEQPC(K,LL+1) / ONEMIL
    WPC(K) = TTRHPM(K) + PKEQPC(K,LL+1)
130  CONTINUE
    DO 140 K=1,NWT
    WPO(K) = CPLBR(K) + CPMAT(K) + CPEQ(K) + PFPOWC(K)
140  CONTINUE
    DO 150 K=1,NWT
    DO 150 I=1,LY
    PFDECC(K) = PFDECC(K) + PFCST(K,I) / ONEMIL
150  CONTINUE
    DO 155 K=1,NWT
    PFDECC(K) = PFDECC(K) * DCOMCF
    TWPC(K) = WPC(K) + WPO(K) + PFDECC(K)
155  CONTINUE
    DO 160 K=1,NWT
    WPSUB(1) = WPSUB(1) + TTRHPM(K)
    WPSUB(2) = WPSUB(2) + PKEQPC(K,LL+1)
    WPSUB(3) = WPSUB(3) + WPC(K)
    WPSUB(4) = WPSUB(4) + CPLBR(K)

```

```

WPSUB(5) = WPSUB(5) + CPMAT(K)
WPSUB(6) = WPSUB(6) + CPEQ(K)
WPSUB(7) = WPSUB(7) + PFPOWC(K)
WPSUB(8) = WPSUB(8) + WPO(K)
WPSUB(9) = WPSUB(9) + PFDECC(K)
WPSUB(10) = WPSUB(10) + TWPC(K)
160  CONTINUE
      TREFC = (TCRHRM + TCCHRM)
      RCC = TREFC + REEQCT(LL+1)
      CEQPT(1) = CEQPT(1) - REEQCT(LL+1)
      RECUC = RFPOWC + RSTMC + RDSLC
      RCO = CLABOR(1) + CMATL(1) + CEQPT(1) + RECUC
      DO 170 K=1,NWT
      DO 170 I=1,LY
170    RFDECC = RFDECC + RFCST(K,I) / ONEMIL
      CONTINUE
      RFDECC = RFDECC * DCOMCF
      TTRCC = RCC + RCO + RFDECC
      WSC = SFWSC + SFWSE
      WSFTSC = (SFTSLC(3) + SFTSLC(4))
      TWSC = WSC + WSFTSC
      WTC = TTCAPC + TREQCT(LL+1)
      CEQPT(2) = CEQPT(2) - TREQCT(LL+1)
      WTO = CLABOR(2) + CMATL(2) + CEQPT(2) + UDSLC
      TWTC = WTC + WTO
      DO 180 K=1,NWT
      DO 190 I=1,LL
      MINEC = MINEC + ((RMNLBR(K,I) * OVRHDF * ROLBRC(1)) +
190    + (RMMATR(K,I) * ROMATC(1)))
      CONTINUE
      DO 194 I=LB,LY
      ORMMAS = ORMMAS + RMFILD(K,I+LED) * RKVOL(K) * DENSTY
      OSCMAS = OSCMAS + CORVOL(K,I) * DENSTY
194  CONTINUE
180  CONTINUE
      DO 182 K=1,NWT
      SRKVOL = SRKVOL + RKVOL(K)
      SRMUVL = SRMUVL + RMUVOL(K)
      DO 182 I=LB,NBKF
      WRBKFL = WRBKFL + (RKBKF(K,I)*ONETHO + OTBKF(K,I))
182  CONTINUE
      ROOMFC = SRKVOL / SRMUVL
      WRBKFL = WRBKFL * ROOMFC
      DO 183 K=1,NWT
      DO 183 I=LB,LY
      ACBKFL = ACBKFL + (RKBKF(K,I) * ONETHO + OTBKF(K,I) )
183  CONTINUE
      ACBKFL = ACBKFL * (1.-ROOMFC)
      MINEC = (MINEC + ROCST(1)) / ONEMIL

```

```

TORMAS = ORMMAS + OSCMAS
TPRMAS = PRMMAS + SPMASS + PSCMAS + CMMAS
RECAPC = MINEC * PRMMAS / TPRMAS
DO 185 I=1,LL
MINEQC = MINEQC + (NEQRPL(1,I) * ROECST(1)) / ONEMIL
185 CONTINUE
DO 205 I=1,LL
RKM1 = RKM1 + SRKMND(I)
205 CONTINUE
RKM2 = ORMMAS + OSCMAS + HDMAS
WEEPC = ROPOWC * HDMAS / RKM2
SEEC = ROPOWC * OSCMAS / RKM2
REPC = ROPOWC * ORMMAS / RKM2
RMMCE = MINEQC * PRMMAS / TPRMAS
ERC = RECAPC + RMMCE
SCELBR = CLABOR(3)
CLABOR(3) = CLABOR(3) * ORMMAS / TORMAS / ONEMIL
CMATL(3) = CMATL(3) * ORMMAS / TORMAS / ONEMIL
CEQPT(3) = CEQPT(3) * ORMMAS / TORMAS / ONEMIL
ERO = CLABOR(3) + CMATL(3) + CEQPT(3) + REPC
REBFC = WRBKFL * BKFLC / ONEMIL
TTREC = ERC + ERO + REBFC
EEC = TECAPC + EXEQCT(LL+1)
CEQPT(4) = CEQPT(4) - EXEQCT(LL+1)
CEQPT(5) = CEQPT(5) - EMEQCT(LL+1)
EEO = CLABOR(4) + CMATL(4) + CEQPT(4) + WEEPC
TEEC = EEC + EEO
WEC = TUCAPC + EMEQCT(LL+1)
WEO = CLABOR(5) + CMATL(5) + CEQPT(5)
TEWC = WEC + WEO
WHSUB(1) = TREFC + SFWSC + TTCAPC + RECAPC + TECAPC + TUCAPC
WHSUB(2) = REEQCT(LL+1) + SFWSE + TREQCT(LL+1) + RMMCE +
+ (EXEQCT(LL+1) + EMEQCT(LL+1))
WHSUB(3) = WSC + WTC + ERC + EEC + WEC + RCC
DO 200 J= 1,5
WHSUB(4) = WHSUB(4) + CLABOR(J)
200 CONTINUE
DO 210 J= 1,5
WHSUB(5) = WHSUB(5) + CMATL(J)
210 CONTINUE
DO 220 J= 1,5
WHSUB(6) = WHSUB(6) + CEQPT(J)
220 CONTINUE
WHSUB(7) = RECUC + UDSLCL + REPC + WEEPC
WHSUB(8) = RCO + WTO + ERO + EEO + WEO
WHSUB(9) = WSFTSC + REBFC
WHSUB(10) = RFDECC
WHSUB(11) = TTRCC + TWSC + TWTC + TTREC + TEEC + TEWC
SMMSC = SFMMC + SFTHC(1)

```

```

TMMSC = SMMSC + SFTSLC(1)
DO 228 K=1,NWT
DO 228 I=1,LL
HASFEC = HASFEC + (NEQRPL(2,I) * ROECST(2) * WAFMV(K)) / ONEMIL
228 CONTINUE
HASFEC = HASFEC + SFTHE(7)
RHSC = SFRHC + HASFEC
DO 230 K=1,NWT
DO 230 I=LB,LEE
RKHLC = RKHLC + (RKHLBR(K,I) * OVRHDF * ROLBRC(2)) / ONEMIL
RKHMC = RKHMC + (RKHMAT(K,I) * ROMATC(2)) / ONEMIL
230 CONTINUE
DO 240 K=1,NWT
DO 240 I=LB,LEE
RKHERC = RKHERC + (NEQRPL(2,I) * ROECST(2) * WAFMV(K)) / ONEMIL
240 CONTINUE
RHOC = RKHLC + RKHMC + RKHERC
TRHSC = RHSC + RHOC + SFTSLC(7)
SPECAC = MINEC * SPMAS / TPRMAS
SPMEQC = MINEQC * SPMAS / TPRMAS
SPEC = SPECAC + SPMEQC
DO 245 K=1,NWT
DO 245 I=NBKF+1,LEE
SPBFC = SPBFC + (RKBKF(K,I) * ONETHO + OTBKF(K,I))
245 CONTINUE
SPBFC = SPBFC * BKFLC / ONEMIL
TSPEC = SPEC + SPBFC
SCMEQC = MINEQC * (PSCMAS + CMMAS) / TPRMAS
PSCMAS = MINEC * (PSCMAS + CMMAS) / TPRMAS
SCEXC = PSCMAS + SCMEQC
SCELBR = SCELBR * OSCMAS / TORMAS / ONEMIL
DO 280 K=1,NWT
DO 290 I=LB,LY
SCEMTC = SCEMTC + (RMMATR(K,I) * ROMATC(1))
290 CONTINUE
280 CONTINUE
SCEMTC = SCEMTC * OSCMAS / TORMAS / ONEMIL
DO 300 I=LB,LY
SCEERC = SCEERC + NEQRPL(1,I) * ROECST(1)
300 CONTINUE
SCEERC = SCEERC * OSCMAS / TORMAS / ONEMIL
CEOC = SCELBR + SCEMTC + SCEERC + SCEC
ACBFC = ACBKFL * BKFLC / ONEMIL
TSCEC = SCEXC + CEOC + ACBFC
DO 310 K=1,NWT
DO 320 I=1,LL
RKHSFC = RKHSFC + (SRHLBR(K,I) * OVRHDF * ROLBRC(3) + SRHMAT(K,I)
+ * ROMATC(3))
RKHSEC = RKHSEC + (NEQRPL(3,I) * ROECST(3) * WAFMV(K)) / ONEMIL

```

```

320  CONTINUE
310  CONTINUE
    RKHSFC = (RKHSFC + ROCST(3)) / ONEMIL
    SRHC = RKHSFC + RKHSEC
    DO 330 K=1,NWT
    DO 340 I=LB,LEE
    SRHLC = SRHLC + (SRHLBR(K,I) * OVRHDF * ROLBRC(3)) / ONEMIL
    SRHMC = SRHMC + (SRHMAT(K,I) * ROMATC(3)) / ONEMIL
340  CONTINUE
330  CONTINUE
    DO 350 I=LB,LEE
    SRHERC = SRHERC + (NEQRPL(3,I) * ROECST(3)) / ONEMIL
350  CONTINUE
    SRHOC = SRHLC + SRHMC + SRHERC
    SRHLC = SRHLC + (SRHLC / SRHOC * TRCC(4))
    SRHMC = SRHMC + (SRHMC / SRHOC * TRCC(4))
    SRHERC = SRHERC + (SRHERC / SRHOC * TRCC(4))
    SRHOC = SRHLC + SRHMC + SRHERC
    TSRHC = SRHC + SRHOC
    RESUB(1) = SFMMC + SFRHC + SPECAC + PSCMAS + RKHSFC
    RESUB(2) = SFTHE(1) + HASFEC + SPMEQC + SCMEQC + RKHSEC
    RESUB(3) = SMMSC + RHSC + SPEC + SCEXC + SRHC
    RESUB(4) = RKHLC + SCELBR + SRHLC
    RESUB(5) = RKHMC + SCEMTC + SRHMC
    RESUB(6) = RKHERC + SCEERC + SRHERC
    RESUB(7) = SCEC
    RESUB(8) = RHOC + CEOC + SRHOC
    RESUB(9) = SFTSLC(1) + SFTSLC(7) + SPBFC + ACBFC
    RESUB(10) = TMMSC + TRHSC + TSPEC + TSCEC + TSRHC
    VSSC = SFVSC + SFTHE(2)
    VENTUC = VENTPC + ACPOWC
    SVSS = VENTUC
    TVSS = VSSC + SVSS + SFTSLC(2)
    VEXSC = SFEXC + SFTHE(6)
    TVEX = VEXSC + SFTSLC(6)
    VMXSC = SFMXC + SFTHE(5)
    TVMX = VMXSC + SFTSLC(5)
    VSUB(1) = SFVSC + SFEXC + SFMXC
    VSUB(2) = SFTHE(2) + SFTHE(6) + SFTHE(5)
    VSUB(3) = VSSC + VEXSC + VMXSC
    VSUB(4) = VENTUC
    VSUB(5) = SVSS
    VSUB(6) = SFTSLC(2) + SFTSLC(6) + SFTSLC(5)
    VSUB(7) = VSUB(3) + VSUB(5) + VSUB(6)
    DO 360 L=1,NSC
    SUPFCC = SUPFCC + SFCST(L)
360  CONTINUE
    DO 370 I=1,LL
    SUPFEC = SUPFEC + SFEQCT(I)

```

```

370  CONTINUE
      SUPCC = SUPFCC + SUPFEC
      DO 380 I=LB,LY
        SUPLBC = SUPLBC + OWLBR(I)
        SUPEQC = SUPEQC + SFEQCT(I)
        SUPMTC = SUPMTC + OTMTCT(I)
380  CONTINUE
      DO 395 I=LB,LY
        DO 395 K=1,NWT
          WATC = WATC + WATERC(K,I) / ONEMIL
395  CONTINUE
      SUPUTC = SFPOWC + OSTMC + SDSLC + WATC
      SUPSUB = SUPLBC + SUPMTC + SUPEQC + SUPUTC
      CNTEQC = 0.0
      DO 400 K=1,NWT
        DO 410 I=1,LY
          SFDECC = SFDECC + (SFCAPC(K,I)) / ONEMIL
410  CONTINUE
400  CONTINUE
      SFDECC = SFDECC * DCOMCF
      TSUPP = SUPCC + SUPSUB + SFDECC
      CNTFCC = SITSUB * CONFAC(1) + WPSUB(1) * CONFAC(1)
      + + TREFC * CONFAC(1) + SFWSC * CONFAC(1) + TTCAPC * CONFAC(1)
      + + RECAPC * CONFAC(2) + (TECAPC + TUCAPC) * CONFAC(3)
      + + VSUB(1) * CONFAC(1) + SUPFCC * CONFAC(1) + SFMMC * CONFAC(1)
      + + (SHDFC(7)-SFTHE(7)) * CONFAC(1) + (SFRHC - SHDFC(7) + SFTHE(7))
      + * CONFAC(2) + (SPECAC + PSCMAS + RKHSFC) * CONFAC(2)
      CNTEQC = WPSUB(2) * CONFAC(3) + SFWSE * CONFAC(1)
      + + RMMCE * CONFAC(2) + (REEQCT(LL+1)+ TREQCT(LL+1)+ EXEQCT(LL+1)+
      + EMEQCT(LL+1)) * CONFAC(3) + VSUB(2) * CONFAC(1)
      + + SUPFEC * CONFAC(3)
      + + SFTHE(1) * CONFAC(1) + (HASFEC - SFTHE(7)) * CONFAC(2)
      + + SFTHE(7) * CONFAC(1) + (SPMEQC + SCMEQC + RKHSEC) * CONFAC(2)
      CNTCSB = CNTFCC + CNTEQC
      CNTLBR = WPSUB(4) * CONFAC(3) + (WHSUB(4) - CLABOR(3)) * CONFAC(3)
      + + RKHLC * CONFAC(2) + (SCELBR + SRHLC) * CONFAC(2)
      + + SUPLBC * CONFAC(3) + CLABOR(3) * CONFAC(2)
      CNTMAT = WPSUB(5) * CONFAC(3) + (WHSUB(5) - CMATL(3)) * CONFAC(3)
      + + CMATL(3) * CONFAC(2) + RKHMC * CONFAC(2) + (SCEMTC + SRHMC)
      + * CONFAC(2) + SUPMTC * CONFAC(3)
      CNTEQR = WPSUB(6) * CONFAC(3) + (WHSUB(6) - CEQPT(3)) * CONFAC(3)
      + + CEQPT(3) * CONFAC(2) + RKHERC * CONFAC(2)
      + + SCEERC * CONFAC(2) + SRHERC * CONFAC(2) + SUPEQC * CONFAC(3)
      CNTUTL = (WPSUB(7) + RECUC + UDSLC + REPC + WEEPC + SCEC +
      + VENTUC + SUPUTC) * CONFAC(3)
      CNTOSB = CNTLBR + CNTMAT + CNTEQR + CNTUTL
      CNTBKF = (WSFTSC + REBFC + SFTSLC(1) + SFTSLC(7) + SPBFC
      + + ACBFC + SFTSLC(2) + SFTSLC(6) + SFTSLC(5)) * CONFAC(4)
      CNTDD = (WPSUB(9) + RFDECC + SFDECC) * CONFAC(4)

```


TCONT = CNTCSB + CNTOSB + CNTBKF + CNTDD

C
C

TOTAL(1) = SITSUB + WPSUB(1) + WHSUB(1) + RESUB(1) + VSUB(1)
 + + SUPFCC + CNTFCC
 TOTAL(2) = WPSUB(2) + WHSUB(2) + RESUB(2) + VSUB(2)
 + + SUPFEC + CNTEQC
 TOTAL(3) = SITSUB + WPSUB(3) + WHSUB(3) + RESUB(3) + VSUB(3)
 + + SUPCC + CNTCSB
 TOTAL(4) = WPSUB(4) + WHSUB(4) + RESUB(4) + SUPLBC + CNTLBR
 TOTAL(5) = WPSUB(5) + WHSUB(5) + RESUB(5) + SUPMTC + CNTMAT
 TOTAL(6) = WPSUB(6) + WHSUB(6) + RESUB(6) + SUPEQC + CNTEQR
 TOTAL(7) = WPSUB(7) + WHSUB(7) + RESUB(7) + VSUB(4) + SUPUTC +
 + CNTUTL
 TOTAL(8) = WPSUB(8) + WHSUB(8) + RESUB(8) + SUPSUB + CNTOSB
 + + VSUB(5)
 TOTAL(9) = WHSUB(9) + RESUB(9) + VSUB(6) + CNTBKF
 TOTAL(10) = WPSUB(9) + WHSUB(10) + SFDECC + CNTDD
 TOTAL(11) = WPSUB(10) + WHSUB(11) + RESUB(10) + VSUB(7)
 + + TSUPP + TCONT + SITSUB

WRITE(IO,1400)
 1400 FORMAT(' ',T51,'TABLE 22. REPOSITORY COST MATRIX'
 1 /T51,9('-'))
 WRITE(IO,1405) IYDOL
 1405 FORMAT('0',T69,'COST IN MILLIONS OF ',I5,' DOLLARS'/
 1 ,T34,99('-'))
 WRITE(IO,1410)
 1410 FORMAT(' ',T37,'CAPITAL CONSTRUCTION',T78,'OPERATIONS'
 1 ,T107,'DECOMMISSIONING'
 2 /T34,26('-'),1X,44('-'),1X,18('-'))
 WRITE(IO,1415)
 1415 FORMAT(' ',T35,'FACILITY CAPITAL',T80,'EQUIP',
 1 T106,'BACKFILL')
 WRITE(IO,1420)
 1420 FORMAT(' ',T12,'SYSTEM',T35,'CONSTR. EQUIP. SUBTOTAL '
 1 , ' LABOR MATERIAL REPLACE. UTILITY SUBTOTAL & SEAL',
 2 ' DECOMM. TOTAL')
 WRITE(IO,1422)
 1422 FORMAT(' '1X,30('-'),1X,10(1X,8('-')),2X,8('-'))
 WRITE(IO,1425)
 1425 FORMAT(' SITE')
 WRITE(IO,1430) TLDCST(NWP1),TLDCST(NWP1),TLDCST(NWP1)
 1430 FORMAT(' LAND',T33,F9.2,T51,F9.2,T123,F10.2)
 WRITE(IO,1435) TLCCST(NWP1),TLCCST(NWP1),TLCCST(NWP1)
 1435 FORMAT(' LICENSING',T33,F9.2,T51,F9.2,T123,F10.2)
 WRITE(IO,1440) TSPCST(NWP1),TSPCST(NWP1),TSPCST(NWP1)
 1440 FORMAT(' SITE PREPARATION',T33,F9.2,T51,F9.2,T123,F10.2)
 WRITE(IO,1442) PRECST,PRECST,PRECST
 1442 FORMAT(' OTHER PRESTARTUP COSTS',T33,F9.2,T51,F9.2,T123,F10.2)

```

WRITE(IO,1445) SITSUB,SITSUB,SITSUB
1445 FORMAT(' SUBTOTAL ',T33,F9.2,T51,F9.2,T123,F10.2)
WRITE(IO,1450)
1450 FORMAT('O WASTE PACKAGE PREPARATION')
DO 1455 K=1,NWT
WRITE(IO,1460) (WSTYP(L,K),L=1,3),TTRHPM(K),PKEQPC(K,LL+1),
+ WPC(K),CPLBR(K),CPMAT(K),CPEQ(K),PFPOWC(K),WPO(K),PFDECC(K),
+ TWPC(K)
1460 FORMAT(' ',3A4,T33,8F9.2,T114,F9.2,F10.2)
1455 CONTINUE
WRITE(IO,1465) (WPSUB(J),J=1,10)
1465 FORMAT(' SUBTOTAL ',T33,8F9.2,T114,F9.2,F10.2)
WRITE(IO,1470)
1470 FORMAT('O WASTE HANDLING AND EMPLACEMENT')
WRITE(IO,1475) TREFC,REEQCT(LL+1),RCC,CLABOR(1),CMATL(1),
+ CEQPT(1),RECUC,RCO,RFDECC,TTRCC
1475 FORMAT(' RECEIVING ',T33,8F9.2,T114,F9.2,F10.2)
WRITE(IO,1480) SFWSC,SFWSE,WSC,WSFTSC,TWSC
1480 FORMAT(' WASTE SHAFT ',T33,3F9.2,T105,F9.2,T123,F10.2)
WRITE(IO,1485) TTCAPC,TREQCT(LL+1),WTC,CLABOR(2),CMATL(2),
+ CEQPT(2),UDSLC,WTO,TWTC
1485 FORMAT(' WASTE TRANSFER ',T33,8F9.2,T123,F10.2)
WRITE(IO,1490) RECAPC,RMMCE,ERC,CLABOR(3),CMATL(3),CEQPT(3),
+ REPC,ERO,REBFC,TTREC
1490 FORMAT(' ROOM EXCAVATION ',T33,9F9.2,T123,F10.2)
WRITE(IO,1495) TECAPC,EXEQCT(LL+1),EEC,CLABOR(4),CMATL(4),
+ CEQPT(4),WEEPC,EEO,TEEC
1495 FORMAT(' WASTE EMPLACEMENT EXCAVATION ',T33,8F9.2,T123,F10.2)
WRITE(IO,1500) TUCAPC,EMEQCT(LL+1),WEC,CLABOR(5),CMATL(5),
+ CEQPT(5),WEO,TEWC
1500 FORMAT(' WASTE EMPLACEMENT ',T33,6F9.2,T96,F9.2,T123,F10.2)
WRITE(IO,1505) (WHSUB(J),J=1,11)
1505 FORMAT(' SUBTOTAL ',T33,10F9.2,F10.2)
WRITE(IO,1510)
1510 FORMAT('O OTHER ROCK EXCAV. & HANDLING')
WRITE(IO,1515) SFMMC,SFTHE(1),SMMSC,SFTSLC(1),TMMSC
1515 FORMAT(' MEN & MATERIALS SHAFTS ',T33,3F9.2,T105,F9.2,
+ T123,F10.2)
WRITE(IO,1520) SFRHC,HASFEC,RHSC,RKHLC,RKHMC,RKHERC,
+ RHOC,SFTSLC(7),TRHSC
1520 FORMAT(' ROCK HOISTING & R.H. SHAFTS ',T33,6F9.2,T96,2F9.2,
+ T123,F10.2)
WRITE(IO,1525) SPECAC,SPMEQC,SPEC,SPBFC,TSPEC
1525 FORMAT(' SHAFT PILLAR EXCAVATION ',T33,3F9.2,T105,F9.2,
+ T123,F10.2)
WRITE(IO,1530) PSCMAS,SCMEQC,SCEXC,SCELBR,SCEMTC,SCEERC,SCEC,CEOC,
+ ACBFC,TSCEC
1530 FORMAT(' ALL CORRIDOR EXCAVATION ',T33,9F9.2,T123,F10.2)
WRITE(IO,1535) RKHSFC,RKHSEC,SRHC,SRHLC,SRHMC,SRHERC,SRHOC,TSRHC

```

```

1535  FORMAT('    SURFACE ROCK HANDL,STOR & DISP',T33,6F9.2,
+   T96,F9.2,T123,F10.2)
      WRITE(IO,1540) (RESUB(J),J=1,10)
1540  FORMAT('    SUBTOTAL',T33,9F9.2,T123,F10.2)
      WRITE(IO,1545)
1545  FORMAT('0 VENTILATION')
      WRITE(IO,1555) SFVSC,SFTHE(2),VSSC,VENTUC,SVSS,SFTSLC(2),TVSS
1555  FORMAT('    SUPPLY SHAFT',T33,3F9.2,T87,2F9.2,T105,F9.2,
+   T123,F10.2)
      WRITE(IO,1560) SFEXC,SFTHE(6),VEXSC,SFTSLC(6),TVEX
1560  FORMAT('    WASTE EMPL. EXHAUST SHAFT',T33,3F9.2,T105,F9.2,
+   T123,F10.2)
      WRITE(IO,1565) SFMXC,SFTHE(5),VMXSC,SFTSLC(5),TVMX
1565  FORMAT('    MINING EXHAUST SHAFT',T33,3F9.2,T105,F9.2,
+   T123,F10.2)
      WRITE(IO,1575) (VSUB(J),J=1,7)
1575  FORMAT('    SUBTOTAL',T33,3F9.2,T87,2F9.2,T105,F9.2,T123,F10.2)
      WRITE(IO,1580) SUPFCC,SUPFEC,SUPCC,SUPLBC,SUPMTC,SUPEQC,SUPUTC,
+   SUPSUB,SFDECC,TSUPP
1580  FORMAT('0 SUPPORT',T33,8F9.2,T114,F9.2,T123,F10.2)
      WRITE(IO,1585) CNTFCC,CNTEQC,CNTCSB,CNTLBR,CNTMAT,CNTEQR,CNTUTL,
+   CNTOSB,CNTBKF,CNTDD,TCONT
1585  FORMAT('0 CONTINGENCY',T33,10F9.2,F10.2)
      WRITE(IO,1590) (TOTAL(J),J=1,11)
1590  FORMAT('0 TOTAL',T33,10F9.2,F10.2)

```

C

```

      RETURN
      END
      SUBROUTINE POSTOP

```

C

```

      INCLUDE PARAM
      INCLUDE OUTCOM
      INCLUDE PSTCOM
      INCLUDE SSTCOM
      INCLUDE INCOM
      INCLUDE MNDCOM
      INCLUDE WAFCOM
      INCLUDE SFTCOM

```

C

1

C

```

      CONTINUE
      SHAFT SEALING
      LB = NBKF + 1
      LE = NBKF + IDD
      DO 450 L = 1,NST
      IF(SFTNUM(L).EQ.0) GO TO 450
      D = SFTDD(L) + 2 * SFTXR(L)
      DO 400 K=1,NWT
      WAF = WAFMV(K)
      IF(L.EQ.1) WAF = WAFLBR(K)

```

```

      IF(L.EQ.3) WAF = WAFRHP(K)
      IF(L.EQ.3.AND.SFTNUM(4).EQ.0) WAF = WAFUGW(K)
      IF(L.EQ.4) WAF = WAFNC(K)
      II = 0
      DO 400 I=LB,LE
      II = II + 1
      SEALCT(K,I) = SKFUNC(SSCON1,SSCON2,D,IEQTP5) * SFTXD(L) * SSDF(II)
+ * SFTNUM(L) * WAF + SEALCT(K,I)
400  CONTINUE
450  CONTINUE
      DO 475 K=1,NWT
      DO 475 I=LB,LE
      ANSLCT(I) = ANSLCT(I)+SEALCT(K,I)
475  CONTINUE
C
C   FACILITY DECOM
C
      DO 500 K=1,NWT
      II = 0
      DO 500 I=LB,LE
      II = II + 1
      DECOMC(K,I) = SFCCST(K) * DCOMCF * FDDF(II)
      ANDDCT(I) = ANDDCT(I) + DECOMC(K,I)
500  CONTINUE
C
C   SURVEILLANCE
C
      SURVC = 0.0
      IT = LE + 1
      ITE = ITSURV + IT - 1
      DO 600 I = IT,ITE
      SURVC = SURCST / ((1. + DRATE) **I) + SURVC
600  CONTINUE
      I = LE
      DO 700 K=1,NWT
      SURVCT(K,I) = SURVC * WAFMV(K)
      TSURVC(I) = TSURVC(I) + SURVCT(K,I)
700  CONTINUE
C
      DO 1000 L=1,NST
      D = SFTDD(L) + 2*SFTXR(L)
      SFTSLC(L) = SKFUNC(SSCON1,SSCON2,D,IEQTP5) * SFTXD(L) / ONEMIL
+ * SFTNUM(L)
1000 CONTINUE
      RETURN
      DEBUG SUBCHK
      AT 1
      END
      SUBROUTINE PPUT(K,I)

```

FEB23

```

C
C      MAKES PRIMARY WASTE EMPLACEMENT
C
      INCLUDE PARAM
      INCLUDE SFTCOM
      INCLUDE SOPCOM
      INCLUDE UOPCOM
      INCLUDE INCOM
      INCLUDE COCOM
C
9999  CONTINUE
C
      BSFILD(K,I) = FLOAT(NUMPKG(K,I)) / FLOAT(NPKGPR(K,I))
      PNLREQ(K,I) = BSFILD(K,I) * AREAPP(K) / RMPILU(K) * RMSPBS(K)
      RMFILD(K,I) = BSFILD(K,I) * RMSPBS(K)
C
      RETURN
      DEBUG SUBCHK
      AT 9999
      END
      SUBROUTINE PREOP
C
      INCLUDE PARAM
      INCLUDE PRECOM
      INCLUDE WAFCOM
      INCLUDE HUBCOM
      INCLUDE SSTCOM
      INCLUDE INCOM
C
      DIMENSION TOTFLD(NY)
1      CONTINUE
      DO 100 I=1,LL
      CSTSU(I) = 0.0
      CSTAE(I) = 0.0
      DO 100 K=1,NW
      PRODC(K,I) = 0.0
      FLDCST(K,I) = 0.0
100    CONTINUE
      DO 300 K=1,NWT
      PRODC(K,1) = TCLAND * WAFMV(K)
      DO 200 I=1,LL
      FLDCST(K,I) = OSFC(K,I) + UGFC(K,I) + SHCPY(K,I) +
+   ROCAPC(K,I) + (CPCMM(K,I) + CORMAS(K,I)) * VARMC
      1 + CSTSP(I) * WAFMV(K)
      TOTFLD(I) = TOTFLD(I) + FLDCST(K,I)
200    CONTINUE
      DO 250 I=1,LL
      STARTC(K,I) = STPCT * FLDCST(K,I)
      PRODC(K,I) = PRODC(K,I) + STARTC(K,I) + AESERV(K,I)

```

```

250 CONTINUE
300 CONTINUE
C
DO 310 K=1,NWT
DO 310 I=1,LL
AECON = AECON1 * (FLDCST(K,I)/TOTFLD(I))
AESERV(K,I) = SKFUNC(AECON,AECON2,FLDCST(K,I),IEQTP4)
310 CONTINUE
C
DO 350 I=1,LL
CSTLND(I) = TCLAND * CLNDF(I)
CSTLIC(I) = TCLICE * CLCDF(I)
DO 350 K=1,NWT
COSTLD(K,I) = CSTLND(I) * WAFMV(K)
COSTLC(K,I) = CSTLIC(I) * WAFMV(K)
CSTAE(I) = CSTAE(I) + AESERV(K,I)
CSTSU(I) = CSTSU(I) + STARTC(K,I)
350 CONTINUE
C
RETURN
DEBUG SUBCHK
AT 1
END
SUBROUTINE RECEVE
C
INCLUDE PARAM
INCLUDE RECCOM
INCLUDE SOPCOM
INCLUDE WAFCOM
C
C
1
C
CONTINUE
DO 125 I=1,NY
SRHSHP(I) = 0.0
SCHSHP(I) = 0.0
TRSHHP(I) = 0.
125 CONTINUE
DO 300 I=1,LY
DO 200 K=1,NWT
IF(RECUNT(K,I).EQ.0.OR.(RCCAP(K) + TCCAP(K)).EQ.0.) GO TO 200
TSHPCN(K,I) = RECUNT(K,I) * (CRRAIL(K,I) / RCCAP(K)
1 + (1.-CRRAIL(K,I)) / TCCAP(K))
IF(KIND (K).EQ.2) GO TO 150
SRHSHP(I) = SRHSHP(I) + TSHPCN (K,I)
GO TO 175
150 CONTINUE
SCHSHP(I) = SCHSHP (I) + TSHPCN (K,I)
175 CONTINUE

```

```

        TRSHP(I) = TRSHP(I)+(SRHSHP(I)+SCHSHP(I)) * CRRAIL(K,I)
200 CONTINUE
300 CONTINUE
C
    DO 400 I=1,LY
        NMODRH = MAXO(NMODRH,NMRH(I))
        NMODCH = MAXO(NMODCH,NRMCH(I))
        NMODPH = MAXO(NMODPH,NPMRH(I))
400 CONTINUE
C
        TCRHRM = WOCST(1) * NMODRH
        TCCHRM = WOCST(2) * NMODCH
        TCRHPM = WOCST(3) * NMODPH
C
        DO 500 K=1,NWT
            TTRHPM(K) = TCRHPM * WAFRHP(K) / ONEMIL
500 CONTINUE
        RETURN
        DEBUG SUBCHK
        AT 1
        END
        SUBROUTINE REPLAS (NA,NB,KD,NU,TP,AC,EL,NW,IFY)
C
C     NA= INTEGER MODULE REQUIREMENT FOR CURRENT YEAR
C
C     NB= INTEGER MODULE REQUIREMENT FOR PREVIOUS YEAR
C
C     AC = ANNUAL CAPACITY
C
C     KD= NOT USED
C
C     NU = NEW EQUIPMENT
C
C     TP = ANNUAL THRUPUT
C
C     EL= EQUIPMENT LIFE IN PROCESS UNITS
C
C     IFY= FIRST YEAR
C
C     NW= NOT USED
C
C     EL = EQUIPMENT LIFE
C
        PARAMETER MEOP = 200
        DIMENSION EXPLIF(MEOP), USDLIF(MEOP)
C
1     CONTINUE
        IF(IFY.NE.1) GO TO 200
        DO 100 M=1,MEOP

```

```

        EXPLIF(M) = 0.
        USDLIF(M) = 0.
100 CONTINUE
200 CONTINUE
    IF(NA.LE.NB) GO TO 630
    NU = NA - NB
    NAA = NA
    GO TO 635
630 CONTINUE
    NU = 0
    NA = NB
    NAA = NB
635 CONTINUE
    IRP = 0
    N = 1
    DO 650 M=1,NAA
        USDLIF(N) = EXPLIF(N)
        EXPLIF(N) =AMIN1(TP,AC,(EL-EXPLIF(N)))+ EXPLIF(N)
        TP = TP -AMIN1(TP,AC,(EL - EXPLIF(N)))
        IEQPL = (EXPLIF(N)+.1)
        IEL = (EL)
        IF(IEQPL.LT.IEL) GO TO 645
        IRP = IRP+1
        NA = NA-1
        IF(NA.NE.0.OR.TP.LE.0.0) GOTO 645
        NU = NU+1
        NA = 1
        N = N+1
        EXPLIF(N) = USDLIF(N-1)+TP-EL
645 CONTINUE
        N = N+1
650 CONTINUE
        NB = NA
C
C     STACK DOWN
C
        IF(IRP.EQ.0) GO TO 690
        DO 670 L=1,IRP
            NM = NA-L
            IF(NM.LE.0) NM = 1
            DO 660 M=1,NM
                EXPLIF(M) = EXPLIF(M+1)
660 CONTINUE
                EXPLIF(NM+1) = 0.
670 CONTINUE
690 CONTINUE
C
        RETURN
C

```



```

        DEBUG SUBCHK
        AT 1
        END
        SUBROUTINE ROCKOP
C
        INCLUDE PARAM
        INCLUDE PSTCOM
        INCLUDE MNDCOM
        INCLUDE WAFCOM
        INCLUDE SFTCOM
C
        DIMENSION RCKREM(NYRO)
C
C
1      CONTINUE
C      BACKFILLING - ROOMS + CORRIDORS
C
        LB = NBYR+IYRPRO-IYRCON+1
        LBB = LY+1
        IF(LB.LT.LBB) LB = LBB
        IF(BKRATE.NE.0.) GO TO 50
C      ASSUME BACKFILLING RATE = SHAFT ROCK LOWERING RATE SINCE NO
C      BACKFILLING RATE
C      IS INPUT
        BKRATE = (MXMMRH * TNPT2 * SFTNUM(1)
1      + MXMPRH * TNPT1 * NDSMP) * 3. * DPY
        50 CONTINUE
C      CALCULATE ROOMS REMAINING TO BE BACKFILLED
        BFLEFF = PCTFIL / XPANFC
        DO 100 K=1,NWT
            REMBKF(K) = UBKVOL(K) * (DENSTY*(1.-OTHBFL)+OTBDEN*OTHBFL)
1      * BFLEFF
            TREM = TREM + REMBKF(K)
        100 CONTINUE
C      CALCULATE BACKFILLING PERIOD PRIOR TO DECOMMISSIONING
        BKFYRS = TREM / BKRATE
        NBKYRS = BKFYRS + .999
        BKFLYR = BKFYRS - NBKYRS + 1.
        NBKF = NBKYRS + LB-1
        PRINT *, ' LB,NBKF,NBKYRS,BKFLYR,TREM,BKRATE',
+      LB,NBKF,NBKYRS,BKFLYR,TREM,BKRATE
C
C      BACKFILL REMAINING ROOMS
C
        IF(LB.GT.NBKF) GO TO 155
        DO 150 I=LB,NBKF
        DO 150 K=1,NWT
            RKBKF(K,I) = UBKVOL(K) / BKFYRS * (1 - OTHBFL) * DENSTY*BFLEFF
            OTBKF(K,I) = UBKVOL(K) / BKFYRS * OTHBFL * OTBDEN*BFLEFF

```

```

      IF(I.EQ.NBKF)RKBKF(K,I) = UBKVOL(K) * BKFLYR / BKFYRS
1      * (1-OTHBFL) * DENSTY*BFLEFF
      IF(I.EQ.NBKF) OTBKF(K,I) = UBKVOL(K) * BKFLYR / BKFYRS
1      * OTHBFL * OTBDEN*BFLEFF
150 CONTINUE
155 CONTINUE
C
C BACKFILL MAIN CORRIDORS
C
      II = 0
      LB = NBKF + 1
      LE = NBKF + IDD
      IF(LE.GT.NYRO) WRITE(IO,9999)
9999 FORMAT(' NUMBER OF YEARS HAS EXCEEDED THE DIMENSION NYRO'
+ /' MAKE ADJUSTMENT TO BACKFILLING RATE AND RESUBMIT')
      IF(LE.GT.NYRO) STOP
      DO 160 I=LB,LE
      II = II + 1
      DO 160 K=1,NWT.
      RKBKF(K,I) = (CORARA*M2ACRE*CORHT(1)*DENSTY*SRMM(K)/SRM*DBDF(II)
1+ TCPCLN * CORWTH(1)*CORHT(1)*DENSTY*SRMM(K)/SRM*DBDF(II))
      2 *(1-OTHBFL) * BFLEFF
      OTBKF (K,I) = RKBKF(K,I) / (1 - OTHBFL) * OTHBFL * OTBDEN / DENSTY
160 CONTINUE
C ASSUME SHAFT BACKFILLING INCLUDED IN SEALING COSTS
      LB = LY + 1
      DO 166 I=LB,LE
      DO 165 K=1,NWT
      SRKBKF(I) = SRKBKF(I) + RKBKF(K,I)
      SOTBKF(I) = SOTBKF(I) + OTBKF(K,I)
165 CONTINUE
166 CONTINUE
C ASSUME THE SAME RESOURCE REQUIREMENTS PER TONNE FOR ROCK AND
C SUPPLEMENTARY BACKFILL
C
C CALCULATE NUMBER OF BACKFILLING MACHINES OVER ALL OPERATING PERIODS
C
      IF(MSWTCH.EQ.0) GO TO 190
      NLAST = 0
      II = 0
      DO 185 I=1,LE
      BKSFT(I) = 3.
      IF (I.LE.LL) BKSFT(I) = MSHIFT(1)
      IF (I.GT.LL.AND.I.LE.LY) BKSFT(I) = MSHIFT(2)
      IF(SRRMR(I).EQ.0.) GO TO 185
      II = II+1
      NRON=5
      TP=0
      DO 170 K=1,NWT

```

```

      TP=RKBKF(K,I) + OTBKF(K,I) + TP
      TPP = TP
170  CONTINUE
      ANEQCP=ROEQPC(NRON)*BKSHFT(I)*DPY
      NBEQP = TP / ANEQCP + .9999
      CALL REPLAS(NBEQP,NLAST,K,NEQRPL(NRON,I),TP,ANEQCP,ROEQPL(NRON),
1    NW,II)
C
C  ASSUME THREE SHIFT OPERATION FOR BACKFILLING AFTER WASTE OPERATIONS
C  HAVE ENDED
C  CALCULATE BACKFILLING LABOR
      DO 180 K=1,NWT
        BKFLBR(K,I) = ROLBRR(NRON)/DPY/BKSHFT(I)*WAFMV(K)
        + * INT(TPP /ROEQPC(NRON) + .9999)
C  CALCULATE BACKFILL MATERIALS (BESIDES SUPPLEMENTARY BACKFILL)
        BKFMAT(K,I) = ROMATU(NRON) * (RKBKF(K,I) + OTBKF(K,I))
180  CONTINUE
185  CONTINUE
190  CONTINUE
C
C  ROCK LOWERING DURING BACKFILL OPERATIONS AFTER MINING HAS ENDED
C
      DO 300 I=LB,LE
        DO 300 K=1,NWT
          RCKHST(K,I) = RKBKF(K,I) + OTBKF(K,I)
          SRKHST(I) = SRKHST(I) + RCKHST(K,I)
          RCKREM(I) = -RKBKF(K,I) + RCKREM(I)
300  CONTINUE
C
C  CALCULATE ROCK HOISTING RESOURCE REQUIREMENTS
C
      IF(MSWTCH.EQ.0) GO TO 330
      NRON = 2
      NLAST = 0
      II = 0
      DO 325 I=1,LE
C  CALCULATE ROCK HOISTING EQUIPMENT REQUIREMENTS
        IF(I.LT.LL+1) BKSHFT(I) = 3.
        IF (BKSHFT(I).EQ.0.) GO TO 325
        II = II+1
        DSRKHT = ABS(SRKHST(I))
        ANEQCP = ROEQPC(NRON) * BKSHFT(I) * DPY
        NBEQP = SRKHST(I) / ANEQCP + .9999
        CALL REPLAS(NBEQP,NLAST,K,NEQRPL(NRON,I),DSRKHT,ANEQCP,
1    ROEQPL(NRON),NW,II)
C  CALCULATE ROCK HOISTING LABOR
        DO 320 K=1,NWT
          RKHLBR(K,I) = ROLBRR(NRON) / DPY / BKSHFT(I)
1    * WAFMV(K) * INT(SRKHST(I) / ROEQPC(NRON) + .9999)

```

```

C  CALCULATE ROCK HOISTING MATERIALS
    RKHMAT(K,I) = ROMATU(NRON) * RCKHST(K,I)
320 CONTINUE
325 CONTINUE
330 CONTINUE
    K = 0

C
C  SURFACE ROCK HANDLING OPERATIONS
C
    NRON = 3
    DO 350 I=LB,LE
      RKSURF(I) = RKSURF(I-1) + RCKREM(I)
350 CONTINUE
      IF(MSWTCH.EQ.0) GO TO 380
      NLAST = 0
      II = 0
      DO 370 I=1,LE
C  CALCULATE RESOURCE REQUIREMENTS
        IF(BKSHFT(I).EQ.0.) GO TO 370
        II = II+1
        DSRKHT = SRKHST(I)
        ANEQCP = ROEQPC(NRON) * BKSHFT(I) * DPY
        NBEQP = SRKHST(I) / ANEQCP + .9999
        CALL REPLAS(NBEQP,NLAST,K,NEQRPL(NRON,I),DSRKHT,ANEQCP,
1 ROEQPL(NRON),NW,II)
        DO 360 K=1,NWT
          SRHLBR(K,I) = ROLBRR(NRON) / DPY / BKSHFT(I) *
1 WAFMV(K) * INT(SRKHST(I) / ROEQPC(NRON) + .9999)
          SRHMAT(K,I) = ROMATU(NRON) * RCKHST(K,I)
360 CONTINUE
370 CONTINUE
380 CONTINUE

C
    RETURN
    DEBUG SUBCHK
    AT 1
    END
    SUBROUTINE SHAFT

C
    INCLUDE PARAM
    INCLUDE SFTCOM
    INCLUDE MNDCOM
    INCLUDE SOPCOM

C
    DIMENSION TSRRMR(200), HSTCAP(200)

C
C
1 CONTINUE
C  ROCK HOISTING SHAFT REQUIREMENTS

```

```

C      NDSMP = 0
      DO 350 I=1,LY
      IF(MXMPRH.EQ.0) GO TO 370
      IF(TNPT1.EQ.0.) GO TO 370
      TPYMP = MXMPRH * 3. * TNPT1 * DPY
      TPYMM = MXMMRH * 3. * TNPT2 * DPY
      NX = (SRKHST(I) - TPYMM) / TPYMP + .999
      NDSMP = MAX0(NDSMP,NX)
      TSRRMR(I) = SRRMR(I)
      IF(I.LE.LL) TSRRMR(I) = 3.
      HSTCAP(I) = (MXMPRH * NDSMP * TNPT1 + MXMMRH * SFTNUM(1)
1      * TNPT2) * TSRRMR(I) * DPY
      IHSTFL(I) = 0
      IF(SRKHST(I).GT.HSTCAP(I)) IHSTFL(I) = 1
350  CONTINUE
      SFTNUM(7) = NDSMP
370  CONTINUE

C      SHAFT  COST
C
C      DO 400 L=1,NST
      IF(SFTNUM(L).EQ.0) GO TO 400
      D = SFTDD(L) + 2 * SFTXR(L)
      SFTLC(L) = SKFUNC(CCCON1,CCCON2,D,IEQTP2) * SFTLD(L)
      SFTWC(L) = SKFUNC(WCCON1,WCCON2,D,IEQTP3)
      SFTSC(L) = SKFUNC(SCCON1,SCCON2,D,IEQTP1) * SFTXD(L)
      SHDFC(L) = SFTSC(L) + SFTLC(L) + SFTHE(L) + SFTWC(L) + SFTSS(L)
400  CONTINUE

C      RETURN
C      DEBUG SUBCHK
      AT 1 .
      END
      FUNCTION SKFUNC(A,B,D,I)

C
C      IF (I.EQ.1) GO TO 10
      SKFUNC = A*EXP(B*D)
      GO TO 20
10    CONTINUE
      SKFUNC = A+B*D
20    CONTINUE
      RETURN
      END
      SUBROUTINE SPUT(K,I)

C
C      MAKES SECONDARY WASTE EMPLACEMENT

```

```

C      INCLUDE PARAM
      INCLUDE SFTCOM
      INCLUDE SOPCOM
      INCLUDE UOPCOM
      INCLUDE INCOM
      INCLUDE COCOM

C
C      CALCULATE NUMBER OF SECONDARY WASTE PACKAGES
9999  CONTINUE
C
      PACKS = RECUNT(K,I) / WUPP(K)
      KN = K-MPRIME

C
      DO 100 M=1,MPRIME
      IF(COCON(KN,M).EQ.0.0) GOTO 100

C
C      LOOK FOR EMPTY ROOMS FROM PERVIOUS YEARS
C
      DO 200 II=1,I

C
C      CALCULATE THE NUMBER OF PACKAGES THAT CAN BE EMPLACED
C
      FULL = RMFILD(M,II) * NPKGPR(M,II) * COCON(KN,M) * PCTCON(KN,M)
      IF(COWAST(KN,M,II).GE.FULL) GOTO 200
      B = FULL - COWAST(KN,M,II)
      A = PACKS - B
      IF(A.LE.0.0) GOTO 210
      COWAST(KN,M,II) = FULL
      PACKS = PACKS - B
      GOTO 200
210  CONTINUE
      COWAST(KN,M,II) = COWAST(KN,M,II) + PACKS
      PACKS = 0.0
      GOTO 300

C
200  CONTINUE
100  CONTINUE
300  CONTINUE

C
C      STORE IF THERE IS NOT ENOUGH ROOM TO EMPLACE NHG WASTE
C
      IF(PACKS.LE.0.0) RETURN
      STOR(K) = STOR(K) + PACKS
      RETURN
      DEBUG SUBCHK
      AT 9999
      END
      SUBROUTINE SSTOR(K,I)

```

```

C      INCLUDE PARAM
      INCLUDE SFTCOM
      INCLUDE SOPCOM
      INCLUDE UOPCOM
      INCLUDE INCOM
      INCLUDE COCOM
C
9999  CONTINUE
      KN = K - MPRIME
      IF (STOR(K).LE.0.0) RETURN
      IF(STOR(K).GT.STOMAX(K)) GOTO 10
      IF(I.LT.ICLR) RETURN
10    CONTINUE
      IF(KIND(K).NE.3) GOTO 20
C
C      MINE FOR TRU EMPLACEMENT
C
      IF(STOR(K).GT.STOMAX(K).AND.I.LT.ICLR) GOTO 30
C
C      ICLR EXCEEDED, CLEAR OUT ALL UNEMPLACED STORED WASTE
C
      RMFILD(K,I) = STOR(K) / SWPPR(K)
      STOR(K) = 0.0
      RETURN
30    CONTINUE
C
C      STORAGE EXCEED REDUCE TO MEET STORAGE CAPACITY
C
      RMFILD(K,I) = (STOR(K) - STOMAX(K)) / SWPPR(K)
      STOR(K) = STOMAX(K)
C
      RETURN
C
20    CONTINUE
C
C      CH WASTE TO GO WITH TRU ONLY, OR BY ITSELF
C
      IF(STOR(K).EQ.0.0.OR.KIND(K).NE.2) RETURN
      DO 35 II = 1,I
      DO 40 KK = MPRIME+1,NWT
      IF(KIND(KK).NE.3) GOTO 40
      FULL = RMFILD(KK,II) * CONTRU(K) * SWPPR(K)
      B = FULL - CHTRU(II)
      IF(B.LE.0.0) GOTO 40
C
C      EMPLACEMENT CAPACITY IS AVAILABLE
C
      IF(STOR(K).GT.STOMAX(K).AND.I.LT.ICLR) GOTO 50

```

```

        IF(STOR(K).GT.B) GOTO 60
        CHTRU(II) = CHTRU(II) + STOR(K)
        STOR(K) = 0.0
        RETURN
60    CONTINUE
        CHTRU(II) = FULL
        STOR(K) = STOR(K) - B
        GOTO 40
50    CONTINUE
        C = STOR(K) - STOMAX(K)
        IF(C.GT.B) GOTO 40
        CHTRU(II) = CHTRU(II) + C
        STOR(K) = STOMAX(K)
        RETURN
40    CONTINUE
35    CONTINUE
C
C    WE HAVE SOME SH LEFT OVER AND MUST NOW MINE EMPLACEMENT ROOMS
C
        IF(STOR(K).LE.0.0) RETURN
        IF(STOR(K).GT.STOMAX(K).AND.I.LT.ICLR) GOTO 80
        RMFILD(K,I) = STOR(K) / SWPPR(K)
        STOR(K) = 0.0
        RETURN
80    CONTINUE
        B = STOR(K) - STOMAX(K)
        RMFILD(K,I) = B / SWPPR(K)
        STOR(K) = STOMAX(K)
        RETURN
        DEBUG SUBCHK
        AT 9999
        END
C
C    SUBROUTINE SURFST
C
        INCLUDE PARAM
        INCLUDE SSTCOM
        INCLUDE WAFCOM
        INCLUDE INCOM
        INCLUDE RECCOM
        INCLUDE MNDCOM
        INCLUDE SOPCOM
        INCLUDE UOPCOM
C
        IB = LL + 2
        LE = LY + IDD
C
C    CONTINUE
C

```



```

      CALL WAFAM2
C
C SURFACE FACILITY SECTION
C
C SUPPORT FACILITY CAPITAL COSTS
C
      DO 200 I=1,LL
      DO 200 K=1,NWT
      DO 200 L=1,NSC
      SFCAPC(K,I) = SFCST(L) * WAFMV(K) * SFCS(I) + SFCAPC(K,I)
200 CONTINUE
C
C SURFACE WASTE OPERATIONS CAPITAL COSTS ARE CALCULATED IN HUB
C
C UNDERGROUND WASTE OPERATIONS CAPITAL COSTS
C
      DO 300 I=1,LL
      DO 300 K=1,NWT
      DO 210 L=1,NUO
      UFCAPC(K,I) = UOCST(L) * WAFMV(K) * SFCS(I)
      1 + UFCAPC(K,I)
210 CONTINUE
C
C TRANSPORTATION OPERATIONS CAPITAL COSTS
C
      L = KIND(K)
      IF (KIND(K).EQ.3) L=1
      WAF = WAFCN(K)
      IF(KIND(K).EQ.2) WAF = WAFNC(K)
      TRCAPC(K,I) = TOCST(L) * WAF * SFCS(I) + TRCAPC(K,I)
C
C ROCK OPERATIONS CAPITAL COSTS
C
      DO 250 L=1,NRO
      ROCAPC(K,I) = ROCST(L) * WAFMV(K) * SFCS(I)
      1 + ROCAPC(K,I)
250 CONTINUE
C
C WASTE EMPLACEMENT EXCAVATION CAPITAL COSTS
C
      DO 270 L=1,NEE
      EECAPC(K,I) = EECST(L) * WAFMET(K,L) * SFCS(I)
      1 + EECAPC(K,I)
270 CONTINUE
300 CONTINUE
C
C SUPPORT FACILITY EQUIPMENT COSTS

```

C

```

LB = LL + 1
DO 400 L=1,NSC
DO 400 K=1,NWT
DO 350 I=1,LL
SFEQPC(K,LB) = SFECST(L) * SFCS(I) * WAFMV(K) + SFEQPC(K,LB)
TSFEC(L) = TSFEC(L) + SFECST(L) * SFCS(I) * WAFMV(K)
350 CONTINUE
II = 0
DO 400 I=LB,LY
II = II + 1
NSFELF = SFEQPL(L)
IF(II.GT.NSFELF) GO TO 360
GO TO 400
360 CONTINUE
SFEQPC(K,I) = SFEQPC(K,I) + SFECST(L) * WAFMV(K)
TSFEC(L) = TSFEC(L) + SFECST(L) * WAFMV(K)
II = 0
400 CONTINUE
DO 450 I=LB,LY
DO 450 K=1,NWT
SFEQCT(I) = SFEQCT(I) + SFEQPC(K,I)
450 CONTINUE

```

C
C
C
C

SITE PREPARATION COSTS

```

DO 650 I=1,LL
CSTSP(I) = SPUC * SFCS(I) * SOA
DO 650 K=1,NWT
ASPC(K,I) = SPUC * SOA * WAFMV(K) * SFCS(I)
650 CONTINUE

```

C
C
C

SUPPORT FACILITY LABOR REQUIRMENTS

```

DO 800 L=1,NSC
S = SLBR(L)
DO 800 I=LB,LY
IF(S.EQ.-1) S=SRRMR(I)
IF(S.EQ.-2) S=SRRMC(I)
DO 800 K=1,NWT
OWLBR(I) = SFLBRR(L) * S * SFLBRC(L) * WAFLBR(K) * OVRHDF
1 + OWLBR(I)
OWOPC(K,I) = OWOPC(K,I) + SFMATU(L) * SFMATC(L) * WAFMV(K)
+ +SFLBRR(L) * S * SFLBRC(L) * WAFLBR(K) * OVRHDF
OTMATC(K,I) = SFMATC(L) * SFMATU(L) * WAFMV(K)
OTMTCT(I) = OTMTCT(I) + OTMATC(K,I)
TMATCT(I) = TMATCT(I) + OTMATC(K,I)
800 CONTINUE

```



```

      B = FULL - COWAST(KN,M,II)
      A = PACKS - B
      IF(A.LT.0.0) GOTO 310
      COWAST(KN,M,II) = FULL
      PACKS = PACKS - B
      GOTO 200
310  CONTINUE
      COWAST(KN,M,II) = COWAST(KN,M,II) + PACKS
      PACKS = 0.0
      GOTO 400
300  CONTINUE
200  CONTINUE
100  CONTINUE
400  CONTINUE
C
C  STORE IS THERE IS NOT ENOUGH ROOM TO EMPLACE CH WASTES
C
      IF(PACKS.LE.0.0) RETURN
      STOR(K) = STOR(K) + PACKS
      RETURN
      DEBUG SUBCHK
      AT 9999
      END
      SUBROUTINE UTILITY
C
      INCLUDE PARAM
      INCLUDE OUTCOM
      INCLUDE UTLCOM
      INCLUDE WAFCOM
      INCLUDE RECCOM
      INCLUDE SOPCOM
      INCLUDE MNDCOM
C
1    CONTINUE
      LE=LY
      LB = LL + 1
C
C    ELECTRIC POWER
C
      DO 200 I=LB,LE
      HPY = 8760
      IF(MOD(I,4).EQ.0) HPY = 8784
      DO 200 K=1,NWT
      RPPOW= RAPP*NRMRH(I)*WAFCN(K)*8.*AMAX1(SRRMR(I),SRRMC(I))*WRKYR
      SFSPOW= SFSP*8.*AMAX1(SRRMR(I),SRRMC(I))*WRKYR*WAFCN(K)
      OWHPOW= OWHP*8.*AMAX1(SRRMR(I),SRRMC(I))*WRKYR*WAFLBR(K)
      SFPPOW = SFP * HPY * WAFLBR(K)
      ROPOW= MROP*RKMSM(K,I)*8.*AMAX1(SRRMR(I),SRRMC(I))*WRKYR

```

```

VNTPOW = RVP * SIZE * HPY * WAFMVY(K,I)
ACPOW = ACP * HPY * WAFMVY(K,I)
POWER(K,I) = RPPOW + SFSPOW + OWHPOW + SFPPOW + ROPOW + VNTPOW +
1 ACPOW
POWER(NWP1,I) = POWER(NWP1,I) + POWER(K,I)
POWCST(K,I) = POWER(K,I) * UPC / 1000.
POWCST(NWP1,I) = POWCST(NWP1,I) + POWCST(K,I)
RFPOWC = RFPOWC + ((RPFOW + OWHPOW) * UPC / 1000. ) / ONEMIL
PFPOWC(K) = PFPOWC(K) + (SFSPOW * UPC / 1000.) / ONEMIL
SFPOWC = SFPOWC + (SFPPOW * UPC / 1000.) / ONEMIL
ROPOWC = ROPOWC + (ROPOW * UPC / 1000.) / ONEMIL
VENTPC = VENTPC + (VNTPOW * UPC / 1000.) / ONEMIL
ACPOWC = ACPOWC + (ACPOW * UPC / 1000.) / ONEMIL
200 CONTINUE

C
C STEAM POWER
C
DO 300 I=LB,LE
DO 300 K=1,NWT
HEAT = HEATS * ((UBFRPY(I) * RMLN(K) * RMWTH(K) + TCPCLN
1 * CORWTH(1))/ M2ACRE + CORARA) * WAFMVY(K,I)
RSTEAM = RAPS * NRMHR(I) * WAFCN(K)
STEAM(K,I) = HEAT + RSTEAM + ORS * WAFLBR(K)
STEAM(NWP1,I)=STEAM(NWP1,I) + STEAM(K,I)
STMCST(K,I) = STEAM(K,I) * STMC
STMCST(NWP1,I) = STMCST(NWP1,I) + STMCST(K,I)
RSTMC = RSTMC + (RSTEAM * STMC) / ONEMIL
OSTMC = OSTMC + ((HEAT + ORS * WAFLBR(K))*STMC) / ONEMIL
300 CONTINUE

C
C WATER REQUIREMENTS
C
DO 400 I=LB,LE
DO 400 K=1,NWT
WATER(K,I) = (VOLW + VOLDW)*WAFLBR(K)
WATER(NWP1,I) = WATER(NWP1,I) + WATER(K,I)
WATERC(K,I) = (VOLW * WTRC + VOLDW * DWTRC) * WAFLBR(K)
WATERC(NWP1,I) = WATERC(NWP1,I) + WATERC(K,I)
400 CONTINUE

C
C DIESEL FUEL REQUIREMENTS
C
DO 500 I=LB,LE
DO 500 K=1,NWT
DSURF = SUD * WRKYR * WAFAL(K)
DSMINE = UGD * NUNPKG(K,I)
DSRAIL = RED * TSHPCN(K,I)
DIESEL(K,I) = DSURF + DSMINE + DSRAIL
DIESEL(NWP1,I) = DIESEL(NWP1,I) + DIESEL(K,I)

```

```

        DSLCST(K,I) = (DSURF + DSMINE + DSRAIL) * DSLC
        DSLCST(NWP1,I) = DSLCST(NWP1,I) + DSLCST(K,I)
        SDSLC = SDSLC + (DSURF * DSLC) / ONEMIL
        UDSLC = UDSLC + (DSMINE * DSLC) / ONEMIL
        RDSLC = RDSLC + (DSRAIL * DSLC) / ONEMIL
500  CONTINUE
C
C      TOTAL
C
      DO 600 I=LB,LE
      DO 600 K=1,NWT
      ANUTC(K,I) = POWCST(K,I) + STMCST(K,I) + WATERC(K,I) + DSLCST(K,I)
      ANUTC(NWP1,I) = ANUTC(NWP1,I) + ANUTC(K,I)
600  CONTINUE
C
      RETURN
      DEBUG SUBCHK
      AT 1
      END
      SUBROUTINE WAFAM
C
      INCLUDE PARAM
      INCLUDE WAFCOM
      INCLUDE SOPCOM
      INCLUDE MNDCOM
      INCLUDE SSTCOM
      INCLUDE RECCOM
      INCLUDE UTLCOM
      INCLUDE UOPCOM
      INCLUDE PSTCOM
      INCLUDE COCOM
C
      DIMENSION TLBRWT(NW)
1    CONTINUE
      LE = LY + IDD
C
      DO 30 K=1,NWT
      NPSMWT(K)=0.0
      RECUWT(K) = 0.0
      RHPKG(K) = 0.
      CHRECW(K)=0.0
      SMNDWT(K) = 0.0
      RHRECW(K) = 0.0
      NPSMWT(K) = 0
30   CONTINUE
      DO 50 I=1,LE
      NPKSTM(I) = 0
50   CONTINUE
C

```

```

SUMREC = 0.0
NPKSUM = 0.0
RHRECU = 0.0
CHRECU = 0.0
SRHPKG = 0.
DO 100 I=1,LE
DO 100 K=1,NWT
SUMREC = SUMREC + RECUNT(K,I)
NPKSTM(I) = NPKSTM(I) + NUNPKG(K,I)
NPKSUM = NPKSUM + NUNPKG(K,I)
NPSMWT(K) = NPSMWT(K) + NUNPKG(K,I)
RECUWT(K) = RECUWT(K) + RECUNT(K,I)
IF(KIND(K).EQ.2) GO TO 70
RHRECU = RHRECU + RECUNT(K,I)
RHRECUW(K) = RHRECUW(K) + RECUNT(K,I)
RHPKG(K) = RHPKG(K) + NUNPKG(K,I)
SRHPKG = SRHPKG + NUNPKG(K,I)
GO TO 100
70 CONTINUE
CHRECU = CHRECU + RECUNT(K,I)
CHRECUW(K) = CHRECUW(K) + RECUNT(K,I)
100 CONTINUE
SUMIND = 0.
DO 200 I=1,LE
DO 200 K=1,NWT
SUMIND = SUMIND + RKMSM(K,I)
SMNDWT(K) = SMNDWT(K) + RKMSM(K,I)
200 CONTINUE
DO 300 K=1,NWT
WAFAL (K) = RECUWT(K)/SUMREC
WAFCN (K) = RHRECUW(K) / RHRECU
WAFRHP(K) = RHPKG(K) / SRHPKG
IF(CHRECU.NE.0.0) WAFNC(K) = CHRECUW(K)/CHRECU
WAFMV(K) = SMNDWT(K)/SUMIND
300 CONTINUE
DO 400 K=1,NWT
WAFUGW(K) = FLOAT(NPSMWT(K))/FLOAT(NPKSUM)
DO 400 I=1,LE
IF(SRKMD(I).EQ.0.) GO TO 400
WAFMVY(K,I) = RKMSM(K,I)/SRKMD(I)
400 CONTINUE
C
DO 435 L=1,NEE
DO 430 K=1,NWT
CALL EMTYPE(K,IMET(K),KIND(K),IEMP(K),MPRIME)
IF(IEMP(K).NE.L) GO TO 430
DO 425 I=1,LY
SUMMET(K,L) = SUMMET(K,L) + NUNPKG(K,I)
SUMET(L) = SUMET(L) + NUNPKG(K,I)

```

```

425 CONTINUE
430 CONTINUE
435 CONTINUE
    DO 440 K=1,NWT
    DO 440 L=1,NEE
    IF(SUMET(L).EQ.0.) GO TO 440
    WAFMET(K,L) = SUMMET(K,L) / SUMET(L)
440 CONTINUE
C
C
    RETURN
C
    ENTRY WAFAM2
C
C
    DO 450 K=1,NWT
    WAFUTL(K) = 0.
    WAFSHC(K) = 0.
    WAFRSH(K) = 0.
450 CONTINUE
    SUMUTL = 0.
    SUMSHC = 0.
    SUMRSH = 0.
    DO 500 I=1,LE
    DO 500 K=1,NWT
    WAFUTL(K) = WAFUTL(K) + ANUTC(K,I)
    SUMUTL = SUMUTL + ANUTC(K,I)
    WAFSHC(K) = WAFSHC(K) + TSHPCN(K,I)
    SUMSHC = SUMSHC + TSHPCN(K,I)
    WAFRSH(K) = WAFRSH(K) + TSHPCN(K,I) * CRRAIL(K,I)
    SUMRSH = SUMRSH + TSHPCN(K,I) * CRRAIL(K,I)
500 CONTINUE
    DO 700 K=1,NWT
    WAFUTL(K) = WAFUTL(K) / SUMUTL
    WAFSHC(K) = WAFSHC(K) / SUMSHC
    WAFRSH(K) = WAFRSH(K) / SUMRSH
700 CONTINUE
C
C
    DO 800 I=LL+1,LY
    DO 800 K=1,NWT
    SUMLBR = SUMLBR + RECLBR(K,I) + PKGLBR(K,I) + TRNLBR(K,I)
    + + EMPLBR(K,I) + EEXLBR(K,I) + RMNLBR(K,I) + RKHLBR(K,I)
    + + SRHLBR(K,I) + BKFLBR(K,I)
C
    TLBRWT(K) = TLBRWT(K) + RECLBR(K,I) + PKGLBR(K,I) + TRNLBR(K,I)
    + + EMPLBR(K,I) + EEXLBR(K,I) + RMNLBR(K,I) + RKHLBR(K,I)
    + + SRHLBR(K,I) + BKFLBR(K,I)
800 CONTINUE

```



```

DO 810 K=1,NWT
WAFLBR(K) = TLBRWT(K) / SUMLBR
810 CONTINUE
RETURN
DEBUG SUBCHK
AT 1
END
SUBROUTINE WASTOP
C
  INCLUDE PARAM
  INCLUDE SFICOM
  INCLUDE SOPCOM
  INCLUDE UOPCOM
  INCLUDE INCOM
  INCLUDE COCOM
  INCLUDE WAFCOM
  INCLUDE MNDCOM
C
  DIMENSION EXMAC(NW)
  DIMENSION SRRMP(NY), TTANF(NW)
C
  NSHFLG=0
1  CONTINUE
  DO 2 I=1,4
  MODFLG(I) = 0
2  CONTINUE
C
  DO 100 I=1,NY
  AVL RPA(I) = 0.
  DO 100 K=1,NWT
C
  NPKGPR(K,I) = 0
  WSTVOL(K,I) = 0.
100 CONTINUE
C
  DO 101 K=1,NWT
  CALL EMTYPE(K,IMET(K),KIND(K),IEMP(K),MPRIME)
  AREAPP(K)=PANLEN(K)*PANWTH(K)/M2ACRE
101 CONTINUE
C
C  LABOR SECTION
C
C
C  ROOM PILLAR UNITS AND MAX PACKAGES/ROW
C
  DO 190 K=1,NWT
  RMPILU(K)=PANLEN(K)*XRASHO(K)/RMWTH(K)
  RMSPBS(K) = FLOAT(RMPILU(K)) / FLOAT(NBSPPA(K))
  HOLD(K) = PKGLN(K) * NPKPEU(K) + CAPLEN(K) + HOLDEL(K)

```

```

      TANF(K)=RMLN(K)*RMWTH(K)/XRASHO(K)/M2ACRE
      IF(IMET(K).EQ.4.OR.IMET(K).GT.7.AND.IMET(K).LT.13)
+      TANF(K) = HPMIN(K) * (HOLD(K) + RMWTH(K))
      1 /M2ACRE
190 CONTINUE
C
      DPY = WRKYR
      NM = 1
      LB = LL + 1
      DO 150 K=1,NWT
      IF(KIND(K).EQ.2) GO TO 130
      NMRH(LB) = 1
      NPMRH(LB) = 1
      GO TO 150
130 CONTINUE
      NRMCH(LB) = 1
150 CONTINUE
      DO 900 I=LB,LY
      DO 300 K=1,NWT
      RECUNT(K,I) = WSTSUP(K,I)
300 CONTINUE
400 CONTINUE
      SURR= 0.
      SRMCAP = 0.
      SURC = 0.
      SRSFTR = 0.
      SRSFTC = 0.
      SUPR = 0.
      SUPC = 0.
      SCMCAP = 0.
      SPMCAP = 0.
      DO 430 K=1,NWT
      IF(KIND(K).EQ.2) GO TO 420
C
C      SUM RECEIVING UNITS - RH
C
      SURR = SURR + RECUNT(K,I)
C
C      SUM PACKAGING UNITS - RH
C
      SUPR = SUPR + RECUNT(K,I) / WUPP(K)
C
C      RECEIVING REQUIREMENTS RH
C
      RMCAP = 0.0
      IF(TTTR(K).EQ.0.0) GO TO 410
      RMCAP = ( TTTR(K) * RECUNT(K,I)/RCCAP(K) * CRRAIL(K,I))
410 CONTINUE
      IF(TTTK(K).EQ.0.0) GO TO 415

```

```

      RMCAP = RMCAP + ( TTTK(K) * RECUNT(K,I) / TCCAP(K)
+      * (1. - CRRAIL(K,I)))
415  CONTINUE
C
C      SUM RECEIVING MODULE CAPACITY - RH
C
      SRMCAP = SRMCAP + RMCAP
C
C      PACKAGING MODULE CAPACITY
C
C      SUM PACKAGING MODULE CAPACITY OVER WASTE TYPE
C
      SPMCAP = SPMCAP + RECUNT(K,I) / WUPP(K) / PKGCAP(K)
      GO TO 430
420  CONTINUE
C
C      RECEIVING REQUIREMENTS - CH
C
      CMCAP = 0.0
      IF (TTTR(K).NE.0.0)
1    CMCAP = (TTTR(K) * RECUNT(K,I) / RCCAP(K) * CRRAIL(K,I))
      IF (TTTK(K).NE.0.0)
1    CMCAP = CMCAP+ ( TTTK(K) * RECUNT(K,I) / TCCAP(K)
+      * (1. - CRRAIL(K,I)))
C
C      SUM RECEIVING MODULE CAPACITY - CH
C
      SCMCAP = SCMCAP + CMCAP
C
C      SUM RECEIVING UNITS - CH
C
      SURC = SURC + RECUNT(K,I)
      SUPC = SUPC + RECUNT(K,I) / WUPP(K)
430  CONTINUE
C
C      MODULE REQUIREMENTS - RH RECEIVING
C
      FMOD = 0.0
      IF(NRBPRM.EQ.0) GOTO 460
      FMOD = SRMCAP / DPY / NALSH / NRBPBM
      NRMHR(I) = FMOD + .9999
      NRMHR(I) = MAX0(NRMHR(I),NRMHR(I-1))
      IF(NRMHR(I).LE.NRHRMA) GO TO 450
C
C      MAX MODULE EXCEEDED - REDUCE RECEIVING UNITS
C
C
      SET FLAG 'ALLOWABLE NUMBER OF R-H RECEIVING MODULES CONSTRAINING

```

```

C      R WASTE RECEIPT RATES BELOW INPUT VALUES
C
C      MODFLG(1) = 1
C
C      NRMRH(I) = NRHRMA
C      DO 440 K=1,NWT
C      IF(KIND(K).EQ.2) GOTO 440
C      RECUNT(K,I) = RECUNT(K,I) * NRHRMA / FMOD
C      IF(RECUNT(K,I).LE.0.) RECUNT(K,I)= 0.
440  CONTINUE
C      GO TO 400
450  CONTINUE
C
C      SHIFT REQUIREMENT FOR RECEIVING MODULE - REMOTE HANDLED WASTE
C
C      SRRMR(I) = SRMCAP / NRMRH(I) / DPY / NRBPRM
C
C      460  CONTINUE
C      RH PACKAGING MODULE REQUIREMENTS
C
C      ASSUME SHIFT REQUIREMENTS FOR RH PACKAGING & RECEIVING ARE EQUAL
C
C      FMOD = SPMCAP / NALSH / DPY
C      NPMRH(I) = FMOD + .9999
C      NPMRH(I) = MAX0(NPMRH(I) , NPMRH(I-1))
C      SRRMP(I) = SPMCAP/DPY/NPMRH(I)
C      IF (SRRMP(I).GT.SRRMR(I)) SRRMR(I) = SRRMP(I)
C
C      MODULE REQUIREMENTS - CH RECEIVING
C
C      FMOD = 0.0
C      IF(NCBPRM.EQ.0) GOTO 490
C      FMOD = SCMCAP / DPY / NALSH / NCBPRM
C      NRMCH(I) = FMOD + .9999
C      NRMCH(I) = MAX0(NRMCH(I) , NRMCH(I-1))
C      IF(NRMCH(I).LE.NCHRMA) GO TO 500
C
C      MAX MODULE EXCEEDED - REDUCE RECEIVING UNITS
C
C      SETS FLAG - ALLOWABLE NUMBER OF C-H RECEIVING MODULES IS
C      CONSTRAINING CONTACT WASTE RECEIPT RATES BELOW INPUT VALUES
C
C      MODFLG(2) = 1
C
C      NRMCH(I) = NCHRMA
C      DO 480 K=1,NWT
C      IF(KIND(K).EQ.1.OR.KIND(K).EQ.3) GO TO 480
C      RECUNT(K,I) = RECUNT(K,I) * NCHRMA / FMOD
C      IF(RECUNT(K,I).LE.0.) RECUNT(K,I)= 0.

```

```

480 CONTINUE
    GO TO 400
500 CONTINUE
C
C   SHIFT REQUIREMENTS - CH RECEIVING
C
470 CONTINUE
    SRRMC(I) = 0.0
    IF (SCMCAP.NE.0.0) SRRMC(I) = SCMCAP / NRMCH(I) / DPY / NCBPRM
C
C   SHAFT SHIFT REQUIREMENTS
C
490 CONTINUE
    DO 540 K=1,NWT
    IF(KIND(K).EQ.2) GO TO 520
    SRSFTR = SRSFTR + (RECUNT(K,I)/WUPP(K))/(NPKPTR(K)*MTRPSD*DPY)
    GO TO 540
520 CONTINUE
    SRSFTC = SRSFTC + (RECUNT(K,I)/WUPP(K))/(NPKPTC(K)*MTCPSD*DPY)
540 CONTINUE
    ISTOP = 0
    DO 545 K=1,NWT
    IF(KIND(K).EQ.2) GOTO 547
    IF (RECUNT(K,I).NE.0.0.AND.SRSFTR.EQ.0.0.OR.NPKPTR(K).EQ.0)
    & ISTOP = 1
    GOTO 545
547 CONTINUE
    IF (RECUNT(K,I).NE.0.0.AND.SRSFTC.EQ.0.0.OR.NPKPTC(K).EQ.0)
    & ISTOP = 1
545 CONTINUE
    IF(ISTOP.EQ.1) GOTO 1630
    NFLG = 0
    NFLG1 = 0
555 CONTINUE
    IF(SFTNUM(3).LT.0) GO TO 570
    IF(SFTNUM(3).EQ.0) GO TO 580
    IF(SRSFTR.LE.NALSH * SFTNUM(3)) GO TO 580
    IF (NSHFLG.EQ.1) GOTO 570
    IF(NFLG1.GT.0) GOTO 580
C
C   SET FLAG - CAPACITY OF R-H WASTE SHAFT CONSTRAINING WASTE
C   RECEIPT RATES BELOW INPUT VALUES
C
MODFLG(3) = 1
C
FMOD = SRSFTR / (NALSH * SFTNUM(3))
DO 560 K=1,NWT
    IF(KIND(K).EQ.2. AND.SFTNUM(4).NE.0) GO TO 560
    RECUNT(K,I) = AINT(RECUNT(K,I) / FMOD)

```

```

      IF(RECUNT(K,I).LT.0.) RECUNT(K,I) = 0.
560  CONTINUE
      NFLG1 = NFLG1+1
      GO TO 400
570  CONTINUE
      NSHFLG = 1
      IF(SFTNUM(4).NE.0) GO TO 575
      SRSFTR = SRSFTR + SRSFTC
      SRSFTC = 0.
      NFLG = 1
575  IF(SRSFTR.NE.0.0) SFTNUM(3) = (SRSFTR / NALSH) + .9999
      SRSFTR = SRSFTR / SFTNUM(3)
580  CONTINUE
      IF(NFLG.EQ.1) GO TO 610
      IF(SFTNUM(4).LT.0) GO TO 600
      IF(SFTNUM(4).GT.0) GO TO 585
      IF(SFTNUM(3).EQ.0) GO TO 1600
      SRSFTR = SRSFTR + SRSFTC
      SRSFTC = 0.
      NFLG = 1
      GO TO 555
585  CONTINUE
      IF(SRSFTC.LE. NALSH * SFTNUM(4)) GO TO 610
C
C      SET FLAG - CAPACITY OF C-H WASTE SHAFT CONSTRAINING
C      CONTACT HANDLED WASTE RECEIPT RATES BELOW INPUT VALUES
C
      MODFLG(4) = 1
C
      FMOD = SRSFTC / (NALSH * SFTNUM(4))
      DO 590 K=1,NWT
      IF(KIND(K).EQ.1.OR.KIND(K).EQ.3) GO TO 590
      RECUNT(K,I) = RECUNT(K,I) / FMOD
      IF(RECUNT(K,I).LT.0.) RECUNT(K,I) = 0.
590  CONTINUE
      GO TO 400
600  CONTINUE
      IF(SRSFTC.NE.0.0) SFTNUM(4) = (SRSFTC / NALSH) + .9999
      SRSFTC = SRSFTC / SFTNUM(4)
610  CONTINUE
      SRRMR(I) = AMAX1(SRRMR(I),SRSFTR)
      SRRMC(I) = AMAX1(SRRMC(I),SRSFTC)
      DO 650 K=1,NWT
      IF(SRRMR(I).EQ.0.0) GO TO 620
      IF(WUPP(K).EQ.0.) GO TO 650
      NUPPKG(K,I) = RECUNT(K,I) / WUPP(K) + .9999
      NEMPU(K,I) = NUPPKG(K,I) / NPKPEU(K) + .9999
      IF(KIND(K).EQ.2) GO TO 620
      SUMRPK(I) = SUMRPK(I) + NUPPKG(K,I)

```

```

SUMRRC(I) = SUMRRC(I) + RECUNT(K,I)
TRANSPT(1,I) = NUNPKG(K,I) / TOEQPC(1) / DPY / SRRMR(I)+
+ TRANSPT(1,I)
EMPMCH(1,I) = NEMPU(K,I)*NPKPEU(K) / (UOEQPC(1) * SRRMR(I) * DPY )
1 + EMPMCH(1,I)
GO TO 650
620 CONTINUE
IF(SRRMC(I).EQ.0.0) GO TO 650
SUMCPK(I) = SUMCPK(I) + NUNPKG(K,I)
SUMCRC(I) = SUMCRC(I)+RECUNT(K,I)
TRANSPT(2,I) = NUNPKG(K,I) / TOEQPC(2) / DPY / SRRMC(I)+
+ TRANSPT(2,I)
EMPMCH(2,I) = NEMPU(K,I)*NPKPEU(K) / ( UOEQPC(2) * SRRMC(I) * DPY)
1 + EMPMCH(2,I)
650 CONTINUE
TRANSPT(1,I) = AINT(TRANSPT(1,I) + .9999)
TRANSPT(2,I) = AINT(TRANSPT(2,I) + .9999)
EMPMCH(1,I) = AINT(EMPMCH(1,I) + .9999)
EMPMCH(2,I) = AINT(EMPMCH(2,I) + .9999)
655 CONTINUE
A = 0.0
IF(SRMCAP.EQ.0.0) GOTO 656
A = SURR / SRMCAP
656 CONTINUE
B = 0.0
IF(SCMCAP.EQ.0.0) GOTO 657
B = SURC / SCMCAP
657 CONTINUE
C = 0.0
IF(SPMCAP.EQ.0.0) GOTO 658
C = SUPR / SPMCAP
658 CONTINUE
RMMAX = AMAX1(RMMAX,A)
CMMAX = AMAX1(CMMAX,B)
PMMAX = AMAX1(PMMAX,C)
C
C MANPOWER REQUIREMENTS
C
ISR = SRRMR(I) + .999
DO 670 L=1,ISR
SFTR(L) = AMIN1(SRRMR(I) - (L-1), 1.)
670 CONTINUE
ISC = SRRMC(I) + .999
DO 680 L=1,ISC
SFTC(L) = AMIN1(SRRMC(I) - (L-1), 1.)
680 CONTINUE
C
C RH
C

```

```

DO 690 L=1,ISR
C
C RECEIVING
C
RHRLBR(I,L) = WOLBRR(1) * SFTR(L) * NMRH(I)
C
C PACKAGING
C
RHPLBR(I,L) = WOLBRR(3) * SFTR(L) * NPMRH(I)
C
C
C UNDERGROUND
C
C TRANSPORTATION
C   TRRLBR(I,L) = TRRLBR(I,L) + TOLBRR(1) * SFTR(L) * TRNSPT(1,I)
C EMPLACEMENT
C   ERRLBR(I,L) = ERRLBR(I,L) + UOLBRR(1) * SFTR(L) * EMPMCH(1,I)
690 CONTINUE
C
C CH
C
DO 750 L=1,ISC
C
C RECEIVING
C
CHRLBR(I,L) = WOLBRR(2) * SFTC(L) * NRMCH(I)
C
C
C UNDERGROUND
C
C TRANSPORTATION
C   TRCLBR(I,L) = TRCLBR(I,L) + TOLBRR(2) * SFTC(L) * TRNSPT(2,I)
C EMPLACEMENT
C   ERCLBR(I,L) = ERCLBR(I,L) + UOLBRR(2) * SFTR(L) * EMPMCH(2,I)
750 CONTINUE
DO 756 K=1,NWT
IS = ISR
IF(KIND(K).EQ.2) IS = ISC
DO 755 L=1,IS
IF(KIND(K).EQ.2) GO TO 754
IF(SUMRRC(I).NE.0.0)
+RECLBR(K,I) = RECLBR(K,I) + RHRLBR(I,L) * RECUNT(K,I) / SUMRRC(I)
IF (SUMRPK(I).NE.0.0)
+ PKGLBR(K,I) = PKGLBR(K,I) + RHPLBR(I,L) * NUMPKG(K,I) / SUMRPK(I)
IF (SUMRPK(I).NE.0.0)
+TRNLBR(K,I) = TRNLBR(K,I) + TRRLBR(I,L) * NUMPKG(K,I) / SUMRPK(I)
IF (SUMRPK(I).NE.0.0)

```



```

      +EMPLBR(K,I) = EMPLBR(K,I) + ERRLBR(I,L) * NUMPKG(K,I) / SUMRPK(I)
      GO TO 755
754  CONTINUE
      RECLBR(K,I) = RECLBR(K,I) + CHRLBR(I,L) * NUMPKG(K,I) / SUMCPK(I)
      TRNLBR(K,I) = TRNLBR(K,I) + TRCLBR(I,L) * NUMPKG(K,I) / SUMCPK(I)
      EMPLBR(K,I) = EMPLBR(K,I) + ERCLBR(I,L) * NUMPKG(K,I) / SUMCPK(I)
755  CONTINUE
      WOLABR(K,I) = RECLBR(K,I) + PKGLBR(K,I) + TRNLBR(K,I) + EMPLBR(K,I)
756  CONTINUE
C
C  MATERIALS REQUIREMENTS FOR RECEIVING
C
      DO 1100 K=1,NWT
      RCMATR(K,I) = (WOMATU(1,K) + WOMATU(2,K)) * RECUNT(K,I)
C  MATERIALS REQUIREMENTS FOR PACKAGING
      PKMATR(K,I) = WOMATU(3,K) * NUMPKG(K,I)
C  MATERIALS REQUIREMENTS FOR TRANSPORT
      TRMATR(K,I) = (TOMATU(1,K) + TOMATU(2,K)) * NUMPKG(K,I)
1100  CONTINUE
C
C
C  EMPLACEMENT DENSITY
C
      DO 700 K=1,NWT
      IF(PITCH(K,I).EQ.0.) PITCH(K,I)=HPMIN(K)
      IF(K.GT.MPRIME) GOTO 695
      NPPRW (K,I) = AINT((RMLN(K)-2.*CLSEMP(K))/PITCH (K,I) + 1.) *
&    PKGPUP(K)
      NPKGPR(K,I)=NPPRW(K,I)*NRWSPR(K)
      HTLDNF(K,I)=NPKGPR(K,I)*TDWI(I,K)*WUPP(K)/TANF(K)
      IF(IMET(K).EQ.4.OR.IMET(K).GT.7.AND.IMET(K).LT.13)
+    HTLDNF(K,I) = PKGPUP(K)*TDWI(I,K)*WUPP(K)/TANF(K)
      IF(HTLDNF(K,I).LE.NFTL(K)) GO TO 645
C  NEAR FIELD THERMAL LIMIT EXCEEDED
      IF(IMET(K).NE.4.AND.IMET(K).LE.7.OR.IMET(K).GT.12) GOTO 640
      TTANF(K) = PKGPUP(K)*TDWI(I,K)*WUPP(K)/NFTL(K)
      PITCH(K,I) = TTANF(K)*M2ACRE/(HOLD(K)+RMWTH(K))
      NPPRW(K,I) = AINT((RMLN(K)-2.*CLSEMP(K))/PITCH(K,I)+1.) * PKGPUP(K)
&    )
      NPKGPR(K,I)=NPPRW(K,I)*NRWSPR(K)
      HTLDNF(K,I) = PKGPUP(K)*TDWI(I,K)*WUPP(K)/TTANF(K)
      GO TO 645
640  CONTINUE
      PKGTD=TDWI(I,K)*WUPP(K)
      NHLPPA=NFTL(K)/PKGTD
      NPKGPR(K,I) = NHLPPA * TANF(K)
      NPPRW(K,I)=NPKGPR(K,I)/NRWSPR(K)
      PITCH(K,I)= (RMLN(K)-2.*CLSEMP(K))/((NPPRW(K,I)-1.) * PKGPUP(K)
      HTLDNF(K,I)=NPKGPR(K,I)*TDWI(I,K)*WUPP(K)/TANF(K)

```

```

645 CONTINUE
    FFHTLD(K,I)=RMPILU(K)*NPKGPR(K,I)*TDWI(I,K)*WUPP(K)/AREAPP(K)
    IF(IMET(K).EQ.4.OR.IMET(K).GT.7.AND.IMET(K).LT.13)
+   FFHTLD(K,I) = FFHTLD(K,I)/RMPILU(K)*NBSPPA(K)
    IF(FFHTLD(K,I).LE.FFTL) GO TO 685
C   FAR FIELD THERMAL LIMIT EXCEEDED
    NHLPPA = FFTL/(TDWI(I,K)*WUPP(K))
    NPKGPR (K,I) = NHLPPA/(RMPILU (K)/AREAPP(K))
    IF(IMET(K).EQ.4.OR.IMET(K).GT.7.AND.IMET(K).LT.13)
+   NPKGPR(K,I) = NHLPPA/(NBSPPA(K)/AREAPP(K))
    NPPRW (K,I) = NPKGPR (K,I)/NRWSPR(K)
    PITCH (K,I) = (RMLN(K) - 2.*CLSEMP(K))/(NPPRW(K,I)-1.) * PKGPUP(K)
    FFHTLD(K,I) = RMPILU(K)*NPKGPR(K,I)*TDWI(I,K)*WUPP(K)/AREAPP(K)
    IF(IMET(K).EQ.4.OR.IMET(K).GT.7.AND.IMET(K).LT.13)
+   FFHTLD(K,I) = FFHTLD(K,I)/RMPILU(K)*NBSPPA(K)
    IF(IMET(K).LT.5) GOTO 685
    WRITE(IO,684)
684 FORMAT('0--STOP BECAUSE INPUT PITCH IS BEING ADJUSTED'/
+         '      ' WHICH INVALIDATES SECONDARY-TO-PRIMARY'/
+         '      ' INPUT RATIOS IN COCON--')
    STOP
685 CONTINUE
695 CONTINUE
C
C   REPOSITORY LIFE
C
    CALL EMPLAC(K,I)
C
700 CONTINUE
    DO 720 K=1,NWT
    ANPAR(I)=ANPAR(I)+PNLREQ(K,I)
720 CONTINUE
870 CONTINUE
    IF (I.EQ.1) CUMPAR(1) = ANPAR(1)
    IF(I.EQ.1) GO TO 730
    CUMPAR(I)=CUMPAR(I-1)+ANPAR(I)
730 CONTINUE
    CORARA = CORLEN(1,1)*CORWTH(1)/M2ACRE
    BEGPA=(SIZE-CPA-GXEF)
    AVLRPA(I)=BEGPA-CUMPAR(I)
    IF(AVLRPA(I) .LE. 0.) GO TO 910
900 CONTINUE
C   END TIME LOOP
    IF(I.GT.LY) I=LY
C
910 CONTINUE
    RMCAP = RMMAX
    PMCAP = PMMAX
    CMCAP = CMMAX

```

```

C   RECALCULATE LAST YEAR
    I=I
    LY=I
    IF(AVLRPA(I).GE.0.) GO TO 960
    CUMPAR(I)=CUMPAR(I)+AVLRPA(I)
    ANPAR(I)=ANPAR(I)+AVLRPA(I)
    SRMF=0.
    DO 930 K=1,NWT
    SRMF=SRMF+BSFILD(K,I)
930 CONTINUE
    DO 940 K=1,NWT
    PNLREQ(K,I)=BSFILD(K,I)/SRMF*ANPAR(I)
940 CONTINUE
    DO 950 K=1,NWT
    BSFILD(K,I)=PNLREQ(K,I)/AREAPP(K)*RMPILU(K)/RMSPBS(K)
    NUPKG(K,I)=BSFILD(K,I)*NPKGPR(K,I)
    RECUNT(K,I) = NUPKG(K,I) * WUPP(K)
    RMFILD(K,I) = BSFILD(K,I)*RMSPBS(K)
950 CONTINUE
960 CONTINUE
    THEET = 0.0
    SRMFLD = 0.0
    DO 970 K=1,NWT
    DO 970 I=1,LY
    SRMFLD = SRMFLD + RMFILD(K,I)
970 CONTINUE
    DO 980 K=1,NWT
    DO 975 I=1,LY
    WTHEET(K) =WTHEET(K)+NPKGPR(K,I)*RMFILD(K,I)*TDWI(I,K)*WUPP(K)
    SRMWT(K) = SRMWT(K)+RMFILD(K,I)
975 CONTINUE
    THEET=THEET+WTHEET(K)
    TNFTA=TNFTA +TANF(K)
980 CONTINUE

```

```

C
C
C   EQUIPMENT REQUIREMENTS FOR RECEIVING

```

```

C
C   REMOTE
C

```

```

    KD = 2
    NEQLY = 0
    II = 0
    DO 1200 I=1,LY
    DRECNT = SUMRRC(I)
    IF(SRRMR(I).EQ.0.) GO TO 1200
    II = II+1
    CAPCTY = WOEQPC(1) * SRRMR(I) * DPY
    NEQC = SUMRRC(I) /(WOEQPC(1) *SRRMR(I) * DPY)

```

```

1 + .999
  CALL REPLAS(NEQC,NEQLY,KD,NEWEQP,DRECNT,CAPCTY,WOEQPL(1),
6          NW,II)
  IF(SUMRRC(I).EQ.0.0) GO TO 1200
  DO 1220 K=1,NWT
  IF(KIND(K).EQ.2) GOTO 1220
  EQRREC(K,I) = NEWEQP*RECUNT(K,I)/SUMRRC(I)
1220 CONTINUE
1200 CONTINUE
C
  KD = 0
C CONTACT
C
  NEQLY = 0
  II = 0
  DO 1250 I=1,LY
  IF(SRRMC(I).EQ.0.) GO TO 1250
  II = II+1
  CAPCTY = WOEQPC(2) * SRRMC(I) * DPY
  DRECNT = SUMCRC(I)
  NEQC = SUMCRC(I) / (WOEQPC(2) * SRRMC(I) * DPY)
1 + .999
  CALL REPLAS(NEQC,NEQLY,KD,NEWEQP,DRECNT,CAPCTY,WOEQPL(2),
6          NW,II)
  IF(SUMCRC(I).EQ.0) GOTO 1250
  DO 1260 K=1,NWT
  IF(KIND(K).EQ.1.OR.KIND(K).EQ.3) GOTO 1260
  EQRREC(K,I) = NEWEQP*RECUNT(K,I)/SUMCRC(I)
1260 CONTINUE
1250 CONTINUE
C
C EQUIPMENT REQUIREMENTS FOR PACKAGING
C
  II = 0
  NEQLY = 0
  DO 1270 I=1,LY
  IF(SRRMR(I).EQ.0.) GO TO 1270
  II = II + 1
  CAPCTY = WOEQPC(3) * SRRMR(I) * DPY
  DRECNT = SUMRPK(I)
  NEQC = SUMRPK(I) / (WOEQPC(3) * SRRMR(I) * DPY)
1 + .999
  CALL REPLAS(NEQC,NEQLY,KD,NEWEQP,DRECNT,CAPCTY,WOEQPL(3),
6          NW,II)
  IF(SUMRPK(I).EQ.0.0) GOTO 1270
  DO 1275 K=1,NWT
  IF(KIND(K).EQ.2) GOTO 1275
  EQRPKG(K,I) = NEWEQP*NUMPKG(K,I)/SUMRPK(I)
1275 CONTINUE

```

1270 CONTINUE

C
C EQUIPMENT REQUIREMENTS FOR TRANSPORTATION
C

NEQLY = 0
KD = 0
DO 1300 I=1,LY
NEQC = TRANSPT(1,I)+.9999
DUMPKG = SUMRPK(I)
CAPCTY = TOEQPC(1) * SRRMR(I) * DPY
EQLIFE = TOEQPL(1)
CALL REPLAS(NEQC,NEQLY,KD,NEWEQP,DUMPKG, CAPCTY,EQLIFE,
6 NW,I)
IF(SUMRPK(I).EQ.0.0) GO TO 1300
DO 1310 K=1,NWT
IF(KIND(K).EQ.2) GO TO 1310
EQRTN(K,I) = NEWEQP*NUMPKG(K,I)/SUMRPK(I)

1310 CONTINUE

1300 CONTINUE

KD = 0
NEQLY = 0
DO 1325 I=1,LY
NEQC = TRANSPT(2,I)+.9999
DUMPKG = SUMCPK(I)
CAPCTY = TOEQPC(2) * SRRMC(I) * DPY
EQLIFE = TOEQPL(2)
IF(I.EQ.1) NEQLY= 0
CALL REPLAS(NEQC,NEQLY,KD,NEWEQP,DUMPKG, CAPCTY,EQLIFE,
6 NW,I)
IF (SUMCPK(I).EQ.0.0) GOTO 1340
DO 1330 K=1,NWT
IF (KIND(K).EQ.1.OR.KIND(K).EQ.3) GOTO 1330
EQRTN(K,I) = NEWEQP*NUMPKG(K,I)/SUMCPK(I)

1330 CONTINUE

1340 CONTINUE

1325 CONTINUE

C
C EMPLACEMENT MATERIALS COST CALCULATIONS
C
C
C

C UNDERGROUND OPERATIONS
C
C

DO 1660 K=1,NWT
IF(IMET(K).NE.1) GO TO 1660
WABOR(K) = REFDIA(K) / 2 + WABATH(K)
SLEVOR(K) = WABOR(K) + SLEETH(K)
HOLDIA(K) = PKGDIA(K) + (WABATH(K)+SLEETH(K)+BCKFTH(K))

```

1 * 2.
HOLD(K) = PKGLN(K) * NPKPEU(K) + CAPLEN(K) + HOLDEL(K)
1660 CONTINUE
IF(IDET.NE.1) GO TO 1305
DO 1700 K=1,NWT
IF(IEMP(K).NE.1) GOTO 1650
UOMATC(1,K) = HOLMC(WABATH(K),WABOR(K),SLEETH(K),SLEVOR(K),
1BCKFTH(K),HOLDIA(K),REFDIA(K),REFMAT(K),
2 PKGDIA(K),REFSLV(K), REFBKF(K),REFDEP(K),NTYP )
1 * HOLD(K) / PKGPUP(K)
GO TO 1700
1650 CONTINUE
IF(IEMP(K).NE.2) GOTO 1670
UOMATC(2,K) = CHMTEM(K) / PKGPUP(K)
GO TO 1700
1670 CONTINUE
UOMATC(1,K) = RCKCST(K) / PKGPRK(K)
1700 CONTINUE
1305 CONTINUE
DO 1306 I=1,LY
DO 1306 K=1,NWT
IF(KIND(K).EQ.2)GO TO 1307
IF(IUHDCM.EQ.1) EEMATR(K,I) = UOMATU(1,K)
EMMATC(K,I) = UOMATC(1,K) * NEMPU(K,I) * UOMATU(1,K)
GO TO 1306
1307 CONTINUE
IF(IUHDCM.EQ.1) EEMATR(K,I) = UOMATU(2,K)
C EMBLACEMENT MATERIALS COST CALCULATION
EMMATC(K,I) = UOMATC(2,K) * NEMPU(K,I) * UOMATU(2,K)
1306 CONTINUE
C
C EMBLACEMENT EXCAVATION OPERATIONS
C
IICNT = 0
ICNT = 0
DO 1800 K=1,NWT
NEQLY = 0
II = 0
IF(IEMP(K).EQ.1) GO TO 1810
IF(IEMP(K).EQ.2) GO TO 1860
IF(IEMP(K).EQ.3) GO TO 1870
WRITE(IO,3000)
3000 FORMAT('1',T5,'***** ERROR *** NO EMBLACEMENT METHOD SPECIFIED')
STOP
C HOLE DRILLING EXCAVATION REQUIREMENTS CALCULATIONS
1810 CONTINUE
IF(IMET(K).EQ.4) GOTO 1811
IF(IMET(K).GT.7.AND.IMET(K).LT.13) GOTO 1811
GOTO 1812

```

```

1811 IF(PKGLN(K).GT.RMWITH(K)) GOTO 1610
1812 CONTINUE
      IF(IUHDCM.EQ.1) GO TO 1830
      IF(ICNT.EQ.1) GOTO 1800
      DO 1820 I=1,LY
      SUMEPU(I) = 0.0
      TEXTLBR = 0.0
      DO 1815 KK=1,NWT
      IF(IEMP(KK).NE.1) GOTO 1815
      SUMEPU(I) = SUMEPU(I)+NEMPU(KK,I)
      EEMATR(KK,I) = EEMATU(1) * NEMPU(KK,I)
1815 CONTINUE
      DNEMPU = SUMEPU(I)
      IF(SRRMR(I).EQ.0.) GO TO 1820
      II = II+1
      NEQC = SUMEPU(I) / EEEQPC(1) / SRRMR(I) / DPY
      1 + .999
      EXMACH(1,I) = NEQC
      CAPCTY = EEEQPC(1) * SRRMR(I) * DPY
      EQLIFE = EEEQPL(1)
      CALL REPLAS(NEQC,NEQLY,KD,NEWEQP,DNEMPU,CAPCTY,EQLIFE,NW,II)
      DO 1805 KK=1,NWT
      IF (IEMP(KK).NE.1) GOTO 1805
      EQREEX(KK,I) = NEWEQP*NEMPU(KK,I)/SUMEPU(I)
1805 CONTINUE
      DO 1825 L=1,ISR
      EXLBR(I,L) = EELBRR(1) * SFTR(L) * EXMACH(1,I)
      TEXTLBR = TEXTLBR + EXLBR(I,L)
1825 CONTINUE
      DO 1840 KK=1,NWT
      IF (IEMP(KK).NE.1) GOTO 1840
      IF(SUMRPK(I).EQ.0.0) GOTO 1840
      EEXLBR(KK,I) = EEXLBR(KK,I) + TEXTLBR*NEMPU(KK,I)/SUMEPU(I)
      EEXRKM(KK,I) = PI*(HOLDIA(KK)**2)/4.*HOLD(KK)*DENSTY*NEMPU(KK,I)
1840 CONTINUE
      ICNT = 1
1820 CONTINUE
      GO TO 1800
1830 CONTINUE
      HOLXC(K) = HOLDC(HDCON1,HDCON2,HOLDIA(K),NTYP) * HOLD(K)
      DO 1835 I=1,LY
      IF(EEMATC(1).NE.0.0)
      +EEMATR(K,I) = HOLXC(K) * HDMCFC *NEMPU(K,I) / EEMATC(1)
      EXMAC(K) = EXMAC(K) + HOLXC(K) * NEMPU(K,I)
      IF(EELBRC(1).NE.0.0)
      +EEXLBR(K,I) = HOLXC(K) * HDLCFC *NEMPU(K,I) / EELBRC(1)
      DO 1837 KK=1,ISR
      EXLBR(I,KK) = EXLBR(I,KK) + EEXLBR(K,I) /. ISR
1837 CONTINUE

```

```

      EEXRKM(K,I) = (HOLDIA(K)**2) / 4. * HOLD(K) * DENSTY *
1 NEMPU(K,I) * PI
1835 CONTINUE
      GO TO 1800
C NO EMPLACEMENT EXCAVATION IS REQUIRED FOR SIMPLE ROOM PLACEMENT
1860 CONTINUE
      GO TO 1800
C TRENCH EXCAVATION REQUIREMENTS CALCULATIONS
1870 CONTINUE
      IF(IICNT.EQ.1) GOTO 1800
      II = 0
      DO 1875 I=1,LY
      SUMEPU(I) = 0.0
      TEXTLBR = 0.0
      DO 2000 KK = 1,NWT
      IF(IEMP(KK).NE.3) GOTO 2000
      SUMEPU(I) = SUMEPU(I) + NEMPU(KK,I)
      EEMATR(KK,I) = EEMATU(3) * NEMPU(KK,I)
2000 CONTINUE
      IF(SRRMR(I).EQ.0.) GO TO 1875
      II = II+1
      DNEMPU = SUMEPU(I)
      NEQC = SUMEPU(I) / EEEQPC(3) / SRRMR(I) / DPY
1 + .999
      EXMACH(2,I) = NEQC
      CAPCTY = EEEQPC(3) * SRRMR(I) * DPY
      EQLIFE = EEEQPL(3)
      CALL REPLAS(NEQC,NEQLY,KD,NEWEQP,DNEMPU,CAPCTY,EQLIFE,NW,I)
      DO 2010 KK=1,NWT
      IF(IEMP(KK).NE.3) GOTO 2010
      EQREEX(KK,I) = NEWEQP * NEMPU(KK,I)/SUMEPU(I)
2010 CONTINUE
      DO 1873 L=1,ISR
      EXLBR(I,L) = EELBRR(3) * SFTR(L) * EXMACH(2,I)
      TEXTLBR = TEXTLBR + EXLBR(I,L)
1873 CONTINUE
      DO 2030 KK=1,NWT
      IF(SUMEPU(I).EQ.0.0) GOTO 2030
      IF(IEMP(KK).NE.3) GO TO 2030
      EEXLBR(KK,I) = EEXLBR(KK,I) + TEXTLBR * NEMPU(KK,I) /SUMEPU(I)
      EEXRKM(KK,I) = TRCHWD(KK) * (PKGLN(KK)+CAPLEN(KK)+HOLDEL(KK))
1 * PITCH(KK,I) * NEMPU(KK,I) * DENSTY
2030 CONTINUE
1875 CONTINUE
      IICNT = 1
1800 CONTINUE
C
      IF(IUHDCM.NE.1) GOTO 2040
      LLL = LL + 1

```



```

DO 2050 I=LLL,LY
DO 2060 K=1,NWT
IF(EEECST(1).EQ.0) GOTO 2060
EQREEX(K,I) = EXMAC(K) * HDECFC / EEECS(1) / (LY - LL)
EXMACH(1,I) = EXMACH(1,I) + EQREEX(K,I)
2060 CONTINUE
2050 CONTINUE
2040 CONTINUE
C EQUIPMENT REQUIREMENT FOR EMPLACEMENT
DO 1350 I=1,LY
IF(SUMRPK(I).EQ.0.0) GOTO 1350
CAPCTY = UOEQPC(1) * SRRMR(I) * DPY
EQLIFE = UOEQPL(1)
DUMPKG = SUMRPK(I)
NEQC = EMPMCH(1,I)
IF(I.EQ.1) NEQLY = 0
CALL REPLAS(NEQC,NEQLY,KD,NEWEQP,DUMPKG, CAPCTY,EQLIFE,NW,I)
DO 1360 K=1,NWT
IF(KIND(K).EQ.2) GO TO 1360
EQREMP(K,I) = NEWEQP*NUMPKG(K,I)/SUMRPK(I)
1360 CONTINUE
1350 CONTINUE
DO 1370 I=1,LY
IF(SUMCPK(I).EQ.0.0) GOTO 1370
CAPCTY = UOEQPC(2)*SRRMC(I)*DPY
EQLIFE = UOEQPL(2)
DUMPKG = SUMCPK(I)
NEQC = EMPMCH(2,I)
IF(I.EQ.1) NEQLY = 0
CALL REPLAS(NEQC,NEQLY,KD,NEWEQP,DUMPKG, CAPCTY,EQLIFE,NW,I)
DO 1380 K=1,NWT
IF(KIND(K).EQ.1.OR.KIND(K).EQ.3) GOTO 1380
EQREMP(K,I) = NEWEQP*NUMPKG(K,I)/SUMCPK(I)
1380 CONTINUE
1370 CONTINUE
C
C LABOR COST CALCULATIONS
C
DO 758 I=1,LY
ARLBCT(I) = 0.
APLBCT(I) = 0.
ATLBCT(I) = 0.
AMLBCT(I) = 0.
REEQCT(I) = 0.
PARQCT(I) = 0.
TREQCT(I) = 0.
758 CONTINUE
DO 1500 I=1,LY
SUMEPU(I) = 0.0

```

```

DO 1505 K=1,NWT
IF (KIND(K).EQ.2) SUMEPU(I) = SUMEPU(I) + NEMPU(K,I)
1505 CONTINUE
DO 760 K=1,NWT
IS = SRRMR(I)+.999
IF (KIND(K).EQ.2) IS = SRRMC(I)+.999
DO 760 L=1,IS
IF(KIND(K).EQ.2) GO TO 765
IF(SUMRRC(I).EQ.0.) GO TO 1910
RELBCT(K,I) = RHRLBR(I,L) * WOLBRC(1) * SHCFAC(L) * OVRHDF *
1 RECUNT(K,I) / SUMRRC(I) + RELBCT(K,I)
1910 CONTINUE
IF(SUMRPK(I).EQ.0.) GO TO 1920
PALBCT(K,I) = RHPLBR(I,L) * WOLBRC(3) * SHCFAC(L) * OVRHDF *
1 Numpkg(K,I) / SUMRPK(I) + PALBCT(K,I)
TRLBCT(K,I) = TRRLBR(I,L) * TOLBRC(1) * SHCFAC(L) * OVRHDF *
1 Numpkg(K,I) / SUMRPK(I) + TRLBCT(K,I)
EMLBCT(K,I) = EMLBCT(K,I)+ERRLBR(I,L)*UOLBRC(1)*SHCFAC(L) *
1 OVRHDF * Numpkg(K,I) / SUMRPK(I)
EXLBCT(K,I) = EXLBCT(K,I) + EXLBR(I,L) * EELBRC(1) * SHCFAC(L) *
1 OVRHDF * Numpkg(K,I) / SUMRPK(I)
1920 CONTINUE
GO TO 770
765 CONTINUE
IF(SUMCPK(I).EQ.0.) GO TO 770
RELBCT(K,I) = CHRLBR(I,L) * WOLBRC(2) * SHCFAC(L) * OVRHDF *
1 Numpkg(K,I) / SUMCPK(I) + RELBCT(K,I)
TRLBCT(K,I) = TRCLBR(I,L) * TOLBRC(2) * SHCFAC(L) * OVRHDF *
1 Numpkg(K,I) / SUMCPK(I) + TRLBCT(K,I)
IF(SUMEPU(I).EQ.0.0) GOTO 770
EMLBCT(K,I) = EMLBCT(K,I)+ERCLBR(I,L)*UOLBRC(2)*SHCFAC(L) *
1 OVRHDF * NEMPU(K,I) / SUMEPU(I)
770 CONTINUE
760 CONTINUE
DO 780 K=1,NWT
ARLBCT(I) = ARLBCT(I) + RELBCT(K,I)
APLBCT(I) = APLBCT(I) + PALBCT(K,I)
ATLBCT(I) = ATLBCT(I) + TRLBCT(K,I)
AMLBCT(I) = AMLBCT(I) + EMLBCT(K,I)
AXLBCT(I) = AXLBCT(I) + EXLBCT(K,I)
780 CONTINUE
C
C MATERIALS COST CALCULATIONS
C
DO 1110 K=1,NWT
C FOR RECEIVING
IF (KIND(K).NE.2) RCMATC(K,I) = RCMATR(K,I) * WOMATC(1,K)
IF (KIND(K).EQ.2) RCMATC(K,I) = RCMATR(K,I) * WOMATC(2,K)
C FOR PACKAGING

```

```

      PKMATC(K,I) = PKMATR(K,I) * WOMATC(3,K)
C   FOR TRANSPORT
      IF (KIND(K).NE.2) TRMATC(K,I) = TRMATR(K,I) * TOMATC(1,K)
      IF (KIND(K).EQ.2) TRMATC(K,I) = TRMATR(K,I) * TOMATC(2,K)
C   FOR EMPLACEMENT EXCAVATIONS
      EXMATC(K,I) = EEMATR(K,I) * EEMATC(IEMP(K))
C   MISCELLANEOUS MATERIALS COSTS
      REMTCT(I) = REMTCT(I) + RCMATC(K,I)
      PAMTCT(I) = PAMTCT(I) + PKMATC(K,I)
      TRMTCT(I) = TRMTCT(I) + TRMATC(K,I)
      EMMTCT(I) = EMMTCT(I) + EMMATC(K,I)
      EXMTCT(I) = EXMTCT(I) + EXMATC(K,I)
1110  CONTINUE
      TMTCT(I) = REMTCT(I) + PAMTCT(I) + TRMTCT(I) + EMMTCT(I) + EXMTCT(I)
C
C   EQUIPMENT COSTS
C
      DO 1400 K=1,NWT
      IF(KIND(K).EQ.2) GO TO 1410
      RCEQPC(K,I) = EQRREC(K,I) * WOECST(1)
      GO TO 1400
1410  CONTINUE
      RCEQPC(K,I) = EQRREC(K,I) * WOECST(2)
1400  CONTINUE
      DO 1420 K=1,NWT
      PKEQPC(K,I) = EQRPKG(K,I) * WOECST(3)
1420  CONTINUE
      DO 1430 K=1,NWT
      IF(KIND(K).EQ.2) GO TO 1440
      TREQPC(K,I) = EQRTN(K,I) * TOECST(1)
      EMEQPC(K,I) = EQREMP(K,I) * UOECST(1)
      GO TO 1430
1440  CONTINUE
      TREQPC(K,I) = EQRTN(K,I) * TOECST(2)
      EMEQPC(K,I) = EQREMP(K,I) * UOECST(2)
1430  CONTINUE
      DO 1445 K=1,NWT
      EXEQPC(K,I) = EQREEX(K,I) * EEEEST(IEMP(K))
1445  CONTINUE
      DO 1450 K=1,NWT
      REEQCT(I) = REEQCT(I) + RCEQPC(K,I)
      PARQCT(I) = PARQCT(I) + PKEQPC(K,I)
      TREQCT(I) = TREQCT(I) + TREQPC(K,I)
      EMEQCT(I) = EMEQCT(I) + EMEQPC(K,I)
      EXEQCT(I) = EXEQCT(I) + EXEQPC(K,I)
1450  CONTINUE
1500  CONTINUE
C
C

```

C

```
      DO 1050 K=1,NWT
      DO 1050 I=1,LY
      WSTVOL(K,I) = WVOLPU(K)*RECUNT(K,I)
1050  CONTINUE
      RETURN
1600  CONTINUE
      WRITE(IO,1111)
1111  FORMAT('1',T25,'STOP - STOP - STOP',/////,T20,'NO WASTE SHAFTS SPE
1CIFIED')
      STOP
1610  CONTINUE
      WRITE(IO,1620)
1620  FORMAT('0  -- STOP  -- STOP --'////
1      ' PACKAGE LENGTH IS LONGER THAN SPECIFIED ROOM WIDTH')
      STOP
1630  CONTINUE
      WRITE(IO,1640)
1640  FORMAT(' ',//,'STOP      STOP      STOP'/
& ' CASE TERMINATED DUE TO ERROR IN CALCULATING SHAFT SHIFT '
& ' REQUIREMENTS',/' CHECK INPUT VALUES FOR WUPP(K)',
& ', DPY, NPKPTR(K), MTRPSD, OR, NPKPTC(K), MTCPSD')
      STOP
      DEBUG SUBCHK
      AT 1
      END
```

PARAM PROC

C

REAL NFTL
REAL MSA
REAL MROP
INTEGER SFTNUM

C

PARAMETER NY = 60
PARAMETER IO = 6
PARAMETER IOUT = 7
PARAMETER IN = 5
PARAMETER IB = 9
PARAMETER NW = 10
PARAMETER NWP1 = NW +1
PARAMETER NC = 10
PARAMETER NCR = 6
PARAMETER NS = 3
PARAMETER NST = 7
PARAMETER NDC = 2
PARAMETER NFAC = 50
PARAMETER NRO = 5
PARAMETER NTO = 2
PARAMETER NUO = 2
PARAMETER NWO = 3
PARAMETER NEE = 3
PARAMETER NEO = 5
PARAMETER NYRO = NY+40
PARAMETER NEQP = 24

C

END

C

C

C

OPTION PROC

COMMON/OPTION/ IOPTA
COMMON/OPTION/ IOPTB(10,10)
COMMON/OPTION/ IOPTC(10)
COMMON/OPTION/ INUM(NW)
COMMON/OPTION/ INWT
COMMON/OPTION/ IWNUM(NW)

END

C

C

C

CHARACTER VARIABLES

INPUT PROC

CHARACTER*80 ICARD, DATYPE, TITLE
CHARACTER*48 CORDSC, SFTITL, DDTITL
CHARACTER*16 RPTYP, PKGDS1(NW), PKGDS2(NW), PKGDS3(NW)
CHARACTER*16 TEKNEK

CHARACTER*16 PLACE
 CHARACTER*12 WSTYP
 INTEGER TABLES
 COMMON/INPUT/ ICARD
 COMMON/INPUT/ TITLE
 COMMON/INPUT/ DATYPE(12)
 COMMON/INPUT/ CORDSC(NCR)
 COMMON/INPUT/ SFTITL(NFAC)
 COMMON/INPUT/ DDTITL(NDC)
 COMMON/INPUT/ RPTYP
 COMMON/INPUT/ PKGDS1
 COMMON/INPUT/ PKGDS2
 COMMON/INPUT/ PKGDS3
 COMMON/INPUT/ PLACE
 COMMON/INPUT/ TEKNEK
 COMMON/INPUT/ TABLES (27)
 COMMON/INPUT/ WSTYP(NW)

C

COMMON/INPUT/ DENSTY
 COMMON/INPUT/ ACP
 COMMON/INPUT/ AECON1
 COMMON/INPUT/ AECON2
 COMMON/INPUT/ BCKFTH(NW)
 COMMON/INPUT/ BKFLC
 COMMON/INPUT/ BKRATE
 COMMON/INPUT/ BLANK
 COMMON/INPUT/ CAPLEN(NW)
 COMMON/INPUT/ CCON1
 COMMON/INPUT/ CCON2
 COMMON/INPUT/ CFSPCT
 COMMON/INPUT/ CHMTEM(NW)
 COMMON/INPUT/ CHRECW(NW)
 COMMON/INPUT/ CLCDF(NC)
 COMMON/INPUT/ CLNDF(NC)
 COMMON/INPUT/ CLSEMP(NW)
 COMMON/INPUT/ COCON(NW,NW)
 COMMON/INPUT/ CONFAC(4)
 COMMON/INPUT/ CONTRU(NW)
 COMMON/INPUT/ CORHT(NCR)
 COMMON/INPUT/ CORLEN(NCR,NW)
 COMMON/INPUT/ CORWTH(NCR)
 COMMON/INPUT/ CPA
 COMMON/INPUT/ CRRAIL(NW,NY)
 COMMON/INPUT/ DBDF(NC)
 COMMON/INPUT/ DCOMCF
 COMMON/INPUT/ DDCST (NDC)
 COMMON/INPUT/ DDEQPC(NDC)
 COMMON/INPUT/ DDFRAC
 COMMON/INPUT/ DDLBRC(NDC)

COMMON/INPUT/	DDLBRN(NDC)
COMMON/INPUT/	DDMATC(NDC)
COMMON/INPUT/	DELRHT(NW)
COMMON/INPUT/	DEPTH
COMMON/INPUT/	DRATE
COMMON/INPUT/	DSL
COMMON/INPUT/	DWTRC
COMMON/INPUT/	EECST (NEO)
COMMON/INPUT/	EEECST(NEO)
COMMON/INPUT/	EEEQPC(NEO)
COMMON/INPUT/	EEEQPL(NEO)
COMMON/INPUT/	EELBRC(NEO)
COMMON/INPUT/	EELBRN(NEO)
COMMON/INPUT/	EEMATC(NEO)
COMMON/INPUT/	EEMATU(NEO)
COMMON/INPUT/	FDDF(NC)
COMMON/INPUT/	FFTL
COMMON/INPUT/	GXF
COMMON/INPUT/	HDCON1
COMMON/INPUT/	HDCON2
COMMON/INPUT/	HDECFC
COMMON/INPUT/	HDLCFC
COMMON/INPUT/	HDMCFC
COMMON/INPUT/	HEATS
COMMON/INPUT/	HOLDEL(NW)
COMMON/INPUT/	HPMIN(NW)
COMMON/INPUT/	ICLR
COMMON/INPUT/	ID
COMMON/INPUT/	IDDYR
COMMON/INPUT/	IDET
COMMON/INPUT/	IEQTP1
COMMON/INPUT/	IEQTP2
COMMON/INPUT/	IEQTP3
COMMON/INPUT/	IEQTP4
COMMON/INPUT/	IEQTP5
COMMON/INPUT/	IMET(NW)
COMMON/INPUT/	ITSURV
COMMON/INPUT/	IUHDCM
COMMON/INPUT/	IYDOL
COMMON/INPUT/	IYRCON
COMMON/INPUT/	IYRDIS
COMMON/INPUT/	IYRPRO
COMMON/INPUT/	KIND(NW)
COMMON/INPUT/	LASTYR
COMMON/INPUT/	LEAD
COMMON/INPUT/	LY
COMMON/INPUT/	LYY
COMMON/INPUT/	MPRIME
COMMON/INPUT/	MROP

COMMON/INPUT/	MSA
COMMON/INPUT/	MSHIFT(2)
COMMON/INPUT/	MSWTCH
COMMON/INPUT/	MTCPSD
COMMON/INPUT/	MTRPSD
COMMON/INPUT/	MXMMRH
COMMON/INPUT/	MXMPRH
COMMON/INPUT/	NALSH
COMMON/INPUT/	NCBPRM
COMMON/INPUT/	NRBPRM
COMMON/INPUT/	NBSPPA(NW)
COMMON/INPUT/	NBYR
COMMON/INPUT/	NCHRNA
COMMON/INPUT/	NCOR
COMMON/INPUT/	NCORPP(NCR)
COMMON/INPUT/	NDDC
COMMON/INPUT/	NFTL (NW)
COMMON/INPUT/	NNHG
COMMON/INPUT/	NPKPEU(NW)
COMMON/INPUT/	NPKPTC(NW)
COMMON/INPUT/	NPKPTR(NW)
COMMON/INPUT/	NRHRMA
COMMON/INPUT/	NRWSPR(NW)
COMMON/INPUT/	NSC
COMMON/INPUT/	NTYP
COMMON/INPUT/	NWT
COMMON/INPUT/	OFTADC
COMMON/INPUT/	ORS
COMMON/INPUT/	OTBDEN
COMMON/INPUT/	OTHBFL
COMMON/INPUT/	OVHRDF
COMMON/INPUT/	OWHP
COMMON/INPUT/	PANLEN(NW)
COMMON/INPUT/	PANWTH(NW)
COMMON/INPUT/	PCTCON(NW,NW)
COMMON/INPUT/	PCTFIL
COMMON/INPUT/	PKGCAP(NW)
COMMON/INPUT/	PKGDI(NW)
COMMON/INPUT/	PKGLN(NW)
COMMON/INPUT/	PKGPRK(NW)
COMMON/INPUT/	PKGUP(NW)
COMMON/INPUT/	PORMDF(2,NC)
COMMON/INPUT/	RAPP
COMMON/INPUT/	RAPS
COMMON/INPUT/	RCCAP(NW)
COMMON/INPUT/	RCKCST(NW)
COMMON/INPUT/	RCKHC
COMMON/INPUT/	RED
COMMON/INPUT/	REFBKF(NW)

COMMON/INPUT/	REFDEP(NW)
COMMON/INPUT/	REFDIA(NW)
COMMON/INPUT/	REFMAT(NW)
COMMON/INPUT/	REFSLV(NW)
COMMON/INPUT/	RMCWS1(NC)
COMMON/INPUT/	RMCWS2(NC)
COMMON/INPUT/	RMHMIN(NW)
COMMON/INPUT/	RMLN(NW)
COMMON/INPUT/	RMNWS1(NC)
COMMON/INPUT/	RMNWS2(NC)
COMMON/INPUT/	RMWTH(NW)
COMMON/INPUT/	ROCST (NRO)
COMMON/INPUT/	ROECST(NRO)
COMMON/INPUT/	ROEQPC(NRO)
COMMON/INPUT/	ROEQPL(NRO)
COMMON/INPUT/	ROLBRC(NRO)
COMMON/INPUT/	ROLBRR(NRO)
COMMON/INPUT/	ROMATC(NRO)
COMMON/INPUT/	ROMATU(NRO)
COMMON/INPUT/	RVP
COMMON/INPUT/	SCCON1
COMMON/INPUT/	SCCON2
COMMON/INPUT/	SCCST
COMMON/INPUT/	SCDF(NC)
COMMON/INPUT/	SCEQPC
COMMON/INPUT/	SCLBRC
COMMON/INPUT/	SCLBRR
COMMON/INPUT/	SCMATC
COMMON/INPUT/	SCMATU
COMMON/INPUT/	SCSKD(NC)
COMMON/INPUT/	SFCS(NC)
COMMON/INPUT/	SFCST(NFAC)
COMMON/INPUT/	SFECST(NFAC)
COMMON/INPUT/	SFEQPL(NFAC)
COMMON/INPUT/	SFLBRC(NFAC)
COMMON/INPUT/	SFLBRR(NFAC)
COMMON/INPUT/	SFMATC(NFAC)
COMMON/INPUT/	SFMATU(NFAC)
COMMON/INPUT/	SFP
COMMON/INPUT/	SFSP
COMMON/INPUT/	SFTDD(NST)
COMMON/INPUT/	SFTHE(NST)
COMMON/INPUT/	SFTLD(NST)
COMMON/INPUT/	SFTNUM(NST)
COMMON/INPUT/	SFTXD(NST)
COMMON/INPUT/	SFTSS(NST)
COMMON/INPUT/	SFTXR(NST)
COMMON/INPUT/	SHCFAC(NS)
COMMON/INPUT/	SIZE

COMMON/INPUT/ SLBR(NFAC)
COMMON/INPUT/ SLEETH(NW)
COMMON/INPUT/ SOA
COMMON/INPUT/ SPUC
COMMON/INPUT/ SRKHST(NYRO)
COMMON/INPUT/ SSCON1
COMMON/INPUT/ SSCON2
COMMON/INPUT/ SSDF(NC)
COMMON/INPUT/ STMC
COMMON/INPUT/ STOMAX(NW)
COMMON/INPUT/ STPCT
COMMON/INPUT/ SUD
COMMON/INPUT/ SURCST
COMMON/INPUT/ SURFDC
COMMON/INPUT/ SWPPR(NW)
COMMON/INPUT/ TCCAP(NW)
COMMON/INPUT/ TCLAND
COMMON/INPUT/ TCLICE
COMMON/INPUT/ TCPCLN
COMMON/INPUT/ TDWI(NY,NW)
COMMON/INPUT/ TNPT1
COMMON/INPUT/ TNPT2
COMMON/INPUT/ TOCST(NT0)
COMMON/INPUT/ TOECST(NT0)
COMMON/INPUT/ TOEQPC(NT0)
COMMON/INPUT/ TOEQPL(NT0)
COMMON/INPUT/ TOLBRC(NT0)
COMMON/INPUT/ TOLBRR(NT0)
COMMON/INPUT/ TOMATC(NT0,NW)
COMMON/INPUT/ TOMATU(NT0,NW)
COMMON/INPUT/ TONPAR
COMMON/INPUT/ TRCHWD(NW)
COMMON/INPUT/ TTTK(NW)
COMMON/INPUT/ TTTR(NW)
COMMON/INPUT/ UGD
COMMON/INPUT/ UNTMAS(NW)
COMMON/INPUT/ UOCST (NUO)
COMMON/INPUT/ UOECST(NUO)
COMMON/INPUT/ UOEQPC(NUO)
COMMON/INPUT/ UOEQPL(NUO)
COMMON/INPUT/ UOLBRC(NUO)
COMMON/INPUT/ UOLBRR(NUO)
COMMON/INPUT/ UOMATC(NUO,NW)
COMMON/INPUT/ UOMATU(NUO,NW)
COMMON/INPUT/ UPC
COMMON/INPUT/ VARMC
COMMON/INPUT/ VOLDW
COMMON/INPUT/ VOLW
COMMON/INPUT/ WABATH(NW)

```

COMMON/INPUT/ WCCON1
COMMON/INPUT/ WCCON2
COMMON/INPUT/ WOCST(NWO)
COMMON/INPUT/ WOECST(NWO)
COMMON/INPUT/ WOEQPC(NWO)
COMMON/INPUT/ WOEQPL(NWO)
COMMON/INPUT/ WOLBRC(NWO)
COMMON/INPUT/ WOLBRR(NWO)
COMMON/INPUT/ WOMATC(NWO,NW)
COMMON/INPUT/ WOMATU(NWO,NW)
COMMON/INPUT/ WRKYR
COMMON/INPUT/ WSTSUP(NW,NY)
COMMON/INPUT/ WTRC
COMMON/INPUT/ WUPP(NW)
COMMON/INPUT/ WVOLPU(NW)
COMMON/INPUT/ XPANFC
COMMON/INPUT/ XRASHO(NW)
END
C
  INCLUDE PARAM
  INCLUDE OPTION
  DIMENSION ISEC(10)
  CHARACTER*1 IANS
C
  ICNT=0
  IOPT=0
  IOPTA=0
C
  100 CONTINUE
C
C  ZERO WASTE NUMBER ARRAY
  DO 80 I=1,NW
    IWNUM(I)=0
80  CONTINUE
C
C  ZERO OPTION MATRIX
  DO 110 I=1,10
    DO 110 J=1,10
      IOPTB(I,J)=0
      IOPTC(I)=0
110 CONTINUE
C
C  MAIN MENU
C
115  WRITE (10,120)
120  FORMAT ('0  OPTION'
+         /'
+         /'0  1. DEVELOP DATA FOR NEW CASE'
+         /'0  2. UPDATE EXISTING CASE'
+         /'0  3. EXIT FROM PREPROCESSOR'

```

```

+      /'0          ENTER THE DESIRED OPTION 1 - 3')
130  READ (IN,180,ERR=115) IOPT
180  FORMAT (I1)
      GO TO (1170,190,1360)IOPT
      WRITE(IO,185)
185  FORMAT(' YOUR RESPONSE IS OUT OF RANGE..... TRY AGAIN.')
      GO TO 130
190  WRITE (IO,200)
200  FORMAT ('0      OPTION'/
+      /'          1.  UPDATE ALL ITEMS'
+      /'0        2.  UPDATE SPECIFIC ITEMS ONLY'
+      /'0          ENTER DESIRED OPTION, 1 OR 2')
220  READ (IN,180,ERR=190) IOPTA
      GO TO (1170,230) IOPTA
      WRITE(IO,185)
      GO TO 220
C
230  CONTINUE
      INWT=NW
      ISECF=0
C
C      ZERO OUT SPECIFIC AREA OPTIONS
      DO 240 I=1,10
          DO 240 J=1,10
              IOPTB(I,J)=0
              IOPTC(I)=0
240  CONTINUE
C
242  WRITE(IO,245)
245  FORMAT ('0          0.  EXIT TO MAIN MENU')
      WRITE (IO,250)
250  FORMAT ('0          1.  GENERAL SYSTEMS DATA')
      WRITE (IO,260)
260  FORMAT ('0          2.  WASTE HANDLING SYSTEMS')
      WRITE (IO,270)
270  FORMAT ('0          3.  REPOSITORY SYSTEMS')
      WRITE (IO,280)
280  FORMAT ('0          4.  SHAFT DESCRIPTIONS AND COSTS')
      WRITE (IO,290)
290  FORMAT ('0          5.  ROCK HANDLING - DISPOSAL DESCRIPTION',
+ ' & COSTS')
      WRITE (IO,300)
300  FORMAT ('0          6.  SUPPORT FACILITIES')
      WRITE (IO,310)
310  FORMAT ('0          7.  UTILITY PARAMETERS AND COSTS')
      WRITE (IO,320)
320  FORMAT ('0          8.  PRE-OPERATIONAL SUPPORT')
      WRITE (IO,330)
330  FORMAT ('0          9.  DECOMMISSIONING AND SURVEILANCE')

```

```

        WRITE (IO,340)
340  FORMAT ('0                      ENTER DESIRED CATEGORY OPTION, 0 - 9')
350  CONTINUE
        READ (IN,180,ERR=242) INDEX
        IF (INDEX.EQ.0) GO TO 100
        IF (INDEX.GT.0.AND.INDEX.LE.9) GO TO 370
        WRITE (IO,360)
360  FORMAT ('0  ERROR IN CATEGORY NUMBER'
+          '/' RE-ENTER CORRECT OPTION')
        GO TO 350
370  CONTINUE
        IOPTC(INDEX)=1
C      GO TO (380,560,690,810,980,1130,1140,1150,1160) INDEX
C
C      1.  GENERAL SYSTEMS
C      -----
C
380  CONTINUE
        WRITE (IO,250)
        WRITE (IO,385)
385  FORMAT(17X,20('='))
        WRITE (IO,390)
390  FORMAT ('0  OPTION      SUB-CATEGORY')
        WRITE (IO,400)
400  FORMAT ('0      0.  EXIT TO CATEGORY MENU')
        WRITE (IO,410)
410  FORMAT ('0      1.  CASE TITLE'
+          '/'0      2.  SITE PARAMETERS'
+          '/'0      3.  CONSTRAINTS'
+          '/'0      4.  REPOSITORY SCHEDULE AND WASTE '
+'GENERATION DATA'
+          '/'0      5.  CONSTRUCTION AND COST EXPENDITURE ',
+'SCHEDULES'
+          '/'0      6.  MISCELLANEOUS COST PARAMETERS'
+          '/'0      7.  ALL THE ABOVE SECTIONS'
+          '/'0      ENTER THE DESIRED SECTION OPTION, 0 - 7')
480  CONTINUE
        ISFLG=0
        READ (IN,490,ERR=380) ISEC
490  FORMAT (10(I1,1X))
        DO 500 I=1,10
            IF (ISEC(I).GE.0.AND.ISEC(I).LE.7) GO TO 500
        ISFLG=1
500  CONTINUE
        IF (ISFLG.GT.0) WRITE (IO,510)
510  FORMAT ('0  ERROR IN SUB-CATEGORY NUMBER'
+          '/' RE-ENTER CORRECT VALUE')
        IF (ISFLG.GT.0) GO TO 480

```

```

      IF (ISEC(1).EQ.0) GO TO 1387
      IF (ISEC(1).EQ.7) GO TO 530
      DO 520 J=1,6
        IF (ISEC(J).EQ.0) GO TO 520
        K=ISEC(J)
        IOPTB(K,INDEX)=1
      ISECF=1
520  CONTINUE
      GO TO 550
530  CONTINUE
      DO 540 J=1,6
        IOPTB(J,INDEX)=1
540  CONTINUE
550  CONTINUE
      IF (IOPTB(4,INDEX).EQ.1.OR.IOPTB(3,INDEX).EQ.1) IFLG=1
      GO TO 1190

```

C
C 2. WASTE HANDLING SYSTEMS
C -----
C

```

560  CONTINUE
      WRITE (IO,260)
      WRITE(IO,565)
565  FORMAT(17X,22('='))
      WRITE (IO,390)
      WRITE (IO,400)
      WRITE (IO,570)
570  FORMAT ('0      1. GENERAL WASTE DESCRIPTION'
+          /'0      2. WASTE RECEIVING PARAMETERS AND '
+ 'COST DATA'
+          /'0      3. WASTE PACKAGING PARAMETERS AND '
+ 'DATA'
+          /'0      4. WASTE TRANSFER PARAMETERS AND COSTS'
+          /'0      5. WASTE EMPLACEMENT PARAMETERS AND '
+ 'COSTS'
+          /'0      6. ALL THE ABOVE SECTIONS'
+ /'0      ENTER THE DESIRED SECTION OPTION, 0 - 6')
630  CONTINUE
      ISFLG=0
      READ (IN,490,ERR=560) ISEC
      DO 640 I=1,10
        IF (ISEC(I).GE.0.AND.ISEC(I).LE.6) GO TO 640
        ISFLG=1
640  CONTINUE
      IF (ISFLG.GT.0) WRITE (IO,510)
      IF (ISFLG.GT.0) GO TO 630
      IF (ISEC(1).EQ.0) GO TO 1387
      IF (ISEC(1).EQ.6) GO TO 660
      DO 650 J=1,5

```

```

        IF (ISEC(J).EQ.0) GO TO 650
        K=ISEC(J)
        IOPTB(K,INDEX)=1
        ISECF=1
650  CONTINUE
        GO TO 680
660  CONTINUE
        DO 670 J=1,5
            IOPTB(J,INDEX)=1
670  CONTINUE
680  CONTINUE
        IFLG=1
        GO TO 1190

```

```

C
C  3. REPOSITORY SYSTEMS
C  -----
C

```

```

690  CONTINUE
        WRITE (IO,270)
        WRITE (IO,695)
695  FORMAT(17X,18('='))
        WRITE (IO,390)
        WRITE (IO,400)
        WRITE (IO,700)
700  FORMAT ('0      1. PANEL PARAMETERS'
+          /'0      2. ROOM PARAMETERS'
+          /'0      3. CORRIDOR PARAMETERS'
+          /'0      4. ALL THE ABOVE SECTIONS'
+          /'0      ENTER THE DESIRED SECTION OPTION, 0 - 4')
        IFLG = 1
750  CONTINUE
        ISFLG=0
        READ (IN,490,ERR=690) ISEC
        DO 760 I=1,10
            IF (ISEC(I).GE.0.AND.ISEC(I).LE.4) GO TO 760
            ISFLG=1
760  CONTINUE
        IF (ISFLG.GT.0) WRITE (IO,510)
        IF (ISFLG.GT.0) GO TO 750
        IF (ISEC(1).EQ.0) GO TO 1387
        IF (ISEC(1).EQ.4) GO TO 780
        DO 770 J=1,3
            IF (ISEC(J).EQ.0) GO TO 770
            K=ISEC(J)
            IOPTB(K,INDEX)=1
            ISECF=1
770  CONTINUE
        GO TO 800
780  CONTINUE

```

```

      DO 790 J=1,3
        IOPTB(J,INDEX)=1
790 CONTINUE
800 CONTINUE
      GO TO 1190
C
C 4. SHAFT DESCRIPTION
C -----
C
810 CONTINUE
      WRITE (IO,280)
      WRITE(IO,815)
815 FORMAT(17X,28('='))
      WRITE (IO,390)
      WRITE (IO,400)
      WRITE (IO,820)
820 FORMAT ('0      1. MEN & MATERIAL SHAFT'
+         /'0      2. REMOTE-HANDLED WASTE'
+ ' SHAFT'
+         /'0      3. CONTACT-HANDLED WASTE'
+ ' SHAFT'
+         /'0      4. MINE VENTILATION SHAFT'
+         /'0      5. MINING EXHAUST SHAFT'
+         /'0      6. WASTE EXHAUST SHAFT'
+         /'0      7. ROCK HOISTING SHAFT'
+         /'0      8. SHAFT COST FUNCTIONS '
+         /'0      9. ALL THE ABOVE SECTIONS'
+ ' 0 - 9')
      ENTER THE DESIRED SECTION OPTION,'
920 CONTINUE
      ISFLG=0
      READ (IN,490,ERR=810) ISEC
      DO 930 I=1,10
        IF (ISEC(I).GE.0.AND.ISEC(I).LE.9) GO TO 930
        ISFLG=1
930 CONTINUE
      IF (ISFLG.GT.0) WRITE (IO,510)
      IF (ISFLG.GT.0) GO TO 920
      IF (ISEC(1).EQ.0) GO TO 1387
      IF (ISEC(1).EQ.9) GO TO 950
      DO 940 J=1,8
        IF (ISEC(J).EQ.0) GO TO 940
        K=ISEC(J)
        IOPTB(K,INDEX)=1
        ISECF=1
940 CONTINUE
      GO TO 970
950 CONTINUE
      DO 960 J=1,8

```



```

      IOPTB(J,INDEX)=1
960 CONTINUE
970 CONTINUE
      GO TO 1190
C
C   5.  ROCK HANDLING SYSTEMS
C   -----
C
980 CONTINUE
      WRITE (IO,290)
      WRITE(IO,985)
985 FORMAT(17X,44('='))
      WRITE (IO,390)
      WRITE (IO,400)
      WRITE (IO,990)
990 FORMAT ('0      1.  GENERAL INFORMATION'
+         /'0      2.  ROCK EXCAVATION'
+         /'0      3.  ROCK HOISTING'
+         /'0      4.  SURFACE STORAGE'
+         /'0      5.  OFFSITE DISPOSAL'
+         /'0      6.  BACKFILLING'
+         /'0      7.  ALL THE ABOVE SECTIONS'
+         /'0      ENTER THE DESIRED SECTION '
+ 'OPTION, 0 - 7')
1070 CONTINUE
      ISFLG=0
      READ (IN,490,ERR=980) ISEC
      DO 1080 I=1,10
        IF (ISEC(I).GE.0.AND.ISEC(I).LE.7) GO TO 1080
        ISFLG=1
1080 CONTINUE
      IF (ISFLG.GT.0) WRITE (IO,510)
      IF (ISFLG.GT.0) GO TO 1070
      IF (ISEC(1).EQ.0) GO TO 1387
      IF (ISEC(1).EQ.7) GO TO 1100
      DO 1090 J=1,6
        IF (ISEC(J).EQ.0) GO TO 1090
        K=ISEC(J)
        IOPTB(K,INDEX)=1
        ISECF=1
1090 CONTINUE
      GO TO 1120
1100 CONTINUE
      DO 1110 J=1,6
        IOPTB(J,INDEX)=1
1110 CONTINUE
1120 CONTINUE
      GO TO 1190
C

```

C 6. SUPPORT FACILITIES

C

C

1130 CONTINUE
IOPTB(1,INDEX)=1
GO TO 1190

C

C 7. UTILITIES

C

C

1140 CONTINUE
WRITE(IO,310)
WRITE(IO,815)
WRITE(IO,390)
WRITE(IO,400)
WRITE(IO,3000)

3000 FORMAT('0 1. ELECTRICAL '
+ /'0 2. STEAM '
+ /'0 3. POTABLE WATER '
+ /'0 4. DEMINERALIZED WATER '
+ /'0 5. DIESEL FUEL '
+ /'0 6. ALL THE ABOVE SECTIONS '
+ /'0 ENTER THE DESIRED SECTION OPTION, 0 - 6')

3070 CONTINUE
ISFLG = 0
READ(IN,490,ERR=1140) ISEC
DO 3080 I=1,10
IF(ISEC(I).GE.0.AND.ISEC(I).LE.6) GO TO 3080
ISFLG = 1

3080 CONTINUE
IF(ISFLG.GT.0) WRITE(IO,510)
IF(ISFLG.GT.0) GOTO 3070
IF(ISEC(1).EQ.0) GOTO 1387
IF(ISEC(1).EQ.6) GOTO 3090
DO 3100 J=1,5
IF(ISEC(J).EQ.0) GOTO 3100
K=ISEC(J)
IOPTB(K,INDEX) = 1
ISECF=1

3100 CONTINUE
GOTO 3110

3090 CONTINUE
DO 3120 J=1,5
IOPTB(J,INDEX)=1

3120 CONTINUE

3110 CONTINUE
GO TO 1190

C

C 8. PRE-OPERATIONAL SUPPORT

```

C -----
C
1150 CONTINUE
      IOPTB(1,INDEX)=1
      GO TO 1190

C
C 9. DECOMMISSIONING AND SURVEILLANCE
C -----
C
1160 CONTINUE
      IOPTB(1,INDEX)=1
      GO TO 1190

C
C CATEGORY AND SUBCATEGORY PROCESSING ROUTINES
C -----
C
C SET OPTION MATRIX TO ACCESS ALL AREAS
C
1170 CONTINUE
      DO 1180 I=1,10
        DO 1180 J=1,10
          IOPTB(I,J)=1
          IOPTC(I)=1
1180 CONTINUE
      INDEX=0
1185 INDEX=INDEX+1
C
1190 CONTINUE
      DO 1200 K=1,NW
        INUM(K)=K
1200 CONTINUE
      INWT=NW
      IF (IOPTA.EQ.1) GO TO 1350
      IF (IFLG.NE.1) GO TO 1350
      DO 1220 I=1,NW
        IWNUM(I) = 0
1220 CONTINUE
      WRITE (IO,1210)
1210 FORMAT (' WASTE TYPE SELECTION: '/
* ' DO YOU WISH TO UPDATE ALL WASTE TYPES OR SPECIFIC ONES? '/
* ' ENTER 0 = ALL WASTE TYPES '/
* ' 1 = SPECIFIC WASTE TYPES')
1214 READ(IN,*,ERR=1201) N
      GOTO 1213
1201 WRITE(IO,1216)
1216 FORMAT('0 ERROR IN INPUT'/' TRY AGAIN')
      GOTO 1214
1213 IF (N.EQ.0) GOTO 1270
      WRITE (IO,1211)

```

```

1211  FORMAT('  ENTER THE NUMBER OF WASTE TYPES YOU WISH TO UPDATE'//
*      FOLLOWED BY THE CORRESPONDING WASTE TYPE NUMBER.'/'
*      SEPARATE ENTRIES WITH A COMMA OR A SPACE')
      READ(IN,*) N,(IWNUM(J),J=1,N)
C
C  SORT WASTE INDEXES INTO ASCENDING ORDER
C
      DO 1230 I=1,NW
          INUM(I)=0
1230  CONTINUE
      INWT=0
      K=0
      INWT=0
      DO 1240 J=1,NW
          IF (IWNUM(J).EQ.0) GO TO 1240
          K=K+1
          INUM(K)=IWNUM(J)
          INWT=INWT+1
1240  CONTINUE
      IF (INWT.EQ.1) GOTO 1270
      L=INWT-1
      DO 1260 I=1,L
          K=I+1
          DO 1250 J=K,INWT
              IF (INUM(I).LT.INUM(J)) GO TO 1250
              NUM=INUM(I)
              INUM(I)=INUM(J)
              INUM(J)=NUM
1250  CONTINUE
1260  CONTINUE
1270  CONTINUE
1350  CONTINUE
C
C  READ OLD DATA FILE IF NECESSARY
C
1360  ICNT=ICNT+1
      IF (ICNT.GT.1.OR.IOPT.EQ.1) GO TO 1370
      CALL OLDCAS
C
C  ENTER DATA FOR CATEGORY SELECTED
C
1370  CONTINUE
      IF(IOPT.EQ.3) GO TO 1430
      IF(INDEX.GT.9) GO TO 100
      CALL READ (INDEX)
      IFLG = 0
      IF(IOPTA.NE.2) GO TO 1185
C
      IF(ISECF.EQ.0) GO TO 1387

```

```

1380 CONTINUE
      WRITE(IO,1385)
1385  FORMAT('0      ARE THERE ADDITIONAL SUBCATEGORIES TO UPDATE? ',
+         'ENTER 0 = NO'/
+         '1 = YES')
      READ(IN,1410) IANS
      IF(IANS.EQ.'1') GOTO 1402
      IF(IANS.EQ.'0') GOTO 1387
      WRITE(IO,1420)
      GO TO 1380
1402 CONTINUE
      DO 1386 K=1,10
        IOPTB(K,INDEX) = 0
1386 CONTINUE
      GO TO (380,560,690,810,980,1130,1140,1150,1160) INDEX
1387 CONTINUE
      WRITE (IO,1390)
1390 FORMAT ('0      ARE THERE ADDITIONAL CATEGORIES TO UPDATE? '
+         'ENTER 0 = NO'/
+         '1 = YES')
1400 CONTINUE
      READ (IN,1410) IANS
1410 FORMAT (A1)
      IF (IANS.EQ.'1') GO TO 230
      IF (IANS.EQ.'0') GO TO 100
      WRITE (IO,1420)
1420 FORMAT (' YOU MUST SPECIFY EITHER 1 OR 0 FOR PROPER RESPONSE'
+         '/ ' INPUT ANSWER AGAIN')
      GO TO 1400

C
C  EXIT FROM PREPROCESSOR
C  -----
C
1430 CONTINUE
      CALL WRITE
C
      CALL OUTPUT
C
      CALL TABLE
C
      WRITE (IO,1440)
1440 FORMAT ('0      THE PREPROCESSOR HAS COMPLETED EXECUTION.'
+         '/ ' YOUR UPDATED DATA FILE IS IN FILE 8.'
+         '/ ' YOU MAY NOW EXECUTE RECON.')
      END
      SUBROUTINE DCODEI(A,N)
C
      CHARACTER*80 A
      CHARACTER*1 BLANK

```

```

CHARACTER*10 C
DATA BLANK/1H /
DO 100 J=1,80
IF(A(J:J).EQ.BLANK) GO TO 100
  IF (ISTRT.EQ.0.AND.IEND.EQ.0) ISTRT = J
  IEND = IEND+1
100 CONTINUE
IF(IEND.EQ.0) RETURN
  IEND = IEND+ISTRT-1
  DO 10 K=1,10
    C(K:K)= BLANK
10 CONTINUE
  IT = 10-IEND+1
  C(IT:10) = A(ISTRT:IEND)
  DECODE(10,1,C,L) IA
  N = IA
200 CONTINUE
  IEND = 0
  RETURN
1 FORMAT(I10)
END
SUBROUTINE DCODER(A,B)

```

```

C
CHARACTER*80 A
CHARACTER*1 BLANK
CHARACTER*15 C
DATA BLANK/1H /
DO 100 J=1,80
IF(A(J:J).EQ.BLANK) GO TO 100
  IF (ISTRT.EQ.0.AND.IEND.EQ.0) ISTRT = J
  IEND = IEND+1
100 CONTINUE
IF(IEND.EQ.0) RETURN
  IEND = IEND+ISTRT-1
  DO 10 K=1,15
    C(K:K)= BLANK
10 CONTINUE
  IT = 15-IEND+1
  C(IT:15) = A(ISTRT:IEND)
  DECODE(15,1,C,L) D
  B = D
200 CONTINUE
  IEND = 0
  RETURN
1 FORMAT(G15.0)
END
SUBROUTINE OLDCAS
C
  INCLUDE PARAM

```

```

C
C   INCLUDE INPUT
C
C   ICIN = 11
C
C
C   READ(ICIN,1100) DATYPE(1)
C   READ(ICIN,1000) TITLE
C   READ(ICIN,1020) RPTYP,PLACE,TEKNEK
C   READ(ICIN,*) IYRCON,IYRPRO,LASTYR,IYDOL,IDDYR,NWT
1  , DRATE
C   IYRDIS = IYDOL
C   READ(ICIN,*) DEPTH,DENSTY,SPUC
1,CONFAC
C   NSTM1 = NST-1
C   READ(ICIN,*) (SFTNUM(L),L=1,NSTM1)
C   DO 75 K=1,NWT
C   READ(ICIN,1040) WSTYP(K),PKGDS1(K),PKGDS2(K),PKGDS3(K)
75 CONTINUE
C   READ(ICIN,*) (KIND(K),K=1,NWT),(UNTMAS(K),K=1,NWT)
C
C
C   LL=IYRPRO-IYRCON
C   LY=LASTYR-IYRCON+1
C   IDD=IDDYR
C   LYY=LY
C
C
C   WASTOP
C
C   READ(ICIN,1100) DATYPE(2)
C   READ(ICIN,*) SIZE, CPA
1  ,GXEF, (NFTL(K),K=1,NWT), FFTL
2  ,NALSH,NRHRMA,NCHRNA,WRKYR,MTRPSD,MTCPSD
3  ,(SHCFAC(L),L=1,NS),OVRHDF
C   READ(ICIN,*) (PANLEN(K),K=1,NWT)
C   READ(ICIN,*) (PANWTH(K),K=1,NWT)
C   READ(ICIN,*) (WUPP (K),K=1,NWT), (WVOLPU(K),K=1,NWT)
C   READ(ICIN,*) (TTTR (K),K=1,NWT), (TTTK (K),K=1,NWT)
C   READ(ICIN,*) (RCCAP (K),K=1,NWT), (TCCAP (K),K=1,NWT)
C   READ(ICIN,*) (PKGCAP(K),K=1,NWT)
C   READ(ICIN,*) (RMLN (K),K=1,NWT), (RMWTH (K),K=1,NWT)
C   READ(ICIN,*) (CLSEMP(K),K=1,NWT), (XRASHO(K),K=1,NWT)
C   READ(ICIN,*) (HPMIN (K),K=1,NWT), (PKGUPP(K),K=1,NWT)
C   READ(ICIN,*) (NBSPPA(K),K=1,NWT)
C   DO 50 K=1,NWT
C   READ(ICIN,*) (WSTSUP(K,I),I=1,LY)
50 CONTINUE
C   DO 60 K=1,NWT
C   READ(ICIN,*) (TDWI(I,K),I=1,LY)

```

```

60 CONTINUE
   DO 70 K=1,NWT
     READ(ICIN,*) (CRRAIL(K,I),I=1,LY)
70 CONTINUE
   DO 120 L = 1,NWO
     READ(ICIN,*) WOCST(L),WOLBRR(L),WOLBRC(L),WOEQPL(L),
1 WOEQPC(L),WOECST(L)
     READ(ICIN,*) (WOMATU(L,K),K=1,NWT),(WOMATC(L,K),K=1,NWT)
120 CONTINUE
C
C MINDEV
C
   READ(ICIN,1100) DATYPE(3)
   READ(ICIN,*) LEAD, NCOR,TCPCLN,PCTFIL,XPANFC,MSA,TONPAR,
1 OFTADC,NBYR,OTHBFL,OTBDEN,BKRATE,MSHIFT(1),MSHIFT(2)
   READ(ICIN,*) (RMHMIN(K),K=1,NWT)
   NCOR = NCOR + 1
   DO 145 M= 1, NCOR
     READ(ICIN,1010) CORDSC(M)
     READ(ICIN,*) CORWTH(M), CORHT(M), NCORPP(M)
     READ(ICIN,*) (CORLEN(M,K),K=1,NWT)
145 CONTINUE
   READ(ICIN,*)(DELRHT(K),K=1,NWT), (PKGLN(K),K=1,NWT),(NRWSPR(K),
1 K=1,NWT)
   READ(ICIN,*) (PORMDF(1,I),I=1,LL),(PORMDF(2,I),I=1,LL)
1 , (SCSKD(I),I=1,LL)
   READ(ICIN,*) (SFTDD(L),SFTXR(L),SFTXD(L),L=1,NST)
   READ(ICIN,*)MSWTCH
   IF(MSWTCH.EQ.0) GO TO 142
   DO 140 L=1,NRO
     READ(ICIN,*) ROCST(L),ROLBRR(L),ROLBRC(L),ROEQPL(L),
1 ROEQPC(L),ROECST(L),ROMATU(L),ROMATC(L)
140 CONTINUE
   GO TO 144
142 CONTINUE
   READ(ICIN,*) VARMC,RCKHC,SURFDC,BKFLC
144 CONTINUE
C
C RECEVE
C
   READ(ICIN,1100) DATYPE(4)
   READ(ICIN,*) NRBPBM, NCBPRM
   READ(ICIN,*)(RMCWS1(I),I=1,LL) , (RMCWS2(I),I=1,LL)
   READ(ICIN,*) (RMNWS1(I),I=1,LL) , (RMNWS2(I),I=1,LL)
C
C UGOPR
C
   READ(ICIN,1100) DATYPE(5)
   DO 130 L =1,NT0

```



```

      READ(ICIN,*) TOCST(L),TOLBRR(L),TOLBRC(L),TOEQPL(L),
1TOEQPC(L),TOECST(L)
      READ(ICIN,*) (TOMATU(L,K),K=1,NWT),(TOMATC(L,K),K=1,NWT)
130  CONTINUE
      READ(ICIN,*) IDET,IUHDCM
      READ(ICIN,*) (TRCHWD(K),K=1,NWT)
      DO 200 K=1,NWT
      READ(ICIN,*) NPKPEU(K),PKGDI A(K),REFDIA(K),HOLDEL(K),IMET(K)
200  CONTINUE
C
C  CO-EMPLACEMENT DATA
C
      READ(ICIN,*) MPRIME, NNHG, ICLR
      IF(MPRIME.EQ.NWT) GOTO 235
      READ(ICIN,*) (STOMAX(K),K=1,NWT)
      IF(NNHG.NE.NWT-MPRIME) NNHG = NWT-MPRIME
      IF(NNHG.EQ.0) GOTO 235
      DO 2000 K=1,MPRIME
      DO 2000 J=1,NNHG
      READ(ICIN,*) COCON(J,K),PCTCON(J,K)
2000 CONTINUE
      READ(ICIN,*) (SWPPR(K),K=1,NWT)
      DO 230 K=1,NWT
      IF(IMET(K).NE.2) GOTO 230
      IF(IMET(K).LE.4.OR.IMET(K).EQ.7.OR.IMET(K).EQ.10.OR.
+  IMET(K).GE.13) GOTO 230
      ICH = 1
230  CONTINUE
      IF(ICH.EQ.1) READ(ICIN,*) (CONTRU(J),J=1,NWT)
235  CONTINUE
      DO 250 L=1,NUO
      READ(ICIN,*) UOCST(L),UOLBRR(L), UOLBRC(L),UOEQPL(L),UOEQPC(L),
1UOECST(L)
      IF(IDET.EQ.1) GO TO 250
      DO 240 K=1,NWT
      IF(KIND(K).EQ.1) M=1
      IF(KIND(K).EQ.2) M=2
      IF(KIND(K).EQ.3) M=1
      IF(M.NE.L) GO TO 240
      READ(ICIN,*) UOMATU(L,K),UOMATC(L,K)
240 CONTINUE
250 CONTINUE
      DO 295 K=1,NWT
      READ(ICIN,*) WABATH(K),SLEETH(K),BCKFTH(K),CAPLEN(K)
295 CONTINUE
      IF(IDET.NE.1) GO TO 300
      DO 270 K=1,NWT
      READ(ICIN,*) REFMAT(K), REFSLV(K),REFBKF(K)
1,REFDEP(K)

```

```

    READ(ICIN,*) CHMTEM(K),PKGPRK(K),RCKCST(K)
270 CONTINUE
300 CONTINUE
    DO 155 L=1,NEE
    IF(IUHDCM.EQ.1.AND.L.EQ.1) GO TO 350
    READ(ICIN,*) EECST(L),EELBRR(L),EELBRC(L),EEEQPL(L),EEEQPC(L),
1 EEEECST(L),EEMATU(L),EEMATC(L)
    GO TO 155
350 CONTINUE
    READ(ICIN,*) EELBRC(L),EEECST(L),EEMATC(L),
1 HDLCFC,HDECFC,HDMCFC,HDCON1,HDCON2,NTYP
155 CONTINUE
C
C SHAFT
C
    READ(ICIN,1100) DATYPE(6)
    READ(ICIN,*) MXMPRH,MXMMRH, TNPT1, TNPT2,
1 SCCON1, SCCON2,IEQTP1, CCON1, CCON2, IEQTP2
2 ,WCCON1,WCCON2,IEQTP3
    READ(ICIN,*) (SFTLD(L),L=1,NST)
    READ(ICIN,*) (SFTHE(L),L=1,NST)
    READ(ICIN,*) (SFTSS(L),L=1,NST)
    READ(ICIN,*) (NPKPTR(K),K=1,NWT),(NPKPTC(K),K=1,NWT)
C
C SURFST
C
    READ(ICIN,1100) DATYPE(7)
    READ(ICIN,*) (SFCS(L),L=1,LL)
    READ(ICIN,*) NSC,SOA
    DO 150 L=1,NSC
    READ(ICIN,1010) SFTITL(L)
    READ(ICIN,*) SFCST(L),SFLBRR(L),SLBR(L),SFLBRC(L),SFEQPL(L),
1 SFECST(L),SFMATU(L),SFMATC(L)
150 CONTINUE
C
C UTILITY
C
    READ(ICIN,1100) DATYPE(8)
    READ(ICIN,*) UPC,STMC,DWTRC,WTRC,DSLCL
    READ(ICIN,*) RAPP,SFSP,OWHP,SFP,MROP,RVP,
1 ACP,HEATS,RAPS,ORS
    READ(ICIN,*) VOLW,VOLDW,SUD,UGD,RED
C
C PREOP
C
    READ(ICIN,1100) DATYPE(9)
    READ(ICIN,*) (SCDF(I),I=1,LL)
1, (CLNDF(I),I=1,LL)
2, (CLCDF(I),I=1,LL)

```

```

      READ(ICIN,*) TCLAND,TCLICE,STPCT,AECON1,AECON2,IEQTP4
C
C  POSTOP
C
      READ(ICIN,1100) DATYPE(10)
      IF (IDD.EQ.0) GOTO 190
      READ(ICIN,*) (DBDF(I),I=1,IDD)
      READ(ICIN,*) (SSDF(I),I=1,IDD)
      READ(ICIN,*) (FDDF(I),I=1,IDD)
190  CONTINUE
      READ(ICIN,*) SSSCON1, SSSCON2, IEQTP5, DCOMCF, SURCST, ITSURV
      READ(ICIN,*) NDDC,DDFRAC
      IF(NDDC.EQ.0) GO TO 165
      DO 160 L=1,NDDC
      READ(ICIN,1010) DDTITL(L)
      READ(ICIN,*) DDCST(L),DDLBR(L),DDLBR(L),DDEQPC(L),DDMATC(L)
160  CONTINUE
165  CONTINUE
      READ(ICIN,*) SCCST,SCLBR,LCLBR,SCEQPC,SCMATU,SCMATC
C
C
1100 FORMAT(A80)
C
C  TABLES TO BE PRINTED
C
      DO 400 I=1,27
      TABLES(I)=0
400  CONTINUE
      DO 410 I=1,27
      READ(ICIN,*,ERR=420) TABLES
410  CONTINUE
420  IF(TABLES(1).NE.0) RETURN
      DO 430 I=1,27
      TABLES(I)=I
430  CONTINUE
      RETURN
1000 FORMAT(A80)
1010 FORMAT(A48)
1020 FORMAT(3A16)
1040 FORMAT(A12,3A16)
      END
      SUBROUTINE OTDECM
      INCLUDE PARAM
      INCLUDE INPUT
      INCLUDE OPTION
      CHARACTER*1
+      BLANK
      DATA BLANK /1H /
C

```

C V. DECOMMISSIONING AND SURVEILLANCE ACTIVITIES

```

C
  WRITE (IOUT,100)
100 FORMAT (' '//, ***** DECOMMISSIONING AND SURVEILLANCE',
+ ' *****',
+3X,76('=')/,3X,76('='))
  IF (IEQTP5.EQ.1) WRITE (IOUT,110)
110 FORMAT (' '//, IEQTP5 (GENERIC - ALL)'
*      /'      SHAFT SEALING FUNCTIONAL RELATIONSHIP '
*      'IS LINEAR')
  IF (IEQTP5.EQ.2) WRITE (IOUT,120)
120 FORMAT (' '//, IEQTP5 (GENERIC - ALL)'
*      /'      SHAFT SEALING FUNCTION RELATIONSHIP '
*      'IS EXPONENTIAL')
  WRITE (IOUT,130)
130 FORMAT (' '//, TO CHANGE FUNCTIONAL RELATIONSHIP ENTER THE'
* ' CORRESPONDING INTEGER'
*      /'      1 = LINEAR'/'      2 = EXPONENTIAL')
  IF (IEQTP5.EQ.1) WRITE (IOUT,140)
140 FORMAT (' '//, FUNCTION = A + B * D'
*      /'      WHERE D = .SFTDD(L) + 2*SFTXR(L)'
*      /'      WHERE L=1,NST')
  IF (IEQTP5.EQ.2) WRITE (IOUT,150)
150 FORMAT (' '//, FUNCTION = A * EXP ( B * D )'
*      /'      WHERE D = SFTDD(L) + 2*SFTXR(L)'
*      /'      WHERE L=1,NST')
  WRITE (IOUT,160) SSCON1
160 FORMAT (' '//, SSCON1 (GENERIC - ALL)'
*      /'      SHAFT SEALING FUNCTION',T50,G15.5
*      /'      COEFFICIENT A')
  WRITE (IOUT,170) SSCON2
170 FORMAT (' '//, SSCON2 (GENERIC - ALL)'
*      /'      SHAFT SEALING FUNCTION ',T50,G15.5
*      /'      COEFFICIENT B')
  WRITE (IOUT,180) DCOMCF
180 FORMAT (' '//, DCOMCF (GENERIC - ALL)'
*      /'      SUPPORT FACILITY',T50,G15.5
*      /'      DECOMMISSIONING COST, FRACTION OF'
*      /'      CONSTRUCTION COST')
  WRITE (IOUT,190) SURCST
190 FORMAT (' '//, SURCST (GENERIC - ALL)'
*      /'      ANNUAL SURVEILLANCE COST, $',T50,G15.5)
  DO 200 J=1,NDDC
    DDCST(J)=0.0
    DDLBRR(J)=0.0
    DDLBRC(J)=0.0
    DDEQPC(J)=0.0
    DDMATC(J)=0.0
200 CONTINUE

```

```

C
C   SEALING COSTS
C
  SCCST=0.0
  SCLBRR=0.0
  SCLBRC=0.0
  SCEQPC=0.0
  SCMATU=0.0
  SCMATC=0.0

C
C   ZERO OUT REMAINING YEARS
C
  LL=LASTYR-IYRCON
  DO 210 I=LL+1,NC
    RMCWS1(I)=0.0
    RMCWS2(I)=0.0
    RMNWS1(I)=0.0
    RMNWS2(I)=0.0
    SFCS(I)=0.0
    SCSKD(I)=0.0
    CLCDF(I)=0.0
    CLNDF(I)=0.0
    PORMDF(1,I)=0.0
    PORMDF(2,I)=0.0
  210 CONTINUE

C
  DO 220 I=IDDYR+1,NC
    DBDF(I)=0.0
    SSDF(I)=0.0
    FDDF(I)=0.0
  220 CONTINUE
  LY=LASTYR-IYRCON+1
  DO 240 K=1,NWT
    DO 230 I=LY+1,NC
      WTSUP(K,I)=0.0
      TDWI(I,K)=0.0
      CRRAIL(K,I)=0.0
    230 CONTINUE
  240 CONTINUE
  RETURN
  END
  SUBROUTINE OTGEN

C
  INCLUDE PARAM
  INCLUDE OPTION
  INCLUDE INPUT
  CHARACTER*1
+   BLANK
  DATA BLANK /1H /

```

```

C I. GENERAL SYSTEMS DATA
C
C
C A. CASE DESCRIPTION AND TITLE DESCRIPTION
C
      WRITE(IOUT,5)
5     FORMAT(' ' /'          ***** GENERAL SYSTEMS DATA *****',/,
      +3X,76('=')/,3X,76('='))
C
C
      WRITE(IOUT,8)
8     FORMAT(' ' /'          CASE TITLE DESCRIPTION' /6X,22('='))
      WRITE (IOUT,100) TITLE
100   FORMAT (' ' /'          CASE TITLE DESCRIPTION '
      *      /T8,A80)
C
C B. SITE PARAMETERS
C
      WRITE (IOUT,110)
110   FORMAT (' ' /'          SITE PARAMETERS' /4X,15('='))
      WRITE (IOUT,120) PLACE
120   FORMAT (' ' /'          PLACE'
      *      /'          REPOSITORY LOCATION, DESCRIPTION ',T50,A16
      *      /'          (LIMIT 16 CHARACTERS)')
      WRITE (IOUT,130) RPTYP
130   FORMAT (' ' /'          RPTYP'
      *      /'          REPOSITORY TYPE,',T50,A16
      *      /'          (LIMIT 16 CHARACTERS)')
      WRITE (IOUT,140) DENSTY
140   FORMAT (' ' /'          DENSTY'
      *      /'          ROCK DENSITY, G/CM**3',T50,G15.5)
      WRITE (IOUT,150) SIZE
150   FORMAT (' ' /'          SIZE'
      *      /'          UNDERGROUND REPOSITORY AREA, ACRE ',T50,G15.5)
      WRITE (IOUT,160) DEPTH
160   FORMAT (' ' /'          DEPTH'
      *      /'          DEPTH, METERS ',T50,G15.5)
      WRITE (IOUT,170) TCLAND
170   FORMAT (' ' /'          TCLAND'
      *      /'          LAND COST, $ ',T50,G15.3)
      WRITE (IOUT,180) SPUC
180   FORMAT (' ' /'          SPUC'
      *      /'          SITE PREPARATION COST, ',T50,G15.3
      *      /'          $/ACRE')
C
C C. CONSTRAINTS
C

```

```

      WRITE (IOUT,190)
190  FORMAT (' '///' CONSTRAINTS'/5X,11('='))
      WRITE (IOUT,200)
200  FORMAT (' '///'          NFTL(K)          BY WASTE TYPE'
*          /'          NEAR-FIELD THERMAL LIMIT ',
*          /'          KW/ACRE')
      DO 201 K=1,NWT
      WRITE (IOUT,202) K,NFTL(I)
202  FORMAT(' ' /T8,'WASTE TYPE ',I3,T25,G15.5)
201  CONTINUE
      WRITE (IOUT,210) FFTL
210  FORMAT (' '///'          FFTL'
*          /'          FAR-FIELD THERMAL LIMIT ',T50,G15.5
*          /'          KW/ACRE')
      WRITE (IOUT,220) NWT
220  FORMAT (' '///'          NWT'
*          /'          NUMBER OF WASTE TYPES FOR EMPLACEMENT'
*          ,T50,I10)
      WRITE (IOUT,230)
230  FORMAT (' '///'          HPMIN(K)  (BY WASTE TYPE)'
*          /'          MINIMUM HOLE PITCH, METERS'/)
      DO 250 K=1,NWT
      WRITE (IOUT,240) K,HPMIN(K)
240  FORMAT (' ',T8,'WASTE TYPE ',I2,T51,G15.5)
250  CONTINUE
      WRITE (IOUT,260) NALSH
260  FORMAT (' '///'          NALSH          (GENERIC - ALL)'
*          /'          MAXIMUM NUMBER OF SHIFTS ALLOWED'
*          ,T50,I10)
      WRITE (IOUT,270) WRKYR
270  FORMAT (' '///'          WRKYR          (GENERIC - ALL)'
*          /'          NUMBER OF WORK DAYS PER YEAR',T50,G15.5)
C
C  D.  REPOSITORY TIMING AND SOURCE TERM DATA
C
C
C
      WRITE (IOUT,280)
280  FORMAT (' '///'          REPOSITORY SCHEDULE AND WASTE GENERATION DATA'
*          /5X,45('='))
      WRITE (IOUT,290) IYRCON
290  FORMAT (' '///'          IYRCON'
*          /'          CALENDAR YEAR CONSTRUCTION BEGINS ',T55,I5)
      WRITE (IOUT,300) IYRPRO
300  FORMAT (' '///'          IYRPRO'
*          /'          CALENDAR YEAR PRODUCTION BEGINS ',T55,I5)
      WRITE (IOUT,310) LASTYR
310  FORMAT (' '///'          LASTYR'
*          /'          LAST YEAR OF WASTE RECEIPTS ',T55,I5)

```

```

      WRITE (IOUT,320) NBYR
320  FORMAT (' ','/' NBYR'
*      /' 'YEARS AFTER PRODUCTION STARTUP THAT',T55,I5
*      /' 'BACKFILLING BEGINS')
      WRITE (IOUT,330) IDDYR
330  FORMAT (' ','/' IDDYR'
*      /' 'NUMBER OF YEARS OF DECOMMISSIONING',T50,I10)

```

C

```

      MM=LASTYR-IYRCON+1
      DO 370 K=1,NWT
        WRITE (IOUT,340)
340    FORMAT (' ',T8,'WASTE QUANTITY AVAILABLE FOR PROCESSING'
*      /T9,'UNITS/YEAR')
        WRITE (IOUT,350) K
350    FORMAT (' ',T8,'WSTSUP(K,I) (BY WASTE TYPE AND YEAR)
*      /T10,'FOR WASTE TYPE',I5/)
        DO 370 J=1,MM
          J1=IYRCON+J-1
          WRITE (IOUT,360) J1,WSTSUP(K,J)
360    FORMAT (' ',T8,'YEAR:',I5,3X,G15.5)
370  CONTINUE

```

C

C

C

C

C

D. CONSTRUCTION AND COST DISTRIBUTION SCHEDULES

C

```

      LL=IYRPRO-IYRCON
      WRITE (IOUT,380)
380  FORMAT (' '//' CONSTRUCTION AND COST DISTRIBUTION SCHEDULES'
*      /5X,44('='))
      WRITE (IOUT,390)
390  FORMAT (' '//' RMCWS1 (R-H - ALL BY YEAR)'
*      /' 'R-H WASTE RECEIVING MODULE CONSTRUCTION '
*      /' 'SCHEDULE'
*      /' 'PRIOR TO OPERATIONS, FRACTION/YEAR')
      DO 410 J=1,LL
        J1=IYRCON+J-1
        WRITE (IOUT,400) J1,RMCWS1(J)
400    FORMAT (' ',T8,'YEAR ',I5,3X,G15.5)
410  CONTINUE
      WRITE (IOUT,420)
420  FORMAT (' '//' RMCWS2 (R-H - ALL BY YEAR)'
*      /' 'R-H WASTE RECEIVING MODULE CONSTRUCTION '
*      /' 'SCHEDULE' /' 'DURING OPERATIONS, FRACTION/YEAR')
      DO 430 J=1,LL
        J1=IYRCON+J-1
        WRITE (IOUT,360) J1,RMCWS2(J)
430  CONTINUE

```



```

WRITE (IOUT,440)
440 FORMAT (' ' /' RMNWS1 (C-H - ALL BY YEAR)'
* ' /' C-H WASTE RECEIVING MODULE CONSTRUCTION '
* 'SCHEDULE' /' PRIOR TO OPERATIONS, FRACTION/YEAR')
DO 460 J=1,LL
J1=IYRCON+J-1
WRITE (IOUT,450) J1,RMNWS1(J)
450 FORMAT (' ',T8,'YEAR:',I5,3X,G15.5)
460 CONTINUE
WRITE (IOUT,470)
470 FORMAT (' ' /' RMNWS2 (C-H - ALL BY YEAR)'
* ' /' C-H WASTE RECEIVING MODULE CONSTRUCTION '
* 'SCHEDULE' /' DURING OPERATIONS, FRACTION/YEAR')
DO 480 J=1,LL
J1=IYRCON+J-1
WRITE (IOUT,450) J1,RMNWS2(J)
480 CONTINUE
WRITE (IOUT,490)
490 FORMAT (' ' /' SFCS (GENERIC - ALL BY YEAR)'
* ' /' OTHER SURFACE FACILITY CONSTRUCTION '
* ' /' SCHEDULE, FRACTION/YEAR')
DO 500 J=1,LL
J1=IYRCON+J-1
WRITE (IOUT,450) J1,SFCS(J)
500 CONTINUE
WRITE (IOUT,510)
510 FORMAT (' ' /' PORMDF(1,I) (GENERIC - ALL BY YEAR)'
* ' /' SHAFT PILLAR MINING COST DISTRIBUTION '
* ' /' SCHEDULE, FRACTION/YEAR')
DO 520 J=1,LL
J1=IYRCON+J-1
WRITE (IOUT,450) J1,PORMDF(1,J)
520 CONTINUE
WRITE (IOUT,530)
530 FORMAT (' ' /' PORMDF(2,I) (GENERIC - ALL BY YEAR)'
* ' /' MAIN CORRIDOR MINING COST DISTRIBUTION '
* ' /' SCHEDULE, FRACTION/YEAR')
DO 540 J=1,LL
J1=IYRCON+J-1
WRITE (IOUT,450) J1,PORMDF(2,J)
540 CONTINUE
WRITE (IOUT,550)
550 FORMAT (' ' /' SCSKD (GENERIC - ALL BY YEAR)'
* ' /' SHAFT CONSTRUCTION COST DISTRIBUTION '
* ' /' SCHEDULE, FRACTION/YEAR')
DO 560 J=1,LL
J1=IYRCON+J-1
WRITE (IOUT,450) J1,SCSKD(J)
560 CONTINUE

```

```

WRITE (IOUT,570)
570 FORMAT (' ' /'          CLCDF          (GENERIC - ALL BY YEAR)'
*          /'          LICENSING COST DISTRIBUTION SCHEDULE,'
*          /'          FRACTION/YEAR')
DO 580 J=1,LL
J1=IYRCON+J-1
WRITE (IOUT,450) J1,CLCDF(J)
580 CONTINUE
WRITE (IOUT,590)
590 FORMAT (' ' /'          CLNDF          (GENERIC - ALL BY YEAR)'
*          /'          LAND COST DISTRIBUTION SCHEDULE,'
*          /'          FRACTION/YEAR')
DO 600 J=1,LL
J1=IYRCON+J-1
WRITE (IOUT,450) J1,CLNDF(J)
600 CONTINUE
WRITE (IOUT,610)
610 FORMAT (' ' /'          SCDF          (GENERIC - ALL BY YEAR)'
*          /'          STARTUP COST DISTRIBUTION SCHEDULE,'
*          /'          FRACTION/YEAR')
DO 620 J=1,LL
J1=IYRCON+J-1
WRITE (IOUT,450) J1,SCDF(J)
620 CONTINUE
WRITE (IOUT,630)
630 FORMAT (' ' /'          DBDF          (GENERIC - ALL BY YEAR)'
*          /'          SHAFT PILLAR & CORRIDOR BACKFILLING COST'
*          /'          DISTRIBUTION SCHEDULE, FRACTION/YEAR')
DO 640 J=1,IDDYR
WRITE (IOUT,450) J,DBDF(J)
640 CONTINUE
WRITE (IOUT,650)
650 FORMAT (' ' /'          SSDF          (GENERIC - ALL BY YEAR)'
*          /'          SHAFT SEALING COST DISTRIBUTION SCHEDULE,'
*          /'          FRACTION/YEAR')
DO 660 J=1,IDDYR
WRITE (IOUT,450) J,SSDF(J)
660 CONTINUE
WRITE (IOUT,670)
670 FORMAT (' ' /'          FDDF          (GENERIC - ALL BY YEAR)'
*          /'          FACILITY DECOMMISSIONING COST DISTRIBUTION'
*          /'          SCHEDULE, FRACTION/YEAR')
DO 680 J=1,IDDYR
WRITE (IOUT,450) J,FDDF(J)
680 CONTINUE
WRITE (IOUT,690) ITSURV
690 FORMAT (' ' /'          ITSURV          (GENERIC - ALL)'
*          /'          SURVEILANCE TIME PERIOD, YEARS',T55,I5)

```

C

C E. MISCELLANEOUS COST INPUT DATA

C
C

```

WRITE (IOUT,700)
700 FORMAT (' '//' MISCELLANEOUS COST PARAMETERS'/5X,29('='))
WRITE (IOUT,710) SHCFAC(1)
710 FORMAT (' '//' SHCFAC(1) (GENERIC - ALL)'
* /' SHIFT 1, COST DIFFERENTIALS',T50,G15.5)
WRITE (IOUT,720) SHCFAC(2)
720 FORMAT (' '//' SHCFAC(2) (GENERIC - ALL)'
* /' SHIFT 2, 1.+ FRACTION OF SHIFT 1',T50,G15.5)
WRITE (IOUT,730) SHCFAC(3)
730 FORMAT (' '//' SHCFAC(3) (GENERIC - ALL)'
* /' SHIFT 3, 1.+ FRACTION OF SHIFT 1',T50,G15.5)
WRITE (IOUT,740) OVRHDF
740 FORMAT (' '//' OVRHDF (GENERIC - ALL)'
* /' LABOR OVERHEAD FACTOR, ',T50,G15.5
* /' 1+FRACTION OF SALARIES')
WRITE (IOUT,760)
760 FORMAT (' '//' CONTINGENCY FACTORS')
WRITE (IOUT,770) CONFAC(1)
770 FORMAT (' '//' CONFAC(1) (GENERIC - ALL)'
* /' SITE PREPARATION, FRACTION ',T50,G15.5)
WRITE (IOUT,780) CONFAC(2)
780 FORMAT (' '//' CONFAC(2) (GENERIC - ALL)'
* /' ROCK OPERATIONS, FRACTION',T50,G15.5)
WRITE (IOUT,790) CONFAC(3)
790 FORMAT (' '//' CONFAC(3) (GENERIC - ALL)'
* /' WASTE OPERATIONS, FRACTION',T50,G15.5)
WRITE (IOUT,800) CONFAC(4)
800 FORMAT (' '//' CONFAC(4) (GENERIC - ALL)'
* /' BACKFILLING + DECOMMISSIONING, ',T50,G15.5
* /' FRACTION')
WRITE (IOUT,810) DRATE
810 FORMAT (' '//' DRATE (GENERIC - ALL)'
* /' DISCOUNT RATE, FRACTION',T50,G15.5)
WRITE (IOUT,820) IYDOL
820 FORMAT (' '//' IYDOL (GENERIC - ALL)'
* /' BASE YEAR FOR DISCOUNTING ',T55,I5)

```

C

```

RETURN
END
SUBROUTINE OTPREP
INCLUDE PARAM
INCLUDE OPTION
INCLUDE INPUT
CHARACTER*1
+ BLANK
DATA BLANK /1H /

```

C
C
C

IV. PRE-OPERATIONAL SUPPORT ACTIVITIES AND COSTS

```

WRITE (IOUT,100)
100 FORMAT (' '//' ***** PRE-OPERATIONAL SUPPORT *****'/3X
+,76('=')/3X,76('='))
WRITE (IOUT,110) TCLICE
110 FORMAT (' '//' TCLICE (GENERIC - ALL)'
* /' LICENSING COST, $',T50,G15.5)
WRITE (IOUT,120) STPCT
120 FORMAT (' '//' STPCT (GENERIC - ALL)'
* /' START-UP COST, FRACTION',T50,G15.5
* /' OF FIELD COST')
IF (IEQTP4.EQ.1) WRITE (IOUT,130)
130 FORMAT (' '//' IEQTP4'
* /' A-E COST FUNCTIONAL RELATIONSHIP IS LINEAR')
IF (IEQTP4.EQ.2) WRITE (IOUT,140)
140 FORMAT (' '//' IEQTP4'
* /' A-E COST FUNCTIONAL RELATIONSHIP IS '
* 'EXPONENTIAL')
WRITE (IOUT,150)
150 FORMAT (' '//' TO CHANGE ENTRY ENTER THE CORRESPONDING'
* ' INTEGER',
* /' 1 = LINEAR'
* /' 2 = EXPONENTIAL')
IF (IEQTP4.EQ.1) WRITE (IOUT,160)
160 FORMAT (' '//' FUNCTION = A + B * D'
* /' WHERE D = OSFC + UGFC + SHDF + S[UC * '
* 'SOA'
* /' AECON1 = A/(NWT) * (CONSTRUCTION YEARS)'
* /' AECON2 = A/(NWT) * (CONSTRUCTION YEARS)')
IF (IEQTP4.EQ.2) WRITE (IOUT,170)
170 FORMAT (' '//' FUNCTION = A * EXP ( B * D )'
* /' WHERE D = OSFC + UGFC + SHDF + S[UC * '
* 'SOA'
* /' AECON1 = A/(NWT) * (CONSTRUCTION YEARS)'
* /' AECON2 = A/(NWT) * (CONSTRUCTION YEARS)')
WRITE (IOUT,180) AECON1
180 FORMAT (' '//' AECON1 (GENERIC - ALL)'
* /' A-E COST FUNCTION CONSTANT AECON1',T50,G15.5
*)
WRITE (IOUT,190) AECON2
190 FORMAT (' '//' AECON2 (GENERIC - ALL)'
* /' A-E COST FUNCTION CONSTANT AECON2',T50,G15.5
*)

```

C

```

RETURN
END
SUBROUTINE OTROCK

```

```

        INCLUDE PARAM
        INCLUDE OPTION
        INCLUDE INPUT
        CHARACTER*1
        +   BLANK
        DATA BLANK /1H /

C
C
C   ROCK HANDLING AND DISPOSAL DESCRIPTIONS AND COST ELEMENTS
C
C   C1  GENERAL INFORMATION
C
C
      WRITE (IOUT,100)
100 FORMAT (' ' /' ***** ROCK HANDLING - DISPOSAL DESCRIPTION',
+ 'AND COSTS *****' /3X,76('=')/3X,76('='))
      WRITE (IOUT,110)
110 FORMAT (' ' /' GENERAL INFORMATION' /5X,19('='))
      WRITE (IOUT,120) TEKNEK
120 FORMAT (' ' /' TEKNEK (GENERIC - ALL)'
*           /' MINING TECHNIQUE,' ,T50,A16
*           /' (LIMIT 16 CHARACTERS)')
      WRITE (IOUT,130) LEAD
130 FORMAT (' ' /' LEAD (GENERIC - ALL)'
*           /' TIME LAG BETWEEN ROOM EXCAVATION',T50,I10
*           /' AND WASTE EMPLACEMENT')
      WRITE (IOUT,140) MSHIFT(1)
140 FORMAT (' ' /' MSHIFT(1) (GENERIC - ALL)'
*           /' NUMBER OF SHIFT/DAY WORKED',T50,I10
*           /' (250 DAY YEAR) FOR MINING DURING'
*           /' REPOSITORY CONSTRUCTION')
      WRITE (IOUT,150) MSHIFT(2)
150 FORMAT (' ' /' MSHIFT(2) (GENERIC - ALL)'
*           /' NUMBER OF SHIFT/DAY WORKED',T50,I10
*           /' (250 DAY YEAR) FOR MINING DURING'
*           /' REPOSITORY OPERATION')
      WRITE (IOUT,160) MSWTCH
160 FORMAT (' ' /' MSWTCH (GENERIC - ALL)'
*           /' ROCK HOISTING COST CALCULATION'
*           /' OPTION = ',T50,I3
*           /' WHERE '
*           /' OPTION = 0 : CALCULATE USING UNIT'
*           /' , ' COST PER TONNE OF ROCK'
*           /' OPTION = 1 : CALCULATE USING RESOURCE'
*           /' REQUIREMENT & COSTS')

C
C
C   ROCK EXCAVATION
C
      WRITE (IOUT,170)

```

```

170 FORMAT (' '/'      ROCK EXCAVATION',/5X,15('='))
    IF (MSWTCH.EQ.1) GO TO 190
C
C  OPTION 1 MSWTCH = 0
C
    WRITE (IOUT,180) VARMC
180 FORMAT (' '/'      VARMC      (GENERIC - ALL)'
*          /'          MINING UNIT COST, $/TONNE',T50,G15.5)
    GO TO 280
C
190 CONTINUE
C
C  OPTION 2 MSWTCH = 1
C
    WRITE (IOUT,200) ROLBRR(1)
200 FORMAT (' '/'      ROLBRR(1)      (GENERIC - ALL)'
*          /'          ROCK EXCAVATION & ROOM',T50,G15.5
*          /'          DEVELOPMENT LABOR REQUIREMENT,'
*          /'          MAN-YEARS/EQUIPMENT UNIT/SHIFT')
    WRITE (IOUT,210) ROEQPL(1)
210 FORMAT (' '/'      ROEQPL(1)      (GENERIC - ALL)'
*          /'          ROCK EXCAVATION & ROOM',T50,G15.5
*          /'          DEVELOPMENT EQUIPMENT LIFE,'
*          /'          TONNES/UNIT')
    WRITE (IOUT,220) ROEQPC(1)
220 FORMAT (' '/'      ROEQPC(1)      (GENERIC - ALL)'
*          /'          ROCK EXCAVATION & ROOM',T50,G15.5
*          /'          DEVELOPMENT EQUIPMENT CAPACITY')
    WRITE (IOUT,230) ROMATU(1)
230 FORMAT (' '/'      ROMATU(1)      (GENERIC - ALL)'
*          /'          ROCK EXCAVATION & ROOM',T50,G15.5
*          /'          DEVELOPMENT MATERIAL REQUIREMENT,'
*          /'          MATERIAL UNIT/TONNE')
    WRITE (IOUT,240) ROCST(1)
240 FORMAT (' '/'      ROCST(1)      (GENERIC - ALL)'
*          /'          ROCK EXCAVATION AND ROOM',T50,G15.5
*          /'          DEVELOPMENT CAPITAL COST, $')
    WRITE (IOUT,250) ROLBRC(1)
250 FORMAT (' '/'      ROLBRC(1)      (GENERIC - ALL)'
*          /'          ROCK EXCAVATION AND ROOM',T50,G15.5
*          /'          DEVELOPMENT UNIT LABOR COST, $/MAN-YEAR')
    WRITE (IOUT,260) ROECST(1)
260 FORMAT (' '/'      ROECST(1)      (GENERIC - ALL)'
*          /'          ROCK EXCAVATION AND ROOM',T50,G15.5
*          /'          DEVELOPMENT UNIT EQUIPMENT COST, $/UNIT')
    WRITE (IOUT,270) ROMATC(1)
270 FORMAT (' '/'      ROMATC(1)      (GENERIC - ALL)'
*          /'          ROCK EXCAVATION AND ROOM',T50,G15.5
*          /'          DEVELOPMENT MATERIAL COST,'

```

```

      *          /'          $/MATERIAL UNIT')
280 CONTINUE
C
C C3  ROCK HOISTING
C
C
      WRITE (IOUT,290)
290  FORMAT (' ' /'          ROCK HOISTING'/5X,13('='))
      IF (MSWTCH.EQ.1) GO TO 310
C
C  MSWTCH = 0
C
      WRITE (IOUT,300) RCKHC
300  FORMAT (' ' /'          RCKHC          (GENERIC - ALL)'
      *          /'          ROCK HOISTING UNIT COST,',T50,G15.5
      *          /'          $/TONNE')
      GO TO 400
C
C  MSWTCH = 1
C
310 CONTINUE
      WRITE (IOUT,320) ROLBRR(2)
320  FORMAT (' ' /'          ROLBRR(2)          (GENERIC - ALL)'
      *          /'          HOISTING LABOR REQUIREMENT,',T50,G15.5
      *          /'          MAN-YEARS/UNIT/SHIFT')
      WRITE (IOUT,330) ROEQPL(2)
330  FORMAT (' ' /'          ROEQPL(2)          (GENERIC - ALL)'
      *          /'          HOISTING EQUIPMENT LIFE,',T50,G15.5
      *          /'          TONNES/UNIT')
      WRITE (IOUT,340) ROEQPC(2)
340  FORMAT (' ' /'          ROEQPC(2)          (GENERIC - ALL)'
      *          /'          HOISTING EQUIPMENT CAPACITY,',T50,G15.5
      *          /'          TONNES/UNIT/SHIFT')
      WRITE (IOUT,350) ROMATU(2)
350  FORMAT (' ' /'          ROMATU(2)          (GENERIC - ALL)'
      *          /'          HOISTING MATERIAL REQUIREMENT,',T50,G15.5
      *          /'          TONNES/UNIT/SHIFT')
      WRITE (IOUT,360) ROCST(2)
360  FORMAT (' ' /'          ROCST(2)          (GENERIC - ALL)'
      *          /'          HOISTING CAPITAL COST, $',T50,G15.5)
      WRITE (IOUT,370) ROLBRC(2)
370  FORMAT (' ' /'          ROLBRC(2)          (GENERIC - ALL)'
      *          /'          HOISTING UNIT LABOR COST,',T50,G15.5
      *          /'          $/MAN-YEAR')
      WRITE (IOUT,380) ROECST(2)
380  FORMAT (' ' /'          ROECST(2)          (GENERIC - ALL)'
      *          /'          HOISTING EQUIPMENT COST,',T50,G15.5
      *          /'          $/UNIT')
      WRITE (IOUT,390) ROMATC(2)

```

```

390 FORMAT (' '//' ROMATC(2) (GENERIC - ALL)'
*          /' HOISTING MATERIAL COST,',T50,G15.5
*          /' $/MATERIAL UNIT')
400 CONTINUE
C
C
WRITE (IOUT,410) MSA
410 FORMAT (' '//' MSA (REAL NUMBER) (GENERIC - ALL)'
*          /' MINED ROCK STORAGE AREA, ACRES',T50,G15.5
*          /' (ZERO RECON WILL CALCULATE)')
WRITE (IOUT,420) TONPAR
420 FORMAT (' '//' TONPAR (GENERIC - ALL)'
*          /' ROCK STORED PER UNIT SURFACE',T50,G15.5
*          /' AREA, TONNES/ACRE')
C
C C4 ROCK SURFACE STORAGE
C
WRITE (IOUT,425)
425 FORMAT (' '//' SURFACE STORAGE'/5X,15('='))
C
C MSWTCH = 0
C
IF (MSWTCH.EQ.1) GO TO 440
WRITE (IOUT,430) SURFDC
430 FORMAT (' '//' SURFDC (GENERIC - ALL)'
*          /' UNIT SURFACE STORAGE COST,',T50,G15.5
*          /' $/TONNE')
GO TO 530
C
C MSWTCH = 1
C
440 CONTINUE
WRITE (IOUT,450) ROLBRR(3)
450 FORMAT (' '//' ROLBRR(3) (GENERIC - ALL)'
*          /' ROCK STORAGE LABOR REQUIREMENT,',T50,G15.5
*          /' MAN-YEARS/UNIT/SHIFT')
WRITE (IOUT,460) ROEQPL(3)
460 FORMAT (' '//' ROEQPL(3) (GENERIC - ALL)'
*          /' ROCK STORAGE EQUIPMENT LIFE,',T50,G15.5
*          /' TONNES/UNIT')
WRITE (IOUT,470) ROEQPC(3)
470 FORMAT (' '//' ROEQPC(3) (GENERIC - ALL)'
*          /' ROCK STORAGE EQUIPMENT',T50,G15.5
*          /' CAPACITY, TONNES/UNIT/SHIFT')
WRITE (IOUT,480) ROMATU(3)
480 FORMAT (' '//' ROMATU(3) (GENERIC - ALL)'
*          /' ROCK STORAGE MATERIAL',T50,G15.5
*          /' REQUIREMENT, MATERIAL UNIT/TONNE')
WRITE (IOUT,490) ROCST(3)

```



```

490 FORMAT (' '/' ROCST(3) (GENERIC - ALL)'
*          '/' ROCK STORAGE CAPITAL COST, '$',G15.5)
WRITE (IOUT,500) ROLBRC(3)
500 FORMAT (' '/' ROLBRC(3) (GENERIC - ALL)'
*          '/' ROCK STORAGE UNIT LABOR COST, ',T50,G15.5
*          '/' , MAN-YEARS/UNIT/SHIFT')
WRITE (IOUT,510) ROECST(3)
510 FORMAT (' '/' ROECST(3) (GENERIC - ALL)'
*          '/' ROCK STORAGE EQUIPMENT COST, ',T50,G15.5
*          '/' $/UNIT')
WRITE (IOUT,520) ROMATC(3)
520 FORMAT (' '/' ROMATC(3) (GENERIC - ALL)'
*          '/' ROCK STORAGE MATERIAL COST, ',T50,G15.5
*          '/' $/MATERIAL UNIT')
530 CONTINUE

```

```

C
C C5 OFF SITE DISPOSAL
C
C

```

```

WRITE (IOUT,540)
540 FORMAT (' '// OFFSITE DISPOSAL'/5X,16('='))

```

```

C
C
C
C
WRITE (IOUT,550) OFTADC
550 FORMAT (' '/' OFTADC (GENERIC - ALL)'
*          '/' UNIT OFFSITE TRANSPORTATION',T50,G15.5
*          '/' AND DISPOSAL COST')

```

```

C
C
C
C
MSWTCH = 1

```

```

IF(MSWTCH.EQ.0) GOTO 650
560 CONTINUE
WRITE (IOUT,570) ROLBRR(4)
570 FORMAT (' '/' ROLBRR(4) (GENERIC - ALL)'
*          '/' OFFSITE TRANSPORTATION AND',T50,G15.5
*          '/' LABOR REQUIREMENT,MAN-YEARS/UNIT/SHIFT')
WRITE (IOUT,580) ROEQPL(4)
580 FORMAT (' '/' ROEQPL(4) (GENERIC - ALL)'
*          '/' OFFSITE TRANSPORTATION AND DISPOSAL',T50,
*          '/' *G15.5 EQUIPMENT LIFE, TONNES/UNIT')
WRITE (IOUT,590) ROEQPC(4)
590 FORMAT (' '/' ROEQPC(4) (GENERIC - ALL)'
*          '/' OFFSITE TRANSPORTATION AND',T50,G15.5
*          '/' DISPOSAL EQUIPMENT CAPACITY, '
*          '/' TONNES/UNIT/SHIFT/YR')
WRITE (IOUT,600) ROMATU(4)
600 FORMAT (' '/' ROMATU(4) (GENERIC - ALL)'
*          '/' OFFSITE TRANSPORTATION AND',T50,G15.5

```

```

*          /'          DISPOSAL MATERIAL REQUIREMENT, '
*          /'          MATERIAL UNIT/TONNE')
WRITE (IOUT,610) ROCST(4)
610 FORMAT (' ' /'          ROCST(4)          (GENERIC - ALL)'
*          /'          OFFSITE TRANSPORTATION AND',T50,G15.5
*          /'          DISPOSAL CAPITAL COST, $')
WRITE (IOUT,620) ROLBRC(4)
620 FORMAT (' ' /'          ROLBRC(4)          (GENERIC - ALL)'
*          /'          OFFSITE TRANSPORTATION AND',T50,G15.5
*          /'          DISPOSAL UNIT LABOR COST, $/MAN-YEAR')
WRITE (IOUT,630) ROECST(4)
630 FORMAT (' ' /'          ROECST(4)          (GENERIC - ALL)'
*          /'          OFFSITE TRANSPORTATION AND',T50,G15.5
*          /'          DISPOSAL EQUIPMENT COST, $/UNIT')
WRITE (IOUT,640) ROMATC(4)
640 FORMAT (' ' /'          ROMATC(4)          (GENERIC - ALL)'
*          /'          OFFSITE TRANSPORTATION AND',T50,G15.5
*          /'          DISPOSAL MATERIAL COST, MATERIAL UNIT',
*          '/TONNE')
650 CONTINUE

```

C
C
C
C
C6. BACKFILLING

```

WRITE (IOUT,660)
660 FORMAT (' ' // '          BACKFILLING'/5X,11('='))
WRITE (IOUT,670) PCTFIL
670 FORMAT (' ' /'          PCTFIL          (GENERIC - ALL)'
*          /'          PERCENTAGE OF EACH ROOM',T50,G15.5
*          /'          BACKFILLED')
WRITE (IOUT,680) XPANFC
680 FORMAT (' ' /'          XPANFC          (GENERIC - ALL)'
*          /'          EXPANSION IN ROCK VOLUME',T50,G15.5
*          /'          AFTER MINING, 1+FRACTION')
WRITE (IOUT,690) OTHBFL
690 FORMAT (' ' /'          OTHBFL          (GENERIC - ALL)'
*          /'          PERCENTAGE OF BACKFILL VOLUME',T50,G15.5
*          /'          COMPRISED OF A SUPPLEMENTARY MATERIAL'
*          ', FRACTION')
WRITE (IOUT,700) OTBDEN
700 FORMAT (' ' /'          OTBDEN          (GENERIC - ALL)'
*          /'          DENSITY OF SUPPLEMENTARY',T50,G15.5
*          /'          BACKFILL MATERIAL, G/CM**3')
WRITE (IOUT,710) BKRATE
710 FORMAT (' ' /'          BKRATE          (GENERIC - ALL)'
*          /'          BACKFILLING RATE, TONNES/YEAR',T50,G15.5
*          /'          (USES SHAFT CAPACITY IF ZERO)')

```

C
C MSWTCH = 0

```

C      IF (MSWTCH.EQ.1) GO TO 730
      WRITE (IOUT,720) BKFLC
720  FORMAT (' ' /'          BKFLC      (GENERIC - ALL)'
*           /'          UNIT BACKFILLING COST',T50,G15.5
*           /'          $/TONNE')
      GO TO 820

C
C  MSWTCH = 1
C
730  CONTINUE
      WRITE (IOUT,740) ROLBRR(5)
740  FORMAT (' ' /'          ROLBRR(5)      (GENERIC - ALL)'
*           /'          BACKFILLING LABOR REQUIREMENT',T50,G15.5
*           /'          MAN-YEAR/UNIT/SHIFT')
      WRITE (IOUT,750) ROEQPL(5)
750  FORMAT (' ' /'          ROEQPL(5)      (GENERIC - ALL)'
*           /'          BACKFILLING EQUIPMENT LIFE',T50,G15.5
*           /'          TONNES/UNIT')
      WRITE (IOUT,760) ROEQPC(5)
760  FORMAT (' ' /'          ROEQPC(5)      (GENERIC - ALL)'
*           /'          BACKFILLING EQUIPMENT',T50,G15.5
*           /'          CAPACITY, TONNES/UNIT/SHIFT')
      WRITE (IOUT,770) ROMATU(5)
770  FORMAT (' ' /'          ROMATU(5)      (GENERIC - ALL)'
*           /'          BACKFILLING MATERIAL',T50,G15.5
*           /'          REQUIREMENT, MATERIAL UNIT/TONNE')
      WRITE (IOUT,780) ROCST(5)
780  FORMAT (' ' /'          ROCST(5)      (GENERIC - ALL)'
*           /'          BACKFILLING CAPITAL COST, $',T50,G15.5)
      WRITE (IOUT,790) ROLBRC(5)
790  FORMAT (' ' /'          ROLBRC(5)      (GENERIC - ALL)'
*           /'          BACKFILLING UNIT LABOR COST',T50,G15.5
*           /'          $/MAN-YEAR')
      WRITE (IOUT,800) ROECST(5)
800  FORMAT (' ' /'          ROECST(5)      (GENERIC - ALL)'
*           /'          BACKFILLING UNIT EQUIPMENT',T50,G15.5
*           /'          COST, $/UNIT')
      WRITE (IOUT,810) ROMATC(5)
810  FORMAT (' ' /'          ROMATC(5)      (GENERIC - ALL)'
*           /'          BACKFILLING UNIT MATERIAL',T50,G15.5
*           /'          COSTS, $/MATERIAL UNIT')
820  CONTINUE

C      RETURN
      END
      SUBROUTINE OTRPOS
C
      INCLUDE PARAM,LIST

```

```
INCLUDE OPTION,LIST
INCLUDE INPUT,LIST
CHARACTER*1
```

III. REPOSITORY SYSTEM DESCRIPTION

```

      WRITE (IOUT,100)
100  FORMAT (' '//' ***** REPOSITORY SYSTEMS *****' /
+3X,76('=')/3X,76('='))

```

PANEL DESCRIPTION

```

WRITE (IOUT,110)
110  FORMAT (' '/'/ '      PANEL PARAMETERS'/5X,16('='))
120  FORMAT (' '//'      WASTE TYPE',I5)
    WRITE (IOUT,130)
130  FORMAT (' '/'/ '      PANLEN          (GENERIC - ALL)'
*      '/'/ '      PANEL LENGTH, METERS')
    DO 400 K=1,NWT
    WRITE(IOUT,410) K,PANLEN(K)
410  FORMAT(' 'T10,'WASTE TYPE ',I3,T50,G15.5)
400  CONTINUE
    WRITE (IOUT,140)
140  FORMAT (' '/'/ '      PANWTH          (GENERIC - ALL)'
*      '/'/ '      PANEL WIDTH, METERS')
    DO 420 K=1,NWT
    WRITE(IOUT,410) K,PANWTH(K)
420  CONTINUE
150  CONTINUE
    WRITE (IOUT,170) GXEF
170  FORMAT (' '/'/ '      GXEF          (GENERIC - ALL)'
*      '/'/ '      GEOMETRIC EXCLUSION FACTOR','T50,G15.5
*      '/'/ '      FRACTION')
    WRITE (IOUT,180) CPA
180  FORMAT (' '/'/ '      CPA          (GENERIC - ALL)'
*      '/'/ '      CENTRAL PILLAR AREA, ACRES',T50,G15.5)
C
    WRITE (IOUT,160)
160  FORMAT (' '/'/ '      NBSPPA(K)      (BY WASTE TYPE)'
*      '/'/ '      NUMBER OF BORE SECTIONS'
*      '/'/ '      PER PANEL')
    DO 165 K=1,NWT
    WRITE(IOUT,415) K,NBSPPA(K)

```

```

415  FORMAT(' ',T10,'WASTE TYPE ',I3,T50,I10)
165  CONTINUE
C
C
C      ROOM DESCRIPTIONS
C
      WRITE (IOUT,330)
330  FORMAT (' '//'      ROOM PARAMETERS  (BY WASTE TYPE)'/
*    5X,32('=')//)
      WRITE (IOUT,340)
340  FORMAT (' '//'      RMLN(K)    (BY WASTE TYPE)'
*          '/'      ROOM LENGTH,METERS')
      DO 345 K=1,NWT
      WRITE(IOUT,410) K,RMLN(K)
345  CONTINUE
      WRITE (IOUT,350)
350  FORMAT (' '//'      RMWTH(K)    (BY WASTE TYPE)'
*          '/'      ROOM WIDTH, METERS')
      DO 355 K=1,NWT
      WRITE(IOUT,410) K,RMWTH(K)
355  CONTINUE
      WRITE (IOUT,360)
360  FORMAT (' '//'      RMHMIN(K)    (BY WASTE TYPE)'
*          '/'      MINIMUM ROOM HEIGHT, METERS')
      DO 365 K=1,NWT
      WRITE(IOUT,410) K,RMHMIN(K)
365  CONTINUE
      WRITE (IOUT,380)
380  FORMAT (' '//'      DELRHT(K)    (BY WASTE TYPE)'
*          '/'      INCREMENTAL ROOM HEIGHT'
*          '/'      OVER PACKAGE HEIGHT (FOR VERTICAL'
*          ' BOREHOLES)')
      DO 385 K=1,NWT
      WRITE(IOUT,410) K,DELRHT(K)
385  CONTINUE
      WRITE (IOUT,390)
390  FORMAT (' '//'      XRASHO(K)    (BY WASTE TYPE)'
*          '/'      EXTRACTION RATIO, METERS'
*          '/'      (ROOM WIDTH)(#ROOMS)/(PANEL LENGTH)')
C    *          '/'      (RMWTH * NBSPPA) / PANLEN')
      DO 395 K=1,NWT
      WRITE(IOUT,410) K,XRASHO(K)
395  CONTINUE
C
C      CORRIDOR DESCRIPTIONS
C
      WRITE (IOUT,190)
190  FORMAT (' '//'      CORRIDOR PARAMETERS'//5X,19('='))
C

```

```

C MAIN CORRIDOR
C
  WRITE (IOUT,200) CORWTH(1)
200  FORMAT (' ' / ' CORWTH(1)'
*      / ' MAIN CORRIDOR WIDTH, METERS',T50,G15.5)
  WRITE (IOUT,210) CORLEN(1,1)
210  FORMAT (' ' / ' CORLEN(1,1)'
*      / ' MAIN CORRIDOR LENGTH, METERS',T50,G15.5)
  WRITE (IOUT,220) CORHT(1)
220  FORMAT (' ' / ' CORHT(1)'
*      / ' MAIN CORRIDOR HEIGHT, METERS',T50,G15.5)
  WRITE (IOUT,230) TCPCLN
230  FORMAT (' ' / ' TCPCLN'
*      / ' TOTAL CENTRAL PILLAR CORRIDOR',T50,G15.5
*      / ' LENGTH, METERS')
  NACOR=NCOR-1
  WRITE (IOUT,240) NACOR
240  FORMAT (' ' / ' NCOR-1'
*      / ' NUMBER OF ADDITIONAL CORRIDORS = ',
*      T50,I10)
  IF (NCOR.LE.1) GO TO 320
  WRITE (IOUT,250)
250  FORMAT (' ' / ' OTHER CORRIDOR DESCRIPTIONS'/T9,27('-'))
  DO 310 JJ=2,NCOR
  J=JJ-1
  WRITE (IOUT,260) J,CORDSC(JJ)
260  FORMAT (' ' / ' CORDSC(J) (ADDITIONAL CORRIDOR',I3,')'
*      / ' DESCRIPTION (LIMIT 48 CHARACTERS)'/
*      ,T15,A48)
  WRITE (IOUT,270) J,CORWTH(JJ)
270  FORMAT (' ' / ' CORWTH(J) (ADDITIONAL CORRIDOR',I3,')'
*      / ' OTHER CORRIDOR WIDTH, METERS',T50,G15.5)
  WRITE (IOUT,280) J
280  FORMAT (' ' / ' CORLEN(J,K) (ADDITIONAL CORRIDOR',I3,
*      / ' , BY WASTE TYPE)'
*      / ' OTHER CORRIDOR LENGTH, METERS')
  DO 282 K=1,NWT
  WRITE(IOUT,450) K,CORLEN(JJ,K)
450  FORMAT(' ',T10,'WASTE TYPE ',I3,T50,G15.5)
282  CONTINUE
  WRITE (IOUT,290) J,CORHT(JJ)
290  FORMAT (' ' / ' CORHT(J) (ADDITIONAL CORRIDOR',I3,')'
*      / ' OTHER CORRIDOR HEIGHT, METERS',T50,G15.5)
  WRITE (IOUT,300) J,NCORPP(JJ)
300  FORMAT (' ' / ' NCORPP(J) (ADDITIONAL CORRIDOR',I3,')'
*      / ' NUMBER CORRIDORS PER PANEL',T50,I10)
310  CONTINUE
320  CONTINUE
C

```

C

```

RETURN
END
SUBROUTINE OTSFT
INCLUDE PARAM
INCLUDE OPTION
INCLUDE INPUT
CHARACTER*1
+   BLANK
DATA BLANK /1H /

```

C

C

C

E. SHAFT DESCRIPTIONS AND COSTS

```

WRITE (IOUT,100)
100 FORMAT (' '///' ***** SHAFT DESCRIPTIONS AND COSTS *****'
* /3X,76('=')/3X,76('='))

```

C

C

C

C

B.1 MEN AND MATERIALS AND ROCK

```

WRITE (IOUT,110)
110 FORMAT (' '///' MEN & MATERIALS SHAFT'/5X,21('='))
WRITE (IOUT,120)
120 FORMAT (' '///' SHAFT TYPE',T38,'NUMBER OF SHAFTS',
* /T11,10('-'),T38,16('-'))
WRITE (IOUT,130) SFTNUM(1)
130 FORMAT (' '///' SFTNUM(1)'
* /' MEN & MATERIALS ',T40,I10)
WRITE (IOUT,140) SFTDD(1)
140 FORMAT (' '///' SFTDD(1)'
* /' DIAMETER, METERS',T50,G15.5)
WRITE (IOUT,150) SFTXR(1)
150 FORMAT (' '///' SFTXR(1)'
* /' INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
* /' METERS')
WRITE (IOUT,160) SFTXD(1)
160 FORMAT (' '///' SFTXD(1)'
* /' EXCAVATED DEPTH, METERS',T50,G15.5)
WRITE (IOUT,170) SFTLD(1)
170 FORMAT (' '///' SFTLD(1)'
* /' LINED DEPTH, METERS',T50,G15.5)
WRITE (IOUT,180) MXMMRH
180 FORMAT (' '///' MXMMRH'
* /' MAXIMUM TRIPS PER SHIFT FOR',T50,I10
* /' ROCK HOISTING FUNCTIONS')
WRITE (IOUT,190) TNPT2
190 FORMAT (' '///' TNPT2'
* /' ROCK HOISTING CAPACITY,',T50,G15.5
* /' TONNES PER TRIP')

```

```

        WRITE (IOUT,200) SFTHE(1)
200  FORMAT (' '/' SFTHE(1)'
*          /' SHAFT HEADFRAME AND CAPITAL ',T50,G15.5
*          /' EQUIPMENT COST, $')
        WRITE (IOUT,210) SFTSS(1)
210  FORMAT (' '/' SFTSS(1)'
*          /' MEN & MATERIALS SHAFT STATION',T50,G15.5
*          /' COST, $')
220  CONTINUE
C
C B2      REMOTE HANDLED WASTE SHAFT
C
C
        WRITE (IOUT,240)
240  FORMAT (' '/' REMOTE - HANDLED WASTE SHAFT'/5X,28('='))
        WRITE (IOUT,250)
250  FORMAT (' '/' ENTER THE NUMBER OF SHAFTS OF THE FOLLOWING TYPES'
+/' (ENTERING A -1 FOR WASTE-SHAFTS ALLOWS RECON TO'/' CALCULATE
+THE NUMBER NEEDED. IF CONTACT-HANDLED'/' WASTE IS PRESENT AND 0
+CONTACT-HANDLED WASTE '/' SHAFTS ARE SPECIFIED, IT WILL BE TRANSP
+ORTED IN'/' THE REMOTE-HANDLED WASTE SHAFT.))'
        WRITE (IOUT,120)
        WRITE (IOUT,260) SFTNUM(3)
260  FORMAT (' '/' SFTNUM(3)'
*          /' R-H WASTE ',T40,I10)
        WRITE (IOUT,270) SFTDD(3)
270  FORMAT (' '/' SFTDD(3)'
*          /' DIAMETER, METERS',T50,G15.5)
        WRITE (IOUT,280) SFTXR(3)
280  FORMAT (' '/' SFTXR(3)'
*          /' INCREMENTAL EXCAVATED RADIUS (METERS)',
+T50,G15.5)
        WRITE (IOUT,290) SFTXD(3)
290  FORMAT (' '/' SFTXD(3)'
*          /' EXCAVATED DEPTH, METERS',T50,G15.5)
        WRITE (IOUT,300) SFTLD(3)
300  FORMAT (' '/' SFTLD(3)'
*          /' LINED DEPTH, METERS',T50,G15.5)
        WRITE (IOUT,310) MTRPSD
310  FORMAT (' '/' MTRPSD'
*          /' MAXIMUM TRIPS PER SHAFT FOR R-H WASTE '
*          ,T50,I10)
        DO 330 K=1,NWT
        IF (KIND(K).EQ.2) GO TO 330
        WRITE (IOUT,320) NPKPTR(K),K
320  FORMAT (' '/' NPKPTR(K) (BY R-H WASTE TYPE)'
*          /' PACKAGES PER TRIP',T50,I11
*          /' FOR R-H WASTE TYPE ',I5)

```

C


```

330 CONTINUE
340 CONTINUE
    WRITE (IOUT,350) SFTHE(3)
350 FORMAT (' ' /' ' SFTHE(3) '
*           /'          SHAFT HEADFRAME AND CAPITAL',T50,G15.5
*           /'          EQUIPMENT COST, $')
    WRITE (IOUT,360) SFTSS(3)
360 FORMAT (' ' /' ' SFTSS(3) '
*           /'          -R-H WASTE SHAFT STATION',T50,G15.5
*           /'          COST, $')
370 CONTINUE
C
C B3 CONTACT HANDLED WASTE SHAFT
C
C
    IFLG=0
    DO 390 K=1,NWT
        IF (KIND(K).EQ.2) IFLG=1
390 CONTINUE
    WRITE (IOUT,400)
400 FORMAT (' ' /' ' CONTACT = HANDLED WASTE SHAFT'/5X,29('='))
    WRITE (IOUT,120)
    WRITE (IOUT,410) SFTNUM(4)
410 FORMAT (' ' /' ' SFTNUM(4) '
*           /'          C-H WASTE ',T40,I10)
    IF (IFLG.EQ.0) GO TO 510
    WRITE (IOUT,420) SFTDD(4)
420 FORMAT (' ' /' ' SFTDD(4) '
*           /'          DIAMETER, METERS',T50,G15.5)
    WRITE (IOUT,430) SFTXR(4)
430 FORMAT (' ' /' ' SFTXR(4) '
*           /'          INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
*           /'          METERS')
    WRITE (IOUT,440) SFTXD(4)
440 FORMAT (' ' /' ' SFTXD(4) '
*           /'          EXCAVATED DEPTH, METERS',T50,G15.5)
    WRITE (IOUT,450) SFTLD(4)
450 FORMAT (' ' /' ' SFTLD(4) '
*           /'          LINED DEPTH, METERS',T50,G15.5)
    WRITE (IOUT,460) MTCPSD
460 FORMAT (' ' /' ' MTCPSD '
*           /'          MAXIMUM TRIPS PER SHIFT',T50,I10)
    DO 480 K=1,NWT
        IF (KIND(K).NE.2) GO TO 480
        WRITE (IOUT,470) NPKPTC(K),K
470 FORMAT (' ' /' ' NPKPTC(K) (BY C-H WASTE TYPE) '
*           /'          PACKAGES PER TRIP',T50,I10
*           /'          FOR C-H WASTE TYPE',I4)
480 CONTINUE

```

```

      WRITE (IOUT,490) SFTHE(4)
490 FORMAT (' ' /'          SFTHE(4)'
*           /'          SHAFT HEADFRAME & CAPITAL',T50,G15.5
*           /'          EQUIPMENT COST, $')
      WRITE (IOUT,500) SFTSS(4)
500 FORMAT (' ' /'          SFTSS(4)'
*           /'          C-H WASTE SHAFT STATION',T50,G15.5
*           /'          COST, $')
      GO TO 550
510 CONTINUE
      DO 530 K=1,NWT
        IF (KIND(K).NE.2) GO TO 530
        WRITE (IOUT,520) NPKPTC(K),K
520  FORMAT (' ' /'          NPKPTC(K) (BY WASTE TYPE)'
*           /'          PACKAGES PER TRIP',T50,I10
+           /'          FOR C-H WASTE TYPE',I4,
*           /'          IN REMOTE HANDLED WASTE SHAFT')
530 CONTINUE
540 CONTINUE
550 CONTINUE

```

```

C
C  B4  MINE VENTILATION SHAFT
C
C

```

```

      WRITE (IOUT,560)
560 FORMAT (' ' /'          MINE VENTILATION SHAFT'/5X,22('='))
      WRITE (IOUT,120)
      WRITE (IOUT,570) SFTNUM(2)
570 FORMAT (' ' /'          SFTNUM(2)'
*           /'          MINE VENTILATION',T40,I10)

```

```

C  B4  MINE VENTILATION SHAFT
C

```

```

      WRITE (IOUT,580) SFTDD(2)
580 FORMAT (' ' /'          SFTDD(2)'
*           /'          DIAMETER, METERS',T50,G15.5)
      WRITE (IOUT,590) SFTXR(2)
590 FORMAT (' ' /'          SFTXR(2)'
*           /'          INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
*           /'          METERS')
      WRITE (IOUT,600) SFTXD(2)
600 FORMAT (' ' /'          SFTXD(2)'
*           /'          EXCAVATED DEPTH, METERS',T50,G15.5)
      WRITE (IOUT,610) SFTLD(2)
610 FORMAT (' ' /'          SFTLD(2)'
*           /'          LINED DEPTH, METERS',T50,G1
+5.5)
      WRITE (IOUT,620) SFTHE(2)
620 FORMAT (' ' /'          SFTHE(2)'
*           /'          SHAFT HEADFRAME & CAPITAL',T50,G15.5

```

```

      *          /'      EQUIPMENT COST, $')
      WRITE (IOUT,630) SFTSS(2)
630  FORMAT (' ' /'      SFTSS(2)')
      *          /'
      *          /'      MINING VENTILATION SHAFT',T50,G15.5
      *          /'      STATION EQUIPMENT COST, $')

```

```

640  CONTINUE

```

C
C
C
C

```

B5  MINING EXHAUST SHAFT

```

```

      WRITE (IOUT,660)
660  FORMAT (' ' /'      MINING EXHAUST SHAFT'/5X,20('='))
      WRITE (IOUT,120)
      WRITE (IOUT,670) SFTNUM(5)
670  FORMAT (' ' /'      SFTNUM(5)')
      *          /'
      *          /'      MINING EXHAUST  ',T40,I10)
      WRITE (IOUT,680) SFTDD(5)
680  FORMAT (' ' /'      SFTDD(5)')
      *          /'
      *          /'      DIAMETER, METERS',T50,G15.5
      +)
      WRITE (IOUT,690) SFTXR(5)
690  FORMAT (' ' /'      SFTXR(5)')
      *          /'
      *          /'      INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
      *          /'      METERS')
      WRITE (IOUT,700) SFTXD(5)
700  FORMAT (' ' /'      SFTXD(5)')
      *          /'
      *          /'      EXCAVATED DEPTH, METERS',T50,G15.5)
      WRITE (IOUT,710) SFTLD(5)
710  FORMAT (' ' /'      SFTLD(5)')
      *          /'
      *          /'      LINED DEPTH, METERS',T50,G15.5)
      WRITE (IOUT,720) SFTHE(5)
720  FORMAT (' ' /'      SFTHE(5)')
      *          /'
      *          /'      SHAFT HEADFRAME & CAPITAL',T50,G15.5
      *          /'      EQUIPMENT COST, $ ')
      WRITE (IOUT,730) SFTSS(5)
730  FORMAT (' ' /'      SFTSS(5)')
      *          /'
      *          /'      MINING EXHAUST SHAFT',T50,G15.5
      *          /'      STATION EQUIPMENT COST, $')

```

```

740  CONTINUE

```

C
C
C
C

```

B6  WASTE EXHAUST SHAFT

```

```

      WRITE (IOUT,760)
760  FORMAT (' ' /'      WASTE EXHAUST SHAFT'/5X,19('='))
      WRITE (IOUT,250)
      WRITE (IOUT,120)
      WRITE (IOUT,770) SFTNUM(6)
770  FORMAT (' ' /'      SFTNUM(6)')

```

```

      *          /'          WASTE EXHAUST  ',T40,I10)
      WRITE (IOUT,780) SFTDD(6)
780  FORMAT (' ' /'          SFTDD(6) '
      *          /'          DIAMETER, METERS',T50,G15.5)
      WRITE (IOUT,790) SFTXR(6)
790  FORMAT (' ' /'          SFTXR(6) '
      *          /'          INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
      *          /'          METERS')
      WRITE (IOUT,800) SFTXD(6)
800  FORMAT (' ' /'          SFTXD(6) '
      *          /'          EXCAVATED DEPTH, METERS',T50,G15.5)
      WRITE (IOUT,810) SFTLD(6)
810  FORMAT (' ' /'          SFTLD(6) '
      *          /'          LINED DEPTH, METERS',T50,G15.5)
      WRITE (IOUT,820) SFTHE(6)
820  FORMAT (' ' /'          SFTHE(6) '
      *          /'          SHAFT HEADFRAME & CAPITAL',T50,G15.5
      *          /'          EQUIPMENT COST')
      WRITE (IOUT,830) SFTSS(6)
830  FORMAT (' ' /'          SFTSS(6) '
      *          /'          WASTE EXHAUST SHAFT',T50,G15.5
      *          /'          STATION EQUIPMENT COST, $')
840  CONTINUE

```

C
C
C
C

B7 ROCK HOISTING SHAFT

```

      WRITE (IOUT,860)
860  FORMAT (' ' /'          ROCK HOISTING SHAFT'/5X,19('='))
      WRITE (IOUT,120)
      WRITE (IOUT,870) SFTNUM(7)
870  FORMAT (' ' /'          SFTNUM(7) '
      *          /'          ROCK HOISTING ONLY  ',T40,
      +I10)
      WRITE (IOUT,880) SFTDD(7)
880  FORMAT (' ' /'          SFTDD(7) '
      *          /'          DIAMETER, METERS',T50,G15.5)
      WRITE (IOUT,890) SFTXR(7)
890  FORMAT (' ' /'          SFTXR(7) '
      *          /'          INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
      *          /'          METERS')
      WRITE (IOUT,900) SFTXD(7)
900  FORMAT (' ' /'          SFTXD(7) '
      *          /'          EXCAVATED DEPTH, METERS',T50,G15.5)
      WRITE (IOUT,910) SFTLD(7)
910  FORMAT (' ' /'          SFTLD(7) '
      *          /'          LINED DEPTH, METERS',T50,G15.5)
      WRITE (IOUT,920) MXMPRH
920  FORMAT (' ' /'          MXMPRH '

```

```

*          /'          MAXIMUM TRIPS PER SHAFT FOR',T50,I10
*          /'          ROCK HOISTING (MINE PRODUCTION)')
WRITE (IOUT,930) TNPT1
930 FORMAT (' ' /'          TNPT1'
*          /'          ROCK HOISTING CAPACITY ',T50,G15.5
*          /'          TONNES PER TRIP')
WRITE (IOUT,940) SFTHE(7)
940 FORMAT (' ' /'          SFTHE(7)'
*          /'          SHAFT HEADFRAME AND CAPITAL',T50,G15.5
*          /'          EQUIPMENT COST, $')
WRITE (IOUT,950) SFTSS(7)
950 FORMAT (' ' /'          SFTSS(7)'
*          /'          ROCK HOISTING SHAFT',T50,G15.5
*          /'          STATION EQUIPMENT COST, $')
960 CONTINUE
C
C B8 SHAFT COST FUNCTION COEFFICIENTS
C
WRITE (IOUT,980)
980 FORMAT (' ' // '          SHAFT COST FUNCTIONS'/5X,20('='))
IF (IEQTP1.EQ.1) WRITE (IOUT,990)
990 FORMAT (' ' /'          IEQTP1'
*          /'          SINKING COST FUNCTION FORM IS',T50,'LINEAR')
IF (IEQTP1.EQ.2) WRITE (IOUT,1005)
1005 FORMAT (' ' /'          IEQTP1'
*          /'          SINKING COST FUNCTION FORM IS',T50,
*          ' EXPONENTIAL')
WRITE (IOUT,1010)
1010 FORMAT (' ' /'          TO CHANGE ENTRY ENTER THE CORRESPONDING INTEGER
+ '          /'          1 = LINEAR'/'          2 = EXPONENTIAL')
IF (IEQTP1.EQ.1) WRITE (IOUT,1020)
1020 FORMAT (' ' /'          FUNCTION = A + B * D'
*          /'          WHERE D = SFTDD(L)+2*SFTXR(L)'
*          /'          WHERE L=1,NST')
IF (IEQTP1.EQ.2) WRITE (IOUT,1030)
1030 FORMAT (' ' /'          FUNCTION = A * EXP ( B * D )'
*          /'          WHERE D = SFTDD(L)+2*SFTXR(L)'
*          /'          WHERE L=1,NST')
WRITE (IOUT,1040) SCON1
1040 FORMAT (' ' /'          SCON1'
*          /'          SINKING COST COEFFICIENT A ',G15.5)
WRITE (IOUT,1050) SCON2
1050 FORMAT (' ' /'          SCON2'
*          /'          SINKING COST COEFFICIENT B ',G15.5)
IF (IEQTP2.EQ.1) WRITE (IOUT,1060)
1060 FORMAT (' ' /'          IEQTP2'
*          /'          LINING COST FUNCTIONAL FORM IS ',T50,'LINEAR
* ')
IF (IEQTP2.EQ.2) WRITE (IOUT,1070)

```

```

1070 FORMAT (' '//' IEQTP2'
*          '/' LINING COST FUNCTIONAL FORM IS ',T50,
* 'EXPONENTIAL')
WRITE (IOUT,1010)
IF (IEQTP2.EQ.1) WRITE (IOUT,1020)
IF (IEQTP2.EQ.2) WRITE (IOUT,1030)
WRITE (IOUT,1080) CCCON1
1080 FORMAT (' '//' CCCON1'
*          '/' LINING COST COEFFICIENT A ',G15.5)
WRITE (IOUT,1090) CCCON2
1090 FORMAT (' '//' CCCON2'
*          '/' LINING COST COEFFICIENT B ',G15.5)
IF (IEQTP3.EQ.1) WRITE (IOUT,1100) IEQTP3
1100 FORMAT (' '//' IEQTP3'
*          '/' WATER CONTROL COST FUNCTIONAL FORM IS '
* ,T50,'LINEAR')
IF (IEQTP3.EQ.2) WRITE (IOUT,1110)
1110 FORMAT (' '//' IEQTP3'
*          '/' WATER CONTROL COST FUNCTIONAL FORM IS '
* ,T50,'EXPONENTIAL')
WRITE (IOUT,1010)
IF (IEQTP3.EQ.1) WRITE (IOUT,1020)
IF (IEQTP3.EQ.2) WRITE (IOUT,1030)
WRITE (IOUT,1120) WCCON1
1120 FORMAT (' '//' WCCON1'
*          '/' WATER CONTROL COST COEFFICIENT A ',G15.5)
WRITE (IOUT,1130) WCCON2
1130 FORMAT (' '//' WCCON2'
*          '/' WATER CONTROL COST COEFFICIENT B ',G15.5)
C
RETURN
END
SUBROUTINE OTSUPT
INCLUDE PARAM
INCLUDE OPTION
INCLUDE INPUT
CHARACTER*1
+ BLANK
DATA BLANK/1H /
C D. SUPPORT FACILITIES DESCRIPTIONS
C
WRITE (IOUT,100)
100 FORMAT (' '//' ***** SUPPORT FACILITIES *****'/
+3X,76('=')/3X,76('='))
WRITE (IOUT,110) SOA
110 FORMAT (' '//' SOA'
*          '/' SURFACE OPERATIONS AREA, ACRES',T50,G15.5)
WRITE (IOUT,120) NSC
120 FORMAT (' '//' NSC (LIMIT 50)'

```

```

*          '/'          NUMBER OF SUPPORT FACILITY DESCRIPTIONS',
+,T50,I5)
DO 220 N=1,NSC
WRITE(IOUT,1000) N
1000 FORMAT(' '/T8,'SUPPORT FACILITY NO. ',I5/T8,26('-'))/
WRITE (IOUT,130) N,SFTITL(N)
130 FORMAT (' '/' SFTITL(N) (FACILITY',I3,')'
*          '/'          SUPPORT FACILITY NAME, (LIMIT 48'
*' CHARACTERS)'/,T5,A48)
WRITE (IOUT,140) N,SFLBRR(N)
140 FORMAT (' '/' SFLBRR(N) (FACILITY',I3,')'
*          '/'          SUPPORT FACILITY LABOR',T50,G15.5
*          '/'          REQUIREMENT, MAN YEARS/FACILITY/SHIFT')
WRITE (IOUT,150) N,SLBR(N)
150 FORMAT (' '/' SLBR(N) (FACILITY',I3,')'
*          '/'          NUMBER OF SHIFTS PER FACILITY',T50,G15.5
*          '/'          (IF = -1 THE R-H OPERATIONS SHIFT'
*          , ' REQUIREMENTS USED)'
*          '/'          (IF = -2 THE C-H OPERATIONS SHIFT',
*          ' REQUIREMENTS USED)')
WRITE (IOUT,160) N,SFEQPL(N)
160 FORMAT (' '/' SFEQPL(N) (FACILITY',I3,')'
*          '/'          SUPPORT FACILITY EQUIPMENT',T50,G15.5
*          '/'          LIFE, YEARS')
WRITE (IOUT,170) N,SFCST(N)
170 FORMAT (' '/' SFCST(N) (FACILITY',I3,')'
*          '/'          SUPPORT FACILITY CAPITAL',T50,G15.5
*          '/'          COST, $')
WRITE (IOUT,180) N,SFLBRC(N)
180 FORMAT (' '/' SFLBRC(N) (FACILITY',I3,')'
*          '/'          SUPPORT FACILITY UNIT LABOR',T50,G15.5
*          '/'          COST, $/MAN-YEAR')
WRITE (IOUT,190) N,SFECST(N)
190 FORMAT (' '/' SFECST(N) (FACILITY',I3,')'
*          '/'          SUPPORT FACILITY UNIT',T50,G15.5
*          '/'          EQUIPMENT COST, $/FACILITY')
WRITE (IOUT,200) N,SFMATU(N)
200 FORMAT (' '/' SFMATU(N) (FACILITY',I3,')'
*          '/'          SUPPORT FACILITY MATERIAL',T50,G15.5
*          '/'          REQUIREMENTS, MATERIAL COSTS/FACILITY')
WRITE (IOUT,210) N,SFMATC(N)
210 FORMAT (' '/' SFMATC(N) (FACILITY',I3,')'
*          '/'          SUPPORT FACILITY MATERIAL',T50,G15.5
*          '/'          COST, $/MATERIAL UNIT')
220 CONTINUE
C
RETURN
END
SUBROUTINE OTUTLT

```

```

C
C
    INCLUDE PARAM
    INCLUDE INPUT
    INCLUDE OPTION
    CHARACTER*1
    +    BLANK
    DATA BLANK /1H /
C E. UTILITIES DESCRIPTION
C
C
C E1. ELECTRIC POWER REQUIREMENTS
C
    WRITE (IOUT,100)
100 FORMAT (' '// ' ***** UTILITY PARAMETERS AND COSTS *****' /
    +3X,76('=')/3X,76('='))
    WRITE (IOUT,110)
110 FORMAT (' '// ' ELECTRICAL'/5X,10('='))
    WRITE (IOUT,120) RAPP
120 FORMAT (' '// ' RAPP (GENERIC - ALL)'
    *      /' ELECTRICAL LOAD-RECEIVING, ',T50,G15.5
    *      /' AND PACKAGING, KW/MODULE')
    WRITE (IOUT,130) SFSP
130 FORMAT (' '// ' SFSP (GENERIC - ALL)'
    *      /' ELECTRICAL LOAD-LAG STORAGE',T50,G15.5
    *      /' POOL, KW')
    WRITE (IOUT,140) OWHP
140 FORMAT (' '// ' OWHP (GENERIC - ALL)'
    *      /' ELECTRICAL LOAD-WASTE',T50,G15.5
    +      /' HANDLING BUILDING, KW')
    WRITE (IOUT,150) SFP
150 FORMAT (' '// ' SFP (GENERIC - ALL)'
    *      /' ELECTRICAL LOAD-SUPPORT',T50,G15.5
    *      /' FACILITIES, KW')
    WRITE (IOUT,160) MROP
160 FORMAT (' '// ' MROP (REAL NUMBER) (GENERIC - ALL)'
    *      /' ELECTRICAL LOAD-ROCK',T50,G15.5
    *      /' OPERATIONS, KW/TONNE/YEAR')
    WRITE (IOUT,170) RVP
170 FORMAT (' '// ' RVP (GENERIC - ALL)'
    *      /' ELECTRICAL LOAD-MINE',T50,G15.5
    *      /' VENTILATION, KW/ACRE')
    WRITE (IOUT,180) ACP
180 FORMAT (' '// ' ACP (GENERIC - ALL)'
    *      /' ELECTRICAL LOAD-AIR',T50,G15.5
    *      /' CONDITIONING, KW')
    WRITE (IOUT,190) UPC
190 FORMAT (' '// ' UPC (GENERIC - ALL)'
    *      /' ELECTRICITY UNIT COST, ',T50,G15.5

```



```

      *          /'          MILLS/KWH')
C
C E2. STEAM REQUIREMENTS
C
      WRITE (IOUT,200)
200 FORMAT (' ' /'          STEAM'/5X,5('='))
      WRITE (IOUT,210) HEATS
210 FORMAT (' ' /'          HEATS          (GENERIC - ALL)'
*          /'          STEAM REQUIREMENTS-UNDERGROUND',T50,G15.5
*          /'          HEATING, 10**3 LB/ACRE')
      WRITE (IOUT,220) RAPS
220 FORMAT (' ' /'          RAPS          (GENERIC - ALL)'
*          /'          STEAM REQUIREMENTS-RECEIVING',T50,G15.5
*          /'          FACILITY HEATING, 10**3 LB/MODULE')
      WRITE (IOUT,230) ORS
230 FORMAT (' ' /'          ORS          (GENERIC - ALL)'
*          /'          STEAM REQUIREMENT-OTHER',T50,G15.5
*          /'          FACILITIES, 10**3 LB')
      WRITE (IOUT,240) STMC
240 FORMAT (' ' /'          STMC          (GENERIC - ALL)'
*          /'          STEAM UNIT COST,',T50,G15.5
*          /'          $/10**3 LB')
C
C E3. WATER REQUIREMENTS
C
      WRITE (IOUT,250)
250 FORMAT (' ' /'          POTABLE WATER'/5X,13('='))
      WRITE (IOUT,260) VOLW
260 FORMAT (' ' /'          VOLW          (GENERIC - ALL)'
*          /'          POTABLE WATER REQUIREMENT,',T50,G15.5
*          /'          10**6 LITERS')
      WRITE (IOUT,270) WTRC
270 FORMAT (' ' /'          WTRC          (GENERIC - ALL)'
*          /'          POTABLE WATER UNIT COST',T50,G15.5
*          /'          $/10**6 LITERS')
C
C DEMINERALIZED WATER
C
      WRITE (IOUT,275)
275 FORMAT (' ' /'          DEMINERALIZED WATER'/5X,19('='))
C
      WRITE (IOUT,280) VOLDW
280 FORMAT (' ' /'          VOLDW          (GENERIC - ALL)'
*          /'          DEMINERALIZED WATER REQUIREMENT',T50,G15.5
*          /'          10**6 LITERS')
      WRITE (IOUT,290) DWTRC
290 FORMAT (' ' /'          DWTRC          (GENERIC - ALL)'

```

```

      *          /'          DEMINERALIZED WATER UNIT',T50,G15.5
      *          /'          COST, $/10**6 LITERS')
C
C
C E4. DIESEL FUEL REQUIREMENTS
C
      WRITE (IOUT,300)
300 FORMAT (' ' /'          DIESEL FUEL' /5X,11('='))
      WRITE (IOUT,310) SUD
310 FORMAT (' ' /'          SUD          (GENERIC - ALL)'
      *          /'          DIESEL FUEL REQUIREMENT',T50,G15.5
      *          /'          SURFACE,LITERS/DAY')
      WRITE (IOUT,320) UGD
320 FORMAT (' ' /'          UGD          (GENERIC - ALL)'
      *          /'          DIESEL FUEL REQUIREMENTS-',T50,G15.5
      *          /'          UNDERGROUND, LITERS/WASTE UNIT')
      WRITE (IOUT,330) RED
330 FORMAT (' ' /'          RED          (GENERIC - ALL)'
      *          /'          DIESEL FUEL REQUIREMENTS-RAIL',T50,G15.5
      *          /'          USE, LITERS/SHIPPING CONTAINER')
      WRITE (IOUT,340) DSLC
340 FORMAT (' ' /'          DSLC          (GENERIC - ALL)'
      *          /'          DIESEL FUEL UNIT COST,',T50,G15.5
      *          /'          $/LITER')
C
      RETURN
      END
C
      SUBROUTINE OTWAST
C
      INCLUDE PARAM
      INCLUDE OPTION
      INCLUDE INPUT
      CHARACTER*1
      +      BLANK
      DATA BLANK /1H /
C
C II. WASTE HANDLING SYSTEMS DESCRIPTION
C
C A. GENERAL WASTE DESCRIPTION
C
      MCNT=0
      IRFLG = 0
      ICFLG = 0
      WRITE (IOUT,100)
100 FORMAT (' ' //10X,' ***** WASTE HANDLING SYSTEMS *****' /
      +3X,76('=')/3X,76('='))
C

```

C

```

DO 2290 K=1,NWT
  WRITE (IOUT,110) K
110 FORMAT (' '//' GENERAL WASTE DESCRIPTION (WASTE TYPE'
  +,I3,')'/5X,42('='))
  WRITE (IOUT,130) K,WSTYP(K)
130 FORMAT (' '//' WSTYP(K) (WASTE TYPE',I3,')'
  *      '/' DESCRIPTION: ',T50,A12)
  WRITE (IOUT,140) K,KIND(K)
140 FORMAT (' '//' KIND(K) (WASTE TYPE',I3,')'
  *      '/' WASTE TYPE DESCRIPTION CODE = ',I5,' , WHERE'
  *      '/' 1 = REMOTE HANDLED'
  *      '/' 2 = CONTACT HANDLED'
  *      '/' 3 = TRU, REMOTE HANDLED')
  IF (KIND(K).EQ.1.OR.KIND(K).EQ.3) IRFLG=IRFLG+1
  IF (KIND(K).EQ.2) ICFLG=ICFLG+1
  WRITE (IOUT,150) K,UNTMAS(K)
150 FORMAT (' '//' UNTMAS(K) (WASTE TYPE',I3,')'
  *      '/' UNIT MASS, KGU ',T50,G15.5)
  WRITE (IOUT,670) K,WVOLPU(K)
670 FORMAT (' '//' WVOLPU(K) (WASTE TYPE',I3,')'
  *      '/' WASTE VOLUME PER UNIT, M3',T50,G15.5)
  WRITE (IOUT,170) K
170 FORMAT (' '//' TDWI(I,K) (WASTE TYPE',I3,' , BY YEAR)'
  *      '/' THERMAL DENSITY, KW/WASTE UNIT'/)
  L=LASTYR-IYRCON+1
  DO 190 I=1,L
  IYR=IYRCON+I-1
  WRITE (IOUT,180) IYR,TDWI(I,K)
180 FORMAT (' ',T8,'YEAR ',I5,3X,G15.5)
190 CONTINUE

```

C
C
C
C
C
C

B. WASTE RECEIVING DESCRIPTION

```

  WRITE (IOUT,200) K
200 FORMAT (' '//' WASTE RECEIVING PARAMETERS AND COSTS',
  + ' (WASTE TYPE',I3,')'/5X,52('='))
  WRITE (IOUT,210)
210 FORMAT (' '//' PARAMETERS'/6X,10('-'))
  IF (KIND(K).EQ.2) GO TO 240
  IF (IRFLG.GT.1) GO TO 270
  WRITE (IOUT,220) NRHRMA
220 FORMAT (' '//' NRHRMA (GENERIC: R-H WASTE)'
  *      '/' MAXIMUM NUMBER OF R-H WASTE RECEIVING',
  * T50,I10 '/' MODULES ALLOWED ')
  WRITE (IOUT,230) NRBPRM
230 FORMAT (' '//' NRBPRM (GENERIC: R-H WASTE)'

```

```

*          /'          NUMBER OF RECEIVING BAYS PER MODULE',
*      T50,I10)
GO TO 270
240 CONTINUE
    IF (ICFLG.GT.1) GO TO 270
    WRITE (IOUT,250) NCHRMA
250 FORMAT (' '/'          NCHRMA          (GENERIC: C-H WASTE)'
*          /'          MAXIMUM NUMBER OF C-H WASTE RECEIVING',
*      T50,I10 /'          MODULES ALLOWED ')
    WRITE (IOUT,260) NCBPRM
260 FORMAT (' '/'          NCBPRM          (GENERIC: C-H WASTE)'
*          /'          NUMBER OF RECEIVING BAYS PER MODULE',
*      T50,I10)
270 CONTINUE
    WRITE (IOUT,280)
280 FORMAT (' '/'          AVERAGE RECEIVING TURNAROUND TIME IN '
* 'SHIFTS FOR')
    WRITE (IOUT,290) K,TTTR(K)
290 FORMAT (' '/'          TTTR(K)          (WASTE TYPE',I3,')'
*          /'          - RAIL CAR',T50,G15.5)
    WRITE (IOUT,300) K,TTTK(K)
300 FORMAT (' '/'          TTTK(K)          (WASTE TYPE',I3,')'
*          /'          - TRUCK 'T50,G15.5)
    WRITE (IOUT,310) K,RCCAP(K)
310 FORMAT (' '/'          RCCAP(K)          (WASTE TYPE',I3,')'
*          /'          RAIL CAR CAPACITY,',T50,G15.5
*          /'          WASTE UNITS / RAIL CAR')
    WRITE (IOUT,320) K,TCCAP(K)
320 FORMAT (' '/'          TCCAP(K)          (WASTE TYPE',I3,')'
*          /'          TRUCK CAPACITY,',T50,G15.5
*          /'          WASTE UNITS / TRUCK')
    WRITE (IOUT,330) K
330 FORMAT (' '/'          CRRAIL(K,I)          (WASTE TYPE',I3,',' , BY YEAR)'
*          /'          PERCENT OF WASTE UNITS SHIPPED BY RAIL,'
* ' FRACTION')
    II=LASTYR-IYRCON+1
    DO 350 I=1,II
        IYR=IYRCON+I-1
        WRITE (IOUT,180) IYR,CRRAIL(K,I)
350 CONTINUE
    IF (KIND(K).EQ.2) GO TO 410

C
C REMOTE HANDLED
C
    IF (IRFLG.GT.1) GO TO 390
    WRITE (IOUT,360) WOLBRR(1)
360 FORMAT (' '/'          WOLBRR(1)          (GENERIC: R-H WASTE)'
*          /'          WASTE RECEIVING LABOR REQUIREMENT ',
*      T50,G15.5 /'          MAN YEARS/SHIFT/MODULE')

```

```

        WRITE (IOUT,370) WOEQPL(1)
370 FORMAT (' '//, WOEQPL(1) (GENERIC: R-H WASTE)'
*           /' WASTE RECEIVING EQUIPMENT LIFE',T50,G15.3
*           /' WASTE UNITS ')
        WRITE (IOUT,380) WOEQPC(1)
380 FORMAT (' '//, WOEQPC(1) (GENERIC: R-H WASTE)'
*           /' WASTE RECEIVING EQUIPMENT CAPACITY',T50,
* G15.5, /' WASTE UNITS/EQUIPMENT UNIT/SHIFT')
390 CONTINUE
        WRITE (IOUT,400) K,WOMATU(1,K)
400 FORMAT (' '//, WOMATU(1,K) (R-H WASTE TYPE ('I2,') )'
*           /' WASTE RECEIVING MATERIAL REQUIREMENT,'
* T50,G15.5 /' MATERIAL UNIT/WASTE UNIT')
        GO TO 470
C
C CONTACT HANDLED
C
410 CONTINUE
        IF (ICFLG.GT.1) GO TO 450
        WRITE (IOUT,420) WOLBRR(2)
420 FORMAT (' '//, WOLBRR(2) (GENERIC: C-H WASTE)'
*           /' C-H RECEIVING UNIT LABOR REQUIREMENT,'
* T50,G15.5 /' MAN YEARS/SHIFT/MODULE')
        WRITE (IOUT,430) WOEQPL(2)
430 FORMAT (' '//, WOEQPL(2) (GENERIC: C-H WASTE)'
*           /' C-H RECEIVING UNIT EQUIPMENT LIFE,',
* T50,G15.5 /' RECEIVING UNITS')
        WRITE (IOUT,440) WOEQPC(2)
440 FORMAT (' '//, WOEQPC(2) (GENERIC: C-H WASTE)'
*           /' C-H RECEIVING EQUIPMENT CAPACITY,'
* T50,G15.5 /' RECEIVING UNITS/EQUIPMENT UNIT/SHIFT')
450 CONTINUE
        WRITE (IOUT,460) K,WOMATU(2,K)
460 FORMAT (' '//, WOMATU(2,K) (C-H WASTE TYPE ('I2,') )'
*           /' C-H MATERIALS REQUIREMENT, 'T50,G15.5
*           /' MATERIALS UNIT/RECEIVING UNIT')
        GO TO 540
C
C C. WASTE RECEIVING COST DATA
C
C REMOTE HANDLED
C
470 CONTINUE
        WRITE (IOUT,480)
480 FORMAT (' '//, COSTS'/6X,5('-'))
        IF (IRFLG.GT.1) GO TO 520
        WRITE (IOUT,490) WOCST(1)
490 FORMAT (' '//, WOCST(1) (GENERIC: R-H WASTE)'
*           /' R-H RECEIVING MODULE CAPITAL COST, 'T50,

```

```

      * G15.3      /'      $/MODULE')
      WRITE (IOUT,500) WOLBRC(1)
500 FORMAT (' ' /'      WOLBRC(1)      (GENERIC: R-H WASTE)'
      *      /'      R-H RECEIVING MODULE LABOR COST, ',T50,
      *G15.3      /'      $/MAN-YEAR')
      WRITE (IOUT,510) WOECST(1)
510 FORMAT (' ' /'      WOECST(1)      (GENERIC: R-H WASTE)'
      *      /'      R-H RECEIVING MODULE EQUIPMENT ',T50,
      * G15.3      /'      COSTS, $/EQUIPMENT UNIT')
520 CONTINUE
      WRITE (IOUT,530) K,WOMATC(1,K)
530 FORMAT (' ' /'      WOMATC(1,K) (R-H WASTE TYPE (' ,I2,' ) )'
      *      /'      R-H RECEIVING MODULE MATERIALS ',T50,
      * G15.5      /'      COSTS, $/MATERIAL UNIT')
      GO TO 600

C
C CONTACT HANDLED
C
540 CONTINUE
      WRITE (IOUT,480)
      IF (ICFLG.GT.1) GO TO 580
      WRITE (IOUT,550) WOCST(2)
550 FORMAT (' ' /'      WOCST(2)      (GENERIC: C-H WASTE)'
      *      /'      C-H RECEIVING MODULE CAPITAL COST, ',T50,
      * G15.5,      /'      $/MODULE')
      WRITE (IOUT,560) WOLBRC(2)
560 FORMAT (' ' /'      WOLBRC(2)      (GENERIC: C-H WASTE)'
      *      /'      C-H RECEIVING MODULE LABOR COST, ',T50,
      *G15.3      /'      $/MAN-YEAR')
      WRITE (IOUT,570) WOECST(2)
570 FORMAT (' ' /'      WOECST(2)      (GENERIC: C-H WASTE)'
      *      /'      C-H RECEIVING MODULE EQUIPMENT COST, ',
      * T50,G15.5, /'      $/EQUIPMENT UNIT')
580 CONTINUE
      WRITE (IOUT,590) K,WOMATC(2,K)
590 FORMAT (' ' /'      WOMATC(2,K) (C-H WASTE TYPE (' ,I2,' ) )'
      *      /'      C-H RECEIVING MODULE MATERIALS ',T50,
      * G15.5      /'      COSTS, $/MATERIAL UNIT')
600 CONTINUE

C
C
C
C D. WASTE PACKAGING DESCRIPTION
C
      WRITE (IOUT,610) K
610 FORMAT (' ' /'      WASTE PACKAGING PARAMETERS AND COSTS',
      + ' (WASTE TYPE',I3,' )' /5X,53('='))
      WRITE (IOUT,620)
620 FORMAT (' ' /'      PARAMETERS' /6X,10('-'))

```

```

        WRITE (IOUT,630) K,PKGDS1(K)
630 FORMAT (' ','/' PKGDS1(K) (WASTE TYPE',I3,')'
*          '/' PACKAGE DESCRIPTION,',T50,A16
*          '/' OUTPUT LINE ONE (LIMIT 16 CHARACTERS)')
        WRITE (IOUT,640) K,PKGDS2(K)
640 FORMAT (' ','/' PKGDS2(K) (WASTE TYPE',I3,')'
*          '/' PACKAGE DESCRIPTION,',T50,A16
*          '/' OUTPUT LINE TWO (LIMIT 16 CHARACTERS)')
        WRITE (IOUT,650) K,PKGDS3(K)
650 FORMAT (' ','/' PKGDS3(K) (WASTE TYPE',I3,')'
*          '/' PACKAGE DESCRIPTION,',T50,A16
*          '/' OUTPUT LINE THREE (LIMIT 16 CHARACTERS)')
        WRITE (IOUT,660) K,WUPP(K)
660 FORMAT (' ','/' WUPP(K) (WASTE TYPE',I3,')'
*          '/' NUMBER OF WASTE UNITS PER PACKAGE ',T50,
*          G15.5)
        IF (KIND(K).EQ.2) GOTO 735
        WRITE (IOUT,680) K,PKGCAP(K)
680 FORMAT (' ','/' PKGCAP(K) (R-H WASTE TYPE',I3,')'
*          '/' WASTE MODULE PACKAGING CAPACITY',T50,
*          G15.5 '/' PACKAGE/SHIFT')
        IF (IRFLG.GT.1) GO TO 720
        WRITE (IOUT,690) WOLBRR(3)
690 FORMAT (' ','/' WOLBRR(3) (GENERIC: R-H)'
*          '/' WASTE PACKAGING LABOR REQUIREMENT,',T50,
*          G15.5 '/' MAN-YEARS/SHIFT/MODULE')
        WRITE (IOUT,700) WOEQPL(3)
700 FORMAT (' ','/' WOEQPL(3) (GENERIC: R-H)'
*          '/' WASTE PACKAGING EQUIPMENT LIFE,',T50,
*          G15.5 '/' PACKAGING UNITS')
        WRITE (IOUT,710) WOEQPC(3)
710 FORMAT (' ','/' WOEQPC(3) (GENERIC: R-H)'
*          '/' WASTE PACKAGING EQUIPMENT CAPACITY,',T50,
*          G15.5 '/' PACKAGING UNITS/EQUIPMENT UNIT/SHIFT')
720 CONTINUE
        WRITE (IOUT,730) K,WOMATU(3,K)
730 FORMAT (' ','/' WOMATU(3,K) (R-H WASTE TYPE',I3,')'
*          '/' WASTE PACKAGING MATERIAL REQUIREMENTS,',
*          T50,G15.5 '/' MATERIAL UNITS/PACKAGE')
735 CONTINUE
        WRITE (IOUT,740) K,PKGDIAM(K)
740 FORMAT (' ','/' PKGDIAM(K) (WASTE TYPE',I3,')'
*          '/' PACKAGE DIAMETER, METERS',T50,G15.5)
        WRITE (IOUT,750) K,REFDIA(K)
750 FORMAT (' ','/' REFDIA(K) (WASTE TYPE',I3,')'
*          '/' REFERENCE PACKAGE DIAMETER USED FOR',T50,
*          G15.5 '/' PACKAGE AND HOLE COST, METERS')
        WRITE (IOUT,760) K,PKGLN(K)
760 FORMAT (' ','/' PKGLN(K) (WASTE TYPE',I3,')'

```

```

      *          /'          PACKAGE LENGTH, METERS',T50,G15.5)
C
C E. WASTE PACKAGING COST DATA
C
      IF (KIND(K).EQ.2) GOTO 825
      WRITE (IOUT,770)
770 FORMAT (' '///'          COSTS'/6X,5('-'))
      IF (IRFLG.GT.1) GO TO 810
      WRITE (IOUT,780) WOCST(3)
780 FORMAT (' '///'          WOCST(3)          (GENERIC: R-H)'
      *          /'          WASTE PACKAGING MODULE CAPITAL COST,',T50,
      * G15.3          /'          $/MODULE')
      WRITE (IOUT,790) WOLBRC(3)
790 FORMAT (' '///'          WOLBRC(3)          (GENERIC: R-H)'
      *          /'          WASTE PACKAGING MODULE LABOR COST,',T50,
      * G15.3          /'          $/MAN-YEAR')
      WRITE (IOUT,800) WOECST(3)
800 FORMAT (' '///'          WOECST(3)          (GENERIC: R-H)'
      *          /'          WASTE PACKAGING MODULE EQUIPMENT COST,'
      * ,T50,G15.3 /'          $/EQUIP UNIT')
810 CONTINUE
      WRITE (IOUT,820) K,WOMATC(3,K)
820 FORMAT (' '///'          WOMATC(3,K)          (R-H WASTE TYPE',I3,')'
      *          /'          WASTE PACKAGING MODULE MATERIAL COST,'
      * ,T50,G15.5, /'          $/MATERIAL UNIT')
C
825 CONTINUE
C
C F TRANSFER OPERATIONS DESCRIPTION
C
      WRITE (IOUT,830) K
830 FORMAT (' '///'          WASTE TRANSFER PARAMETERS AND COSTS',
      + ' (WASTE TYPE',I3,')'/5X,52('='))
      IF (KIND(K).EQ.2) GO TO 890
      IF (IRFLG.GT.1) GO TO 870
C
C REMOTE HANDLED
C
      WRITE (IOUT,840)
840 FORMAT (' '///'          PARAMETERS'/6X,10('-'))
      WRITE (IOUT,850) TOEQPL(1)
850 FORMAT (' '///'          TOEQPL(1)          (GENERIC: R-H WASTE)'
      *          /'          R-H TRANSPORTATION UNIT,',T50,G15.5
      *          /'          EQUIPMENT LIFE, PACKAGES')
      WRITE (IOUT,860) TOEQPC(1)
860 FORMAT (' '///'          TOEQPC(1)          (GENERIC: R-H WASTE)'
      *          /'          R-H TRANSPORTATION UNIT',T50,G15.5
      *          /'          EQUIPMENT CAPACITY,'

```



```

      * ' PACKAGES/EQUIPMENT-UNIT/SHIFT')
      WRITE (IOUT,950) TOLBRR(1)
950  FORMAT (' '//' TOLBRR(1) (GENERIC: R-H WASTE)'
      *           '/' R-H TRANSPORTATION UNIT',T50,G15.5
      *           '/' LABOR REQUIREMENT, MAN-YEARS/EQUIPMENT UNIT
      +')
870  CONTINUE
      WRITE (IOUT,880) K,TOMATU(1,K)
880  FORMAT (' '//' TOMATU(1,K) (R-H WASTE TYPE (' ,I2,') )'
      *           '/' R-H TRANSPORTATION MATERIAL',T50,G15.5
      *           '/' REQUIREMENT MATERIAL UNIT/PACKAGE')
C
      GO TO 960
C CONTACT HANDLED
C
890  CONTINUE
      IF (ICFLG.GT.1) GO TO 920
      WRITE (IOUT,900) TOEQPL(2)
900  FORMAT (' '//' TOEQPL(2) (GENERIC: C-H WASTE)'
      *           '/' C-H TRANSPORTATION UNIT',T50,G15.5
      *           '/' EQUIPMENT LIFE, PACKAGES')
      WRITE (IOUT,910) TOEQPC(2)
910  FORMAT (' '//' TOEQPC(2) (GENERIC: C-H WASTE)'
      *           '/' C-H TRANSPORTATION UNIT',T50,G15.5
      *           '/' EQUIPMENT CAPACITY, PACKAGES/EQUIPMENT',
      *           '/' UNIT/SHIFT')
      WRITE (IOUT,970) TOLBRR(2)
970  FORMAT (' '//' TOLBRR(2) (GENERIC: C-H WASTE)'
      *           '/' C-H TRANSPORTATION UNIT',T50,G15.5
      *           '/' LABOR REQUIREMENT,MAN-YEARS/EQUIPMENT UNIT'
      + )
920  CONTINUE
      WRITE (IOUT,930) K,TOMATU(2,K)
930  FORMAT (' '//' TOMATU(2,K) (C-H WASTE TYPE (' ,I2,') )'
      *           '/' C-H TRANSPORTATION UNIT',T50,G15.5
      *           '/' MATERIAL REQUIREMENT, MATERIAL UNITS'
      *           '/' /PACKAGE')
960  CONTINUE
C
C G. TRANSFER OPERATIONS COST DATA
C
C REMOTE HANDLED
C
980  CONTINUE
      WRITE (IOUT,990)
      IF (KIND(K).EQ.2) GOTO 1050
990  FORMAT (' '//' COSTS' /6X,5('-'))
      IF (IRFLG.GT.1) GO TO 1030

```

```

      WRITE (IOUT,1000) TOCST(1)
1000 FORMAT (' '// TOCST(1) (GENERIC: R-H WASTE)'
*           /' R-H WASTE TRANSFER CAPITAL COST,',T50,
* G15.5 /' $ ')
      WRITE (IOUT,1010) TOLBRC(1)
1010 FORMAT (' '// TOLBRC(1) (GENERIC: R-H WASTE)'
*           /' R-H WASTE TRANSFER UNIT',T50,G15.5
*           /' LABOR COST, $/MAN YEAR')
      WRITE (IOUT,1020) TOECST(1)
1020 FORMAT (' '// TOECST(1) (GENERIC: R-H WASTE)'
*           /' R-H WASTE TRANSFER UNIT',T50,G15.5
*           /' EQUIPMENT COST, $/EQUIPMENT UNIT')
1030 CONTINUE
      WRITE (IOUT,1040) K,TOMATC(1,K)
1040 FORMAT (' '// TOMATC(1,K) (R-H WASTE TYPE (' ,I2,' ) )'
*           /' R-H WASTE TRANSFER MATERIALS',T50,G15.5
*           /' COST, $')
      GO TO 1110
1050 CONTINUE
C
C CONTACT HANDLED
C
      WRITE (IOUT,830)
      IF (ICFLG.GT.1) GO TO 1090
      WRITE (IOUT,1060) TOCST(2)
1060 FORMAT (' '// TOCST(2) (GENERIC: C-H WASTE)'
*           /' C-H WASTE TRANSPORTATION',T50,G15.5
*           /' CAPITAL COST, $')
      WRITE (IOUT,1070) TOLBRC(2)
1070 FORMAT (' '// TOLBRC(2) (GENERIC: C-H WASTE)'
*           /' C-H WASTE TRANSFER UNIT',T50,G15.5
*           /' LABOR COST, $/MAY-YEAR')
      WRITE (IOUT,1080) TOECST(2)
1080 FORMAT (' '// TOECST(2) (GENERIC: C-H WASTE)'
*           /' C-H EMPLACEMENT UNIT',T50,G15.5
*           /' EQUIPMENT COST, $/EQUIPMENT UNIT')
1090 CONTINUE
      WRITE (IOUT,1100) K,TOMATC(2,K)
1100 FORMAT (' '// TOMATC(2,K) (C-H WASTE TYPE (' ,I2,' ) )'
*           /' C-H WASTE TRANSFER UNIT',T50,G15.5
*           /' MATERIALS COST, $/MATERIAL UNIT')
C
C
C
C H. EMPLACEMENT OPERATIONS DESCRIPTION
C
1110 CONTINUE
      WRITE (IOUT,1120) K
1120 FORMAT (' '// WASTE EMPLACEMENT PARAMETERS AND COSTS',

```

```

+ ' (WASTE TYPE',I3,')'
+ / (NOTE: PRIMARY WASTE TYPES MUST BE FIRST)'
+ /5X,54('-'))
WRITE (IOUT,1130)
1130 FORMAT (' ' / ' PARAMETERS' /6X,10('-'))
WRITE(IOUT,1152) K
1152 FORMAT(' ' / ' IMET(K) (WASTE TYPE',I3,')')
WRITE(IOUT,1150) IMET(K)
1150 FORMAT(' ' / ' EMPLACEMENT METHOD OPTION = ',I3,', WHERE' /
+ ' PRIMARY SECONDARY '
+ ' TERTIARY' /
+ ' 1 = VERTICAL BOREHOLE NONE '
+ ' NONE' /
+ ' 2 = ROOM EMPLACEMENT NONE '
+ ' NONE' /
+ ' 3 = TRENCHING NONE '
+ ' NONE' /
+ ' 4 = HORIZONTAL BOREHOLE NONE '
+ ' NONE' /
+ ' 5 = VERTICAL BOREHOLE TRU IN VBH'
+ ' NONE' /
+ ' 6 = VERTICAL BOREHOLE TRU IN VBH'
+ ' CH IN RE' /
+ ' 7 = VERTICAL BOREHOLE CH IN RE '
+ ' NONE' /
+ ' 8 = HORIZONTAL BOREHOLE TRU IN HBH'
+ ' NONE' /
+ ' 9 = HORIZONTAL BOREHOLE TRU IN HBH'
+ ' CH IN RE')
WRITE(IOUT,1154)
1154 FORMAT(' 10 = HORIZONTAL BOREHOLE CH IN RE '
+ ' NONE' /
+ ' 11 = HORIZONTAL BOREHOLE TRU IN T '
+ ' NONE' /
+ ' 12 = HORIZONTAL BOREHOLE TRU IN T '
+ ' CH IN RE' /
+ ' 13 = ROOM EMPLACEMENT TRU IN RE '
+ ' NONE' /
+ ' 14 = ROOM EMPLACEMENT CH IN RE '
+ ' NONE' /
+ ' 15 = ROOM EMPLACEMENT TRU IN RE '
+ ' CH IN RE' /)
IF (K.GT.1.AND.MCNT.GE.1) GO TO 1200
MCNT=MCNT+1
WRITE (IOUT,1170) MPRIME
1170 FORMAT (' ' / ' MPRIME (GENERIC: ALL)'
* / ' NUMBER OF PRIMARY HEAT GENERATING',T50,I10
* / ' WASTE TYPES')
WRITE (IOUT,1180) NNHG

```

```

1180 FORMAT (' '/' NNHG (GENERIC: ALL)'
*          '/' NUMBER OF NON-HEAT GENERATING',T50,I10
*          '/' WASTE TYPES')
      WRITE (IOUT,1190) ICLR
1190 FORMAT (' '/' ICLR (GENERIC: ALL)'
*          '/' THE YEAR INVENTORIES OF STORED NON-HEAT',
* T50,I10 '/' GENERATING WASTE CAN BE CLEARED BY SOLO',
*          '/' EMPLACEMENT')
1200 CONTINUE
      IF (NNHG.EQ.0) GO TO 1300
      WRITE (IOUT,1210) STOMAX(K)
1210 FORMAT (' '/' STOMAX(K) (BY SECONDARY WASTE TYPE)'
*          '/' THE STORAGE CAPACITY LIMIT FOR',T50,
* G15.5 '/' SECONDARY WASTE (PACKAGES)')
      WRITE (IOUT,1220)
1220 FORMAT (' '/' NUMBER OF NON-HEAT GENERATING WASTE'
*          '/' PACKAGES THAT CAN BE COEMPLACED PER'
*          '/' PRIMARY WASTE PACKAGE AND FRACTION'
*          '/' SECONDARY WASTE EMPLACEMENT CAPACITY'
*          '/' AVAILABLE FOR THIS SECONDARY WASTE TYPE')
      IF (K.GT.MPRIME) GO TO 1310
      WRITE (IOUT,1230) K
1230 FORMAT (' '/' PRIMARY WASTE TYPE ',I5)
      DO 1260 M=1,NNHG
      MM=MPRIME+M
      WRITE (IOUT,1240) MM
1240 FORMAT (' '/' FOR SECONDARY WASTE TYPE',I5)
      WRITE (IOUT,1250) COCON(M,K),PCTCON(M,K)
1250 FORMAT (' '/' COCON(MM,K) (BY SECONDARY WASTE BY'
* ' PRIMARY WASTE TYPE)'
*          '/' NUMBER OF PACKAGES PER',T50,G15.5
*          '/' PRIMARY PACKAGE'
*          '/' PCTCON(MM,K) (BY SECONDARY WASTE BY '
* 'PRIMARY WASTE TYPE'
*          '/' FRACTION OF SECONDARY CAPACITY',T50,G15.5)
1260 CONTINUE
      WRITE (IOUT,1270)
1270 FORMAT (' '/' SWPPR(M) (BY SECONDARY WASTE)')
      DO 1280 J=MPRIME+1,NWT
      WRITE(IOUT,1272) J,SWPPR(J)
1272 FORMAT(' ', '/' SECONDARY WASTE NO.',I5
*          '/' SECONDARY WASTE PACKAGES PER ROOM',T50
* ,G15.5 '/' SOLO EMPLACEMENT')
1280 CONTINUE
      IF (KIND(K).EQ.1.OR.KIND(K).EQ.3) GO TO 1300
      IF (IMET(K).LE.4.OR.IMET(K).EQ.7.OR.IMET(K).EQ.10.OR.IMET(K).GT.
+ 12) GO TO 1310
      WRITE (IOUT,1290) CONTRU(K)
1290 FORMAT (' '/' CONTRU(M) (BY SECONDARY C-H WASTE)'

```

```

*           /'          NUMBER OF C-H PACKAGES THAT',T50,G15.5
*           /'          CAN BE EMPLACED PER TRU WASTE PACKAGE'
*           /'          FOR PRIMARY HG WASTE TYPE',I5)
1300  CONTINUE
1310  CONTINUE
      WRITE (IOUT,1320) K,NPKPEU(K)
1320  FORMAT (' ' /'          NPKPEU(K)          (WASTE TYPE',I3,')'
*           /'          NUMBER OF PACKAGES PER',T50,I10
*           /'          EMPLACEMENT UNIT')
      WRITE (IOUT,1330) K,PKGPPUP(K)
1330  FORMAT (' ' /'          PKGPPUP(K)          (WASTE TYPE',I3,')'
*           /'          EMPLACEMENT UNITS PER UNIT PITCH',T50,G15.5
*           /'          PER ROW')
      IF (KIND(K).EQ.2) GO TO 1350
      IF (IRFLG.GT.1) GO TO 1420
      WRITE (IOUT,1340) UOLBRR(1)
1340  FORMAT (' ' /'          UOLBRR(1)          (GENERIC: R-H WASTE)'
*           /'          R-H WASTE EMPLACEMENT LABOR',T50,G15.5
*           /'          REQUIREMENT, MAN-YEARS/EQUIPMENT UNIT')
      GO TO 1370
1350  CONTINUE
      IF (ICFLG.GT.1) GO TO 1420
      WRITE (IOUT,1360) UOLBRR(2)
1360  FORMAT (' ' /'          UOLBRR(2)          (GENERIC: C-H WASTE)'
*           /'          C-H WASTE EMPLACEMENT LABOR',T50,G15.5
*           /'          REQUIREMENT, MAN-YEARS/EQUIPMENT UNIT')
      GO TO 1400
1370  CONTINUE
      WRITE (IOUT,1380) UOEQPL(1)
1380  FORMAT (' ' /'          UOEQPL(1)          (GENERIC: R-H WASTE)'
*           /'          R-H WASTE EMPLACEMENT',T50,G15.5
*           /'          EQUIPMENT LIFE, PACKAGES')
      WRITE (IOUT,1390) UOEQPC(1)
1390  FORMAT (' ' /'          UOEQPC(1)          (GENERIC: R-H WASTE)'
*           /'          R-H WASTE EMPLACEMENT',T50,G15.5
*           /'          EQUIPMENT CAPACITY, PACKAGES/EQUIPMENT'
*           ' UNIT/SHIFT')
      GO TO 1420
1400  CONTINUE
      WRITE (IOUT,1382) UOEQPL(2)
1382  FORMAT (' ' /'          UOEQPL(2)          (GENERIC: C-H WASTE)'
*           /'          C-H WASTE EMPLACEMENT',T50,G15.5
*           /'          EQUIPMENT LIFE, PACKAGES')
      WRITE (IOUT,1410) UOEQPC(2)
1410  FORMAT (' ' /'          UOEQPC(2)          (GENERIC: C-H WASTE)'
*           /'          C-H WASTE EMPLACEMENT',T50,G15.5
*           /'          EQUIPMENT CAPACITY, PACKAGES/EQUIPMENT '
*           ' UNIT/SHIFT')
1420  CONTINUE

```

```

        WRITE (IOUT,1430) K,CLSEMP(K)
1430 FORMAT (' '//'      CLSEMP(K) (WASTE TYPE',I3,')'
*           //'      CLOSEST EMPLACEMENT TO',T50,G15.5
*           //'      CORRIDOR, METERS')
        WRITE (IOUT,1440) K,NRWSPR(K)
1440 FORMAT (' '//'      NRWSPR(K) (WASTE TYPE',I3,')'
*           //'      NUMBER OF ROWS PER ROOM',T50,I10)
        WRITE (IOUT,1450) K,WABATH(K)
1450 FORMAT (' '//'      WABATH(K) (WASTE TYPE',I3,')'
*           //'      WATER BARRIER MATERIAL',T50,G15.5
*           //'      THICKNESS, METERS')
        WRITE (IOUT,1460) K,SLEETH(K)
1460 FORMAT (' '//'      SLEETH(K) (WASTE TYPE',I3,')'
*           //'      SLEEVE THICKNESS, METERS '
+           ',T50,G15.5)
        WRITE (IOUT,1470) K,BCKFTH(K)
1470 FORMAT (' '//'      BCKFTH(K) (WASTE TYPE',I3,')'
*           //'      BACKFILL MATERIAL THICKNESS,',T50,G15.5
*           //'      METERS')
        WRITE (IOUT,1480) K,HOLDEL(K)
1480 FORMAT (' '//'      HOLDEL(K) (WASTE TYPE',I3,')'
*           //'      INCREMENTAL HOLE DEPTH BEYOND',T50,G15.5
*           //'      PACKAGE LENGTH, METERS')
        WRITE (IOUT,1490) K,CAPLEN(K)
1490 FORMAT (' '//'      CAPLEN(K) (WASTE TYPE',I3,')'
*           //'      LENGTH OF EMPLACEMENT HOLE',T50,G15.5
*           //'      PLUG, METERS')

```

C

C I. EMPLACEMENT OPERATIONS COST DATA

C

```

        WRITE (IOUT,1500)
1500 FORMAT (' '//'      COST'/6X,5('-'))
        IF (KIND(K).EQ.2) GO TO 1520
        IF (IRFLG.GT.1) GO TO 1620
        WRITE (IOUT,1510) UOCST(1)
1510 FORMAT (' '//'      UOCST(1) (GENERIC: R-H WASTE)'
*           //'      R-H WASTE EMPLACEMENT',T50,G15.5
*           //'      CAPITAL COST, $')
        GO TO 1540
1520 CONTINUE
        IF (ICFLG.GT.1) GO TO 1620
        WRITE (IOUT,1530) UOCST(2)
1530 FORMAT (' '//'      UOCST(2) (GENERIC: C-H WASTE)'
*           //'      C-H WASTE EMPLACEMENT CAPITAL',T50,G15.5
*           //'      COST, $')
        GO TO 1560
1540 CONTINUE
        WRITE (IOUT,1550) UOLBRC(1)
1550 FORMAT (' '//'      UOLBRC(1) (GENERIC: R-H WASTE)'

```

```

      *          /'          R-H WASTE EMPLACEMENT UNIT',T50,G15.5
      *          /'          LABOR COST, $/MAN-YEAR')
      GO TO 1580
1560  CONTINUE
      WRITE (IOUT,1570) UOLBRC(2)
1570  FORMAT (' ' /'          UOLBRC(2)          (GENERIC: C-H WASTE)'
      *          /'          C-H WASTE EMPLACEMENT UNIT',T50,G15.5
      *          /'          LABOR COST, $/MAN-YEAR')
      GO TO 1600
1580  CONTINUE
      WRITE (IOUT,1590) UOECST(1)
1590  FORMAT (' ' /'          UOECST(1)          (GENERIC: R-H WASTE)'
      *          /'          R-H WASTE EMPLACEMENT UNIT',T50,G15.5
      *          /'          EQUIPMENT COST, $/EQUIPMENT UNIT')
      GO TO 1620
1600  CONTINUE
      WRITE (IOUT,1610) UOECST(2)
1610  FORMAT (' ' /'          UOECST(2)          (GENERIC: C-H WASTE)'
      *          /'          C-H WASTE EMPLACEMENT UNIT',T50,G15.5
      *          /'          EQUIPMENT COST')
1620  CONTINUE
      IF (K.GT.1) GO TO 1640
      WRITE (IOUT,1630) IDET
1630  FORMAT (' ' /'          IDET          (GENERIC: ALL)'
      *          /'          INDIVIDUAL MATERIAL COST OPTION ='T55,I3,
      *          ' , WHERE'/'          0 = AGGREGATE EMPLACEMENT COSTS'
      *          /'          1 = DETAILED EMPLACEMENT MATERIALS'
      *          /'          DESCRIPTION')
C
C  EMPLACEMENT METHODS
C
1640  CONTINUE
      WRITE (IOUT,1650)
1650  FORMAT (' ' /,5X,'EMPLACEMENT METHODS',/5X,19('-'))
C
C      GO TO (1660,2010,2140,1660,1660,1660,1660,1660,1660,1660,16
C  + 60,2010,2010,2010)IMET(K)
C  IMET = 1
C
1660  CONTINUE
      IF (KIND(K).EQ.2) GO TO 1730
      IF (IDET.EQ.1) GO TO 1760
      WRITE (IOUT,1710) K,UOMATU(1,K)
1710  FORMAT (' ' /'          UOMATU(1,K)          (R-H WASTE TYPE (' ,I2,' )'
      *          /'          R-H WASTE BARRIER MATERIAL',T50,G15.5
      *          /'          UNIT REQUIREMENT, MATERIAL UNIT/EMPLACEMENT
      + UNIT')
      WRITE (IOUT,1720) K,UOMATC(1,K)
1720  FORMAT (' ' /'          UOMATC(1,K)          (R-H WASTE TYPE (' ,I2,' )'

```

```

*           /'          R-H WASTE BARRIER MATERIAL',T50,G15.5
*           /'          UNIT COST, $ MATERIAL UNIT/EMPLACEMENT UNIT
& ')
  GO TO 1760
1730  CONTINUE
      IF (IDET.EQ.1) GO TO 1760
      WRITE (IOUT,1740) K,UOMATU(2,K)
1740  FORMAT (' /'          UOMATU(2,K) (C-H WASTE TYPE (' ,I2,') )'
*           /'          C-H WASTE BARRIER MATERIAL',T50,G15.5
*           /'          UNIT REQUIREMENT, MATERIAL '
* 'UNIT/EMPLACEMENT UNIT')
      WRITE (IOUT,1750) K,UOMATC(2,K)
1750  FORMAT (' /'          UOMATC(2,K) (C-H WASTE TYPE (' ,I2,') )'
*           /'          C-H WASTE BARRIER MATERIAL',T50,G15.5
*           /'          UNIT COST, $ MATERIAL UNIT/EMPLACEMENT UNIT
& ')
1760  CONTINUE
      IF (IDET.NE.1) GOTO 1810
      WRITE (IOUT,1770) K,REFMAT(K)
1770  FORMAT (' /'          REFMAT(K) (WASTE TYPE',I3,')'
*           /'          REFERENCE HOLE LINER MATERIAL',T50,G15.5
*           /'          COST, $/PACKAGES')
      WRITE (IOUT,1780) K,REFSLV(K)
1780  FORMAT (' /'          REFSLV(K) (WASTE TYPE',I3,')'
*           /'          REFERENCE HOLE SLEEVE COST',T50,G15.5
*           /'          $/SLEEVE')
      WRITE (IOUT,1790) K,REFBKF(K)
1790  FORMAT (' /'          REFBKF(K) (WASTE TYPE',I3,')'
*           /'          REFERENCE BACKFILL MATERIAL',T50,G15.5
*           /'          COST, $/HOLE')
      WRITE (IOUT,1800) K,REFDEP(K)
1800  FORMAT (' /'          REFDEP(K) (WASTE TYPE',I3,')'
*           /'          REFERENCE HOLE DEPTH FOR',T50,G15.5
*           /'          REFERENCE COSTS, METERS')
C
C      (ALSO SEE REFERENCE PACKAGE DIAMETER UNDER ITEM II)
C
1810  CONTINUE
C  CALCULATE HOLE DRILLING COSTS USING IUHDCM=1
C
      IF (K.GT.1) GO TO 2010
      WRITE (IOUT,1820) IUHDCM
1820  FORMAT (' /'          IUHDCM (GENERIC: ALL)'
*           /'          CALCULATE HOLE DRILLING COSTS'
*           /'          WHERE OPTION =' ,I5
*           /'          0 = HOLE DRILLING MATERIAL '
* 'REQUIREMENT,'
*           /'          MATERIAL UNIT/BOREHOLE'
*           /'          1 = UNIT HOLE DRILLING COST')

```



```

      IF (IUHDCM.EQ.1) WRITE(IOUT,8000)
8000  FORMAT(' ','/' IF IUHDCM = 1 THEN THE FOLLOWING 5 VALUES'
* ' ARE ZERO. THEY ARE PRINTED OUT FOR CLARITY')
      IF(IUHDCM.EQ.0) EELBRR(1) = 0.0
      WRITE (IOUT,1830) EELBRR(1)
1830  FORMAT (' ','/' EELBRR(1) (GENERIC: ALL)'
*          '/' HOLE DRILLING LABOR',T50,G15.5
*          '/' REQUIREMENT, MAN-YEARS/EQUIPMENT',
* ' UNIT/SHIFT/DAY')
      IF(IUHDCM.EQ.0) EEEQPC(1) = 0.0
      WRITE (IOUT,1840) EEEQPL(1)
1840  FORMAT (' ','/' EEEQPL(1) (GENERIC: ALL)'
*          '/' HOLE DRILLING UNIT EQUIPMENT',T50,G15.5
*          '/' LIFE, EMPLACEMENT UNITS/EQUIPMENT UNIT')
      IF (IUHDCM.EQ.0) EEEQPC(1) = 0.0
      WRITE (IOUT,1850) EEEQPC(1)
1850  FORMAT (' ','/' EEEQPC(1) (GENERIC: ALL)'
*          '/' HOLE DRILLING EMPLACEMENT UNIT',T50,G15.5
*          '/' EQUIPMENT CAPACITY, '
*          '/' EMPLACEMENT UNITS/EQUIPMENT UNIT/SHIFT/DAY')
      IF (IUHDCM.EQ.0) EEMATU(1) = 0.0
      WRITE (IOUT,1860) EEMATU(1)
1860  FORMAT (' ','/' EEMATU(1) (GENERIC: ALL)'
*          '/' HOLE DRILLING MATERIAL',T50,G15.5
*          '/' REQUIREMENT, MATERIAL UNITS/EMPLACEMENT UNIT')
      IF (IUHDCM.EQ.0) EECST(1) = 0.0
      WRITE (IOUT,1870) EECST(1)
1870  FORMAT (' ','/' EECST(1) (GENERIC: ALL)'
*          '/' HOLE DRILLING CAPITAL',T50,G15.5
*          '/' COSTS, '$')
1880  CONTINUE
      WRITE (IOUT,1670) EELBRC(1)
1670  FORMAT (' ','/' EELBRC(1) (GENERIC: ALL)'
*          '/' HOLE DRILLING UNIT LABOR COST',T50,G15.5
*          '/' $/MAN-YEAR')
      WRITE (IOUT,1680) EEECST(1)
1680  FORMAT (' ','/' EEECST(1) (GENERIC: ALL)'
*          '/' HOLE DRILLING UNIT EQUIPMENT',T50,G15.5
*          '/' COST, $/EQUIPMENT UNIT')
      WRITE (IOUT,1690) EEMATC(1)
1690  FORMAT (' ','/' EEMATC(1) (GENERIC: ALL)'
*          '/' HOLE DRILLING UNIT MATERIALS',T50,G15.5
*          '/' COST, $/MATERIAL UNIT')
1700  CONTINUE
      IF (IUHDCM.NE.1) GOTO 2000
1890  CONTINUE
      WRITE (IOUT,1900) HDLCFC
1900  FORMAT(' ','/' PORTION OF HOLE DRILLING UNIT COST, FRACTION
+ '/'

```

```

*          /'          HDLCFC      (GENERIC:  ALL)'
*          /'          - ATTRIBUTED TO LABOR',T50,G15.5)
      WRITE (IOUT,1910) HDECFC
1910 FORMAT (' ' /'          HDECFC      (GENERIC:  ALL)'
*          /'          - ATTRIBUTED TO EQUIPMENT',T50,G15.5)
      WRITE (IOUT,1920) HDMCFC
1920 FORMAT (' ' /'          HDMCFC      (GENERIC:  ALL)'
*          /'          - ATTRIBUTED TO MATERIALS',T50,G15.5)
      IF (NTYP.EQ.1) WRITE (IOUT,1930) NTYP
1930 FORMAT (' ' /'          NTYP      (GENERIC:  ALL)'
*          /'          HOLE DRILLING COST FUNCTION TYPE IS',T50
*          , ' EXPONENTIAL')
      IF (NTYP.EQ.0) WRITE (IOUT,1940)
1940 FORMAT (' ' /'          NTYP      (GENERIC:  ALL)'
*          /'          HOLE DRILLING COST FUNCTION TYPE IS',T50,
*          ' POWER')
      WRITE (IOUT,1950)
1950 FORMAT (' ' /'          TO CHANGE ENTRY ENTER THE CORRESPONDING INTE'
+ , 'GER',          /'          1 = LINEAR'
*          /'          2 = EXPONENTIAL')
      IF (NTYP.EQ.1) WRITE (IOUT,1960)
1960 FORMAT (' ' /'          FUNCTION = A + B * D'
*          /'          WHERE D = HOLDIA(K),K=1,NWT')
      IF (NTYP.EQ.2) WRITE (IOUT,1970)
1970 FORMAT (' ' /'          FUNCTION = A * EXP ( B * D )'
*          /'          WHERE D = HOLDIA(K),K=1,NWT')
      WRITE (IOUT,1980) HDCON1
1980 FORMAT (' ' /'          HDCON1      (GENERIC:  ALL)'
*          /'          HOLE DRILLING COST FUNCTION',T50,G10.5
*          /'          CONSTANT A')
      WRITE (IOUT,1990) HDCON2
1990 FORMAT (' ' /'          HDCON2      (GENERIC:  ALL)'
*          /'          HOLE DRILLING COST FUNCTION',T50,G15.5
*          /'          CONSTANT B')
2000  CONTINUE
C
C  EMPLACEMENT METHOD EXCAVATION - ROOM EMPLACEMENT
C
2010  CONTINUE
      IF (IDET.NE.1) GO TO 2060
      IF (KIND(K).NE.2) GOTO 2060
2040  CONTINUE
      WRITE (IOUT,2050) K,CHMTEM(K)
2050 FORMAT (' ' /'          CHMTEM(K) (C-H WASTE TYPE (' ,I2,' ) )'
*          /'          C-H WASTE MATERIALS COST',T50,G15.5
*          /'          $/UNIT PITCH')
2060  CONTINUE
      IF (KIND(K).NE.2.AND.IDET.NE.1) CHMTEM(K) = 0.0
      IF (K.GT.1) GO TO 2140

```

```

      WRITE (IOUT,2070) EECST(2)
2070 FORMAT (' ','/' EECST(2) (GENERIC: ALL)'
*           /' ROOM EMPLACEMENT CAPITAL',T50,G15.5
*           /' COST, $ ')
      WRITE (IOUT,2080) EELBRR(2)
2080 FORMAT (' ','/' EELBRR(2) (GENERIC: ALL)'
*           /' ROOM EMPLACEMENT LABOR',T50,G15.5
*           /' REQUIREMENT, MAN-YEARS/EMPLACEMENT'
*           , ' UNIT/SHIFT')
      WRITE (IOUT,2090) EELBRC(2)
2090 FORMAT (' ','/' EELBRC(2) (GENERIC: ALL)'
*           /' ROOM EMPLACEMENT LABOR COST, ',T50,G15.5
*           /' $/MAN-YEAR')
      WRITE (IOUT,2100) EEEQPL(2)
2100 FORMAT (' ','/' EEEQPL(2) (GENERIC: ALL)'
*           /' ROOM EMPLACEMENT UNIT',T50,G15.5
*           /' EQUIPMENT LIFE, PACKAGES/EQUIPMENT LIFE')
      WRITE (IOUT,2110) EEEQPC(2)
2110 FORMAT (' ','/' EEEQPC(2) (GENERIC: ALL)'
*           /' ROOM EMPLACEMENT UNIT',T50,G15.5
*           /' EQUIPMENT CAPACITY, PACKAGES EQUIPMENT'
*           ' UNIT/SHIFT')
      WRITE (IOUT,2120) EEECST(2)
2120 FORMAT (' ','/' EEECST(2) (GENERIC: ALL)'
*           /' ROOM EMPLACEMENT UNIT',T50,G15.5
*           /' EQUIPMENT COST, PACKAGES/EQUIPMENT',
*           ' UNIT/SHIFT')
      WRITE (IOUT,2130) EEMATU(2)
2130 FORMAT (' ','/' EEMATU(2) (GENERIC: ALL)'
*           /' ROOM EMPLACEMENT MATERIAL',T50,G15.5
*           /' REQUIREMENT, MATERIAL UNITS/PACKAGES')
      WRITE (IOUT,2030) EEMATC(2)
2030 FORMAT (' ','/' EEMATC(2) (GENERIC: ALL)'
*           /' ROOM EMPLACEMENT MATERIAL',T50,G15.5
*           /' UNIT COST, $/MATERIAL UNIT')

```

C
C
C
C

J.3 EMPLACEMENT METHOD EXCAVATION - TRENCHING

```

2140 CONTINUE
      WRITE (IOUT,2150) K,TRCHWD(K)
2150 FORMAT (' ','/' TRCHWD(K) (WASTE TYPE',I3,')'
*           /' TRENCH WIDTH, METERS',T50,G15.5)
      IF (K.GT.1) GO TO 2240
      WRITE (IOUT,2160) EELBRR(3)
2160 FORMAT (' ','/' EELBRR(3) (GENERIC: ALL)'
*           /' TRENCHING LABOR REQUIREMENT,',T50,G15.5
*           /' MAN-YEARS/EMPLACEMENT UNIT/SHIFT')
      WRITE (IOUT,2170) EEEQPL(3)

```

```

2170 FORMAT (' ','/' EEEQPL(3) (GENERIC: ALL)'
*          /' TRENCHING UNIT EQUIPMENT LIFE, ',T50,G15.5
*          /' PACKAGES/EQUIPMENT UNIT')
      WRITE (IOUT,2180) EEEQPC(3)
2180 FORMAT (' ','/' EEEQPC(3) (GENERIC: ALL)'
*          /' TRENCHING UNIT EQUIPMENT',T50,G15.5
*          /' CAPACITY, PACKAGES/EQUIPMENT UNIT/SHIFT')
      WRITE (IOUT,2190) EEMATU(3)
2190 FORMAT (' ','/' EEMATU(3) (GENERIC: ALL)'
*          /' TRENCHING MATERIAL',T50,G15.5
*          /' REQUIREMENT, MATERIAL UNITS/PACKAGE')
      WRITE (IOUT,2200) EECST(3)
2200 FORMAT (' ','/' EECST(3) (GENERIC: ALL)'
*          /' TRENCHING CAPITAL COST, $',T50,G15.5)
      WRITE (IOUT,2210) EELBRC(3)
2210 FORMAT (' ','/' EELBRC(3) (GENERIC: ALL)'
*          /' TRENCHING UNIT LABOR COST, 'T50,G15.5
*          /' $/MAN-YEAR')
      WRITE (IOUT,2220) EEEEST(3)
2220 FORMAT (' ','/' EEEEST(3) (GENERIC: ALL)'
*          /' TRENCHING UNIT EQUIPMENT',T50,G15.5
*          /' COST, $/EQUIPMENT UNIT')
      IF (IDET.NE.0) GO TO 2240
      WRITE (IOUT,2230) EEMATC(3)
2230 FORMAT (' ','/' EEMATC(3) (GENERIC: ALL)'
*          /' TRENCHING UNIT MATERIAL',T50,G15.5
*          /' COST, $/MATERIAL UNIT')
2240 CONTINUE
      IF (IDET.NE.1) GOTO 2280
2250 CONTINUE
      WRITE (IOUT,2260) K,PKGPRK(K)
2260 FORMAT (' ','/' PKGPRK(K) (WASTE TYPE',I3,')'
*          /' PACKAGES EMPLACED PER RACK',T50,G15.5)
      WRITE (IOUT,2270) K,RCKCST(K)
2270 FORMAT (' ','/' RCKCST(K) (WASTE TYPE',I3,')'
*          /' UNIT RACK COST, $/RACK',T50,G15.5)
2280 CONTINUE
C
2290 CONTINUE
      RETURN
      END
      SUBROUTINE OUTPUT
C
C DIRECTS THE SEQUENCE TO OUTPUT
C
      CALL OTGEN
C
      CALL OTWAST
C

```

```

      CALL OTRPOS
C
      CALL OTSHFT
C
      CALL OTROCK
C
      CALL OTSUPT
C
      CALL OTUTLT
C
      CALL OTPREP
C
      CALL OTDECM
C
      RETURN
      END
      SUBROUTINE RDDECM
      INCLUDE PARAM
      INCLUDE INPUT
      INCLUDE OPTION
      CHARACTER*1
      +   BLANK
      DATA BLANK /1H /
C
C  V.  DECOMMISSIONING AND SURVEILANCE ACTIVITIES
C
9000  FORMAT(A80)
      WRITE (IO,100)
100  FORMAT (' '///' DECOMMISSIONING AND SURVEILANCE',/
+3X,32('=')/,3X,32('='))
      IF (IEQTP5.EQ.1) WRITE (IO,110)
110  FORMAT (' '///' IEQTP5 (GENERIC - ALL)'
*      /' SHAFT SEALING FUNCTIONAL RELATIONSHIP '
*      'IS LINEAR')
      IF (IEQTP5.EQ.2) WRITE (IO,120)
120  FORMAT (' '///' IEQTP5 (GENERIC - ALL)'
*      /' SHAFT SEALING FUNCTION RELATIONSHIP '
*      'IS EXPONENTIAL')
      WRITE (IO,130)
130  FORMAT (' '///' TO CHANGE FUNCTIONAL RELATIONSHIP ENTER THE'
*      ' CORRESPONDING INTEGER'
*      /' 1 = LINEAR'/' 2 = EXPONENTIAL')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,IEQTP5)
      IF (IEQTP5.EQ.1) WRITE (IO,140)
140  FORMAT (' '///' FUNCTION = A + B * D'
*      /' WHERE D = SFTDD(L) + 2*SFTXR(L)'
*      /' WHERE, L=1,NST')
      IF (IEQTP5.EQ.2) WRITE (IO,150)

```

```

150 FORMAT ( ' '/' ' FUNCTION = A * EXP ( B * D ) '
*           /' WHERE D = SFTDD(L) + 2*SFTXR(L) '
*           /' WHERE L=1,NST')
WRITE (IO,160) SSSCON1
160 FORMAT ( ' '/' ' SSSCON1 (GENERIC - ALL) '
*           /' SHAFT SEALING FUNCTION',T50,G15.5
*           /' COEFFICIENT A')
READ(IN,9000) ICARD
CALL DCODER(ICARD,SSCON1)
WRITE (IO,170) SSSCON2
170 FORMAT ( ' '/' ' SSSCON2 (GENERIC - ALL) '
*           /' SHAFT SEALING FUNCTION ',T50,G15.5
*           /' COEFFICIENT B')
READ(IN,9000) ICARD
CALL DCODER(ICARD,SSCON2)
WRITE (IO,180) DCOMCF
180 FORMAT ( ' '/' ' DCOMCF (GENERIC - ALL) '
*           /' SUPPORT FACILITY',T50,G15.5
*           /' DECOMMISSIONING COST, FRACTION OF '
*           /' CONSTRUCTION COST')
READ(IN,9000) ICARD
CALL DCODER(ICARD,DCOMCF)
WRITE (IO,190) SURCST
190 FORMAT ( ' '/' ' SURCST (GENERIC - ALL) '
*           /' ANNUAL SURVEILLANCE COST, $',T50,G15.5)
READ(IN,9000) ICARD
CALL DCODER(ICARD,SURCST)
DO 200 J=1,NDDC
DDCST(J)=0.0
DDLBRJ(J)=0.0
DDLBRJ(J)=0.0
DDEQPC(J)=0.0
DDMATC(J)=0.0
200 CONTINUE
C
C SEALING COSTS
C
SCCST=0.0
SCLBRJ=0.0
SCLBRJ=0.0
SCEQPC=0.0
SCMATU=0.0
SCMATC=0.0
C
C ZERO OUT REMAINING YEARS
C
LL=LASTYR-IYRCON
DO 210 I=LL+1,NC
RMCWS1(I)=0.0

```

```

      RMCWS2(I)=0.0
      RMNWS1(I)=0.0
      RMNWS2(I)=0.0
      SFCS(I)=0.0
      SCSKD(I)=0.0
      CLCDF(I)=0.0
      CLNDF(I)=0.0
      PORMDF(1,I)=0.0
      PORMDF(2,I)=0.0
210  CONTINUE
C
      DO 220 I=IDDYR+1,NC
          DBDF(I)=0.0
          SSDF(I)=0.0
          FDDF(I)=0.0
220  CONTINUE
      LY=LASTYR-IYRCON+1
      DO 240 K=1,NWT
          DO 230 I=LY+1,NC
              WTSUP(K,I)=0.0
              TDWI(I,K)=0.0
              CRRAIL(K,I)=0.0
230  CONTINUE
240  CONTINUE
      RETURN
      END
      SUBROUTINE RDGEN
C
      INCLUDE PARAM
      INCLUDE OPTION
      INCLUDE INPUT
      CHARACTER*1
+      BLANK
      DATA BLANK /1H /
C
9000  FORMAT(A80)
C
C  I.  GENERAL SYSTEMS DATA
C
C  I.  GENERAL SYSTEMS DATA
C
      IF(IOPTA.EQ.1) WRITE(IO,5)
5  FORMAT(' ' / ' ***** GENERAL SYSTEMS DATA *****' , / ,
+3X,76('=') / , 3X,76('='))
C
C  A.  CASE TITLE DESCRIPTION
C

```

```

      IF (IOPTB(1,1).NE.1) GOTO 1000
C
      WRITE(IO,8)
8      FORMAT(' '/'          CASE TITLE DESCRIPTION'/6X,22('='))
      WRITE (IO,100) TITLE
100     FORMAT (' '/'          CASE TITLE DESCRIPTION '
      *      /T8,A80)
      READ(IN,9000) ICARD
      IFLG = 0
      DO 102 I=1,80
      IF(ICARD(I:I).EQ.BLANK) GOTO 102
      IFLG = 1
102     CONTINUE
      IF(IFLG.EQ.0) GOTO 104
      TITLE = ICARD
104     CONTINUE
C
C      B. SITE PARAMETERS
C
1000    IF(IOPTB(2,1).NE.1) GO TO 1005
      WRITE (IO,110)
110     FORMAT (' '/'          SITE PARAMETERS'/4X,15('='))
      WRITE (IO,120) PLACE
120     FORMAT (' '/'          PLACE'
      *      /'          REPOSITORY LOCATION, DESCRIPTION ',T50,A16
      *      /'          (LIMIT 16 CHARACTERS)')
      READ(IN,9000) ICARD
      IFLG = 0
      DO 122 I=1,80
      IF(ICARD(I:I).EQ.BLANK) GOTO 122
      IFLG = 1
122     CONTINUE
      IF(IFLG.EQ.0) GOTO 124
      PLACE(1:16) = ICARD(1:16)
124     CONTINUE
      WRITE (IO,130) RPTYP
130     FORMAT (' '/'          RPTYP'
      *      /'          REPOSITORY TYPE,',T50,A16
      *      /'          (LIMIT 16 CHARACTERS)')
      READ(IN,9000) ICARD
      IFLG = 0
      DO 132 I=1,80
      IF(ICARD(I:I).EQ.BLANK) GOTO 132
      IFLG = 1
132     CONTINUE
      IF(IFLG.EQ.0) GOTO 134
      RPTYP(1:16) = ICARD(1:16)
134     CONTINUE
      WRITE (IO,140) DENSTY

```



```

140 FORMAT (' '/' DENSTY'
*          '/' ROCK DENSITY, G/CM**3',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,DENSTY)
      WRITE (IO,150) SIZE
150 FORMAT (' '/' SIZE'
*          '/' UNDERGROUND REPOSITORY AREA, ACRE ',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SIZE)
      WRITE (IO,160) DEPTH
160 FORMAT (' '/' DEPTH'
*          '/' DEPTH, METERS ',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,DEPTH)
      WRITE (IO,170) TCLAND
170 FORMAT (' '/' TCLAND'
*          '/' LAND COST, $ ',T50,G15.3)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TCLAND)
      WRITE (IO,180) SPUC
180 FORMAT (' '/' SPUC'
*          '/' SITE PREPARATION COST, ',T50,G15.3
*          '/' $/ACRE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SPUC)

C
C   C. CONSTRAINTS
C
1005 IF (IOPTB(3,1).NE.1) GOTO 1010
C
      WRITE (IO,190)
190 FORMAT (' '/' CONSTRAINTS'/5X,11('='))
      WRITE (IO,200)
200 FORMAT (' '/' NFTL (BY WASTE TYPE)'
*          '/' NEAR-FIELD THERMAL LIMIT ',
*          '/' KW/ACRE')
      IF (IWNUM(1).EQ.0) INWT=NWT
      NUM = INUM(INWT)
      DO 201 KNUM=1,INWT
      K=INUM(KNUM)
      WRITE(IO,240) K,NFTL(K)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,NFTL(K))
201 CONTINUE
      WRITE (IO,210) FFTL
210 FORMAT (' '/' FFTL'
*          '/' FAR-FIELD THERMAL LIMIT ',T50,G15.5
*          '/' KW/ACRE')

```



```

      CALL DCODEI(ICARD,IYRPRO)
      WRITE (IO,310) LASTYR
310  FORMAT (' ' /' ' LASTYR'
*      /' ' LAST YEAR OF WASTE RECEIPTS ',T55,I5)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,LASTYR)
      WRITE (IO,320) NBYR
320  FORMAT (' ' /' ' NBYR'
*      /' ' YEARS AFTER PRODUCTION STARTUP THAT',T55,I5
*      /' ' BACKFILLING BEGINS')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,NBYR)
      WRITE (IO,330) IDDYR
330  FORMAT (' ' /' ' IDDYR'
*      /' ' NUMBER OF YEARS OF DECOMMISSIONING',T50,I10)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,IDDYR)

```

C

```

      MM=LASTYR-IYRCON+1
      IF (IWNUM(1).EQ.0) INWT=NWT
      NUM = INUM(INWT)
      DO 370 KNUM=1,INWT
      K=INUM(KNUM)
      WRITE (IO,340)
340  FORMAT (' ' /,T8,'WASTE QUANTITY AVAILABLE FOR PROCESSING'
*      /T9,'UNITS/YEAR')
      WRITE (IO,350) K
350  FORMAT (' ' /,T8,'WSTSUP(K,I) (BY WASTE TYPE AND YEAR)
*      ' /T10,'FOR WASTE TYPE',I5/)
      DO 370 J=1,MM
      J1=IYRCON+J-1
      WRITE (IO,360) J1,WSTSUP(K,J)
360  FORMAT (' ' ,T8,'YEAR:',I5,3X,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WSTSUP(K,J))
370  CONTINUE

```

C

C

D. CONSTRUCTION AND COST EXPENDITURE SCHEDULES

C

C

1020 IF (IOPTB(5,1).NE.1) GOTO 1030

C

LL=IYRPRO-IYRCON

C

```

      WRITE (IO,380)
380  FORMAT (' ' /' ' CONSTRUCTION AND COST DISTRIBUTION SCHEDULES'
*      /5X,44('='))
      WRITE (IO,390)
390  FORMAT (' ' /' ' RMCWS1 (R-H - ALL BY YEAR)'

```

```

*          /'          R-H WASTE RECEIVING MODULE CONSTRUCTION '
*, 'SCHEDULE' /'
*          /'          PRIOR TO OPERATIONS, FRACTION/YEAR')
DO 410 J=1,LL
  J1=IYRCON+J-1
  WRITE (IO,400) J1,RMCWS1(J)
400  FORMAT (' ',T8,'YEAR ',I5,3X,G15.5)
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,RMCWS1(J))
410  CONTINUE
  WRITE (IO,420)
420  FORMAT (' ' /'          RMCWS2          (R-H - ALL BY YEAR)'
*          /'          R-H WASTE RECEIVING MODULE CONSTRUCTION '
*, 'SCHEDULE' /'          DURING OPERATIONS, FRACTION/YEAR')
DO 430 J=1,LL
  J1=IYRCON+J-1
  WRITE (IO,360) J1,RMCWS2(J)
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,RMCWS2(J))
430  CONTINUE
  WRITE (IO,440)
440  FORMAT (' ' /'          RMNWS1          (C-H - ALL BY YEAR)'
*          /'          C-H WASTE RECEIVING MODULE CONSTRUCTION '
*, 'SCHEDULE' /'          PRIOR TO OPERATIONS, FRACTION/YEAR')
DO 460 J=1,LL
  J1=IYRCON+J-1
  WRITE (IO,450) J1,RMNWS1(J)
450  FORMAT (' ',T8,'YEAR:',I5,3X,G15.5)
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,RMNWS1(J))
460  CONTINUE
  WRITE (IO,470)
470  FORMAT (' ' /'          RMNWS2          (C-H - ALL BY YEAR)'
*          /'          C-H WASTE RECEIVING MODULE CONSTRUCTION '
*, 'SCHEDULE' /'          DURING OPERATIONS, FRACTION/YEAR')
DO 480 J=1,LL
  J1=IYRCON+J-1
  WRITE (IO,450) J1,RMNWS2(J)
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,RMNWS2(J))
480  CONTINUE
  WRITE (IO,490)
490  FORMAT (' ' /'          SFCS          (GENERIC - ALL BY YEAR)'
*          /'          OTHER SURFACE FACILITY CONSTRUCTION '
*          /'          SCHEDULE, FRACTION/YEAR')
DO 500 J=1,LL
  J1=IYRCON+J-1
  WRITE (IO,450) J1,SFCS(J)
  READ(IN,9000) ICARD

```

```

        CALL DCODER(ICARD,SFCS(J))
500 CONTINUE
    WRITE (IO,510)
510 FORMAT (' '/'          PORMDF(1,I)      (GENERIC - ALL BY YEAR)'
*          /'          SHAFT PILLAR MINING COST DISTRIBUTION'
*          /'          SCHEDULE, FRACTION/YEAR')
    DO 520 J=1,LL
        J1=IYRCON+J-1
        WRITE (IO,450) J1,PORMDF(1,J)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,PORMDF(1,J))
520 CONTINUE
    WRITE (IO,530)
530 FORMAT (' '/'          PORMDF(2,I)      (GENERIC - ALL BY YEAR)'
*          /'          MAIN CORRIDOR MINING COST DISTRIBUTION'
*          /'          SCHEDULE, FRACTION/YEAR')
    DO 540 J=1,LL
        J1=IYRCON+J-1
        WRITE (IO,450) J1,PORMDF(2,J)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,PORMDF(2,J))
540 CONTINUE
    WRITE (IO,550)
550 FORMAT (' '/'          SCSKD          (GENERIC - ALL BY YEAR)'
*          /'          SHAFT CONSTRUCTION COST DISTRIBUTION'
*          /'          SCHEDULE, FRACTION/YEAR')
    DO 560 J=1,LL
        J1=IYRCON+J-1
        WRITE (IO,450) J1,SCSKD(J)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SCSKD(J))
560 CONTINUE
    WRITE (IO,570)
570 FORMAT (' '/'          CLCDF          (GENERIC - ALL BY YEAR)'
*          /'          LICENSING COST DISTRIBUTION SCHEDULE,'
*          /'          FRACTION/YEAR')
    DO 580 J=1,LL
        J1=IYRCON+J-1
        WRITE (IO,450) J1,CLCDF(J)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,CLCDF(J))
580 CONTINUE
    WRITE (IO,590)
590 FORMAT (' '/'          CLNDF          (GENERIC - ALL BY YEAR)'
*          /'          LAND COST DISTRIBUTION SCHEDULE,'
*          /'          FRACTION/YEAR')
    DO 600 J=1,LL
        J1=IYRCON+J-1
        WRITE (IO,450) J1,CLNDF(J)

```

```

      READ(IN,9000) ICARD
      CALL DCODER(ICARD,CLNDF(J))
600 CONTINUE
      WRITE (IO,610)
610 FORMAT (' ' /'          SCDF          (GENERIC - ALL BY YEAR)'
*          /'          STARTUP COST DISTRIBUTION SCHEDULE,'
*          /'          FRACTION/YEAR')
      DO 620 J=1,LL
        J1=IYRCON+J-1
        WRITE (IO,450) J1,SCDF(J)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SCDF(J))
620 CONTINUE
      WRITE (IO,630)
630 FORMAT (' ' /'          DBDF          (GENERIC - ALL BY YEAR)'
*          /'          SHAFT PILLAR & CORRIDOR BACKFILLING COST'
*          /'          DISTRIBUTION SCHEDULE, FRACTION/YEAR')
      DO 640 J=1,IDDYR
        WRITE (IO,450) J,DBDF(J)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,DBDF(J))
640 CONTINUE
      WRITE (IO,650)
650 FORMAT (' ' /'          SSDF          (GENERIC - ALL BY YEAR)'
*          /'          SHAFT SEALING COST DISTRIBUTION SCHEDULE,'
*          /'          FRACTION/YEAR')
      DO 660 J=1,IDDYR
        WRITE (IO,450) J,SSDF(J)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SSDF(J))
660 CONTINUE
      WRITE (IO,670)
670 FORMAT (' ' /'          FDDF          (GENERIC - ALL BY YEAR)'
*          /'          FACILITY DECOMMISSIONING COST DISTRIBUTION'
*          /'          SCHEDULE, FRACTION/YEAR')
      DO 680 J=1,IDDYR
        WRITE (IO,450) J,FDDF(J)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,FDDF(J))
680 CONTINUE
      WRITE (IO,690) ITSURV
690 FORMAT (' ' /'          ITSURV          (GENERIC - ALL)'
*          /'          SURVEILANCE TIME PERIOD, YEARS',T55,I5)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,ITSURV)
C
C E. MISCELLANEOUS COST INPUT DATA
C
1030 IF (IOPTB(6,1).NE.1) RETURN

```

C

```

WRITE (IO,700)
700 FORMAT (' '/' MISCELLANEOUS COST PARAMETERS'/5X,29('='))
WRITE (IO,710) SHCFAC(1)
710 FORMAT (' '/' SHCFAC(1) (GENERIC - ALL)'
*      '/' SHIFT 1, COST DIFFERENTIALS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SHCFAC(1))
WRITE (IO,720) SHCFAC(2)
720 FORMAT (' '/' SHCFAC(2) (GENERIC - ALL)'
*      '/' SHIFT 2, 1.+ FRACTION OF SHIFT 1',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SHCFAC(2))
WRITE (IO,730) SHCFAC(3)
730 FORMAT (' '/' SHCFAC(3) (GENERIC - ALL)'
*      '/' SHIFT 3, 1.+ FRACTION OF SHIFT 1',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SHCFAC(3))
WRITE (IO,740) OVRHDF
740 FORMAT (' '/' OVRHDF (GENERIC - ALL)'
*      '/' LABOR OVERHEAD FACTOR, ',T50,G15.5
*      '/' 1+FRACTION OF SALARIES')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,OVRHDF)
WRITE (IO,760)
760 FORMAT (' '/' CONTINGENCY FACTORS')
WRITE (IO,770) CONFAC(1)
770 FORMAT (' '/' CONFAC(1) (GENERIC - ALL)'
*      '/' SITE PREPARATION, FRACTION ',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,CONFAC(1))
WRITE (IO,780) CONFAC(2)
780 FORMAT (' '/' CONFAC(2) (GENERIC - ALL))'
*      '/' ROCK OPERATIONS, FRACTION',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,CONFAC(2))
WRITE (IO,790) CONFAC(3)
790 FORMAT (' '/' CONFAC(3) (GENERIC - ALL)'
*      '/' WASTE OPERATIONS, FRACTION',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,CONFAC(3))
WRITE (IO,800) CONFAC(4)
800 FORMAT (' '/' CONFAC(4) (GENERIC - ALL)'
*      '/' BACKFILLING + DECOMMISSIONING, ',T50,G15.5
*      '/' FRACTION')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,CONFAC(4))
WRITE (IO,810) DRATE
810 FORMAT (' '/' DRATE (GENERIC - ALL)'

```

```

*          /'          DISCOUNT RATE, FRACTION',T50,G15.5)
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,DRATE)
  WRITE (IO,820) IYDOL
820 FORMAT (' ' /'          IYDOL          (GENERIC - ALL)'
*          /'          BASE YEAR FOR DISCOUNTING ',T55,I5)
  READ(IN,9000) ICARD
  CALL DCODEI(ICARD,IYDOL)

```

C

```

  RETURN
  END
  SUBROUTINE RDPREP
  INCLUDE PARAM
  INCLUDE OPTION
  INCLUDE INPUT
  CHARACTER*1
+   BLANK
  DATA BLANK /1H /

```

C

C IV. PRE-OPERATIONAL SUPPORT ACTIVITIES AND COSTS

C

```

9000 FORMAT(A80)
  IF (IOPTA.EQ.1) WRITE (IO,100)
100 FORMAT (' ' /'          ***** PRE-OPERATIONAL SUPPORT *****' /3X
+ ,76('=')/3X,76('='))
  WRITE (IO,110) TCLICE
110 FORMAT (' ' /'          TCLICE          (GENERIC - ALL)'
*          /'          LICENSING COST, $',T50,G15.5)
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,TCLICE)
  WRITE (IO,120) STPCT
120 FORMAT (' ' /'          STPCT          (GENERIC - ALL)'
*          /'          START-UP COST, FRACTION',T50,G15.5
*          /'          OF FIELD COST')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,STPCT)
  IF (IEQTP4.EQ.1) WRITE (IO,130)
130 FORMAT (' ' /'          IEQTP4'
*          /'          A-E COST FUNCTIONAL RELATIONSHIP IS LINEAR')
  IF (IEQTP4.EQ.2) WRITE (IO,140)
140 FORMAT (' ' /'          IEQTP4'
*          /'          A-E COST FUNCTIONAL RELATIONSHIP IS '
*          'EXPONENTIAL')
  WRITE (IO,150)
150 FORMAT (' ' /'          TO CHANGE ENTRY ENTER THE CORRESPONDING'
*          ' INTEGER',
*          /'          1 = LINEAR'
*          /'          2 = EXPONENTIAL').
  READ(IN,9000) ICARD

```



```

      CALL DCODEI(ICARD,IEQTP4)
      IF (IEQTP4.EQ.1) WRITE (IO,160)
160  FORMAT (' ' /'          FUNCTION = A + B * D'
*          /'          WHERE D = OSFC + UGFC + SHDF + SPUC * '
* 'SOA'
*          /'          AECON1 = A / (NWT) * (CONSTRUCTION YEARS)'
*          /'          AECON2 = A / (NWT) * (CONSTRUCTION YEARS)')
      IF (IEQTP4.EQ.2) WRITE (IO,170)
170  FORMAT (' ' /'          FUNCTION = A * EXP ( B * D )'
*          /'          WHERE D = OSFC + UGFC + SHDF + SPUC * '
* 'SOA'
*          /'          AECON1 = A / (NWT) * (CONSTRUCTION YEARS)'
*          /'          AECON2 = A / (NWT) * (CONSTRUCTION YEARS)')
      WRITE (IO,180) AECON1
180  FORMAT (' ' /'          AECON1      (GENERIC - ALL)'
*          /'          A-E COST FUNCTION CONSTANT AECON1',T50,G15.5
*)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,AECON1)
      WRITE (IO,190) AECON2
190  FORMAT (' ' /'          AECON2      (GENERIC - ALL)'
*          /'          A-E COST FUNCTION CONSTANT AECON2',T50,G15.5
*)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,AECON2)

```

C

```

      RETURN
      END
      SUBROUTINE RDROCK
      INCLUDE PARAM
      INCLUDE OPTION
      INCLUDE INPUT
      CHARACTER*1
+      BLANK
      DATA BLANK /1H /

```

C

```

9000 FORMAT(A80)

```

C

```

C  ROCK HANDLING AND DISPOSAL DESCRIPTIONS AND COST ELEMENTS

```

C

```

C1  GENERAL INFORMATION

```

C

C

```

      IF(IOPTA.EQ.1) WRITE (IO,100)
100  FORMAT (' ' /'          ***** ROCK HANDLING - DISPOSAL DESCRIPTION',
+ 'AND COSTS *****'/3X,76('=')/3X,76('='))
      IF (IOPTB(1,5).NE.1) GOTO 1000
      WRITE (IO,110)
110  FORMAT (' ' /'          GENERAL INFORMATION'/5X,19('='))

```

```

WRITE (IO,120) TEKNEK
120 FORMAT (' ' /'          TEKNEK      (GENERIC - ALL)'
*          /'          MINING TECHNIQUE, ',T50,A16
*          /'          (LIMIT 16 CHARACTERS)')
      READ(IN,9000) ICARD
      IFLG = 0
      DO 122 I=1,80
      IF(ICARD(I:I).EQ.BLANK) GOTO 122
      IFLG = 1
122    CONTINUE
      IF(IFLG.EQ.0) GOTO 124
      TEKNEK = ICARD
124    CONTINUE
      WRITE (IO,130) LEAD
130 FORMAT (' ' /'          LEAD      (GENERIC - ALL)'
*          /'          TIME LAG BETWEEN ROOM EXCAVATION',T50,I10
*          /'          AND WASTE EMPLACEMENT')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,LEAD)
      WRITE (IO,140) MSHIFT(1)
140 FORMAT (' ' /'          MSHIFT(1)      (GENERIC - ALL)'
*          /'          NUMBER OF SHIFT/DAY WORKED',T50,I10
*          /'          (250 DAY YEAR) FOR MINING DURING'
*          /'          REPOSITORY CONSTRUCTION')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,MSHIFT(1))
      WRITE (IO,150) MSHIFT(2)
150 FORMAT (' ' /'          MSHIFT(2)      (GENERIC - ALL)'
*          /'          NUMBER OF SHIFT/DAY WORKED',T50,I10
*          /'          (250 DAY YEAR) FOR MINING DURING'
*          /'          REPOSITORY OPERATION')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,MSHIFT(2))
      MFLG = MSWTCH
      WRITE (IO,160) MSWTCH
160 FORMAT (' ' /'          MSWTCH      (GENERIC - ALL)'
*          /'          ROCK HOISTING COST CALCULATION'
*          /'          OPTION = ',T50,I3
*          /'          WHERE '
*          /'          OPTION = 0 : CALCULATE USING UNIT'
*          /'          , ' COST PER TONNE OF ROCK'
*          /'          OPTION = 1 : CALCULATE USING RESOURCE'
*          /'          ' REQUIREMENT & COSTS')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,MSWTCH)
C
      IF (MFLG.EQ.MSWTCH) GOTO 1000
      WRITE(IO,1001)
1001 FORMAT(' ' /'          ALL OPTIONS HAVE BEEN SET FOR INPUT DUE TO THE'

```

```

      * ' CHANGE OF MSWTCH')
      DO 165 N=1,6
      IOPTB(N,5) = 1
165   CONTINUE
C     ROCK EXCAVATION
C
1000  IF (IOPTB(2,5).NE.1) GOTO 1010
C
      WRITE (IO,170)
170  FORMAT (' ' /'          ROCK EXCAVATION',/5X,15('='))
      IF (MSWTCH.EQ.1) GO TO 190
C
C   OPTION 1 MSWTCH = 0
C
      WRITE (IO,180) VARMC
180  FORMAT (' ' /'          VARMC      (GENERIC - ALL)'
      *           /'          MINING UNIT COST, $/TONNE',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,VARMC)
      GO TO 280
C
190  CONTINUE
C
C   OPTION 2 MSWTCH = 1
C
      WRITE (IO,200) ROLBRR(1)
200  FORMAT (' ' /'          ROLBRR(1)      (GENERIC - ALL)'
      *           /'          ROCK EXCAVATION & ROOM',T50,G15.5
      *           /'          DEVELOPMENT LABOR REQUIREMENT, '
      *           /'          MAN-YEARS/EQUIPMENT UNIT/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROLBRR(1))
      WRITE (IO,210) ROEQPL(1)
210  FORMAT (' ' /'          ROEQPL(1)      (GENERIC - ALL)'
      *           /'          ROCK EXCAVATION & ROOM',T50,G15.5
      *           /'          DEVELOPMENT EQUIPMENT LIFE, '
      *           /'          TONNES/UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROEQPL(1))
      WRITE (IO,220) ROEQPC(1)
220  FORMAT (' ' /'          ROEQPC(1)      (GENERIC - ALL)'
      *           /'          ROCK EXCAVATION & ROOM',T50,G15.5
      *           /'          DEVELOPMENT EQUIPMENT CAPACITY')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROEQPC(1))
      WRITE (IO,230) ROMATU(1)
230  FORMAT (' ' /'          ROMATU(1)      (GENERIC - ALL)'
      *           /'          ROCK EXCAVATION & ROOM',T50,G15.5
      *           /'          DEVELOPMENT MATERIAL REQUIREMENT, '

```

```

*          /'          MATERIAL UNIT/TONNE')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,ROMATU(1))
  WRITE (IO,240) ROCST(1)
240 FORMAT (' ' /'          ROCST(1)          (GENERIC - ALL)'
*          /'          ROCK EXCAVATION AND ROOM',T50,G15.5
*          /'          DEVELOPMENT CAPITAL COST, '$')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,ROCST(1))
  WRITE (IO,250) ROLBRC(1)
250 FORMAT (' ' /'          ROLBRC(1)          (GENERIC - ALL)'
*          /'          ROCK EXCAVATION AND ROOM',T50,G15.5
*          /'          DEVELOPMENT UNIT LABOR COST, '$/MAN-YEAR')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,ROLBRC(1))
  WRITE (IO,260) ROECST(1)
260 FORMAT (' ' /'          ROECST(1)          (GENERIC - ALL)'
*          /'          ROCK EXCAVATION AND ROOM',T50,G15.5
*          /'          DEVELOPMENT UNIT EQUIPMENT COST, '$/UNIT')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,ROECST(1))
  WRITE (IO,270) ROMATC(1)
270 FORMAT (' ' /'          ROMATC(1)          (GENERIC - ALL)'
*          /'          ROCK EXCAVATION AND ROOM',T50,G15.5
*          /'          DEVELOPMENT MATERIAL COST, '$/MATERIAL UNIT')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,ROMATC(1))
280 CONTINUE
C
C C3 ROCK HOISTING
C
1010 IF (IOPTB(3,5).NE.1) GOTO 1020
C
  WRITE (IO,290)
290 FORMAT (' ' /'          ROCK HOISTING'/5X,13('='))
  IF (MSWTCH.EQ.1) GO TO 310
C
C MSWTCH = 0
C
  WRITE (IO,300) RCKHC
300 FORMAT (' ' /'          RCKHC          (GENERIC - ALL)'
*          /'          ROCK HOISTING UNIT COST, ',T50,G15.5
*          /'          $/TONNE')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,RCKHC)
  GO TO 400
C
C MSWTCH = 1

```

C

```

310 CONTINUE
    WRITE (IO,320) ROLBRR(2)
320 FORMAT (' ' /'          ROLBRR(2)      (GENERIC - ALL)'
*          /'          HOISTING LABOR REQUIREMENT, ',T50,G15.5
*          /'          MAN-YEARS/UNIT/SHIFT')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROLBRR(2))
    WRITE (IO,330) ROEQPL(2)
330 FORMAT (' ' /'          ROEQPL(2)      (GENERIC - ALL)'
*          /'          HOISTING EQUIPMENT LIFE, ',T50,G15.5
*          /'          TONNES/UNIT')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROEQPL(2))
    WRITE (IO,340) ROEQPC(2)
340 FORMAT (' ' /'          ROEQPC(2)      (GENERIC - ALL)'
*          /'          HOISTING EQUIPMENT CAPACITY, ',T50,G15.5
*          /'          TONNES/UNIT/SHIFT')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROEQPC(2))
    WRITE (IO,350) ROMATU(2)
350 FORMAT (' ' /'          ROMATU(2)      (GENERIC - ALL)'
*          /'          HOISTING MATERIAL REQUIREMENT, ',T50,G15.5
*          /'          TONNES/UNIT/SHIFT')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROMATU(2))
    WRITE (IO,360) ROCST(2)
360 FORMAT (' ' /'          ROCST(2)      (GENERIC - ALL)'
*          /'          HOISTING CAPITAL COST, $',T50,G15.5)
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROCST(2))
    WRITE (IO,370) ROLBRC(2)
370 FORMAT (' ' /'          ROLBRC(2)      (GENERIC - ALL)'
*          /'          HOISTING UNIT LABOR COST, ',T50,G15.5
*          /'          $/MAN-YEAR')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROLBRC(2))
    WRITE (IO,380) ROECST(2)
380 FORMAT (' ' /'          ROECST(2)      (GENERIC - ALL)'
*          /'          HOISTING EQUIPMENT COST, ',T50,G15.5
*          /'          $/UNIT')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROECST(2))
    WRITE (IO,390) ROMATC(2)
390 FORMAT (' ' /'          ROMATC(2)      (GENERIC - ALL)'
*          /'          HOISTING MATERIAL COST, ',T50,G15.5
*          /'          $/MATERIAL UNIT')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROMATC(2))

```

```

400 CONTINUE
C
C C4 ROCK SURFACE STORAGE
C
1020 IF (IOPTB(4,5).NE.1) GOTO 1030
      WRITE (IO,405)
405  FORMAT (' ' /' SURFACE STORAGE',/5X,15('='))
      WRITE (IO,410) MSA
410  FORMAT (' ' /' MSA (REAL NUMBER) (GENERIC - ALL)'
*         /' MINED ROCK STORAGE AREA, ACRES',T50,G15.5
*         /' (ZERO RECON WILL CALCULATE)')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,MSA)
      WRITE (IO,420) TONPAR
420  FORMAT (' ' /' TONPAR (GENERIC - ALL)'
*         /' ROCK STORED PER UNIT SURFACE',T50,G15.5
*         /' AREA, TONNES/ACRE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TONPAR)
C
C MSWTCH = 0
C
      IF (MSWTCH.EQ.1) GO TO 440
      WRITE (IO,430) SURFDC
430  FORMAT (' ' /' SURFDC (GENERIC - ALL)'
*         /' UNIT SURFACE STORAGE COST,',T50,G15.5
*         /' $/TONNE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SURFDC)
      GO TO 530
C
C MSWTCH = 1
C
440 CONTINUE
      WRITE (IO,450) ROLBRR(3)
450  FORMAT (' ' /' ROLBRR(3) (GENERIC - ALL)'
*         /' ROCK STORAGE LABOR REQUIREMENT,',T50,G15.5
*         /' MAN-YEARS/UNIT/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROLBRR(3))
      WRITE (IO,460) ROEQPL(3)
460  FORMAT (' ' /' ROEQPL(3) (GENERIC - ALL)'
*         /' ROCK STORAGE EQUIPMENT LIFE,',T50,G15.5
*         /' TONNES/UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROEQPL(3))
      WRITE (IO,470) ROEQPC(3)
470  FORMAT (' ' /' ROEQPC(3) (GENERIC - ALL)'
*         /' ROCK STORAGE EQUIPMENT',T50,G15.5

```

```

*          /'          CAPACITY, TONNES/UNIT/SHIFT')
  READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROEQPC(3))
  WRITE (IO,480) ROMATU(3)
480 FORMAT (' ' /'          ROMATU(3)          (GENERIC - ALL)'
*          /'          ROCK STORAGE MATERIAL',T50,G15.5
*          /'          REQUIREMENT, MATERIAL UNIT/TONNE')
  READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROMATU(3))
  WRITE (IO,490) ROCST(3)
490 FORMAT (' ' /'          ROCST(3)          (GENERIC - ALL)'
*          /'          ROCK STORAGE CAPITAL COST, $',G15.5)
  READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROCST(3))
  WRITE (IO,500) ROLBRC(3)
500 FORMAT (' ' /'          ROLBRC(3)          (GENERIC - ALL)'
*          /'          ROCK STORAGE UNIT LABOR COST,',T50,G15.5
*          /'          , MAN-YEARS/UNIT/SHIFT')
  READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROLBRC(3))
  WRITE (IO,510) ROECST(3)
510 FORMAT (' ' /'          ROECST(3)          (GENERIC - ALL)'
*          /'          ROCK STORAGE EQUIPMENT COST,',T50,G15.5
*          /'          $/UNIT')
  READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROECST(3))
  WRITE (IO,520) ROMATC(3)
520 FORMAT (' ' /'          ROMATC(3)          (GENERIC - ALL)'
*          /'          ROCK STORAGE MATERIAL COST,',T50,G15.5
*          /'          $/MATERIAL UNIT')
  READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROMATC(3))
530 CONTINUE
C
C C5 OFF SITE DISPOSAL
C
1030 IF (IOPTB(5,5).NE.1) GOTO 1040
C
  WRITE (IO,540)
540 FORMAT (' ' /'          OFFSITE DISPOSAL'/5X,16('='))
  WRITE (IO,550) OFTADC
550 FORMAT (' ' /'          OFTADC          (GENERIC - ALL)'
*          /'          UNIT OFFSITE TRANSPORTATION',T50,G15.5
*          /'          AND DISPOSAL COST')
  READ(IN,9000) ICARD
    CALL DCODER(ICARD,OFTADC)
C
C
C MSWTC = 1

```

C

```

560 CONTINUE
    IF (MSWTCH.NE.1) GOTO 650
    WRITE (IO,570) ROLBRR(4)
570 FORMAT (' '/' ' ROLBRR(4) (GENERIC - ALL)'
*          /' OFFSITE TRANSPORTATION AND',T50,G15.5
*          /' LABOR REQUIREMENT,MAN-YEARS/UNIT/SHIFT')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROLBRR(4))
    WRITE (IO,580) ROEQPL(4)
580 FORMAT (' '/' ' ROEQPL(4) (GENERIC - ALL)'
*          /' OFFSITE TRANSPORTATION AND DISPOSAL',T50,
*G15.5 /' EQUIPMENT LIFE, TONNES/UNIT')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROEQPL(4))
    WRITE (IO,590) ROEQPC(4)
590 FORMAT (' '/' ' ROEQPC(4) (GENERIC - ALL)'
*          /' OFFSITE TRANSPORTATION AND',T50,G15.5
*          /' DISPOSAL EQUIPMENT CAPACITY,'
*          /' TONNES/UNIT/SHIFT/YR')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROEQPC(4))
    WRITE (IO,600) ROMATU(4)
600 FORMAT (' '/' ' ROMATU(4) (GENERIC - ALL)'
*          /' OFFSITE TRANSPORTATION AND',T50,G15.5
*          /' DISPOSAL MATERIAL REQUIREMENT,'
*          /' MATERIAL UNIT/TONNE')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROMATU(4))
    WRITE (IO,610) ROCST(4)
610 FORMAT (' '/' ' ROCST(4) (GENERIC - ALL)'
*          /' OFFSITE TRANSPORTATION AND',T50,G15.5
*          /' DISPOSAL CAPITAL COST, $')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROCST(4))
    WRITE (IO,620) ROLBRC(4)
620 FORMAT (' '/' ' ROLBRC(4) (GENERIC - ALL)'
*          /' OFFSITE TRANSPORTATION AND',T50,G15.5
*          /' DISPOSAL UNIT LABOR COST, $/MAN-YEAR')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROLBRC(4))
    WRITE (IO,630) ROECST(4)
630 FORMAT (' '/' ' ROECST(4) (GENERIC - ALL)'
*          /' OFFSITE TRANSPORTATION AND',T50,G15.5
*          /' DISPOSAL EQUIPMENT COST, $/UNIT')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,ROECST(4))
    WRITE (IO,640) ROMATC(4)
640 FORMAT (' '/' ' ROMATC(4) (GENERIC - ALL)'

```



```

*          /'      OFFSITE TRANSPORTATION AND',T50,G15.5
*          /'      DISPOSAL MATERIAL COST, MATERIAL UNIT',
* '/TONNE')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,ROMATC(4))
650 CONTINUE
C
C C6. BACKFILLING
C
1040 IF (IOPTB(6,5).NE.1) RETURN
C
  WRITE (IO,660)
660 FORMAT (' '//'      BACKFILLING'/5X,11('='))
  WRITE (IO,670) PCTFIL
670 FORMAT (' '//'      PCTFIL      (GENERIC - ALL)'
*          /'      PERCENTAGE OF EACH ROOM',T50,G15.5
*          /'      BACKFILLED')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,PCTFIL)
  WRITE (IO,680) XPANFC
680 FORMAT (' '//'      XPANFC      (GENERIC - ALL)'
*          /'      EXPANSION IN ROCK VOLUME',T50,G15.5
*          /'      AFTER MINING, 1+FRACTION')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,XPANFC)
  WRITE (IO,690) OTHBFL
690 FORMAT (' '//'      OTHBFL      (GENERIC - ALL)'
*          /'      PERCENTAGE OF BACKFILL VOLUME',T50,G15.5
*          /'      COMPRISED OF A SUPPLEMENTARY MATERIAL'
*          /'      , FRACTION')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,OTHBFL)
  WRITE (IO,700) OTBDEN
700 FORMAT (' '//'      OTBDEN      (GENERIC - ALL)'
*          /'      DENSITY OF SUPPLEMENTARY',T50,G15.5
*          /'      BACKFILL MATERIAL, G/CM**3')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,OTBDEN)
  WRITE (IO,710) BKRATE
710 FORMAT (' '//'      BKRATE      (GENERIC - ALL)'
*          /'      BACKFILLING RATE, TONNES/YEAR',T50,G15.5
*          /'      (USES SHAFT CAPACITY IF ZERO)')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,BKRATE)
C
C MSWTC = 0
C
  IF (MSWTC.EQ.1) GO TO 730
  WRITE (IO,720) BKFLC

```

```

720 FORMAT (' '/' BKFLC (GENERIC - ALL)'
*          '/' UNIT BACKFILLING COST',T50,G15.5
*          '/' $/TONNE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,BKFLC)
      GO TO 820

C
C MSWTC = 1
C
730 CONTINUE
      WRITE (IO,740) ROLBRR(5)
740 FORMAT (' '/' ROLBRR(5) (GENERIC - ALL)'
*          '/' BACKFILLING LABOR REQUIREMENT,',T50,G15.5
*          '/' MAN-YEAR/UNIT/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROLBRR(5))
      WRITE (IO,750) ROEQPL(5)
750 FORMAT (' '/' ROEQPL(5) (GENERIC - ALL)'
*          '/' BACKFILLING EQUIPMENT LIFE,',T50,G15.5
*          '/' TONNES/UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROEQPL(5))
      WRITE (IO,760) ROEQPC(5)
760 FORMAT (' '/' ROEQPC(5) (GENERIC - ALL)'
*          '/' BACKFILLING EQUIPMENT',T50,G15.5
*          '/' CAPACITY, TONNES/UNIT/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROEQPC(5))
      WRITE (IO,770) ROMATU(5)
770 FORMAT (' '/' ROMATU(5) (GENERIC - ALL)'
*          '/' BACKFILLING MATERIAL',T50,G15.5
*          '/' REQUIREMENT, MATERIAL UNIT/TONNE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROMATU(5))
      WRITE (IO,780) ROCST(5)
780 FORMAT (' '/' ROCST(5) (GENERIC - ALL)'
*          '/' BACKFILLING CAPITAL COST, $',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROCST(5))
      WRITE (IO,790) ROLBRC(5)
790 FORMAT (' '/' ROLBRC(5) (GENERIC - ALL)'
*          '/' BACKFILLING UNIT LABOR COST,',T50,G15.5
*          '/' $/MAN-YEAR')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROLBRC(5))
      WRITE (IO,800) ROECST(5)
800 FORMAT (' '/' ROECST(5) (GENERIC - ALL)'
*          '/' BACKFILLING UNIT EQUIPMENT',T50,G15.5
*          '/' COST, $/UNIT')

```

```

      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROECST(5))
      WRITE (IO,810) ROMATC(5)
810  FORMAT (' ' /'          ROMATC(5)          (GENERIC - ALL)'
*          /'          BACKFILLING UNIT MATERIAL',T50,G15.5
*          /'          COSTS, $/MATERIAL UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ROMATC(5))
820  CONTINUE
C
C
      RETURN
      END
      SUBROUTINE RDRPOS
C
      INCLUDE PARAM,LIST
      INCLUDE OPTION,LIST
      INCLUDE INPUT,LIST
      CHARACTER*1
+      BLANK
      DATA BLANK /1H /
C
C  III.  REPOSITORY SYSTEM DESCRIPTION
C
C
      IF(IOPTA.EQ.1) WRITE (IO,100)
100  FORMAT (' ' // ' ***** REPOSITORY SYSTEMS *****' /
+3X,76('=')/3X,76('='))
C
C
      IF(IWNUM(1).EQ.0) INWT=NWT
C
      IF (IOPTB(1,3).NE.1) GOTO 2000
1000  FORMAT(A80)
C
C      PANEL DESCRIPTION
C
      WRITE (IO,110)
110  FORMAT (' ' /'          PANEL PARAMETERS'/5X,16('='))
      WRITE (IO,130)
130  FORMAT (' ' /'          PANLEN(K)          (BY WASTE TYPE)'
*          /'          PANEL LENGTH, METERS')
      DO 400 KNUM=1,INWT
      K=INUM(KNUM)
      WRITE(IO,410) K,PANLEN(K)
410  FORMAT(' ',T10,'WASTE TYPE ',I3,T50,G15.5)
      READ(IN,1000) ICARD
      CALL DCODER(ICARD,PANLEN(K))
400  CONTINUE

```

```

        WRITE (IO,140)
140    FORMAT (' ' /'          PANWTH(K)          (BY WASTE TYPE)'
*          /'          PANEL WIDTH, METERS')
        DO 420 KNUM=1,INWT
            K=INUM(KNUM)
            WRITE(IO,410) K,PANWTH(K)
            READ(IN,1000) ICARD
            CALL DCODER(ICARD,PANWTH(K))
420    CONTINUE
150    CONTINUE
        WRITE (IO,170) GXEF
170    FORMAT (' ' /'          GXEF          (GENERIC - ALL)'
*          /'          GEOMETRIC EXCLUSION FACTOR,',T50,G15.5
*          /'          FRACTION')
            READ(IN,1000) ICARD
            CALL DCODER(ICARD,GXEF)
            WRITE (IO,180) CPA
180    FORMAT (' ' /'          CPA          (GENERIC - ALL)'
*          /'          CENTRAL PILLAR AREA, ACRES',T50,G15.5)
            READ(IN,1000) ICARD
            CALL DCODER(ICARD,CPA)
C
        WRITE (IO,160)
160    FORMAT (' ' /'          NBSPPA(K)          (BY WASTE TYPE)'
*          /'          NUMBER OF BORE SECTIONS'
*          /'          PER PANEL')
        DO 165 KNUM=1,INWT
            K=INUM(KNUM)
            WRITE(IO,415) K,NBSPPA(K)
415    FORMAT(' ',T10,'WASTE TYPE ',I3,T50,I10)
            READ(IN,1000) ICARD
            CALL DCODEI(ICARD,NBSPPA(K))
165    CONTINUE
C
C    ROOM DESCRIPTIONS
C
2000    IF (IOPTB(2,3).NE.1) GOTO 2010
C
        WRITE (IO,330)
330    FORMAT (' ' /'          ROOM PARAMETERS (BY WASTE TYPE)'/5X,31('='))
        WRITE (IO,340)
340    FORMAT (' ' /'          RMLN(K)          (BY WASTE TYPE)'
*          /'          ROOM LENGTH,METERS')
        DO 345 KNUM=1,INWT
            K=INUM(KNUM)
            WRITE(IO,410) K,RMLN(K)
            READ(IN,1000) ICARD
            CALL DCODER(ICARD,RMLN(K))
345    CONTINUE

```

```

        WRITE (IO,350)
350  FORMAT (' ' /'          RMWTH(K)      (BY WASTE TYPE)'
*      /'          ROOM WIDTH, METERS')
        DO 355 KNUM=1,INWT
            K=INUM(KNUM)
            WRITE(IO,410) K,RMWTH(K)
            READ(IN,1000) ICARD
            CALL DCODER(ICARD,RMWTH(K))
355  CONTINUE
        WRITE (IO,360)
360  FORMAT (' ' /'          RMHMIN(K)      (BY WASTE TYPE)'
*      /'          MINIMUM ROOM HEIGHT, METERS')
        DO 365 KNUM=1,INWT
            K=INUM(KNUM)
            WRITE(IO,410) K,RMHMIN(K)
            READ(IN,1000) ICARD
            CALL DCODER(ICARD,RMHMIN(K))
365  CONTINUE
        WRITE (IO,380)
380  FORMAT(' ' /'          DELRHT(K)      (BY WASTE TYPE)'
*      /'          INCREMENTAL ROOM HEIGHT'
*      /'          OVER PACKAGE HEIGHT (FOR VERTICAL'
*      ' BOREHOLES)')
        DO 385 KNUM=1,INWT
            K=INUM(KNUM)
            WRITE(IO,410) K,DELRHT(K)
            READ(IN,1000) ICARD
            CALL DCODER(ICARD,DELRHT(K))
385  CONTINUE
        WRITE (IO,390)
390  FORMAT (' ' /'          XRASHO(K)      (BY WASTE TYPE)'
*      /'          EXTRACTION RATIO, METERS'
*      /'          (ROOM WIDTH)(#ROOMS)/(PANEL LENGTH)'
C  *      /'          (RMWTH * NBSPPA) / PANLEN')
        DO 395 KNUM=1,INWT
            K=INUM(KNUM)
            WRITE(IO,410) K,XRASHO(K)
            READ(IN,1000) ICARD
            CALL DCODER(ICARD,XRASHO(K))
395  CONTINUE
C
C  CORRIDOR DESCRIPTIONS
C
2010      IF(IOPTB(3,3).NE.1) RETURN
C
        WRITE (IO,190)
190  FORMAT (' ' /'          CORRIDOR PARAMETERS' /5X,19('='))
C
C  MAIN CORRIDOR

```

C

```

WRITE (IO,200) CORWTH(1)
200 FORMAT (' ' / ' ' CORWTH(1) '
*          / '          MAIN CORRIDOR WIDTH, METERS',T50,G15.5)
      READ(IN,1000) ICARD
      CALL DCODER(ICARD,CORWTH(1))
      WRITE (IO,210) CORLEN(1,1)
210 FORMAT (' ' / ' ' CORLEN(1,1) '
*          / '          MAIN CORRIDOR LENGTH, METERS',T50,G15.5)
      READ(IN,1000) ICARD
      CALL DCODER(ICARD,CORLEN(1,1))
      WRITE (IO,220) CORHT(1)
220 FORMAT (' ' / ' ' CORHT(1) '
*          / '          MAIN CORRIDOR HEIGHT, METERS',T50,G15.5)
      READ(IN,1000) ICARD
      CALL DCODER(ICARD,CORHT(1))
      WRITE (IO,230) TCPCLN
230 FORMAT (' ' / ' ' TCPCLN '
*          / '          TOTAL CENTRAL PILLAR CORRIDOR',T50,G15.5
*          / '          LENGTH, METERS')
      READ(IN,1000) ICARD
      CALL DCODER(ICARD,TCPCLN)
      NACOR=NCOR-1
      WRITE (IO,240) NACOR
240 FORMAT (' ' / ' ' NCOR - 1 '
*          / '          NUMBER OF ADDITIONAL CORRIDORS = ',
*          T50,I10)
      READ(IN,1000) ICARD
      CALL DCODEI(ICARD,NACOR)
      NCOR=NCOR+1
      IF (NCOR.LE.1) GO TO 320
      WRITE (IO,250)
250 FORMAT (' ' / ' ' OTHER CORRIDOR DESCRIPTIONS'/T9,27('-'))
      DO 310 JJ=2,NCOR
      J=JJ-1
      WRITE (IO,260) J,CORDSC(JJ)
260 FORMAT (' ' / ' ' CORDSC(JJ) (ADDITIONAL CORRIDOR',I3,') '
*          / '          DESCRIPTION (LIMIT 48 CHARACTERS)'/
*          ,T15,A48)
      READ(IN,1000) ICARD
      DO 263 I=1,80
      IF (ICARD(I:I).EQ.BLANK) GOTO 263
      IFLG=1
263 CONTINUE
      IF (IFLG.EQ.0) GOTO 264
      CORDSC(JJ)=ICARD(1:48)
264 CONTINUE
      WRITE (IO,270) J,CORWTH(JJ)
270 FORMAT (' ' / ' ' CORWTH(JJ) (ADDITIONAL CORRIDOR',I3,') '

```

```

*          /'          OTHER CORRIDOR WIDTH, METERS',T50,G15.5)
      READ(IN,1000) ICARD
      CALL DCODER(ICARD,CORWTH(JJ))
      WRITE (IO,280) J
280  FORMAT (' ' /'          CORLEN(J,K)    (ADDITIONAL CORRIDOR',I3,
*  ' BY WASTE TYPE)'
*          /'          OTHER CORRIDOR LENGTH, METERS'/)
      DO 282 KNUM=1,INWT
      K=INUM(KNUM)
      WRITE(IO,450) K,CORLEN(JJ,K)
450  FORMAT(' ',T10,'WASTE TYPE ',I3,T50,G15.5)
      READ(IN,1000) ICARD
      CALL DCODER(ICARD,CORLEN(JJ,K))
411  FORMAT(' ',T50,G15.5)
282  CONTINUE
      WRITE (IO,290) J,CORHT(JJ)
290  FORMAT (' ' /'          CORHT(J)    (ADDITIONAL CORRIDOR',I3,')'
*          /'          OTHER CORRIDOR HEIGHT, METERS',T50,G15.5)
      READ(IN,1000) ICARD
      CALL DCODER(ICARD,CORHT(JJ))
      WRITE (IO,300) J,NCORPP(JJ)
300  FORMAT (' ' /'          NCORPP(J)    (ADDITIONAL CORRIDOR',I3,')'
*          /'          NUMBER CORRIDORS PER PANEL',T50,I10)
      READ(IN,1000) ICARD
      CALL DCODEI(ICARD,NCORPP(JJ))
310  CONTINUE
320  CONTINUE
C
C
      RETURN
      END
      SUBROUTINE RDSHFT
      INCLUDE PARAM
      INCLUDE OPTION
      INCLUDE INPUT
      CHARACTER*1
      +   BLANK
      DATA BLANK /1H /
C
C 9000  FORMAT(A80)
C
C  E.  SHAFT DESCRIPTIONS AND COSTS
C
      IF(IOPTA.EQ.1) WRITE (IO,100)
100  FORMAT (' ' /' ***** SHAFT DESCRIPTIONS AND COSTS *****'
*  /3X,76('=')/3X,76('='))
C
      IF(IWNUM(1).EQ.0) INWT=NWT
C

```

```

C   B.1  MEN AND MATERIALS AND ROCK
C
      IF (IOPTB(1,4).NE.1) GOTO 5000
C
      WRITE (IO,110)
110  FORMAT ('/'/'      MEN & MATERIALS SHAFT'/5X,21('='))
      WRITE (IO,120)
120  FORMAT ('/'/'      SHAFT TYPE',T38,'NUMBER OF SHAFTS',
* /T11,10('-'),T38,16('-'))
      WRITE (IO,130) SFTNUM(1)
130  FORMAT ('/'/'      SFTNUM(1)'
* /'/'      MEN & MATERIALS ',T40,I10)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,SFTNUM(1))
      IF (SFTNUM(1).EQ.0) GO TO 220
      WRITE (IO,140) SFTDD(1)
140  FORMAT ('/'/'      SFTDD(1)'
* /'/'      DIAMETER, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTDD(1))
      WRITE (IO,150) SFTXR(1)
150  FORMAT ('/'/'      SFTXR(1)'
* /'/'      INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
* /'/'      METERS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTXR(1))
      WRITE (IO,160) SFTXD(1)
160  FORMAT ('/'/'      SFTXD(1)'
* /'/'      EXCAVATED DEPTH, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTXD(1))
      WRITE (IO,170) SFTLD(1)
170  FORMAT ('/'/'      SFTLD(1)'
* /'/'      LINED DEPTH, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTLD(1))
      WRITE (IO,180) MXMMRH
180  FORMAT ('/'/'      MXMMRH'
* /'/'      MAXIMUM TRIPS PER SHIFT FOR',T50,I10
* /'/'      ROCK HOISTING FUNCTIONS')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,MXMMRH)
      WRITE (IO,190) TNPT2
190  FORMAT ('/'/'      TNPT2'
* /'/'      ROCK HOISTING CAPACITY,',T50,G15.5
* /'/'      TONNES PER TRIP')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TNPT2)
      WRITE (IO,200) SFTHE(1)

```



```

200 FORMAT (' '/' SFTHE(1)'
*          '/' SHAFT HEADFRAME AND CAPITAL',T50,G15.5
*          '/' EQUIPMENT COST, '$')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTHE(1))
      WRITE (IO,210) SFTSS(1)
210 FORMAT (' '/' SFTSS(1)'
*          '/' MEN & MATERIALS SHAFT STATION',T50,G15.5
*          '/' COST, '$')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTSS(1))
      GO TO 230
220 CONTINUE
      SFTDD(1)=0.0
      SFTXR(1)=0.0
      SFTXD(1)=0.0
      SFTLD(1)=0.0
      MXMMRH=0
      TNPT2=0.0
      SFTHE(1)=0.0
      SFTSS(1)=0.0
230 CONTINUE
C
C B2      REMOTE HANDLED WASTE SHAFT
C
5000 IF (IOPTB(2,4).NE.1) GOTO 5010
C
      WRITE (IO,240)
240 FORMAT (' '/' REMOTE - HANDLED WASTE SHAFT'/5X,28('='))
      WRITE (IO,250)
250 FORMAT (' '/' ENTER THE NUMBER OF SHAFTS OF THE FOLLOWING TYPES'
+ '/' (ENTERING A -1 FOR WASTE-SHAFTS ALLOWS RECON TO '/' CALCULATE
+THE NUMBER NEEDED. IF CONTACT-HANDLED '/' WASTE IS PRESENT AND 0
+CONTACT-HANDLED WASTE '/' SHAFTS ARE SPECIFIED, IT WILL BE TRANSP
+ORTED IN '/' THE REMOTE-HANDLED WASTE SHAFT.))
      WRITE (IO,120)
      WRITE (IO,260) SFTNUM(3)
260 FORMAT (' '/' SFTNUM(3)'
*          '/' R-H WASTE ',T40,I10)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,SFTNUM(3))
      IF (SFTNUM(3).EQ.0) GO TO 370
      WRITE (IO,270) SFTDD(3)
270 FORMAT (' '/' SFTDD(3)'
*          '/' DIAMETER, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTDD(3))
      WRITE (IO,280) SFTXR(3)
280 FORMAT (' '/' SFTXR(3)'

```

```

*          /'          INCREMENTAL EXCAVATED RADIUS, ',T50,G15.5
*          /'          METERS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTXR(3))
      WRITE (IO,290) SFTXD(3)
290  FORMAT (' ' /'          SFTXD(3)'
*          /'          EXCAVATED DEPTH, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTXD(3))
      WRITE (IO,300) SFTLD(3)
300  FORMAT (' ' /'          SFTLD(3)'
*          /'          LINED DEPTH, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTLD(3))
      WRITE (IO,310) MTRPSD
310  FORMAT (' ' /'          MTRPSD'
*          /'          MAXIMUM TRIPS PER SHAFT FOR R-H WASTE '
*          ,T50,I10)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,MTRPSD)
      DO 330 KNUM=1,INWT
      K=INUM(KNUM)
      IF (KIND(K).EQ.2) GO TO 330
      WRITE (IO,320) NPKPTR(K),K
320  FORMAT (' ' /'          NPKPTR(K)      (BY R-H WASTE TYPE)'
*          /'          PACKAGES PER TRIP',T50,I11
*          /'          FOR R-H WASTE TYPE ',I5)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,NPKPTR(K))
330  CONTINUE
      IF (SFTNUM(3).EQ.0) GO TO 340
340  CONTINUE
      WRITE (IO,350) SFTHE(3)
350  FORMAT (' ' /'          SFTHE(3)'
*          /'          SHAFT HEADFRAME AND CAPITAL',T50,G15.5
*          /'          EQUIPMENT COST, $')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTHE(3))
      WRITE (IO,360) SFTSS(3)
360  FORMAT (' ' /'          SFTSS(3)'
*          /'          R-H WASTE SHAFT STATION',T50,G15.5
*          /'          COST, $')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTSS(3))
      GO TO 380
370  CONTINUE
      SFTDD(3)=0.0
      SFTXR(3)=0.0
      SFTXD(3)=0.0

```

```

      SFTLD(3)=0.0
      SFTHE(3)=0.0
      SFTSS(3)=0.0
      MTRPSD = 0
380  CONTINUE
C
C  B3  CONTACT HANDLED WASTE SHAFT
C
5010  IF (IOPTB(3,4).NE.1) GOTO 5020
C
      IFLG=0
      DO 390 KNUM=1,INWT
      K=INUM(KNUM)
      IF (KIND(K).EQ.2) IFLG=1
390  CONTINUE
      WRITE (IO,400)
400  FORMAT (' ' /'          CONTACT - HANDLED WASTE SHAFT' /5X,29('='))
      WRITE (IO,120)
      WRITE (IO,410) SFTNUM(4)
410  FORMAT (' ' /'          SFTNUM(4)'
*      /'          C-H WASTE  ',T40,I10)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,SFTNUM(4))
      IF (IFLG.EQ.0) GO TO 510
      IF (SFTNUM(4).EQ.0) GO TO 510
      WRITE (IO,420) SFTDD(4)
420  FORMAT (' ' /'          SFTDD(4)'
*      /'          DIAMETER, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTDD(4))
      WRITE (IO,430) SFTXR(4)
430  FORMAT (' ' /'          SFTXR(4)'
*      /'          INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
*      /'          METERS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTXR(4))
      WRITE (IO,440) SFTXD(4)
440  FORMAT (' ' /'          SFTXD(4)'
*      /'          EXCAVATED DEPTH, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTXD(4))
      WRITE (IO,450) SFTLD(4)
450  FORMAT (' ' /'          SFTLD(4)'
*      /'          LINED DEPTH, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTLD(4))
      WRITE (IO,460) MTCPSD
460  FORMAT (' ' /'          MTCPSD'
*      /'          MAXIMUM TRIPS PER SHIFT',T50,I10)

```

```

      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,MTCPSD)
DO 480 KNUM=1,INWT
K=INUM(KNUM)
  IF (KIND(K).NE.2) GO TO 480
  WRITE (IO,470) NPKPTC(K),K
470  FORMAT (' ' /'      NPKPTC(K)      (BY WASTE TYPE) '
*          /'      PACKAGES PER TRIP',T50,I10
*          /'      FOR C-H WASTE TYPE',I4)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,NPKPTC(K))
480  CONTINUE
      WRITE (IO,490) SFTHE(4)
490  FORMAT (' ' /'      SFTHE(4) '
*          /'      SHAFT HEADFRAME & CAPITAL',T50,G15.5
*          /'      EQUIPMENT COST, $')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTHE(4))
      WRITE (IO,500) SFTSS(4)
500  FORMAT (' ' /'      SFTSS(4) '
*          /'      C-H WASTE SHAFT STATION',T50,G15.5
*          /'      COST, $')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTSS(4))
      GO TO 550
510  CONTINUE
      WRITE (IO,460) MTCPSD
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,MTCPSD)
DO 530 KNUM=1,INWT
K=INUM(KNUM)
  IF (KIND(K).NE.2) GO TO 530
  WRITE (IO,520) NPKPTC(K),K
520  FORMAT (' ' /'      NPKPTC(K)      (BY C-H WASTE TYPE) '
*          /'      PACKAGES PER TRIP',T50,I10
+          /'      FOR C-H WASTE TYPE',I4,
*          /'      IN REMOTE HANDLED WASTE SHAFT')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,NPKPTC(K))
530  CONTINUE
      GOTO 550
540  CONTINUE
      SFTDD(4)=0.0
      SFTXR(4)=0.0
      SFTXD(4)=0.0
      SFTLD(4)=0.0
      MTCPSD=0
      SFTHE(4)=0.0
      SFTSS(4)=0.0

```

```

550 CONTINUE
C
C   B4   MINE VENTILATION SHAFT
C
5020 IF (IOPTB(4,4).NE.1) GOTO 5030
C
      WRITE (IO,560)
560  FORMAT (' ' /'          MINE VENTILATION SHAFT' /5X,22('='))
      WRITE (IO,120)
      WRITE (IO,570) SFTNUM(2)
570  FORMAT (' ' /'          SFTNUM(2)'
*         /'          MINE VENTILATION',T40,I10)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,SFTNUM(2))
C   B4   MINE VENTILATION SHAFT
C
      IF (SFTNUM(2).EQ.0) GO TO 640
      WRITE (IO,580) SFTDD(2)
580  FORMAT (' ' /'          SFTDD(2)'
*         /'          DIAMETER, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTDD(2))
      WRITE (IO,590) SFTXR(2)
590  FORMAT (' ' /'          SFTXR(2)'
*         /'          INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
*         /'          METERS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTXR(2))
      WRITE (IO,600) SFTXD(2)
600  FORMAT (' ' /'          SFTXD(2)'
*         /'          EXCAVATED DEPTH, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTXD(2))
      WRITE (IO,610) SFTLD(2)
610  FORMAT (' ' /'          SFTLD(2)'
*         /'          LINED DEPTH, METERS',T50,G1
+5.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTLD(2))
      WRITE (IO,620) SFTHE(2)
620  FORMAT (' ' /'          SFTHE(2)'
*         /'          SHAFT HEADFRAME & CAPITAL',T50,G15.5
*         /'          EQUIPMENT COST, $')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTHE(2))
      WRITE (IO,630) SFTSS(2)
630  FORMAT (' ' /'          SFTSS(2)'
*         /'          MINING VENTILATION SHAFT',T50,G15.5
*         /'          STATION EQUIPMENT COST, $')

```

```

        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SFTSS(2))
        GO TO 650
640  CONTINUE
        SFTXR(2)=0.0
        SFTXD(2)=0.0
        SFTLD(2)=0.0
        SFTHE(2)=0.0
        SFTSS(2)=0.0
650  CONTINUE
C
C  B5  MINING EXHAUST SHAFT
C
5030 IF (IOPTB(5,4).NE.1) GOTO 5040
C
        WRITE (IO,660)
660  FORMAT (' ' /'          MINING EXHAUST SHAFT' /5X,20('='))
        WRITE (IO,120)
        WRITE (IO,670) SFTNUM(5)
670  FORMAT (' ' /'          SFTNUM(5)'
*          /'          MINING EXHAUST  ',T40,I10)
        READ(IN,9000) ICARD
        CALL DCODEI(ICARD,SFTNUM(5))
        IF (SFTNUM(5).EQ.0) GO TO 740
        WRITE (IO,680) SFTDD(5)
680  FORMAT (' ' /'          SFTDD(5)'
*          /'          DIAMETER, METERS',T50,G15.5
+)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SFTDD(5))
        WRITE (IO,690) SFTXR(5)
690  FORMAT (' ' /'          SFTXR(5)'
*          /'          INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
*          /'          METERS')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SFTXR(5))
        WRITE (IO,700) SFTXD(5)
700  FORMAT (' ' /'          SFTXD(5)'
*          /'          EXCAVATED DEPTH, METERS',T50,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SFTXD(5))
        WRITE (IO,710) SFTLD(5)
710  FORMAT (' ' /'          SFTLD(5)'
*          /'          LINED DEPTH, METERS',T50,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SFTLD(5))
        WRITE (IO,720) SFTHE(5)
720  FORMAT (' ' /'          SFTHE(5)'
*          /'          SHAFT HEADFRAME & CAPITAL',T50,G15.5

```

```

*          /'          EQUIPMENT COST, $ ')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTHE(5))
      WRITE (IO,730) SFTSS(5)
730  FORMAT (' ' /'          SFTSS(5)'
*          /'          MINING EXHAUST SHAFT',T50,G15.5
*          /'          STATION EQUIPMENT COST, $')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTSS(5))
      GO TO 750
740  CONTINUE
      SFTXR(5)=0.0
      SFTXD(5)=0.0
      SFTXD(5)=0.0
      SFTLD(5)=0.0
      SFTHE(5)=0.0
      SFTSS(5)=0.0
750  CONTINUE
C
C  B6  WASTE EXHAUST SHAFT
C
5040 IF(IOPTB(6,4).NE.1) GOTO 5050
C
      WRITE (IO,760)
760  FORMAT (' ' /'          WASTE EXHAUST SHAFT'/5X,19('='))
      WRITE (IO,250)
      WRITE (IO,120)
      WRITE (IO,770) SFTNUM(6)
770  FORMAT (' ' /'          SFTNUM(6)'
*          /'          WASTE EXHAUST  ',T40,I10)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,SFTNUM(6))
      IF (SFTNUM(6).EQ.0) GO TO 840
      WRITE (IO,780) SFTDD(6)
780  FORMAT (' ' /'          SFTDD(6)'
*          /'          DIAMETER, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTDD(6))
      WRITE (IO,790) SFTXR(6)
790  FORMAT (' ' /'          SFTXR(6)'
*          /'          INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
*          /'          METERS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTXR(6))
      WRITE (IO,800) SFTXD(6)
800  FORMAT (' ' /'          SFTXD(6)'
*          /'          EXCAVATED DEPTH, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTXD(6))

```

```

        WRITE (IO,810) SFTLD(6)
810  FORMAT (' ' /'          SFTLD(6)'
*          /'          LINED DEPTH, METERS',T50,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SFTLD(6))
        WRITE (IO,820) SFTHE(6)
820  FORMAT (' ' /'          SFTHE(6)'
*          /'          SHAFT HEADFRAME & CAPITAL',T50,G15.5
*          /'          EQUIPMENT COST')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SFTHE(6))
        WRITE (IO,830) SFTSS(6)
830  FORMAT (' ' /'          SFTSS(6)'
*          /'          WASTE EXHAUST SHAFT',T50,G15.5
*          /'          STATION EQUIPMENT COST, $')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SFTSS(6))
        GO TO 850
840  CONTINUE
        SFTXR(6)=0.0
        SFTXD(6)=0.0
        SFTLD(6)=0.0
        SFTHE(6)=0.0
        SFTSS(6)=0.0
850  CONTINUE
C
C  B7  ROCK HOISTING SHAFT
C
5050 IF (IOPTB(7,4).NE.1) GO TO 5060
C
        WRITE (IO,860)
860  FORMAT (' ' /'          ROCK HOISTING SHAFT' /5X,19('='))
        WRITE (IO,120)
        WRITE (IO,870)
870  FORMAT (' ' /'          ROCK HOISTING ONLY  ')
        WRITE (IO,880) SFTDD(7)
880  FORMAT (' ' /'          SFTDD(7)'
*          /'          DIAMETER, METERS',T50,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SFTDD(7))
        WRITE (IO,890) SFTXR(7)
890  FORMAT (' ' /'          SFTXR(7)'
*          /'          INCREMENTAL EXCAVATED RADIUS,',T50,G15.5
*          /'          METERS')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,SFTXR(7))
        WRITE (IO,900) SFTXD(7)
900  FORMAT (' ' /'          SFTXD(7)'
*          /'          EXCAVATED DEPTH, METERS',T50,G15.5)

```



```

      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTXD(7))
      WRITE (IO,910) SFTLD(7)
910  FORMAT (' '//' SFTLD(7)'
      *          '/' LINED DEPTH, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTLD(7))
      WRITE (IO,920) MXMPRH
920  FORMAT (' '//' MXMPRH'
      *          '/' MAXIMUM TRIPS PER SHAFT FOR',T50,I10
      *          '/' ROCK HOISTING (MINE PRODUCTION)')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,MXMPRH)
      WRITE (IO,930) TNPT1
930  FORMAT (' '//' TNPT1'
      *          '/' ROCK HOISTING CAPACITY ',T50,G15.5
      *          '/' TONNES PER TRIP')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TNPT1)
      WRITE (IO,940) SFTHE(7)
940  FORMAT (' '//' SFTHE(7)'
      *          '/' SHAFT HEADFRAME AND CAPITAL',T50,G15.5
      *          '/' EQUIPMENT COST, '$')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTHE(7))
      WRITE (IO,950) SFTSS(7)
950  FORMAT (' '//' SFTSS(7)'
      *          '/' ROCK HOISTING SHAFT',T50,G15.5
      *          '/' STATION EQUIPMENT COST, '$')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFTSS(7))
970  CONTINUE
C
C  B8  SHAFT COST FUNCTION COEFFICIENTS
C
5060 IF(IOPTB(8,4).NE.1) RETURN
      WRITE (IO,980)
980  FORMAT (' '//' SHAFT COST FUNCTIONS'/5X,20('='))
      IF (IEQTP1.EQ.1) WRITE (IO,990)
990  FORMAT (' '//' IEQTP1'
      *          '/' SINKING COST FUNCTION FORM IS',T50,'LINEAR')
      IF (IEQTP1.EQ.2) WRITE (IO,1000)
1000 FORMAT (' '//' IEQTP1'
      *          '/' SINKING COST FUNCTION FORM IS',T50,
      *          ' EXPONENTIAL')
      WRITE (IO,1010)
1010 FORMAT (' '//' TO CHANGE ENTRY ENTER THE CORRESPONDING INTEGER
      +',          '/' 1 = LINEAR'/' 2 = EXPONENTIAL')
      READ(IN,9000) ICARD

```

```

      CALL DCODEI(ICARD,IEQTP1)
      IF (IEQTP1.EQ.1) WRITE (IO,1020)
1020  FORMAT (' ' /'          FUNCTION = A + B * D'
*          /'          WHERE D = SFTDD(L)+2*SFTXR(L)'
*          /'          WHERE L=1,NST')
      IF (IEQTP1.EQ.2) WRITE (IO,1030)
1030  FORMAT (' ' /'          FUNCTION = A * EXP ( B * D )'
*          /'          WHERE D = SFTDD(L)+2*SFTXR(L)'
*          /'          WHERE L=1,NST')
      WRITE (IO,1040) SCCON1
1040  FORMAT (' ' /'          SCCON1'
*          /'          SINKING COST COEFFICIENT A  ',G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SCCON1)
      WRITE (IO,1050) SCCON2
1050  FORMAT (' ' /'          SCCON2'
*          /'          SINKING COST COEFFICIENT B  ',G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SCCON2)
      IF (IEQTP2.EQ.1) WRITE (IO,1060)
1060  FORMAT (' ' /'          IEQTP2'
*          /'          LINING COST FUNCTIONAL FORM IS ',T50,'LINEAR
*')
      IF (IEQTP2.EQ.2) WRITE (IO,1070)
1070  FORMAT (' ' /'          IEQTP2'
*          /'          LINING COST FUNCTIONAL FORM IS ',T50,
* 'EXPONENTIAL')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,IEQTP2)
      WRITE (IO,1010)
      IF (IEQTP2.EQ.1) WRITE (IO,1020)
      IF (IEQTP2.EQ.2) WRITE (IO,1030)
      WRITE (IO,1080) CCCON1
1080  FORMAT (' ' /'          CCCON1'
*          /'          LINING COST COEFFICIENT A  ',G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,CCCON1)
      WRITE (IO,1090) CCCON2
1090  FORMAT (' ' /'          CCCON2'
*          /'          LINING COST COEFFICIENT B  ',G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,CCCON2)
      IF (IEQTP3.EQ.1) WRITE (IO,1100) IEQTP3
1100  FORMAT (' ' /'          IEQTP3'
*          /'          WATER CONTROL COST FUNCTIONAL FORM IS '
* ',T50,'LINEAR')
      IF (IEQTP3.EQ.2) WRITE (IO,1110)
1110  FORMAT (' ' /'          IEQTP3'
*          /'          WATER CONTROL COST FUNCTIONAL FORM IS '

```

```

      * ,T50,'EXPONENTIAL')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,IEQTP3)
      WRITE (IO,1010)
      IF (IEQTP3.EQ.1) WRITE (IO,1020)
      IF (IEQTP3.EQ.2) WRITE (IO,1030)
      WRITE (IO,1120) WCCON1
1120  FORMAT (' '//'      WCCON1'
      *          /'      WATER CONTROL COST COEFFICIENT A  ',G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WCCON1)
      WRITE (IO,1130) WCCON2
1130  FORMAT (' '//'      WCCON2'
      *          /'      WATER CONTROL COST COEFFICIENT B  ',G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WCCON2)
C
      RETURN
      END
      SUBROUTINE RDSUPT
      INCLUDE PARAM
      INCLUDE OPTION
      INCLUDE INPUT
      CHARACTER*1
      +      BLANK
      DATA BLANK/1H /
C
9000  FORMAT(A80)
C
C      D.  SUPPORT FACILITIES DESCRIPTIONS
C
      IF (IOPTA.EQ.1) WRITE (IO,100)
100  FORMAT (' '//'      ***** SUPPORT FACILITIES *****'/
      +3X,76('=')/3X,76('='))
      WRITE (IO,110) SOA
110  FORMAT (' '//'      SOA'
      *          /'      SURFACE OPERATIONS AREA, ACRES',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SOA)
      WRITE (IO,120) NSC
120  FORMAT (' '//'      NSC (LIMIT 50)'
      *          /'      NUMBER OF SUPPORT FACILITY DESCRIPTIONS',
      +T50,I5)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,NSC)
300  IF (NSC.EQ.0) RETURN
      WRITE(IO,410) NSC
410  FORMAT(' '//'      THERE ARE CURRENTLY',I5,' SUPPORT FACILITIES'
      *          /'      FOR THIS CASE.  DO YOU WISH TO EDIT ALL FACILITIES'

```

```

* /'      OR A SPECIFIC FACILITY? ENTER THE APPROPRIATE'
* /'      RESPONSE '/'      0 = ALL'
* /'      1 = SPECIFIC FACILITY')
READ (IN,9010) N
9010 FORMAT(I1)
IF (N.NE.0) GOTO 320
M=1
MM=NSC
GOTO 310
320 CONTINUE
WRITE(IO,420)
420 FORMAT(' ' /'      WHICH FACILITY DO YOU WISH TO EDIT?'
* /'      ENTER THE FACILITY NUMBER')
READ (IN,9000) ICARD
CALL DCODEI(ICARD,NFCLT)
M=NFCLT
MM=NFCLT
310 DO 220 N=M,MM
WRITE(IO,1000) N
1000 FORMAT(' ' /T8,'SUPPORT FACILITY NO. 'I5/T8,26('-')/)
WRITE (IO,130) N,SFTITL(N)
130 FORMAT (' ' /'      SFTITL(N)      (FACILITY',I3,')'
* /'      SUPPORT FACILITY NAME, (LIMIT 48'
* ' CHARACTERS)' /,T5,A48)
READ(IN,9000) ICARD
IFLG = 0
DO 132 I=1,80
IF(ICARD(I:I).EQ.BLANK) GOTO 132
IFLG = 1
132 CONTINUE
IF(IFLG.EQ.0) GOTO 134
SFTITL(N) = ICARD
134 CONTINUE
WRITE (IO,140) N,SFLBRR(N)
140 FORMAT (' ' /'      SFLBRR(N)      (FACILITY',I3,')'
* /'      SUPPORT FACILITY LABOR',T50,G15.5
* /'      REQUIREMENT, MAN YEARS/FACILITY/SHIFT')
READ(IN,9000) ICARD
CALL DCODER(ICARD,SFLBRR(N))
WRITE (IO,150) N,SLBR(N)
150 FORMAT (' ' /'      SLBR(N)      (FACILITY',I3,')'
* /'      NUMBER OF SHIFTS PER FACILITY',T50,G15.5
* /'      (IF = -1 THE R-H OPERATIONS SHIFT'
* , ' REQUIREMENTS USED)'
* /'      (IF = -2 THE C-H OPERATIONS SHIFT',
* ' REQUIREMENTS USED)')
READ(IN,9000) ICARD
CALL DCODER(ICARD,SLBR(N))
WRITE (IO,160) N,SFEQPL(N)

```

```

160  FORMAT (' '/'          SFEQPL(N)          (FACILITY',I3,')'
*      '/'          SUPPORT FACILITY EQUIPMENT',T50,G15.5
*      '/'          LIFE, YEARS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFEQPL(N))
      WRITE (IO,170) N,SFCST(N)
170  FORMAT (' '/'          SFCST(N)          (FACILITY',I3,')'
*      '/'          SUPPORT FACILITY CAPITAL',T50,G15.5
*      '/'          COST, $')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFCST(N))
      WRITE (IO,180) N,SFLBRC(N)
180  FORMAT (' '/'          SFLBRC(N)          (FACILITY',I3,')'
*      '/'          SUPPORT FACILITY UNIT LABOR',T50,G15.5
*      '/'          COST, $/MAN-YEAR')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFLBRC(N))
      WRITE (IO,190) N,SFECST(N)
190  FORMAT (' '/'          SFECST(N)          (FACILITY',I3,')'
*      '/'          SUPPORT FACILITY UNIT',T50,G15.5
*      '/'          EQUIPMENT COST, $/FACILITY')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFECST(N))
      WRITE (IO,200) N,SFMATU(N)
200  FORMAT (' '/'          SFMATU(N)          (FACILITY',I3,')'
*      '/'          SUPPORT FACILITY MATERIAL',T50,G15.5
*      '/'          REQUIREMENTS, MATERIAL COSTS/FACILITY')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFMATU(N))
      WRITE (IO,210) N,SFMATC(N)
210  FORMAT (' '/'          SFMATC(N)          (FACILITY',I3,')'
*      '/'          SUPPORT FACILITY MATERIAL',T50,G15.5
*      '/'          COST, $/MATERIAL UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFMATC(N))
220  CONTINUE
      WRITE(IO,430)
430  FORMAT(' '/'          DO YOU WISH TO EDIT ANOTHER FACILITY?'
*      '/'          0 = NO'
*      '/'          1 = YES')
      READ (IN,9010) N
      IF (N.EQ.1) GOTO 320
C
      RETURN
      END
      SUBROUTINE RDUTLT
      INCLUDE PARAM
      INCLUDE OPTION
      INCLUDE INPUT

```

```

      CHARACTER*1
      +      BLANK
      DATA BLANK /1H /
C
C E. UTILITIES DESCRIPTION
C
9000 FORMAT(A80)
C
      IF (IOPTB(1,7).NE.1) GOTO 1000
C
C E1. ELECTRIC POWER REQUIREMENTS
C
      IF(IOPTA.EQ.1) WRITE (IO,100)
100 FORMAT (' '//' ***** UTILITY PARAMETERS AND COSTS *****'/
+3X,76('=')/3X,76('='))
      WRITE (IO,110)
110 FORMAT (' '//' ELECTRICAL'/5X,10('='))
      WRITE (IO,120) RAPP
120 FORMAT (' '//' RAPP (GENERIC - ALL)'
*      /' ELECTRICAL LOAD-RECEIVING',T50,G15.5
*      /' AND PACKAGING, KW/MODULE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,RAPP)
      WRITE (IO,130) SFSP
130 FORMAT (' '//' SFSP (GENERIC - ALL)'
*      /' ELECTRICAL LOAD-LAG STORAGE',T50,G15.5
*      /' POOL, KW')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFSP)
      WRITE (IO,140) OWHP
140 FORMAT (' '//' OWHP (GENERIC - ALL)'
*      /' ELECTRICAL LOAD-WASTE',T50,G15.5
+      /' HANDLING BUILDING, KW')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,OWHP)
      WRITE (IO,150) SFP
150 FORMAT (' '//' SFP (GENERIC - ALL)'
*      /' ELECTRICAL LOAD-SUPPORT',T50,G15.5
*      /' FACILITIES, KW')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SFP)
      WRITE (IO,160) MROP
160 FORMAT (' '//' MROP (REAL NUMBER) (GENERIC - ALL)'
*      /' ELECTRICAL LOAD-ROCK',T50,G15.5
*      /' OPERATIONS, KW/TONNE/YEAR')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,MROP)
      WRITE (IO,170) RVP
170 FORMAT (' '//' RVP (GENERIC - ALL)'

```

```

*          /'          ELECTRICAL LOAD-MINE',T50,G15.5
*          /'          VENTILATION, KW/ACRE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,RVP)
      WRITE (IO,180) ACP
180  FORMAT (' ' /'          ACP          (GENERIC - ALL)'
*          /'          ELECTRICAL LOAD-AIR',T50,G15.5
*          /'          CONDITIONING, KW')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ACP)
      WRITE (IO,190) UPC
190  FORMAT (' ' /'          UPC          (GENERIC - ALL)'
*          /'          ELECTRICITY UNIT COST,',T50,G15.5
*          /'          MILLS/KWH')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,UPC)
C
C  E2. STEAM REQUIREMENTS
C
1000 IF (IOPTB(2,7).NE.1) GOTO 1010
C
      WRITE (IO,200)
200  FORMAT (' ' /'          STEAM'/5X,5('='))
      WRITE (IO,210) HEATS
210  FORMAT (' ' /'          HEATS          (GENERIC - ALL)'
*          /'          STEAM REQUIREMENTS-UNDERGROUND',T50,G15.5
*          /'          HEATING, 10**3 LB/ACRE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,HEATS)
      WRITE (IO,220) RAPS
220  FORMAT (' ' /'          RAPS          (GENERIC - ALL)'
*          /'          STEAM REQUIREMENTS-RECEIVING',T50,G15.5
*          /'          FACILITY HEATING, 10**3 LB/MODULE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,RAPS)
      WRITE (IO,230) ORS
230  FORMAT (' ' /'          ORS          (GENERIC - ALL)'
*          /'          STEAM REQUIREMENT-OTHER',T50,G15.5
*          /'          FACILITIES, 10**3 LB')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,ORS)
      WRITE (IO,240) STMC
240  FORMAT (' ' /'          STMC          (GENERIC - ALL)'
*          /'          STEAM UNIT COST,',T50,G15.5
*          /'          $/10**3 LB')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,STMC)
C
C  E3. WATER REQUIREMENTS

```

```

C
1010 IF (IOPTB(3,7).NE.1) GOTO 1020
C
    WRITE (IO,250)
250 FORMAT (' '/' POTABLE WATER'/5X,13('='))
    WRITE (IO,260) VOLW
260 FORMAT (' '/' VOLW (GENERIC - ALL)'
*          /' POTABLE WATER REQUIREMENT',T50,G15.5
*          /' 10**6 LITERS')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,VOLW)
    WRITE (IO,270) WTRC
270 FORMAT (' '/' WTRC (GENERIC - ALL)'
*          /' POTABLE WATER UNIT COST',T50,G15.5
*          /' $/10**6 LITERS')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,WTRC)
C
C DEMINERALIZED WATER
C
1020 IF (IOPTB(4,7).NE.1) GOTO 1030
C
    WRITE (IO,275)
275 FORMAT (' '/' DEMINERALIZED WATER'/5X,19('='))
    WRITE (IO,280) VOLDW
280 FORMAT (' '/' VOLDW (GENERIC - ALL)'
*          /' DEMINERALIZED WATER REQUIREMENT',T50,G15.5
*          /' 10**6 LITERS')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,VOLDW)
    WRITE (IO,290) DWTRC
290 FORMAT (' '/' DWTRC (GENERIC - ALL)'
*          /' DEMINERALIZED WATER UNIT',T50,G15.5
*          /' COST, $/10**6 LITERS')
    READ(IN,9000) ICARD
    CALL DCODER(ICARD,DWTRC)
C
C E4. DIESEL FUEL REQUIREMENTS
C
1030 IF (IOPTB(5,7).NE.1) RETURN
C
    WRITE (IO,300)
300 FORMAT (' '/' DIESEL FUEL'/5X,11('='))
    WRITE (IO,310) SUD
310 FORMAT (' '/' SUD (GENERIC - ALL)'
*          /' DIESEL FUEL REQUIREMENT',T50,G15.5
*          /' SURFACE,LITERS/DAY')
    READ(IN,9000) ICARD

```



```

        CALL DCODER(ICARD,SUD)
        WRITE (IO,320) UGD
320  FORMAT (' ' /'          UGD          (GENERIC - ALL) '
*          /'          DIESEL FUEL REQUIREMENTS-',T50,G15.5
*          /'          UNDERGROUND, LITERS/WASTE UNIT')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,UGD)
        WRITE (IO,330) RED
330  FORMAT (' ' /'          RED          (GENERIC - ALL) '
*          /'          DIESEL FUEL REQUIREMENTS-RAIL',T50,G15.5
*          /'          USE, LITERS/SHIPPING CONTAINER')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,RED)
        WRITE (IO,340) DSLC
340  FORMAT (' ' /'          DSLC          (GENERIC - ALL) '
*          /'          DIESEL FUEL UNIT COST,',T50,G15.5
*          /'          $/LITER')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,DSLC)
C
        RETURN
        END
        SUBROUTINE RDWAST
C
C
        INCLUDE PARAM
        INCLUDE OPTION
        INCLUDE INPUT
        CHARACTER*1
+        BLANK
        DATA BLANK /1H /
C
9000  FORMAT(A80)
C
C II.  WASTE HANDLING SYSTEMS DESCRIPTION
C
C A.   GENERAL WASTE DESCRIPTION
C
        MCNT=0
        IRFLG = 0
        ICFLG = 0
        IF (IWNUM(1).EQ.0) INWT = NWT
        NUM = INUM(INWT)
        IF(IOPTA.EQ.1) WRITE (IO,100)
100  FORMAT (' ' //10X, ' ***** WASTE HANDLING SYSTEMS *****' /
+3X,76('=')/3X,76('='))
        DO 2290 KNUM=1,INWT
        K=INUM(KNUM)
C

```

```

        IF (IOPTB(1,2).NE.1) GOTO 5000
C
        WRITE (IO,110) K
110  FORMAT (' '//' GENERAL WASTE DESCRIPTION (WASTE TYPE',
        +I3,')'/5X,42('='))
        WRITE (IO,130) K,WSTYP(K)
130  FORMAT (' '//' WSTYP(K) (WASTE TYPE',I3,')'/
        +' DESCRIPTION: ',T50,A12)
        READ(IN,9000) ICARD
        IFLG = 0
        DO 135 I=1,80
        IF(ICARD(I:I).EQ.BLANK) GOTO 135
        IFLG = 1
135  CONTINUE
        IF(IFLG.EQ.0) GOTO 137
        WSTYP(K) = ICARD
137  CONTINUE
        WRITE (IO,140) K,KIND(K)
140  FORMAT (' '//' KIND(K) (WASTE TYPE',I3,')'/
        +' WASTE TYPE DESCRIPTION CODE = ',10X,I5/
        +3X,'WHERE :          1 = REMOTE HANDLED'/
        +'                   2 = CONTACT HANDLED'/
        +'                   3 = TRU, REMOTE HANDLED')
        READ(IN,9000) ICARD
        CALL DCODEI(ICARD,KIND(K))
        WRITE (IO,150) K,UNTMAS(K)
150  FORMAT (' '//' UNTMAS(K) (WASTE TYPE',I3,')'/
        +' UNIT MASS, KGU ',T50,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,UNTMAS(K))
        WRITE (IO,160) K,WVOLPU(K)
160  FORMAT (' '//' WVOLPU(K) (WASTE TYPE',I3,')'/
        +' WASTE VOLUME PER UNIT, M**3',T50,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,WVOLPU(K))
        WRITE (IO,170) K
170  FORMAT (' '//' TDWI(I,K) (WASTE TYPE',I3,', BY YEAR)'
        *      /' THERMAL DENSITY, KW/WASTE UNIT'/)
        L=LASTYR-IYRCON+1
        DO 190 I=1,L
        IYR=IYRCON+I-1
        WRITE (IO,180) IYR,TDWI(I,K)
180  FORMAT (' ',T8,'YEAR:',I5,3X,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,TDWI(I,K))
190  CONTINUE
C
C
5000  IF (KIND(K).EQ.1.OR.KIND(K).EQ.3) IRFLG=IRFLG+1

```

```
IF (KIND(K).EQ.2) ICFLG=ICFLG+1
IF (IOPTB(2,2).NE.1) GOTO 5010
```

```
C
C B. WASTE RECEIVING DESCRIPTION
C
```

```
WRITE (IO,200) K
200 FORMAT (' '/' WASTE RECEIVING PARAMETERS AND COSTS',
+ ' (WASTE TYPE',I3,') '/5X,52('='))
WRITE (IO,210)
210 FORMAT (' '/' PARAMETERS'/6X,10('-'))
IF (KIND(K).EQ.2) GO TO 240
IF (IRFLG.GT.1) GO TO 270
WRITE (IO,220) NRHRMA
220 FORMAT (' '/' NRHRMA (GENERIC: R-H WASTE)'
* '/' MAXIMUM NUMBER OF R-H WASTE RECEIVING',
* T50,I10 '/' MODULES ALLOWED ')
READ(IN,9000) ICARD
CALL DCODEI(ICARD,NRHRMA)
WRITE (IO,230) NRBPRM
230 FORMAT (' '/' NRBPRM (GENERIC: R-H WASTE)'
* '/' NUMBER OF RECEIVING BAYS PER MODULE',
* T50,I10)
READ(IN,9000) ICARD
CALL DCODEI(ICARD,NRBPRM)
GO TO 270
240 CONTINUE
IF (ICFLG.GT.1) GO TO 270
WRITE (IO,250) NCHRMA
250 FORMAT (' '/' NCHRMA (GENERIC: C-H WASTE)'
* '/' MAXIMUM NUMBER OF C-H WASTE RECEIVING',
* T50,I10 '/' MODULES ALLOWED ')
READ(IN,9000) ICARD
CALL DCODEI(ICARD,NCHRMA)
WRITE (IO,260) NCBPRM
260 FORMAT (' '/' NCBPRM (GENERIC: C-H WASTE)'
* '/' NUMBER OF RECEIVING BAYS PER MODULE',
* T50,I10)
READ(IN,9000) ICARD
CALL DCODEI(ICARD,NCBPRM)
270 CONTINUE
WRITE (IO,280)
280 FORMAT (' '/' AVERAGE RECEIVING TURNAROUND TIME IN '
* 'SHIFTS FOR')
WRITE (IO,290) K,TTTR(K)
290 FORMAT (' '/' TTTR(K) (WASTE TYPE',I3,')'
* '/' - RAIL CAR',T50,G15.5)
READ(IN,9000) ICARD
CALL DCODER(ICARD,TTTR(K))
WRITE (IO,300) K,TTTK(K)
```

```

300 FORMAT (' '/' TTTK(K) (WASTE TYPE',I3,')'
*          /' - TRUCK 'T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TTTK(K))
      WRITE (IO,310) K,RCCAP(K)
310 FORMAT (' '/' RCCAP(K) (WASTE TYPE',I3,')'
*          /' RAIL CAR CAPACITY,',T50,G15.5
*          /' WASTE UNITS / RAIL CAR')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,RCCAP(K))
      WRITE (IO,320) K,TCCAP(K)
320 FORMAT (' '/' TCCAP(K) (WASTE TYPE',I3,')'
*          /' TRUCK CAPACITY,',T50,G15.5
*          /' WASTE UNITS / TRUCK')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TCCAP(K))
      WRITE (IO,330) K
330 FORMAT (' '/' CRRAIL(K,I) (WASTE TYPE',I3,',' , BY YEAR)'
*          /' PERCENT OF WASTE UNITS SHIPPED BY RAIL,'
*          ' FRACTION')
      II=LASTYR-IYRCON+1
      DO 350 I=1,II
        IYR=IYRCON+I-1
        WRITE (IO,180) IYR,CRRAIL(K,I)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,CRRAIL(K,I))
350 CONTINUE
      IF (KIND(K).EQ.2) GO TO 410

```

C
C
C

```

      REMOTE HANDLED

      IF (IRFLG.GT.1) GO TO 390
      WRITE (IO,360) WOLBRR(1)
360 FORMAT (' '/' WOLBRR(1) (GENERIC: R-H WASTE)'
*          /' WASTE RECEIVING LABOR REQUIREMENT ',
*          'T50,G15.5 /' MAN YEARS/SHIFT/MODULE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOLBRR(1))
      WRITE (IO,370) WOEQPL(1)
370 FORMAT (' '/' WOEQPL(1) (GENERIC: R-H WASTE)'
*          /' WASTE RECEIVING EQUIPMENT LIFE',T50,G15.3
*          /' WASTE UNITS ')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOEQPL(1))
      WRITE (IO,380) WOEQPC(1)
380 FORMAT (' '/' WOEQPC(1) (GENERIC: R-H WASTE)'
*          /' WASTE RECEIVING EQUIPMENT CAPACITY',T50,
*          'G15.5, /' WASTE UNITS/EQUIPMENT UNIT/SHIFT')
      READ(IN,9000) ICARD

```

```

      CALL DCODER(ICARD,WOEQPC(1))
390  CONTINUE
      WRITE (IO,400) K,WOMATU(1,K)
400  FORMAT (' '//'      WOMATU(1,K) (R-H WASTE TYPE (' ,I2,' ) )'
*      /'      WASTE RECEIVING MATERIAL REQUIREMENT,'
* T50,G15.5 /'      MATERIAL UNIT/WASTE UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOMATU(1,K))
      GO TO 470

C
C CONTACT HANDLED
C
410  CONTINUE
      IF (ICFLG.GT.1) GO TO 450
      WRITE (IO,420) WOLBRR(2)
420  FORMAT (' '//'      WOLBRR(2) (GENERIC: C-H WASTE)'
*      /'      C-H RECEIVING UNIT LABOR REQUIREMENT,'
* ,T50,G15.5 /'      MAN YEARS/SHIFT/MODULE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOLBRR(2))
      WRITE (IO,430) WOEQPL(2)
430  FORMAT (' '//'      WOEQPL(2) (GENERIC: C-H WASTE)'
*      /'      C-H RECEIVING UNIT EQUIPMENT LIFE,',
* T50,G15.5 /'      RECEIVING UNITS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOEQPL(2))
      WRITE (IO,440) WOEQPC(2)
440  FORMAT (' '//'      WOEQPC(2) (GENERIC: C-H WASTE)'
*      /'      C-H RECEIVING EQUIPMENT CAPACITY,'
* ,T50,G15.5 /'      RECEIVING UNITS/EQUIPMENT UNIT/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOEQPC(2))
450  CONTINUE
      WRITE (IO,460) K,WOMATU(2,K)
460  FORMAT (' '//'      WOMATU(2,K) (C-H WASTE TYPE (' ,I2,' ) )'
*      /'      C-H MATERIALS REQUIREMENT, ' ,T50,G15.5
*      /'      MATERIALS UNIT/RECEIVING UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOMATU(2,K))
      GO TO 540

C
C C. WASTE RECEIVING COST DATA
C
C REMOTE HANDLED
C
470  CONTINUE
      WRITE (IO,480)
480  FORMAT (' '//'      COSTS'/6X,5('-'))
      IF (IRFLG.GT.1) GO TO 520

```

```

      WRITE (IO,490) WOCST(1)
490 FORMAT (' /' WOCST(1) (GENERIC: R-H WASTE)'
*          /' R-H RECEIVING MODULE CAPITAL COST, ',T50,
* G15.3 /' $/MODULE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOCST(1))
      WRITE (IO,500) WOLBRC(1)
500 FORMAT (' /' WOLBRC(1) (GENERIC: R-H WASTE)'
*          /' R-H RECEIVING MODULE LABOR COST, ',T50,
* G15.3 /' $/MAN-YEAR')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOLBRC(1))
      WRITE (IO,510) WOECST(1)
510 FORMAT (' /' WOECST(1) (GENERIC: R-H WASTE)'
*          /' R-H RECEIVING MODULE EQUIPMENT ',T50,
* G15.3 /' COSTS, $/EQUIPMENT UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOECST(1))
520 CONTINUE
      WRITE (IO,530) K,WOMATC(1,K)
530 FORMAT (' /' WOMATC(1,K) (R-H WASTE TYPE (' ,I2,') )'
*          /' R-H RECEIVING MODULE MATERIALS ',T50,
* G15.5 /' COSTS, $/MATERIAL UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOMATC(1,K))
      WOMATC(2,K) = 0.0
      GO TO 600

C
C CONTACT HANDLED
C
540 CONTINUE
      WRITE (IO,480)
      IF (ICFLG.GT.1) GO TO 580
      WRITE (IO,550) WOCST(2)
550 FORMAT (' /' WOCST(2) (GENERIC: C-H WASTE)'
*          /' C-H RECEIVING MODULE CAPITAL COST, ',T50,
* G15.5, /' $/MODULE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOCST(2))
      WRITE (IO,560) WOLBRC(2)
560 FORMAT (' /' WOLBRC(2) (GENERIC: C-H WASTE)'
*          /' C-H RECEIVING MODULE LABOR COST, ',T50,
* G15.3 /' $/MAN-YEAR')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOLBRC(2))
      WRITE (IO,570) WOECST(2)
570 FORMAT (' /' WOECST(2) (GENERIC: C-H WASTE)'
*          /' C-H RECEIVING MODULE EQUIPMENT COST, ',
* T50,G15.5, /' $/EQUIPMENT UNIT')

```

```

        READ(IN,9000) ICARD
        CALL DCODER(ICARD,WOECST(2))
580    CONTINUE
        WRITE (IO,590) K,WOMATC(2,K)
590    FORMAT (' '/'      WOMATC(2,K) (C-H WASTE TYPE (' ,I2,' ) )'
*          '/'      C-H RECEIVING MODULE MATERIALS ',T50,
* G15.5      '/'      COSTS, $/MATERIAL UNIT')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,WOMATC(2,K))
        WOMATC(1,K) = 0.0
600    CONTINUE
C
C
5010   IF (IOPTB(3,2).NE.1) GOTO 5020
C
C   D. WASTE PACKAGING DESCRIPTION
C
        WRITE (IO,610) K
610    FORMAT (' '/'      WASTE PACKAGING PARAMETERS AND COSTS',
+ ' (WASTE TYPE',I3,')' /5X,53('='))
        WRITE (IO,620)
620    FORMAT (' '/'      PARAMETERS' /6X,10('-'))
        WRITE (IO,630) K,PKGDS1(K)
630    FORMAT (' '/'      PKGDS1(K) (WASTE TYPE',I3,')'
*          '/'      PACKAGE DESCRIPTION,',T50,A16
*          '/'      OUTPUT LINE ONE (LIMIT 16 CHARACTERS)')
        READ(IN,9000) ICARD
        IFLG = 0
        DO 632 I=1,80
        IF(ICARD(I:I).EQ.BLANK) GOTO 632
        IFLG = 1
632    CONTINUE
        IF(IFLG.EQ.0) GOTO 634
        PKGDS1(K) = ICARD
634    CONTINUE
        WRITE (IO,640) K,PKGDS2(K)
640    FORMAT (' '/'      PKGDS2(K) (WASTE TYPE',I3,')'
*          '/'      PACKAGE DESCRIPTION,',T50,A16
*          '/'      OUTPUT LINE TWO (LIMIT 16 CHARACTERS)')
        READ(IN,9000) ICARD
        IFLG = 0
        DO 642 I=1,80
        IF(ICARD(I:I).EQ.BLANK) GOTO 642
        IFLG = 1
642    CONTINUE
        IF(IFLG.EQ.0) GOTO 644
        PKGDS2(K) = ICARD
644    CONTINUE
        WRITE (IO,650) K,PKGDS3(K)

```

```

650 FORMAT (' '/' PKGDS3(K) (WASTE TYPE',I3,')'
*          /' PACKAGE DESCRIPTION,',T50,A16
*          /' OUTPUT LINE THREE (LIMIT 16 CHARACTERS)')
      READ(IN,9000) ICARD
      IFLG = 0
      DO 652 I=1,80
      IF(ICARD(I:I).EQ.BLANK) GOTO 652
      IFLG = 1
652    CONTINUE
      IF(IFLG.EQ.0) GOTO 654
      PKGDS3(K) = ICARD
654    CONTINUE
      WRITE (IO,660) K,WUPP(K)
660    FORMAT (' '/' WUPP(K) (WASTE TYPE',I3,')'
*          /' NUMBER OF WASTE UNITS PER PACKAGE ',T50,
*          G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WUPP(K))
      IF (KIND(K).EQ.2) GOTO 735
      WRITE (IO,680) K,PKGCAP(K)
680    FORMAT (' '/' PKGCAP(K) (RH WASTE TYPE',I3,')'
*          /' WASTE MODULE PACKAGING CAPACITY',T50,
*          G15.5 /' PACKAGE/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,PKGCAP(K))
      IF (IRFLG.GT.1) GO TO 720
      WRITE (IO,690) WOLBRR(3)
690    FORMAT (' '/' WOLBRR(3) (GENERIC: R-H)'
*          /' WASTE PACKAGING LABOR REQUIREMENT,',T50,
*          G15.5 /' MAN-YEARS/SHIFT/MODULE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOLBRR(3))
      WRITE (IO,700) WOEQPL(3)
700    FORMAT (' '/' WOEQPL(3) (GENERIC: R-H)'
*          /' WASTE PACKAGING EQUIPMENT LIFE,',T50,
*          G15.5 /' PACKAGING UNITS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOEQPL(3))
      WRITE (IO,710) WOEQPC(3)
710    FORMAT (' '/' WOEQPC(3) (GENERIC: R-H)'
*          /' WASTE PACKAGING EQUIPMENT CAPACITY,',T50,
*          G15.5 /' PACKAGING UNITS/EQUIPMENT UNIT/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOEQPC(3))
720    CONTINUE
      WRITE (IO,730) K,WOMATU(3,K)
730    FORMAT (' '/' WOMATU(3,K) (R-H WASTE TYPE',I3,')'
*          /' WASTE PACKAGING MATERIAL REQUIREMENTS,',
*          T50,G15.5 /' MATERIAL UNITS/PACKAGE')

```



```

      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOMATU(3,K))
      GOTO 738
735  CONTINUE
      PKGCAP(K) = 0.0
      WOMATU(3,K) = 0.0
      WOMATC(3,K) = 0.0
738  CONTINUE
      WRITE (IO,740) K,PKGDI(K)
740  FORMAT (' '//'      PKGDI(K)      (WASTE TYPE',I3,')'
*         '/'      PACKAGE DIAMETER, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,PKGDI(K))
      WRITE (IO,750) K,REFDI(K)
750  FORMAT (' '//'      REFDI(K)      (WASTE TYPE',I3,')'
*         '/'      REFERENCE PACKAGE DIAMETER USED FOR',T50,
* G15.5      '/'      PACKAGE AND HOLE COST, METERS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,REFDI(K))
      WRITE (IO,760) K,PKGLN(K)
760  FORMAT (' '//'      PKGLN(K)      (WASTE TYPE',I3,')'
*         '/'      PACKAGE LENGTH, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,PKGLN(K))
C
C  E.  WASTE PACKAGING COST DATA
C
      IF (KIND(K).EQ.2) GOTO 5020
      WRITE (IO,770)
770  FORMAT (' '///      COSTS'/6X,5(' - '))
      IF (IRFLG.GT.1) GO TO 810
      WRITE (IO,780) WOCST(3)
780  FORMAT (' '//'      WOCST(3)      (GENERIC: R-H)'
*         '/'      WASTE PACKAGING MODULE CAPITAL COST,',T50,
* G15.3      '/'      $/MODULE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOCST(3))
      WRITE (IO,790) WOLBRC(3)
790  FORMAT (' '//'      WOLBRC(3)      (GENERIC: R-H)'
*         '/'      WASTE PACKAGING MODULE LABOR COST,',T50,
* G15.3      '/'      $/MAN-YEAR')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOLBRC(3))
      WRITE (IO,800) WOECST(3)
800  FORMAT (' '//'      WOECST(3)      (GENERIC: R-H)'
*         '/'      WASTE PACKAGING MODULE EQUIPMENT COST,
* ,T50,G15.3 '/'      $/EQUIP UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOECST(3))

```

```

810  CONTINUE
      WRITE (IO,820) K,WOMATC(3,K)
820  FORMAT (' ' /' ' WOMATC(3,K) (R-H WASTE TYPE',I3,'')'
*      /' ' WASTE PACKAGING MODULE MATERIAL COST,'
*      ,T50,G15.5, /' ' $/MATERIAL UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WOMATC(3,K))
C
C
5020 IF (IOPTB(4,2).NE.1) GOTO 5030
C
C  F  TRANSFER OPERATIONS DESCRIPTION
C
      WRITE (IO,830) K
830  FORMAT (' ' /' ' WASTE TRANSFER PARAMETERS AND COSTS',
+ ' (WASTE TYPE',I3,'') /5X,52('='))
      IF (KIND(K).EQ.2) GO TO 890
      IF (IRFLG.GT.1) GO TO 870
C
C  REMOTE HANDLED
C
      WRITE (IO,840)
840  FORMAT (' ' /' ' PARAMETERS' /6X,10('-'))
      WRITE (IO,850) TOEQPL(1)
850  FORMAT (' ' /' ' TOEQPL(1) (GENERIC: R-H WASTE)'
*      /' ' R-H TRANSPORTATION UNIT',T50,G15.5
*      /' ' EQUIPMENT LIFE, PACKAGES')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TOEQPL(1))
      WRITE (IO,860) TOEQPC(1)
860  FORMAT (' ' /' ' TOEQPC(1) (GENERIC: R-H WASTE)'
*      /' ' R-H TRANSPORTATION UNIT',T50,G15.5
*      /' ' EQUIPMENT CAPACITY,'
*      ' PACKAGES/EQUIPMENT-UNIT/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TOEQPC(1))
      WRITE (IO,950) TOLBRR(1)
950  FORMAT (' ' /' ' TOLBRR(1) (GENERIC: R-H WASTE)'
*      /' ' R-H TRANSPORTATION UNIT',T50,G15.5
*      /' ' LABOR REQUIREMENT, MAN-YEARS/EQUIPMENT UNIT
+ ')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TOLBRR(1))
870  CONTINUE
      WRITE (IO,880) K,TOMATU(1,K)
880  FORMAT (' ' /' ' TOMATU(1,K) (R-H WASTE TYPE (' ,I2,' ) )'
*      /' ' R-H TRANSPORTATION MATERIAL',T50,G15.5
*      /' ' REQUIREMENT MATERIAL UNIT/PACKAGE')
      READ(IN,9000) ICARD

```

```

      CALL DCODER(ICARD,TOMATU(1,K))
      GOTO 960
C
C CONTACT HANDLED
C
890  CONTINUE
      IF (ICFLG.GT.1) GO TO 920
      WRITE (IO,900) TOEQPL(2)
900  FORMAT (' '/' TOEQPL(2) (GENERIC: C-H WASTE)'
*          /' C-H TRANSPORTATION UNIT',T50,G15.5
*          /' EQUIPMENT LIFE, PACKAGES')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TOEQPL(2))
      WRITE (IO,910) TOEQPC(2)
910  FORMAT (' '/' TOEQPC(2) (GENERIC: C-H WASTE)'
*          /' C-H TRANSPORTATION UNIT',T50,G15.5
*          /' EQUIPMENT CAPACITY, PACKAGES/EQUIPMENT',
*          /' UNIT/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TOEQPC(2))
      WRITE (IO,970) TOLBRR(2)
970  FORMAT (' '/' TOLBRR(2) (GENERIC: C-H WASTE)'
*          /' C-H TRANSPORTATION UNIT',T50,G15.5
*          /' LABOR REQUIREMENT,MAN-YEARS/EQUIPMENT UNIT'
+ )
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TOLBRR(2))
920  CONTINUE
      WRITE (IO,930) K,TOMATU(2,K)
930  FORMAT (' '/' TOMATU(2,K) (C-H WASTE TYPE ('I2,') )'
*          /' C-H TRANSPORTATION UNIT',T50,G15.5
*          /' MATERIAL REQUIREMENT, MATERIAL UNITS'
*          '/PACKAGE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TOMATU(2,K))
      IF (ICFLG.GT.1) GO TO 1050
960  CONTINUE
C
C G. TRANSFER OPERATIONS COST DATA
C
C REMOTE HANDLED
C
980  CONTINUE
      WRITE (IO,990)
990  FORMAT (' '/' COSTS'/6X,5('-'))
C
      IF (KIND(K).EQ.2) GOTO 1050
      IF (IRFLG.GT.1) GO TO 1030

```

```

        WRITE (IO,1000) TOCST(1)
1000 FORMAT (' /' TOCST(1) (GENERIC: R-H WASTE)'
*          /' R-H WASTE TRANSFER CAPITAL COST, ',T50,
* G15.5 /' $ ')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,TOCST(1))
        WRITE (IO,1010) TOLBRC(1)
1010 FORMAT (' /' TOLBRC(1) (GENERIC: R-H WASTE)'
*          /' R-H WASTE TRANSFER UNIT',T50,G15.5
*          /' LABOR COST, $/MAN YEAR')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,TOLBRC(1))
        WRITE (IO,1020) TOECST(1)
1020 FORMAT (' /' TOECST(1) (GENERIC: R-H WASTE)'
*          /' R-H WASTE TRANSFER UNIT',T50,G15.5
*          /' EQUIPMENT COST, $/EQUIPMENT UNIT')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,TOECST(1))
1030 CONTINUE
        WRITE (IO,1040) K,TOMATC(1,K)
1040 FORMAT (' /' TOMATC(1,K) (R-H WASTE TYPE (' ,I2,') )'
*          /' R-H WASTE TRANSFER MATERIALS',T50,G15.5
*          /' COST, $')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,TOMATC(1,K))
        GO TO 5030
1050 CONTINUE
C
C CONTACT HANDLED
C
        WRITE (IO,830)
        IF (ICFLG.GT.1) GO TO 1090
        WRITE (IO,1060) TOCST(2)
1060 FORMAT (' /' TOCST(2) (GENERIC: C-H WASTE)'
*          /' C-H WASTE TRANSPORTATION',T50,G15.5
*          /' CAPITAL COST, $')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,TOCST(2))
        WRITE (IO,1070) TOLBRC(2)
1070 FORMAT (' /' TOLBRC(2) (GENERIC: C-H WASTE)'
*          /' C-H WASTE TRANSFER UNIT',T50,G15.5
*          /' LABOR COST, $/MAY-YEAR')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,TOLBRC(2))
        WRITE (IO,1080) TOECST(2)
1080 FORMAT (' /' TOECST(2) (GENERIC: C-H WASTE)'
*          /' C-H EMPLACEMENT UNIT',T50,G15.5
*          /' EQUIPMENT COST, $/EQUIPMENT UNIT')
        READ(IN,9000) ICARD

```

```

      CALL DCODER(ICARD,TOECST(2))
1090  CONTINUE
      WRITE (IO,1100) K,TOMATC(2,K)
1100  FORMAT (' '/' TOMATC(2,K) (C-H WASTE TYPE (' ,I2,' ) )'
*      /' C-H WASTE TRANSFER UNIT',T50,G15.5
*      /' MATERIALS COST, $/MATERIAL UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TOMATC(2,K))
C
5030  IF (IOPTB(5,2).NE.1) GO TO 2290
C
C
C  H.  EMPLACEMENT OPERATIONS DESCRIPTION
C
1110  CONTINUE
      WRITE (IO,1120) K
1120  FORMAT (' '/' WASTE EMPLACEMENT PARAMETERS AND COSTS',
+ ' WASTE TYPE',I3,'')
* / (NOTE: PRIMARY WASTE TYPES MUST BE FIRST)'
+ /5X,53('='))
      WRITE (IO,1130)
1130  FORMAT (' '/' PARAMETERS'/6X,10('-'))
      WRITE(IO,1152) K
1152  FORMAT(' '/' IMET(K) (WASTE TYPE',I3,'')')
      WRITE(IO,1150) IMET(K)
1150  FORMAT(' '/' EMPLACEMENT METHOD OPTION = ',I3,', WHERE'/
+ ' PRIMARY SECONDARY '
+ ' TERTIARY'/
+ ' 1 = VERTICAL BOREHOLE NONE '
+ ' NONE'/
+ ' 2 = ROOM EMPLACEMENT NONE '
+ ' NONE'/
+ ' 3 = TRENCHING NONE '
+ ' NONE'/
+ ' 4 = HORIZONTAL BOREHOLE NONE '
+ ' NONE '/
+ ' 5 = VERTICAL BOREHOLE TRU IN VBH'
+ ' NONE'/
+ ' 6 = VERTICAL BOREHOLE TRU IN VBH'
+ ' CH IN RE'/
+ ' 7 = VERTICAL BOREHOLE CH IN RE '
+ ' NONE'/
+ ' 8 = HORIZONTAL BOREHOLE TRU IN HBH'
+ ' NONE'/
+ ' 9 = HORIZONTAL BOREHOLE TRU IN HBH'
+ ' CH IN RE')
      WRITE(IO,1154)
1154  FORMAT(' 10 = HORIZONTAL BOREHOLE CH IN RE '
+ ' NONE'/

```

```

+      '      11 = HORIZONTAL BOREHOLE      TRU IN T '
+      '      NONE'/'
+      '      12 = HORIZONTAL BOREHOLE      TRU IN T '
+      '      CH IN RE'/'
+      '      13 = ROOM EMPLACEMENT        TRU IN RE '
+      '      NONE'/'
+      '      14 = ROOM EMPLACEMENT        CH IN RE '
+      '      NONE'/'
+      '      15 = ROOM EMPLACEMENT        TRU IN RE '
+      '      CH IN RE'/'
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,IMET(K))
      IF (K.GT.1.AND.MCNT.GE.1) GO TO 1200
      MCNT=MCNT+1
      WRITE (IO,1170) MPRIME
1170 FORMAT (' '/' MPRIME (GENERIC: ALL)'
*          /' NUMBER OF PRIMARY HEAT GENERATING',T50,I10
*          /' WASTE TYPES')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,MPRIME)
      WRITE (IO,1180) NNHG
1180 FORMAT (' '/' NNHG (GENERIC: ALL)'
*          /' NUMBER OF NON-HEAT GENERATING',T50,I10
*          /' WASTE TYPES')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,NNHG)
      WRITE (IO,1190) ICLR
1190 FORMAT (' '/' ICLR (GENERIC: ALL)'
*          /' THE YEAR INVENTORIES OF STORED NON-HEAT',
* T50,I10 /' GENERATING WASTE CAN BE CLEARED BY SOLO',
*          /' EMPLACEMENT')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,ICLR)
1200 CONTINUE
      IF (NNHG.EQ.0) GO TO 1300
      WRITE (IO,1210) K,STOMAX(K)
1210 FORMAT (' '/' STOMAX(K) (WASTE TYPE',I3,')'
*          /' THE STORAGE CAPACITY LIMIT FOR',T50,
* G15.5 /' SECONDARY WASTE (PACKAGES)')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,STOMAX(K))
      WRITE (IO,1220)
1220 FORMAT (' '/' NUMBER OF NON-HEAT GENERATING WASTE'
*          /' PACKAGES THAT CAN BE COEMPLACED PER'
*          /' PRIMARY WASTE PACKAGE AND FRACTION'
*          /' SECONDARY WASTE EMPLACEMENT CAPACITY'
*          /' AVAILABLE FOR THIS SECONDARY WASTE TYPE')
      IF (K.GT.MPRIME) GO TO 1310
      WRITE (IO,1230) K

```

```

1230 FORMAT (' ' / '          PRIMARY WASTE TYPE ',I5)
      DO 1260 M=1,NNHG
      MM=MPRIME+M
      WRITE (IO,1240) MM
1240  FORMAT (' ' / '          FOR SECONDARY WASTE TYPE',I5)
      WRITE (IO,1250) COCON(M,K)
1250  FORMAT (' ' / '          COCON(MM,K) (BY SECONDARY WASTE TYPE BY '
*      ' PRIMARY WASTE TYPE)'
*      / '          NUMBER OF PACKAGES PER',T50,G15.5
*      / '          PRIMARY PACKAGE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,COCON(M,K))
      WRITE(IO,1255) PCTCON(M,K)
1255  FORMAT(' ' / '          PCTCON(MM,K) (BY SECONDARY WASTE TYPE BY '
*      ' PRIMARY WASTE TYPE)'
*      / '          FRACTION OF SECONDARY CAPACITY',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,PCTCON(M,K))
1260  CONTINUE
      WRITE (IO,1270)
1270  FORMAT (' ' / '          SWPPR(MM) (BY SECONDARY WASTE)' )
      DO 1280 J=MPRIME+1,NWT
      WRITE (IO,1272) J,SWPPR(J)
1272  FORMAT (' ' / '          SECONDARY WASTE NO. 'I5
*      / '          SECONDARY WASTE PACKAGES PER ROOM',T50
*      / '          SOLO EMPLACEMENT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SWPPR(J))
1280  CONTINUE
      IF (KIND(K).EQ.1.OR.KIND(K).EQ.3) GO TO 1300
      IF (IMET(K).LE.4.OR.IMET(K).EQ.7.OR.IMET(K).EQ.10.OR.IMET(K).GT.
+      12) GO TO 1310
      WRITE (IO,1290) CONTRU(K)
1290  FORMAT (' ' / '          CONTRU(M) (BY SECONDARY C-H WASTE)'
*      / '          NUMBER OF C-H PACKAGES THAT',T50,G15.5
*      / '          CAN BE EMPLACED PER TRU WASTE PACKAGE'
*      / '          FOR PRIMARY HG WASTE TYPE',I5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,CONTRU(K))
1300  CONTINUE
1310  CONTINUE
      WRITE (IO,1320) K,NPKPEU(K)
1320  FORMAT (' ' / '          NPKPEU(K) (WASTE TYPE',I3,')'
*      / '          NUMBER OF PACKAGES PER',T50,I10
*      / '          EMPLACEMENT UNIT')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,NPKPEU(K))
      WRITE (IO,1330) K,PKGUP(K)
1330  FORMAT (' ' / '          PKGUP(K) (WASTE TYPE',I3,')'

```

```

*          /'          EMPLACEMENT UNITS PER UNIT PITCH',T50,G15.5
*          /'          PER ROW')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,PKGUP(K))
      IF (KIND(K).EQ.2) GO TO 1350
      IF (IRFLG.GT.1) GO TO 1420
      WRITE (IO,1340) UOLBRR(1)
1340 FORMAT (' ' /'          UOLBRR(1)    (GENERIC: R-H WASTE)'
*          /'          R-H WASTE EMPLACEMENT LABOR',T50,G15.5
*          /'          REQUIREMENT, MAN-YEARS/EQUIPMENT UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,UOLBRR(1))
      GO TO 1370
1350 CONTINUE
      IF (ICFLG.GT.1) GO TO 1420
      WRITE (IO,1360) UOLBRR(2)
1360 FORMAT (' ' /'          UOLBRR(2)    (GENERIC: C-H WASTE)'
*          /'          C-H WASTE EMPLACEMENT LABOR',T50,G15.5
*          /'          REQUIREMENT, MAN-YEARS/EQUIPMENT UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,UOLBRR(2))
      GO TO 1400
1370 CONTINUE
      WRITE (IO,1380) UOEQPL(1)
1380 FORMAT (' ' /'          UOEQPL(1)    (GENERIC: R-H WASTE)'
*          /'          R-H WASTE EMPLACEMENT',T50,G15.5
*          /'          EQUIPMENT LIFE, PACKAGES')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,UOEQPL(1))
      WRITE (IO,1390) UOEQPC(1)
1390 FORMAT (' ' /'          UOEQPC(1)    (GENERIC: R-H WASTE)'
*          /'          R-H WASTE EMPLACEMENT',T50,G15.5
*          /'          EQUIPMENT CAPACITY, PACKAGES/EQUIPMENT
*          ' UNIT/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,UOEQPC(1))
      GO TO 1420
1400 CONTINUE
      WRITE (IO,1382) UOEQPL(2)
1382 FORMAT (' ' /'          UOEQPL(2)    (GENERIC: C-H WASTE)'
*          /'          C-H WASTE EMPLACEMENT',T50,G15.5
*          /'          EQUIPMENT LIFE, PACKAGES')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,UOEQPL(2))
      WRITE (IO,1410) UOEQPC(2)
1410 FORMAT (' ' /'          UOEQPC(2)    (GENERIC: C-H WASTE)'
*          /'          C-H WASTE EMPLACEMENT',T50,G15.5
*          /'          EQUIPMENT CAPACITY, PACKAGES/EQUIPMENT
*          ' UNIT/SHIFT')

```



```

      READ(IN,9000) ICARD
      CALL DCODER(ICARD,UOEQPC(2))
1420  CONTINUE
      WRITE (IO,1430) K,CLSEMP(K)
1430  FORMAT (' ' /'          CLSEMP(K) (WASTE TYPE',I3,')'
*          /'          CLOSEST EMPLACEMENT TO',T50,G15.5
*          /'          CORRIDOR, METERS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,CLSEMP(K))
      WRITE (IO,1440) K,NRWSPR(K)
1440  FORMAT (' ' /'          NRWSPR(K) (WASTE TYPE',I3,')'
*          /'          NUMBER OF ROWS PER ROOM',T50,I10)
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,NRWSPR(K))
      WRITE (IO,1450) K,WABATH(K)
1450  FORMAT (' ' /'          WABATH(K) (WASTE TYPE',I3,')'
*          /'          WATER BARRIER MATERIAL',T50,G15.5
*          /'          THICKNESS, METERS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,WABATH(K))
      WRITE (IO,1460) K,SLEETH(K)
1460  FORMAT (' ' /'          SLEETH(K) (WASTE TYPE',I3,')'
*          /'          SLEEVE THICKNESS, METERS
+          ,T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,SLEETH(K))
      WRITE (IO,1470) K,BCKFTH(K)
1470  FORMAT (' ' /'          BCKFTH(K) (WASTE TYPE',I3,')'
*          /'          BACKFILL MATERIAL THICKNESS,',T50,G15.5
*          /'          METERS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,BCKFTH(K))
      WRITE (IO,1480) K,HOLDEL(K)
1480  FORMAT (' ' /'          HOLDEL(K) (WASTE TYPE',I3,')'
*          /'          INCREMENTAL HOLE DEPTH BEYOND',T50,G15.5
*          /'          PACKAGE LENGTH, METERS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,HOLDEL(K))
      WRITE (IO,1490) K,CAPLEN(K)
1490  FORMAT (' ' /'          CAPLEN(K) (WASTE TYPE',I3,')'
*          /'          LENGTH OF EMPLACEMENT HOLE',T50,G15.5
*          /'          PLUG, METERS')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,CAPLEN(K))
C
C  I.  EMPLACEMENT OPERATIONS COST DATA
C
      WRITE (IO,1500)
1500  FORMAT (' ' /'          COST'/6X,5('-'))

```

```

        IF (KIND(K).EQ.2) GO TO 1520
        IF (IRFLG.GT.1) GO TO 1620
        WRITE (IO,1510) UOCST(1)
1510 FORMAT (' ' /'          UOCST(1) (GENERIC: R-H WASTE)'
*          /'          R-H WASTE EMPLACEMENT',T50,G15.5
*          /'          CAPITAL COST, $')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,UOCST(1))
        GO TO 1540
1520 CONTINUE
        IF (ICFLG.GT.1) GO TO 1620
        WRITE (IO,1530) UOCST(2)
1530 FORMAT (' ' /'          UOCST(2) (GENERIC: C-H WASTE)'
*          /'          C-H WASTE EMPLACEMENT CAPITAL',T50,G15.5
*          /'          COST, $')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,UOCST(2))
        GO TO 1560
1540 CONTINUE
        WRITE (IO,1550) UOLBRC(1)
1550 FORMAT (' ' /'          UOLBRC(1) (GENERIC: R-H WASTE)'
*          /'          R-H WASTE EMPLACEMENT UNIT',T50,G15.5
*          /'          LABOR COST, $/MAN-YEAR')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,UOLBRC(1))
        GO TO 1580
1560 CONTINUE
        WRITE (IO,1570) UOLBRC(2)
1570 FORMAT (' ' /'          UOLBRC(2) (GENERIC: C-H WASTE)'
*          /'          C-H WASTE EMPLACEMENT UNIT',T50,G15.5
*          /'          LABOR COST, $/MAN-YEAR')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,UOLBRC(2))
        GO TO 1600
1580 CONTINUE
        WRITE (IO,1590) UOECST(1)
1590 FORMAT (' ' /'          UOECST(1) (GENERIC: R-H WASTE)'
*          /'          R-H WASTE EMPLACEMENT UNIT',T50,G15.5
*          /'          EQUIPMENT COST, $/EQUIPMENT UNIT')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,UOECST(1))
        GO TO 1620
1600 CONTINUE
        WRITE (IO,1610) UOECST(2)
1610 FORMAT (' ' /'          UOECST(2) (GENERIC: C-H WASTE)'
*          /'          C-H WASTE EMPLACEMENT UNIT',T50,G15.5
*          /'          EQUIPMENT COST, $/EQUIPMENT UNIT')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,UOECST(2))

```

```

1620  CONTINUE
      IF (K.GT.1) GO TO 1640
      WRITE (IO,1630) IDET
1630  FORMAT (' ' /' ' IDET (GENERIC: ALL)'
*          /' ' INDIVIDUAL MATERIAL COST OPTION ='T55,I3,
*          ' , WHERE' /' ' 0 = AGGREGATE EMPLACEMENT COSTS'
*          /' ' 1 = DETAILED EMPLACEMENT MATERIALS'
*          /' ' DESCRIPTION')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,IDET)
C
C  EMPLACEMENT METHODS
C
1640  CONTINUE
      WRITE (IO,1650)
1650  FORMAT (' ' /,5X,'EMPLACEMENT METHODS',/5X,19('-'))
C
C  GO TO (1660,2010,2140,1660,1660,1660,1660,1660,1660,1660,16
C  + 60,2010,2010,2010) IMET(K)
C  IMET = 1
C
1660  CONTINUE
      IMETF1=IMETF1+1
      WRITE(IO,1665)
1665  FORMAT(' ' //T8,'METHOD OF EMPLACEMENT IS HOLE DRILLING'/
*  T9,38('-')//)
1700  CONTINUE
      IF (KIND(K).EQ.2) GO TO 1730
      IF (IDET.EQ.1) GO TO 1760
      WRITE (IO,1710) K,UOMATU(1,K)
1710  FORMAT (' ' /' ' UOMATU(1,K) (R-H WASTE TYPE (' ,I2,' ) )'
*          /' ' R-H WASTE BARRIER MATERIAL',T50,G15.5
*          /' ' UNIT REQUIREMENT, MATERIAL UNIT/EMPLACEMENT
+ UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,UOMATU(1,K))
      WRITE (IO,1720) K,UOMATC(1,K)
1720  FORMAT (' ' /' ' UOMATC(1,K) (R-H WASTE TYPE (' ,I2,' ) )'
*          /' ' R-H WASTE BARRIER MATERIAL',T50,G15.5
*          /' ' UNIT COST, $/EMPLACEMENT UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,UOMATC(1,K))
      GO TO 1810
1730  CONTINUE
      IF (IDET.EQ.1) GO TO 1760
      WRITE (IO,1740) K,UOMATU(2,K)
1740  FORMAT (' ' /' ' UOMATU(2,K) (C-H WASTE TYPE (' ,I2,' ) )'
*          /' ' C-H WASTE BARRIER MATERIAL',T50,G15.5
*          /' ' UNIT REQUIREMENT, MATERIAL '

```

```

* 'UNIT/EMPLACEMENT UNIT')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,UOMATU(2,K))
  WRITE (IO,1750) K,UOMATC(2,K)
1750 FORMAT (' ' /' UOMATC(2,K) (C-H WASTE TYPE (' ,I2,' ) )'
*          /' C-H WASTE BARRIER MATERIAL',T50,G15.5
*          /' UNIT COST, $/MATERIAL UNIT')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,UOMATC(2,K))
1760 CONTINUE
  IF (IDET.NE.1) GOTO 1810
  WRITE (IO,1770) K,REFMAT(K)
1770 FORMAT (' ' /' REFMAT(K) (WASTE TYPE',I3,' )'
*          /' REFERENCE HOLE LINER MATERIAL',T50,G15.5
*          /' COST, $/PACKAGES')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,REFMAT(K))
  WRITE (IO,1780) K,REFSLV(K)
1780 FORMAT (' ' /' REFSLV(K) (WASTE TYPE',I3,' )'
*          /' REFERENCE HOLE SLEEVE COST',T50,G15.5
*          /' $/SLEEVE')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,REFSLV(K))
  WRITE (IO,1790) K,REFBKF(K)
1790 FORMAT (' ' /' REFBKF(K) (WASTE TYPE',I3,' )'
*          /' REFERENCE BACKFILL MATERIAL',T50,G15.5
*          /' COST, $/HOLE')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,REFBKF(K))
  WRITE (IO,1800) K,REFDEP(K)
1800 FORMAT (' ' /' REFDEP(K) (WASTE TYPE',I3,' )'
*          /' REFERENCE HOLE DEPTH FOR',T50,G15.5
*          /' REFERENCE COSTS, METERS')
  READ(IN,9000) ICARD
  CALL DCODER(ICARD,REFDEP(K))
C
C      (ALSO SEE REFERENCE PACKAGE DIAMETER UNDER ITEM II)
C
1810 CONTINUE
C  CALCULATE HOLE DRILLING COSTS USING IUHDCM=1
C
  IF (IMETF1.GT.1) GO TO 2000
  WRITE (IO,1820) IUHDCM
1820 FORMAT (' ' /' IUHDCM (GENERIC: ALL)'
*          /' CALCULATE HOLE DRILLING COSTS'
*          /' WHERE OPTION =' ,I5
*          /' 0 = HOLE DRILLING MATERIAL '
*' REQUIREMENT,'
*          /' MATERIAL UNIT/BOREHOLE'

```

```

*          /'          1 = UNIT HOLE DRILLING COST')
      READ(IN,9000) ICARD
      CALL DCODEI(ICARD,IUHDCM)
      IF (IUHDCM.EQ.1) GO TO 1880
      WRITE (IO,1830) EELBRR(1)
1830 FORMAT (' ' /'          EELBRR(1)      (GENERIC: ALL)'
*          /'          HOLE DRILLING LABOR',T50,G15.5
*          /'          REQUIREMENT, MAN-YEARS/EQUIPMENT',
* ' UNIT/SHIFT/DAY')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EELBRR(1))
      WRITE (IO,1840) EEEQPL(1)
1840 FORMAT (' ' /'          EEEQPL(1)      (GENERIC: ALL)'
*          /'          HOLE DRILLING UNIT EQUIPMENT',T50,G15.5
*          /'          LIFE, EMPLACEMENT UNITS/EQUIPMENT UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EEEQPL(1))
      WRITE (IO,1850) EEEQPC(1)
1850 FORMAT (' ' /'          EEEQPC(1)      (GENERIC: ALL)'
*          /'          HOLE DRILLING EMPLACEMENT UNIT',T50,G15.5
*          /'          EQUIPMENT CAPACITY, '
*          /'          EMPLACEMENT UNITS/EQUIPMENT UNIT/SHIFT/DAY')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EEEQPC(1))
      WRITE (IO,1860) EEMATU(1)
1860 FORMAT (' ' /'          EEMATU(1)      (GENERIC: ALL)'
*          /'          HOLE DRILLING MATERIAL',T50,G15.5
*          /'          REQUIREMENT, MATERIAL UNITS/EMPLACEMENT UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EEMATU(1))
      WRITE (IO,1870) EECST(1)
1870 FORMAT (' ' /'          EECST(1)      (GENERIC: ALL)'
*          /'          HOLE DRILLING CAPITAL',T50,G15.5
*          /'          COSTS, $')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EECST(1))
1880 CONTINUE
      IF (IUHDCM.NE.1) GOTO 1885
      EELBRR(1) = 0.0
      EEEQPC(1) = 0.0
      EEEQPC(1) = 0.0
      EEMATU(1) = 0.0
      EECST(1) = 0.0
1885 CONTINUE
      WRITE (IO,1670) EELBRC(1)
1670 FORMAT (' ' /'          EELBRC(1)      (GENERIC: ALL)'
*          /'          HOLE DRILLING UNIT LABOR COST',T50,G15.5
*          /'          $/MAN-YEAR')
      READ(IN,9000) ICARD

```

```

        CALL DCODER(ICARD,EELBRC(1))
        WRITE (IO,1680) EEECS(1)
1680 FORMAT (' '//' EEECS(1) (GENERIC: ALL)'
*           //' HOLE DRILLING UNIT EQUIPMENT',T50,G15.5
*           //' COST, $/EQUIPMENT UNIT')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,EEECST(1))
        WRITE (IO,1690) EEMATC(1)
1690 FORMAT (' '//' EEMATC(1) (GENERIC: ALL)'
*           //' HOLE DRILLING UNIT MATERIALS',T50,G15.5
*           //' COST, $/MATERIAL UNIT')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,EEMATC(1))
        IF (IUHDCM.EQ.0) GO TO 2000
1890 CONTINUE
        WRITE (IO,1900) HDLCFC
1900 FORMAT (' '//' PORTION OF HOLE DRILLING UNIT COST, FRACTION
+'//
*           //' HDLCFC (GENERIC: ALL)'
*           //' - ATTRIBUTED TO LABOR',T50,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,HDLCFC)
        WRITE (IO,1910) HDECFC
1910 FORMAT (' '//' HDECFC (GENERIC: ALL)'
*           //' - ATTRIBUTED TO EQUIPMENT',T50,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,HDECFC)
        WRITE (IO,1920) HDMCFC
1920 FORMAT (' '//' HDMCFC (GENERIC: ALL)'
*           //' - ATTRIBUTED TO MATERIALS',T50,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,HDMCFC)
        IF (NTYP.EQ.1) WRITE (IO,1930) NTYP
1930 FORMAT (' '//' NTYP (GENERIC: ALL)'
*           //' HOLE DRILLING COST FUNCTION TYPE IS',T50
*           //' EXPONENTIAL')
        IF (NTYP.EQ.0) WRITE (IO,1940)
1940 FORMAT (' '//' NTYP (GENERIC: ALL)'
*           //' HOLE DRILLING COST FUNCTION TYPE IS',T50,
*           //' POWER')
        WRITE (IO,1950)
1950 FORMAT (' '//' TO CHANGE ENTRY ENTER THE CORRESPONDING INTE',
+'GER', //' 1 = LINEAR'
*           //' 2 = EXPONENTIAL')
        READ(IN,9000) ICARD
        CALL DCODEI(ICARD,NTYP)
        IF (NTYP.EQ.1) WRITE (IO,1960)
1960 FORMAT (' '//' FUNCTION = A + B * D'
*           //' WHERE D = HOLDIA(K),K=1,NWT')

```

```

      IF (NTYP.EQ.2) WRITE (IO,1970)
1970 FORMAT (' '//'      FUNCTION = A * EXP ( B * D )'
*           /'      WHERE D = HOLDIA(K),K=1,NWT')
      WRITE (IO,1980) HDCON1
1980 FORMAT (' '//'      HDCON1      (GENERIC: ALL)'
*           /'      HOLE DRILLING COST FUNCTION',T50,G10.5
*           /'      CONSTANT A')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,HDCON1)
      WRITE (IO,1990) HDCON2
1990 FORMAT (' '//'      HDCON2      (GENERIC: ALL)'
*           /'      HOLE DRILLING COST FUNCTION',T50,G15.5
*           /'      CONSTANT B')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,HDCON2)
2000  CONTINUE
      GOTO 2240
C
C  EMPLACEMENT METHOD EXCAVATION - ROOM EMPLACEMENT
C
      IMETF2=IMETF2+1
      WRITE(IO,2011)
2011  FORMAT(' '//'T8,'METHOD OF EMPLACEMENT IS ROOM EMPLACEMENT'/
*          T9,41('-')//)
2010  CONTINUE
2040  CONTINUE
      IF (IDET.NE.1) GOTO 2060
      IF (KIND(K).NE.2) GOTO 2060
      WRITE (IO,2050) K,CHMTEM(K)
2050  FORMAT (' '//'      CHMTEM(K) (C-H WASTE TYPE (' ,I2,') )'
*           /'      C-H WASTE MATERIALS COST,' ,T50,G15.5
*           /'      $/UNIT PITCH')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,CHMTEM(K))
2060  CONTINUE
      IF(KIND(K).NE.2) CHMTEM(K)= 0.0
      IF (IMETF2.GT.1) GO TO 2001
      WRITE (IO,2070) EECST(2)
2070  FORMAT (' '//'      EECST(2)      (GENERIC: ALL)'
*           /'      ROOM EMPLACEMENT CAPITAL',T50,G15.5
*           /'      COST, $ ')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EECST(2))
      WRITE (IO,2080) EELBRR(2)
2080  FORMAT (' '//'      EELBRR(2)      (GENERIC: ALL)'
*           /'      ROOM EMPLACEMENT LABOR',T50,G15.5
*           /'      REQUIREMENT, MAN-YEARS/EMPLACEMENT'
*           , ' UNIT/SHIFT')
      READ(IN,9000) ICARD

```

```

        CALL DCODER(ICARD,EELBRR(2))
        WRITE (IO,2090) EELBRC(2)
2090 FORMAT (' '/' ' EELBRC(2) (GENERIC: ALL)'
*          /'          ROOM EMPLACEMENT LABOR COST, ',T50,G15.5
*          /'          $/MAN-YEAR')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,EELBRC(2))
        WRITE (IO,2100) EEEQPL(2)
2100 FORMAT (' '/' ' EEEQPL(2) (GENERIC: ALL)'
*          /'          ROOM EMPLACEMENT UNIT',T50,G15.5
*          /'          EQUIPMENT LIFE, PACKAGES/EQUIPMENT LIFE')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,EEEQPL(2))
        WRITE (IO,2110) EEEQPC(2)
2110 FORMAT (' '/' ' EEEQPC(2) (GENERIC: ALL)'
*          /'          ROOM EMPLACEMENT UNIT',T50,G15.5
*          /'          EQUIPMENT CAPACITY, PACKAGES EQUIPMENT'
*          ' UNIT/SHIFT')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,EEEQPC(2))
        WRITE (IO,2120) EEECST(2)
2120 FORMAT (' '/' ' EEECST(2) (GENERIC: ALL)'
*          /'          ROOM EMPLACEMENT UNIT',T50,G15.5
*          /'          EQUIPMENT COST, PACKAGES/EQUIPMENT',
*          ' UNIT/SHIFT')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,EEECST(2))
        WRITE (IO,2130) EEMATU(2)
2130 FORMAT (' '/' ' EEMATU(2) (GENERIC: ALL)'
*          /'          ROOM EMPLACEMENT MATERIAL',T50,G15.5
*          /'          REQUIREMENT, MATERIAL UNITS/PACKAGES')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,EEMATU(2))
        WRITE (IO,2030) EEMATC(2)
2030 FORMAT (' '/' ' EEMATC(2) (GENERIC: C-H WASTE)'
*          /'          C-H EMPLACEMENT MATERIAL',T50,G15.5
*          /'          UNIT COST, $/MATERIAL UNIT')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,EEMATC(2))
C
2001 CONTINUE
      GOTO 2240
C
C J.3 EMPLACEMENT METHOD EXCAVATION - TRENCHING
C
      IMETF3=IMETF3+1
      WRITE(IO,2145)
2145 FORMAT(' '///T8,'METHOD OF EMPLACEMENT IS TRENCHING'/
*          T9,34('-')/)

```



```

2140  CONTINUE
      WRITE (IO,2150) K,TRCHWD(K)
2150  FORMAT (' ' /'          TRCHWD(K) (WASTE TYPE',I3,')'
*          /'          TRENCH WIDTH, METERS',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,TRCHWD(K))
      IF (IMETF3.GT.1) GO TO 2240
      WRITE (IO,2160) EELBRR(3)
2160  FORMAT (' ' /'          EELBRR(3) (GENERIC: ALL)'
*          /'          TRENCHING LABOR REQUIREMENT,',T50,G15.5
*          /'          MAN-YEARS/EMPLACEMENT UNIT/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EELBRR(3))
      WRITE (IO,2170) EEEQPL(3)
2170  FORMAT (' ' /'          EEEQPL(3) (GENERIC: ALL)'
*          /'          TRENCHING UNIT EQUIPMENT LIFE,',T50,G15.5
*          /'          PACKAGES/EQUIPMENT UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EEEQPL(3))
      WRITE (IO,2180) EEEQPC(3)
2180  FORMAT (' ' /'          EEEQPC(3) (GENERIC: ALL)'
*          /'          TRENCHING UNIT EQUIPMENT',T50,G15.5
*          /'          CAPACITY, PACKAGES/EQUIPMENT UNIT/SHIFT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EEEQPC(3))
      WRITE (IO,2190) EEMATU(3)
2190  FORMAT (' ' /'          EEMATU(3) (GENERIC: ALL)'
*          /'          TRENCHING MATERIAL',T50,G15.5
*          /'          REQUIREMENT, MATERIAL UNITS/PACKAGE')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EEMATU(3))
      WRITE (IO,2200) EECST(3)
2200  FORMAT (' ' /'          EECST(3) (GENERIC: ALL)'
*          /'          TRENCHING CAPITAL COST, $',T50,G15.5)
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EECST(3))
      WRITE (IO,2210) EELBRC(3)
2210  FORMAT (' ' /'          EELBRC(3) (GENERIC: ALL)'
*          /'          TRENCHING UNIT LABOR COST,',T50,G15.5
*          /'          $/MAN-YEAR')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EELBRC(3))
      WRITE (IO,2220) EEECST(3)
2220  FORMAT (' ' /'          EEECST(3) (GENERIC: ALL)'
*          /'          TRENCHING UNIT EQUIPMENT',T50,G15.5
*          /'          COST, $/EQUIPMENT UNIT')
      READ(IN,9000) ICARD
      CALL DCODER(ICARD,EEECST(3))
      IF (IDET.NE.0) GO TO 2240

```

```

        WRITE (IO,2230) EEMATC(3)
2230 FORMAT (' ' / '      EEMATC(3)      (GENERIC: ALL)'
*          / '      TRENCHING UNIT MATERIAL',T50,G15.5
*          / '      COST, $/MATERIAL UNIT')
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,EEMATC(3))
2240 CONTINUE
        IF (IDET.NE.1) GO TO 2280
2250 CONTINUE
        WRITE (IO,2260) K,PKGPRK(K)
2260 FORMAT (' ' / '      PKGPRK(K) (WASTE TYPE',I3,')'
*          / '      PACKAGES EMPLACED PER RACK',T50,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,PKGPRK(K))
        WRITE (IO,2270) K,RCKCST(K)
2270 FORMAT (' ' / '      RCKCST(K) (WASTE TYPE',I3,')'
*          / '      UNIT RACK COST, $/RACK',T50,G15.5)
        READ(IN,9000) ICARD
        CALL DCODER(ICARD,RCKCST(K))
2280 CONTINUE
C
2290 CONTINUE
        RETURN
        END
        SUBROUTINE READ(INDEX)
C
C   DIRECTS THE PREPROCESSOR TO GO TO A SPECIFIC CATEGORY
C
C       GO TO ( 10, 20, 30, 40, 50, 60, 70, 80, 90 ) INDEX
C
C   GENERAL
C
C   10 CALL RDGEN
C       RETURN
C
C   WASTE
C
C   20 CALL RDWAST
C       RETURN
C
C   REPOSITORY
C
C   30 CALL RDRPOS
C       RETURN
C
C   SHAFT
C
C   40 CALL RDSHFT
C       RETURN

```

```

C
C  ROCK
C
C  50  CALL RDROCK
C      RETURN
C
C  SUPPORT
C
C  60  CALL RDSUPT
C      RETURN
C
C  UTILITIES
C
C  70  CALL RDUTLT
C      RETURN
C
C  PRE-OPERATIONAL
C
C      RETURN
C  80  CALL RDPREP
C      RETURN
C
C  DECOMMISSIONING AND SURVEILANCE
C
C  90  CALL RDDECM
C      RETURN
C      END
C      SUBROUTINE TABLE
C
C  THIS SUBROUTINE REQUESTS THE USER WHICH OUTPUT TABLES HE WISHES PRINTED
C
C
C      INCLUDE PARAM
C      INCLUDE INPUT
C      INTEGER TA
C      DIMENSION TA (27)
C
C      IZERO = 0
C      IC = 8
C
C      DO 5 I=1,27
C          TA(I) = 0
C          TABLES(I) = 0
C  5      CONTINUE
C
C      WRITE (IOUT,5010)
C  5010  FORMAT(' ' //)
C      WRITE (IO,100)
C  100   FORMAT(' ', 'DO YOU WISH TO PRINT OUT SPECIFIC TABLES OR ALL '

```

```

* ' TABLES? ' / '      ENTER 0 = ALL ' / '      1 = SPECIFIC TABLES ' )
10  READ (IN,*,ERR=110) IANS
    IF (IANS.EQ.0) GOTO 3000
    GOTO 20
110  WRITE (IO,115)
115  FORMAT(' ', 'YOU MUST ENTER A 0 OR 1 FOR THE PROPER RESPONSE' /
*      'TRY AGAIN')
    GOTO 10
20  CONTINUE
    WRITE (IO,120)
120  FORMAT(' ', 'DO YOU KNOW WHAT TABLES YOU WISH TO PRINT OR WOULD'
* / ' YOU LIKE TO SEE THE TABLE HEADINGS? '
* / ' ENTER 0 = NO TABLE HEADINGS '
* / ' 1 = PRINT TABLE HEADINGS ' )
130  READ (IN,*,ERR=125) IANS
    GOTO 30
125  WRITE (IO,115)
    GOTO 130
C
30  CONTINUE
    IF (IANS.EQ.0) GOTO 4000
C
C  WRITE OUT HEADINGS AS THEY APPEAR IN RECON
C
C  TABLE 1
C
    WRITE (IO,150) RPTYP
150  FORMAT(' TABLE 1. MINE DEVELOPMENT PARAMETERS FOR A REPOSITORY'
* , ' IN ', A16)
    WRITE (IO,160) RPTYP
160  FORMAT(' TABLE 2. RECEIVING AND PROCESSING FOR A REPOSITORY'
* , ' IN ', A16)
    WRITE (IO,170) RPTYP
170  FORMAT(' TABLE 3. WASTE SHIPPING DISTRIBUTION FOR A REPOSITORY'
* , ' IN ', A16)
    WRITE (IO,180) RPTYP, RPTYP
180  FORMAT(' TABLE 4A. WASTE SHIPPING DISTRIBUTION FOR A REPOSITORY'
* , ' IN ', A16 /
*      ' TABLE 4B. RECEIVING AND PROCESSING FOR A REPOSITORY'
* , ' IN ', A16 / ' ENTER AS TABLE 4 ' )
    WRITE (IO,200)
200  FORMAT(' TABLE 5. WASTE OPERATIONS PARAMETERS FOR (EACH WASTE',
* ' TYPE) ' )
    WRITE (IO,210) RPTYP
210  FORMAT(' TABLE 6. EMPLACEMENT PACKAGE THERMAL OUTPUT FOR A',
* ' REPOSITORY IN ' / 13X, A16)
    WRITE (IO,220) RPTYP
220  FORMAT(' TABLE 7. WASTE LOADING PARAMETERS FOR A REPOSITORY '

```

```

*, 'IN ', A16)
WRITE (IO,230) RPTYP
230  FORMAT('  TABLE 8.  REPOSITORY DEVELOPMENT SCHEDULE FOR EACH'
*, ' WASTE TYPE IN '/13X, A16)
WRITE (IO,245) RPTYP
245  FORMAT('  TABLE 9.  SUMMARY DEVELOPMENT SCHEDULE FOR A REPOSITORY'
*, ' IN ', A16)
C
WRITE (IO,255)
255  FORMAT(' ')
WRITE (IO,250)
250  FORMAT('  ENTER THE TABLE NUMBERS, SEPARATED BY COMMAS OR SPACES'
*, ', OF THE TABLES'/'  YOUR WANT FROM THIS FIRST SECTION.  IF NO',
*, ' TABLES ARE REQUESTED FROM THIS '/'  SECTION, SIMPLY ENTER A',
*, ' CARRIAGE RETURN')
C
READ(IN,9000,END=1010) (TA(J),J=1,9)
9000  FORMAT(9(I1,1X))
C
1010  CONTINUE
WRITE (IO,240)
240  FORMAT('  TABLE 10.  SHAFT PARAMETERS AND COSTS')
WRITE (IO,260) RPTYP
260  FORMAT('  TABLE 11.  WASTE FACILITY CAPITAL FOR A REPOSITORY IN '
*, A16)
WRITE (IO,270) RPTYP
270  FORMAT('  TABLE 12.  EQUIPMENT LIFE AND COST FOR A REPOSITORY IN '
*, A16)
WRITE (IO,280) RPTYP
280  FORMAT('  TABLE 13.  CONSTRUCTION AND PREPARATION COSTS FOR A'
*, ' REPOSITORY IN '/13X, A16)
WRITE (IO,290) RPTYP
290  FORMAT('  TABLE 14.  MINING AND ROCK REMOVAL OPERATIONS COST',
*, ' SUMMARY FOR A REPOSITORY '/13X, A16)
WRITE (IO,300) RPTYP
300  FORMAT('  TABLE 15.  WASTE OPERATIONS LABOR COST SUMMARY FOR A',
*, ' REPOSITORY IN '/13X, A16)
WRITE (IO,310) RPTYP
310  FORMAT('  TABLE 16.  WASTE OPERATIONS MATERIALS COSTS SUMMARY',
*, ' FOR A REPOSITORY '/13X, A16)
WRITE (IO,320) RPTYP
320  FORMAT('  TABLE 17.  WASTE OPERATIONS EQUIPMENT COST SUMMARY',
*, ' FOR A REPOSITORY IN '/13X, A16)
WRITE (IO,330) RPTYP
330  FORMAT('  TABLE 18.  UTILITIES REQUIREMENTS AND COST SUMMARY',
*, ' FOR A REPOSITORY IN '/13X, A16)
WRITE (IO,340)
340  FORMAT('  TABLE 19.  WASTE OPERATIONS COSTS SUMMARY FOR EACH',
*, ' WASTE TYPE')

```

```

      WRITE (IO,350) RPTYP
350  FORMAT('  TABLE 20.  BACKFILLING AND DECOMMISSIONING SUMMARY',
      *' FOR A REPOSITORY IN '/13X,A16)
      WRITE (IO,360) RPTYP
360  FORMAT('  TABLE 21.  COST SUMMARY FOR A REPOSITORY IN ',A16)
      WRITE (IO,370)
370  FORMAT('  TABLE 22.  REPOSITORY COST MATRIX')
C
      WRITE (IO,255)
      WRITE (IO,251)
251  FORMAT('  ENTER THE TABLE NUMBERS, SEPARATED BY COMMAS OR SPACES'
      *,',', OF THE TABLES'/' YOUR WANT FROM THIS SECOND SECTION.  IF NO',
      *' TABLES ARE REQUESTED FROM THIS '/' SECTION, SIMPLY ENTER A',
      *' CARRIAGE RETURN')
C
      READ(IN,9010,END=1020) (TA(J),J=10,22)
9010  FORMAT(13(I2,1X))
C
1020  WRITE (IO,380) RPTYP
380  FORMAT('  TABLE 23.  CASH FLOW SUMMARY FOR A REPOSITORY IN ',A16)
      WRITE (IO,390) RPTYP
390  FORMAT('  TABLE 24.  UNIT COST SUMMARY FOR A REPOSITORY IN ',A16
      *'/' DOLLARS PER KILOGRAM HEAVY METAL')
      WRITE (IO,400) RPTYP
400  FORMAT('  TABLE 25.  UNIT COST SUMMARY FOR A REPOSITORY IN ',A16
      *'/' DOLLARS PER CUBIC METER OF WASTE')
      WRITE (IO,410) RPTYP
410  FORMAT('  TABLE 26.  UNIT COST SUMMARY FOR A REPOSITORY IN ',A16
      *'/' DOLLARS PER PACKAGE')
      WRITE (IO,420)
420  FORMAT('  TABLE 27.  COEMPLACEMENT ARRAY')
C
      WRITE (IO,255)
      WRITE (IO,252)
252  FORMAT('  ENTER THE TABLE NUMBERS, SEPARATED BY COMMAS OR SPACES'
      *,',', OF THE TABLES'/' YOUR WANT FROM THIS THIRD SECTION.  IF NO',
      *' TABLES ARE REQUESTED FROM THIS '/' SECTION, SIMPLY ENTER A',
      *' CARRIAGE RETURN')
C
      READ(IN,9020) (TA(J),J=23,27)
9020  FORMAT(5(I2,1X))
C
1000  CONTINUE
C
2010  NUM = 0
      JFLG = 0
      DO 2000 J=1,27
      IF (TA(J).LT.0.OR.TA(J).GT.27) JFLG = 1
      IF (TA(J).LE.0.OR.TA(J).GT.27) GOTO 2000

```

```

        NUM = NUM+1
        TABLES(NUM) = TA(J)
2000  CONTINUE
C
        IF (JFLG.EQ.1) GOTO 7000
C
C  SORT TABLES IN NUMERICAL ORDER
C
        IF (NUM.LE.1) GOTO 2020
        L = NUM-1
        DO 2030 I =1,L
            K = I+1
            DO 2040 J = K,NUM
                IF (TABLES(I).LT.TABLES(J)) GOTO 2040
                M = TABLES(I)
                TABLES(I) = TABLES(J)
                TABLES(J) = M
2040  CONTINUE
2030  CONTINUE
2020  CONTINUE
        GOTO 5000
C
C  SET TABLES MATRIX TO PRINT ALL TABLES
C
3000  CONTINUE
        WRITE (IC,*) IZERO
        WRITE (IOUT,3001)
3001  FORMAT(' ALL OUTPUT TABLES WILL BE OUTPUT FROM RECON.')
```

RETURN

```

C
C  USER KNOWS WHAT TABLES HE WANTS PRINTED
C
4000  CONTINUE
        WRITE (IO,255)
        WRITE (IO,3010)
3010  FORMAT(' ENTER THE NUMBER OF TABLES FOLLOWED BY THE'/
* ' CORRESPONDING TABLE NUMBERS YOU WISH TO SEE OUTPUT')
        READ(IN,*,ERR=1000,END=1000) N,(TA(I),I=1,N)
C
C  CHECK TABLE NUMBERS FOR SEQUENCE
C
        GOTO 1000
C
5000  CONTINUE
        DO 5020 I = 1,NUM,5
            J = I+4
            IF (J.GT.NUM) J=NUM
            WRITE (IC,*) (TABLES(L),L=I,J)
            WRITE (IOUT,5001) (TABLES(K),K=1,NUM)

```

```

5001 FORMAT(' THE FOLLOWING TABLES WILL BE OUTPUT FROM RECON'/
* 27(I3,', '))
5020 CONTINUE
      RETURN
7000 CONTINUE
      WRITE (IO,7010)
7010 FORMAT(' ONE OR MORE TABLE NUMBERS REQUESTED WAS IN ERROR'/
* ' RE-INPUT REQUESTED NUMBERS')
      GOTO 30
      END
      SUBROUTINE WRITE
C
      INCLUDE PARAM
      INCLUDE INPUT,LIST
C
C
      DATYPE(1)(1:11) = 'INPUT DATA '
      DATYPE(2)(1:17) = 'WASTE OPERATIONS '
      DATYPE(3)(1:17) = 'MINE DEVELOPMENT '
      DATYPE(4)(1:21) = 'RECEIVING OPERATIONS '
      DATYPE(5)(1:22) = 'UNDERGROUND OPERATION '
      DATYPE(6)(1:17) = 'SHAFT OPERATIONS '
      DATYPE(7)(1:19) = 'SURFACE OPERATIONS '
      DATYPE(8)(1:10) = 'UTILITIES '
      DATYPE(9)(1:15) = 'PRE-OPERATIONS '
      DATYPE(10)(1:16) = 'POST-OPERATIONS '
      IC = 8
C
      WRITE (IC,100) DATYPE(1)
100 FORMAT (A80)
      WRITE (IC,100) TITLE
110 FORMAT (A48)
      WRITE (IC,120) RPTYP,PLACE,TEKNEK
120 FORMAT (3A16)
      WRITE (IC,*) IYRCON,IYRPRO,LASTYR,IYDOL,IDDYR
      WRITE (IC,*) NWT,DRATE
      IYRDIS = IYDOL
      WRITE (IC,*) DEPTH,DENSTY,SPUC
      WRITE (IC,*) CONFAC
      NSTM1 = NST-1
      DO 1120 J=1,NSTM1,5
      JJ= J+4
      IF(JJ.GT.NSTM1) JJ=NSTM1
      WRITE(IC,*) (SFTNUM(L),L=J,JJ)
1120 CONTINUE
      DO 140 K=1,NWT
      WRITE (IC,130) WSTYP(K),PKGDS1(K),PKGDS2(K),PKGDS3(K)
130 FORMAT (A12,3A16)
140 CONTINUE

```



```

DO 141 J=1,NWT,5
JJ=J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (KIND(L),L=J,JJ)
141 CONTINUE
DO 143 J=1,NWT,5
JJ= J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (UNTMAS(L),L=J,JJ)
143 CONTINUE

C
LL = IYRPRO-IYRCON
LY = LASTYR-IYRCON+1
IDD = IDDYR
LYY = LY

C
C WASTOP
C

WRITE (IC,100) DATYPE(2)
WRITE (IC,*) SIZE,CPA
WRITE (IC,*) GXEF
DO 4001 J=1,NWT,5
JJ=J+4
IF (JJ.GT.NWT) JJ=NWT
WRITE (IC,*) (NFTL(L),L=J,JJ)
4001 CONTINUE
WRITE (IC,*) FFTL,NALSH,NRHRMA
WRITE (IC,*) NCHRNA,WRKYR,MTRPSD,MTCPSD
WRITE (IC,*) (SHCFAC(L),L = 1,NS),OVRHDF
DO 4010 J=1,NWT,5
JJ=J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (PANLEN(L),L=J,JJ)
4010 CONTINUE
DO 4015 J=1,NWT,5
JJ=J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (PANWTH(L),L=J,JJ)
4015 CONTINUE
DO 4000 J=1,NWT,5
JJ=J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (WUPP(L),L=J,JJ)
4000 CONTINUE
DO 4020 J=1,NWT,5
JJ=J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (WVOLPU(L),L=J,JJ)

```

```

4020  CONTINUE
      DO 4030 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (TTTR(L),L=J,JJ)
4030  CONTINUE
      DO 4500 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (TTTK(L),L=J,JJ)
4500  CONTINUE
      DO 4040 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (RCCAP(L),L=J,JJ)
4040  CONTINUE
      DO 4050 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (TCCAP(L),L=J,JJ)
4050  CONTINUE
      DO 4060 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (PKGCAP(L),L=J,JJ)
4060  CONTINUE
      DO 4070 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (RMLN(L),L=J,JJ)
4070  CONTINUE
      DO 4080 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (RMWTH(L),L=J,JJ)
4080  CONTINUE
      DO 4090 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (CLSEMP(L),L=J,JJ)
4090  CONTINUE
      DO 4100 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
4100  CONTINUE
      DO 4110 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT

```

```

WRITE(IC,*) (HPMIN(L),L=J,JJ)
4110 CONTINUE
DO 4120 J=1,NWT,5
JJ=J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (PKGUP(L),L=J,JJ)
4120 CONTINUE
DO 4130 J=1,NWT,5
JJ=J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (NBSPPA(L),L=J,JJ)
4130 CONTINUE
C
DO 4140 K=1,NWT
DO 4140 I=1,LY,5
II = I+4
IF(II.GT.LY) II=LY
WRITE(IC,*) (WSTSUP(K,J),J=I,II)
4140 CONTINUE
C
DO 4150 K=1,NWT
DO 4150 I=1,LY,5
II=I+4
IF(II.GT.LY) II=LY
WRITE(IC,*) (TDWI(J,K),J=I,II)
4150 CONTINUE
C
DO 4160 K=1,NWT
DO 4160 I=1,LY,5
II=I+4
IF(II.GT.LY) II=LY
WRITE(IC,*) (CRRAIL(K,J),J=I,II)
4160 CONTINUE
DO 210 L = 1,NWO
WRITE (IC,*) WOCST(L),WOLBRR(L),WOLBRC(L),WOEQPL(L),WOEQPC(L)
WRITE (IC,*) WOECST(L)
DO 5010 J=1,NWT,5
JJ=J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (WOMATU(L,K),K=J,JJ)
5010 CONTINUE
DO 5020 J=1,NWT,5
JJ=J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (WOMATC(L,K),K=J,JJ)
5020 CONTINUE
210 CONTINUE
C
C MINDEV

```

C

```

WRITE (IC,100) DATYPE(3)
NCOR = NCOR-1
WRITE (IC,*) LEAD,NCOR,TCPCLN,PCTFIL
WRITE (IC,*) XPANFC,MSA,TONPAR,OFTADC,NBYR
WRITE (IC,*) OTHBFL,OTBDEN,BKRATE,MSHIFT(1),MSHIFT(2)
DO 4400 J=1,NWT,5
  JJ=J+4
  IF(JJ.GT.NWT) JJ=NWT
  WRITE(IC,*) (RMHMIN(L),L=J,JJ)
4400 CONTINUE
  NCOR = NCOR+1
  DO 220 M = 1,NCOR
    WRITE (IC,110) CORDSC(M)
    WRITE (IC,*) CORWTH(M),CORHT(M),NCORPP(M)
  DO 4405 J=1,NWT,5
    JJ=J+4
    IF(JJ.GT.NWT) JJ=NWT
    WRITE(IC,*) (CORLEN(M,L),L=J,JJ)
4405 CONTINUE
  220 CONTINUE
  DO 4402 J=1,NWT,5
    JJ=J+4
    IF(JJ.GT.NWT) JJ=NWT
    WRITE(IC,*) (DELRHT(K),K=J,JJ)
4402 CONTINUE
  DO 4410 J=1,NWT,5
    JJ=J+4
    IF(JJ.GT.NWT) JJ=NWT
    WRITE(IC,*) (PKGLN(K),K=J,JJ)
4410 CONTINUE
  DO 4420 J=1,NWT,5
    JJ=J+4
    IF(JJ.GT.NWT) JJ=NWT
    WRITE(IC,*) (NRWSPR(K),K=J,JJ)
4420 CONTINUE
  DO 4170 J=1,LL,5
    JJ=J+4
    IF(JJ.GT.LL) JJ= LL
    WRITE(IC,*) (PORMDF(1,L),L=J,JJ)
4170 CONTINUE
  DO 4180 J=1,LL,5
    JJ=J+4
    IF(JJ.GT.LL) JJ=LL
    WRITE(IC,*) (PORMDF(2,L),L=J,JJ)
4180 CONTINUE
  DO 4190 J=1,LL,5
    JJ=J+4
    IF(JJ.GT.LL) JJ=LL

```

```

        WRITE(IC,*) (SCSKD(I),I=J,JJ)
4190  CONTINUE
        DO 225 L=1,NST
        WRITE (IC,*) SFTDD(L),SFTXR(L),SFTXD(L)
225  CONTINUE
        WRITE (IC,*) MSWTCH
        IF (MSWTCH.EQ.0) GO TO 240
        DO 230 L = 1,NRO
            WRITE (IC,*) ROCST(L),ROLBRR(L),ROLBRC(L),ROEQPL(L),ROEQPC(L)
        WRITE(IC,*) ROECST(L),ROMATU(L),ROMATC(L)
230  CONTINUE
        GO TO 250
240  CONTINUE
        WRITE(IC,*) VARMC,RCKHC,SURFDC,BKFLC
250  CONTINUE
C
C  RECEVE
C
        WRITE (IC,100) DATYPE(4)
        WRITE (IC,*) NRBPRM, NCBPRM
        DO 4200 J=1,LL,5
            JJ=J+4
            IF(JJ.GT.LL) JJ=LL
        WRITE(IC,*) (RMCWS1(L),L=J,JJ)
4200  CONTINUE
        DO 4210 J=1,LL,5
            JJ=J+4
            IF(JJ.GT.LL) JJ=LL
        WRITE(IC,*) (RMCWS2(L),L=J,JJ)
4210  CONTINUE
        DO 4220 J=1,LL,5
            JJ=J+4
            IF(JJ.GT.LL) JJ=LL
        WRITE(IC,*) (RMNWS1(L),L=J,JJ)
4220  CONTINUE
        DO 4230 J=1,LL,5
            JJ=J+4
            IF(JJ.GT.LL) JJ=LL
        WRITE(IC,*) (RMNWS2(L),L=J,JJ)
4230  CONTINUE
C
C  UGOPR
C
        WRITE (IC,100) DATYPE(5)
        DO 260 L = 1,NT0
            WRITE (IC,*) TOCST(L),TOLBRR(L),TOLBRC(L),TOEQPL(L),TOEQPC(L)
            WRITE(IC,*) TOECST(L)
        DO 4240 J=1,NWT,5
            JJ=J+4

```

```

      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (TOMATU(L,M),M=J,JJ)
4240  CONTINUE
      DO 4250 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (TOMATC(L,M),M=J,JJ)
4250  CONTINUE
260   CONTINUE
      WRITE (IC,*) IDET,IUHDCM
      DO 4440 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (TRCHWD(K),K=J,JJ)
4440  CONTINUE
      DO 270 K = 1,NWT
      WRITE (IC,*) NPKPEU(K),PKGDI(K),REFDIA(K),HOLDEL(K),IMET(K)
270   CONTINUE
C
C   CO-EMPLACEMENT DATA
C
      WRITE (IC,*) MPRIME, NNHG, ICLR
      IF(MPRIME.EQ.NWT) GOTO 235
      DO 4700 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (STOMAX(K),K=J,JJ)
4700  CONTINUE
      IF(NNHG.NE.NWT-MPRIME) NNHG = NWT-MPRIME
      IF(NNHG.EQ.0) GOTO 235
      DO 1000 K=1,MPRIME
      DO 1000,J=1,NNHG
      WRITE(IC,*) COCON(J,K),PCTCON(J,K)
1000  CONTINUE
      DO 4450 J=1,NWT,5
      JJ=J+4
      IF(JJ.GT.NWT) JJ=NWT
      WRITE(IC,*) (SWPPR(K),K=J,JJ)
4450  CONTINUE
      ICH = 0
      DO 275 K=1,NWT
      IF(KIND(K).NE.2) GOTO 275
      IF(IMET(K).LE.4.OR.IMET(K).EQ.7.OR.IMET(K).EQ.10.OR.
+      IMET(K).GE.13) GOTO 275
      ICH = 1
275   CONTINUE
      IF(ICH.EQ.0) GOTO 4470
      DO 4460 J=1,NWT,5
      JJ=J+4

```

```

        WRITE(IC,*) (CONTRU(K),K=J,JJ)
4460  CONTINUE
4470  CONTINUE
235   CONTINUE
        DO 290 L = 1,NUO
            WRITE (IC,*) UOCST(L),UOLBRR(L),UOLBRC(L),UOEQPL(L),UOEQPC(L)
            WRITE (IC,*) UOECST(L)
            IF (IDET.EQ.1) GO TO 290
            DO 280 K = 1,NWT
                IF(KIND(K).EQ.1) M=1
                IF(KIND(K).EQ.2) M=2
                IF(KIND(K).EQ.3) M=1
                IF (M.NE.L) GO TO 280
                WRITE (IC,*) UOMATU(L,K),UOMATC(L,K)
280    CONTINUE
290    CONTINUE
            DO 295 K=1,NWT
                WRITE (IC,*) WABATH(K),SLEETH(K),BCKFTH(K),CAPLEN(K)
295    CONTINUE
            IF (IDET.NE.1) GO TO 310
            DO 300 K = 1,NWT
                WRITE (IC,*) REFMAT(K),REFSLV(K),REFBKF(K),REFDEP(K)
                WRITE (IC,*) CHMTEM(K),PKGPRK(K),RCKCST(K)
300    CONTINUE
310    CONTINUE
            DO 330 L = 1,NEE
                IF (IUHDCM.EQ.1.AND.L.EQ.1) GO TO 320
                WRITE (IC,*) EECST(L),EELBRR(L),EELBRC(L),EEEQPL(L),EEEQPC(L)
                WRITE (IC,*) EEECSST(L),EEMATU(L),EEMATC(M)
                GO TO 330
320    CONTINUE
                WRITE (IC,*) EELBRC(L),EEECST(L),EEMATC(L),HDLFCFC,HDECFC
                WRITE (IC,*) HDMCFC,HDCON1,HDCON2,NTYP
330    CONTINUE
C
C  SHAFT
C
        WRITE (IC,100) DATYPE(6)
        WRITE (IC,*) MXMPRH,MXMMRH,TNPT1,TNPT2,SCCON1
        WRITE (IC,*) SCCON2,IEQTP1,CCCON1,CCCON2,IEQTP2
        WRITE (IC,*) WCCON1,WCCON2,IEQTP3
        DO 4550 J=1,NST,5
            JJ=J+4
            IF(JJ.GT.NST) JJ=NST
            WRITE(IC,*) (SFTLD(L),L=J,JJ)
4550  CONTINUE
        DO 4560 J=1,NST,5
            JJ=J+4
            IF(JJ.GT.NST) JJ=NST

```

```

WRITE(IC,*) (SFTHE(L),L=J,JJ)
4560 CONTINUE
DO 4570 J=1,NST,5
JJ=J+4
IF(JJ.GT.NST) JJ=NST
WRITE(IC,*) (SFTSS(L),L=J,JJ)
4570 CONTINUE
DO 4480 J=1,NWT,5
JJ=J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (NPKPTR(L),L=J,JJ)
4480 CONTINUE
DO 4490 J=1,NWT,5
JJ=J+4
IF(JJ.GT.NWT) JJ=NWT
WRITE(IC,*) (NPKPTC(L),L=J,JJ)
4490 CONTINUE
C
C SURFST
C
WRITE (IC,100) DATYPE(7)
DO 4300 J=1,LL,5
JJ=J+4
IF(JJ.GT.LL) JJ=LL
WRITE(IC,*) (SFCS(L),L=J,JJ)
4300 CONTINUE
WRITE (IC,*) NSC,SOA
DO 340 L = 1,NSC
WRITE (IC,110) SFTITL(L)
WRITE (IC,*) SFCST(L),SFLBRR(L),SLBR(L),SFLBRC(L),SFEQPL(L)
WRITE (IC,*) SFECST(L),SFMATU(L),SFMATC(L)
340 CONTINUE
C
C UTILITY
C
WRITE (IC,100) DATYPE(8)
WRITE (IC,*) UPC,STMC,DWTRC,WTRC,DSLCL
WRITE (IC,*) RAPP,SFSP,OWHP,SFP,MROP
WRITE (IC,*) RVP,ACP,HEATS,RAPS,ORS
WRITE (IC,*) VOLW,VOLDW,SUD,UGD,RED
C
C PREOP
C
WRITE (IC,100) DATYPE(9)
DO 4310 J=1,LL,5
JJ=J+4
IF(JJ.GT.LL) JJ=LL
WRITE(IC,*) (SCDF(L),L=J,JJ)
4310 CONTINUE

```



```

DO 4330 J=1,LL,5
JJ=J+4
IF(JJ.GT.LL) JJ=LL
WRITE(IC,*) (CLNDF(L),L=J,JJ)
4330 CONTINUE
DO 4340 J=1,LL,5
JJ=J+4
IF(JJ.GT.LL) JJ=LL
WRITE(IC,*) (CLCDF(L),L=J,JJ)
4340 CONTINUE
1070 CONTINUE
WRITE (IC,*) TCLAND,TCLICE,STPCT,AECON1,AECON2
WRITE (IC,*) IEQTP4
C
C POSTOP
C
WRITE (IC,100) DATYPE(10)
IF (IDD.EQ.0) GOTO 190
DO 4360 J=1,IDD,5
JJ=J+4
IF(JJ.GT.IDD) JJ=IDD
WRITE(IC,*) (DBDF(L),L=J,JJ)
4360 CONTINUE
DO 4370 J=1,IDD,5
JJ=J+4
IF(JJ.GT.IDD) JJ=IDD
WRITE(IC,*) (SSDF(L),L=J,JJ)
4370 CONTINUE
DO 4380 J=1,IDD,5
JJ=J+4
IF(JJ.GT.IDD) JJ=IDD
WRITE(IC,*) (FDDF(L),L=J,JJ)
4380 CONTINUE
190 CONTINUE
WRITE (IC,*) SSSCON1,SSCON2,IEQTP5,DCOMCF,SURCST
WRITE (IC,*) ITSURV
WRITE (IC,*) NDDC,DDFRAC
IF (NDDC.EQ.0) GO TO 360
DO 350 L = 1,NDDC
WRITE (IC,110) DDTITL(L)
WRITE (IC,*) DDCST(L),DDLBR(L),DDLBR(L),DDEQPC(L),DDMATC(L)
350 CONTINUE
360 CONTINUE
WRITE (IC,*) SCCST,SCLBR,LCLBR,SCEQPC,SCMATU
WRITE (IC,*) SCMATC
C
C
RETURN
END

```

INPUT DATA - TEST									
SAMPLE CASE FOR DOCUMENTATION									
SALT	GULF INTERIOR				BORING				
	1990		1998		2023		0		1
	1	.00000000							
727.00000		2.1600000			11211.000				
.20 .20.	.20 .20								
	0		1		-1		0		1
	1								
HLW-	HLW		HLW				22" X 161"		
	1								
9740.0000									
WASTE OPERATIONS									
2000.0000		380.00000							
90.000000									
120.00000									
120.00000			3		5				
	5	250.00000			16		32		
1.0000000		1.1200000			1.1200000		1.0000000		
1370.0000									
192.00000									
1.0000000									
866.50000									
1.8800000									
1.8800000									
4.0000000									
1.0000000									
17.250000									
152.40000									
4.5700000									
2.9900000									
.15400000									
12.200000									
1.0000000									
	46								
.00000000		.00000000		.00000000		.00000000		.00000000	
.00000000		.00000000		.00000000		185.00000		185.00000	
185.00000		185.00000		185.00000		308.00000		308.00000	
308.00000		308.00000		308.00000		308.00000		308.00000	
308.00000		308.00000		308.00000		308.00000		308.00000	
308.00000		308.00000		308.00000		308.00000		308.00000	
308.00000		308.00000		308.00000		308.00000		308.00000	
308.00000		308.00000		308.00000		102.00000			
10.000000		10.000000		10.000000		10.000000		10.000000	
10.000000		10.000000		10.000000		10.000000		10.000000	
10.000000		10.000000		10.000000		10.000000		10.000000	
10.000000		10.000000		10.000000		10.000000		10.000000	
10.000000		10.000000		10.000000		10.000000		10.000000	
10.000000		10.000000		10.000000		10.000000		10.000000	
10.000000		10.000000		10.000000		10.000000		10.000000	

10.000000	10.000000	10.000000	10.000000	
1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
40000000.	7.3000000	37500.000	1000000.0	20.000000
.00000000				
1.0000000				
303.56000				
2710000.0	2.0000000	37500.000	40000.000	40.000000
.00000000				
.00000000				
.00000000				
.14779600+009	39.000000	37500.000	1000000.0	20.000000
63729000.				
1.0000000				
37620.000				
MINE DEVELOPMENT				
2	1	87722.000	.97000000	
1.4200000	1000.0000	12000.000	8.5980000	2
.00000000	.00000000	.00000000		1
5.1800000				
MAIN CORR.				
4.8800000	4.5700000		0	
147103.00				
PASSAGEWAYS				
9.2000000	4.6000000		2	
1195.0000				
1.6000000				
4.1900000				
1				
.00000000	.00000000	.00000000	.00000000	.20000000
.50000000	.30000000	.00000000		
.00000000	.00000000	.00000000	.00000000	.00000000
.30000000	.40000000	.30000000		
.00000000	.10000000+000	.30000000	.30000000	.25000000
.50000000-001	.00000000	.00000000		
.00000000	.00000000	.00000000		
6.6000000	.00000000	769.00000		
6.7000000	.00000000	744.00000		
.00000000	.00000000	.00000000		
6.1000000	.00000000	733.00000		
6.5000000	.00000000	722.00000		
9.4000000	.00000000	847.00000		
1				
.11690900+009	150.00000	37500.000	20000000.	10000.000

2463000.0	1.0000000	.88600000		
.00000000	47.000000	37500.000	15000000.	10284.000
8615000.0	1.0000000	.49500000		
22865000.	18.000000	37500.000	1000000.0	4120.0000
.00000000	1.0000000	11.212000		
34298000.	18.000000	37500.000	1000000.0	6200.0000
777000.00	1.0000000	11.212000		
46764000.	172.00000	37500.000	10000000.	5000.0000
985000.00	1.0000000	.88600000		

RECEIVING OPERATIONS

3	1			
.18000000-001	.13700000	.59000000-001	.77000000-001	.31100000
.23300000	.10000000+000	.65000000-001		
.18000000-001	.13700000	.59000000-001	.77000000-001	.31100000
.23300000	.10000000+000	.65000000-001		
.18000000-001	.13700000	.59000000-001	.77000000-001	.31100000
.23300000	.10000000+000	.65000000-001		
.18000000-001	.13700000	.59000000-001	.77000000-001	.31100000
.23300000	.10000000+000	.65000000-001		

UNDERGROUND OPERATION

2661000.0	26.000000	37500.000	4100.0000	16.000000
2406000.0				
1.0000000				
1000.0000				
.00000000	2.0000000	37500.000	4000.0000	16.000000
33000.000				
.00000000				
.00000000				

0	0			
.00000000				
1	.88000000	.88000000	.50000000	1
1	0	2024		
.00000000	.00000000	.00000000	.00000000	.00000000
.00000000				
.00000000	4.6900000			
.00000000	2.0000000	56375.000	5000.0000	10.000000
.00000000				
.00000000	.64000000-003	.15180000	.37000000	
.00000000	7.0000000	77000.000	50000.000	12.000000
1330000.0	1.0000000	85.000000		
.00000000	2.0000000	56375.000	5000.0000	10.000000
33000.000	.00000000	85.000000		
.00000000	.00000000	.00000000	.00000000	.00000000
.00000000	.00000000	85.000000		

SHAFT OPERATIONS

120	0	85.700000	.00000000	2932.0000
.28800000	2	.00000000	.00000000	1
.00000000	.00000000			
.00000000	728.00000	732.00000	.00000000	733.00000

722.00000	771.00000			
.00000000	4608000.0	24816000.	.00000000	.00000000
.00000000	14696000.			
.00000000	3346000.0	10528000.	.00000000	.00000000
.00000000	14217000.			
1				
0				
SURFACE OPERATIONS				
.18000000-001	.13700000	.59000000-001	.77000000-001	.31100000
.23300000	.10000000+000	.65000000-001		
5	3800.0000			
ON-SITE IMPROVEMENTS				
.00000000	.00000000	1.0000000	.00000000	.00000000
.00000000	1.0000000	1140000.0		
VENTILATION STRUCTURES & FLOW PATHS				
30281000.	74.000000	1.0000000	33330.000	30.000000
51276000.	1.0000000	5456000.0		
SUPPORT (NO UTILITIES)				
7167000.0	424.00000	1.0000000	33330.000	30.000000
35121000.	1.0000000	4052000.0		
EXTRA				
.00000000	.00000000	1.0000000	.00000000	.00000000
.00000000	.00000000	.00000000		
EXTRA-EXTRA				
.00000000	.00000000	1.0000000	.00000000	.00000000
.00000000	.00000000	.00000000		
UTILITIES				
13.200000	.00000000	.00000000	.00000000	.35200000
.00000000	.00000000	3950.0000	3160.0000	.10900000
1.2700000	4650.0000	.00000000	.00000000	.00000000
3.8200000	.00000000	5000.0000	.00000000	.00000000
PRE-OPERATIONS				
1.0000000	.00000000	.00000000	.00000000	.00000000
.00000000	.00000000	.00000000		
1.0000000	.00000000	.00000000	.00000000	.00000000
.00000000	.00000000	.00000000		
1.0000000	.00000000	.00000000	.00000000	.00000000
.00000000	.00000000	.00000000		
13333000.	.00000000	.00000000	.00000000	.12000000
1				
POST-OPERATIONS				
1.0000000				
1.0000000				
1.0000000				
.00000000	.00000000	1	.50000000	.00000000
24				
0	.00000000			
.00000000	.00000000	.00000000	.00000000	.00000000
.00000000				

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 1 MINE DEVELOPMENT PARAMETERS FOR A REPOSITORY IN SALT

LOCATION	GULF INTERIOR
YEAR CONSTRUCTION BEGINS	1990
YEAR OPERATION BEGINS	1998
DEPTH, METERS (A)	727.0
TOTAL SURFACE AREA, HECTARES (B)	3800.0
TOTAL UNDERGROUND AREA, HECTARES (B)	2000.0
MINING METHOD	BORING
CENTRAL PILLAR AREA, HECTARES (B)	380.0
MAIN CORR.	
WIDTH, METERS (A)	4.9
HEIGHT, METERS (A)	4.6
LENGTH, METERS (A)	147103.0
PASSAGEWAYS	
CORRIDORS/PANEL	2
WIDTH, METERS (A)	9.2
HEIGHT, METERS (A)	4.6
LENGTH BY WASTE, METERS (A)	
HLW-	1195.0
GEOMETRIC EXCLUSION FACTOR	90.0
SITE PREPARATION COST, \$/HECTARE (B)	11211.0
DISCOUNT RATE, PERCENT	.0

B.303

WASTE TYPE

HLW-

EXTRACTION RATIO	.154
PANEL WIDTH, METERS (A)	192.0
PANEL LENGTH, METERS (A)	1370.0
ROOM WIDTH, METERS (A)	4.6
ROOM HEIGHT, METERS (A)	6.8
ROOMS PER PANEL	46

(A) 1 METER = 3.281 FEET
(B) 1 HECTARE = 2.471 ACRES

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE; 09/25/84
TIME; 16.18.11

TABLE 2 RECEIVING AND PROCESSING FOR A REPOSITORY IN SALT

	RECEIVING PACKAGING	

R.H. WASTE PROCESSING RATE PER MODULE, UNITS/SHIFT	2.1	17.2
G.H. WASTE PROCESSING RATE PER MODULE, UNITS/SHIFT	.0	
	WASTE TYPE	

	HLW-	

RAIL CONTAINER CAPACITY, UNITS	4.0	
TRUCK CONTAINER CAPACITY, UNITS	1.0	

B.305

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 3 WASTE SHIPPING DISTRIBUTION FOR A REPOSITORY IN SALT

PERCENTAGE OF UNITS SHIPPED BY RAIL FOR:

YEAR	HLW-
1990	100.0
1991	100.0
1992	100.0
1993	100.0
1994	100.0
1995	100.0
1996	100.0
1997	100.0
1998	100.0
1999	100.0
2000	100.0
2001	100.0
2002	100.0
2003	100.0
2004	100.0
2005	100.0
2006	100.0
2007	100.0
2008	100.0
2009	100.0
2010	100.0
2011	100.0
2012	100.0
2013	100.0
2014	100.0
2015	100.0
2016	100.0
2017	100.0
2018	100.0
2019	100.0
2020	100.0
2021	100.0
2022	100.0
2023	100.0

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE; 09/25/84
TIME; 16.18.11

TABLE 4A RECEIVING AND PROCESSING FOR A REPOSITORY IN SALT

YEAR	NUMBER OF MODULES REQUIRED		
	REMOTE-HANDLED RECEIVING	REMOTE-HANDLED PACKAGING	CONTACT-HANDLED RECEIVING
1990	0	0	0
1991	0	0	0
1992	0	0	0
1993	0	0	0
1994	0	0	0
1995	0	0	0
1996	0	0	0
1997	0	0	0
1998	1	1	0
1999	1	1	0
2000	1	1	0
2001	1	1	0
2002	1	1	0
2003	1	1	0
2004	1	1	0
2005	1	1	0
2006	1	1	0
2007	1	1	0
2008	1	1	0
2009	1	1	0
2010	1	1	0
2011	1	1	0
2012	1	1	0
2013	1	1	0
2014	1	1	0
2015	1	1	0
2016	1	1	0
2017	1	1	0
2018	1	1	0
2019	1	1	0
2020	1	1	0
2021	1	1	0
2022	1	1	0
2023	1	1	0

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 4B RECEIVING AND PROCESSING FOR A REPOSITORY IN SALT

NUMBER OF CONTAINER SHIPMENTS FOR:		
YEAR	HLW-	TOTAL
1990	.0	.0
1991	.0	.0
1992	.0	.0
1993	.0	.0
1994	.0	.0
1995	.0	.0
1996	.0	.0
1997	.0	.0
1998	46.2	46.2
1999	46.2	46.2
2000	46.2	46.2
2001	46.2	46.2
2002	46.2	46.2
2003	77.0	77.0
2004	77.0	77.0
2005	77.0	77.0
2006	77.0	77.0
2007	77.0	77.0
2008	77.0	77.0
2009	77.0	77.0
2010	77.0	77.0
2011	77.0	77.0
2012	77.0	77.0
2013	77.0	77.0
2014	77.0	77.0
2015	77.0	77.0
2016	77.0	77.0
2017	77.0	77.0
2018	77.0	77.0
2019	77.0	77.0
2020	77.0	77.0
2021	77.0	77.0
2022	77.0	77.0
2023	25.5	25.5
	1796.7	1796.7

B.308

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 5A. WASTE OPERATIONS PARAMETERS FOR HLW-
IN SALT

YEAR	NUMBER OF SHIFTS	UNITS RECEIVED	NUMBER OF PACKAGES	NUMBER OF PERSONNEL	NUMBER OF UNDERGROUND TRANSPORTERS	NUMBER OF EMPLACEMENT MACHINES	NUMBER OF DRILLING MACHINES	NUMBER OF ROOMS FILLED
1990	.00	.0	0	.0				.0
1991	.00	.0	0	.0				.0
1992	.00	.0	0	.0				.0
1993	.00	.0	0	.0				.0
1994	.00	.0	0	.0				.0
1995	.00	.0	0	.0				.0
1996	.00	.0	0	.0				.0
1997	.00	.0	0	.0				.0
1998	.12	185.0	185.0	8.8				14.2
1999	.12	185.0	185.0	8.8				14.2
2000	.12	185.0	185.0	8.8				14.2
2001	.12	185.0	185.0	8.8				14.2
2002	.12	185.0	185.0	8.8				14.2
2003	.19	308.0	308.0	14.4				23.7
2004	.19	308.0	308.0	14.4				23.7
2005	.19	308.0	308.0	14.4				23.7
2006	.19	308.0	308.0	14.4				23.7
2007	.19	308.0	308.0	14.4				23.7
2008	.19	308.0	308.0	14.4				23.7
2009	.19	308.0	308.0	14.4				23.7
2010	.19	308.0	308.0	14.4				23.7
2011	.19	308.0	308.0	14.4				23.7
2012	.19	308.0	308.0	14.4				23.7
2013	.19	308.0	308.0	14.4				23.7
2014	.19	308.0	308.0	14.4				23.7
2015	.19	308.0	308.0	14.4				23.7
2016	.19	308.0	308.0	14.4				23.7
2017	.19	308.0	308.0	14.4				23.7
2018	.19	308.0	308.0	14.4				23.7
2019	.19	308.0	308.0	14.4				23.7
2020	.19	308.0	308.0	14.4				23.7
2021	.19	308.0	308.0	14.4				23.7
2022	.19	308.0	308.0	14.4				23.7
2023	.06	102.0	102.0	5.1				7.8
TOTAL		7187.0	7187					552.8

B.309

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 5B. WASTE OPERATIONS PARAMETERS SUMMARY FOR A REPOSITORY IN SALT

YEAR	NUMBER OF SHIFTS	UNITS RECEIVED	NUMBER OF PACKAGES	NUMBER OF PERSONNEL	NUMBER OF UNDERGROUND TRANSPORTERS	NUMBER OF EMPLACEMENT MACHINES	NUMBER OF DRILLING MACHINES	NUMBER OF ROOMS FILLED
1990	.00	.0	0	.0	.0	.0	.0	.0
1991	.00	.0	0	.0	.0	.0	.0	.0
1992	.00	.0	0	.0	.0	.0	.0	.0
1993	.00	.0	0	.0	.0	.0	.0	.0
1994	.00	.0	0	.0	.0	.0	.0	.0
1995	.00	.0	0	.0	.0	.0	.0	.0
1996	.00	.0	0	.0	.0	.0	.0	.0
1997	.00	.0	0	.0	.0	.0	.0	.0
1998	.12	185.0	185	8.8	1.0	.0	1.0	14.2
1999	.12	185.0	185	8.8	1.0	.0	1.0	14.2
2000	.12	185.0	185	8.8	1.0	.0	1.0	14.2
2001	.12	185.0	185	8.8	1.0	.0	1.0	14.2
2002	.12	185.0	185	8.8	1.0	.0	1.0	14.2
2003	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2004	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2005	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2006	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2007	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2008	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2009	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2010	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2011	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2012	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2013	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2014	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2015	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2016	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2017	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2018	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2019	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2020	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2021	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2022	.19	308.0	308	14.4	1.0	.0	1.0	23.7
2023	.06	102.0	102	5.1	1.0	.0	1.0	7.8
TOTAL		7187.0	7187					552.8

B.310

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 6 EMPLACEMENT PACKAGE THERMAL OUTPUT FOR A REPOSITORY IN SALT

HLW-		
YEAR	NUMBER OF PACKAGES	AVERAGE HEAT RATE, KW/PACKAGE
1990	0	10.00
1991	0	10.00
1992	0	10.00
1993	0	10.00
1994	0	10.00
1995	0	10.00
1996	0	10.00
1997	0	10.00
1998	185	10.00
1999	185	10.00
2000	185	10.00
2001	185	10.00
2002	185	10.00
2003	308	10.00
2004	308	10.00
2005	308	10.00
2006	308	10.00
2007	308	10.00
2008	308	10.00
2009	308	10.00
2010	308	10.00
2011	308	10.00
2012	308	10.00
2013	308	10.00
2014	308	10.00
2015	308	10.00
2016	308	10.00
2017	308	10.00
2018	308	10.00
2019	308	10.00
2020	308	10.00
2021	308	10.00
2022	308	10.00
2023	102	10.00

TOTAL	7187	

B.311

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 7 WASTE LOADING PARAMETERS FOR A REPOSITORY IN SALT

YEAR	HLW-		
	PITCH, METERS	PACKAGES PER ROW	PACKAGES PER ROOM
1990	.0	0	0
1991	.0	0	0
1992	.0	0	0
1993	.0	0	0
1994	.0	0	0
1995	.0	0	0
1996	.0	0	0
1997	.0	0	0
1998	12.2	13	13
1999	12.2	13	13
2000	12.2	13	13
2001	12.2	13	13
2002	12.2	13	13
2003	12.2	13	13
2004	12.2	13	13
2005	12.2	13	13
2006	12.2	13	13
2007	12.2	13	13
2008	12.2	13	13
2009	12.2	13	13
2010	12.2	13	13
2011	12.2	13	13
2012	12.2	13	13
2013	12.2	13	13
2014	12.2	13	13
2015	12.2	13	13
2016	12.2	13	13
2017	12.2	13	13
2018	12.2	13	13
2019	12.2	13	13
2020	12.2	13	13
2021	12.2	13	13
2022	12.2	13	13
2023	12.2	13	13

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 8A. REPOSITORY DEVELOPMENT SCHEDULE FOR HLW-
IN SALT

YEAR	ROCK, THOUSAND OF TONNES (1)			ROOMS		AREA REQUIRED, ACRES	
	MINED	BACKFILLED	TRANSPORT TO SURFACE	EXCAVATED	FILLED	ANNUAL	CUMULATIVE
1990	0.	0.	0.	00	00	00	00
1991	21.	0.	21.	00	00	00	00
1992	63.	0.	63.	00	00	00	00
1993	63.	0.	63.	00	00	00	00
1994	898.	0.	898.	00	00	00	00
1995	4249.	0.	4249.	00	00	00	00
1996	4315.	0.	4315.	14.23	00	00	00
1997	2339.	0.	2339.	14.23	00	00	00
1998	215.	0.	215.	14.23	14.23	20.11	20.11
1999	215.	0.	215.	14.23	14.23	20.11	40.22
2000	215.	213.	2.	14.23	14.23	20.11	60.32
2001	356.	354.	2.	23.69	14.23	20.11	80.43
2002	356.	160.	197.	23.69	14.23	20.11	100.54
2003	358.	242.	116.	23.69	23.69	33.48	134.01
2004	358.	242.	116.	23.69	23.69	33.48	167.49
2005	358.	242.	116.	23.69	23.69	33.48	200.97
2006	358.	242.	116.	23.69	23.69	33.48	234.44
2007	358.	242.	116.	23.69	23.69	33.48	267.92
2008	358.	242.	116.	23.69	23.69	33.48	301.40
2009	358.	242.	116.	23.69	23.69	33.48	334.87
2010	358.	242.	116.	23.69	23.69	33.48	368.35
2011	358.	242.	116.	23.69	23.69	33.48	401.82
2012	358.	242.	116.	23.69	23.69	33.48	435.30
2013	358.	242.	116.	23.69	23.69	33.48	468.78
2014	358.	242.	116.	23.69	23.69	33.48	502.25
2015	358.	242.	116.	23.69	23.69	33.48	535.73
2016	358.	242.	116.	23.69	23.69	33.48	569.21
2017	358.	242.	116.	23.69	23.69	33.48	602.68
2018	358.	242.	116.	23.69	23.69	33.48	636.16
2019	358.	242.	116.	23.69	23.69	33.48	669.64
2020	358.	242.	116.	23.69	23.69	33.48	703.11
2021	121.	242.	121.	7.85	23.69	33.48	736.59
2022	4.	242.	238.	00	23.69	33.48	770.06
2023	1.	80.	79.	00	7.85	11.09	781.15
2024		7727.	7727.				
TOTAL	19875.	13373.	22832.	552.85	552.85		

(1) QUANTITIES INCLUDE ALL ROCK EXCAVATED (PANELS, CORRIDORS, SHAFTS, BOREHOLES, PILLARS, ETC.)

THOUSAND OF TONNES EXCAVATED FOR SHAFT ARE 212.

THOUSAND OF TONNES EXCAVATED FOR HOLE DRILLING ARE 87.

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 9 SUMMARY DEVELOPMENT SCHEDULE FOR A REPOSITORY IN SALT

YEAR	ROCK, THOUSAND OF TONNES (1)					ROOMS		WASTE STORAGE AREA ACRES		
	MINED	BACKFILLED	TRANSPORT TO SURFACE	SURFACE STORAGE	TRANSPORT OFFSITE	EXCAVATED	FILLED	ANNUAL REQUIRED	CUMULATIVE REQUIRED	REMAINING AVAILABLE
1990	0	0	0	0	0	0	0	0	0	0
1991	21	0	21	21	0	0	0	0	0	0
1992	63	0	63	85	0	0	0	0	0	0
1993	63	0	63	148	0	0	0	0	0	0
1994	898	0	898	1046	0	0	0	0	0	0
1995	4249	0	4249	5295	0	0	0	0	0	0
1996	4315	0	4315	9610	0	14.2	0	0	0	0
1997	2339	0	2339	11949	0	14.2	0	0	0	0
1998	215	0	215	12000	164	14.2	14.2	20.1	20.1	1509.9
1999	215	0	215	12000	215	14.2	14.2	20.1	40.2	1489.8
2000	215	213	2	12000	2	14.2	14.2	20.1	60.3	1469.7
2001	356	354	2	12000	2	23.7	14.2	20.1	80.4	1449.6
2002	356	160	197	12000	197	23.7	14.2	20.1	100.5	1429.5
2003	356	242	116	12000	116	23.7	23.7	33.5	134.0	1396.0
2004	356	242	116	12000	116	23.7	23.7	33.5	167.5	1362.5
2005	356	242	116	12000	116	23.7	23.7	33.5	201.0	1329.0
2006	356	242	116	12000	116	23.7	23.7	33.5	234.4	1295.6
2007	356	242	116	12000	116	23.7	23.7	33.5	267.9	1262.1
2008	356	242	116	12000	116	23.7	23.7	33.5	301.4	1228.6
2009	356	242	116	12000	116	23.7	23.7	33.5	334.9	1195.1
2010	356	242	116	12000	116	23.7	23.7	33.5	368.3	1161.7
2011	356	242	116	12000	116	23.7	23.7	33.5	401.8	1128.2
2012	356	242	116	12000	116	23.7	23.7	33.5	435.3	1094.7
2013	356	242	116	12000	116	23.7	23.7	33.5	468.8	1061.2
2014	356	242	116	12000	116	23.7	23.7	33.5	502.3	1027.7
2015	356	242	116	12000	116	23.7	23.7	33.5	535.7	994.3
2016	356	242	116	12000	116	23.7	23.7	33.5	569.2	960.8
2017	356	242	116	12000	116	23.7	23.7	33.5	602.7	927.3
2018	356	242	116	12000	116	23.7	23.7	33.5	636.2	893.8
2019	356	242	116	12000	116	23.7	23.7	33.5	669.6	860.4
2020	356	242	116	12000	116	23.7	23.7	33.5	703.1	826.9
2021	121	242	121	11879	0	7.8	23.7	33.5	736.6	793.4
2022	4	242	238	11641	0	0	23.7	33.5	770.1	759.9
2023	1	80	79	11562	0	0	7.8	11.1	781.2	748.8
2024		7727	7727	3835						
TOTAL	19875	13373	22832		2667	552.8	552.8			

(1) QUANTITIES INCLUDE ALL ROCK EXCAVATED (PANELS, CORRIDORS, SHAFTS, BOREHOLES, PILLARS, ETC.)

THOUSAND OF TONNES EXCAVATED FOR SHAFT ARE 212.

THOUSAND OF TONNES EXCAVATED FOR HOLE DRILLING ARE 87.

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 10 SHAFT PARAMETERS AND COSTS

	MEN AND MATERIALS	MINING VENT	WASTE TRANSPORT		EXHAUST		ROCK HOISTING	TOTAL
			REMOTE	CONTACT	MINING	WASTE		
DESIGN DIAMETER, METERS	.0	6.6	6.7	.0	6.1	6.5	9.4	
INCREMENTAL EXCAVATED RADIUS, METERS	.0	.0	.0	.0	.0	.0	.0	
EXCAVATED DEPTH, METERS	.0	769.0	744.0	.0	733.0	722.0	847.0	
LINED DEPTH, METERS	.0	728.0	732.0	.0	733.0	722.0	771.0	
WASTE TRANSPORT CAPACITY								
TRIPS PER SHIFT			16	32				48
PACKAGES PER TRIP								
HLW-			1					
ROCK HOISTING CAPACITY								
TRIPS PER SHIFT	.0						120	
TONNES PER TRIP	0.						86.	
NUMBER OF SHAFTS	0	1	1	0	1	1	1	5
SHAFT COSTS, MILLIONS OF 0 DOLLARS								
SINKING	.0	15.1	15.0	.0	12.5	13.8	37.2	93.5
WATER CONTROL	.0	.0	.0	.0	.0	.0	.0	.0
LINING	.0	.0	.0	.0	.0	.0	.0	.0
EQUIPMENT	.0	4.6	24.8	.0	.0	.0	14.7	44.1
SHAFT STATIONS	.0	3.3	10.5	.0	.0	.0	14.2	28.1
TOTAL	.0	23.0	50.4	.0	12.5	13.8	66.1	165.8

B.315

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 11 WASTE FACILITY CAPITAL COSTS FOR A REPOSITORY IN SALT

	MILLIONS OF	0 \$
WASTE RECEIVING MODULE - REMOTE HANDLED	40.00	
WASTE RECEIVING MODULE - CONTACT HANDLED	.00	
WASTE PACKAGING MODULE	147.80	
UNDERGROUND TRANSPORTATION FACILITY	2.66	
EMPLACEMENT EXCAVATION FACILITY	.00	
EMPLACEMENT SUPPORT FACILITY	.00	
ON-SITE IMPROVEMENTS	.00	
VENTILATION STRUCTURES & FLOW PATHS	30.28	
SUPPORT (NO UTILITIES)	7.17	
EXTRA	.00	
EXTRA-EXTRA	.00	
TOTAL	227.90	

B.316

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 12 EQUIPMENT LIFE AND COST FOR A REPOSITORY IN SALT

	LIFE		COSTS MILLIONS \$
SURFACE WASTE OPERATIONS			
RECEIVING MODULE - REMOTE HANDLED WASTE	1000000.	RECEIVING UNITS	.00
RECEIVING MODULE - CONTACT HANDLED WASTE	40000.	RECEIVING UNITS	.00
PACKAGING MODULE	1000000.	RECEIVING UNITS	63.73
UNDERGROUND WASTE OPERATIONS			
TRANSPORTER - REMOTE HANDLED WASTE	4100.	PACKAGES	4.81
TRANSPORTER - CONTACT HANDLED WASTE	4000.	PACKAGES	.00
EMPLACEMENT - REMOTE HANDLED WASTE	0.	PACKAGES	.00
EMPLACEMENT - CONTACT HANDLED WASTE	5000.	PACKAGES	.00
ROCK OPERATIONS			
EXCAVATION AND HAULAGE	20000000.	TONNES ROCK	2.46
HOISTING	15000000.	TONNES ROCK	25.84
ROCK STOCKPILE	1000000.	TONNES ROCK	.00
OFFSITE DISPOSAL	1000000.	TONNES ROCK	.00
BACKFILLING AND HAULAGE	10000000.	TONNES ROCK	.98
SUPPORT OPERATIONS			
ON-SITE IMPROVEMENTS	0.	YEARS	.00
VENTILATION STRUCTURES & FLOW PATHS	30.	YEARS	51.28
SUPPORT (NO UTILITIES)	30.	YEARS	35.12
EXTRA	0.	YEARS	.00
EXTRA-EXTRA	0.	YEARS	.00
EMPLACEMENT EXCAVATION METHODS			
HOLE DRILLING	50000.	PACKAGES	1.33
ROOM EMPLACEMENT	5000.	PACKAGES	.00
TRENCHING	0.	PACKAGES	.00
TOTAL			185.56

B.317

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 13 CONSTRUCTION AND PREPARATION COSTS FOR A REPOSITORY IN SALT
(MILLIONS OF 0 DOLLARS)

YEAR	LAND	LICENSE	SITE PREPARE	SHAFT CONSTRUCT	FACILITY CONSTRUCTION		ARCHITECT ENGINEER	STARTUP	TOTAL
					SURFACE	SUBSURFACE			
1990	13.33	.00	.77	.00	4.05	.05	1.06	.00	19.26
1991	.00	.00	5.84	16.58	30.86	.36	10.07	.00	63.70
1992	.00	.00	2.51	49.73	13.29	.16	9.45	.00	75.13
1993	.00	.00	3.28	49.73	17.34	.20	10.51	.00	81.06
1994	.00	.00	13.25	41.44	70.05	.83	25.01	.00	150.58
1995	.00	.00	9.93	8.29	52.48	.62	23.28	.00	94.59
1996	.00	.00	4.26	.00	22.52	.27	14.16	.00	41.22
1997	.00	.00	2.77	.00	14.64	.17	8.12	.00	25.70
1998					.00				.00
1999					.00				.00
2000					.00				.00
2001					.00				.00
2002					.00				.00
2003					.00				.00
2004					.00				.00
2005					.00				.00
2006					.00				.00
2007					.00				.00
2008					.00				.00
2009					.00				.00
2010					.00				.00
2011					.00				.00
2012					.00				.00
2013					.00				.00
2014					.00				.00
2015					.00				.00
2016					.00				.00
2017					.00				.00
2018					.00				.00
2019					.00				.00
2020					.00				.00
2021					.00				.00
2022					.00				.00
2023					.00				.00
TOTAL	13.33	.00	42.60	165.75	225.24	2.66	101.65	.00	551.25

B.318

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 14 MINING AND ROCK REMOVAL OPERATIONS COSTS SUMMARY FOR A REPOSITORY IN SALT
----- COSTS IN MILLIONS OF 0 DOLLARS

YEAR	MINING (1)	HOISTING (1)	SURFACE HANDLE AND STORE (2)	OFFSITE DISPOSAL (2)	TOTAL
1990	.00	.00	.00	.00	.00
1991	.00	.00	.26	.00	.26
1992	.00	.00	.79	.00	.79
1993	.00	.00	.79	.00	.79
1994	14.20	1.64	11.22	.00	27.06
1995	71.20	8.23	53.09	.00	132.52
1996	72.48	8.38	53.91	.00	134.77
1997	39.28	4.54	29.22	.00	73.04
1998	3.57	.42	2.69	1.41	8.09
1999	3.57	.42	2.69	1.85	8.53
2000	3.57	.00	.03	.02	3.62
2001	5.95	.00	.03	.02	6.00
2002	5.95	.38	2.46	1.69	10.48
2003	5.95	.23	1.45	1.00	8.62
2004	5.95	.23	1.45	1.00	8.62
2005	5.95	.23	1.45	1.00	8.62
2006	5.95	.23	1.45	1.00	8.62
2007	5.95	.23	1.45	1.00	8.62
2008	5.95	.23	1.45	1.00	8.62
2009	5.95	.23	1.45	1.00	8.62
2010	5.95	.23	1.45	1.00	8.62
2011	5.95	.23	1.45	1.00	8.62
2012	5.95	.23	1.45	1.00	8.62
2013	5.95	.23	1.45	1.00	8.62
2014	5.95	.23	1.45	1.00	8.62
2015	5.95	.23	1.45	1.00	8.62
2016	5.95	.23	1.45	1.00	8.62
2017	5.95	.23	1.45	1.00	8.62
2018	5.95	.23	1.45	1.00	8.62
2019	5.95	.23	1.45	1.00	8.62
2020	5.95	.23	1.45	1.00	8.62
2021	1.97	.23	1.51	.00	3.72
2022	.00	.46	2.98	.00	3.44
2023	.00	.15	.99	.00	1.14
2024	.00	15.01	96.55	.00	111.56
TOTAL	328.82	43.94	285.28	22.93	680.97

(1) QUANTITIES INCLUDE ALL ROCK EXCAVATED EXCEPT SHAFT (PANELS, CORRIDORS, BOREHOLES, PILLARS, ETC.)

(2) INCLUDES ALL ROCK EXCAVATED (PANELS, CORRIDORS, SHAFTS, BOREHOLES, PILLARS, ETC.)

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 15 WASTE OPERATIONS LABOR COST SUMMARY FOR A REPOSITORY IN SALT

MILLIONS OF 0 DOLLARS

YEAR	RECEIVING	PACKAGING	TRANSPORTATION	EMPLACEMENT EXCAVATION	EMPLACEMENT OPERATIONS	OTHER	TOTAL
1990	.00	.00	.00	.00	.00	.00	.00
1991	.00	.00	.00	.00	.00	.00	.00
1992	.00	.00	.00	.00	.00	.00	.00
1993	.00	.00	.00	.00	.00	.00	.00
1994	.00	.00	.00	.00	.00	.00	.00
1995	.00	.00	.00	.00	.00	.00	.00
1996	.00	.00	.00	.00	.00	.00	.00
1997	.00	.00	.00	.00	.00	.00	.00
1998	.03	.17	.11	.03	.00	16.60	16.95
1999	.03	.17	.11	.03	.00	16.60	16.95
2000	.03	.17	.11	.03	.00	16.60	16.95
2001	.03	.17	.11	.03	.00	16.60	16.95
2002	.03	.17	.11	.03	.00	16.60	16.95
2003	.05	.28	.19	.03	.00	16.60	17.16
2004	.05	.28	.19	.03	.00	16.60	17.16
2005	.05	.28	.19	.03	.00	16.60	17.16
2006	.05	.28	.19	.03	.00	16.60	17.16
2007	.05	.28	.19	.03	.00	16.60	17.16
2008	.05	.28	.19	.03	.00	16.60	17.16
2009	.05	.28	.19	.03	.00	16.60	17.16
2010	.05	.28	.19	.03	.00	16.60	17.18
2011	.05	.28	.19	.03	.00	16.60	17.16
2012	.05	.28	.19	.03	.00	16.60	17.16
2013	.05	.28	.19	.03	.00	16.60	17.18
2014	.05	.28	.19	.03	.00	16.60	17.18
2015	.05	.28	.19	.03	.00	16.60	17.16
2016	.05	.28	.19	.03	.00	16.60	17.16
2017	.05	.28	.19	.03	.00	16.60	17.16
2018	.05	.28	.19	.03	.00	16.60	17.16
2019	.05	.28	.19	.03	.00	16.60	17.16
2020	.05	.28	.19	.03	.00	16.60	17.16
2021	.05	.28	.19	.03	.00	16.60	17.16
2022	.05	.28	.19	.03	.00	16.60	17.16
2023	.02	.09	.06	.03	.00	16.60	16.81
TOTAL	1.23	6.59	4.39	.90	.00	431.56	444.66

B.320

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE; 09/25/84
TIME; 16.18.11

TABLE 16 WASTE OPERATIONS MATERIALS COST SUMMARY FOR A REPOSITORY IN SALT

MILLIONS OF 0 DOLLARS

YEAR	RECEIVING	PACKAGING	TRANSPORTATION	EMPLACEMENT EXCAVATION	EMPLACEMENT OPERATIONS	OTHER	TOTAL
1990	.00	.00	.00	.00	.00	.00	.00
1991	.00	.00	.00	.00	.00	.00	.00
1992	.00	.00	.00	.00	.00	.00	.00
1993	.00	.00	.00	.00	.00	.00	.00
1994	.00	.00	.00	.00	.00	.00	.00
1995	.00	.00	.00	.00	.00	.00	.00
1996	.00	.00	.00	.00	.00	.00	.00
1997	.00	.00	.00	.00	.00	.00	.00
1998	.06	96.96	.18	.02	.00	10.65	17.86
1999	.06	96.96	.18	.02	.00	10.65	17.86
2000	.06	96.96	.18	.02	.00	10.65	17.86
2001	.06	96.96	.18	.02	.00	10.65	17.86
2002	.06	96.96	.18	.02	.00	10.65	17.86
2003	.09	11.59	.31	.03	.00	10.65	22.66
2004	.09	11.59	.31	.03	.00	10.65	22.66
2005	.09	11.59	.31	.03	.00	10.65	22.66
2006	.09	11.59	.31	.03	.00	10.65	22.66
2007	.09	11.59	.31	.03	.00	10.65	22.66
2008	.09	11.59	.31	.03	.00	10.65	22.66
2009	.09	11.59	.31	.03	.00	10.65	22.66
2010	.09	11.59	.31	.03	.00	10.65	22.66
2011	.09	11.59	.31	.03	.00	10.65	22.66
2012	.09	11.59	.31	.03	.00	10.65	22.66
2013	.09	11.59	.31	.03	.00	10.65	22.66
2014	.09	11.59	.31	.03	.00	10.65	22.66
2015	.09	11.59	.31	.03	.00	10.65	22.66
2016	.09	11.59	.31	.03	.00	10.65	22.66
2017	.09	11.59	.31	.03	.00	10.65	22.66
2018	.09	11.59	.31	.03	.00	10.65	22.66
2019	.09	11.59	.31	.03	.00	10.65	22.66
2020	.09	11.59	.31	.03	.00	10.65	22.66
2021	.09	11.59	.31	.03	.00	10.65	22.66
2022	.09	11.59	.31	.03	.00	10.65	22.66
2023	.03	3.84	.10	.01	.00	10.65	14.63
TOTAL	2.18	270.37	7.19	.61	.00	276.85	557.20

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 17 WASTE OPERATIONS EQUIPMENT COST SUMMARY FOR A REPOSITORY IN SALT

MILLIONS OF 0 DOLLARS

YEAR	RECEIVING	PACKAGING	TRANSPORTATION	EMPLACEMENT EXCAVATION	EMPLACEMENT OPERATIONS	OTHER	TOTAL
1990	.00	.00	.00	.00	.00	.00	.00
1991	.00	.00	.00	.00	.00	.00	.00
1992	.00	.00	.00	.00	.00	.00	.00
1993	.00	.00	.00	.00	.00	.00	.00
1994	.00	.00	.00	.00	.00	.00	.00
1995	.00	.00	.00	.00	.00	.00	.00
1996	.00	.00	.00	.00	.00	.00	.00
1997	.00	.00	.00	.00	.00	.00	.00
1998	.00	63.73	2.41	1.33	.00	86.40	153.86
1999	.00	.00	.00	.00	.00	.00	.00
2000	.00	.00	.00	.00	.00	.00	.00
2001	.00	.00	.00	.00	.00	.00	.00
2002	.00	.00	.00	.00	.00	.00	.00
2003	.00	.00	.00	.00	.00	.00	.00
2004	.00	.00	.00	.00	.00	.00	.00
2005	.00	.00	.00	.00	.00	.00	.00
2006	.00	.00	.00	.00	.00	.00	.00
2007	.00	.00	.00	.00	.00	.00	.00
2008	.00	.00	.00	.00	.00	.00	.00
2009	.00	.00	.00	.00	.00	.00	.00
2010	.00	.00	.00	.00	.00	.00	.00
2011	.00	.00	.00	.00	.00	.00	.00
2012	.00	.00	.00	.00	.00	.00	.00
2013	.00	.00	2.41	.00	.00	.00	2.41
2014	.00	.00	.00	.00	.00	.00	.00
2015	.00	.00	.00	.00	.00	.00	.00
2016	.00	.00	.00	.00	.00	.00	.00
2017	.00	.00	.00	.00	.00	.00	.00
2018	.00	.00	.00	.00	.00	.00	.00
2019	.00	.00	.00	.00	.00	.00	.00
2020	.00	.00	.00	.00	.00	.00	.00
2021	.00	.00	.00	.00	.00	.00	.00
2022	.00	.00	.00	.00	.00	.00	.00
2023	.00	.00	.00	.00	.00	.00	.00
TOTAL	.00	63.73	4.81	1.33	.00	86.40	156.27

B.322

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 18 UTILITIES REQUIREMENTS AND COST SUMMARY FOR A REPOSITORY IN SALT

REQUIREMENTS					COSTS, MILLIONS OF 0 DOLLARS				
	ELECTRICITY THOUSANDS OF KWH	STEAM THOUSANDS OF MJ(A)	WATER MILLIONS OF LITERS	DIESEL THOUSANDS OF LITERS	ELECTRICITY	STEAM	WATER	DIESEL	TOTAL
1990	.00	.00	.00	.00	.00	.00	.00	.00	.00
1991	.00	.00	.00	.00	.00	.00	.00	.00	.00
1992	.00	.00	.00	.00	.00	.00	.00	.00	.00
1993	.00	.00	.00	.00	.00	.00	.00	.00	.00
1994	.00	.00	.00	.00	.00	.00	.00	.00	.00
1995	.00	.00	.00	.00	.00	.00	.00	.00	.00
1996	.00	.00	.00	.00	.00	.00	.00	.00	.00
1997	.00	.00	.00	.00	.00	.00	.00	.00	.00
1998	97014.81	.00	3.82	1250.00	1.28	.00	.00	.44	1.72
1999	97014.81	.00	3.82	1250.00	1.28	.00	.00	.44	1.72
2000	97014.81	.00	3.82	1250.00	1.28	.00	.00	.44	1.72
2001	100837.91	.00	3.82	1250.00	1.33	.00	.00	.44	1.77
2002	100589.51	.00	3.82	1250.00	1.33	.00	.00	.44	1.77
2003	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2004	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2005	107498.07	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2006	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2007	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2008	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2009	107498.07	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2010	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2011	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2012	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2013	107498.07	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2014	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2015	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2016	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2017	107498.07	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2018	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2019	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2020	107249.67	.00	3.82	1250.00	1.42	.00	.00	.44	1.86
2021	97530.69	.00	3.82	1250.00	1.29	.00	.00	.44	1.73
2022	29362.59	.00	3.82	1250.00	.39	.00	.00	.44	.83
2023	28203.70	.00	3.82	1250.00	.37	.00	.00	.44	.81
<hr/>									
TOTAL	2579056.37	.00	99.32	32500.00	34.04	.00	.00	11.44	45.48

(A) MJ(MEGA JOULE)= .977 POUNDS OF STEAM AT 100C.

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 19A. WASTE OPERATIONS COST SUMMARY FOR HLW-
----- IN SALT
MILLIONS OF 0 DOLLARS

YEAR	RECEIVING	PACKAGING	TRANSPORTATION	EMPLACEMENT EXCAVATION	EMPLACEMENT OPERATIONS	OTHER	TOTAL
1990	.00	.00	.00	.00	.00	.00	.00
1991	.00	.00	.00	.00	.00	.00	.00
1992	.00	.00	.00	.00	.00	.00	.00
1993	.00	.00	.00	.00	.00	.00	.00
1994	.00	.00	.00	.00	.00	.00	.00
1995	.00	.00	.00	.00	.00	.00	.00
1996	.00	.00	.00	.00	.00	.00	.00
1997	.00	.00	.00	.00	.00	.00	.00
1998	.09	70.86	2.70	1.38	.00	115.36	190.39
1999	.09	7.13	.30	.05	.00	28.97	36.53
2000	.09	7.13	.30	.05	.00	28.97	36.53
2001	.09	7.13	.30	.05	.00	29.02	36.58
2002	.09	7.13	.30	.05	.00	29.01	36.58
2003	.11	11.87	.50	.06	.00	29.10	41.67
2004	.11	11.87	.50	.06	.00	29.11	41.68
2005	.11	11.87	.50	.06	.00	29.10	41.67
2006	.11	11.87	.50	.06	.00	29.10	41.67
2007	.11	11.87	.50	.06	.00	29.10	41.67
2008	.11	11.87	.50	.06	.00	29.11	41.68
2009	.11	11.87	.50	.06	.00	29.10	41.67
2010	.11	11.87	.50	.06	.00	29.10	41.67
2011	.11	11.87	.50	.06	.00	29.10	41.67
2012	.11	11.87	.50	.06	.00	29.11	44.08
2013	.11	11.87	2.90	.06	.00	29.10	41.67
2014	.11	11.87	.50	.06	.00	29.10	41.67
2015	.11	11.87	.50	.06	.00	29.10	41.67
2016	.11	11.87	.50	.06	.00	29.10	41.67
2017	.11	11.87	.50	.06	.00	29.11	41.68
2018	.11	11.87	.50	.06	.00	29.10	41.67
2019	.11	11.87	.50	.06	.00	29.10	41.67
2020	.11	11.87	.50	.06	.00	29.10	41.67
2021	.11	11.87	.50	.06	.00	28.97	41.55
2022	.15	11.87	.50	.06	.00	28.07	40.85
2023	.05	3.93	.16	.04	.00	28.06	32.25
TOTAL	3.41	340.69	16.39	2.84	.00	840.29	1203.62

B.324

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 19B. WASTE OPERATIONS COST SUMMARY FOR A REPOSITORY IN SALT

MILLIONS OF 0 DOLLARS

YEAR	RECEIVING	PACKAGING	TRANSPORTATION	EMPLACEMENT EXCAVATION	EMPLACEMENT OPERATIONS	OTHER	TOTAL
1990	.00	.00	.00	.00	.00	.00	.00
1991	.00	.00	.00	.00	.00	.00	.00
1992	.00	.00	.00	.00	.00	.00	.00
1993	.00	.00	.00	.00	.00	.00	.00
1994	.00	.00	.00	.00	.00	.00	.00
1995	.00	.00	.00	.00	.00	.00	.00
1996	.00	.00	.00	.00	.00	.00	.00
1997	.00	.00	.00	.00	.00	.00	.00
1998	.09	70.86	2.70	1.38	.00	115.36	190.39
1999	.09	7.13	.30	.05	.00	28.97	36.53
2000	.09	7.13	.30	.05	.00	28.97	36.53
2001	.09	7.13	.30	.05	.00	29.02	36.58
2002	.09	7.13	.30	.05	.00	29.01	36.58
2003	.15	.87	.50	.06	.00	29.10	41.67
2004	.15	.87	.50	.06	.00	29.10	41.67
2005	.15	.87	.50	.06	.00	29.11	41.68
2006	.15	.87	.50	.06	.00	29.10	41.67
2007	.15	.87	.50	.06	.00	29.10	41.67
2008	.15	.87	.50	.06	.00	29.10	41.67
2009	.15	.87	.50	.08	.00	29.11	41.68
2010	.15	.87	.50	.06	.00	29.10	41.67
2011	.15	.87	.50	.06	.00	29.10	41.67
2012	.15	.87	.50	.06	.00	29.10	41.67
2013	.15	.87	2.00	.06	.00	29.11	44.08
2014	.15	.87	.50	.08	.00	29.10	41.67
2015	.15	.87	.50	.06	.00	29.10	41.67
2016	.15	.87	.50	.06	.00	29.10	41.67
2017	.15	.87	.50	.06	.00	29.11	41.68
2018	.15	.87	.50	.06	.00	29.10	41.67
2019	.15	.87	.50	.06	.00	29.10	41.67
2020	.15	.87	.50	.06	.00	29.10	41.67
2021	.15	.87	.50	.06	.00	28.97	41.55
2022	.15	.87	.50	.06	.00	28.07	40.65
2023	.05	3.93	.16	.04	.00	28.06	32.25
TOTAL	3.41	340.69	16.39	2.84	.00	840.29	1203.62

B.325

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 20 BACKFILLING AND DECOMMISSIONING SUMMARY FOR A REPOSITORY IN SALT

YEAR	BACKFILL MATERIALS, THOUSAND TONNES			COSTS IN MILLIONS OF 0 DOLLARS				
	ROCK	OTHER	TOTAL	BACKFILLING	SEALING	DECOMMISSION	SURVEILLANCE	TOTAL
1990	.00	.00	.00	.00	.00	.00	.00	.00
1991	.00	.00	.00	.00	.00	.00	.00	.00
1992	.00	.00	.00	.00	.00	.00	.00	.00
1993	.00	.00	.00	.00	.00	.00	.00	.00
1994	.00	.00	.00	.00	.00	.00	.00	.00
1995	.00	.00	.00	.00	.00	.00	.00	.00
1996	.00	.00	.00	.00	.00	.00	.00	.00
1997	.00	.00	.00	.00	.00	.00	.00	.00
1998	.00	.00	.00	.00	.00	.00	.00	.00
1999	.00	.00	.00	.00	.00	.00	.00	.00
2000	212.74	.00	212.74	1.31	.00	.00	.00	1.31
2001	354.18	.00	354.18	2.18	.00	.00	.00	2.18
2002	159.69	.00	159.69	.98	.00	.00	.00	.98
2003	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2004	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2005	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2006	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2007	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2008	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2009	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2010	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2011	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2012	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2013	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2014	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2015	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2016	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2017	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2018	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2019	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2020	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2021	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2022	241.94	.00	241.94	1.49	.00	.00	.00	1.49
2023	80.12	.00	80.12	.49	.00	.00	.00	.49
2024	7727.12	.00	7727.12	47.52	.00	112.62	.00	160.14
TOTAL	13372.60	.00	13372.60	82.24	.00	112.62	.00	194.86

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 21 COST SUMMARY FOR A REPOSITORY IN SALT
----- MILLIONS OF 0 DOLLARS

FROM TABLE	COST CATEGORY	WASTE TYPE	
		HLW-	TOTAL
13	CONSTRUCTION AND PREOPERATION		
	LAND	13.33	13.33
	LICENSE	.00	.00
	SITE PREPARATION	42.60	42.60
	SHAFT	165.75	165.75
	SURFACE FACILITIES	225.24	225.24
	UNDERGROUND FACILITIES	2.66	2.66
	ARCHITECT ENGINEER	101.65	101.65
	STARTUP	.00	.00
	SUBTOTAL	551.25	551.25
14	MINING AND ROCK REMOVAL OPERATIONS		
	MINING	328.82	328.82
	HOISTING	43.94	43.94
	SURFACE HANDLE AND STORE	285.28	285.28
	OFFSITE DISPOSAL	22.93	22.93
	SUBTOTAL	680.97	680.97
19	WASTE OPERATIONS		
	RECEIVING	3.41	3.41
	PACKAGING	340.69	340.69
	TRANSPORTATION	16.39	16.39
	EMPLACEMENT EXCAVATION	2.84	2.84
	EMPLACEMENT OPERATIONS	.00	.00
	OTHER	840.29	840.29
	SUBTOTAL	1203.62	1203.62
20	BACKFILLING AND DECOMMISSIONING		
	BACKFILLING	82.24	82.24
	SEALING	.00	.00
	DECOMMISSIONING	112.62	112.62
	SURVEILLANCE	.00	.00
	SUBTOTAL	194.86	194.86
	CONTINGENCY	526.14	526.14
TOTAL		3156.84	3156.84

B.327

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 22. REPOSITORY COST MATRIX

SYSTEM	COST IN MILLIONS OF 0 DOLLARS										
	CAPITAL CONSTRUCTION			OPERATIONS				DECOMMISSIONING			
	FACILITY CONSTR.	CAPITAL EQUIP.	SUBTOTAL	LABOR	MATERIAL	EQUIP REPLACE.	UTILITY	SUBTOTAL	BACKFILL & SEAL	DECOMM.	TOTAL
SITE											
LAND	13.33		13.33								13.33
LICENSING	.00		.00								.00
SITE PREPARATION	42.60		42.60								42.60
OTHER PRESTARTUP COSTS	101.65		101.65								101.65
SUBTOTAL	157.59		157.59								157.59
WASTE PACKAGE PREPARATION											
HLW-	147.80	63.73	211.52	6.59	270.37	.00	.00	276.96		73.90	562.38
SUBTOTAL	147.80	63.73	211.52	6.59	270.37	.00	.00	276.96		73.90	562.38
WASTE HANDLING AND EMPLACEMENT											
RECEIVING	40.00	.00	40.00	1.23	2.18	.00	.47	3.88		20.00	63.88
WASTE SHAFT	25.55	24.82	50.37						.00		50.37
WASTE TRANSFER	2.66	2.41	5.07	4.39	7.19	2.41	.00	13.98			19.05
ROOM EXCAVATION	4.61	.06	4.68	92.11	4.74	.00	2.77	99.62	23.69		127.98
WASTE EMPLACEMENT EXCAVATION	.00	1.33	1.33	.90	.61	.00	.04	1.55			2.88
WASTE EMPLACEMENT	.00	.00	.00	.00	.00	.00		.00			.00
SUBTOTAL	72.83	28.61	101.44	98.63	14.72	2.41	3.28	119.04	23.69	20.00	264.16
OTHER ROCK EXCAV. & HANDLING											
MEN & MATERIALS SHAFTS	.00	.00	.00						.00		.00
ROCK HOISTING & R.H. SHAFTS	60.09	23.31	83.40	4.05	5.39	17.23		26.67	.00		110.07
SHAFT PILLAR EXCAVATION	67.16	.90	68.06						47.52		115.58
ALL CORRIDOR EXCAVATION	112.63	1.50	114.13	42.89	2.21	.00	1.29	46.39	11.03		171.55
SURFACE ROCK HANDL,STOR & DI	159.45	.00	159.45	4.51	144.26	.00		148.77			308.22
SUBTOTAL	399.34	25.71	425.05	51.45	151.86	17.23	1.29	221.82	58.55		705.42
VENTILATION											
SUPPLY SHAFT	18.43	4.61	23.04				19.97	19.97	.00		43.01
WASTE EMPL. EXHAUST SHAFT	13.76	.00	13.76						.00		13.76
MINING EXHAUST SHAFT	12.45	.00	12.45						.00		12.45
SUBTOTAL	44.65	4.61	49.26				19.97	19.97	.00		69.22
SUPPORT	37.45	.00	37.45	431.56	276.85	86.40	20.95	815.75		18.72	871.92
CONTINGENCY	171.93	24.53	196.46	117.64	142.76	21.21	9.10	290.71	16.45	22.52	526.14
TOTAL	1031.57	147.20	1178.77	705.86	856.56	127.24	54.58	1744.24	98.69	135.15	3156.84

B.328

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 23 CASH FLOW SUMMARY FOR A REPOSITORY INSALT
----- MILLIONS OF 0 DOLLARS

YEAR	CONSTRUCTION, STARTUP COSTS	OPERATING COSTS			CONTINGENCIES	TOTAL
		OPERATIONS MINING	WASTE OPERATIONS	BACKFILLING AND DECOMMISSIONING		
1990	19.26	.00	.00	.00	3.85	23.12
1991	63.70	.26	.00	.00	12.79	76.76
1992	75.13	.79	.00	.00	15.18	91.11
1993	81.06	.79	.00	.00	16.37	98.23
1994	150.58	27.08	.00	.00	35.53	213.16
1995	94.59	132.52	.00	.00	45.42	272.54
1996	41.22	134.77	.00	.00	35.20	211.18
1997	25.70	73.04	.00	.00	19.75	118.49
1998	.00	8.09	190.39	.00	39.70	238.18
1999	.00	8.53	36.53	.00	9.01	54.07
2000	.00	3.82	36.53	1.31	8.29	49.76
2001	.00	6.00	36.58	2.18	8.95	53.71
2002	.00	10.48	36.58	.98	9.61	57.65
2003	.00	8.62	41.67	1.49	10.36	62.14
2004	.00	8.62	41.67	1.49	10.36	62.14
2005	.00	8.62	41.68	1.49	10.36	62.14
2006	.00	8.62	41.67	1.49	10.36	62.14
2007	.00	8.62	41.67	1.49	10.36	62.14
2008	.00	8.62	41.67	1.49	10.36	62.14
2009	.00	8.62	41.68	1.49	10.36	62.14
2010	.00	8.62	41.67	1.49	10.36	62.14
2011	.00	8.62	41.67	1.49	10.36	62.14
2012	.00	8.62	41.67	1.49	10.36	62.14
2013	.00	8.62	44.08	1.49	10.84	65.03
2014	.00	8.62	41.67	1.49	10.36	62.14
2015	.00	8.62	41.67	1.49	10.36	62.14
2016	.00	8.62	41.67	1.49	10.36	62.14
2017	.00	8.62	41.68	1.49	10.36	62.14
2018	.00	8.62	41.67	1.49	10.36	62.14
2019	.00	8.62	41.67	1.49	10.36	62.14
2020	.00	8.62	41.67	1.49	10.36	62.14
2021	.00	3.72	41.55	1.49	9.35	56.10
2022	.00	3.44	40.65	1.49	9.11	54.69
2023	.00	1.14	32.25	.49	6.78	40.65
2024	.00	111.56		160.14	54.34	326.04

TOTAL	551.25	680.97	1203.62	194.86	526.14	3156.84

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11

TABLE 24 UNIT COST SUMMARY FOR A REPOSITORY IN SALT
0 DOLLARS PER KILOGRAM HEAVY METAL

FROM TABLE	COST CATEGORY	WASTE TYPE
		HLW-
13	CONSTRUCTION AND PREOPERATION	
	LAND	.19
	LICENSE	.00
	SITE PREPARATION	.61
	SHAFT	2.37
	SURFACE FACILITIES	3.22
	UNDERGROUND FACILITIES	.04
	ARCHITECT ENGINEER	1.45
	STARTUP	.00
	SUBTOTAL	7.87
14	MINING AND ROCK REMOVAL OPERATIONS	
	MINING	4.70
	HOISTING	.63
	SURFACE HANDLE AND STORE	4.08
	OFFSITE DISPOSAL	.33
	SUBTOTAL	9.73
19	WASTE OPERATIONS	
	RECEIVING	.05
	PACKAGING	4.87
	TRANSPORTATION	.23
	EMPLACEMENT EXCAVATION	.04
	EMPLACEMENT OPERATIONS	.00
	OTHER	12.00
	SUBTOTAL	17.19
20	BACKFILLING AND DECOMMISSIONING	
	BACKFILLING	1.17
	SEALING	.00
	DECOMMISSIONING	1.61
	SURVEILLANCE	.00
	SUBTOTAL	2.78
	CONTINGENCY	7.52
TOTAL		45.10

B.330

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE; 09/25/84
TIME; 16.18.11

TABLE 25 UNIT COST SUMMARY FOR A REPOSITORY IN SALT
0 DOLLARS PER CUBIC METER OF WASTE

FROM TABLE	COST CATEGORY	WASTE TYPE HLW-
13	CONSTRUCTION AND PREOPERATION	
	LAND	2.14
	LICENSE	.00
	SITE PREPARATION	6.84
	SHAFT	26.62
	SURFACE FACILITIES	36.17
	UNDERGROUND FACILITIES	.43
	ARCHITECT ENGINEER	16.32
	STARTUP	.00
	SUBTOTAL	88.52
14	MINING AND ROCK REMOVAL OPERATIONS	
	MINING	52.80
	HOISTING	7.06
	SURFACE HANDLE AND STORE	45.81
	OFFSITE DISPOSAL	3.68
	SUBTOTAL	109.35
19	WASTE OPERATIONS	
	RECEIVING	.55
	PACKAGING	54.71
	TRANSPORTATION	2.63
	EMPLACEMENT EXCAVATION	.46
	EMPLACEMENT OPERATIONS	.00
	OTHER	134.93
	SUBTOTAL	193.27
20	BACKFILLING AND DECOMMISSIONING	
	BACKFILLING	13.21
	SEALING	.00
	DECOMMISSIONING	18.08
	SURVEILLANCE	.00
	SUBTOTAL	31.29
	CONTINGENCY	84.49
TOTAL		506.92

B.331

CASE DESCRIPTION - SAMPLE CASE FOR DOCUMENTATION

DATE: 09/25/84
TIME: 16.18.11TABLE 26 UNIT COST SUMMARY FOR A REPOSITORY IN SALT
----- 0 DOLLARS PER PACKAGE

FROM TABLE	COST CATEGORY	WASTE TYPE ----- HLW-----
13	CONSTRUCTION AND PREOPERATION	
	LAND	1855.18
	LICENSE	.00
	SITE PREPARATION	5927.62
	SHAFT	23083.11
	SURFACE FACILITIES	31340.47
	UNDERGROUND FACILITIES	370.25
	ARCHITECT ENGINEER	14143.85
	STARTUP	.00
	SUBTOTAL	76700.45
14	MINING AND ROCK REMOVAL OPERATIONS	
	MINING	45751.68
	HOISTING	6113.93
	SURFACE HANDLE AND STORE	39694.48
	OFFSITE DISPOSAL	3190.69
	SUBTOTAL	94750.77
19	WASTE OPERATIONS	
	RECEIVING	475.11
	PACKAGING	47403.75
	TRANSPORTATION	2280.54
	EMPLACEMENT EXCAVATION	394.69
	EMPLACEMENT OPERATIONS	.00
	OTHER	116917.39
	SUBTOTAL	167471.49
20	BACKFILLING AND DECOMMISSIONING	
	BACKFILLING	11442.92
	SEALING	.00
	DECOMMISSIONING	15670.24
	SURVEILLANCE	.00
	SUBTOTAL	27113.16
	CONTINGENCY	73207.15
TOTAL		439243.03

B.332

COEMPLACEMENT ARRAY COWAST

PRIMARY WASTE TYPE I

I YR
 1990
 1991
 1992
 1993
 1994
 1995
 1996
 1997
 1998
 1999
 2000
 2001
 2002
 2003
 2004
 2005
 2006
 2007
 2008
 2009
 2010
 2011
 2012
 2013
 2014
 2015
 2016
 2017
 2018
 2019
 2020
 2021
 2022
 2023
 2024
 2025
 2026
 2027
 2028
 2029
 2030
 2031
 2032
 2033

B.333

DISTRIBUTION

No. of
Copies

No. of
Copies

OFFSITE

C. W. Conner
RW-33
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
Washington, D.C. 20585

J. J. Fiori
RW-22
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
Washington, D.C. 20585

M. W. Frei
RW-23
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
Washington, D.C. 20585

T. H. Issacs
RW-42
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
Washington, D.C. 20585

T. D. Nguyen
RW-33
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
Washington, D.C. 20585

S. P. Schneider
RW-22
Office of Civilian Radioactive
Waste Management
U.S. Department of Energy
Washington, D.C. 20585

D. G. Dippold
Office of Nuclear Waste Isolation
505 King Avenue
Columbus, OH 43201

Nancy Finley
Division 9413
Sandia Laboratory
Albuquerque, NM 87185

Richard Morissette
GA Technologies
P.O. Box 81608
San Diego, CA 92138

W. B. Kerr
Exxon Nuclear Idaho
P.O. Box 2800
Bldg. WCB S-1
Idaho Falls, ID 83401

Rudy Rozsa
Lawrence Livermore Labs
L-365
Livermore, CA 94550

Peter Baumgardner
Atomic Energy of Canada Ltd.
Whiteshell Nuclear Research
Establishment
Pinawa, Manitoba, Canada ROE1L0

No. of
Copies

Gary Simmons
Atomic Energy of Canada Ltd.
Whiteshell Nuclear Research
Establishment
Pinawa, Manitoba, Canada ROE1L0

20 Frank Haines
Roy F. Weston Associates
2301 Research Blvd. 3rd Floor
Rockville, MD 20850

30 DOE Technical Information Center

ONSITE

3 DOE Richland Operations Office

P. A. Craig
D. C. Langstaff
H. E. Ransom

34 Pacific Northwest Laboratory

M. K. Altenhofen
A. J. Boegel
J. B. Burnham
L. L. Clark (8)
M. O. Cloninger
B. M. Cole
D. A. Dingee
R. L. Engel
R. M. Fleischman
C. M. Heeb
G. M. Holter
J. H. Jarrett
A. T. Luksic
R. W. McKee
G. W. McNair
E. T. Merrill
P. W. Reimus
M. R. Shay
R. I. Smith
M. K. White
Technical Information (5)
Publishing Coordination (2)