AN INTEGRATED COMPUTER SIMULATOR FOR SURFACE MINE PLANNING AND DESIGN.

by

Amal Chakraborty

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE in

Mining Engineering

APPROVED:

__________________________
E. Topuz, Chairman

__________________________  ________________________
C. Haycocks                  G. J. Faulkner

__________________________
J. Richard Lucas, Department Head

30th August, 1985
Blacksburg, Virginia
ACKNOWLEDGEMENTS

The author would like to thank his committee chairman, Dr. E. Topuz, for his valuable guidance during the course of this study. In particular, his patience and constant encouragement during the tedious stage of programming work is deeply appreciated. The author would also like to acknowledge the assistance and cooperation of Dr. G. Faulkner and Dr. C. Haycocks in the preparation of this thesis.

The author wishes to thank his colleagues in the department, also Mr. L. Daniels and Mr. Carl Zipper of the Agronomy Department for many stimulating discussions on this subject. A warm and sincere thanks is also extended to the secretarial staff of the Mining Engineering Department.

A special tribute is paid to the author's mother, Mrs. A. P. Chakraborty, for her continuous encouragement and good wishes throughout this study period. The author dedicates this thesis to her.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td></td>
<td>ii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td></td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td></td>
<td>vi</td>
</tr>
<tr>
<td>I.</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II.</td>
<td>MODEL FORMULATION AND PROGRAM DESCRIPTION</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>Conceptual Model</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>Subroutines of the Program</td>
<td>13</td>
</tr>
<tr>
<td>2.2.1.</td>
<td>Subroutine HAUL</td>
<td>13</td>
</tr>
<tr>
<td>2.2.2.</td>
<td>Function AVBRP</td>
<td>23</td>
</tr>
<tr>
<td>2.2.3.</td>
<td>Subroutine MHDL</td>
<td>24</td>
</tr>
<tr>
<td>2.2.4.</td>
<td>Subroutine TRUCK</td>
<td>25</td>
</tr>
<tr>
<td>2.2.5.</td>
<td>Function RAND</td>
<td>29</td>
</tr>
<tr>
<td>2.2.6.</td>
<td>Function SAMPLE</td>
<td>29</td>
</tr>
<tr>
<td>2.2.7.</td>
<td>Subroutine DRILL</td>
<td>31</td>
</tr>
<tr>
<td>2.2.8.</td>
<td>Subroutine COST</td>
<td>32</td>
</tr>
<tr>
<td>2.3</td>
<td>Assumption and Options of the Program</td>
<td>36</td>
</tr>
<tr>
<td>2.3.1.</td>
<td>Assumptions</td>
<td>36</td>
</tr>
<tr>
<td>2.3.2.</td>
<td>Options</td>
<td>36</td>
</tr>
<tr>
<td>III.</td>
<td>PREPARATION OF DATA AND RUNNING THE PROGRAM</td>
<td>37</td>
</tr>
<tr>
<td>3.1.</td>
<td>Data Acquisition and Analysis</td>
<td>38</td>
</tr>
<tr>
<td>3.1.1.</td>
<td>Distribution Identification</td>
<td>38</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Figure 1:</td>
<td>Contour Plan of a Complete Mountaintop Removal</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2:</td>
<td>Cross-Ridge Method for complete Mountaintop Removal</td>
<td>6</td>
</tr>
<tr>
<td>Figure 3:</td>
<td>Subdivisions of a Mine into Blocks</td>
<td>8</td>
</tr>
<tr>
<td>Figure 4:</td>
<td>Subdivisions of a Mine into Lifts</td>
<td>9</td>
</tr>
<tr>
<td>Figure 5:</td>
<td>Division of Post-Mining Topography into Fill Cells</td>
<td>10</td>
</tr>
<tr>
<td>Figure 6:</td>
<td>Flow-chart of the Program</td>
<td>14</td>
</tr>
<tr>
<td>Figure 7:</td>
<td>Speed-Rimpull-Gradeability of 773B (Caterpillar)</td>
<td>21</td>
</tr>
<tr>
<td>Figure 8:</td>
<td>Flow-chart of Subroutine MHDL</td>
<td>27</td>
</tr>
<tr>
<td>Figure 9:</td>
<td>Amos Ridge Mining Site</td>
<td>75</td>
</tr>
<tr>
<td>Figure 10:</td>
<td>Overburden Sequence of Amos Ridge Mine</td>
<td>76</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>I</td>
<td>Name and Purpose of the Subprograms</td>
<td>19</td>
</tr>
<tr>
<td>II</td>
<td>Input Data Type A</td>
<td>42</td>
</tr>
<tr>
<td>III</td>
<td>Input Data Type B</td>
<td>43</td>
</tr>
<tr>
<td>IV</td>
<td>Input Data Type C</td>
<td>44</td>
</tr>
<tr>
<td>V</td>
<td>Input Data Type D</td>
<td>45</td>
</tr>
<tr>
<td>VI</td>
<td>Input Data Type E</td>
<td>47</td>
</tr>
<tr>
<td>VII</td>
<td>Input Data Type F</td>
<td>48</td>
</tr>
<tr>
<td>VIII</td>
<td>Input Data Type G</td>
<td>49</td>
</tr>
<tr>
<td>IX</td>
<td>Input Data Type H</td>
<td>51</td>
</tr>
<tr>
<td>X</td>
<td>Input Data Type I</td>
<td>53</td>
</tr>
<tr>
<td>XI</td>
<td>Input Data Type J</td>
<td>55</td>
</tr>
<tr>
<td>XII</td>
<td>Input Data Type K</td>
<td>56</td>
</tr>
<tr>
<td>XIII</td>
<td>Input Data Type L</td>
<td>57</td>
</tr>
<tr>
<td>XIV</td>
<td>Input Data Type M</td>
<td>58</td>
</tr>
<tr>
<td>XV</td>
<td>Input Data Type N</td>
<td>59</td>
</tr>
<tr>
<td>XVI</td>
<td>Input Data Type O</td>
<td>60</td>
</tr>
<tr>
<td>XVII</td>
<td>Blocks and Lifts Dimension</td>
<td>71</td>
</tr>
<tr>
<td>XVIII</td>
<td>Coal Production in Amos Ridge</td>
<td>72</td>
</tr>
<tr>
<td>XIX</td>
<td>Fraction of Lift Removed by Each Operation</td>
<td>74</td>
</tr>
<tr>
<td>XX</td>
<td>Inventory of Major Equipment</td>
<td>73</td>
</tr>
<tr>
<td>XXI</td>
<td>Input Data File</td>
<td>78</td>
</tr>
<tr>
<td>XXII</td>
<td>Salient Outputs of the First Run</td>
<td>81</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

The Appalachian Region of the United States is an important producer of coal, both by surface and underground mining methods. For surface coal mining in particular, the Appalachian share of the total U.S. market has been increasing steadily in recent years. Clearly, since much of the coal that is surface-mined in Central Appalachia is low-sulphur, steam coal, this area will play an important role in the regional program to increase the production and use of domestic coal while maintaining environmental quality standards. Unfortunately, there are a few problems, some of which primarily affect the Central Appalachian surface coal mining industry, and others which affect the surface coal mining industry nationwide. A major result of the latter type is the alarming decline in surface coal mine productivity since the late 1960's. The causes of this productivity loss, although still a subject of debate, are felt to be a declining reserve and increasingly stringent reclamation laws that affect current mining practices. Conventional contour mining, while environmentally undesirable, has many advantages from the mine operator's point of view. Among these are low operating costs and relatively small capital requirements. Complete mountaintop
removal mining theoretically has many advantages over contour mining but, in order to be profitable, it requires extensive planning of materials handling. The computer program described in this thesis was developed as a tool to analyze alternative materials handling plans.

The approach taken in developing the simulator had two parts. First, a literature survey was conducted to recognize the predominant physical conditions of a mountaintop type of surface mining, to identify important operational variables, and to develop a generalized model. Second, a computer program was written to simulate mountaintop removal mining operations.

The computer memory required for the program is about 320K. The maximum number of blocks and lifts that can be simulated by the program is ten with five lifts in each block. Such limitations have been imposed so that the program can be run in most micro-computers. However, if the user wants to run the program in a mainframe computer or in a micro-computer with a large memory he could use the program to simulate a bigger mine simply by changing the dimension statements.

The program can be used to model mountaintop removal mining with a reasonable degree of accuracy. The accuracy of
the result will, of course, depend on the accuracy of the input data. For validation of the model, the program was run with field data collected from a mine. The output of the program and the actual results differed by around 15%. This discrepancy could probably be attributed to the high degree of approximation that had to be made about cost data and haul time. A conservative approach was taken in approximating cost data and thus resulted in a higher cost per ton of coal mined.
II. MODEL FORMULATION AND PROGRAM DESCRIPTION.

During the late 1960's the first large scale demonstration of mountaintop removal surface mining was successfully accomplished in West Virginia. Typically, mountaintop removal methods could be considered as a logical extension of contour mining. Two types of mountaintop removal method are prevalent. In the first method, the first few cuts are taken along the contour. This method is called the "Contour Plan Mountaintop Removal Method." The other method is called the "Cross Ridge Mountaintop Removal Method." In that method the cuts are taken across the ridge of the mountain. Figures 1 and 2 show these methods. Both methods have some advantages and disadvantages. The main objective of this study is the development of a computer simulation model that may be used to minimize the cost of a mining operation in mountaintop removal type of surface mining. In this section some details are presented about the formulation of the simulation model.

2.1. Conceptual Model

To model a mountaintop removal mining operation the pre-mining topography was divided into blocks, and each block was divided into lifts.
Figure 1. Contour Plan for Complete Mountaintop Removal
Figure 2. Crossridge Plan for Complete Mountaintop Removal
The coal is exposed by removing blocks of overburden. Each block is removed as a vertical sequence of lifts; therefore, the unit for overburden removal operations is the lift. The subdivision of a mine into blocks and lifts is shown in Figures 3 and 4. The post mining topography was segmented into a number of fill cells, each with a specified volume, to be filled with overburden. Figure 5 shows the division of post-mining topography into fill cells. The destination of overburden removed from each lift could be one of the four fill cells:

1. Fixed dump sites
2. Extra spoil disposal area
3. Backfilling areas
4. Temporary spoil storage area.

In most operations the majority of the spoil will be backfilled. To accommodate additional spoil the number of other fill cells available should be specified. The fill cells are to be filled-in according to a sequence specified by the user. The simulator performs calculations using volumetric capacities of fill cells and the volume of spoil being produced through removing overburden from the various lifts as it moves through sequence. Filling a fill cell to capacity results in the next increment of overburden being placed in the next fill cell in the sequence.
Figure 3. Subdivision of a Mine into Blocks
Figure 4. Subdivision of a Mine into Lifts
Figure 5: Division of Post-Mining Topography into Fill Cells
The term fixed dump sites refers to reclamation areas, or any other areas which are to receive spoil, but where the mine planner does not wish to consider them as a part of the back filling sequence. The user has to specify a fixed dump site for any lift which he desires to treat specially.

The total volume of the spoil minus the volume of spoil to be filled in backfilling areas and fixed dump sites will be dumped into extra spoil disposal area. If no extra spoil disposal area is specified the program would assume the volume to be zero, and will issue a warning message if the total volume of spoil minus the volume of spoil to be filled in backfilling areas and fixed dump sites is not zero. The program will check each time the spoil has to be hauled to an extra spoil disposal area if the maximum amount has already been dumped. If the volumetric capacity of extra spoil disposal areas has already been filled, then the overburden will be hauled to the extra spoil storage area. This overburden will be dumped back when fill cells are available. The amount of overburden that has to be taken to extra spoil storage areas constitutes "the amount of overburden rehandled." Decisions about where to haul the portions of each lift will be handled by the following logic: 1) First, the program will check if a fixed dump
site has been specified for the lift. 2) If not it will check whether it can be dumped into backfilling areas. 3) If the distances between backfilling areas and current working places are less than the minimum value specified, it will try to dump in the extra spoil disposal area, if possible. Otherwise it will dump into an extra spoil storage area.

Unit operations on any lift for overburden removal are the following:

1. drilling and shooting,
2. dozing,
3. loading and carrying
4. loading and hauling.

User has to specify the percentage of overburden that has to be removed by each operation. If certain operations will not be done in a particular lift the input will be zero.

When the overburden removal of all lifts in a block is completed, the underlying coal may be removed. In the program it is assumed that the coal will be hauled by contract haulers. However, some information regarding haulers is necessary to simulate coal removal operations. Construction of sediment ponds, grading and seeding of
reclaimed areas, etc., are not simulated in the model. The costs of these operations are to be specified by the users. A flow chart of the program is shown in Figure 6.

The program reads input data and then initializes all variables. Next the program goes through a loop which checks if any work is to be done and calls subroutines accordingly. Then the program finds the operations that will be completed next, and it updates the simulation clock and other attributes. This process continues until work on all blocks is over. Once all blocks are mined the program computes and prints a summary of operations. Then the control goes to subroutine COST where the cost of mining is calculated and printed. If the number of simulation runs is more than one, the control comes back to initialize variables for the second run and continues.

2.2. Subroutines of the program:

The subroutines and functions used in the program are given in Table I. The more important ones are discussed below.

2.2.1. Subroutine HAUL

It is known that if rate of acceleration is constant the rectilinear motion of the vehicle can be expressed by the following set of relationships:

\[ v_2 = v_1 + a \cdot t \]
Figure 6: Flow-chart of the Program
Figure 6: continued
Figure 6: continued
Figure 6: continued
Figure 6: continued
### TABLE I: NAME AND PURPOSE OF THE SUBPROGRAMS

<table>
<thead>
<tr>
<th>NAME</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMULT</td>
<td>Simulates mining activities.</td>
</tr>
<tr>
<td>BULDOZ</td>
<td>Calculates the time required to doze a lift. Also accumulates total time dozer has worked.</td>
</tr>
<tr>
<td>Haul</td>
<td>Calculates hauling time for haulers</td>
</tr>
<tr>
<td>AVBRP</td>
<td>Calculates the available rimpull given the speed.</td>
</tr>
<tr>
<td>MHDL</td>
<td>This subroutine is called when overburden is to be loaded or hauled. The subroutine finds out the destination of overburden and then calls subroutine TRUCK.</td>
</tr>
<tr>
<td>TRUCK</td>
<td>This calculates time required to load and haul overburden or coal. Also accumulates total time trucks have worked.</td>
</tr>
<tr>
<td>RAND</td>
<td>Generates a random value from UNIFORM (0,1) distribution.</td>
</tr>
<tr>
<td>SAMPLE</td>
<td>Generates a random variate of a given distribution.</td>
</tr>
<tr>
<td>DRILL</td>
<td>Calculate time to drill a lift.</td>
</tr>
<tr>
<td>COST</td>
<td>Calculates the cost of mining.</td>
</tr>
</tbody>
</table>
\[ s_2 = s_1 + v_1 \cdot t + \frac{1}{2} a \cdot t^2 \]
\[ v_2^2 = v_1^2 + 2 \cdot a \cdot s_2 - 2 \cdot a \cdot s_1 \]

where \( v_1 \) and \( s_1 \) are velocity and distance at time \( t_1 \), and \( v_2 \) and \( s_2 \) are velocity and distance at time \( t_2 \).

\( t = t_2 - t_1 \) and \( a \) is acceleration.

Figure 7 shows a typical truck performance curve as supplied by a manufacturer of off-highway trucks. It is evident that rimpull and, consequently, acceleration, vary greatly with vehicle speed so that the constant acceleration equations seem inapplicable. To circumvent this impasse, consider vehicle motion during any short period of time. If the selected period is short enough, acceleration during the period is approximately constant, and the constant acceleration formula is applicable, (O'Neil, 1966).

Three sets of data have to be supplied for the program to simulate vehicular motion. They are:

1. speed and rimpull in each gear,
2. haul road characteristics, and
3. travel constraints.

Haul road characteristics input consists of:

1. rolling resistance for the whole mine
2. number of segments of each haul road, and
3. grade and length of each section.
Figure 7: Speed-Rimpull-Gradeability of 7738 (Caterpillar)
Every mine has its own peculiar operating conditions and practices that distinguish it from other operations. The following constraints must be defined for the model if the user wants to calculate haulage time:

1. maximum velocity allowed,
2. deceleration rate,
3. maximum acceleration allowed, and
4. maximum down hill speeds.

Though the program is capable of calculating haul time for trucks, obtaining data regarding haul road characteristics at the design stage could be a difficult task for the user. Thus, an option has been incorporated into the program to allow the user to provide approximate haul time for haul roads. Subroutine HAUL calculates haul time by using the following equations:

\[
\begin{align*}
DT &= (S_2 - S_1) / A \\
S_2 &= (S_1^2 + 2.A .HL / 20 )^{1/2} \\
A &= AF/( WMAT . 2000 . 32.2 ) \\
AF &= (AVRP . 1000 /453.6) - RES \\
AVRP &= F( S_1, SPEED , RP ,K ) \\
RES &= ( RORI + GRADE ) . 20 . WMAT
\end{align*}
\]

where:

\[
\begin{align*}
DT &= \text{time taken to travel the road section} \\
S_2 &= \text{final speed}
\end{align*}
\]
S₁ = initial speed
A = acceleration
HL = length of the road section
AF = acceleration force
WMAT = weight of the truck including load
AVRP = available rimpull
RES = resistance to motion
F = mapping function of S₁, SPEED, RP into AVBRP. This function is discussed later.

SPEED = speed at Kth gear
RP = rimpull available at Kth gear
RORI = rolling resistance to mine
GRADE = grade of the road section

2.2.2 Function AVB RP

This function calculates the available rimpull for truck at a given speed. Inputs are speed-rimpull array and the speed at which available rimpull is to be calculated.

S₁ = speed
NGEAR = number of gear
SPEED(K) = speed at Kth gear
RP(K) = rimpull at Kth gear

Program finds K such that:
SPEED(K) ≥ S₁
If SPEED(K) = S₁ and K > 1 ≤ N
then AVBRP = RP(K)

If SPEED(K) = S1 and K = 1 then

\[
AVBRP = \frac{RP(K) \cdot S1}{SPEED(K)}
\]

otherwise

\[
AVBRP = \frac{RP(K) + (RP(K) + RP(K-1)) \cdot (S1 - SPEED(K-1))}{SPEED(K) - SPEED(K-1)}
\]

2.2.3. Subroutine MHDL

This subroutine is called whenever overburden has to be removed from any lift by truck and loader. This routine finds out the destination of the overburden and calls subroutine TRUCK with appropriate codes. In order to do so, the subroutine keeps status of all back filling areas, extra spoil disposal areas, and extra spoil storage areas. If the spoil in a lift has been specified as one to be treated specially, that is, it has to go to a specified fill cell, then this subroutine calls subroutine TRUCK with a code to indicate destination. Otherwise, it checks if any backfill areas are available for dumping the overburden. If not, the routine looks for a extra spoil disposal area. If spoil disposal areas are not available or the volume of material that has already been dumped in extra spoil disposal areas
is equal to the maximum amount of material that can be dumped there, then the subroutine calls TRUCK with codes to indicate that the lift will go to "extra spoil storage area." If no storage area was specified then an error will occur. A flow chart of the subroutine is shown in Figure 8.

2.2.4. Subroutine TRUCK

Subroutine TRUCK can be called either from subroutine SIMULT or MHDL. If the material to be moved is coal then SIMULT calls this subroutine, otherwise, MHDL calls it. TRUCK itself calls function SAMPLE and subroutine HAUL.

The TRUCK subroutine calculates the time required to remove overburden from a lift. The calculations of cycle time and time required to move overburden are shown below.

\[ SCYCLE = RHT + THT + TDT + TLT + WTR \]

if removing overburden. Otherwise:

\[ SCYCLE = HCTR + TLT \]

\[
TRE = \frac{WWT}{NUMB \cdot SCYCLE \cdot TOTR}
\]

\[
TREC = \frac{CWT}{NUMC \cdot SCYCLE \cdot TCTR}
\]

\[ TDT = SAM + DMP \]

\[ TLT = SAML + TEX \]
where:

\[ \text{SCYCLE} = \text{cycle time (min)} \]
\[ \text{RHT} = \text{return haul time (min)} \]
\[ \text{THT} = \text{forward haul time (min)} \]
\[ \text{TDT} = \text{total dumping time (min)} \]
\[ \text{TLT} = \text{total loading time (min)} \]
\[ \text{WTR} = \text{waiting time for truck at loading point (min)} \]
\[ \text{HCTR} = \text{total haul time (min)} \]
\[ \text{TRE} = \text{time required if material is overburden (min)} \]
\[ \text{WWT} = \text{amount of overburden (cu.yd.)} \]
\[ \text{NUMB} = \text{number of truck} \]
\[ \text{TOTR} = \text{truck capacity (cu.yd.)} \]
\[ \text{TREC} = \text{time required if material is coal (min)} \]
\[ \text{CWT} = \text{weight of coal to be removed (tons)} \]
\[ \text{NUMC} = \text{number of coal truck} \]
\[ \text{TCTR} = \text{truck capacity (cu.yd.)} \]
\[ \text{SAM} = \text{time to dump. Generated by SAMPLE from distribution (min)} \]
\[ \text{DMP} = \text{maneuvering time at dumping point (min)} \]
\[ \text{SAML} = \text{loading time. Generated by function SAMPLE (min)} \]
\[ \text{TEX} = \text{exchange time at loading point (min)} \]
Figure 8: Flow-chart of Subroutine MHOL
Figure 8: continued
2.2.5. Function RAND

There are several methods available to generate random numbers which are independent and identically distributed over UNIFORM \{0,1\}. The method used in this program is "congruential generators." The method is described below.

Let \{z_i\} denote a sequence of integers defined by the recursive formula:

\[ z_i \equiv (az_{i-1})(\text{mod} \ m) \]

where the multiplier a, and the modulus m are integers. The convergence sign 'Ξ' means that we determine \( z_i \) algebraically as:

\[ z_i = az_{i-1} - mk_i \]

where \( k_i = (az_{i-1})/m \) denotes the largest positive integer of \( (az_{i-1})/m \). Then the desired random numbers on \[0,1] are obtained by \( v_i = z_i /m \).

The values used for a and m in this model are 16807 and 2147483647 respectively.

2.2.6. Function SAMPLE

A random variate is a particular outcome or sample value of a random variable. Now, let X be the random variable of our concern and F(X) be the cumulative distribution function. By definition,

\[ F(X) = P(x \leq X). \]
Let U = F(X) be a random variable. By definition, the cumulative distribution function of U is given as:

\[ G(u) = P(U \leq u) \text{.} \]

Substituting we get,

\[ G(u) = P(F(X) \leq u) = P(X \leq F^{-1}(u)) \]

\[ = F(F^{-1}(u)) = u \]

Thus \( G(u) = u \). Differentiating with respect to \( u \), we obtain the probability density function of \( u \) as

\[
G(u) \begin{cases} 
1 & \text{if } 0 < u < 1 \\
0 & \text{otherwise}
\end{cases}
\]

which is the probability density function of a uniform distribution. Therefore, we conclude that \( U \sim \text{UNIFORM } (0,1) \).

So, a random variate generation can be summarized in the four steps:

1. Generate a random number \( r \sim U(0,1) \).
2. Set \( r = F(X) = P(x \leq X) \).
3. Solve for \( X \).
4. Deliver \( X \) as the random variate from the desired distribution.

The generation of an exponentially distributed random variable is shown as an example. The probability density function, \( f(x) \), and the cumulative distribution function, \( F(x) \), for an exponential distribution may be written:

\[
f(x) = m \cdot e^{-mx}, \quad x > 0 \quad \text{mean } = 1/m; \quad \text{variance } = 1/m^2
\]

\[
F(x) = 1 - e^{-mx}, \quad x \geq 0.
\]
Then,

\[ U = F(x) = 1 - e^{-mx} \]

\[ e^{-mx} = 1 - U \Rightarrow \ln(e^{-mx}) = \ln(1-U) \]

\[-mx = \ln(1-U) = \ln(u') \text{ where } U' \sim U(0,1)\]

\[ x = -\frac{1}{m} \ln U'. \]

2.2.7. Subroutine DRILL

This subroutine calculates the time required to drill a lift, and also total feet drilled. Input to this subroutine includes lift dimensions, spacing, burden, fraction to be blasted, and number of drilling machines to be used.

Calculation:

\[ \begin{align*}
NH &= \frac{RL \cdot WD}{DS \cdot DB} \\
DL &= AD \cdot FB \\
TF &= NH \cdot DL \\
TR &= TF / DR 
\end{align*} \]

where:

\begin{align*}
NH &= \text{number of holes to be drilled} \\
RL &= \text{length of the lift} \\
WD &= \text{width of the lift} \\
DS &= \text{spacing} \\
DB &= \text{drilling burden} \\
DL &= \text{length of each hole} \\
AD &= \text{depth of the lift}
\end{align*}
FB = fraction to be blasted
TF = total feet drilled
TR = time required to drill all holes
DR = drilling rate.

2.2.8. Subroutine COST

Owning and operating costs for a given model of machine can vary widely because of many influencing factors. Users must be able to estimate, with a reasonable degree of accuracy, what a machine will cost to own and operate per hour. In this model, owning costs are considered as fixed costs, whereas operating costs are variable. So, if a machine works $T_1$ units of time out of a possible $T_2$ time units, and $C_2$ and $C_1$ are owning and operating costs respectively, then the total cost incurred is $C = C_1 T_1 + C_2 T_2$.

Before describing the calculation involved in finding cost, some of the cost items are described below.

Interest, Insurance, and Taxes (IIT) : Many owners charge this cost as part of hourly owning and operating costs while others prefer to consider it as general overhead in their overall operation. Operating manuals of the equipment provide example calculation of this cost item (Caterpillar, 1979).
Undercarriage : Undercarriage expense can be a major portion of the operating costs for truck type machines (Caterpillar, 1979). The primary conditions that affect probable life expectancy of truck type undercarriage are:

1. impact factor
2. abrasiveness factor
3. " Z " factor.

For estimating the undercarriage cost the following procedure may be used:

Step 1 - Select machine and its corresponding basic factor.

Step 2 - Determine impact, abrasiveness, and "Z" factor

Step 3 - \[ C = (I+A+Z) \cdot (\text{Basic Factor}) \]

where \( C \) = cost / hour.

Repair Cost: Repair costs are normally the largest operating costs and include all parts and direct labor (except operators' wages) chargeable to the machines. Shop overhead can be absorbed in the general overhead or charged to machines as a percentage of direct labor costs, whichever is the owner's normal practice, (Caterpillar, 1979).

Repair costs are low initially and increase gradually. In the program they are assumed constant, so average repair costs should be provided as an input.
Values of two factors are needed in the program. First is the "Basic Factor." This factor is based on average costs for the first 10,000 hours of service. If the machines will be used for more than 10,000 hours another factor named "extended life multipliers" is needed. To estimate hourly repair costs, enter basic factor and extended life multipliers. If the machine will be used less than 10,000 hours, then the extended life multipliers will be 1.

The cost per hour for repair

\[ C = BF \cdot EF \]

where BF is the basic factor and EF is extended life multipliers.

Special items: Ground engaging tools, welding, etc. are covered here. This cost ($/hour) should be estimated and entered as input.

Calculation of total cost:

\[ TCOST = DPH + CIIT + CF + CRP + SPL + ALT \]
\[ DPH = \frac{VD}{DPR} \]
\[ VD = DP - TR - RV \]
\[ CIIT = \frac{(TF \cdot DP)}{1000} \]
\[ CF = FUP \cdot FCS \]
\[ CRP = EF \cdot BRF \]
\[ CTIRE = \frac{TR}{ELH} \]
\[ CUDG = (I + A + Z) \cdot BF \]
where

TCOST = total cost

DPH = depreciation per hour

CIIT = cost of insurance, interest, and tax

CF = fuel cost

CRP = cost of repair

SPL = special cost

ALT = CTIRE if tire mounted

ALT = CUDG if crawler mounted

VD = value of depreciation

DPR = depreciation period

DP = delivered price

TR = tire replacement cost

RV = resale value

TF = IIT factor

FUP = unit price for fuel

FCS = fuel consumption per hour

EF = extended use multiplier

BRF = basic repair factor

CTIRE = tire cost

ELH = expected life of tire in hours

CUDG = undercarriage cost

BF = basic factor

I, A, Z = impact, abrasion and Z factor respectively.
2.3. Assumptions and Options in the Program

2.3.1. Assumptions

The following assumptions and simplifications have been made in the program:

- Seeding, sediment ditch construction, etc., have not been considered.
- Production loss due to inclement weather has not been considered.
- Equipment failure has not been considered.
- Dragline or scraper usage is not permitted.
  They can be added as separate subroutines with minor modification of main program.
- Coal will be hauled by contract hauler.

2.3.2. Options

1) Single or multiple seams can be simulated in the program.
2) User can input either haul time or haul road data. In the later case haul time will be calculated in the program.
3) Deterministic, stochastic or hybrid simulation.
4) User can input either equipment cost data or average owning and operating cost. In the former case cost data will be used to calculate owning and operating cost.
4) Output options. Options available are printing and listing file.
III. PREPARATION OF DATA AND RUNNING THE PROGRAM

The user has to prepare a plan for mining the deposit. This plan should include the equipment to be used, block and lift dimensions, and a material handling plan. Once the mine has been divided into a number of blocks and each block into a number of lifts, the surface area and average depth of all the blocks need to be computed. The post mining topography is to be divided into a number of fill cells. Haul road characteristics or approximate haul times are to be estimated from each block to all possible dumping locations. Once this has been done the user can obtain equipment characteristics and cost features from the equipment performance handbook or can use data from some other sources such as current publications or his own experience in other, similar operations. Once all data have been obtained, the user has to create a data file by running the interactive data program described in Section 3.2. Description of the procedure is given in section 3.2. Care should be exercised in using decimal points. A decimal point in any integer variable will cause an error which is sometimes difficult (time consuming) to trouble-shoot in micro-computers. One way to avoid such a possibility is not to use a decimal point unless it is absolutely necessary.
3.1. **Data Acquisition and Analysis**

Both probabilistic and deterministic techniques were adopted to model the system. By formulating a hybrid model, the best features of each technique are combined in the simulator. Probabilistic simulation is used where system behavior is random, and deterministic simulation is used where it does not contain significant random elements.

3.1.1 **Distribution Identification:**

A sampled observation is a chance occurrence of a random variable that ranges over either a discrete or continuous spectrum of values. Sample measurements are vital only as representatives of the parent population. If a simulation model is to reflect the performance of a system in a realistic manner, the input data to the model must reflect the operating characteristics of the system.

To realistically capture the characteristics of a particular random variable in the simulation model, the associated process generator should possess the property that the distribution of the random variable produced is similar to that observed in the real world system. Identification of the distribution of a random variable can be done in the following steps:
1. Identification of the shape of the distribution;

2. Identification of one or more well-known distributions having a shape similar to that reflected by observed data;

3. Estimation of parameters of the hypothesized distributions;

4. Testing the fit of the observed data to the hypothesized distributions.

Detailed description and methodology of any of the steps described above can be found in any standard statistics text book.

If none of the candidate distributions can be accepted as describing the random variable under study, an empirical distribution has to be formed based on observed data. In that case, the cumulative frequency histogram for each input variable must be defined. If, however, the data can be described by a standard probability density function, only characteristic parameters of the distribution are needed by the program. A random element can then be generated by sampling a value from UNIFORM (0,1) distribution and equating that with cumulative distribution function of the input variable under consideration. This not only frees a sizable block of computer memory, but also lessens the
number of calculations to be performed which improves the model's efficiency and economy.

In the model, the following distributions can be used:

(1) uniform,
(2) empirical,
(3) normal,
(4) exponential,
(5) triangular,
(6) weibull.

If the deterministic simulation is desired then the average values of the variables are used instead of the distributions. If some other distributions need to be used, a little modification of the function SAMPLE is required.

3.2. Data Input Procedure

A data file can be created or modified by running the DATA program. The program will ask for values in the terminal. Tables II to XV will appear on the screen when program DATA is executed. The tables include a brief description of these variables. The formats are given so that the user may enter his data easily on the screen. These data are then read by unformatted read statements, therefore, it may be easier to leave a blank space between data values instead of following the formats displayed on
the screen. The codes for the different distributions are given below:

- average : 0
- uniform : 1
- empirical : 2
- normal : 3
- exponential : 4
- triangular : 5
- weibull : 6.

3.3. Running the Program

3.3.1. System Requirement

The microcomputer version of the simulator is written in FORTRAN 77 and runs on the IBM-PC, and PCXT. Minimum system configuration is 320 KB RAM, monitor, and a printer (color or monochrome).

3.2.2 Preparation of the program diskette

A copy of the program diskette should be made as a back-up copy. If working on an IBM-PC the copying can be achieved by using DISKCOPY or COPY commands.
TABLE II: INPUT DATA TYPE A

<table>
<thead>
<tr>
<th>&lt;VARIABLE&gt;</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ISEED&gt;</td>
<td>Initial seed value (any prime number, enter 0 for default value 87).</td>
</tr>
<tr>
<td>&lt;ISTOP&gt;</td>
<td>Code to indicate if antithetic sampling should be used (0 if no, 1 if yes).</td>
</tr>
<tr>
<td>&lt;NBLOCK&gt;</td>
<td>Number of blocks.</td>
</tr>
<tr>
<td>&lt;NFILCL&gt;</td>
<td>Number of backfill cells.</td>
</tr>
<tr>
<td>&lt;NHOLLW&gt;</td>
<td>Number of extra spoil disposal areas.</td>
</tr>
<tr>
<td>&lt;NEX&gt;</td>
<td>Number of extra spoil storage areas.</td>
</tr>
<tr>
<td>&lt;CTIM&gt;</td>
<td>Shift time (min).</td>
</tr>
<tr>
<td>&lt;CDEN&gt;</td>
<td>Density of coal (tons/cu.ft.).</td>
</tr>
<tr>
<td>&lt;NSDY&gt;</td>
<td>Number of shifts per day.</td>
</tr>
<tr>
<td>&lt;NDMH&gt;</td>
<td>Number of days per month.</td>
</tr>
<tr>
<td>&lt;CREVR&gt;</td>
<td>Fraction of coal recoverable.</td>
</tr>
</tbody>
</table>

Enter data in the following format:
ISEED ISTOP NBLOCK NFILCL NHOLLW NEX CTIM CDEN NSDY NDMH CREVR

###   #   ###   ###   ###   #.####   #   ###   #.####

| <NROCK> | Number of types of rock. |
| <PFAC(I)> | Powder factor of Ith type of rock. |
| <DENS(I)> | Density of Ith type of rock (tons/cu.yd.) |
| <SWELF(I)> | Swelling factor of Ith type of rock. |
| <EFFECT(I)> | Factor for drilling rate of Ith type rock. |

Note: I = 1, NROCK
Enter data in the following format:
NROCK PFAC(I) DENS(I) SWELF(I) EFFECT(I)

###   #.###   #.###   #.###   #.###
### TABLE III: INPUT DATA TYPE B

#### INPUT DATA FOR DRILLS

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NPDR&gt;</td>
<td>Number of drills.</td>
</tr>
<tr>
<td>&lt;ISDR&gt;</td>
<td>Code for the probability distribution of drilling rate.</td>
</tr>
<tr>
<td>&lt;DB&gt;</td>
<td>Burden (ft.).</td>
</tr>
<tr>
<td>&lt;DS&gt;</td>
<td>Spacing (ft.).</td>
</tr>
<tr>
<td>&lt;NSPACE(1)&gt;</td>
<td>Number of data points if empirical distribution indicated; otherwise enter 0.</td>
</tr>
<tr>
<td>&lt;TMOVE&gt;</td>
<td>Average time for drill to move from one hole to another (min).</td>
</tr>
</tbody>
</table>

Enter data in the following format:

NPDR ISDR DB DS NSPACE(1) TMOVE
### # ###.# ###.# # ###.##

#### PROBABILITY DISTRIBUTION DATA FOR DRILLING RATE

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;PRM(J)&gt;</td>
<td>If ISDR = 2 then value of Jth point for drilling rate (ft/min). If ISDR is not 2 then Jth parameter of the distribution indicated.</td>
</tr>
<tr>
<td>&lt;HIST(J)&gt;</td>
<td>Corresponding cumulative probability. If ISDR is not 2 enter 0.</td>
</tr>
</tbody>
</table>

Note: J = 1, MNN. MNN = NSPACE(1) if ISDR = 2 otherwise MNN= Number of parameters of the distribution indicated by ISDR.

Enter data in the following format:

PRM(J) HIST(J)
####.## #.##
TABLE IV: INPUT DATA TYPE C

DATA FOR BULLDOZERS

<table>
<thead>
<tr>
<th>&lt;VARIABLE&gt;</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NPBL&gt;</td>
<td>Number of dozers.</td>
</tr>
<tr>
<td>&lt;TRATE1&gt;</td>
<td>Travelling rate for loaded dozer (ft/min).</td>
</tr>
<tr>
<td>&lt;TRATE2&gt;</td>
<td>Travelling rate for empty dozer (ft/min).</td>
</tr>
<tr>
<td>&lt;LTIME&gt;</td>
<td>Loading time (min).</td>
</tr>
<tr>
<td>&lt;DTIME&gt;</td>
<td>Dumping time (min).</td>
</tr>
<tr>
<td>&lt;CONST&gt;</td>
<td>Tram constant (min).</td>
</tr>
<tr>
<td>&lt;CAPB&gt;</td>
<td>Bucket capacity (cu.yd.).</td>
</tr>
</tbody>
</table>

Enter data in the following format:

NPBL TRATE1 TRATE2 LTIME DTIME CONST CAPB
    ## ###.# ###.# ##.## ##.## ##.## ##.## ##.##
### TABLE V: INPUT DATA TYPE D

**INPUT DATA FOR OVERBURDEN AND COAL TRUCK**

<table>
<thead>
<tr>
<th>&lt;VARIABLE&gt;</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;N1&gt;</td>
<td>Number of overburden trucks.</td>
</tr>
<tr>
<td>&lt;NC1&gt;</td>
<td>Number of coal trucks.</td>
</tr>
<tr>
<td>&lt;ISTR&gt;</td>
<td>Code for distribution of dumping times.</td>
</tr>
<tr>
<td>&lt;NSPACE(3)&gt;</td>
<td>Number of data points if ISTR = 2. Otherwise enter 0.</td>
</tr>
<tr>
<td>&lt;TEX&gt;</td>
<td>Maneuvering time for truck at loading point (minute).</td>
</tr>
<tr>
<td>&lt;DMP&gt;</td>
<td>Maneuvering time for truck at dumping point (minute).</td>
</tr>
<tr>
<td>&lt;HCTR&gt;</td>
<td>Total haul time for coal truck (minute)</td>
</tr>
<tr>
<td>&lt;GVW&gt;</td>
<td>Weight of empty overburden truck (tons).</td>
</tr>
<tr>
<td>&lt;TOTR&gt;</td>
<td>Capacity of overburden truck (cu.yd.).</td>
</tr>
<tr>
<td>&lt;GVC&gt;</td>
<td>Weight of empty coal truck (tons).</td>
</tr>
<tr>
<td>&lt;TCTR&gt;</td>
<td>Capacity of coal truck (cu.yd.).</td>
</tr>
<tr>
<td>&lt;NHAUL&gt;</td>
<td>Code to indicate if hauling time should be calculated by the program (yes=0, no=1).</td>
</tr>
</tbody>
</table>

Enter data in the following format:

N1 NC1 ISTR NSPACE(3) TEX DMP HCTR GVW TOTR GVC TCTR NHAUL
## ## # ### #.# #.# ##.## ##.# ##.## ##.# ##.## #

### PROBABILITY DISTRIBUTION DATA FOR DUMPING TIME

| <PTR(J)> | Value of jth point if empirical distribution indicated otherwise Jth parameter of the distribution indicated for dumping time. |
| <HITR(J)> | Corresponding cumulative probability if distribution is empirical. Otherwise enter 0. |

Note: J = 1, NS where NS = NSPACE(3) if distribution is empirical otherwise NS = Number of parameter of the distribution indicated.

Enter data in the following format:

PTR(J) HITR(J)
###.## #.#
### TABLE V: INPUT DATA TYPE D (continued)

#### INPUT FOR GENERAL TRUCK DATA

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NGEAR&gt;</td>
<td>Number of gears in the overburden truck.</td>
</tr>
<tr>
<td>&lt;DA&gt;</td>
<td>Deceleration when brake used (ft/second²).</td>
</tr>
<tr>
<td>&lt;RORI&gt;</td>
<td>Rolling resistance of the mine (%).</td>
</tr>
<tr>
<td>&lt;DGR&gt;</td>
<td>Maximum downhill grade (degrees).</td>
</tr>
<tr>
<td>&lt;AMAX&gt;</td>
<td>Maximum acceleration allowed (ft/second²).</td>
</tr>
<tr>
<td>&lt;VMAX&gt;</td>
<td>Maximum velocity allowed (m.p.h.).</td>
</tr>
</tbody>
</table>

Note: This table will not appear if NHAUL = 1.

Enter data in the following format:

```plaintext
NGEAR DA RORI DGR AMAX VMAX
## #.## #.## ## #.## #.## #.##
```

- **<DHILL(I)>**: Maximum downhill speed permissible at grade I.

Enter data in the following format:

```plaintext
DHILL(1) DHILL(2) DHILL(3) DHILL(4) DHILL(5) .... DHILL(DGR)
##.## ##.## ##.## ##.## ##.## ##.## ##.##
```

- **<SPEED(J)>**: Speed limit at Jth gear (m.p.h.).
- **<RP(J)>**: Corresponding rimpull (lbs).

Note: J= 1,NGEAR

Enter data in the following format:

```plaintext
SPEED(J) RP(J)
##.## #####.##
```

Note: This table will appear if haultime is to be calculated by the program.
TABLE VI: INPUT DATA TYPE E

<table>
<thead>
<tr>
<th>&lt;VARIABLE&gt;</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;N2&gt;</td>
<td>Number of loader (overburden).</td>
</tr>
<tr>
<td>&lt;NC2&gt;</td>
<td>Number of loader (coal).</td>
</tr>
<tr>
<td>&lt;IPSV&gt;</td>
<td>Code for probability distribution of loading time.</td>
</tr>
<tr>
<td>&lt;NSPACE(4)&gt;</td>
<td>Number of data points for loading cycle time</td>
</tr>
<tr>
<td></td>
<td>if IPSV indicated above = 2. Otherwise enter 0.</td>
</tr>
<tr>
<td>&lt;CAPL&gt;</td>
<td>Bucket capacity of overburden loader (cu.yd.).</td>
</tr>
<tr>
<td>&lt;CCPL&gt;</td>
<td>Bucket capacity of coal loader (cu.yd.).</td>
</tr>
<tr>
<td>&lt;AVSPD&gt;</td>
<td>Average speed for overburden loader (ft/min).</td>
</tr>
</tbody>
</table>

Enter data in the following format:

N2  NC2  IPSV  NSPACE(4)  CAPL  CCPL  AVSPD
##  ##  #  ####  ###.##  ##.##  ###.##

##.##  ##.##  ###.##  ##.##  ###.##  ##.##  ###.##
### TABLE VII: INPUT DATA TYPE F

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSV(J)</td>
<td>Value of Jth point if empirical distribution indicated otherwise Jth parameter of the distribution indicated for overburden loader (min).</td>
</tr>
<tr>
<td>HISV(J)</td>
<td>Corresponding cumulative probability if distribution is empirical. Otherwise enter 0.</td>
</tr>
<tr>
<td>CSV(J)</td>
<td>Value of Jth point for loading cycle time if IPSV = 2 otherwise Jth parameter of the distribution indicated for coal loader (min).</td>
</tr>
<tr>
<td>HCSV(J)</td>
<td>Corresponding cumulative probability if distribution empirical otherwise enter 0.</td>
</tr>
</tbody>
</table>

Note: J = 1,NS where NS = NSPACE(4) if distribution empirical otherwise NS = Number of parameter of the distribution.

Enter data in the following format:

PSV(J)  HISV(J)  CSV(J)  HCSV(J)
##.##    #.##     ##.##    #.##
TABLE VIII: INPUT DATA TYPE G

INPUT DATA ABOUT BLOCKS

<table>
<thead>
<tr>
<th>&lt;VARIABLE&gt;</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NLIFT(I)&gt;</td>
<td>Number of lifts of Ith block.</td>
</tr>
<tr>
<td>&lt;DAREA(I)&gt;</td>
<td>Area of Ith block for overburden (sq.ft.).</td>
</tr>
<tr>
<td>&lt;CAREA(I)&gt;</td>
<td>Area of coal of block I (sq.ft.).</td>
</tr>
<tr>
<td>&lt;CTIK(I)&gt;</td>
<td>Coal thickness of block I (ft.).</td>
</tr>
<tr>
<td>&lt;PSBLK(I)&gt;</td>
<td>Block number that has to be finished before work on Ith block can start.</td>
</tr>
<tr>
<td>&lt;PSLFT(I)&gt;</td>
<td>Lift number that has to be finished before work Ith block can start. (This lift is in the block described by PSBLK(I)).</td>
</tr>
<tr>
<td>&lt;BPREP(I)&gt;</td>
<td>Preparation time for block I (hr.).</td>
</tr>
</tbody>
</table>

Note: I=1,NBLOCK

Enter data in the following format:

NLIFT(I)  DAREA(I)  CAREA(I)  CTIK(I)  PSBLK(I)  PSLFT(I)  BPREP(I)
##  #######.#  #######.#  ##.##  ##  ##  #######.#
TABLE VIII: INPUT DATA TYPE G (continued)

<table>
<thead>
<tr>
<th>&lt;VARIABLE&gt;</th>
<th>&lt;DESCRIPTION&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;TPREP(I)&gt;</td>
<td>Time to prepare block I for coal loading (hr.).</td>
</tr>
<tr>
<td>&lt;NST(I)&gt;</td>
<td>Primary backfill cell number (load and haul mode).</td>
</tr>
<tr>
<td>&lt;NSTH(I)&gt;</td>
<td>Primary disposal area number (load and haul mode).</td>
</tr>
<tr>
<td>&lt;LNST(I)&gt;</td>
<td>Primary backfill cell number (loader carry mode).</td>
</tr>
<tr>
<td>&lt;LNSTH(I)&gt;</td>
<td>Primary disposal area number (loader carry mode).</td>
</tr>
<tr>
<td>&lt;BNST(I)&gt;</td>
<td>Primary backfill cell number (dozer push mode).</td>
</tr>
<tr>
<td>&lt;BNSTH(I)&gt;</td>
<td>Primary disposal area number (dozer push mode).</td>
</tr>
</tbody>
</table>

Note: I=1,NBLOCK
Primary areas are the area from which program begins its search for the destination of overburden.

Enter data in the following format:
TPREP(I) NST(I) NSTH(I) LNST(I) LNSTH(I) BNST(I) BNSTH(I)
#####.#  ##  ##  ##  ##  ##  ##  ##
### TABLE IX: INPUT DATA TYPE H

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NCLFT(I,J)&gt;</td>
<td>Code to indicate if lift J of block I is to be treated specially for overburden removal operation (0-no, 1-yes).</td>
</tr>
<tr>
<td>&lt;PRBLK(I,J)&gt;</td>
<td>Block number that has to be stripped before work on Jth lift of block I can be started.</td>
</tr>
<tr>
<td>&lt;PRLFT(I,J)&gt;</td>
<td>Lift that has to be stripped before work on Jth lift of block I can be started. This lift is in the block described by PRBLK(I,J).</td>
</tr>
<tr>
<td>&lt;ADEP(I,J)&gt;</td>
<td>Depth of the lift J of block I (feet).</td>
</tr>
</tbody>
</table>

Note: There are NLIFT(I) number of repetition of data for block I and there are NBLOCK number of blocks.

Enter data in the following format:

`NCLFT(I,J) PRBLK(I,J) PRLFT(I,J) BCF(I,J) ADEP(I,J) # ## #.## ##.##`

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;FTIME(I,J)&gt;</td>
<td>Fraction of loader time dozer is needed for feeding overburden to loader (loader carry).</td>
</tr>
<tr>
<td>&lt;FTIME2(I,J)&gt;</td>
<td>Fraction of loader time dozer is needed for feeding overburden to loader (load and haul).</td>
</tr>
</tbody>
</table>

Note: There are NLIFT(I) number of repetition of data for block I and there are NBLOCK number of blocks.

Enter data in the following format:

`FTIME(I,J) FTIME2(I,J) #.### #.##`
## TABLE IX: INPUT DATA TYPE H (continued).

<table>
<thead>
<tr>
<th>INPUT DATA ABOUT LIFTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;FBLAS(I,J)&gt; Fraction of overburden that has to be blasted of block I lift J.</td>
</tr>
<tr>
<td>&lt;FDOZ(I,J)&gt; Fraction of overburden to be dozed.</td>
</tr>
<tr>
<td>&lt;FRCD(I,J)&gt; Fraction to be loaded and carried.</td>
</tr>
<tr>
<td>&lt;FLNH(I,J)&gt; Fraction of to be loaded and hauled.</td>
</tr>
<tr>
<td>&lt;FROCK(I,J,K)&gt; Fraction of Kth rock present in lift J and block I.</td>
</tr>
</tbody>
</table>

Note: There are NLIFT(I) number of repetition of data for block I and there are NBLOCK number of blocks. For each lift there are NROCK number of FROCK value.

Enter data in the following format:

FBLAS(I,J) FDOZ(I,J) FRCD(I,J) FLNH(I,J) FROCK(I,J,K)  
#.###    #.###    #.###    #.###    #.###

| <BCF(I,J)> Correction factor for dozer production rate of lift J of block I |

Note: I=1,NBLOCK  J=1,NLIFT(I)  
Enter data in the following format:

BCF(I,J)  
#.###
TABLE X: INPUT DATA TYPE I

### INPUT DATA FOR BACKFILL CELLS ###

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;CFILCL(I)&gt;</td>
<td>Capacity of backfill cell I (cu.yd.).</td>
</tr>
<tr>
<td>&lt;BTIME(I)&gt;</td>
<td>Time needed to grade backfill cell I (hr.).</td>
</tr>
<tr>
<td>&lt;NAFIL(I)&gt;</td>
<td>Number of block that has to be mined before this backfill cell is available.</td>
</tr>
<tr>
<td>&lt;NBFIL(I,J)&gt;</td>
<td>Block number of the blocks that has to be mined before this backfill cell I is available.</td>
</tr>
</tbody>
</table>

NOTE: J=1,NAFIL(I)

Enter data in the following format:
CFILCL(I) BTIME(I) NAFIL(I) NBFIL(I,J)  
#####. #####. ## ##

### INPUT DATA ABOUT HAUL TIME ###

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;GFIL(JJ,K,1)&gt;</td>
<td>Haul time from block JJ to backfill cell K (min).</td>
</tr>
<tr>
<td>&lt;FILL(JJ,K,1)&gt;</td>
<td>Distance of block JJ to backfill cell K (feet).</td>
</tr>
</tbody>
</table>

Note: There are NBLOCK*NFILCL number of data for each variable. This table will not appear NHAUL = 0

Enter data in the following format:
GFIL(JJ,K,1) FILL(JJ,K,1)  
###.### ####.##
TABLE X: (continued)

<table>
<thead>
<tr>
<th>INPUT DATA FOR HAUL ROAD CHARACTERISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NSECFL(JJ,K)&gt; Number of section in haul road from block JJ to backfill cell K.</td>
</tr>
<tr>
<td>&lt;GFIL(JJ,K,L)&gt; Grade of segment no. L of block JJ and backfill cell K (%).</td>
</tr>
<tr>
<td>&lt;FILL(JJ,K,L)&gt; Length of segment number L of block JJ and backfill cell K (feet).</td>
</tr>
</tbody>
</table>

Note: L=1,NSECFL(JJ,K). There are NBLOCK*NFILCL number of this card. Repeat last two data NSECFL(JJ,K) times. This table will not appear if NHAUL = 1.

Enter data in the following format:
NSECFL(JJ,K) GFIL(JJ,K,L) FILL(JJ,K,L)
## ###.## ####.##
TABLE XI: INPUT DATA TYPE J

<table>
<thead>
<tr>
<th>&lt;VARIABLE&gt;</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NDCOD(J,K)&gt;</td>
<td>Code to indicate the filcell type of overburden destination (1-backfill cell, 2-spoil disposal area).</td>
</tr>
<tr>
<td>&lt;MCDEST(J,K)&gt;</td>
<td>Fillcell number of the cell specified.</td>
</tr>
</tbody>
</table>

Repeat it from JJ = 1,NBLOK. K=1,NLIFT(JJ)
Note: Provide data for only those lift indicated for special treatment.

Enter data in the following format:
NDCOD(JJ,K)  MCDEST(JJ,K)
###        ##
### TABLE XII: INPUT DATA TYPE K

#### INPUT DATA FOR EXTRA SPOIL STORAGE AREA

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;CEXST(I)&gt;</td>
<td>Capacity of spoil storage area I (cu.yd.).</td>
</tr>
</tbody>
</table>

Note: I=1,NEX

Enter data in the following format:

`CEXST(1)  CEXST(2)  CEXST(3)  ....  CEXST(NEX)
#####  #####  #####  #####
`

#### INPUT DATA ABOUT HAUL TIME

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;GEXST(JJ,K,1)&gt;</td>
<td>Haul time from block JJ to spoil storage area K (min).</td>
</tr>
<tr>
<td>&lt;HEXST(JJ,K,1)&gt;</td>
<td>Distance of block JJ to spoil storage area K (feet).</td>
</tr>
</tbody>
</table>

NOTE: There are NBLOCK *NEX data pairs. This will not appear if NHAUL=0.

Enter data in the following format:

`GEXST(JJ,K,1)  HEXST(JJ,K,1)
###.##  ####.##
`

#### INPUT DATA FOR HAUL ROAD CHARACTERISTICS

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NSECEX(JJ,K)&gt;</td>
<td>Number of section in haul road from block JJ to spoil storage area K.</td>
</tr>
<tr>
<td>&lt;GEXST(JJ,K,L)&gt;</td>
<td>Grade of segment number L of block JJ and spoil storage area K (%).</td>
</tr>
<tr>
<td>&lt;HEXST(JJ,K,L)&gt;</td>
<td>Length of segment number L of block JJ and spoil storage area K (feet).</td>
</tr>
</tbody>
</table>

Note: L=1,NSECEX(JJ,K). There are NBLOCK*NEX data group. Repeat last two data NSECEX(JJ,K) times. This will appear if NHAUL=0.

Enter data in the following format:

`NSECEX(JJ,K)  GEXST(JJ,K,L)  HEXST(JJ,K,L)
##  ###.##  ####.##
`
### TABLE XIII: INPUT DATA TYPE L

#### INPUT DATA FOR EXTRA SPOIL DISPOSAL AREA

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;CHOLLW(I)&gt;</td>
<td>Capacity of spoil disposal area I (cu.yd.).</td>
</tr>
</tbody>
</table>

Note: I=1,NHOLLW

Enter data in the following format:

CHOLLW(1) CHOLLW(2) ..... CHOLLW(I)

#### INPUT DATA ABOUT HAUL TIME

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;GHOLLW(JJ,K,1)&gt;</td>
<td>Haul time from block JJ to spoil disposal area K (min).</td>
</tr>
<tr>
<td>&lt;HHOLLW(JJ,K,1)&gt;</td>
<td>Distance of block JJ to spoil disposal area K (feet).</td>
</tr>
</tbody>
</table>

NOTE: There are NBLOCK*NHOLLW number of data for each variable. This will appear if NHAUL=1.

Enter data in the following format:

GHOLLW(JJ,K,1) HHOLLW(JJ,K,1)

#### INPUT DATA FOR HAUL ROAD CHARACTERISTICS

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NSECHW(JJ,K)&gt;</td>
<td>Number of section in haul road from block JJ to spoil disposal area K.</td>
</tr>
<tr>
<td>&lt;GHOLLW(JJ,K,L)&gt;</td>
<td>Grade of segment number L of block JJ and spoil disposal area K (%)</td>
</tr>
<tr>
<td>&lt;HHOLLW(JJ,K,L)&gt;</td>
<td>Length of segment number L of block JJ and spoil disposal area K (feet).</td>
</tr>
</tbody>
</table>

Note: L=1,NSECHW(JJ,K). There are NBLOCK*NHOLLW number of this data group. Repeat last two data NSECHW(JJ,K) times. This will appear if NHAUL=0.

Enter data in the following format:

NSECHW(JJ,K) GHOLLW(JJ,K,L) HHOLLW(JJ,K,L)

###
### TABLE XIV: INPUT DATA TYPE M

#### INPUT DATA ABOUT HAUL TIME FOR BACKFILL CELL TO STORAGE AREA

| <FGFIL(JJ,K,L)> | Haul time from storage area JJ to backfill cells K (min). |
| <FFILL(JJ,K,L)> | Distance of storage area JJ to backfill cell K (feet). |

**NOTE:** There are \(NEX*NFILCL\) number of data pairs. This will appear if \(NHAUL=1\).

Enter data in the following format:
\[
<\text{FGFIL}(JJ,K,L)\> \quad <\text{FFILL}(JJ,K,L)\>
\]
###.###  ###.###

#### INPUT DATA FOR HAUL ROAD CHARACTERISTICS

| <NSECEF(JJ,K)> | Number of section in haul road of storage area JJ to spoil storage area K. |
| <FGFIL(JJ,K,L)> | Grade of segment no. \(L\) of storage area JJ to backfill cell K (%). |
| <FFILL(JJ,K,L)> | Length of segment number \(L\) of storage area JJ to backfill cell K (feet). |

**Note:** \(L=1,NSECEF(JJ,K)\). There are \(NEX*NFILCL\) number of this data group. Repeat last two data \(NSECEF(JJ,K)\) times. This will appear if \(NHAUL=0\).

Enter data in the following format:
\[
<\text{NSECEF}(JJ,K)\> \quad <\text{FGFIL}(JJ,K,L)\> \quad <\text{FFILL}(JJ,K,L)\>
\]
###  ###.###  ###.###
TABLE XV: INPUT DATA TYPE N

<table>
<thead>
<tr>
<th>INPUT COST DATA</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;CHCOST&gt;</td>
<td>Coal hauling cost ($/ton).</td>
</tr>
<tr>
<td>&lt;ZNHP&gt;</td>
<td>Hourly rate for hourly personnel ($).</td>
</tr>
<tr>
<td>&lt;ZSAL&gt;</td>
<td>Salary of salaried personnel ($/year).</td>
</tr>
<tr>
<td>&lt;NHP&gt;</td>
<td>Number of hourly personnel.</td>
</tr>
<tr>
<td>&lt;NSP&gt;</td>
<td>Number of salaried personnel</td>
</tr>
<tr>
<td>&lt;WELP&gt;</td>
<td>Welfare trust (%).</td>
</tr>
<tr>
<td>&lt;OBFT&gt;</td>
<td>Benefit trust ($/ton).</td>
</tr>
<tr>
<td>&lt;EXRECL&gt;</td>
<td>Grading cost ($/acre).</td>
</tr>
<tr>
<td>&lt;TAREA&gt;</td>
<td>Total area of mine (acres).</td>
</tr>
<tr>
<td>&lt;CSEED&gt;</td>
<td>Seeding cost ($/acres).</td>
</tr>
</tbody>
</table>

Enter data in the following format:

CHCOST ZNHP ZSAL NHP NSP WELP OBFT EXRECL TAREA CSEED
##.## ##.## #####. ### ### ##.## ##.## ####.# #####. ###.#

<table>
<thead>
<tr>
<th>INPUT DATA FOR BLASTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;DDI&gt;</td>
</tr>
<tr>
<td>&lt;WHFT&gt;</td>
</tr>
<tr>
<td>&lt;PRMCST&gt;</td>
</tr>
<tr>
<td>&lt;AFCST&gt;</td>
</tr>
<tr>
<td>&lt;SLCOST&gt;</td>
</tr>
<tr>
<td>&lt;DETCST&gt;</td>
</tr>
<tr>
<td>&lt;BLCST&gt;</td>
</tr>
<tr>
<td>&lt;PRMRT&gt;</td>
</tr>
<tr>
<td>&lt;COPT&gt;</td>
</tr>
</tbody>
</table>

Enter data in the following format:

DDI WHFT PRMCST AFCST SLCOST DETCST BLCST PRMRT
##.# #.## ####.# ####.## ###.## ###.## #.##
### TABLE XVI: INPUT DATA TYPE 0

**INPUT COST DATA FOR EQUIPMENT AND SUPPLY**

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCTIRE</td>
<td>Code to indicate if tire mounted (0-no, 1-yes)</td>
</tr>
<tr>
<td>DPRICE</td>
<td>Delivered price ($/unit).</td>
</tr>
<tr>
<td>RESVAL</td>
<td>Resale value ($/unit).</td>
</tr>
<tr>
<td>DEPPER</td>
<td>Depreciation period (hours).</td>
</tr>
<tr>
<td>TIIFAC</td>
<td>Interest, insurance and tax cost ($/hour).</td>
</tr>
<tr>
<td>FUELUP</td>
<td>Fuel price ($/gallon).</td>
</tr>
<tr>
<td>FUELCS</td>
<td>Fuel consumption (gallon/hour).</td>
</tr>
</tbody>
</table>

Enter data in the following format:

NCTIRE DPRICE RESVAL DEPPER TIIFAC FUELUP FUELCS
# ###### ###### #.## #.## ##.## ###.##

| CLFGPH | Cost of lubricant, grease, etc. ($/hour). |
| EXUMUL | Extended use multiplier. |
| BRF    | Basic repair factor. |
| NOPER  | Number of operator. |
| OPERHW | Operator wage ($/hour). |

Enter data in the following format:

CLFGPH EXUMUL BRF NOPER OPERHW
###.## #.## #.## #.## ### #.##

| I | Impact factor. |
| A | Abrasiveness factor. |
| Z | "Z" factor. |
| BF | Basic factor. |

Note: This card will not appear if NCTIRE = 1

Enter data in the following format:

I A Z BF
#.## #.## #.## #.## #.##

| TRCOST | Tire cost ($/tire). |
| ESTLIH | Estimated life of tires (hour). |

Note: Do not provide this card if NCTIRE = 0

Enter data in the following format:

TRCOST ESTLIH
#####.# #####.#

Note: This table will appear on the screen for every type of equipment if COPT=1 (see table XV).
TABLE XVI: INPUT DATA TYPE O (continued)

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;OPCOST&gt;</td>
<td>Operating cost ($/hour).</td>
</tr>
<tr>
<td>&lt;OWCOST&gt;</td>
<td>Owning cost ($/hour).</td>
</tr>
</tbody>
</table>

Note: This will not appear if COPT = 0

Enter data in the following format:

OPCOST   OWCOST
###.#   ###.#
Once the working copy has been made prepare the data file using the DATA program or by any editor program. Once the data file has been prepared the program can be executed by entering OPSIM and then pressing ENTER. The following will appear on the screen:

SURFACE MINING SIMULATION PROGRAM. FOLLOW INSTRUCTIONS IN ORDER TO RUN THE PROGRAM

ENTER THE NAME OF INPUT FILE

-- enter the name of the file containing data. <RETURN> (e.g. AMAL.DAT).

ENTER OUTPUT OPTION 1: PRINTER

2: FILE

-- enter option <RETURN>. If choice is 2 then it will ask:

ENTER THE NAME OF OUTPUT FILE

-- enter output file name. <RETURN> (e.g. AMAL.LIS)

ENTER THE NUMBER OF SIMULATION RUNS

-- enter number of times to run simulation <RETURN> (e.g. 1)

ENTER MAXIMUM NUMBER OF MACHINES THAT CAN BE USED IN ANY LIFT AT THE SAME TIME
DRILL, DOZER, TRUCK (OVERBURDEN), LOADER, TRUCK (COAL), LOADER
-- enter number <RETURN> (e.g. 1 2 2 1 2 1)

ENTER THE SEQUENCE IN WHICH BACKFILL CELLS ARE TO BE FILLED
-- enter sequence <RETURN> (e.g. 1 2 3 4)

ENTER THE SEQUENCE IN WHICH BLOCKS ARE TO BE STRIPPED.
-- enter value <RETURN> (e.g. 1 2 3 4 5 6 7 8 9 10)

ENTER NUMBER OF BREAK POINTS FOR OVERBURDEN REMOVAL FROM EXTRA SPOIL STORAGE AREA TO FILL CELLS
-- enter value <RETURN> (e.g. 1)

ENTER CORRESPONDING BLOCK NUMBER
-- enter sequence <RETURN> (e.g. 10)

After mining this block the program checks if any overburden stored in any temporary storage area. If yes then they are hauled back to fill cells if available.

EXECUTION STARTED. PLEASE WAIT

If choice of output was the printer then after a few seconds it will start printing results otherwise the output file will be in the disk file with name given as input. The program can be run several times with different sequences and machines to recognize the optimum design and equipment combination.
IV. OUTPUT OF THE PROGRAM

The program can be used to analyze different mining plans as well as to test the economic feasibility of mining a deposit. The program output consists of equipment performance data as well as the cost of mining. From the equipment performance an operator can draw conclusions about whether to increase or decrease the number of units of a particular piece of equipment. If a particular piece of equipment is busy most of the time, and other equipment is idle most of the time, one could increase the number of pieces of the equipment that is busy, or reduce the other equipment type. The decision whether to increase a particular equipment type or reduce others can be made by considering the overall utilization of equipment. If the utilizations were quite low, the number of pieces of equipment should be reduced. However, idling of equipment may be due to poor sequencing. The value of the match factor should help in drawing conclusions regarding truck and loader matching. The term "MATCH FACTOR" is defined as follows:

$$\text{MATCH FACTOR} = \frac{\text{LT} + \text{MT}}{\text{THT} + \text{DT} + \text{MT}}$$

where \( \text{LT} = \) loading time,
MT = maneuvering time (truck),
THT = total haul time,
DT = dumping time.

The amount of "material rehandled" should give a fair idea about the mining sequence. If this value is very high then a new sequencing should be tried. Material rehandling would be high if the backfilling sequence and stripping sequence do not match properly.

4.1. Statistical Analysis of Output Data:

In simulation studies inferences regarding the performance of a system are made based on the results obtained from simulation. If the simulation model contains random elements, the output of the simulation consists of observed samples of random variables. Thus, to make any assertion about the operation of the system, one should consider the inherent variability of the output. This variability can be taken into account through building a confidence interval or through hypothesis testing.

4.1.1. Confidence intervals

The parameter of primary interest in simulation analysis is the population mean. If the samples are independent and identically distributed, then from the Central Limit Theorem, one can say the sample mean $X$ is
approximately normally distributed for a sufficiently large sample size. If we assume that \( \bar{X} \) is normally distributed then it can be shown that \( Z = (\bar{X} - M) / \sqrt{V_X} \) is normally distributed with mean 0 and standard deviation 1 and:

\[
P[-Z_{a/2} < Z < Z_{a/2}] = 1-a
\]

where:

- \( M \) = population mean,
- \( \sqrt{V_X} \) = variance,

and \( Z_{a/2} \) is the value for \( Z \) such that the area to its right on a standard normal curve equals \( a/2 \).

Hence, one can assert with probability (1-a) that

\[
\bar{X} - Z_{a/2} \cdot \sqrt{V_X} < M < \bar{X} + Z_{a/2} \cdot \sqrt{V_X}
\]

Hence, a (1-a) confidence interval for \( M \) using the estimator \( S_X \) is given by:

\[
\bar{X} - t_{a/2, N-1} \cdot S_X < M < \bar{X} + t_{a/2, N-1} \cdot S_X
\]

where \( t_{a/2, N-1} \) is the critical value of the t statistic with \( N-1 \) degrees of freedom and \( N \) is the sample size. \( S_X \) can be substituted with \( S_X/N \) as it was assumed \( \bar{X}_n \) are identically independent distributions.
4.1.2. Initial condition

The initial conditions for a simulation may cause the values obtained from the model to be different from the real system. However, if the real system is of a terminating type one could simulate the system from beginning to end, thus avoiding the complications of the start-up procedure and termination conditions. The same technique has been applied in the model as the system is of a terminating type.

4.1.3 Parameters estimation:

There are several methods available for estimating population mean and variance of the response variable from the simulation results. Some of these are listed below:

1. Replication,
2. Subintervals,
3. Regeneration,
4. Parametric modelling,

As only the replication method can be used for a terminating type of simulation model, it is of particular interest here.

The method of replication: In this procedure the simulation is run independently several times. Suppose the number of
run is n and the observed value for ith run is \( X_i \). Then the estimator for the mean would be:

\[
\bar{X} = \frac{x_1 + x_2 + \ldots + x_n}{n}
\]

and the estimator of the variance would be:

\[
S^2 = \frac{(x_i - \bar{X})^2}{n-1}, \quad i = 1, 2, \ldots, n
\]

4.1.4 Interopretation

It can be seen from the expression of confidence interval that the interval width depends on \( V_X \). The smaller the \( V_X \) the smaller is the width. It is better to reduce the variance so that the probability will be higher for the same interval to contain the population mean. There are several methods available to reduce the variance. The technique suggested by Pritsker (1979), which is of particular interest here and has been applied in our model is antithetic sampling. The antithetic sampling is described below.

In deriving the expression for \( V_X \) samples are assumed identically and independently distributed (IID). If IID is not assumed, then it can be shown that

\[
\text{VAR}[X] = \frac{1}{N^2} \sum_{i=1}^{N} (\text{VAR}[X_i]) + \sum_{i=1}^{N} \sum_{j=i}^{N} \text{COV}[X_i, X_j]
\]
where \( \text{COV}[X_i, X_j] \) is the covariance between the mean of the \( i \)th and \( j \)th run. If the individual runs are independent the covariances are zero. However, if one can make the covariance negative then the variance will be lesser than the variance with independent runs.

If one performs pairs of independent runs in which antithetic sampling is used in the second run of the pair, then the variance calculation for \( 2N \) runs is simplified by combining values across pairs of runs. If \( X_j \) is the antithetic value for \( X_i \), then:

\[
\text{VAR} [X] = \sum_{i=1}^{N} \text{VAR} (U_i/2N)
\]

where \( U_i = (X_i + X_j)/2 \).

When combined in this fashion, covariance terms between runs need not be computed. There are two points to be noted. First, although the correlation between antithetic random numbers is \(-1\), the correlation between samples based on such random numbers may not be \(-1\). The second point is that, for square or higher even power relations, the introduction of a negative correlation can result in an increase in the variance of the sample mean.
V. A CASE STUDY

Detailed operational data were collected from a experimental mining site of the Amos Ridge Coal Co., Wise County, Virginia, by Mr. Carl Zipper of the Department of Agronomy. This data has been used to validate our model and to show, as an example, how the model can be used to analyze alternative mining plans. A summary of the characteristics of the mine is given below.

Three seams are being mined by the mountain top removal method -- the Upper Marker (2 ft.), the Lower Splinter (4 ft.), and Lower Marker (2 ft.).

The data were collected from this mine between January 1st and September 15th, 1984. There were ten blocks mined during this period. Six of these blocks were from the Lower Marker seam, three Lower Splinter and one Upper Marker. The topography and the positions of the blocks are shown in Figure 9. The lithographic structure is shown in Figure 10. The areas of the blocks and the volumes of the lifts are shown in Table XVII. Coal production during this period is shown in Table XVIII. The fraction of material that has been removed by each operation is shown in Table XIX. Table XX shows the inventory of major equipment. Most of the equipment was old and their cost features could not be
Table XVII: Block and Lift Dimensions

<table>
<thead>
<tr>
<th>Block No.</th>
<th>Area of Overburden (sq.ft)</th>
<th>Area of Coal (sq.ft)</th>
<th>Volume of Top Lift (cu.yd)</th>
<th>Volume of Bottom Lift (cu.yd)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-LM</td>
<td>22500</td>
<td>31800</td>
<td>15600</td>
<td>22700</td>
</tr>
<tr>
<td>2-LM</td>
<td>17245</td>
<td>19500</td>
<td>12100</td>
<td>15100</td>
</tr>
<tr>
<td>3-LM</td>
<td>23800</td>
<td>26900</td>
<td>13000</td>
<td>17900</td>
</tr>
<tr>
<td>4-LM</td>
<td>34000</td>
<td>34400</td>
<td>18200</td>
<td>22200</td>
</tr>
<tr>
<td>5-LM</td>
<td>11376</td>
<td>11000</td>
<td>none</td>
<td>7300</td>
</tr>
<tr>
<td>6-LM</td>
<td>28834</td>
<td>23700</td>
<td>14300</td>
<td>19600</td>
</tr>
<tr>
<td>1-LS</td>
<td>12150</td>
<td>13234</td>
<td>7200</td>
<td>7900</td>
</tr>
<tr>
<td>2-LS</td>
<td>28750</td>
<td>22600</td>
<td>54000</td>
<td>13500</td>
</tr>
<tr>
<td>3-LS</td>
<td>18245</td>
<td>23700</td>
<td>6500</td>
<td>17000</td>
</tr>
<tr>
<td>1-UM</td>
<td>12000</td>
<td>12800</td>
<td>11100</td>
<td>11800</td>
</tr>
</tbody>
</table>
Table XVIII: Coal Production in Amos Ridge

<table>
<thead>
<tr>
<th>From</th>
<th>Date</th>
<th>To</th>
<th>Coal Produced (tons)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1-1984</td>
<td>1-15</td>
<td></td>
<td>1341</td>
</tr>
<tr>
<td>1-16</td>
<td>1-31</td>
<td></td>
<td>1356</td>
</tr>
<tr>
<td>2-1</td>
<td>2-15</td>
<td></td>
<td>2113</td>
</tr>
<tr>
<td>2-16</td>
<td>2-29</td>
<td></td>
<td>1985</td>
</tr>
<tr>
<td>3-1</td>
<td>3-15</td>
<td></td>
<td>1584</td>
</tr>
<tr>
<td>3-16</td>
<td>3-31</td>
<td></td>
<td>1207</td>
</tr>
<tr>
<td>4-1</td>
<td>4-16</td>
<td></td>
<td>1999</td>
</tr>
<tr>
<td>4-17</td>
<td>4-30</td>
<td></td>
<td>693</td>
</tr>
<tr>
<td>5-1</td>
<td>5-15</td>
<td></td>
<td>3009</td>
</tr>
<tr>
<td>5-15</td>
<td>5-30</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>6-1</td>
<td>6-15</td>
<td></td>
<td>2075</td>
</tr>
<tr>
<td>6-15</td>
<td>6-30</td>
<td></td>
<td>456</td>
</tr>
<tr>
<td>7-1</td>
<td>7-15</td>
<td></td>
<td>1206</td>
</tr>
<tr>
<td>7-15</td>
<td>7-30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-1</td>
<td>8-15</td>
<td></td>
<td>956</td>
</tr>
</tbody>
</table>
Table XIX: Fraction of Lifts Removed by Each Operation

<table>
<thead>
<tr>
<th>Block No.</th>
<th>Lift No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-LM</td>
<td>Fraction drilled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction dozed</td>
<td>.5</td>
<td>.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction loaded and hauled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td>2-Lm</td>
<td>Fraction drilled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction dozed</td>
<td>.5</td>
<td>.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction loaded and hauled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td>3-LM</td>
<td>Fraction drilled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction dozed</td>
<td>.5</td>
<td>.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction loaded and hauled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td>4-LM</td>
<td>Fraction drilled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction dozed</td>
<td>.5</td>
<td>.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction loaded and hauled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td>5-LM</td>
<td>Fraction drilled</td>
<td>.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction dozed</td>
<td>.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction loaded and hauled</td>
<td>.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-LM</td>
<td>Fraction drilled</td>
<td>.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction dozed</td>
<td>.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction loaded and hauled</td>
<td>.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-US</td>
<td>Fraction drilled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction dozed</td>
<td>.5</td>
<td>.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction loaded and hauled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td>2-LS</td>
<td>Fraction drilled</td>
<td>.8</td>
<td>.8</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>Fraction dozed</td>
<td>.5</td>
<td>.5</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>Fraction loaded and hauled</td>
<td>.8</td>
<td>.8</td>
<td>1.0</td>
</tr>
<tr>
<td>3-LS</td>
<td>Fraction drilled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction dozed</td>
<td>.5</td>
<td>.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction loaded and hauled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td>1-UM</td>
<td>Fraction drilled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction dozed</td>
<td>.5</td>
<td>.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fraction loaded and hauled</td>
<td>.8</td>
<td>.8</td>
<td></td>
</tr>
</tbody>
</table>
Table XX: Inventory of Major Equipment

<table>
<thead>
<tr>
<th>Equipment Type</th>
<th>Model/Manufacturer</th>
<th>Number of Pieces of Equipment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haulers</td>
<td>Terex 773B CAT.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>992 CAT.</td>
<td>1</td>
</tr>
<tr>
<td>Loaders</td>
<td>988 CAT.</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>945 Fiat Allis</td>
<td>1</td>
</tr>
<tr>
<td>Dozers</td>
<td>D9G CAT.</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>FL14 Fiat Allis</td>
<td>1</td>
</tr>
<tr>
<td>Drills</td>
<td>Chicago Pneumatic</td>
<td>1</td>
</tr>
</tbody>
</table>
Figure 9: Amos Ridge Mining Site
sandstone
fossil rock
sandstone

30' - 35' siltstone and stone

toxic layer
Upper Marker Seam

60' - 65' sandstone

siltstone and shale

44'' to 50'' Lower Marker Seam split by approximately 1' of sandstone

15' siltstone and shale

20' sandstone

22'' - 26'' Lower Marker Seam

Figure 10. Overburden Sequence of Amos Ridge Mine
collected from the site, therefore, industry averages were assumed. Accurate haul road characteristics from each block to each fill cell could not be collected. Instead, approximate haul time was given as input.

The program was run first with the design and equipment that has been used in the mine. The input data and the output are shown in Tables XXI to XXIII.

Output:

The salient outputs are shown in Table XXII. The cost of coal per ton was $22.77. In actual operation it was approximately $21.5. The discrepancy could probably be attributed to the high degree of approximation that had to be made about cost data and haul time. A conservative approach was taken in approximating cost data and thus resulted in a higher cost per ton of coal mined. From the output, it can be seen that drilling time was more than the dozing time and loading time. Intuition suggests that dozers and haulers might have spent much time waiting for the drill to finish its job, so the number of drills should be increased. It can also be seen that the match factor was very high. This factor suggest that either the number of trucks should increased or the number of loaders should be decreased.
### TABLE XXI: INPUT DATA FILE

<table>
<thead>
<tr>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0 1 10 4 1 0 480. 0.0450 1 24 0.900 1 1.20 1.35 1.30 1.00</td>
</tr>
<tr>
<td>B</td>
<td>1 1 12.0 12.0 0 1.20 1.10 .00 1.30 .00</td>
</tr>
<tr>
<td>C</td>
<td>3 300.00 400.00 20. .20 .00 20.00</td>
</tr>
<tr>
<td>D</td>
<td>2 1 0 0 0.20 .30 12.00 32.00 30.50 31.00 30.50 1</td>
</tr>
<tr>
<td>E</td>
<td>3 2 0 0 5.25 5.25 200.</td>
</tr>
<tr>
<td>F</td>
<td>2 22500. 31800. 6.00 2 17425. 19500. 2. 1 1 6.00 2 23800. 26900. 2. 1 3 12.00 2 34000. 34400. 2. 2 3 10.00</td>
</tr>
<tr>
<td>G</td>
<td>0 0 0 18.72 .00 .00 0 0 0 27.24 .00 .00 0 0 0 18.74 .00 .00 0 0 0 23.40 .00 .00 0 0 0 14.74 .00 .00</td>
</tr>
</tbody>
</table>
TABLE XXI: (continued)

<table>
<thead>
<tr>
<th></th>
<th>0 0 0 20.30</th>
<th>.00  .00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 0 0 14.45</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 17.62</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 14.40</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 13.40</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 18.34</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 25.35</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 25.35</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 25.35</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 25.35</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 25.35</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 25.35</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 9.62</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 9.62</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 25.00</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>0 0 0 26.55</td>
<td>.00  .00</td>
</tr>
<tr>
<td></td>
<td>.80  .40</td>
<td>.00  .60</td>
</tr>
<tr>
<td></td>
<td>.80  .40</td>
<td>.00  .60</td>
</tr>
<tr>
<td></td>
<td>.80  .40</td>
<td>.00  .60</td>
</tr>
<tr>
<td></td>
<td>.80  .40</td>
<td>.00  .60</td>
</tr>
<tr>
<td></td>
<td>.80  .40</td>
<td>.00  .60</td>
</tr>
<tr>
<td></td>
<td>.80  .40</td>
<td>.00  .60</td>
</tr>
<tr>
<td></td>
<td>.80  .40</td>
<td>.00  .60</td>
</tr>
<tr>
<td></td>
<td>.80  .40</td>
<td>.00  .60</td>
</tr>
<tr>
<td></td>
<td>.80  .40</td>
<td>.00  .60</td>
</tr>
<tr>
<td></td>
<td>.80  .40</td>
<td>.00  .60</td>
</tr>
<tr>
<td></td>
<td>.80  .40</td>
<td>.00  .60</td>
</tr>
<tr>
<td></td>
<td>.80  .30</td>
<td>.00  .70</td>
</tr>
<tr>
<td></td>
<td>.80  .30</td>
<td>.00  .70</td>
</tr>
<tr>
<td></td>
<td>.80  .30</td>
<td>.00  .70</td>
</tr>
<tr>
<td></td>
<td>.80  .30</td>
<td>.00  .70</td>
</tr>
<tr>
<td></td>
<td>.80  .30</td>
<td>.00  .70</td>
</tr>
<tr>
<td></td>
<td>.80  .30</td>
<td>.00  .70</td>
</tr>
<tr>
<td></td>
<td>.80  .00</td>
<td>.00  1.00</td>
</tr>
<tr>
<td></td>
<td>.90  .50</td>
<td>.00  .50</td>
</tr>
<tr>
<td></td>
<td>.90  .50</td>
<td>.00  .50</td>
</tr>
<tr>
<td></td>
<td>.90  .50</td>
<td>.00  .50</td>
</tr>
<tr>
<td></td>
<td>.90  .50</td>
<td>.00  .50</td>
</tr>
<tr>
<td></td>
<td>.90  .50</td>
<td>.00  .50</td>
</tr>
<tr>
<td></td>
<td>.76  .76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.76  .76</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>.76  .76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.76  .76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.76</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td></td>
<td>.76</td>
<td>.76</td>
</tr>
<tr>
<td>G</td>
<td>.76</td>
<td>.76</td>
</tr>
<tr>
<td></td>
<td>15300</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>25950</td>
<td>2.00</td>
</tr>
<tr>
<td>H</td>
<td>20300</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td>15850</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td>.00</td>
<td>.00</td>
</tr>
<tr>
<td></td>
<td>.00</td>
<td>.00</td>
</tr>
<tr>
<td></td>
<td>.5</td>
<td>200.0</td>
</tr>
<tr>
<td></td>
<td>.5</td>
<td>200.6</td>
</tr>
<tr>
<td></td>
<td>.6</td>
<td>300.6</td>
</tr>
<tr>
<td></td>
<td>.4</td>
<td>175.6</td>
</tr>
<tr>
<td></td>
<td>1.1</td>
<td>100.1</td>
</tr>
<tr>
<td></td>
<td>.5</td>
<td>200.5</td>
</tr>
<tr>
<td></td>
<td>653500.0</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>1.20</td>
<td>500.1</td>
</tr>
<tr>
<td></td>
<td>1.10</td>
<td>400.1</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>.5</td>
<td>.05</td>
</tr>
<tr>
<td></td>
<td>1700000.3</td>
<td>30000.2</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td>12000.00</td>
<td>2500.00</td>
</tr>
<tr>
<td></td>
<td>1241000.7</td>
<td>23000.18</td>
</tr>
<tr>
<td></td>
<td>.7</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td>12500.00</td>
<td>2500.00</td>
</tr>
<tr>
<td></td>
<td>1140000.3</td>
<td>36000</td>
</tr>
<tr>
<td></td>
<td>.7</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td>12500.00</td>
<td>1500.00</td>
</tr>
<tr>
<td></td>
<td>1136000.3</td>
<td>34000</td>
</tr>
<tr>
<td></td>
<td>.7</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td>5000.00</td>
<td>2200.00</td>
</tr>
<tr>
<td></td>
<td>11136000.3</td>
<td>34000</td>
</tr>
<tr>
<td></td>
<td>.7</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td>5000.00</td>
<td>2250.00</td>
</tr>
</tbody>
</table>
TABLE XXII: SALIENT OUTPUTS OF FIRST RUN

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Completion time (shifts)</td>
<td>138</td>
</tr>
<tr>
<td>Drilling time (hr.)</td>
<td>708.5</td>
</tr>
<tr>
<td>Dozing time (hr.)</td>
<td>288.1</td>
</tr>
<tr>
<td>Trucking time (hr.)</td>
<td>551.9</td>
</tr>
<tr>
<td>Match factor</td>
<td>1.7</td>
</tr>
<tr>
<td>Cost of coal (dollar/ton)</td>
<td>22.77</td>
</tr>
</tbody>
</table>
It can also be seen that equipment utilization is quite low, so it may be better to reduce the number of pieces of equipment. After these changes, the program was run a second time with one drill, one overburden loader, one coal loader, two overburden trucks, and one dozer. Some outputs are shown in Table XXII. The cost of mining was found as $20.22 per ton.

Recommendation: From the above results it can be seen that equipment utilization in the second case is much better than in the previous one; thus, the cost of mining per ton of coal is less. However, more equipment may be suitable if one can sequence the working of blocks in a better way. Sequencing should be done so that interdependency between blocks would be less, while time to move machinery from one place to another is minimal. The backfilling sequence should be in concordance with the stripping sequence, which will reduce material rehandling.
## TABLE XXIII: SALIENT OUTPUTS OF SECOND RUN

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Completion time (shifts)</td>
<td>162</td>
</tr>
<tr>
<td>Drilling time (hr.)</td>
<td>708.2</td>
</tr>
<tr>
<td>Dozing time (hr.)</td>
<td>428.1</td>
</tr>
<tr>
<td>Trucking time (hr.)</td>
<td>551.9</td>
</tr>
<tr>
<td>Match factor</td>
<td>1.7</td>
</tr>
<tr>
<td>Cost of coal (dollar/ton)</td>
<td>20.22</td>
</tr>
</tbody>
</table>
VI. ADDITIONAL APPLICATIONS AND FUTURE REFINEMENTS

6.1 Additional Applications

Though the program was written to analyze alternative material handling plans, it can also be used to select equipment and to make decisions regarding profitability of mining a particular deposit.

6.2 Suggestion for Future Refinements

In this program, the destination of the overburden from any lift is fixed, given the current status of the fill cells and spoil storage area. However, in reality it is seldom so. In most of the cases the operator uses his judgement depending on the present conditions. This difficulty can be surpassed by use of the "Gaming Method." In this method after each operation is over the operator will be asked regarding the next operation. He would be provided with the current status of each lift, fill cell, equipment availability, etc. This method will create a unique opportunity to combine pre-mining design and operator's experience to find an optimum solution.
REFERENCES


APPENDIX A

Main Program
CSJOB

AMAL,PAGE=50,TIME=4

***********************************************************************
* SURFACE MINING SIMULATION PROGRAM *
***********************************************************************

BY

AMAL CHAKRABORTY

***********************************************************************
* THIS IS APPLICABLE TO MOUNTAINTOP CUT MINING. THE PROGRAM CAN *
* BE USED FOR DIFFERENT TYPE OF RECLAMATION PROCEDURE. FOR *
* DETAILS SEE MANUAL.                                            *
***********************************************************************

COMMON/BRES/DAREA(10),CAREA(10),CTIK(10),AEP(10,5),PBLAS(10,5),
+ CREVR,PDOZ(10,5),FLNH(10,5),PPAC(3),SWELF(3),FROCK(10,5,3),
+ TOVL

COMMON/BTR/TEX,DMP_GWV,SPEED(10),RP(10),HCTR,TCTR,HDIS(10,5),
+ RORI,NSEG,N1,MATCH,THATCH,N2,VC,DA,AMAX,VNAX
+ CWT(20),DHLIL(10),TTR,N1,A,ASV,LACR,ACVL,CHD(10)
+ LHDC,N2,CCPL,CAVL,WNR,GTOVB,CHIL,CTTT
+ K,NTR,TOVB,THL,TTL,SCYCLE,WMATL,HITME,
+ TIME,CYCLE,RHT(10,5),CRHT(20),CTHT(10),CTL(10,5),
+ (10,5),TDI(10,5),CQDL(10),CTDL(10),CQDT(10),WTR(10,5),
+ TTR(10,5),CTTR(10),CAVTR(10),CCT(10)
+ TCCT(10,5),DLCT(20,5),CCLT(10,5),CQLT(10,5),CCLT(10,5)

COMMON /BTR/NAFIL(10),NBFL(10,5),NFCODE(10),NCFIL(10),SEQCL(10),
+ VOLL(5),CHOLW(5),GEIX(5),VOLLEX(5),
+ NCLFT(10,5),FLFT(10,5,5),GHOLLW(10,5,5),HOLLW(10,5,5),GEIXST(5),
+ (10,5,5),NSECAL(10,5,5),NSECCH(10,5,5),CFIRL(10),
+ TMINE,OBMINE,ROBN,WHOLLW,NEX,TYOLLW,VMAX,INDK
+ DIST,ICSV,ISTR,IPSV,PSV(20),HTR(20),PTR(20)
+ TVSV,HGCV(20),CSV(20),PGS,HISV(20),AVSPD,FTIME(10,5),

+ FTIME2(10,5),NSCSEX(10,5),GPIL(10,10,5),NSECFL(10,10),NFIILC

COMMON/GEN/RS(6),NCODE(10,5,8),JOB(10,5,8),CJOB(10),NLIFT(10),
+ NREDO(10,5,8),CTIM,WT(10,5,4),CODE,COF,SLIFT(10,5)
+ ISED,IEBLK,ILIFT,CON(3),RSPACE(6),NBLK,NMACH(10)
+ NREDO(10,5,9),NROCK,ISTOP,TIPREP(10),INDEX,FNAME

COMMON/BNEW/NCLB(10),NCFL(10),ETIME(10),FGFL(5,10,5)
+ FILL(5,10,5),CVOLL(10),NSECFL(5,10),NREADY(10),TIPT(5,10)
+ NMBF(5,2),VFILC(10),EFHT(5),FRHT(5),FDLCT(5),FTLT(5)
+ FWTR(5),FDI5(5),NCSEX(5),FDLFT(5),FTR(5),FHT(5)

COMMON/DNEW/INSH(10),INST(10),BNST(10),BNST(10),
+ NCEL(10),EFFECT(5),FRCD(10,5),NSTD(10),NNEX(5),BFROM(10,10,4),
+ BBF(10,10,4),LFLF(10,10,4),BQW(10,10,4),BST(10)
+ LFROM(10,10,4),AMOUNT(10,10,4),NUMBER(10,4,3),TSTART(10,5,8)
+ TFINISH(5),CDLT(10,5),CPOINT(10,10,4),BCFROM(10,10,4)
+ LCFROM(10,10,4),TCODS(5),PGS(5),NDCOD(10,5),MCDES(10,5)

COMMON/BR/DS,NPDR,ADR,HIST(20),PR(20),ISDR,TFDR(10,5),TDTR
+ TCO(10,5),CCDR,TCDR,MOVE

COMMON/BRUL/NPBL,BGF(10,5),ABUL,HBIL(20),PBFL(20),PBBL(20),ISBL
+ TDZ(10,5),OBBL,ITBL,GTIME(10),NUMBG(10),BTIME(10)

COMMON/ION/PBLK(10),PSLFT(10),PRBLK(10,5),PRLFT(10,5),SEQBB(10)
+ JLIPT(10),BPREP(10),CDEN,NAUL,CTSF,NDY,NDH,NCOD2(10,5),
+ NNDZ(10,5),TRATE1,TRATE2,CONST,ETIME,DTIME,GAP

INTEGER PSBLK,PSLFT,PRBLK,PRLFT,SEQBB,JLIPT,PRDR(7),PRTR
INTEGER SECTON,CCODE,CJOB,PCSV,PCTR,PCFL,PRBL,BFROM,LFROM, +BNST,FREE(8),SLIFT,TTIM,START,DGR,BFROM,SBFROM,BNSTH 
REAL LTIME,NNEX 
CHARACTER*64 FNAME 
CHARACTER*64 FNAME1 
WRITE(*,48) 
48 FORMAT(///) 
WRITE(*,70) 
70 FORMAT(10X,'*********************************************') 
+ /10X,'* OPSIM *' 
+ /10X,'* SURFACE MINING SIMULATION PROGRAM *' 
+ /10X,'* Amal Chakraborty *' 
+ /10X,'* Department of Mining Engineering *' 
+ /10X,'* Virginia Polytechnic Institute and *' 
+ /10X,'* State University. *') 
71 FORMAT(10X,'* August 1985 *') 
+ /10X, '*********************************************') 
WRITE(*,48) 
WRITE(*,*) PRESS <RETURN> TO CONTINUE ' 
WRITE(*,*) 'ENTER THE NAME OF INPUT FILE' 
READ(*,'(A)')FNAME 
OPEN(3,FILE=FNAME) 
WRITE(*,787) 
787 FORMAT(///' ENTER OUTPUT OPTION 1: PRINTER'/ 
2: FILE '///) 
READ(*,*),NLOPT 
IF(NLOPT.EQ.1) Then 
OPEN(6,FILE='PRN') 
ELSE 
WRITE(*,*) 'ENTER THE NAME OF OUTPUT FILE' 
READ(*,'(A)')FNAME1 
OPEN(6,FILE=FNAME1,ACCESS='SEQUENTIAL',STATUS='NEW') 
END IF 
IOUT = 6 
WRITE(*,463) 
463 FORMAT(///' ENTER THE NUMBER OF SIMULATION RUNS') 
READ(*,*),NSIM 
WRITE(*,9005) 
9005 FORMAT(///' ENTER MAXIMUM NUMBER OF THE FOLLOWING MACHINE THAT CAN' 
$' BE USED/ IN ANY LIFT AT THE SAME TIME: '/' 
$' DRILL, DOZER, TRUCK (OVERRIDE)', 
$' LOADER, TRUCK (COAL), LOADER ' 
READ(*,'(I6)')IRED(I),I=1,6) 
READ(3,'*(1EQ.0) ISEED,ISTOP,NBLOCK,NFILCL,NHOLLW,NEX,CTIM,CDEN,NSDY, 
+NDMH,CREV) 
IF(ISEED.EQ.0) ISEED=87 
IF(NFILCL.EQ.0) GO TO 17666 
WRITE(*,27654)
27654 FORMAT(//' ENTER THE SEQUENCE IN WHICH BACKFILL CELLS ARE TO BE FILLED')
READ(*,*) (SEQCL(I),I=1,NFILCL)

17666 CONTINUE
WRITE(*,687)
687 FORMAT(//') ENTER THE SEQUENCE IN WHICH BLOCKS ARE TO BE STRIPPED')
READ(*,*) (SEQBB(IB),IB=1,NBLOCK)
WRITE(*,*) ' ENTER NUMBER OF BREAK POINTS FOR OVERBURDEN REMOVAL'
WRITE(*,*) ' FROM EXTRA SPOIL STORAGE AREA TO FILL CELLS'
READ(*,*) NBREAK
WRITE(*,*) ' ENTER CORRESPONDING BLOCK NUMBERS'
READ(*,*) (NCEL(I),I=1,NBREAK)

924 CONTINUE

C INPUT DATA FOR DRILL

WRITE(*,*) ' EXECUTION STARTED, PLEASE WAIT'
PRDR(1)=1
PRDR(2)=2
PRDR(4)=2
PRDR(5)=1
PRDR(6)=3
PRDR(7)=2

THIS ROUTINE READS INPUT DATA THEN CALL SIMULT TO SIMULATE
READ(3,*) NROCK,(PFAC(I),DENS(I),SWELF(I),EFFECT(I),I=1,NROCK)
READ(3,*) NPDR,ISDR,DB,DS,NSPACE(1),TMOVE
IF (ISDR.EQ.2) THEN
NS=NSPACE(1)
ELSE
NS=PRDR(ISDR+1)
END IF
READ(3,*) (PRM(J),HIST(J),J=1,NS)
READ(3,*) NPBL,TRATE1,TRATE2,LTIME,DTIME,CONST,CAPB
READ(3,*) N1,N2,ISTR,NSPACE(3),TEX,DMP,HCTR,GVW,TOTR,GVC,TCTR,
+NHAUL
IF (ISTR.EQ.2) THEN
NS=NSPACE(3)
ELSE
NS=PRDR(ISTR+1)
END IF
READ(3,*) (PTR(J),HITR(J),J=1,NS)
IF (NHAUL.EQ.0) THEN
READ(3,*) NGEAR,DA,ROI,DGR,AMAX,VMAX
IF (DA.EQ.0) DA=.5
READ(3,*) (DHILL(I),I=1,DGR)
READ(3,*) (SPEED(J),RP(J),J=1,NGEAR)
END IF
READ(3,*) N2,NC2,IPSV,NSPACE(5),CAPL,CCPL,AVSPD
IF (IPSV.EQ.2) THEN
NS=NSPACE(5)
ELSE
NS=PRDR(IPSV+1)
END IF
READ(3,*) (PSV(J), HISV(J), CSV(J), HCSV(J), J=1, NS)
DO 1799 I=1, NBLCK
1799 READ(3,*)NLIFT(I), DAREA(I), CAREA(I), CTIK(I), PSBLK(I), PSLFT(I)
+ BPREP(I)
   READ(3,*) (TPREP(I), NST(I), NSTH(I), LNST(I), LNSTH(I), BNST(I), BNSTH(I)
+ ), I=1, NBLCK)
   DO 400 I=1, NBLCK
      BPREP(I)=BPREP(I)*60.
      TPREP(I)=TPREP(I)*60.
      KK=NLIFT(I)
      DO 400 J=1, KK
400 READ(3,*) NCLFT(I, J), PRBLK(I, J), PRLFT(I, J), ADEP(I, J)
+ FTIME(I, J), FTIME2(I, J)
   DO 401 I=1, NBLCK
      KK=NLIFT(I)
      DO 401 J=1, KK
401 READ(3,*) FBLAS(I, J), FDOZ(I, J), FRCD(I, J),
+ FHINH(I, J), (FROCK(I, J, K), K=1, NRCK)
   DO 4400 I=1, NBLCK
      KK=NLIFT(I)
4400 READ(3,*) (BCF(I, J), J=1, KK)
   CTSF=CTIM/60.
   TCOAL =0
IF (NFILCL .NE. 0) THEN
   DO 402 I=1, NFILCL
      READ(3,*) CFILCL(I), BTIME(I), NA, (NBFIL(I, J), J=1, NA)
      BTIME(I)=BTIME(I)*60.
402 NAFIL(I)=NA
IF (NHAUL .EQ. 1) THEN
   DO 403 I=1, NBLCK
403 READ(3,*) (GFIL(I, J, 1), FILL(I, J, 1), J=1, NFILCL)
   ELSE
   DO 404 I=1, NBLCK
      DO 404 J=1, NFILCL
         READ(3,*) NS, (GFIL(I, J, K), FILL(I, J, K), K=1, NS)
404 NSECFL(I, J)=NS
   END IF
END IF
DO 405 JJ=1, NBLCK
   NF=NLIFT(JJ)
DO 405 K=1, NF
IF (NCLFT(JJ, K).EQ. 1) THEN
   READ(3,*) NDCOD(JJ, K), MCDEST(JJ, K)
END IF
END IF
405 CONTINUE
IF (NEX .NE. 0) THEN
   READ(3,*) (GEXST(I), I=1, NEX)
   IF (NHAUL .EQ. 1) THEN
      DO 407 JJ=1, NBLCK
407 READ(3,*) (GEXST(JJ, K, 1), HEXST(JJ, K, 1), K=1, NEX)
   ELSE
   DO 408 JJ=1, NBLCK
      DO 408 K=1, NEX
         READ(3,*) NS, (GEXST(JJ, K, L), HEXST(JJ, K, L), L=1, NS)
408 NSECNX(JJ, K)=NS
SUBROUTINE SIMULT(NSIM,NBREAK)
!
THIS ROUTINE SIMULATES MINING ACTIVITIES. SUBROUTINE CALLED
BY THIS ROUTINE ARE MHL, TRUCK, BULDOZ, OVERB, DRILL, COMPUT
COMMON /BRES/DAREA(10), CAREA(10), CTIK(10), ADEP(10,5), FBLAS(10,5),
! CREVR, FDOZ(10,5), FNLH(10,5), PFAC(3), SWELF(5), ROCK(10,5,3),
! TODL, TOVL
COMMON /BTR/ BRFIL(10,5), FDFIL(10,5), NFILCL(10,5), NCODE(10,5),
! NCFL(10,5), NSCUS(10,5), VOLLEX(5),
+ NCLET(10,5), NLET(10,5), FHOLLW(10,5,5), HOLLW(10,5,5), GEXST(10,5,5),
+ NSCAL(10,5), NSCHW(10,5), CFILCL(10), MINDIS(10,5),
! TMINE, OBMINE, ROVBH, NOLLW, Nex, TVOLLW, VOLMAX, INDK,
! DIST, ICSV, ISTR, IPSV, PSV(20), HITR(20), PTR(20),
! PRSV, HCSV(20), CSV(20), PCSV, HCSV(20), AVSPD, FTIME(10,5),
! FTIME2(10,5), NSCEX(10,5), GFIL(10,10,5), NSECFL(10,10), NFILCL
!
END
FILE 9 TO 13 ARE OPENED TO WRITE EQUIPMENT SCHEDULING DATA.
At the end of the simulation they are written back to output file.
OPEN(9,FILE='DOZER.LIS',STATUS='NEW')
OPEN(10,FILE='DRILL.LIS',STATUS='NEW')
OPEN(11,FILE='TRUCK.LIS',STATUS='NEW')
OPEN(12,FILE='LOADER.LIS',STATUS='NEW')
OPEN(13,FILE='LOAD2.LIS',STATUS='NEW')
WRITE(9,*) DRILL
WRITE(10,*) DOZER
WRITE(11,*) TRUCK
C INITIALIZE ALL THE VARIABLES.

7702 IOUT=6
OBMINE=0
TOVL=0
TODL=0
DO 3343 J=1,5
3343 TFINISH(J)=0
DO 9998 I=1,NBLOCK
CITR(I)=0
CITH(I)=0
CAVTR(I)=0
CTDT(I)=0
CTLD(I)=0
CTTR(I)=0
KK=NLIFT(I)
DO 9998 J=1,KK
TRR(I,J)=0
TLT(I,J)=0
TDT(I,J)=0
TCCT(I,J)=0
DLCT(I,J)=0
RHT(I,J)=0
TFDR(I,J)=0
CDLI(I,J)=0
CLCT(I,J)=0
CCCT(I,J)=0
FTR(I,J)=0
CDLC(I,J)=0
TDRI(I,J)=0
DO 9998 K=1,8
9998 TSTART(I,J,K)=0
DO 9111 I=1,10
DO 9111 J=1,5
9111 TH(I,J)=0
COAL=0
C NMACH ARE THE NUMBER OF MACHINES AVAILABLE AT ANY TIME. THEY ARE
C INITIALIZED TO THE NUMBER OF MACHINES.
C
NMACH(1)=NPDR
NMACH(2)=NPBL
NMACH(3)=N1
NMACH(4)=NC1
NMACH(5)=N2
NMACH(6)=NC2
ACOST=0
TLITK=0
TTIM=0
ROVBHY=0
TVOLLW=0
DO 1711 J=1,5
NCBL(J)=2
VOLL(J)=0
CODE=0
ADR=0
ATR=0
ABUL=0
ASVL=0
ACRC=0
ACVLI=0
KSHLFT=0
TPBL=0
OBBL=0
PTBL=0
TBBL=0
TOVB=0
THL=0
TTBL=0
CILK=0
CILK=0
TNNATCH=0
CATCH=0
TCDR=0
CDR=0
TPCOAL=0
INDK=1
INEX=1
DO 231 I=1,NBLOCK
  KK=NLIT(I)
  DO 231 J=1,KK
    SLIFT(I,J)=2
    DO 231 K=1,7
      JOB(I,J,K)=2
      NUMB(I,J,K)=0
      NUMB(I,J,K+1)=0
      NUMB(I,J,K+2)=0
      TREQD(I,J,K)=0
      NCODE(I,J,K)=1
  231  DO 431 I=1,NFILCL
    BBTIME(I)=0
    NREADY(I)=1
    NCFCL(I)=2
    VFILCL(I)=CFILCL(I)
    NUMB(I)=0
    NCFL(I)=2
  431  IF(BTIME(I).EQ.0) NCFL(I)=0
  DO 433 I=1,NHOLLW
  433  VOLLWT(I)=CHOLLW(I)
  DO 435 I=1,NEX
  435  }
NNEX(I)=0
DO 435 I=1,10
DO 435 J=1,4
DO 435 K=1,3
435 NUMBER(I,J,K)=0
DO 9019 I=1,NBLOCK
KK=NLIFT(I)
DO 9019 J=1,KK
DO 9019 K=1,7 9019 CONTINUE
DO 731 I=1,NBLOCK
KK=NLIFT(I)
DO 731 J=1,KK
DO 731 K=1,7 731 IF(NCODE(I,J,K).EQ.0) JOB(I,J,K)=0
DO 8731 I=1,NBLOCK
KK=NLIFT(I)
DO 8731 J=1,KK
DO 8731 K=1,4 8731 WT(I,J,K)=0
13350 T=0 NDD=0 DO 1111 I=1,NBLOCK
CJOB(I)=2 1111 NFCODE(I)=2 CALL OVERBN(IOUT,TCOAL) 6666 TMINE=0
T=0 1223 CONTINUE
C FIND OUT IF THE BLOCK IS READY TO BE STRIPPED.
3271 DO 3255 IB=1,NBLOCK
IBLOCK=SEQBB(IB)
PSB=PSBLK(IBLOCK)
PSL=PSLFT(IBLOCK)
IF(PSB.EQ.0.AND. PSL.EQ.0) THEN
BSIGN=0 ELSE IF(PSL.GT.NLIFT(IBLOCK)) THEN
BSIGN=NFCODE(PSB)

ELSE
  BSIGN=SLIFT(PSB,PSL)
END IF
END IF

IF(BSIGN.NE.0) GO TO 3255
5432
  KK=NLIFT(IBLOCK)
  DO 3256 ILIFT=1, KK
  PRB=PRBLK(IBLOCK, ILIFT)
  PLF=PRLFT(IBLOCK, ILIFT)
  IF(PRB.EQ.0 .AND. PLF.EQ.0) THEN
    SIGN=0
  ELSE IF(PLF.GT.NLIFT(IBLOCK)) THEN
    SIGN=SLIFT(PRB, PLF)
  END IF
  IF(SIGN.NE.0) GO TO 3256
  ILIFT=ILIFT+1
END IF
IF(ILIFT.EQ.0) ILIFT=1
IF(DRILLING REQUIRED AND AND CAN BE STARTED THEN CALL DRILL
IF(NCODE(IBLOCK, ILIFT, 1).EQ.1 .AND. JOB(IBLOCK, ILIFT, 1).EQ.2 .+
  (SLIFT(IBLOCK, ILIFT), EQ.0 .OR. ILIFT.EQ.1)) THEN
  NSTART(IBLOCK, ILIFT, 1)=T+BPREP(IBLOCK)
  CALL DRILL
  WRITE(9, 1706) IBLOCK, ILIFT, NUMB(IBLOCK, ILIFT, 1), T/60.,
  + TREQD(IBLOCK, ILIFT, 1) + BPREP(IBLOCK)
END IF
IF(DOZER PUSH REQUIRED AND AND CAN BE STARTED THEN CALL MHDL
IF(NCODE(IBLOCK, ILIFT, 2).EQ.1 .AND. JOB(IBLOCK, ILIFT, 2).EQ.2 .+
  (SLIFT(IBLOCK, ILIFT), EQ.0 .OR. ILIFT.EQ.1)) THEN
  NSTART(IBLOCK, ILIFT, 2)=T+BPREP(IBLOCK)
  CALL MHDL
  WRITE(10, 1706) IBLOCK, ILIFT, NUMB(IBLOCK, ILIFT, 2), T/60., TREQD( +
  IBLOCK, ILIFT, 2) + BPREP(IBLOCK)
END IF
IF LOADER CARRY REQUIRED AND CAN BE STARTED THEN CALL MHDL
IF(NCODE(IBLOCK, ILIFT, 3). EQ. 1. AND. JOB(IBLOCK, ILIFT, 2). EQ. 0
+ AND. JOB(IBLOCK, ILIFT, 3). EQ. 2. AND. JOB(IBLOCK, ILIFT, 1). EQ. 0. AND.
+(SLIFT(IBLOCK, ILIFT). EQ. 0. OR. ILIFT. EQ. 1)) THEN
  IF(NMACH(4). NE. 0) THEN
    IF(NMACH(2). NE. 0. OR. FTIME(IBLOCK, ILIFT). EQ. 0) THEN
      IF(NMACH(4). GT. MRED(4)) NUMB(IBLOCK, ILIFT, 7) = MRED(4)
      IF(NMACH(4). LE. MRED(4)) NUMB(IBLOCK, ILIFT, 7) = NMACH(4)
      IF(FTIME(IBLOCK, ILIFT). NE. 0) THEN
        IF(NMACH(2). GT. MRED(2)) NUMB(IBLOCK, ILIFT, 8) = MRED(2)
        IF(NMACH(2). LE. MRED(2)) NUMB(IBLOCK, ILIFT, 8) = NMACH(2)
      END IF
      NMACH(2) = NMACH(2) - NUMB(IBLOCK, ILIFT, 8)
      CODE = 2. TSTART(IBLOCK, ILIFT, 3) = T
    END IF
    CALL MHD
    JOB(IBLOCK, ILIFT, 3) = 1
    FTR(IBLOCK, ILIFT) = TREQD(IBLOCK, ILIFT, 3)
    WRITE(12, 1706) IBLOCK, ILIFT, NUMB(IBLOCK, ILIFT, 7), T/60., +TREQD(IBLOCK, ILIFT, 3)/60., WCODE(4)
    IF(FTIME(IBLOCK, ILIFT). NE. 0) THEN
      JOB(IBLOCK, ILIFT, 6) = 1
      WRITE(10, 2706) IBLOCK, ILIFT, NUMB(IBLOCK, ILIFT, 8), T/60., +TREQD(IBLOCK, ILIFT, 6)/60., WCODE(6)
    END IF
  END IF
END IF
IF LOAD AND HAUL CAN BE STARTED THEN CALL MHD

IF(NCODE(IBLOCK, ILIFT, 4). EQ. 1. AND. JOB(IBLOCK, ILIFT, 3). EQ. 0
+ AND. JOB(IBLOCK, ILIFT, 2). EQ. 0. AND. JOB(IBLOCK, ILIFT, 4). EQ. 2. AND.
+(SLIFT(IBLOCK, ILIFT). EQ. 0. OR. ILIFT. EQ. 1)) THEN
  IF(NMACH(4). NE. 0. AND. NMACH(3). NE. 0) THEN
    IF(NMACH(2). NE. 0. OR. FTIME2(IBLOCK, ILIFT). EQ. 0. ) THEN
      IF(NMACH(4). GT. MRED(4)) NUMB(IBLOCK, ILIFT, 4) = MRED(4)
      IF(NMACH(4). LE. MRED(4)) NUMB(IBLOCK, ILIFT, 4) = NMACH(4)
      IF(FTIME2(IBLOCK, ILIFT). NE. 0.) THEN
        IF(NMACH(2). GT. MRED(2)) NUMB(IBLOCK, ILIFT, 9) = MRED(2)
        IF(NMACH(2). LE. MRED(2)) NUMB(IBLOCK, ILIFT, 9) = NMACH(2)
      END IF
      NMACH(2) = NMACH(2) - NUMB(IBLOCK, ILIFT, 9)
      CODE = 20. TSTART(IBLOCK, ILIFT, 4) = T
    END IF
  END IF
  NMACH(4) = NMACH(4) - NUMB(IBLOCK, ILIFT, 4)
  NMACH(3) = NMACH(3) - NUMB(IBLOCK, ILIFT, 3)
END IF
TTR(IBLOCK, ILIFT) = TREQD(IBLOCK, ILIFT, 4)
IF(FTIME2(IBLOCK, ILIFT). NE. 0.) THEN
  JOB(IBLOCK, ILIFT, 7) = 1
WRITE(10,1706)IBLOCK,ILIFT,NUMB(IBLOCK,ILIFT,9),T/60.,
+TREQD(IBLOCK,ILIFT,7)/60.,WCODE(7)
END IF
WRITE(11,1706)IBLOCK,ILIFT,NUMB(IBLOCK,ILIFT,3),T/60.,
+TREQD(IBLOCK,ILIFT,4)/60.,WCODE(3)
WRITE(12,1706)IBLOCK,ILIFT,NUMB(IBLOCK,ILIFT,4),T/60.,
+TREQD(IBLOCK,ILIFT,4)/60.,WCODE(4)
END IF
END IF
END IF
3256 CONTINUE
3255 CONTINUE

FIND OUT IF ANY BLOCK IS READY FOR COAL REMOVAL

DO 3268 IBLOCK=1,NBLOCK
  KK=NLIFT(IBLOCK)
  DO 3267 ILIFT=1,KK
    DO 3267 IN=1
      IF(IN.EQ.5) GO TO 3267
      IF(NCODE(IBLOCK,ILIFT,IN).EQ.0.OR.JOB(IBLOCK,ILIFT,IN).EQ.0)
        GO TO 3267
      GO TO 3268
  3267 CONTINUE
IF COAL IS READY TO BE REMOVED AND MACHINES AVAILABLE THEN CALL TRUCK
CWT(IBLOCK)=CDEN*CAREA(IBLOCK)*CTIK(IBLOCK)*CREVR
IF(CJOB(IBLOCK).EQ.2) THEN
  IF(NMACH(5).NE.0.AND.NMACH(6).NE.0) THEN
    ILIFT=1
    IF(NMACH(6).GT.MRED(6)) NUMB(IBLOCK,ILIFT,6)=MRED(6)
    IF(NMACH(6).LE.MRED(6)) NUMB(IBLOCK,ILIFT,6)=NMACH(6)
    NMACH(6)=NMACH(6)-NUMB(IBLOCK,ILIFT,6)
    IF(NMACH(5).GT.MRED(5)) NUMB(IBLOCK,ILIFT,5)=MRED(5)
    IF(NMACH(5).LE.MRED(5)) NUMB(IBLOCK,ILIFT,5)=NMACH(5)
    NMACH(5)=NMACH(5)-NUMB(IBLOCK,ILIFT,5)
    CODE = 1
    TSTART(IBLOCK,ILIFT,5)=T
    CALL TRUCK(NHaul,22,TRE,7,INDK,7)
    WRITE(13,1706)IBLOCK,ILIFT,NUMB(IBLOCK,1,5),T/60.,
    +TREQD(IBLOCK,1,5)/60.,WCODE(5)
  TFIX=TREQD(IBLOCK,1,5)
  END IF
END IF
3268 CONTINUE

FIND OUT IF ANY BACKFILL CELL IS READY TO BE GRADED.

IF(NMACH(2).NE.0) THEN
  DO 8887 IF=1,NFILCL
    IF(NREADY(IF).EQ.0.AND.NCFL(IF).EQ.2) THEN
      NILCEL =IF
      GO TO 8873
    END IF
  8887 CONTINUE

END IF
CONTINUE
GO TO 8878
8873 IF(NMACH(2).GT.MRED(2)) THEN
    NUMBG(NILCEL)=MRED(2)
ELSE
    NUMBG(NILCEL)=NMACH(2)
END IF
NMACH(2)=NMACH(2)-NUMBG(NILCEL)
CALL BULDOZ(2,NILCEL,1,1)
BBTIME(NILCEL)=BTIME(NILCEL)
TSTART(NILCEL,1,7)=T
NCFIL(NILCEL)=1
WRITE(10,1707)NILEL,NUMBG(NILCEL),T/60, BBTIME(NILCEL)/60.,WCODE(8)
1707 FORMAT('BF',12,12X,12,5X,F8.2,1X,F8.2,6X,A25)
8878 CONTINUE
END IF

FIND IF SPOIL IS TO BE REMOVED FROM TEMPORARY STORAGE AREA.

IF(NEX.NE.0) THEN
DO 9701 I=1,NBREAK
  J=NCEL(I)
  IF(NFCODE(J).EQ.0) THEN
    DO 5679 K=1,NEX
      IF(NNEX(K).EQ.0) GO TO 5679
      MZ=K
      GO TO 5680
    CONTINUE
  GO TO 9702
5679 CONTINUE
GO TO 9702
5680 DO 5682 I=1,NFILCL
  K=SEQCL(I)
  IF(NFILCL(K).EQ.0) GO TO 5682
  NM=NAFIL(K)
  IF(NM.EQ.0) THEN
    NCFIL(K)=1
    GO TO 5688
  ELSE
    DO 5677 NNM=1,NM
      IS=NFIL(K,NNM)
      IF(NFCODE(IS).EQ.0) GO TO 5677
      NCFIL(K)=2
      GO TO 5177
    CONTINUE
    NCFIL(K)=1
    GO TO 5177
  END IF
5677 CONTINUE
5682 CONTINUE
5688 NCFIL(K)=1
5177 IF(NFILCL(K).EQ.1) THEN
  IF(NMACH(4).EQ.0.OR.NMACH(3).EQ.0) GO TO 9702
  IF(NMACH(4).GT.MRED(4))NUMBF(MZ,2)=MRED(4)
  IF(NMACH(4).LE.MRED(4))NUMBF(MZ,2)=NMACH(4)
  IF(NMACH(3).GT.MRED(3))NUMBF(MZ,1)=MRED(3)
IF(NMACH(3).LE.MRED(3)) NUMBF(MZ,1)=NMACH(3)
NMACH(4)=NMACH(4)-NUMBF(MZ,2)
NMACH(3)=NMACH(3)-NUMBF(MZ,1)
CODE = 3
CALL MHDL
NCBL(MZ)=1
WRITE(11,1708)MZ,NUMBF(MZ,1),T/60.,ETIME(MZ)/60.,WCODE(3)
1708 FORMAT('ES',12,12X,12,5X,F8.2,1X,F8.2,6X,A25)
WRITE(12,1708)MZ,NUMBF(MZ,2),T/60.,ETIME(MZ)/60.,WCODE(4)
END IF
END IF

9701 CONTINUE
END IF

9702 J=1

CC FIND NEXT EVENT

AMINTM = 999999.
DO 3262 IBLOCK=1,NBLOCK
KK=NLIFT(IBLOCK)
DO 3262 ILIFT=1,KK
DO 3262 KN=1,J
IF(KN.NE.5) THEN
  IF(JOB(IBLOCK,ILIFT,KN).EQ.1) THEN
    IF(AMINTM.BT.TREQLD(IBLOCK,ILIFT,KN))
      INTM=TREQLD(IBLOCK,ILIFT,KN)
    END IF
  END IF
3262 CONTINUE
3299 DO 4299 IBLOCK=1,NBLOCK
IF(CJOB(IBLOCK).EQ.1) THEN
  IF(AMINTM.GT.TREQD(IBLOCK,1,5)) AMINTM=TREQD(IBLOCK,1,5)
END IF
4299 CONTINUE

DO 8885 IE=1,NFILE
IF(NCFL(IE).EQ.1) THEN
  IF(AMINTM.GT.BBTIME(IE)) AMINTM=BBTIME(IE)
END IF
8885 CONTINUE

DO 8886 IE=1,NEX
IF(NCBL(IE).EQ.1) THEN
  IF(AMINTM.GT.ETIME(IE)) AMINTM=ETIME(IE)
END IF
8886 CONTINUE

UPDATE ATTRIBUTES

DO 3263 IBLOCK=1,NBLOCK
MK=NLIFT(IBLOCK)
DO 4563 ILIFT=1,MK
DO 4563 KK=1,J
IF(KK.EQ.5) GO TO 4563
IF(JOB(IBLOCK,ILIFT,KK).EQ.1) THEN
  IF(TREQD(IBLOCK,ILIFT,KK).EQ.AMINTM) GO TO 3264
TREQD( IBLOCK, ILIFT, KK) = TREQD( IBLOCK, ILIFT, KK) - AMINTM

GO TO 4563

3264 JOB( IBLOCK, ILIFT, KK) = 0
IF( KK .LT. 6) THEN
  IF( KK .EQ. 5) THEN
    NMACH( KK) = NMACH( KK) + NUMB( IBLOCK, ILIFT, 7)
  ELSE
    NMACH( KK) = NMACH( KK) + NUMB( IBLOCK, ILIFT, KK)
  END IF
ELSE
  IF( KK .EQ. 3) THEN
    NMACH( 3) = NMACH( 3) + NUMB( IBLOCK, ILIFT, 3)
  ELSE IF( KK .EQ. 6) THEN
    NMACH( 2) = NMACH( 2) + NUMB( IBLOCK, ILIFT, 8)
  ELSE
    NMACH( 2) = NMACH( 2) + NUMB( IBLOCK, ILIFT, 9)
  END IF
END IF

TREQD( IBLOCK, ILIFT, KK) = TREQD( IBLOCK, ILIFT, KK) - AMINTM
END IF

4563 CONTINUE
3278 IF( CJOB( IBLOCK). EQ. 1) THEN
  IF( TREQD( IBLOCK, 1, 5) .EQ. AMINTM) THEN
    TMINE = CWT( IBLOCK) * AMINTM / TFIX
    CJOB( IBLOCK) = 0.
  END IF
  NMACH( 6) = NMACH( 6) + NUMB( IBLOCK, 1, 6)
  NMACH( 5) = NMACH( 5) + NUMB( IBLOCK, 1, 5)
  TREQD( IBLOCK, 1, 5) = TREQD( IBLOCK, 1, 5) - AMINTM
ELSE
  TREQD( IBLOCK, 1, 5) = TREQD( IBLOCK, 1, 5) - AMINTM
END IF

3291 CONTINUE
3263 CONTINUE
4004 CONTINUE
DO 8883 IE = 1, NFILC
  IF( NCFL( IE). EQ. 1) THEN
    IF( BBTIME( IE). LE. AMINTM) THEN
      NMACH( 2) = NMACH( 2) + NUMB( IE)
      NCFL( IE) = 0.
    END IF
    BBTIME( IE) = BBTIME( IE) - AMINTM
  END IF
8883 CONTINUE
DO 9883 IE = 1, NEX
  IF( NCBL( IE). EQ. 1) THEN
    IF( ETIME( IE). LE. AMINTM) THEN
      NMACH( 3) = NMACH( 3) + NUMBF( IE, 1)
      NMACH( 4) = NMACH( 4) + NUMBF( IE, 2)
      NCBL( IE) = 0.
    END IF
    ETIME( IE) = ETIME( IE) - AMINTM
  END IF
9883 CONTINUE
3292 NI1=0
7271 DO 5477 I=1,NBLOCK
   KB=NLIFT(I)
   DO 5477 JJ=1,KB
   DO 5478 K=1,7
   IF(K.EQ.5) GO TO 5478
   IF(JOB(I,JJ,K).NE.0) GO TO 5477
   5478 CONTINUE
   SLIFT(I,JJ)=0
5477 CONTINUE
   DO 8881 I=1,NFILEL
   IF(VFILEL(I).EQ.0.) GO TO 8882
   GO TO 8881
8882 NREADY(I)=0
8881 CONTINUE
   DO 8777 I=1,NBLOCK
   KB=NLIFT(I)
   DO 8778 JJ=1,KB
   IF(SLIFT(I,JJ).NE.0) GO TO 8777
   8778 CONTINUE
   IF(CJOB(I).EQ.0.) NFCODE(I)=0
8777 CONTINUE
   DO 7002 I=1,NBLOCK
   JJ=NLIFT(I)
   DO 7002 J=1, JJ
   IF(NFCODE(I,J).EQ.0) GO TO 7002
   GO TO 1223
7002 CONTINUE
   IF(cncf(i).eq.0.) go to 9013
   go to 1223
9013 CONTINUE
6002 TPCOA L=TPCOAL +TIME
   KSHIFT=INT(T/CTIM)
   CLOSE(9)
   CLOSE(10)
   CLOSE(11)
   CLOSE(12)
   CLOSE(13)
   CLOSE(14)
   OPEN(9,FILE='DRILL.LIS',STATUS='OLD')
   OPEN(10,FILE='DOZER.LIS',STATUS='OLD')
   OPEN(11,FILE='TRUCK.LIS',STATUS='OLD')
   OPEN(12,FILE='LOADER.LIS',STATUS='OLD')
   OPEN(13,FILE='LOADER2.LIS',STATUS='OLD')
   IF(N1.NE.0) THEN
      ATR=ATR/T/N1
      TITK=TITK/N1
   END IF
   IF(N2.NE.0) THEN
      ASVL=ASVL/T/N2
   END IF
(TLTK = TLTK/N2)
END IF
IF(NC2 .NE. 0) THEN
ACVL = ACVL/T/NC2
CTITK = CTITK/NC2
END IF
IF(NPDR .NE. 0) THEN
ADR = ADR/NC/TPDR
TCDR = TCDR/TP/TPDR
END IF
IF(NPBL .NE. 0) THEN
ABUL = ABUL/T/TPBL
TTLBL = TBL/TP/TPBL
END IF
IF(NCI .NE. 0) ACR = ACR/NC/TC
CALL PRINTF(IOUT, NSIM, ISIM, KSHIFT, TCOAL, T)
CONTINUE
RETURN
END SUBROUTINE PRINTF(IOUT, NSIM, ISIM, KSHIFT, TCOAL, T)
COMMON/BRES/DAREA(10), CAREA(10), CTIT(10), ADEP(10,5), FBLAS(10,5), + CREVR, FDOP(10,5), FLNH(10,5), PFAC(3), SWELF(3), FROCK(10,5,3) + TOBL, TOVL
COMMON/BTR/T, DMP, GW, SPEED(10), RP(10), HCTR, TCTR, HDIS(10,5) + RORI, NGEAR, NC1, CMATCH, TMATCH, NG2, GVC, DA, AMAX, VMAX + CWT(20), DHI1(10), TOTR, N1, ATR, ASVL, ACR, ACVL, CHD(10) + LHDC, N2, CCPL, CAPL, NNTR, GTOMVB, CBHL, CTIT + K, NTR, TOVL, THT, TITK, SCYCLE, WMATL, HTIME + TIME, CYCLE, RHT(10,5), CRHT(20), TH(IH(10,5), CTHT(10), TLT + (10,5), TDT(10,5), CTLD(10), CTT(10), TCDT(10), WTR(10,5) + TTR(10,5), CCTR(10), CAPE(10), CCT(10) + TCT(10,5), DLCT(20,5), CDLC(10,5), CLCT(10,5), CCLT(10,5) COMMON/BTR/NAFIL(10), NBFIL(10,5), NFCODE(10), NCFCL(10), SEQCL(10), + VOLL(5), CHOLLW(5), CECS(5), VOLLEX(5), + NCLF(10,5), FILL(10,10,5) + HEXT(10,5,5), GHOLLW(10,5,5), HHOLLW(10,5,5), GEXST + (10,5,5), NSECAL(10,5), NSECWL(10,5), CFILCL(10), MINDIS, + TMINE, OBVBN, HHOVLW, NEX, TOLNL, VOLMAX, INDK + CONTINUE
COMMON/GEN/MRED(6), NCODE(10,5,8), J6B(10,5,8), CAJOB(10,5,5) COMMON/BNEW/NCBL, NCFL(10), NCFL(5,10), FGFIL(10,5), FGFIL(10,5) + CGVOLL(10), NENCEF(5,10), NREADY(10), TTI(5,10) + NUFF(5,2), VFILCL(10), EFHT(5), FRHT(5), FDT(5), FDLC(5), FTTL(5) + FWTR(5), FDIS(5), INX, FITC(5), VOLL(5), FILT(5), FHT(5) COMMON/DNEW/LNSTH(10), NNSTH(10), BSUMT(10,10,4), BFROM(10,10,4) + BFROM(10,10,4), LNFROM(10,10,4), AMOUNT(10,10,4), NUMBER(10,4,3), TSTART(10,5,8) + TFINISH(5), SDL(10,5), CMOUNT(10,10,4), BCFROM(10,10,4) + LCFROM(10,10,4), TCOSD(5), TCOSB(5), NDCOD(10,5), MHCDEST(10,5)
COMMON/BDR/DB, DS, NPDR, ADR, HIST(20), PRM(20), ISDR, TFDR(10,5), TTDR + (10,5), CCDR, TTCB, TMOV
COMMON/BBUL/NPBL, BCF(10,5), ABUL, HIBL(20), PFBL(20), PBBL(20), ISBL + , TDZ(10,5), OBBL, TBLT, GTIME(10), NUMB(10), BTIME(10)
COMMON/ION/PSBLK(10), PSLFT(10), PBRLK(10,5), PRLFT(10,5), SEQBB(10) + , JLIFT(10), BPREP(10), CDEN, NHAUL, CTSF, NSDY, NDMH, NCDZ(10,5), + NNDZ(10,5), RATE1, RATE2, CONST, LTIM, DTIME, CAPB
INTEGER PSBLK, PSLFT, PBRLK, PRLFT, SEQBB, JLIFT, PRDR(7), PRFL, PRTR + INTEGER SECTION, CCODE, GJOB, PCSV, PTRC, PCFL, PBRL, BCFROM, LCFROM, + BSNTH, FREE(8), SLIFT, ITIM, START, DGR, BFROM, BBFROM, BNSY
REAL LTIM, NNX
CHARACTER*64 FNAME
COMMON/BCST/IC, N1, TH, WORKT, CFUEL(6), CIIT(6), DEPPH(6), REPAIR(6), + CTIRE(6), CUND(6), TCOPER(6), SPLITM(6), TAHP, TCOST, NOP, OPCOST(6), + COPT, WCOST(6)
COMMON/BEXTRA/ZNHP, ZSAL, NHP, WEL, OBFT, EXRECL, TAREA, CSEED, + DCP, WHFT, PRMCST, AFCST, SLCOST, DTCST, BLCST, PRMRT
COMMON/BEXTRA/FR(10,5), BBTIME(10)
DIMENSION TTAKEN(5)
CHARACTER*60 FMT(5)
CHARACTER*80 LINE
FMT(1):='(1X,' ' BLOCK NO.'',2X,I2,3X,1(1X,I6),': TOTAL')
FMT(2):='(1X,' ' BLOCK NO.'',2X,I2,3X,2(1X,I6),': TOTAL')
FMT(3):='(1X,' ' BLOCK NO.'',2X,I2,3X,3(1X,I6),': TOTAL')
FMT(4):='(1X,' ' BLOCK NO.'',2X,I2,3X,4(1X,I6),': TOTAL')
FMT(5):='(1X,' ' BLOCK NO.'',2X,I2,3X,5(1X,I6),': TOTAL')
CALL COMPUT(NREAD, T, ADR, AVSL, APL, ATR, ABUL, ACR, ACVL, KSHIFT, NPDR, + NPBL, FBLAS, PFAC, ADEP, TPCOAL, FROCK, IOUT, N1, N2, TPCOAL, DB, DS, DAREA, + NC1, NC2, NROCK, AGAST, TCDR, CCOR, NDR, CTOVB, CHL, CTITX, NTR, TOVB, THL, + NBL, TPVB, PBBL, PSLFT, OBBL, TDZ, ROVB, WCOST, CMATCH, CMATCH, ISIM, + NC2, NROCK, NEFT, CHCOST, CTSF, NSDY, NDMH, NC2, TDL, TOVL)
IF(ISIM.EQ.1) THEN
  WRITE(IOUT,'*')
  WRITE(IOUT,*) ONSIM SIMULATION SUMMARY
  WRITE(IOUT,9393)FNAME
9393 FORMAT(//' INPUT FILE: ',A64)
END IF
WRITE(IOUT,999)ISIM, NSIM
999 FORMAT(//' SIMULATION #', I2, ' OF ', I2)'/'
WRITE(IOUT,992)
992 FORMAT(_OPERATION AND COST SUMMARY, BY BLOCK (OPERATING COST ONLY
+ *)')
DO 132 IB=1, NBLOCK
IF (IB.EQ.1) WRITE(IOUT,131)NBLOCK
IL=NLIPT(IB)
BVOL=0
DO 988 I=1, IL
AVSWL=0
DO 988 IB=1, IL
BVOL=AVSWL+ADEP(IB, I)*DAREA(IB)/27.
DO 988 I=1, IL
BVOL=AVSWL+SWELF(IROCK)*FROCK(IB, I, IROCK)*ADEP(IB, I)*DAREA(IB)/27
AVSWL=AVSWL/BVOL
BLOSWL=BVOL*AVSWL
SRATIO=BVOL/CWT(IB)
WRITE(IOUT,133)IB,NLIFT(IB)
WRITE(IOUT,990)BVOL,AVSWL,BLOSWL,SRATIO
+ FORMAT( ' BLOCK VOLUME (BCY) = '',F8.0/
+ AVERAGE SWELL = '',F8.2/
+ BLOCK VOLUME (LCY) = '',F8.0/
+ STRIPPING RATIO (BCY/TON) = '',F8.2)
+ ILIF=NLIFT(IB)
DO 941 IC=1,ILIF
941 JLIFT(IC)=IC
C
WRITE(IOUT,3678)
WRITE(IOUT,1230)
WRITE(IOUT,FMT(ILIFT))IB,(JLIFT(IC),IC=1,ILIF)
WRITE(IOUT,1230)
WRITE(IOUT,676)
TOTAL=0
DO 791 I=1,ILIF
791 TOTAL=TOTAL+WT(IB,I,1)
TOTAL=TOTAL/BLOSWL
WRITE(IOUT,677)(INT(FBLAS(IB,IC)*100),IC=1,ILIF),INT(TOTAL*100+.5)
TOTAL=0
DO 792 I=1,ILIF
792 TOTAL=TOTAL+TDTR(IB,I)
WRITE(IOUT,917)(INT(TDTR(IB,IC)),IC=1,ILIF),INT(TOTAL)
DO 9162 I=1,ILIF
9162 TDTR(IB,I)=TSTART(IB,I,1)+TDTR(IB,I)
WRITE(IOUT,1381)(TSTART(IB,IC,1)/60.,IC=1,ILIF)
WRITE(IOUT,1382)(TDTR(IC)/60.,IC=1,ILIF)
TOTAL=0
DO 793 I=1,ILIF
793 TOTAL=TOTAL+TDTR(IB,I)
WRITE(IOUT,918)(TDTR(IB,IC)/60.,IC=1,ILIF),TOTAL/60.
TOTAL=0
TOTAL=0
BVOL=0
DO 1794 I=1,ILIF
1794 TCOSD(I)=TDTR(IB,I)/60.*OPCST(1)*NUMB(IB,I,1)
TOTAL=TOTAL+TCOSD(I)
TCOSD(IB,I)=TCOSD(I)/WT(IB,I,1)
BVOL=BVOL+WT(IB,I,1)
ELSE
TOTAL1=TOTAL1+TCOSD(I)
IF(BVOL.EQ.0.) THEN
TOTAL1=0.
ELSE
TOTAL1=TOTAL1/BVOL
END IF
DO 9794 I=1,ILIF
9794 IF(TCOSD(I).EQ.0.) TCOSD(I)=0.
WRITE(IOUT,982)(TCOSD(I),I=1,ILIF),TOTAL1
WRITE(IOUT,981)(TCOSD(I),I=1,ILIF),TOTAL1
982 FORMAT( 'COST ($)',7X,6(1X,F7.1))
981 FORMAT( 'COST ($/LCY)',5X,6(3X,F4.2,1X))
WRITE(IOUT,678)
TOTAL1=0
DO 1796 I=1,ILIF
  TCOSD(I)=FTR(IB,I)/60.*OPCOST(3)*NUMB(IB,I,7)
  TOTAL=TOTAL+TCOSD(I)
  BVOL=BVOL+WT(IB,I,3)
  TCOSB(I)=TCOSD(I)/WT(IB,I,3)
1796 TOTAL=TOTAL1+TCOSD(I)
IF (BVOL.EQ.0.) THEN
  TOTAL1=0.
ELSE
  TOTAL1=TOTAL1/BVOL
END IF
DO 8797 I=1,ILIF
  IF(TCOSD(I).EQ.0.) TCOSB(I)=0.
  WRITE (IOUT,982)(TCOSD(I),I=1,ILIF),TOTAL
  WRITE (IOUT,981)(TCOSB(I),I=1,ILIF),TOTAL1
8797 FORMAT('/ LOADER CARRYING OVERBURDEN ')
WRITE(IOU1,6921)
6921 FORMAT('/ DOZER FEEDING OVERBURDEN ')
DO 6990 IC=1,ILIF
  IF(FTIME(IB,IC).EQ.0.) TSTART(IB,IC,3)=0.
  WRITE(IOUT,1381)(TSTART(IB,IC,3)+FTIME(IB,IC)*FTR(IB,IC))
  WRITE(IOUT,1382)(TFINIS(IC)/60.,IC=1,ILIF)
  TOTAL2=0
DO 6971 I=1,ILIF
  TTAKEN(I)=FTIME(IB,IC)*FTR(IB,IC)
6971 TOTAL2=TOTAL2+FTIME(IB,IC)*FTR(IB,IC)
WRITE(IOUT,1997)(TTAKEN(IC)/60.,IC=1,ILIF),TOTAL2/60.
TOTAL=0
TOTAL1=0
DO 6795 I=1,ILIF
  TCOSD(I)=FTR(IB,I)/60.*OPCOST(2)*FTIME(IB,I)*NUMB(IB,I,8)
  TOTAL=TOTAL+TCOSD(I)
  BVOL=BVOL+WT(IB,I,3)*FTIME(IB,I)
  TCOSB(I)=TCOSD(I)/WT(IB,I,3)/FTIME(IB,I)
6795 TOTAL1=TOTAL1+TCOSD(I)
IF (BVOL.EQ.0.) THEN
  TOTAL1=0.
ELSE
  TOTAL1=TOTAL1/BVOL
END IF
DO 9988 I=1,ILIF
  IF(TCOSD(I).EQ.0.) TCOSB(I)=0.
  WRITE (IOUT,982)(TCOSD(I),I=1,ILIF),TOTAL
  WRITE (IOUT,981)(TCOSB(I),I=1,ILIF),TOTAL1
9988 FORMAT('/ TRAVEL TIME (HR) ')
4681 FORMAT(3X,'TRAVEL TIME (MIN)',5(F6.1,1X))
5681 FORMAT(3X,'TRAVEL TIME (MCY)',1X,5(F6.1,1X))
6681 FORMAT(3X,'CARRIED (%)',3X,5(F6.1,1X))
4676 FORMAT(3X,'CARRIED (LCY)' ,1X,5(F6.1,1X))
4481 FORMAT(3X,'CARRIED (H) ',1X,5(F6.1,1X))
WRITE(IOUT,680)
TOTAL=0
TOT=0
TOTAL1=0
DO 2795 I=1, ILIF
   TOTAL=TOTAL+WT(IB, I, 4)
   TOT=TOT+WT(IB, I, 4)
2795 TOTAL=TOTAL+TTR(IB, I)
TOTAL=TOTAL+BLOWL
WRITE(IOUT, 681)((INT(FINH(IB, IC)*100), IC=1, ILIF), INT(TOTAL*100+.5)
WRITE(IOUT, 1681)((WT(IB, IC, 4)), IC=1, ILIF), INT(TOT)
DO 9164 I=1, ILIF
9164 TFINISH(IB)=TSTART(IB, I, 4)+TTR(IB, I)
WRITE(IOUT, 1381)(TSTART(IB, IC, 4)/60, IC=1, ILIF)
WRITE(IOUT, 1382)(TFINISH(IC)/60, IC=1, ILIF)
WRITE(IOUT, 2681)(TTR(IB, IC)/60, IC=1, ILIF), TOTAL1/60.
WRITE(IOUT, 1676) (DLCT(IB, IC), IC=1, ILIF)
WRITE(IOUT, 481)(DLT(IC, IC), IC=1, ILIF)
WRITE(IOUT, 1081)(THT(IB, IC), IC=1, ILIF)
WRITE(IOUT, 1091)(RHT(IB, IC), IC=1, ILIF)
WRITE(IOUT, 1082)(WTR(IB, IC), IC=1, ILIF)
WRITE(IOUT, 1083)(TDT(IB, IC), IC=1, ILIF)
WRITE(IOUT, 1084)(TCCT(IB, IC), IC=1, ILIF)
TOTAL=TOTAL1
TOTAL1=0
BVAL=0
DO 3795 I=1, ILIF
   TCOSD(IB)=TTR(IB, I, 4)/60.*(OPCOST(3)*NUMB(IB, I, 3)
   +OPCOST(4)*NUMB(IB, I, 4))
   TOTAL=TOTAL+TCOSD(IB)
   BVAL=BVAL+WT(IB, I, 4)
   TCOSB(IB)=TCOSD(IB)/WT(IB, I, 4)
3795 TOTAL=TOTAL+TCOSD(IB)
IF(BVAL.EQ.0.) THEN
   TOTAL1=0
ELSE
   TOTAL1=TOTAL1/BVAL
END IF
DO 9796 I=1, ILIF
9796 IF(TCOSD(IB).EQ.0.) TCOSB(IB)=0.
WRITE(IOUT, 982)(TCOSD(IB), I=1, ILIF), TOTAL
WRITE(IOUT, 981)(TCOSB(IB), I=1, ILIF), TOTAL1
129 CONTINUE
WRITE(IOUT, 921)
TOTAL=0
TOTAL1=0
TOTAL2=0
921 FORMAT(’DOZER FEEDING OVERBURDEN’,
+’TO LOADER’)
DO 1990 IC=1, ILIF
1990 TFIN(IS(IC))=TSTART(IB, IC, 4)+FTIME2(IB, IC)*TTR(IB, IC)
WRITE(IOUT, 1382)(TSTART(IB, IC, 4)/60, IC=1, ILIF)
WRITE(IOUT, 1385)(TFIN(IS(IC))/60, IC=1, ILIF)
DO 2971 I=1, ILIF
2971 TTAKEN(I)=FTIME2(IB, I)*TTR(IB, I)
2971 TOTAL2=TOTAL2+FTIME2(IB, I)*TTR(IB, I)
WRITE(IOUT, 1997)(TAKEN(IC)/60, IC=1, ILIF), TOTAL2/60.
1997 FORMAT( 'TIME TAKEN (HR)' , 1X, 6(1X, F6.1, 1X))
TOTAL=0
TOTAL1=0
BVOL=0
DO 4795 I=1, ILIF
TCOSD(I)=TTR(IB, I)/60. * OPCOST(2)*FTIME2(IB, I)* NUMB(IB, I, 9)
TOTAL=TOTAL+TCOSD(I)
TCOSB(I)=TCOSD(I)/WT(IB, I, 4)/FTIME2(IB, I)
BVOL=BVOL+WT(IB, I, 4)*FTIME2(IB, I)
4795 TOTAL=TOTAL+TCOSD(I)
   IF (BVOL.EQ.0.) THEN
      TOTAL1=0
   ELSE
      TOTAL1=TOTAL1/BVOL
   END IF
DO 9975 I=1, ILIF
9975 IF(TCOSD(I).EQ.0.) TCOSB(I)=0
   WRITE(IOUT, 982)(TCOSD(I), I=1, ILIF), TOTAL
WRITE(IOUT, 981)(TCOSB(I), I=1, ILIF), TOTAL1
TOTAL=0
TOTAL1=0
WRITE(IOUT, 3591)
WRITE(IOUT, 5921) INT(CWT(IB)) ,AIR(IB)/60.
WRITE(IOUT, 438) 'D R I L L I N G A N D B L A S T I N G ')
3678 FORMAT( 'L I F T N U M B E R ')
133 FORMAT( 'TOTAL NUMBER OF BLOCKS' , I2)
438 FORMAT( 'T I M E T A K E N ( H R ) ' , 5(F6.1, 1X))
1281 CONTINUE
WRITE(IOUT,11376)
DO 6375 I=1,NFILLCL
K=NUMBER(I,2,2)
6375 WRITE(IOUT,3375)I,(BBFROM(I,J,2),LLFROM(I,J,2),BMOUNT(I,J,2),
+J=1,K)
WRITE(IOUT,11377)
DO 7375 I=1,NFILLCL
K=NUMBER(I,2,3)
7375 WRITE(IOUT,3375)I,(BCFROM(I,J,2),LCFROM(I,J,2),CMOUNT(I,J,2),
+J=1,K)
3375 FORMAT(' ',I2,ucher,'$
+I4,6X,I2,11X,F7.0
+I8,6X,I2,11X,F7.0))
END IF
IF(NHOLLW.NE.0) THEN WRITE(IOUT,1377)
1377 FORMAT(1X,EXTRA SPOIL DISPOSAL AREA '/
1X,NO. CAPACITY VOLUME LEFT VOLUME RECEIVED FROM AMOUNT' /
1X,UNFILLED BLOCK LIFT ( LCY '"'
WRITE(IOUT,11375)
DO 4375 I=1,NHOLLW
K=NUMBER(I,3,1)
4375 WRITE(IOUT,1373)I,CHOLLW(I),VOLLFT(I),(BFROM(I,J,3),LFROM(I,J +3),AMOUNT(I,J,3),J=1,K)
WRITE(IOUT,11376)
DO 5375 I=1,NHOLLW
K=NUMBER(I,3,2)
5375 WRITE(IOUT,3375)I,(BBFROM(I,J,3),LLFROM(I,J,3),BMOUNT(I,J,3),
+J=1,K)
WRITE(IOUT,11377)
DO 8375 I=1,NHOLLW
K=NUMBER(I,3,3)
8375 WRITE(IOUT,3375)I,(BCFROM(I,J,3),LCFROM(I,J,3),CMOUNT(I,J,3),
+J=1,K)
END IF
IF(NEX.NE.0) THEN WRITE(IOUT,3377)
3377 FORMAT(1X,EXTRA SPOIL STORAGE AREA '/
1X,NO. CAPACITY VOLUME LEFT VOLUME RECEIVED FROM AMOUNT' /
1X,UNFILLED BLOCK LIFT ( LCY '
WRITE(IOUT,11375)
DO 7376 I=1,NEX
K=NUMBER(I,4,1)
7376 WRITE(IOUT,1373)I,CEXT(I),VOLLEX(I),(BFROM(I,J,4),LFROM(I,J +4),AMOUNT(I,J,4),J=1,K)
WRITE(IOUT,11376)
DO 8376 I=1,NEX
K=NUMBER(I,4,2)
8376 WRITE(IOUT,3375)I,(BBFROM(I,J,4),LLFROM(I,J,4),BMOUNT(I,J,4), +J=1,K)
WRITE(IOUT,11377)
DO 8377 I=1,NEX
K=NUMBER(I,4,3)
8377 WRITE(IOUT,3375)I,(BCFROM(I,J,4),LCFROM(I,J,4),CMOUNT(I,J,4), +J=1,K)
END IF
WRITE(IOUT,1230)
OPEN(9,FILE='DRILL.LIS',STATUS='OLD')
OPEN(10,FILE='DOZER.LIS',STATUS='OLD')
OPEN(11,FILE='TRUCK.LIS',STATUS='OLD')
OPEN(12,FILE='LOADER.LIS',STATUS='OLD')
OPEN(13,FILE='LOADER2.LIS',STATUS='OLD')
WRITE(IOUT,3988)
3988 FORMAT(/,'SCHEDULING OF EQUIPMENT : {
+
--------------------------
)
179 READ(9,198,END=30)LINE
198 FORMAT(A80)
WRITE(IOUT,198)LINE
GO TO 179
30 WRITE(IOUT,9978)
9978 FORMAT(/)
945 READ(10,198,END=31)LINE
WRITE(IOUT,198)LINE
GO TO 945
31 WRITE(IOUT,9978)
946 READ(11,198,END=32)LINE
WRITE(IOUT,198)LINE
GO TO 946
32 WRITE(IOUT,9978)
947 READ(12,198,END=33)LINE
WRITE(IOUT,198)LINE
GO TO 947
33 WRITE(IOUT,9978)
948 READ(13,198,END=34)LINE
WRITE(IOUT,198)LINE
GO TO 948
34 WRITE(IOUT,3742)
3742 FORMAT(/,15X,'SUMMARY OF OPERATIONS AND COSTS',/15X,'--------------------------',/15X,' OPERATIONS',/15X,'--------------------------',/15X,' TOTAL COAL MINED (TONS)',27X,18
RMINE=0
AVSWL=0
DO 9017 I=1,NBLOCK
KKK=NLIFT(I)
DO 9017 J=1,KKK
RMINE=RMINE+ADEP(I,J)*DAREA(I)/27.
DO 9017 K=1,NROCK
9017 AVSWL=AVSWL+ROCK(I,J,K)*SWELF(K)*ADEP(I,J)*DAREA(I)
AVSWL=AVSWL/RMINE/27.
STIP = RMINE/TCOAL
WRITE(IOUT,3211)INT(TCOAL),INT(ORBINE),INT(RMINE),AVSWL,STIP
3211 FORMAT(/,5X,'TOTAL COAL MINED (TONS)',27X,18
CALL COST(NREAD, T,ADR, ASVL, AFL, ATR, ABUL, ACR, ACVL, KSHIFT, NPDR, NPBL,
FBLAS, PFAC, ADEP, TCOAL, FROCK, IOUT, N1, N2, TPCOAL, DB, DS, DAREA,
NC1, NC2, NROCK, ACOST, TCDR, CCDR, NDR, CTOVB, CTHL, CTI TK, NTR, TOVB, THL,
SNBL, TPBL, TTBBL, TPBL, OBBL, TDZ, ROVBN, CHCOST, CMATCH, TMATCH, ISIM,
NCODE, NLIFT, NBLOCK, CTSF, NSDY, NDMH, TTTK, TODL, TOVL)
7777 CONTINUE
5000 RETURN
END

SUBROUTINE COMPUT(NREAD, T,ADR, ASVL, AFL, ATR, ABUL, ACR, ACVL, KSHIFT,
+NPDR, NPBL, FBLAS, PFAC, ADEP, TCOAL, FROCK, IOUT, N1, N2, TPCOAL, DB, DS,
+DAREA, NC1, NC2, NROCK, ACOST, TCDR, CCDR, NDR, CTOVB, CTHL, CTITK, NTR, TOVB,
+THL, SNBL, TPBL, TTBBL, TPBL, OBBL, TDZ, ROVBN, CHCOST, CMATCH, TMATCH, ISIM,
+NCODE, NLIFT, NBLOCK, CTSF, NSDY, NDMH, TTTK, TODL, TOVL)
COMMON/BCST/ IC, NI, TH, WORKT, CFUEL(6), CIIT(6), DEPPH(6), REPAIR(6),
+CTIRE(6), CUNDFR(6), TCOPEP(6), SPLIM(6), TAHP, TCOST, NOP
+OCOST(6), TCOST, OCTCOST(6) COMMON/CEXTRA/ZNHP, ZSAL, NHP, NSP, WELP, OBFT, EXRECL, TAREA, CSEED,
+DDI, WHTP, PRMCS, AFCST, SLCOST, DETCST, BLCST, PRMRT
INTEGER SECTON, CCODE, Prsv, CJOB

DIMENSION CWT(10), FBLAS(10,5), PFAC(3), ADEP(10)
+, 5), FROCK(10,5,3), NCODE(10,5,8), NLIFT(10), TDZ(10,5), DAREA(10)

C

---

NOP = 0
SUPPLY = 0
SCOST = 0
TECT = 0
TAHP = 0
TH = T/60.
WORKT = TH*ADR
IF(ISIM.GT.1) GO TO 777
READ(3,*) CHCOST, ZNHP, ZSAL, NHP, NSP, WELP, OBFT, EXRECL, TAREA, CSEED
READ(3,*) DDI, WHTP, PRMCS, AFCST, SLCOST, DETCST, BLCST, PRMRT, COPT

7777 TEX = TAREA*(EXRECL+CSEED)

3450 IF(NPDR.EQ.0) GO TO 123
IG = NPDR
CALL SUBCST(1, ISIM)
123 IF(NPBL.EQ.0) GO TO 125
IG = NPBL
WORKT = TH*ABUL
CALL SUBCST(2, ISIM)
125 IF(N1.EQ.0) GO TO 126
IG = N1
WORKT = TH*ATR
CALL SUBCST(3, ISIM)

C

126 IF(N2.EQ.0) GO TO 127
IG = N2
WORKT = TH*ASVL
CALL SUBCST(4, ISIM)
127 IF(NC2.EQ.0) RETURN
IC=NC2
WORK=TH
ACVL
CALL SUBCST(ISIM)
RETURN
END

SUBROUTINE BULLDOZ(NDCODE,NILCEL,NC,KC,GALT,HALT)
IF(NDCODE.EQ.1) THEN
COMMON/BRES/DAREA(10),CTIK(10),ADEP(10,5),FBLAS(10,5),
+CREV,BDOZ(10,5),FLNH(10,5),PPAC(3),SWELF(3),FROCK(10,5,5),
+TODL,TOVL
COMMON/BTR/TEX,DMP,GWV,SHED(10),RP(10),HCTR,TCTR,HDIS(10,5),
+ROSI,NGEAR,NC1,CMATCH,TMATCH,NC2,GVC,DA,AMAX,VMAX,
+CW(20),DHLL(10),TODR,N1,ATR,ASVL,ACR,ACVL,CHD(10),
+LMDC,N2,CGFL,CAPL,WNTR,TOVBL,CHHL,CHT,
+X,NTL,TOVL,THL,TTIK,SCYC,WMATL,HTIME,
+TIMECYCLE,HTHT(10,5),CTHT(10),THT(10,5),
+(10,5),TDT(10,5),CTLD(10),CTLT(10),CDT(10),WTR(10,5),
+ISTR(10,5),CTTR(10),CAVR(10),CCTR(10),
+GTCT(10,5),DLC(10,5),DLC(10,5),CLCT(10,5),
COMMON/BTR/NAFIL(10),NBFL(10,5),NCF(10),SEQCL(10),
+VOLL(5),CHLLM(5),GEXST(5),VOLLEX(5),
+NCLFT(10,5),HILL(10,5,5),DILL(10,5,5),GEXST(5,10,5),
+TIME,OBMN,RVDN,NNMIW,WVOL,VLMAX,INDK,
+TIME1,CSV,1STR,IPSV,PSV(20),HTIR(20),PTR(20),
+PRSV,HCSV(20),CSV(20),CSV(20),CSV(20),PDMIN(10,5),
+FTIME2(10,5),NSEC(10,5),GFIL(10,10,5),NSECFL(10,10),
COMMON/GEN/HADV(6),NCODE(10,5,8),JOB(10,5,8),CJOB(10),NLIFT(10),
+TREDF(10,5,8),CTIM,WT(10,5,4),DADV,ADDO,SILF(10,5),
+ISEDE,IBLOCK,ILIFT,DEF(3),NSPACE(6),NLIFT(10),
+NIND(10,5,9),NROCK,ISTOP,TPRP(10),INEX,FNAME
COMMON/BNEW/NCFIL(10),NCFL(10),ETIME(10),FGFIL(5,10,5),
+FPFILL(5,10,5),CVOLL(10),NSECFL(10),NREADY(10),IFT(5,10),
+NUMB(5,5),VTLCD(10),ETHT(5),PRDF(5),FDTL(5),PDCT(5),
+FTR(5),FDIS(5),INX,FTCCT(5),VOLLE(5),FTR(5),FHI(5),
COMMON/BNEW/LNSH(10),INSH(10),BN(10),BNST(10),
+NCEL(10),EFFECT(5),PRCD(10,5),NST(10),NEX(5),BFROM(10,10,4),
+BFROM(10,10,4),LHFLR(10,10,4),BMO(10,10,4),NSTH(10),
+LFTOM(10,10,4),AMOUNT(10,10,4),NUMBER(10,10,4),TSTART(10,5,8),
+TFINISH(5),CDLCT(10,5),MOUNT(10,10,4),BFROM(10,10,4),
+LCF(10,10,4),TCDS(5),NDC(10,5),MCDEST(10,5),
COMMON/BDR/DBL,DS,NPDR,ADR,HIST(20),PRM(20),ISDR,TFDR(10,5),TDR,
+HY(10,5),CDDR,TCDR,TQVE
COMMON/BUL/NPFL,BCF(10,5),ABUL,HIBL(20),PFLB(20),PBBL(20),ISBL
+TDZ(10,5),OBBL,ITBL,GTIME(10),NUMBG(10),BTIME(10),
COMMON/ION/PSFIL(10),PSLT(10),PBL(10,5),PLFL(10,5),SEQQB(10,5),
+JLIFT(10),BPREP(10),CDEN,NHAUL,CTSF,NSY,NDLH,NDCDO(10,5),
+NDNDO(10,5),TRATE1,TRATE2,CONST,LTIME,DTIME,CAPB
INTEGER PSFIL,PSLT,PBLK,PLFL,SEQQB,JLIFT,PRDR(7),PRDF,PRTR
INTEGER SECTON,CCOE,COB,PSV,PCTR,PCFL,PBLK,BCFOM,LCF(10,10,4)
+BNST,FREE(8),SLIFT,ITIM,START,DGR,BFROM,BBFROM,BNSTH
REAL LTIME,NEX
CHARACTER*64 FNAME
JOB(IBLOCK,ILIFT,2)=1
PPH=0.
NN=NUMB(IBLOCK,ILIFT,2)
IF(NHAUL.EQ.1) THEN
  DISTAN=HALT
ELSE
  IF(NC.EQ.1) THEN
    DISTAN=FILL(IBLOCK,KC,1)
  ELSE IF(NC.EQ.2) THEN
    NS=NSCFL(IBLOCK,KC)
    DO 890 I=1,NS
    DISTAN=DISTAN+FILL(IBLOCK,KC,I)
  ELSE IF(NC.EQ.3) THEN
    NS=NSCHW(IBLOCK,KC)
    DO 891 I=1,NS
    DISTAN=DISTAN+HOLLOW(IBLOCK,KC,I)
  ELSE
    NS=NECHW(IBLOCK,KC)
    DO 892 I=1,NS
    DISTAN=DISTAN+HEXST(IBLOCK,KC,I)
  END IF
END IF
ATIME=(DISTAN/TRATE1+DISTAN/TRATE2)/BCF(IBLOCK,ILIFT)+CONST
  +DTIME+LTIME
PPH=60./ATIME*CAPB
WPPH=0.
DO 12 I=1,NROCK
  WPPH=WPPH*DESI(I)*FROCK(IBLOCK,ILIFT,I)+WPPH
WWT=0.
DO 66 IR=1,NROCK
  WWT=WWT+WWT(IBLOCK,ILIFT,2)*DESI(II)
TREQD IS TIME TO FINISH THE JOB
TREQD(IBLOCK,ILIFT,2)=TREQD(IBLOCK,ILIFT,2)+WWT/WPPH*60.
OBUL=OBUL+TREQD(IBLOCK,ILIFT,2)*NUMB(IBLOCK,ILIFT,2)
TTBL=TTBL+TREQD(IBLOCK,ILIFT,2)*NUMB(IBLOCK,ILIFT,2)
TDZ(IBLOCK,ILIFT)=TREQD(IBLOCK,ILIFT,2)
RETURN
ELSE
  NCFL(NILCEL)=1
  BTIMB(NILCEL)=BTI(M(NILCEL))/NUMB(NILCEL)
  ABUL=ABUL+BTM(NILCEL)*NUMB(NILCEL)
END IF
RETURN
END

THE SUBROUTINE MHDL FINDS OUT THE DESTINATION OF OVERBURDEN FOR A
LIFT. IT PRESERVES THE STATUS OF EACH FILL CELLS, HOLLOW, AND EXTRA
SPOIL STORAGE AREA. WHEN ONE IS FULL IT PUSH OVERBURDEN TO NEXT

SUBROUTINE MHDL

COMMON/BRES/DAREA(10), CAREA(10), CTIK(10), ADEP(10,5), FBLAS(10,5),
+ CREVR, FDOZ(10,5), FLNH(10,5), PFAC(3), SWEFL(3), FROCK(10,5,3),
+ TODL, TOVL

COMMON/BTR/TEX, DMP, GVW, SPEED(10), RPI(10), HCTR, TCTR, HDIS(10,5),
+ NRT, NGEAR, NC1, CMATCH, TMATCH, NC2, GVC, DA, AMA, VMAX
+ CWT(20), DHLL(10), TOTR, N1, ATR, ASVL, ACR, AGVL, CHD(10)
+ LHD, N2, CCPL, CAFL, NNT, CTVOB, CTNL, CTIT
+ NSTR, TOVBL, TBL, TITK, SCYCLE, WMAT, HTIME
+ TIME, CYCLE, RHT(10,5), CRHT(20), TH(10,5), CTHT(10), TL(T10,5)
+ TTR(10,5), CTR1(10), CAVTR(10), CCT(10)
+ TCCT(10,5), DLCT(20,5), CDLC(10,5), CLCT(10,5), CCLT(10,5)

COMMON /BTR/MPIF(10), NBFIL(10,5), NCODC(10), NCFCL(16), SEQCL(10),
+ VOLL(5), CHOLLW(5), GEXST(5), VOLLEX(5),
+ NCLFT(10,5), FILL(10,10,5)
+ NCEX(10,5), HOLLOW(10,5,5), HOLLW(10,5,5), GEXST
+ NSEC(10,5), NSCW(10,5), FCPLC(10), MINDIS
+ TMINE, OBMIN, ROBN, NHOOLL, NEX, TVOOLL, VOLUM, VOLUMAX
+ DIST, ICSV, ISTR, IPRBLK, PSV(20), PTR(20)
+ PRSV, HGSV(20), CSV(20), PCSV, HSVP(20), AVSPC, FTIME(10,5)
+ FTIME2(10,5), NSCEX(10,5), GFILL(10,10,5), NSCEPL(10,10), NPILCL

COMMON/GEN/HR6D(6), NCODE(10,5,8), JOB(10,5,8), CJOB(10), NLIFT(10)
+ ISEED, IBLOCK, IIFLT, DENS(5), NSPACE(6), NBLK, NMACH(10),
+ NNUMB(10,5,9), NROCK, ISTOP, TPREP(10), INEX, FNAME

COMMON/BNEW/NCBL(10), NCFL(10), ETIME(10), FGFIL(5,10,5),
+FILL(5,10,5), CVOLL(10), NSEEF(5,10), NREADY(10), FTIT(5,10)
+ NSEEF(10,5), FVPLC(10), FTIT(5), FTRIT(5), FDLC(5), FTDL(5)
+ FVTR(5), FDLC(5), INX, FICTT(5), VOLLLF(5), FTTR(5), FHT(5)

COMMON/DNEV/LSHT(10), LNST(10), BNST(10), BNSTH(10),
+ NCELT(10), EFFECT(5), FRCD(10,5), NST(10), NNX(5), BFROM(10,10,4)
+ BBFROM(10,10,4), LLRMR(10,10,4), HOUNT(10,10,4), NSTH(10)
+ LFROM(10,10,4), AMOUNT(10,10,4), NUMBER(10,4,3), TSTART(10,5,8)
+ TFINISH(5), CDMIT(10,5), CMOUNT(10,10,4), BCFROM(10,10,4)
+ LCFCM(10,10,4), TCOOD(5), TCOOD(5), NDCOM(10,10,5), NDCOM(10,10,5)

COMMON/BDR/DBDS, INPT, ADR, HIST(20), PRM(20), ISDR, TFRD(10,5), TDR
+ (10,5), CCNDR, TCDR, TACHERS

COMMON/BBUL/NPBL, BCF(10,5), ABUL, HIBL(20), PBFIL(20), PPBIL(20), ISBL
+ TDLZ(10,5), OBL, TBL, GTIME(10), NUMG(10), BTME(10)

COMMON/ION/PSBL(10), PSFLF(10), PBFIL(10,5), PRFL(10,5), SEQB(10)
+ JLFIT(10), NWFIL, HAILL, CTSF, NSD, NDMH, NDCO(10,5,3),
+ NDCO(10,5), TRATE1, TRATE2, CONSTLIME, DTIME, CAPP

INTEGER PSBL, PSFLF, PBFL, PRFL, SEQB, JLFIT, TDLZ, OBL, TBL, GTIME, NWFIL

REAL LTIME, NEX

CHARACTER*64 FNAME

ILI=ILIFT

IF(CODE.EQ.3.) GO TO 1000
IF(CODE.EQ.2.) THEN
   TREQD(1BL,ILI,3)=0
TREQD( IBL, ILI, 6) = 0
Z = WT( IBL, ILI, 3)
ELSE
  IF (CODE = 0.) THEN
    TREQD( IBL, ILI, 4) = 0
    TREQD( IBL, ILI, 7) = 0
    Z = WT( IBL, ILI, 4)
  ELSE
    TREQD( IBL, ILI, 2) = 0
    Z = WT( IBL, ILI, 2)
  END IF
END IF
END IF

IF (NCLFT( IBL, ILI). EQ. 1) THEN
  NNH = NDCOD( IBLOCK, ILLIFT)
  NND = MODEST( IBLOCK, ILLIFT)
  IF (CODE = 0.) THEN
    NUMBER( NND, 1, 1) = NUMBER( NND, 1, 1) + 1
    BFROM( NND, NUMBER( NND, 1, 1), 1) = IBLK
    LFROM( NND, NUMBER( NND, 1, 1), 1) = ILLIFT
  ELSE
    IF (CODE = 2.) THEN
      NUMBER( NND, 2, 1) = NUMBER( NND, 2, 1) + 1
      BFROM( NND, NUMBER( NND, 2, 1), 1) = IBLK
      LFROM( NND, NUMBER( NND, 2, 1), 1) = ILLIFT
    ELSE
      NUMBER( NND, 3, 1) = NUMBER( NND, 3, 1) + 1
      BCFROM( NND, NUMBER( NND, 3, 1), 1) = IBLK
      LCFROM( NND, NUMBER( NND, 3, 1), 1) = ILLIFT
    END IF
  END IF
END IF

IF (NNH = 0) THEN
  GALT = GFIL( IBLOCK, NND, 1)
  HALT = FILL( IBLOCK, NND, 1)
  NNK = 1
ELSE
  IF (NNH = 2) THEN
    GALT = GHOLLW( IBLOCK, NND, 1)
    HALT = HHOLLW( IBLOCK, NND, 1)
    NNK = 2
  END IF
END IF

IF (CODE = 0.) THEN
  ZZ = Z
  THT( IBLOCK, ILLIFT) = GALT
  HDIS( IBLOCK, ILLIFT) = HALT
  CALL TRUCK( ZZ, TRE, NNK, KF, K)
  TREQD( IBL, ILI, 4) = TRE
  CDLT( IBL, ILI) = TRE
  TREQD( IBL, ILI, 7) = TRE*TIME2( IBL, ILI)*NUMB( IBLOCK, ILLIFT, 4)/
    NUMB( IBLOCK, ILLIFT, 9)
  RETURN
ELSE
  IF (CODE = 2.) THEN
    ZZ = Z
    CALL TRUCK( ZZ, TRE, NNK, KF, K)
TREQD(IBL,ILI,3)=TRE
CDLC(IBL,ILI)=TRE
TREQD(IBL,ILI,6)=TRE*FTIME(IBL,ILI)*NUMB(IBLOCK,ILIFT,7)/
+ NUMB(IBLOCK,ILIFT,8)
ELSE
ZZ=Z
CALL BULDOZ(1,NILCEL,1,1,GALT,HALT)
END IF
RETURN
END IF
IF(NFILCL.EQ.0) GO TO 444
NK=NST(IBL)
DO 111 I=NK,NFILCL
K=SEQCL(I)
IF(NCFCL(K).EQ.0) GO TO 111
NM=NAFIL(K)
IF(NM.EQ.0) THEN
NCFCL(K)=1
GO TO 888
END IF
DO 77 NNM=1,NM
IS=NBFIL(K,NNM)
IF(NFCODE(IS).EQ.0)GO TO 77
NCFCL(K)=2
GO TO 177
77 CONTINUE
NCFCL(K)=1
177 IF(NCFCL(K).EQ.2) GO TO 444
888 CONTINUE
IF(CODE.EQ.0) THEN
NUMBER(K,2,1)=NUMBER(K,2,1)+1
LFROM(K,NUMBER(K,2,1),2)=ILI
ELSE
IF(CODE.EQ.2.) THEN
NUMBER(K,2,2)=NUMBER(K,2,2)+1
BBFROM(K,NUMBER(K,2,2),2)=IBL
LFROM(K,NUMBER(K,2,2),2)=ILI
ELSE
NUMBER(K,2,3)=NUMBER(K,2,3)+1
BCFROM(K,NUMBER(K,2,3),2)=IBL
LCFROM(K,NUMBER(K,2,3),2)=ILI
ENDIF
ENDIF
IF(Z.GT.VFILCL(K)) THEN
NCFCL(K)=0
ZZ=VFILCL(K)
ENDIF
IF(ZZ.MAV.EQ.0) THEN
ENDIF
IF(NHAUL.EQ.1) THEN
THI(IBL,ILI)=GFILL(IBL,K,1)
HDIS(IBL,ILI)=FILL(IBL,K,1)
ENDIF
ENDIF
IF(CODE.EQ.4.) THEN
CALL BULDOZ(1,NILCEL,2,K,GALT,HALT)
COUNT(K,NUMBER(K,2,3),2)=VFILCL(K)
ELSE
CALL TRUCK(ZZ,TRE,1,KF,K)
IF(CODE.EQ.2.) THEN
  BMOUNT(K,NUMBER(K,2,2),2)=VFILCL(K)
  TREQD(IBL,ILI,3)=TREQD(IBL,ILI,3)+TRE
  CDL1(IBL,ILI)=TREQD(IBL,ILI,3)
  TREQD(IBL,ILI,6)=TREQD(IBL,ILI,6)+TRE*FTIME(IBL,ILI)
ELSE
  AMOUNT(K,NUMBER(K,2,1),2)=VFILCL(K)
  TREQD(IBL,ILI,7)=TREQD(IBL,ILI,7)+TRE*FTIME2(IBL,ILI)
  TREQD(IBL,ILI,4)=TREQD(IBL,ILI,4)+TRE
END IF
END IF
Z=Z-VFILCL(K)
VFILCL(K)=0
GO TO 111
ELSE
VFILCL(K)=VFILCL(K)-Z
IF(CODE.EQ.0.) THEN
  IF(NHAUL.EQ.1) THEN
    THT(IBL,ILI)=THT(IBL,ILI)+TRE
    HDIS(IBL,ILI)=FDIS(IBL,ILI)
  END IF
END IF
IF(CODE.EQ.4.) THEN
  CALL BULDOZ(1,NILCEL,2,K,GALT,HALT)
  COUNT(K,NUMBER(K,2,3),2)=Z
ELSE
  CALL TRUCK(ZZ,TRE,1,KF,K)
  IF(CODE.EQ.2.) THEN
    BMOUNT(K,NUMBER(K,2,2),2)=Z
    TREQD(IBL,ILI,3)=TREQD(IBL,ILI,3)+TRE
    CDL1(IBL,ILI)=TREQD(IBL,ILI,3)
    TREQD(IBL,ILI,6)=TREQD(IBL,ILI,6)+FTIME(IBL,ILI)*TRE
  ELSE
    AMOUNT(K,NUMBER(K,2,1),2)=Z
    TREQD(IBL,ILI,4)=TREQD(IBL,ILI,4)+TRE
    TREQD(IBL,ILI,7)=TREQD(IBL,ILI,7)+FTIME2(IBL,ILI)*TRE
  END IF
END IF
END IF
RETURN
111 CONTINUE
444 IF(TVOLLW.GE.VOLMAX) GO TO 555
K=INDK
VOLLFT(INDK)=CHOLLW(INDK)-VOLL(INDK)
IF(CODE.EQ.0.) THEN
  NUMBER(K,3,1)=NUMBER(K,3,1)+1
  BFROM(K,NUMBER(K,3,1),3)=IBL
  LFROM(K,NUMBER(K,3,1),3)=ILI
ELSE
  IF(CODE.EQ.2.) THEN
NUMBER(K,3,2)=NUMBER(K,3,2)+1
BFROM(K,NUMBER(K,3,2),3)=IBL
LFROM(K,NUMBER(K,3,2),3)=ILI
ELSE
    NUMBER(K,3,3)=NUMBER(K,3,3)+1
    BCFROM(K,NUMBER(K,3,3),3)=IBL
    LCFROM(K,NUMBER(K,3,3),3)=ILI
END IF

END IF

IF(VOLLFT(INDK).GE.Z) THEN
    IF((TVOLLW+Z).GT.VOLMAX) THEN
        Z=Z-VOLMAX+TVOLLW
        ZZ=VOLMAX-TVOLLW
        VOLL(INDK)=VOLL(INDK)+ZZ
        VOLLFT(INDK)=VOLLFT(INDK)-ZZ
        TVOLLW=VOLMAX
    ELSE
        ZZ=Z
        VOLL(INDK)=VOLL(INDK)+Z
        VOLLFT(INDK)=VOLLW(INDK)-VOLL(INDK)
        TVOLLW=TVOLLW+Z
        Z=0
    END IF
END IF

IF(CODE.EQ.0.) THEN
    IF(NHAUL.EQ.1) THEN
        THT(IBM,ILI)=GHOLLW(IBM,INDK,1)
        HDIS(IBM,ILI)=HHOLLW(IBM,INDK,1)
    END IF
END IF

IF(CODE.EQ.4.) THEN
    CALL BULD(1,NILCEL,3,K,GALT,HALT)
    CMOUNT(K,NUMBER(K,3,3),3)=ZZ
ELSE
    CALL TRUCK(ZZ,2,2,K,INDK)
END IF

IF(CODE.EQ.2.) THEN
    AMOUNT(INDK NUMBER(K,3,2),3)=ZZ
    TREQD(IBL,ILI,3)=TREQD(IBL,ILI,3)+TRE*FTIME(IBL,ILI)
    ELSE
        AMOUNT(INDK NUMBER(K,3,1),3)=ZZ
        TREQD(IBL,ILI,4)=TREQD(IBL,ILI,4)+TRE
        TREQD(IBL,ILI,7)=TREQD(IBL,ILI,7)+TRE*FTIME2(IBL,ILI)
    END IF
END IF

IF(Z.EQ.0.) RETURN ELSE IF((TVOLLW+VOLLFT(INDK)).GT.VOLMAX) THEN
    Z=Z-VOLMAX+TVOLLW
    ZZ=VOLMAX-TVOLLW
    VOLL(INDK)=VOLL(INDK)+ZZ
    VOLLFT(INDK)=VOLLFT(INDK)-ZZ
    TVOLLW=VOLMAX
ELSE
    ZZ=VOLLFT(INDK)
VOLL(INDK) = VOLL(INDK) + VOLLFT(INDK)
VOLLFT(INDK) = CHOLLW(INDK) - VOLL(INDK)
TVOLLW = TVOLLW + VOLLFT(INDK)
Z = Z - VOLLFT(INDK)
END IF
IF (CODE.EQ.0.) THEN
  IF (NHAUL.EQ.1) THEN
    THIG(IBL,ILI) = GHOLLW(IBL,INDK,1)
    HDIG(IBL,ILI) = HHOLLW(IBL,INDK,1)
  END IF
END IF
IF (CODE.EQ.4.) THEN
  CALL BULDOZ(1, NILCEL, 3, K, GALT, HALT)
  CMOUNT(K, NUMBER(K, 3, 3)) = ZZ
ELSE
  CALL TRUCK(ZZ, TRE, 2, K, INDK)
END IF
IF (CODE.EQ.2.) THEN
  TREQD(IBL,ILI,3) = TREQD(IBL,ILI,3) + TRE
  CDLT(IBL,ILI) = TREQD(IBL,ILI,3) + TRE*FTIME(IBL,ILI)
  BMOUNT(K,NUMBER(K,3,2)) = ZZ
ELSE
  TREQD(IBL,ILI,4) = TREQD(IBL,ILI,4) + TRE
  TREQD(IBL,ILI,7) = TREQD(IBL,ILI,7) + TRE*FTIME2(IBL,ILI)
  AMOUNT(K, NUMBER(K, 3, 1)) = ZZ
END IF
END IF
INDK = INDK + 1
VOLL(INDK) = 0.
IF (Z.EQ.0) RETURN
GO TO 444
END IF
555 IF (NEX.EQ.0) THEN
  WRITE(*,788)
  $ ' NO SPOIL STORAGE AREA DEFINED. PROGRAM TERMINATED'/
  $ ' BECAUSE THERE IS NO PLACE TO DUMP SPOIL'
  STOP
ELSE
  IF (INEX.GT.NEX) THEN
    WRITE(*,789)
    $ ' ALL SPOIL STORAGE AREA FULL. PROGRAM TERMINATED'/
    $ ' BECAUSE THERE IS NO PLACE TO DUMP SPOIL'
    STOP
END IF
END IF
K = INEX
IF (CODE.EQ.0) THEN
  NUMBER(K,4,1) = NUMBER(K,4,1) + 1
  BFROM(K,NUMBER(K,4,1),4) = IBL
  LLFROM(K,NUMBER(K,4,1,4)) = IBL
ELSE
  IF (CODE.EQ.2) THEN
    NUMBER(K,4,2) = NUMBER(K,4,2) + 1
    BFROM(K,NUMBER(K,4,2),4) = IBL
    LLFROM(K,NUMBER(K,4,2),4) = IBL
  END IF
ENDIF
ELSE
  NUMBER(K,4,3)=NUMBER(K,4,3)+1
  BCFROM(K,NUMBER(K,4,3),4)=IBL
  ICFROM(K,NUMBER(K,4,3),4)=ILI
END IF
END IF

IF(VOLLEX(INEX).GT.Z) THEN
  NNEX(INEX)=NNEX(INEX)+Z
  ZZ=Z
  VOLLEX(INEX)=VOLLEX(INEX)-Z
END IF

IF(CODE.EQ.0.) THEN
  IF(NHAUL.EQ.1) THEN
    THT(IBL,ILI)=GEXST(IBL,INEX,1)
    HDIS(IBL,ILI)=HEXST(IBL,INEX,1)
  END IF
END IF

IF(CODE.EQ.4.) THEN
  CALL BULDOZ(1,NILCEL,4,K,GALT,HALT)
  CMOUNT(INEX,NUMBER(INEX,4,3),4)=Z
END IF

ELSE
  CALL TRUCK(ZZ,TRE,4,K,INEX)
END IF

IF(CODE.EQ.2.) THEN
  AMOUNT(INEX,NUMBER(INEX,4,2),4)=Z
  TREQD(IBL,ILI,3)=TRE+TREQD(IBL,ILI,3)
  CDLT(IBL,ILI)=TREQD(IBL,ILI,3)
  TREQD(IBL,ILI,6)=TREQD(IBL,ILI,6)+TRE*FTIME(IBL,ILI)
ELSE
  AMOUNT(INEX,NUMBER(INEX,4,1),4)=Z
  TREQD(IBL,ILI,4)=TREQD(IBL,ILI,4)+TRE
  TREQD(IBL,ILI,7)=TREQD(IBL,ILI,7)+TRE*FTIME2(IBL,ILI)
END IF

END IF

ROVBN=ROVBN+ZZ
RETURN

ELSE
  NNEX(INEX)=GEXST(INEX)
  ZZ=VOLLEX(INEX)
  Z=Z-VOLLEX(INEX)
END IF

IF(CODE.EQ.0.) THEN
  IF(NHAUL.EQ.1) THEN
    THT(IBL,ILI)=GEXST(IBL,INEX,1)
    HDIS(IBL,ILI)=HEXST(IBL,INEX,1)
  END IF
END IF

IF(CODE.EQ.4.) THEN
  CALL BULDOZ(1,NILCEL,4,K,GALT,HALT)
  CMOUNT(INEX,NUMBER(INEX,4,3),4)=VOLLEX(INEX)
ELSE
  CALL TRUCK(ZZ,TRE,4,K,INEX)
END IF

IF(CODE.EQ.2.) THEN
  AMOUNT(INEX,NUMBER(INEX,4,2),4)=VOLLEX(INEX)
  TREQD(IBL,ILI,3)=TRE+TREQD(IBL,ILI,3)
  CDLT(IBL,ILI)=TREQD(IBL,ILI,3)
  TREQD(IBL,ILI,6)=TREQD(IBL,ILI,6)+TRE*FTIME(IBL,ILI)
ELSE
AMOUNT(INEX, NUMBER(INEX, 4), 4) = VOLLEX(INEX)
TREQD( IBL, ILI, 4) = TREQD( IBL, ILI, 4) + TRE
TREQD( IBL, ILI, 7) = TREQD( IBL, ILI, 7) + TRE * FTIME(IBL, ILI)
END IF
ROVBN = ROVBN + ZZ
INEX = INEX + 1
VOLLEX(INEX) = 0.
IF(Z .EQ. 0) GO TO 190
GO TO 555
END IF
END IF
190 RETURN
1000 NN = 0
DO 1001 I = 1, NEX
IF(NNEX(I) .EQ. 0) GO TO 1001
NN = I
INX = NN
ETIME(NN) = 0
1001 CONTINUE
IF (NN .EQ. 0) RETURN
IF(NFILCL .EQ. 0) RETURN
Z = NNEX(NN)
DO 7111 I = 1, NFILCL
K = SEQCL(I)
IF(NCFCL(K) .EQ. 0) GO TO 7111
NM = NAFIL(K)
IF (NM .EQ. 0) THEN
NCFCL(K) = 1
GO TO 7888
END IF
DO 777 NNM = 1, NM
IS = NBFIL(K, NNM)
IF(NFCLCODE(IS) .EQ. 0) GO TO 777
NCFCL(K) = 2
GO TO 7177
777 CONTINUE
NCFCL(K) = 1
7177 IF(NCFCL(K) .EQ. 2) GO TO 7444
7888 CONTINUE
IF(Z .GT. VFILCL(K)) THEN
NNEX(NN) = NNEX(NN) - VFILCL(K)
ZZ = VFILCL(K)
VFILCL(K) = 0
NCFCL(K) = 0
IF (NHAUL .EQ. 1) THEN
ETIME(INX) = FGFILL(NN, K, 1)
FDIS(INX) = FFILL(NN, K, 1)
END IF
CALL TRUCK(NHAUL, ZZ, TRE, 5, INX, K)
END IF
RETURN
1000 NN = 0
ZV=Z
VFILCL(K)=VFILCL(K)-Z
VOLLEX(NN)=CEXST(NN)
IF(NHAUL.EQ.1) THEN
  FHT(NN)=FFIL(NN,K,1)
  FM(NN)=FFIL(NN,K,1)
END IF
CALL TRUCK(NHAUL,ZZ,TRE,5,K,INX)
ETIME(NN)=TRE
END IF
7444 DO 7890 L=1,NFILCL
7890 IF(VFILCL(L).NE.0) RETURN
DO 7891 L=1,NEX
7891 NNEX(L)=O. RETURN
CONTINUE
END SUBROUTINE TRUCK(ZZ,TRE,HCODE,KF,K)
COMMON/BRES/DAREA(10),CAREA(10),CTIK(10),ADEP(10,5),FBLAS(10,5),
+ CREVR,FDOZ(10,5),FLNH(10,5),FFAQ(3),SWELF(3),PRO(10,5,3),
+ TOGL,TOLV
COMMON/BTR/TEX,DMP,GWV,SPEED(10),RP(10),HCNT,TCNT,HDIS(10,5)
+ K,NI,NEAR,NC1,CMATCH,TMATCH,NC2,GVC,DA,AMAX,VMAX
+ CWT(20),DHTIL(10),TNTR,N1,ATR,ASVL,ACR,ACVL,CHD(10)
+ ,LDTC,N2,CCPL,CAPL,NNTR,CVOB,CTHL,CTTT
+ K,TOTV,TOLB,THL,TTLK,SCYCLE,WHATL,HTIME
+ NMT1,NMT2,NMT3,NMT4,CGHT(10,5),CMT(10,5),CLHT(10,5),CMLT(10,5),
+ CHT(10,5),CMHT(10,5),CCT(10,5),CSTM(10,5)
COMMON /BTR/NAFIL(10),NFIL(10,5),NCFIL(10,5),NCFIL(10),SEQCL(10),
+ VOL(5),CHOLL(5),CEBXST(5),VOLLEX(5),
+ ,NCLFT(10,5),FIL(10,10,5)
+ ,HXS(10,5,5),GHOLLW(10,5,5),HOLLW(10,5,5),GEXST
+ ,DCC(10,5,5),NSL(10,5,5),NSLCT(10,5,5),NFILCL(10)
+ ,HOCT(V10,5),HOCT(V10,5),HDC(10,5),HDC(10,5),HDL(10,5),
+ HDT(10,5),HTL(10,5),HTCT(10,5),HDL(10,5),HDLT(10,5),HDT(10,5),
+ HTR(10,5),HTT(10,5),CTT(10,5),CTTR(10,5),SMIL(10,5),
+ TCF(10,5),NLCT(20,5),NLCT(10,5),CCT(10,5)
COMMON /BTR/NAFIL(10),NFIL(10,5),NCFIL(10,5),NCFIL(10),SEQCL(10),
+ VOL(5),CHOLL(5),CEBXST(5),VOLLEX(5),
+ ,NCLFT(10,5),FIL(10,10,5)
+ ,HXS(10,5,5),GHOLLW(10,5,5),HOLLW(10,5,5),GEXST
+ ,DCC(10,5,5),NSL(10,5,5),NSLCT(10,5,5),NFILCL(10)
+ ,HOCT(V10,5),HOCT(V10,5),HDC(10,5),HDC(10,5),HDL(10,5),
+ HDT(10,5),HTL(10,5),HTCT(10,5),HDL(10,5),HDLT(10,5),HDT(10,5),
+ HTR(10,5),HTT(10,5),CTT(10,5),CTTR(10,5),SMIL(10,5),
+ TCF(10,5),NLCT(20,5),NLCT(10,5),CCT(10,5)
COMMON/BBUL,NPBL,BCF(10,5),ABUL,HIBL(20),PFBL(20),PBBL(20),ISBL
+,TDZ(10,5),OBBL,ITBL,GTIME(10),NUMBG(10),BTIME(10)
COMMON/IION/PSBLK(10),PSLFT(10),PRBLK(10,5),PRLF(10,5),SEQBB(10)
+NNDOZ(10,5),TRATE1,TRATE2,CONST,LTIME,DTIME,CAPB
INTEGER PSBLK,PSLFT,PRBLK,PRLF,SEQBB,TRAVE,PRDR(7),PRFL,PRTR
INTEGER SECTON,CCODE,GJOB,PCSV,PCTR,PCLF,PRBL,BCFROM,LCFROM,
+BNST,FREE(8),SLIFT,ITIM,START,DGR,BFROM,BBFROM,BNSTH
REAL LTIME,NNEX
CHARACTER*64 FNAME
INTEGER HCODE LOGICAL ROAD

I= ILIFT
DIST=0
TCYCLE=0.
TRE=0.
NTRUK=0
IF(CODE.EQ.2.) GO TO 7777
IF(CODE.EQ.1.) THEN
CJOB( IBL)=1
ELSE
IF(CODE.EQ.0.) JOB( IBL,ILI,4)=1
END IF
ROAD=. FALSE.
MCODE=0.
TMINE=0.
ENDIF
IF(CODE.EQ.0.) THEN
NOM=NUMB( IBL,ILI,3)/NUMB( IBL,ILI,4)
NRES=MOD(NUMB( IBL,ILI,3),NUMB( IBL,ILI,4))
NOM=NOM+1
SCYCLE=0.
WHATL=GWV+TOTR
ENDIF
IF(NHAUL.EQ.1) GO TO 7703
CALL HAUL(ROAD,2Z,HCODE,KF,K)
WHATL=GWV
THT( IBL,ILI)=HTIME/60.
ROAD=. NOT. ROAD.
CALL HAUL(ROAD,2Z,HCODE,KF,K)
RHT( IBL,ILI)=HTIME/60.

7703
TUT( IBL,ILI)=SAMPLE(ISTR,PTR,HITR,NSPACE,I,3,ISEED,ISTOP)+DMP
IF(NHAUL.EQ.0) THEN
SCYCLE=SCYCLE+RHT( IBL,ILI)+TDT( IBL,ILI)+THT( IBL,ILI)
ELSE
SCYCLE=SCYCLE+THT( IBL,ILI)+TDT( IBL,ILI)
END IF
DLCT( IBL,ILI)=SAMPLE(IPS,PSV,HISV,NSPACE,I,5,ISEED,ISTOP)
TLT( IBL,ILI)=DLCT( IBL,ILI)+ITIM+SCYCLE+THT( IBL,ILI)"+(NRES=(NOM+1)+(NUMB
+(( IBL,ILI,4)-NRES)*NOM)/NUMB( IBL,ILI,4))
IF(WTR( IBL,ILI).LT.0) THEN
WTR( IBL,ILI)=0
ELSE
SCYCLE=SCYCLE+TTL( IBL,ILI)+WTR( IBL,ILI)
END IF

WWT=0.
DO 66 I=1,NROCK
    WWT=WWT+Zz*DENS(I)*FROCK(IBL,ILI,I)
    TRE=WWT/PPH*60./NUMB(IBL,ILI,1)
    TOVB=TOVB+Zz
    IF(NHAUL.EQ.1) DIST = HDIS(IBL,ILI)
    THT=THL+TRE/TCCT(IBLOCK,ILIFT)*DIST*NUMB(IBLOCK,ILIFT,3)*2
    TMATCH=TMATCH+TCCT(IBLOCK,ILIFT)-WTR(IBLOCK,ILIFT)
    CMATCH=CMATCH+TLC(IBLOCK,ILIFT)
    TITK=TITK+TRE*NUMB(IBLOCK,ILIFT,3)
    TLTK=TLTK+TRE*NUMB(IBLOCK,ILIFT,4)
    ATR=ATR+TRE*NUMB(IBLOCK,ILIFT,3)
    ASVL=ASVL+TRE*NUMB(IBLOCK,ILIFT,4)
    ABUL=ABUL+FTIME(IBLOCK,ILIFT)*TRE*NUMB(IBLOCK,ILIFT,9)
ELSE
    FTCCT(INX)=SCYCLE
    WWT=0.
    DO 966 I=1,NROCK
        WWT=WWT+Zz*DENS(I)
        TRE=WWT/PPH*60./NUMBF(INX,1)
        FT0VB=FT0VB+Zz
        IF(NHAUL.EQ.1) DIST = FDIS(INX)
        THT=THL+TRE/FTCCT(INX)*DIST*NUMBF(INX,1)*2
        FTTR(INX)=TRE
        TTTK=TTTK+TRE*NUMBF(INX,1)
        ATR=ATR+TRE*NUMBF(INX,1)
        ASVL=ASVL+TRE*NUMBF(INX,2)
    END IF
END IF
C 876 FORMAT( ' TREQD ',F4.1)
RETURN

ACYCLE=SAMPLE(IPSV,PSV,HISV,NSPACE,1,5,ISEED,ISTOP)
CLCT(IBLOCK,ILIFT)=ACYCLE
JOB(IBLOCK,ILIFT,3)=ACYCLE
ATIME=0
GO TO (1,2,3,4) HCODE
1 IF(NHAUL.EQ.1) THEN
   ISECT = 1
ELSE
   ISECT=NSECFL(IBLOCK,K)
END IF
DO 131 ISEC=1,ISECT
   ATIME=(FILL[IBLOCK,K,ISEC)/AVSPD)+ATIME
131 CONTINUE
GO TO 5
2 IF(NHAUL.EQ.1) THEN
   ISECT = 1
ELSE
   ISECT=NSECCHW(IBLOCK,K)
END IF
DO 132 ISEC=1,ISECT
   ATIME=(HLOW[IBLOCK,K,ISEC)/AVSPD)+ATIME
132 CONTINUE
GO TO 5
4 IF(NHAUL.EQ.1) THEN
   ISECT = 1
IF(CODE.EQ.3.) THEN
INX=KF
NOM=NUMBF(INX,1)/NUMBF(INX,2)
NRES=MOD(NUMBF(INX,1),NUMBF(INX,2))
NOM1=NOM+1
SCYCLE=0.
WMATL=GVW+TOTR
IF(NHAUL.EQ.1) GO TO 17703
CALL HAUL(ROAD,ZZ,HCODE,INX,K)
WMATL=GVW
FHT(INX) = HTIME/60.
ROAD = NOT. ROAD
ROAD = NOT. ROAD
CALL HAUL(ROAD,ZZ,HCODE,KF,K)
FHT(INX) = HTIME/60.
17703
FTDT(INX) = SAMPLE(ISTR, PTR, HITR, NSPACE, I, 3, ISEED, ISTOP) + DMP
IF(NHAUL.EQ.1) THEN
SCYCLE = SCYCLE + FTDT(INX) + FHT(INX)
ELSE
SCYCLE = SCYCLE + FTDT(INX) + FHT(INX)
END IF
FDLCT(INX) = SAMPLE(IPSV, PSV, HISV, NSPACE, I, 5, ISEED, ISTOP)
FTL(I, X) = FDLCT(INX) * TOTR/CAPL+TEX
FWTR(INX) = SCYCLE + FTL(INX) * ((NRES * (NOM + 1)) + (NUMBF(INX, 2) - NRES) * NOM) / NUMBF(INX, 2)
IF(FWTR(INX).LT.0) THEN
FWTR(INX) = 0
END IF
SCYCLE = SCYCLE + FTL(INX) + FWTR(INX)
ELSE
DLCT(INX) = SAMPLE(IPSV, PSV, HISV, NSPACE, I, 5, ISEED, ISTOP)
TLT(INX) = DLCT(INX) * TOTR/CAPL+TEX
SCYCLE = 0
SCYCLE = SCYCLE + HCTR
CTLD(INX) = SAMPLE(ICSV, CSV, HCSV, NSPACE, I, 5, ISEED, ISTOP)
CTLT(INX) = CTLD(INX) * TCTR/CCPL
SCYCLE = SCYCLE + CTLT(INX)
END IF
END IF
CYCLE = 60. / SCYCLE
PPH = 0.
PPH = CYCLE * TOTR + PPH
IF(CODE.EQ.1.) THEN
TREQD(IBLOCK, 1, 5) = GWT(IBL) / PPH * 60. / NUMB(IBL, 1, 5) + TPREP(IBLOCK)
CTTR(IBL) = TREQD(IBL, 1, 5)
CCT(IBLOCK) = SCYCLE
CTOVB = CTOVB + CWT(IBL)
CTTK = CTTK + TREQD(IBL, 1, 5) * NUMB(IBL, 1, 5)
CTHL = CHTH + TREQD(IBL, 1, 5) / CCT(IBL) * DIST * NUMB(IBL, 1, 5) *
+ 2
ACR = ACR + TREQD(IBL, 1, 5)
ACVL = ACVL + TREQD(IBL, 1, 5) * NUMB(IBL, 1, 6)
ELSE
IF(CODE.EQ.0.) THEN
TCCT(IBL, ILL) = SCYCLE
END IF
ELSE
    ISECT=NSEC_EX(IBLOCK,K)
END IF
DO 134 ISEC=1 ISECT
134 ATIME=(HEXST(IBLOCK,K,ISEC)/AVSPD)+ATIME
5 CONTINUE
WWT=0
DO 135 IROCK=1 NROCK
135 WWT=WWT+2*ZDENS(IBLOCK)*FROCK(IBLOCK,ILIFT,IROCK)
ACYCLE=ACYCLE+TIME=2
CMLT(IBLOCK, ILIFT)=CMLT+CLCT(IBLOCK, ILIFT)
TRE=WWT/ACMLT=ACMLT/NUMB(IBLOCK,ILIFT,7)
ASVL=ASVL+TRE*NUMB(IBLOCK,ILIFT,7)
TLTK=TLTK+TRE*NUMB(IBLOCK,ILIFT,7)*FTIME(IBLOCK,ILIFT)
RETURN
END
SUBROUTINE Haul(ROAD,ZZ,HCODE,KF
DIMENSION GRADE(10,10,5),HLENTH(10,10,5),TRADE(10,10,5)
COMMON/BRES/DAREA(10),CAREA(10),CTHR(10),ADEP(10,5),FBLAS(10,5),
    CREVR,EDOZ(10,5),FNLN(10,5),PFAC(3),SWELF(3),FROCK(10,5,3)
    +,TODL,TOVL
COMMON/BTR/TEX,DMP,GWV,SPEED(10),RP(10),HCTR,TCTR,HDIS(10,5)
    +,RORI,NGEAR,NC1,CMATCH,TMATCH,NC2,GVC,DA,AMAX,VMAX
    +,CWT(20),DHILL(10),TSTR,N1,ATR,ASVL,ACR,ACVL,CHD(10)
    +,LHDC,N2,CAPL,NCPL,NNTR,CTOV8,CTHL,CTTW
    +,K,NT,TML,TML,TML,TML,TML,TML,TML,TML
    +,TIME,CYCLE,RHT(10,5),CRHT(20),HTTH(10,5),CTHT(10),TLT
    +,(10,5),TTD(10,5),CMTD(10),CTLT(10),CTDT(10),WTR(10,5)
    +,TTR(10,5),CTTR(10),CAVR(10),CCT(10)
    +,TCPT(10,5),DLCT(20,5),CDLC(10,5),CLCT(10,5),CCLT(10,5)
COMMON/BTR/NAFIL(10),NBFLIL(10,5),NFCODE(10),NFCL(10),SEQCL(10),
    VOLL(5),CHOLLW(5),CEXT(5),VOLLEX(5),
    NCLFT(10,5),FILL(10,10,5)
    +,HEXST(10,5,5),GHOLLW(10,5,5),HHOLLW(10,5,5),GEXST
    +,(10,5,5),NSECAL(10,5),NSECW(10,5),CFILC(10),MINDIS,
    +,TMIN,TMIN,TMIN,TMIN,TMIN,TMIN,TMIN,TMIN
    +,TIME,OMBINE,ROVBE,HOGLW,NEHOLLW,TVOLLW,TVOL,INDK
    +,DIST,ICSV,ISTR,PSV(20),HTR(20),PTR(20)
    +,PSV,HCVS(20),CSSV(20),PSV,HSV(20),AVSPD,FTIME(10,5)
    +,FTIME(10,5),NSECEx(10,5),GFIL(10,10,5),NSECFL(10,10),NFILC
COMMON/GEN/MRED(6),NCODE(10,10,5),JOB(10,5,5),CJOB(10),NLIFT(10)
    +,TREDQ(10,5,8),CTHR,WT(10,5,4),CCODE,COCC,SLFT(10,5)
    +,ISSEE,IBLOCK,ILIFT,DENS(3),NSPACE(6),NBLCK,NMACH(10)
    +,NMB(10,5,5),NROCK,ISTOP,TPREP(10),INEX,FNAME
COMMON/BNEW/NCLFL(10),NFCL(10),ETIME(10),FGFLIL(10,5,5)
    +,FILL(10,5,5),CVOLL(10),NSECFL(10,5),NREADY(10),TFT(5,10)
    +,NUMB(5,5),VFILC(10),FTHT(5),FRHT(5),FDTL(5),FDLCT(5),FTLT(5)
    +,FWTR(5),FDIS(5),INX,FTCTR(5),VOLLFL(5),FTFL(5),FHT(5)
COMMON/DNEW/LNSTH(10),LNST(10),BNST(10),BNSLH(10)
    +,NCE(10),EFECT(5),FRCD(10,5),NSTR(10),NNSX(5),BFROMM(10,10,4)
    +,BBFROMM(10,10,4),LRFROM(10,10,4),BMONTH(10,10,4),NSTRH(10)
    +,LFROM(10,10,4),AMOUNT(10,10,4),NORDER(10,4,3),TSTEGR(10,5,8)
    +,TFINISH(5),CMLT(10,5),CMOUNT(10,10,4),BFROM(10,10,4)
    +,LRFROM(10,10,4),TCOSD(5),TCOSB(5),NDCOD(10,5),MCDEST(10,5)
COMMON/BDR/DB,DS,NPD,ADR,HIST(20),PRH(20),ISP,TFD(10,5),TTDR
RESMOT IS RESISTANCE TO MOTION
DD TOTAL DISTANCE COVERED BY TRUCK(I,J)
GO TO (1,2,3,4,5),HCODE
1 SECTO=NSECFL(IBLOCK,K)
DO 11 ISEC=1,SECTO
GRADE(IBLOCK,ILIFT,ISEC)=GFILL(IBLOCK,K,ISEC)
DIST=FILL(IBLOCK,K,ISEC)
11 HLENTH(IBLOCK,ILIFT,ISEC)=FILL(IBLOCK,K,ISEC)
GO TO 6
2 SECTO=NSECHW(IBLOCK,K)
DO 22 ISEC=1,SECTO
GRADE(IBLOCK,ILIFT,ISEC)=GHOLLW(IBLOCK,K,ISEC)
DIST=HHOLLW(IBLOCK,K,ISEC)+DIST
22 HLENTH(IBLOCK,ILIFT,ISEC)=HHOLLW(IBLOCK,K,ISEC)
GO TO 6
3 SECTO=NSECEX(IBLOCK,K)
DO 44 ISEC=1,SECTO
GRADE(IBLOCK,ILIFT,ISEC)=GEXST(IBLOCK,INEX,ISEC)
DIST=HEXST(IBLOCK,K,ISEC)
44 HLENTH(IBLOCK,ILIFT,ISEC)=HEXST(IBLOCK,K,ISEC)
GO TO 6
4 SECTO=NSECEF(KF,K)
DO 46 ISEC=1,SECTO
GRADE(KF,K,ISEC)=FGFILL(KF,K,ISEC)
DIST=FFILL(KF,K,ISEC)
46 HLENTH(KF,K,ISEC)=FFILL(KF,K,ISEC)
6 CONTINUE
DO 234 II = 1,SECTO
IF(ROAD) THEN
IF(HCODE.NE.5) THEN
TRADE(IBLOCK,ILIFT,II) = -GRADE(IBLOCK,ILIFT,II)
ELSE
TRADE(KF,K,II) = -GRADE(KF,K,II)
END IF
THTR=WMTL-GVW
WMTL=GVW
ELSE
THTR=0.
IF(HCODE.NE.5) THEN
TRADE(IBLOCK,ILIFT,II) = GRADE(IBLOCK,ILIFT,II)
ELSE
TRADE(KF,K,II) = GRADE(KF,K,II)
C END IF
END IF
234 CONTINUE
DS=0.
T=0.
S1=0.
SUM=0.
DD=0.
35 DO 84 II=1,SECTO
IF(HCODE.NE.5) THEN
RESMOT=(RORI+TRADE(IBLOCK,ILIFT,II))*20*WMATL
ELSE
RESMOT=(RORI+TRADE(KF,K,II))*20*WMATL
END IF
END IF
DO 667
KM=1.
AF ACCELERATION FORCE
AF=AVBRPC1,I6SPEED1 RP~NGEAR)*1000/453.6-RESMOT
A=AF/(WMATL*2000.*32.2)
IF(A<AMAX)76,77,77
77 A=AMAX
C HLENGTH(IBLOCK,ILIFT,II) LENGTH OF THE SECTON II
C DS DISTANCE COVERED BY TIME DT
76 IF(HCODE.NE.5) THEN
S2=SQR((S1**2+2*A*HLENGTH(IBLOCK,ILIFT,II))/20.)
ELSE
S2=SQR((S1**2+2*A*HLENGTH(KF,K,II))/20.)
END IF
IF(S2>VMAX) S2=VMAX
IF(HCODE.NE.5) THEN
IF(TRADE(IBLOCK,ILIFT,II).LT.0) GO TO 78
ELSE IF(TRADE(KF,K,II).LT.0) GO TO 78
END IF
GO TO 79
78 IF(HCODE.NE.5) THEN
INTEGR=-INT(TRADE(IBLOCK,ILIFT,II))
ELSE
INTEGR=-INT(TRADE(KF,K,II))
END IF
IF(S2>DHILL(INTEGR))S2=DHILL(INTEGR)
79 DT=(S2-S1)/(A)
T=T+DT
IF(HCODE.NE.5) THEN
DS=DS+HLENGTH(IBLOCK,ILIFT,II)/20.
ELSE
DS=DS+HLENGTH(KF,K,II)/20.
END IF
C DA CONSTANT DEACCELERATION RATE
S1=S2
667 CONTINUE
84 CONTINUE
99 HTIME=T
80 CONTINUE
WMATL=GTV+THTR
RETURN
END
FUNCTION AVBRP(S1, I, SPEED, RP, NGEAR)
DIMENSION SPEED(10), RP(10)
S1 = S1*60./88.
DO 88 K = 1, NGEAR
IF(SPEED(K).EQ.S1) GO TO 71
IF(SPEED(K).GT.S1) GO TO 73
GO TO 88
71 NP = K
GO TO 87
73 IF(K.EQ.1) GO TO 1293
AB = -RP(K) + RP(K-1)
BC = SPEED(K) - SPEED(K-1)
ED = S1 - SPEED(K-1)
AVBRP = RP(K) + AB*ED/BC
GO TO 80
1293 AB = -RP(K)
BC = SPEED(K)
ED = S1
AVBRP = RP(K) + AB*ED/BC
GO TO 80
87 AVBRP = RP(NP)
GO TO 80
88 CONTINUE
S1 = S1*88./60.
RETURN
END
FUNCTION SAMPLE(IDIS, PRM, HIST, NSPACE, LM, ML, ISEED, ISTOP)
DIMENSION HIST(20), PRM(20), NSPACE(5)
PI = 3.14159
IF(IDIS.EQ.0) GO TO 99
R = RAND(ISEED, ISTOP)
GO TO(101, 102, 103, 104, 105, 107), IDIS
99 GO TO(101, 102, 103, 104, 105, 107), IDIS
C AVERAGE VALUE
SAMPLE = PRM(1)
RETURN
C UNIFORM DISTRIBUTION
101 SAMPLE = PRM(1) + R*(PRM(2) - PRM(1))
RETURN
C EMPERICAL DISTRIBUTION
102 NN = NSPACE(ML)
DO 110 I = 1, NN
IF(R.GE.HIST(I)) GO TO 110
IF(I.EQ.1) GO TO 455
SAMPLE = (PRM(I-1) + PRM(I))/2
RETURN
110 CONTINUE
222 SAMPLE = PRM(NN)
RETURN
455 SAMPLE = PRM(1)
RETURN
C EXPONENTIAL DISTRIBUTION
103 SAMPLE = PRM(1) - PRM(2)*ALOG(R)
RETURN
NORMAL DISTRIBUTION
104 R1=R
R2=RAND(ISEED,ISTOP)
SAMPLE =PRM(1)+PRM(2)*SQRT(-2.*ALOG(R1))*COS(2*PI*R2)
A=3.*PRM(2)
IF(SAMPLE .LE. 0.)SAMPLE =.001
IF (SAMPLE.LT.(PRM(1)-A)) SAMPLE=PRM(1)-A
IF (SAMPLE.GT.(PRM(1)+A)) SAMPLE =PRM(1)+A
RETURN

WEIBULL DISTRIBUTION
105 SAMPLE=PRM(1)+PRM(2)*(-ALOG(1.-R))**(1./PRM(3))
RETURN

TRIANGULAR DISTRIBUTION
107 BA=PRM(2)-PRM(1)
CA=PRM(3)-PRM(1)
CB=PRM(3)-PRM(2)
IF(R.LE.(BA/CA))SAMPLE=PRM(1)+SQRT(CA*BA*R)
IF(R.GT.(BA/CA)) SAMPLE=PRM(3)-SQRT(CA*BA*(1.-R))
RETURN
END

FUNCTION RAND(ISEED,ISTOP)
THIS FUNCTION GENERATES RANDOM NUMBERS UNIFORMLY DISTRIBUTED IN
THE RANGE (0,1)

DOUBLE PRECISION Z,D2P31M,DMOD
DATA D2P31M/2147483647.DO/
Z=ISEED Z=DMOD(16807.DO*Z,D2P31M) ISEED=Z
RAND=Z/D2P31M
IF (ISTOP.EQ.-1)RAND=1-RAND
RETURN
END

SUBROUTINE DRILL
COMMON/BRES/DAREA(10),CAREA(10),CTIK(10),ADEF(10,5),FBLAS(10,5),
+ CREVR,FD0Z(10,5),FLNH(10,5),PFAC(3),SWEFL(3),FROWK(10,5,3),
+ TODL,TOLV
COMMON/BTR/TEX,DMP,GW,SPEED(10),RP(10),HCT,TCTR,HDIS(10,5),
+ RORI,NCEAR,NC1,CMATCH,TMATCH,MC2,GVC,DA,AMAX,VMAX
+ CW(20),DHILL(10),TCTR,N1,ATR,ASVL,ACR,ACVL,CHD(10),
+ LHD,CN,CCPL,CAPI,NNTR,CFOV,B,CTH,CCTT
+ N1,NT1,TV0B,THL,TTTK,SCYCLE,WMATL,HTIME,
+ TIME,CYCLE,RHT(10,5),RHT(20),THI(10,5),CTHT(10),TLT(10,5),
+ TDTI(10,5),CRLD(10),CTLT(10),CTDT(10),WTR(10,5),
+ TRT(10,5),CCTR(10),CCTR(10),CCTR(10),
+ TCCT(10,5),DLCT(20,5),DLCT(10,5),CLCT(10,5),
+ COMMON /BTR!/NAFIL(10),NBFIL(10,5),NFCODE(10),NCFCL(10),SEQCL(10),
+ VOLL(5),CHOLL(5),CEXX(5),VOLLEX(5),
+ NLFL(10,5),FILL(10,10,5),
+ HEXST(10,5,5),GOLLW(10,5,5),HHOLLW(10,5,5),GEXST
THIS SUBROUTINE CALCULATES THE DRILLING TIME REQUIRED

I=IBLOCK J=ILIFT EFF=0
JOB(I,J,1)=1 DP=DS*DB
NH IS THE NO OF DRILLING HOLE NH=DAREA(I)/DS/DB
NDEACH=INT(FLOAT(NH)/FLOAT(NUMB(IBLOCK,ILIFT,1)))+1
DLEACH IS THE TIME REQUIRED TO DRILL ONE HOLE
DLEACH=ADEP(I,J)*FBLAS(I,J)
FUN=SAMPLE(ISDR,PRM,HIST,NSPACE,NO,1,ISEED,ISTOP)
DO 7 IROCK=1,NROCK
7 EFF=EFF+EFFECT(IROCK)*FROCK(IBLOCK,ILIFT,IROCK)
FUN=FUN*EFF
TDMAX=DLEACH/FUN
TREQP(I,J,1)=NDEACH*TDMAX+NH*TMOVE
ADR=ADR+TREQP(I,J,1)*NUMB(I,J,1)
TCTRL=DEC+TREQP(I,J,1)*NUMB(I,J,1)
RETURN
THIS PROGRAM IS TO CALCULATE THE AMOUNT OF MATERIAL THAT
HAS TO BE MOVED BY ANY OPERATION.

SUBROUTINE OVERBN(IOUT, TCOAL)

COMMON/BRES/DAREA(10), CAREA(10), CTIK(10), ADEP(10,5), FBLAS(10,5),
+ CREVR, FD02(10,5), FLNH(10,5), PFAG(3), SWELP(3), FROCK(10,5,3),
+ T0DL, TOVL

COMMON/BTR/TEX, DMP, GVW, SPEED(10), RP(10), HTCRT, TCTR, HDIS(10,5),
+ RORI, NGEAR, NC1, CMATCH, TMATCH, NC2, GUC, DA, AMAX, VMAX
+ CWT(20), DHILL(10), TTRR, N1, ATR, ASVL, ACR, ACVL, CHD(10)
+ LHDC, N2, CCP, CAPL, NNT, CTO1B, CTHL, CTIT
+ K, NTR, TOVB, THL, TTRK, SCYCLE, WMLT, HTIME
+ TIME, CYCLE, RHT(10,5), CRHT(20), THT(10,5), CTHT(10,5), TLT
+ (10,5), TDT(10,5), CTDL(10), CTRT(10,5), CDT(10,5), WTR(10,5)
+ TTR(10,5), CTRR(10), CAVR(10), CCT(10)
+ TCR(10,5), DLR(20,5), DCLC(10,5), CLT(10,5), CT(10,5)

COMMON /BTR/NAFIL(10), NBFIL(10,5), NFCODE(10), NCFCL(10), SEQCL(10),
+ VOL(5), CHOLLW(5), GEXST(5), VOLLEX(5),
+ NCLFT(10,5), FILL(10,5,5)
+ HEXST(10,5,5), GCHOLLW(10,5,5), HCHOLLW(10,5,5), GEXST
+ (10,5,5), NSECAL(10,5), NSECCHW(10,5), CFIILC(10), MINDIS,
+ TMINE, OBMINE, ROVBV, NHOLLW, NEX, TVOLL, VOLMAX, INDK
+ DIST, ICSS, IST, IPSV, PSV(20), HITR(20), PTR(20),
+ PRSV, HCsv(20), CSV(20), PCSV, HSV(20), AVSPD, FTIME(10,5)
+ FTIME(10,5), NSECX(10,5), GFIL(10,10,5), NSECFL(10,10), NFILC1

COMMON/GEN/MRED(6), NCODE(10,5,8), J0B(10,5,8), CJOB(10), NLFIL(10)
+ TRED(10,5,8), CTIM, WT(10,5,4), CGCODE, CODE, SLIFT(10,5)
+ ISEED, IBLOCK, ILLFT, DENS(5), NSPACE(6), NBLOCK, NMAC(10)
+ NLF(10,5,9), NROCK, ISTOP, TPER(10), INEX, FNAME

COMMON/NEW/NCBL(10), NCFIL(10), ETIME(10), FGFIL(5,10,5),
+ FFILL(5,10,5), CVOLL(10), NSECCEF(5,10), NREADY(10), FTF(5,10)
+ NLFILC(10,5), VFTILC(10), FTT(5), SFR(5), FDT(5), FDLT(5), FFILT(5)
+ FWR(5,5), FDIS(5), INX, FTC(5), VOLLT(5), FTIR(5), FHT(5)

COMMON/DNEW/LNSTH(10), LNST(10), BNSTH(10),
+ NC(10), NECF(10), PRCD(10,10,4), BFIOM(10,10,4),
+ BBF(10,10,4), BFM(10,10,4), BNFIL(10,10,4),
+ BONR(10,10,4), AMQLT(10,10,4), NUMBER(10,4,3),
+ TSTART(10,5,8)
+ TFIN(5,10,5), CLFIL(10,10,4), CFIL(10,10,4),
+ LCFIL(10,10,4), TCOSD(5), TCOSB(5), NDCOD(10,5), MCDST(10,5)

COMMON/BDR/DBL, DS, NPDR, ADR, HIST(20), PRM(20), ISDR, TFRD(10,5),
+ + (10,5), CDDR, TCDR, MOVE

COMMON/BBUL/NFIL, BFI(10,5), ABUL, HIHL(20), PFB(20), PBBL(20), ISBL
+ TDOZ(10,5), OBB, TTBL, GTIME(10), NUMBG(10,5), BTIME(10)

COMMON/ION/PBILK(10), PSFIL(10), PBBLK(10,5), PRLFT(10,5), SEQBB(10)
+ PLUS(10), BPREP(10), CDEN, NHAUL, CTSF, NSWY, NDHH, NCCOZ(10,5),
+NDOZ(10,5), TRATE1, TRATE2, CONST, LTIME, DTIME, CAPP

INTEGER PSBLK, PSFIL, PBLK, PRLFT, SEQBB, PLUS, BPREP, CDEN, NHAUL, CTSF, NSWY, NDHH, NCCOZ(10,5),
+NDOZ(10,5), TRATE1, TRATE2, CONST, LTIME, DTIME, CAPP

REAL LTIME, NEX
CHARACTER*64 FNAME
TOB=0
TOTTFL=0
OBMINE=0
TCOAL=0
TTLIFT=0
DO 10 IBLOCK=1,NBLOCK
LIFT=NLIFT(IBLOCK)
DO 10 ILIFT=1,LIFT
DO 35 IROCK=1,NROCK
35 OBMINE=OBMINE+DAREA(IBLOCK)*ADEP(IBLOCK,ILI$FT)*FROCK(IBLOCK,ILIFT,IROCK)*SWELF(IROCK)/27
DO 10 K3=1,4
WT(IBLOCK,ILIFT,K3)=0
GWT=DAREA(IBLOCK)*ADEP(IBLOCK,ILIFT)
IF(NCODE(IBLOCK,ILIFT,K3).EQ.0) GO TO 10
GO TO (1,2,3,4),K3
1 WT(IBLOCK,ILIFT,K3)=GWT*FBLAS(IBLOCK,ILIFT)/27.
DO 144 I=1,NROCK
144 WT(IBLOCK,ILIFT,K3)=GWT*FROCK(IBLOCK,ILIFT,IROCK)*SWELF(IROCK)
GO TO 89
2 WT(IBLOCK,ILIFT,K3)=GWT*FDOZ(IBLOCK,ILIFT)/27.
GO TO 89
3 WT(IBLOCK,ILIFT,K3)=GWT*FRCD(IBLOCK,ILIFT)/27.
DO 145 I=1,NROCK
145 WT(IBLOCK,ILIFT,K3)=GWT*FLNH(IBLOCK,ILIFT)/27.
DO 146 I=1,NROCK
146 WT(IBLOCK,ILIFT,K3)=GWT*FRNH(IBLOCK,ILIFT)/27.
DO 147 I=1,NROCK
147 CONTINUE
DWT=0.
DO 5 IROCK=1,NROCK
5 WTOB=WT(IBLOCK,ILIFT,K3)*FROCK(IBLOCK,ILIFT,IROCK)*SWELF(IROCK)
CONTINUE
DO 2231 IBLOCK=1,NBLOCK
CWT(IBLOCK)=CAREA(IBLOCK)*CTIK(IBLOCK)*CDEN*CREVR
2231 TCOAL=TCOAL+CWT(IBLOCK)
DO 878 I=1,NBLOCK
878 KK=NLIFT(I)
DO 878 J=1,KK
IF(NCLFT(I,J).NE.1) THEN
TTLIFT=TTLIFT+WT(I,J,4)+WT(I,J,3)+WT(I,J,2)
ELSE
NMM=NCOD(I,J)
NCC=MCDEST(I,J)
IF(NNM.EQ.1) THEN
VFILCL(NNC)=VFILCL(NNC)-WT(I,J,2)-WT(I,J,3)-WT(I,J,4)
ELSE
VOLLFT(NNC)=VOLLFT(NNC)-WT(I,J,2)-WT(I,J,3)-WT(I,J,4)
END IF
END IF
END IF
CONTINUE
IF(NFILCL.EQ.0) THEN
VOLMAX=TOTTFL
ELSE
DO 989 I=1,NFILCL
989 TOTFIL=TOTFIL+VFILCL(I)
VOLMAX=TOTTFL-TOTFIL
IF(VOLMAX.LT.0) WRITE(IOUT,900)
900 FORMAT('ERROR. FILL CELL VOLUME IS MORE THAN STRIP CELL VOLUME')
END IF
RETURN
END
SUBROUTINE COST(NREAD,T,ADR,ASVL,AFL,ATR,ABUL,ACR,ACVL,KSHIFT,
+NPBL,NVPBL,FBLAS,PFAC,ADEP,TCOAL,CRFACK,IOUT,N1,N2,TPCOAAL,DS,
+DAREA,NC1,NC2,NROCK,ACOST,TCDF,CDR,NDR,CTOVBL,CTRL,CTITK,NTR,TOVB,
+THL,NVL,TPBL,TTBL,OBBL,TZ,ROVBN,CHCOST,CMATCH,MTMATCH,ISIM
+NCODE,NLIFT,NBLOCK,CTSF,NSDY,NDMH,TTTK,TODL,TOVL)
COMMON/BGST/IC,NI,TH,WKRT,CFUEL(6),CLIF(6),DEFF(6),REPAIR(6),
+CTIRE(6),CUNOC(6),TCOPER(6),SPLIT(6),TAHP,TCOST,NOP
+,OPCOST(6),COPCOST(6)
COMMON/CEXTRA/NHPL,ZSAL,NHP,NSP,WELP,OBFT,EXRECL,TAREA,CSEED,
+D12,WHFT,PRMCST,APCST,SLCOST,DCST,BCST,PRMRT
INTEGER SECTON,CCODP,PRSV,CJOB
DIMENSION CWT(IO),FBLAS(IO,5),PFAC(3),ADEP(IO
+,5),FROCK(IO,5,3),NCODE(IO,5,6),NLIFT(IO),TDZ(IO,5),DAREA(IO)
------------------------------
SUPPLY=0
SCOST=0
TEC=0
TAREA=(EXRECL+CSEED)
COMPUTE COST OF SALARIED PERSONNEL
COMPUTE HOURLY PERSONNEL (OTHER THAN OPERATORS) COST
TH=KSHIFT*CTSF
129 THP=THP*TH
THP=THP*TH
ASPC=NSP*ZSAL*T/400./3./NDMH/12.
TOTAL HOURLY PAYROLL
TAP=TAHP+ASPC+THP
MISC. OPERATING EXPENSE RELATED TO LABOR
ZWEL=WELP*TH*(NHP+NOP)
ZOBFT=OBFT*TCOAL
OELM=ZWEL+ZOBFT
COMPUTE TOTAL BLASTING COST
COVER=2.
TOTTFL=0.
DO 171 I=1,NBLOCX
KK=NLIFT(I)

DO 171 J=1, KK
IF(NCODE(I, J, 1).EQ.0) GO TO 171
NHOLES=DAREA(I)/DB/DS
CYFH=DB/DS/27
HSTEM=.7*DB
VOLEFT=CYFH*FBLAS(I, J)*ADEP(I, J)

PEH=0.
DO 113 K=1, NROCK
113 PEH=PFAC(K)*VOLEFT*FROCK(I, J, K)+PEH
AFLB=PEH/(1+PRMRT)*NHOLES*(1-WHT)
SLLBPH=PEH/(1+PRMRT)

PLB=(PEH*PEH/(1+PRMRT))/NHOLES
SLLB=SLLBPH*NHOLES*WHT
TUTBLS=PLB*PRMST/100+SLLB+SLCOST/100+AFLB*AFGST/100+(1.15*(DS+FBL
+AS(I, J))*ADEP(I, J)-COVER)*NHOLES)/100+DETCST/1000+BLCST*/(SLLBPH
+AFLB)/100+TUTBLS

CONTINUE

171 CONTINUE
420 FORMAT(/60X,12(1H-)/)
WRITE(IOUT,472)NPDR, INT(CCDR), TCDR/60.
472 FORMAT(9X,'DOIL IN G'/9a, 1, 9X 'TOTAL NO.

478 FORMAT(/9X,'DOIL IN G'/'9X, 

481 FORMAT(/9X,'OVERBUREN D H AUL I N G'/'9X, 

479 FORMAT(/9X,'TOTAL NO. OF TRUCKS'/'9X, 

491 FORMAT('COAL LOADING AND HAULING'/'9X, 

1895 FORMAT('MATCH FACTOR'/'37X,F10.1///)
WRITE(IOUT,491)

1979 FORMAT(9X,'TOTAL NO. OF LOADER'/'31X, I8/9X,'TOTAL AMOUNT OF COAL

7896 FORMAT(IOUT,7896)
7896 FORMAT(/' / EQUIPMENT COST (OPERATORS WAGES ARE NOT INCLUDED):'
+---------------------------------------------------------------------
 TOTCOS=OPCOST(1)+OWCOST(1)
 THWR=TH*ADR
 THIDL=TH-THWR
 WRITE(OUT,908)OPCOST(1),OWCOST(1),TOTCOS,THWR,THIDL
 908 FORMAT(/' / COST OF DRILL
 + ----------
 + OPERATING COST PER HOUR ,F8.2
 + OWNING COST PER HOUR ,F8.2
 + TOTAL COST PER HOUR ,F8.2
 + TOTAL HOURS WORKED ,F8.2
 + TOTAL HOURS IDLE ,F8.2
 TOTCOS=TOTCOS*THWR*NPD
 THWR=TH*ABUL
 THIDL=TH-THWR
 WRITE(OUT,909)TOTCOS,TIDCOS,CPERYD
 909 FORMAT(/' / TOTAL OPERATING COST
 + ----------
 + OPERATING COST PER HOUR ,F8.2
 + OWNING COST PER HOUR ,F8.2
 + TOTAL COST PER HOUR ,F8.2
 + TOTAL HOURS WORKED ,F8.2
 + TOTAL HOURS IDLE ,F8.2
 TOTCOS=OPCOST(2)+OWCOST(2)
 THWR=TH*ADR
 THIDL=TH-THWR
 WRITE(OUT,910)OPCOST(2),OWCOST(2),TOTCOS,THWR,THIDL
 910 FORMAT(/' / COST OF DOZING
 + ----------
 + OPERATING COST PER HOUR ,F8.2
 + OWNING COST PER HOUR ,F8.2
 + TOTAL COST PER HOUR ,F8.2
 + TOTAL HOURS WORKED ,F8.2
 + TOTAL HOURS IDLE ,F8.2
 TOTCOS=TOTCOS*THWR*NPD
 TIDCOS=THIDL*OWCOST(2)*NPBL
 CPERYD=(TOPCOS+TIDCOS)/OCL
 WRITE(OUT,911)TOPCOS,TIDCOS,CPERYD
 911 FORMAT(/' / TOTAL OPERATING COST
 + ----------
 + OPERATING COST PER HOUR ,F8.2
 + OWNING COST PER HOUR ,F8.2
 + TOTAL COST PER HOUR ,F8.2
 + TOTAL HOURS WORKED ,F8.2
 + TOTAL HOURS IDLE ,F8.2
 TOTCOS=OPCOST(3)+OWCOST(3)
 THWR=TH*ADR
 THIDL=TH-THWR
 WRITE(OUT,912)OPCOST(3),OWCOST(3),TOTCOS,THWR,THIDL
 912 FORMAT(/' / COST OF TRUCK
 + ----------
 + OPERATING COST PER HOUR ,F8.2
 + OWNING COST PER HOUR ,F8.2
 + TOTAL COST PER HOUR ,F8.2
 + TOTAL HOURS WORKED ,F8.2
 + TOTAL HOURS IDLE ,F8.2
 TOTCOS=TOPCOS*THWR*N1
 TIDCOS=THIDL*OWCOST(3)*N1
 CPERYD=(TOPCOS+TIDCOS)/TOVB
WRITE(IOUT, 913) TOPCOST, TIDCOST, CPERYD

913 FORMAT(' TOTAL OPERATING COST ', F10.2
+ / ' TOTAL IDLE COST ', F10.2
+ / ' COST OF OVERBURDEN TRUCK ($/LCY) ', F8.2)
TOTCOST=OPCOST(4)+OWCOST(4)
THWOR=TH*ASVL
THIDL=TH-THWOR
WRITE(IOUT, 914) OPCOST(4), OWCOST(4), TOTCOST, THWOR, THIDL

914 FORMAT(' COST OF OVERBURDEN LOADER
+ / ' OPERATING COST PER HOUR ', F8.2
+ / ' OWNING COST PER HOUR ', F8.2
+ / ' TOTAL COST PER HOUR ', F8.2
+ / ' TOTAL HOURS WORKED ', F8.2
+ / ' TOTAL HOURS IDLE ', F8.2)

TOPCOST=THWOR*TOTCOST*N2
TIDCOST=OWCOST(4)*THIDL*N2
CPERYD=(TOPCOST+TIDCOST)/TOVL
WRITE(IOUT, 915) TOPCOST, TIDCOST, CPERYD

915 FORMAT(' TOTAL OPERATING COST ', F10.2
+ / ' TOTAL IDLE COST ', F10.2
+ / ' COST OF OVERBURDEN LOADER ($/LCY) ', F8.2)

THWOR=TH*ACR
THIDL=TH-THWOR
WRITE(IOUT, 916) OPCOST(6), OWCOST(6), TOTCOST, THWOR, THIDL

916 FORMAT(' COST OF COAL LOADER
+ / ' OPERATING COST PER HOUR ', F8.2
+ / ' OWNING COST PER HOUR ', F8.2
+ / ' TOTAL COST PER HOUR ', F8.2
+ / ' TOTAL HOURS WORKED ', F8.2
+ / ' TOTAL HOURS IDLE ', F8.2)

TOPCOST=TOTCOST*THWOR*N2
TIDCOST=OWCOST(6)*THIDL*N2
CPERYD=(TOPCOST+TIDCOST)/CTOVB
WRITE(IOUT, 917) TOPCOST, TIDCOST, CPERYD

917 FORMAT(' TOTAL OPERATING COST ', F10.2
+ / ' TOTAL IDLE COST ', F10.2
+ / ' COST OF COAL LOADER ($/TON) ', F8.2)
WRITE(IOUT, 1742)

1742 FORMAT(' 9X, 'COSTS' / ' 9X, '-------------' /)
WRITE(IOUT, 99) KSHIFT

99 FORMAT(' 9X, COMPLETION TIME (SHIFTS)', 26X, I10/
IF(COPT.EQ.1) THEN
WRITE(IOUT, 442)

442 FORMAT(10X, 'PAYROLL COSTS' / 9X, '-------------------------------

454 FORMAT(10X, 'TOTAL HOURLY PAYROLL (NON OPERATORS =', I2, ')
WRITE(IOUT, 454) NOP, INT(THP)

455 FORMAT(14X, 'TOTAL HOURLY PAYROLL (OPERATORS =', I2, ')
WRITE(IOUT, 456) NSP, INT(ASPC)
456 FORMAT(1H,9X,'SALARIED PERSONNEL PAYROLL (' ,I2,' PERSONS)' ,11X,I +10)
   WRITE(IOUT,420)
   WRITE(IOUT,458)INT(TAP)
458 FORMAT(1H,9X,'TOTAL PAYROLL COSTS',31X,I10/)
   WRITE(IOUT,464)
464 FORMAT(//10X,'LABOR RELATED OVERHEAD COST'/9X,'-------------- $---------')
   WRITE(IOUT,468)INT(ZWEL)
468 FORMAT(10X,'PENSION TRUST',37X,I10)
   WRITE(IOUT,470)INT(ZOBFT)
470 FORMAT(1H,9X,'BENEFIT TRUST',37X,I10///)
   WRITE(IOUT,471)
471 FORMAT(10X,'OWNING COST OF EQUIPMENT ($)'/10X,'-------------- $---------')
   IF(NPDL.EQ.0) GO TO 860

**OUTPUT OF EQUIPMENT PERFORMANCE AND THEIR OWNING COST**

TENT=TECT+DEPPH(1)+CIIT(1)
   WRITE(IOUT,786)INT(DEPPH(1)),INT(CIIT(1))
786 FORMAT(1H,9X,'OWNING COST FOR DRILLING MACHINE'/20X,'DEPRECIATION' $,'ION COST',23X,I10/20X,'INSURANCE,TAXES ETC.',20X,I10/)
860 IF(NPDL.EQ.0) GO TO 861
   TENT=TECT+DEPPH(2)+CIIT(2)
   WRITE(IOUT,787)INT(DEPPH(2)),INT(CIIT(2))
787 FORMAT(1H,9X,'OWNING COST FOR DOZER.'//20X,'DEPRECIATION ', $,'ON COST',23X,I10/20X,'INSURANCE,TAXES ETC.',20X,I10)

861 IF(N1.EQ.0) GO TO 862
   TENT=TECT+DEPPH(3)+CIIT(3)
   WRITE(IOUT,788)INT(DEPPH(3)),INT(CIIT(3))
788 FORMAT(//10X,'OWNING COST FOR TRUCK(OVERBURDEN)'//20X,'DEPRECIATION' $,'ON COST',23X,I10/20X,'INSURANCE,TAXES ETC.',20X,I10)

862 IF(N2.EQ.0) GO TO 864
   TENT=TECT+DEPPH(4)+CIIT(4)
   WRITE(IOUT,790)INT(DEPPH(4)),INT(CIIT(4))
790 FORMAT(//10X,'OWNING COST FOR LOADER (OVERBURDEN)'//20X,'DEPRECIATION ' $,'ON COST',23X,I10/20X,'INSURANCE,TAXES ETC.',20X,I10)
864 IF(N2.EQ.0) GO TO 865
   TENT=TECT+DEPPH(6)+CIIT(6)
   WRITE(IOUT,791)INT(DEPPH(6)),INT(CIIT(6))
791 FORMAT(//10X,'OWNING COST FOR LOADER(COAL)'//20X,'DEPRECIATION', $,'ON COST',22X,I10/20X,'INSURANCE,TAXES ETC.',19X,I10/)
865 WRITE(IOUT,890)INT(TECT)
890 FORMAT(59X, '---------'/10X,'TOTAL EQUIPMENT OWNING COST',23X +,I10///)

**TTIRE=0**
**TFUEL=0**
**TUNDC=0**
**TITM =0**
TPAIR=0
DO 866 IK=1,6
  IF(IEQ.5) GO TO 866
  IF(CTIRE(IK).NE.0)TTIRE=TTIRE+CTIRE(IK)
  IF(CFUEL(IK).NE.0)TFUEL=TFUEL+CFUEL(IK)
  IF(CUNDC(IK).NE.0)TUND= TUND+CUNDC(IK)
  IF(REPAIR(IK).NE.0)TPAIR=TPAIR+REPAIR(IK)
  IF(SPLITM(IK).NE.0)TITM=TITM+SPLITM(IK)
866 CONTINUE
1987 WRITE(IOUT,867)
867 FORMAT(' ',9X,'SUPPLY COST ($')/10X,'-----------------------

+-----+
       IF(CTIRE. NE. 0) WRITE(IOUT,870)INT(CTIRE)
       IF(FUEL. NE. 0) WRITE(IOUT,871)INT(FUEL)
       IF(TUND. NE. 0) WRITE(IOUT,872)INT(TUND)
       IF(TPAIR. NE. 0) WRITE(IOUT,873)INT(TPAIR)
       IF(TITM. NE. 0) WRITE(IOUT,874)INT(TITM)
       IF(TOTBLG. NE. 0) WRITE(IOUT,875)INT(TOTBLG)
870 FORMAT(10X,'TOTAL TIRE REPLACEMENT COST',35X,I8)
871 FORMAT(10X,'TOTAL FUEL COST',35X,I8)
872 FORMAT(10X,'TOTAL UNDERCARRIAGE COST',35X,I8)
873 FORMAT(10X,'TOTAL REPAIR COST',35X,I8)
874 FORMAT(10X,'MISCELLANEOUS SUPPLY (LUBRICANT ETC.) COST',
      $9X,I8)
875 FORMAT(9X,'TOTAL BLASTING COST',31X,I8)
     TSP=TTIRE+TFUEL+TUND+CUNDC+TPAIR+TITM+TOTBLG
     WRITE(IOUT,518)INT(TSP)
518 FORMAT(1H,9X,'TOTAL SUPPLY COST',33X,I8///)
   END IF
   WRITE(IOUT,986)INT(CHCOST*TCOAL)
986 FORMAT(10X,'COAL HAULING COST',32X,I9)
   WRITE(IOUT,987)INT(TEX)
987 FORMAT(9X,'GRADING AND SEEDING COST',25X,I9)
   TCOS=TEX+TCT+OELM+TAHP+TSP+CHCOST*TCOAL
   WRITE(IOUT,474)INT(TCOS)
474 FORMAT(10X,'TOTAL COST ($'),35X,I9/
       + 9X,'------------------------)
   WRITE(IOUT,989)TCOS/TCOAL
989 FORMAT(10X,'COST OF COAL ($/TON'),29X,F12.2
       + 9X,'------------------------)
   WRITE(IOUT,2344)
2344 FORMAT(39X,'END'/39X,'---'/1')
RETURN
END
SUBROUTINE SUBCST(MC,ISIM)
COMMON/BCST/IC,NZ,NTH,WORKT,CFUEL,CIIT,DEPH,REPAIR,CTIRE,CUNDC,
   TCOPER,SPLITM,TAHP,TCOST,NOP,OFCOST,COPT,OCOST
REAL I
IF(COPT.EQ.1) THEN
   SPLITM(MC)=0
   CIIT(MC)=0
   CFUEL(MC)=0
   CIIT(MC)=0
   CUNDC(MC)=0
RETURN
END
DEPPH(MC) = 0
CTIRE(MC) = 0
REPAIR(MC) = 0
IF(ISIM.GT.1) GO TO 777
READ(3,*) NCTIRE, DPRICE, RESVAL, DEPPER, TIIFAC, FUELUP, FUELCS
READ(3,*) CLFGPH, EXUMUL, BRF, SPL, NOPER, OPERHW
IF(NCTIRE.EQ.0) THEN
  IF(ISIM.EQ.1) READ(3,*) I, A, Z, BF
ELSE
  IF(ISIM.EQ.1) READ(3,*) TRCOST, ESTLIIH
END IF
C VALDEP IS VALUE FOR DEPRECIATION
777 CONTINUE
DEPPH(MC) = (DPRICE-TRCOST-RESVAL)/DEPPER*IC*TH+DEPPH(MC)
CTIRE(MC) = (TIIFAC*DPRICE/1000.)*IC*TH+CTIRE(MC)
CFUEL(MC) = (FUELUP*FUELCS)*IC*WORKT+CFUEL(MC)
IF(NCTIRE.EQ.1)CTIRE(MC) = (TRCOST/ESTLIIH)*IC*WORKT+CTIRE(MC)
IF(NCTIRE.EQ.0) CUND(MC) = BRF*(I+A+Z)*IC*WORKT+CUND(MC)
REPAIR(MC) = EXUMUL*BRF*IC*WORKT+REPAIR(MC)
SPLITM(MC) = SPLITM(MC)+WORKT*(SPL+CLFGPH)*IC
NOP=NOP+NPER
TAHP=TAHP+NOPER*OPERHW
OPCOST(MC) = CTIRE(MC)+CFUEL(MC)+CUND(MC)+REPAIR(MC)+SPLITM(MC)
OWCOST(MC) = CIIT(MC)+DEPPH(MC)
OPCOST(MC) = OPCOST(MC)/WORKT/IC
OWCOST(MC) = OWCOST(MC)/TH/IC
RETURN
ELSE
READ(8,*) OPCOST(MC), OWCOST(MC)
END IF
RETURN
END
APPENDIX B

Data Program
C$JOB AMAL, PAGE=50, TIME=4

Written by: AMAL CHAKRABORTY
DATE : 9/23/85

Purpose: THIS PROGRAM CREATES A DATA FILE FOR THE SIMULATOR
OPSIM. IT CAN ALSO BE USED FOR CHANGING EXISTING FILE.

Program: DATA

COMMON/ACOM/ISEED,ISTOP,CTIM,CDEN,
+NSDY, NDUM, NROCK,PFA(5), DENS(5), SWE(5), CREVR,EFFECT(5)
COMMON/BCOM/NPDR,ISDR,DB, DB, PRM(20),HIST(20), THOVE
+DNR, WNP, TPRT, TRATE, TRATE, LTIME, DTIME, CONST, CAPB
COMMON/DCOM/N1, N2,ISTR,TWX, DMP, HCTR, GVR, TOTR, TCTR, NHAUL
+S,PTR(20),HTR(20),NGEAR, DA, DORI, DGR, AMAX, VMAX, DHIIL(20)
+SPEED(10),RP(10)
COMMON/ECOM/2,NC2, IPSY, CAPL, CCPL, AVSPD
COMMON/FCOM/PSV(20),HISV(20),CSV(20),HC(20)
COMMON/GCOM/DAREA(10),CAREA(10),CTIK(10),PSBLK(10)
+PSLFT(10),BPREP(10),JLFT(10),TPREP(10),NST(10),NSTD(10)
+LNST(10),LNSTH(10),BNST(10),BNSTH(10)
COMMON/HCOM/PBBLK(10,5),PRLFT(10,5),ADEP(10,5),FBLAS(10,5),
+FDOZ(10,5),FLNH(10,5),FROCK(10,5,3),NCLFT(10,5),BCF(10,5),
+FRC(10,5),FTIME(10,5),FTIME2(10,5)
COMMON/ICOM/NAFIL(10),NFIL(10,5),FILL(10,10,5)
+NSECFL(10,10),GFLP(10,10,5),GFLC(10,5),NFILCL,BTIME(10)
COMMON/LCOM/CHOLLW(5),GHOLLW(10,5,5),HNOLLW(10,5,5),NSECW
+(10,5),NUHOLLW
COMMON/JCOM/GALT(10,5,5),HALT(10,5,5)
+NSECAL(10,5)
COMMON/KCOM/HEXT(10,5,5),GEXT(10,5,5),NSECX(10,5,5),CEXT(5)
+NEX
COMMON/MCOM/FFILL(10,5,5),FGFLX(10,5,5),NSEC(10,5)
COMMON/MCOM/HCOMP,CHCST,CHST,CHST,CHST,CHST,CHST,CHST,CHST,CHST
+CHST,CHST,CHST,CHST,CHST,CHST,CHST,CHST,CHST,CHST,CHST
COMMON/OCOM/NCNTRE(6),NPRICE(6),RESV(6),DEP(6),TIFAC(6)
+FUELUP(6),FUELGC,ELPGF,ELPGF,ELPGF,ELPGF,ELPGF,ELPGF,ELPGF
+NOPER(6),OPRHW(6),I(6),A(6),Z(6),BF(6),TRCOST(6),ESTLH(6)
+OCOST(6),OCOST(6)
COMMON/GEN/NSPACE(7),NBLOCK,NLIFT(10)
INTEGER PSBLK,PSLFT,PSBLK,PSLFT,SEQB,JSKT,JSKF,SSKF,SHL,
INGENERAL,GCODE,PSV,CSJ,PRD(7),BNST,BNST
+FREE(8),SLT,STIM,STAR,DR,COPT
REAL II,LTIME
CHARACTER*64 FNAME
CHARACTER*1 OPTION
PRDR(1)=1
PRDR(2)=2
PRDR(4)=2
PRDR(5)=1
PRDR(6)=3
PRDR(7)=2
DATA FILE CREATION PROGRAM
FOR THE PROGRAM OPSIM.

DO YOU WANT TO CREATE A NEW FILE OR MODIFY EXISTING FILE. ENTER OPTION:

1: CREATE FILE
2: CHANGE FILE
3: QUIT

DO YOU WANT TO CREATE A NEW FILE OR MODIFY EXISTING FILE. ENTER OPTION:

1: CREATE FILE
2: CHANGE FILE
3: QUIT

IlLEGAL OPTION. TRY AGAIN

CALL ACARD(NOPT,PRDR,NHOLLW,NEX,NFILCL)
CALL BCARD(NOPT,PRDR)
CALL CCARD(NOPT,PRDR,NPREV)
CALL DCARD(NOPT,PRDR)
CALL ECARD(NOPT,PRDR,IPSV)
CALL GCARD(NOPT,PRDR)
CALL HCARD(NOPT,PRDR,NROCK)
CALL ICARD(NOPT,PRDR,NCLFT,NHAUL,NPREV2)
CALL JCARD(NOPT,PRDR,NCLFT,NHAUL,NPREV3)
CALL KCARD(NOPT,PRDR,NCLFT,NHAUL,NPREV4)
CALL LCARD(NOPT,PRDR,NCLFT,NHAUL,NPREV5)
CALL MCARD(NOPT,PRDR,NCLFT,NHAUL,NPREV8,NEX,NFILCL)
CALL NCARD(NOPT,PRDR)
CALL OCARD(NOPT,PRDR,COPT,NPREV9,NPREVO)
GO TO 9970
1000 GO TO 9991
999 STOP
9990 CONTINUE

MODIFY FILE

2222 WRITE(*,1111)
1111 FORMAT( ' ENTER THE CARD TYPE YOU WANT TO MODIFY ? ' )
READ(*,109)OPTION
109 FORMAT(A1)
IF (OPTION.EQ. 'A'.OR. OPTION.EQ. 'a') THEN
CALL ACARD(NOPT,PRDR,NHOLLW,NEX,NFILCL)
ELSE
IF (OPTION.EQ. 'B'.OR. OPTION.EQ. 'b') THEN
CALL BCARD(NOPT,PRDR)
ELSE
IF (OPTION.EQ. 'C'.OR. OPTION.EQ. 'c') THEN
CALL CCARD(NOPT,PRDR)
ELSE
IF (OPTION.EQ. 'D'.OR. OPTION.EQ. 'd') THEN
CALL DCARD(NOPT,PRDR,NPREV)
ELSE
IF (OPTION.EQ. 'E'.OR. OPTION.EQ. 'e') THEN
CALL ECARD(NOPT,PRDR)
ELSE
IF (OPTION.EQ. 'F'.OR. OPTION.EQ. 'f') THEN
CALL FCARD(NOPT,PRDR,IPSV)
ELSE
IF (OPTION.EQ. 'G'.OR. OPTION.EQ. 'g') THEN
CALL GCARD(NOPT,PRDR)
ELSE
IF (OPTION.EQ. 'H'.OR. OPTION.EQ. 'h') THEN
CALL HCARD(NOPT,PRDR,NROCK)
ELSE
IF (OPTION.EQ. 'I'.OR. OPTION.EQ. 'i') THEN
CALL ICARD(NOPT,PRDR,NCLFT,NHAUL,NPREV2)
ELSE
IF (OPTION.EQ. 'J'.OR. OPTION.EQ. 'j') THEN
CALL JCARD(NOPT,PRDR,NCLFT,NHAUL,NPREV3)
ELSE
IF (OPTION.EQ. 'K'.OR. OPTION.EQ. 'k') THEN
CALL KCARD(NOPT,PRDR,NCLFT,NHAUL,NPREV4)
ELSE
IF (OPTION.EQ. 'L'.OR. OPTION.EQ. 'l') THEN
CALL LCARD(NOPT,PRDR,NCLFT,NHAUL,NPREV5)
ELSE
IF (OPTION.EQ. 'M'.OR. OPTION.EQ. 'm') THEN
CALL MCARD(NOPT,PRDR,NCLFT,NHAUL,NPREV8,NEX,NFILCL)

146
ELSE
   IF (OPTION.EQ. 'N'. OR. OPTION.EQ. 'n')
   THEN CALL NCARD(NOPT, PRDR)
   ELSE IF (OPTION.EQ. 'O'. OR. OPTION.EQ. 'o')
      THEN CALL OCARD(NOPT, PRDR, COPT, NPREV9, NPREVO)
      END IF
   END IF
END IF
END IF
END IF
END IF
END IF
END IF
END IF
END IF
END IF
END IF
END IF
END IF
END IF
END IF
END IF
END IF
WRITE(*,*)' DO YOU WANT TO CHANGE MORE DATA (Y/N)'
READ(*,109) OPTION
IF (OPTION.EQ. 'Y'. OR. OPTION.EQ. 'y') GO TO 2222
GO TO 9970
C 9991 CONTINUE

WRITE(*,50)
WRITE(*,*)' ENTER THE NAME OF INPUT FILE'
READ(*, (A') FNAME
OPEN(3, FILE=FNAME, STATUS='OLD')
READ(3,*) ISEED, ISTOP, NBLOCK, NFILCL, NHOLLW, NEX, CTIM, CDEN, NSDY,
+ NDMH, CREVR
READ(3,*), NROCK, (PFAC(I), DENS(I), SWELF(I), EFFECT(I), I=1, NROCK)
READ(3,*), NPDR, ISDR, DB, DS, NSPACE(1), TMOVE
IF (ISDR.EQ. 2) THEN
   NS=NSPACE(1)
ELSE
   NS=PRDR(ISDR+1)
END IF
READ(3,*) (PRM(J), HIST(J), J=1, NS)
READ(3,*) NPBL, TRATE1, TRATE2, LTIME, DTIME, CONST, CAPB
READ(3,*) N1, NC1, ISTR, NSPACE(3), TEXP, DMP, HCTR, GVW, TOTR, GVC, TCTR,
+ NHAUL
IF(NHAUL.EQ. 0) THEN
   NPREV8=1
   NPREV=1
   NPREV2=1
   NPREV3=1
   NPREV4=1
   NPREV5=1
ELSE
IF (ISTR.EQ.2) THEN
   NS=NSPACE(3)
ELSE
   NS=PRDR(ISTR+1)
END IF
READ(3,*) PTR(J),HITR(J),J=1,NS
IF (NHAUL.EQ.0) THEN
READ(3,*) NGEAR,DA,RORI,DGR,AMAX,VMAX
READ(3,*) DHILL(I),I=1,DGR
READ(3,*) SPEED(J),RP(J),J=1,NGEAR
END IF
READ(3,*) N2,NC2,IPSV,NSPACE(4),CAPL,CCPL,AVSPD
IF (IPSV.EQ.2) THEN
   NS=NSPACE(4)
ELSE
   NS=PRDR(IPSV+1)
END IF
READ(3,*) PSI(J),HISV(J),CSV(J),HCSV(J),J=1,NS
DO 1799 I=1,NBLOC
   1799 READ(3,*) NLIF(I),DAREA(I),CAREA(I),CTIK(I),PSBLK(I),PSLFT(I)
   +,BPREP(I)
READ(3,*) TPREP(I),NST(I),NSTH(I),LNST(I),LNSTH(I),
   +BNST(I),BNSTH(I),I=1,NBLOC
DO 400 I=1,NBLOC
   K=NLIFT(I)
   DO 400 J=1,KK
   400 READ(3,*) NCrit(J),PRBLK(I,J),PRLFT(I,J),ADEP(I,J)
   +,FTIME(I,J),FTTIME2(I,J)
   KK=NLIFT(I)
   DO 401 J=1,KK
   401 READ(3,*) FBLS(I,J),FDOZ(I,J),FRCD(I,J)
   +,FINS(I,J),FROCK(I,J,K),K=1,MRoCK
   DO 3401 I=1,NBLOC
      JJ=NLIFT(I)
   3401 READ(3,*) BCF(I,J),J=1,JJ
   IF (NFILCL.NE.0) THEN
   DO 402 I=1,NFILCL
      READ(3,*) CFILCL(I),BTIME(I),NA,(NBFIL(I,J),J=1,NA)
   402 NAFIL(I)=NA
   END IF
   READ(3,*) (GFIL(I,J,1),FILL(I,J,1),J=1,NFILCL)
ELSE
   DO 404 I=1,NBLOC
      V=1,NFILCL
      READ(3,*) NS,(GFIL(I,J,K),FILL(I,J,K),K=1,NS)
   NSEGFL(I,J)=NS
END IF
END IF
IF (NHAUL.EQ.1) THEN
   DO 405 JJ=1,NBLOCK
      NF=NLIFT(JJ)
      DO 405 K=1,NF
         IF (NCLFT(JJ,K).EQ.1) THEN
            READ(3,*) GALT(JJ,K,1),HALT(JJ,K,1)
         END IF
      CONTINUE
   ELSE
      DO 406 JJ=1,NBLOCK
         NF=NLIFT(JJ)
         DO 406 K=1,NF
            IF (NCLFT(JJ,K).EQ.1) THEN
               READ(3,*) NS, (GALT(JJ,K,L),HALT(JJ,K,L),L=1,NS)
            END IF
         CONTINUE
      END IF
   IF(NEX.NE.0) THEN
      IF (NHAUL.EQ.1) THEN
         DO 407 JJ=1,NBLOCK
            READ(3,*)(GEXST(JJ,K,1),HEXST(JJ,K,1),K=1,NEX)
         ELSE
            DO 408 JJ=1,NBLOCK
               DO 408 K=1,NEX
                  READ(3,*) NS, (GEXST(JJ,K,L),HEXST(JJ,K,L),L=1,NS)
               NSECAL(JJ,K)=NS
            END IF
         END IF
      IF(NHOLLW.NE.0) THEN
         IF (NHAUL.EQ.1) THEN
            DO 409 JJ=1,NBLOCK
               READ(3,*)(GHOLLW(JJ,K,1),HHOLLW(JJ,K,1),K=1,NHOLLW)
         ELSE
            DO 410 JJ=1,NBLOCK
               DO 410 K=1,NHOLLW
                  READ(3,*) NS, (GHOLLW(JJ,K,L),HHOLLW(JJ,K,L),L=1,NS)
               NSECHW(JJ,K)=NS
            END IF
         END IF
      IF(NEX.NE.0) THEN
         IF (NHAUL.EQ.1) THEN
            DO 407 JJ=1,NEX
               READ(3,*)(FGFIL(JJ,K,1),FFILL(JJ,K,1),K=1,NFILCL)
         ELSE
            DO 408 JJ=1,NEX
               DO 408 K=1,NFILCL
                  READ(3,*) NS, (FGFIL(JJ,K,L),FFILL(JJ,K,L),L=1,NS)
               NSECEF(JJ,K)=NS
            END IF
         END IF
      END IF
END IF
READ(3,*) CHCST,ZNHP, ZSAL, NHP, NSP, WELP, OBFT, EXRECL, TAREA, CSEED
READ(3,*) DDI, WHFT, PRMCST, AFCST, SLCOST, DETCST, BLCST, PMRT, COPT
DO 939 I=1,5
IF(COPT.EQ.1) THEN
  NPREV9=1
  READ(3,*) NCTIRE(I), DPRICE(I), RESVAL(I), DEPPER(I),
  + TIIFAC(I), FUELUP(I), FUELCG(I)
  READ(3,*) CLGPH(I), EXUMUL(I), BRF(I), SPL(I), NOPER(I), OPERHW(I)
  IF (NCTIRE(I).EQ.0) THEN
    READ(3,*) II(I), A(I), Z(I), BF(I)
  ELSE
    READ(3,*) TRCOST(I), ESTLIH(I)
  END IF
ELSE
  NPREV0=1
  READ(3,*) OPCOST(I), OWCOST(I)
END IF
939 CONTINUE
GO TO 9990

9970 CONTINUE
WRITE INTO FILE
WRITE(*,50) ENTER THE NAME OF OUTPUT FILE ' READ(*, (A)) FNAME
OPEN(4, FILE=FNAME, STATUS='NEW')
WRITE(4, 1900) ISEED, ISTOP, NBLOCK, NFILCL, NHOLLW,
+ NEX CTIM, CDEN, NSDY, NDML, CREVR
1900 FORMAT(6(I2), 7(I2), F9.4, 2(I2), I2, F5.3)
WRITE(4, 1901) NROCK, (PFAC(I), DENS(I), SWELF(I), EFFECT(I), I=1, NROCK)
1901 FORMAT(8(I2), 4(I2), F5.2)
WRITE(4, 1902) NPDR, ISDR, DB, DS, NSPACE(I), TMOVE
1902 FORMAT(I2, 1X, I2, I2, I2, F4.1, 1X, F4.1, 1X, I3, 1X, F5.2)
IF (ISDR.EQ.2) THEN
  NS=NSPACE(1)
ELSE
  NS=PRDR(ISDR+1)
END IF
WRITE(4, 1903) (PRM(J), HIST(J), J=1, NS)
1903 FORMAT(10(I2, F5.2))
WRITE(4, 1904) NPBL, TRATE1, TRATE2, LTIME, DTIME, CONST, CAPB
1904 FORMAT(12(I2, I2), F5.2, I2)
WRITE(4, 1905) N1, NC1, ISTR, NSPACE(3), TEX, DMP, HCTR, GVW, TOTR, GVC, TCTR,
+ NHAUL
6 FORMAT(4(I2, I3), 7(I2, F6.2), 1X, I1)
IF (ISTR.EQ.2) THEN
  NS=NSPACE(3)
ELSE
  NS=PRDR(ISTR+1)
END IF
WRITE(4, 1906) (PTR(J), HITR(J), J=1, NS)
IF (NHAUL.EQ.0) THEN
WRITE(4,1904)NGEAR,DA,RORI,DGR,AMAX,VMAX
WRITE(4,1903)(DHILL(I),I=1,NGEAR)
WRITE(4,1903)(SPEED(J),RP(J),J=1,NGEAR)
END IF
WRITE(4,6)N2,WC2,IPSV,NSPACE(4),CAPL,CCPL,AVSPD
IF (IPSV.EQ.2) THEN
  NS=NSPACE(4)
ELSE
  NS=PRDR(IPSV+1)
END IF
WRITE(4,1905)(PSV(J),HISV(J),CSV(J),HCSV(J),J=1,NS)
1905 FORMAT(1X,41X,F6.2)
DO 2799 I=1,NBLOCK
2799 WRITE(4,1907)(NLIFT(I),DAREA(I),CAREA(I),CTIK(I),PSBLK(I),PSLFT(I)
  +BPREP(I))
1907 FORMAT(I2,3(1X,F10.2),2(1X,I2),1X,F10.2)
DO 3799 I=1,NBLOCK
3799 WRITE(4,1908)(TPREP(I),NST(I),NSTH(I),LNST(I),LNSTH(I),BNST(I),
  +BNSTR(I))
1908 FORMAT(F10.2,6(1X,I2))
DO 7400 I=1,NBLOCK
  KK=NLIFT(I)
  DO 7400 J=1,KK
7400 WRITE(4,1909)(NCLF(I,J),PRBLK(I,J),PRLFT(I,J),ADEP(I,J)
  +BCF(I,J),J=1,JJ)
1909 FORMAT(3(1X,I2),3(1X,F8.2))
DO 7401 I=1,NBLOCK
  KK=NLIFT(I)
  DO 7401 J=1,KK
7401 WRITE(4,1910)(FBLAS(I,J),FDOZ(I,J),FRCD(I,J),FLNH(I,J)
  +(FROCK(I,J,K),K=1,NROCK)
1910 FORMAT(10(1X,F8.2))
DO 5700 I=1,NBLOCK
  J=NLIFT(I)
  IF(NFILCL.NE.0) THEN
    IF(NFILCL.NE.0) THEN
      DO 7402 I=1,NFILCL
        DO 7402 J=1,NFILCL
7402 WRITE(4,1911)(GFIL(I,J,K),FILL(I,J,K),K=1,NS)
1911 FORMAT(3,1X,10(1X,F8.2))
ELSE
    DO 7404 I=1,NBLOCK
      DO 7404 J=1,NFILCL
7404 WRITE(4,1912)(NS,(GFIL(I,J,K),FILL(I,J,K),K=1,NS)
1912 FORMAT(13,1X,10(1X,F8.2))
END IF
END IF
IF (NHAUL.EQ.1) THEN
  DO 7405 J=1,NBLOCK
    NF=NLIFT(J)
    DO 7405 K=1,NF

IF (NCLFT(JJ,K).EQ.1) THEN
  WRITE(4,1910)GALT(JJ,K,1),HALT(JJ,K,1)
END IF

7405 CONTINUE
ELSE
  DO 7406 JJ=1,NBLOCK
    NF=NLIFT(JJ)
    DO 7406 K=1,NF
      IF(NCLFT(JJ,K).EQ.1) THEN
        NS=NSECAL(JJ,K)
        WRITE(4,1912)NS,(GALT(JJ,K,L),HALT(JJ,K,L),L=1,NS)
      END IF
  END DO
END IF

IF(NEX.NE.0) THEN
  WRITE(4,1913)(CEST(I),I=1,NEX)
  1913 FORMAT(1X,F10.1)
ENDIF

IF(NHAUL.EQ.1) THEN
  DO 7407 JJ=1,NBLOCK
    WRITE(4,1910)(GEXST(JJ,K,1),HEXST(JJ,K,1),K=1,NEX)
  END DO
ELSE
  DO 7408 JJ=1,NBLOCK
    DO 7408 K=1,NEX
      NS=NSECEX(JJ,K)
      WRITE(4,1912)NS,(GEXST(JJ,K,L),HEXST(JJ,K,L),L=1,NS)
    END DO
  END DO
END IF

IF(NHOLLW.NE.0) THEN
  WRITE(4,1913)(CHOLLW(I),I=1,NHOLLW)
  IF(NHAUL.EQ.1) THEN
    DO 7409 JJ=1,NBLOCK
      WRITE(4,1910)(GHOLLW(JJ,K,1),HHOLLW(JJ,K,1),K=1,NHOLLW)
    END DO
ELSE
    DO 7410 JJ=1,NBLOCK
      DO 7410 K=1,NHOLLW
        NS=NSECHW(JJ,K)
        WRITE(4,1912)NS,(GHOLLW(JJ,K,L),HHOLLW(JJ,K,L),L=1,NS)
      END DO
    END DO
END IF

IF(NEX.NE.0) THEN
  IF(NHAUL.EQ.1) THEN
    DO 9407 JJ=1,NEX
      WRITE(4,1910)(FGFIL(JJ,K,1),FFILL(JJ,K,1),K=1,NFILCL)
    END DO
  ELSE
    DO 9408 JJ=1,NEX
      DO 9408 K=1,NFILCL
        NS=NSECEF(JJ,K)
        WRITE(4,1912)NS,(FGFIL(JJ,K,L),FFILL(JJ,K,L),L=1,NS)
      END DO
    END DO
  END IF
END IF

WRITE(4,7)CHCOST,ZNHS,ZNPS,WSFL,OBFT,EXRECL,TAREA,CSEED
7 FORMAT(3(1X,F10.2),2(1X,I2),5(1X,F9.3))
WRITE(4,9)DDI,WHFT,PRMCST,AFCST,SLCOST,DETCST,BLCST,PRRT,COPT
9 FORMAT(8(IX,F8.2),IX,I2)
DO 959 I=1,5
IF(COPT.EQ.1) THEN
WRITE(4,1917) NCTIRE(I),DPRICE(I),RESVAL(I),DEPPER(I),
1917 FORMAT(1X,F5.2),IX,I2,F6.2)
WRITE(4,3)CLEGPH(I),EXUMUL(I),BRF(I),SPL(I),NOPER(I),OPERHW(I)
3 FORMAT(4(IX,F5.2),IX,I2,F6.2)
IF(NCTIRE(I).EQ.0) THEN
WRITE(4,3)N(I),A(I),Z(I),BF(I)
ELSE
WRITE(4,399)TRCOST(I),ESTLIH(I)
399 FORMAT(2(X,F10.2)) END IF ELSE WRITE(4,3)OPCOST(I),OWCOST(I)
END IF
959 CONTINUE
CLOSE(4,STATUS='KEEP') STOP
END SUBROUTINE ECARD(NOPT,PRDR)
COMMONECOM/N2,NC2,IPSV,IPSV,CCPL,CCPL,AVSPD
COMMON/GEN/NSPACE(7),NBLOCK,NLIFT(10)
INTEGER PRDR(7)
WRITE(*,7)
7 FORMAT(100H)
WRITE(*,72)
72 FORMAT(100H)
WRITE(*,172)
172 FORMAT(100H)
IF (NOPT.EQ.1) THEN
WRITE(*,74)
74 FORMAT(100H)
READ(*,*)N2,NC2,IPSV,NSPACE(4),CAPL,CCPL,AVSPD
ELSE
WRITE(*,*)' HERE IS YOUR EXISTING DATA'
WRITE(*,11)N2,NC2,IPSV,NSPACE(4),CAPL,CCPL,AVSPD
FORMAT(3X,I2,3X,I3,5X,I3,3X,I6,3X,3(3X,F4.1))
READ(*,*)N2,NC2,IPSV,NSPACE(4),CAPL,CCPL,AVSPD
END IF
RETURN
END SUBROUTINE FCARD(NOPT PRDR IPSV)
COMMON/FCOM/PSV(20),HISV(20),CSV(20),HCSV(20)
COMMON/GEN/NSPACE(7),NBLK,NLFT(10)
INTEGER PRDR(7)
WRITE(*,81)
WRITE(*,182)
WRITE(*,82)
WRITE(*,83)
81 FORMAT(/,' PROBABILITY DISTRIBUTION DATA FOR LOADING TIME ',/,' <VARIABLE> DEFINITION ',/,' <PSV(J)> VALUE OF JTH POINT IF EMPIRICAL DISTRIBUTION INDICATED OTHERWISE JTH PARAMETER OF THE DISTRIBUTION INDICATED FOR OVERBURDEN LOADER. ',/,' <HISV(J)> CORRESPONDING CUMULATIVE PROBABILITY IF DISTRIBUTION IS EMPIRICAL OTHERWISE ENTER 0. ',/,' <CSV(J)> VALUE OF JTH POINT FOR LOADING CYCLE IF IPSV = 2 OTHERWISE JTH PARAMETER OF THE DISTRIBUTION INDICATED FOR COAL LOADER (MIN. ) ',/,' <HCSV(J)> CORRESPONDING CUMULATIVE PROBABILITY IF DISTRIBUTION IS EMPIRICAL OTHERWISE ENTER 0. ',/,' NOTE : J=1,NS WHERE NS = NSPACE(4) IF DISTRIBUTION EMPIRICAL OTHERWISE NS = NUMBER OF PARAMETER OF THE DISTRIBUTION'
IF (IPSV.EQ.2) THEN
  NS=NSPACE(4)
ELSE
  NS=PRDR(IPSV+1)
END IF
IF (NOPT.EQ.1) THEN
WRITE(*,*)' ENTER DATA IN THE FOLLOWING FORMAT: '
WRITE(*,84)
84 FORMAT(/,'PSV(J) HISV(J) CSV(J) HCSV(J);'/,'###.# #.## ###.# #.## ###.### #.## J=1,NS')
READ(*,*)(PSV(J),HISV(J),CSV(J),HCSV(J),J=1,NS)
ELSE
WRITE(*,*)' HERE IS YOUR EXISTING DATA'
WRITE(*,85) PSV(J), HISV(J), CSV(J), HCSV(J)
85 FORMAT(3X,F5.3,1X)
WRITE(*,88) PSV(J), HISV(J), CSV(J), HCSV(J), J=1,NS
WRITE(*,90) READ(*,85) PSV(J), HISV(J), CSV(J), HCSV(J), J=1,NS
END IF
RETURN
END

SUBROUTINE GCARD(NOPT,PRDR)
COMMON/GCOM/DAREA(10), CAREA(10), CTIK(I), PSBLK(I),
+ PSLFT(I), BPREP(I), JLIFT(I), TPREP(I), NST(10), NSTH(10)
+ , LNST(10), LNSTH(10), BNSST(10), BNSSTH(10)
COMMON/GEN/NSPACE(7), NBLDCK, NLIFT(10)
INTEGER PRDR(7), PSBLK, PSLFT, BNST, BNSSTH
WRITE(*,86)
WRITE(*,186)
WRITE(*,87)
WRITE(*,88)
WRITE(*,89)

86 FORMAT(/,'*=*=***************************************************************'
+ /, '*=*=INPUT DATA ABOUT BLOCKS*=*='
+ /, '*=*=***************************************************************'
+ /, '*=*=<VARIABLE> DEFINITION*=*='
+ /, '*=*=***************************************************************'
+ '/,'<NLIFT(I)> NUMBER OF LIFTS OF ITH BLOCK *=
+ /, '<DAREA(I)> AREA OF ITH BLOCK FOR OVERBURDEN *=
+ /, '<CAREA(I)> AREA OF COAL OF BLOCK I (SQ. FT.) *=
+ '/,'<CTIK(I)> COAL THICKNESS OF BLOCK I (FT) *=
+ '/,'<PSBLK(I)> BLOCK NUMBER THAT TO BE FINISHED *=
+ '/,'<PSLFT(I)> LIFT NUMBER THAT HAS TO BE FINISHED *=
+ '/,'<BPREP(I)> PREPARATION TIME FOR BLOCK I (HR.) *
+ '/,'NOTE: I=1,NBLDCK')

186 FORMAT(/,'*** INPUT DATA ABOUT BLOCKS ***'
+ /,'*** ***************************************************************'
+ /,'*** <VARIABLE> DEFINITION ***'
+ /,'*** ***************************************************************'

87 FORMAT(/,'***<NLIFT(I)> NUMBER OF LIFTS OF ITH BLOCK ***'
+ /,'***<DAREA(I)> AREA OF ITH BLOCK FOR OVERBURDEN ***'
+ /,'***<CAREA(I)> AREA OF COAL OF BLOCK I (SQ. FT.) ***'
+ /,'***<CTIK(I)> COAL THICKNESS OF BLOCK I (FT) ***'
+ /,'***<PSBLK(I)> BLOCK NUMBER THAT TO BE FINISHED ***'
+ /,'***<PSLFT(I)> LIFT NUMBER THAT HAS TO BE FINISHED ***'
+ /,'***<BPREP(I)> PREPARATION TIME FOR BLOCK I (HR.) ***'
+ /,'NOTE: I=1,NBLDCK')

88 FORMAT(/,'*** NUMBER OF LIFTS OF ITH BLOCK ***'
+ /,'*** AREA OF ITH BLOCK FOR OVERBURDEN ***'
+ /,'*** AREA OF COAL OF BLOCK I (SQ. FT.) ***'
+ /,'*** COAL THICKNESS OF BLOCK I (FT) ***'
+ /,'*** BLOCK NUMBER THAT TO BE FINISHED ***'
+ /,'*** LIFT NUMBER THAT HAS TO BE FINISHED ***'
+ /,'*** PREPARATION TIME FOR BLOCK I (HR.) ***'
+ /,'NOTE: I=1,NBLDCK')

97 FORMAT(/,'ENTER DATA IN THE FOLLOWING FORMAT: ' '
+ <|NLIFT(I)| DAREA(I) | CAREA(I) | CTIK(I) | PSBLK(I) | PSLFT(I) | BPREP(I) |
+ <|## | ### | ### | ### | ### | ### | ### |
+ READ(*,*) (NLIFT(I), DAREA(I), CAREA(I), CTIK(I), PSBLK(I), PSLFT(I), BPREP(I), I=1,NBLDCK)
ELSE
WRITE(*,90) HERE IS YOUR EXISTING DATA'
WRITE(*,90) NLIFT(I), DAREA(I), CAREA(I), CTIK(I), PSBLK(I), PSLFT(I)
+ BPREP(I)
DO 1799 I=1,NBLDCK
1799 WRITE(*,89) NLIFT(I), DAREA(I), CAREA(I), CTIK(I), PSBLK(I), PSLFT(I)

89 FORMAT(1X,15,3(1X,F8.1),1X,2I8,1X,F7.0)
WRITE(*,89) ENTER NEW DATA'
READ(*,*) (NLIFT(I), DAREA(I), CAREA(I), CTIK(I), PSBLK(I), PSLFT(I))
+ ,BPREP(I),I=1,NBLOCK)
END IF
WRITE(*,90)
WRITE(*,790)
90 FORMAT(************ TIME TO PREPARE BLOCK I FOR COAL************)
+ / * <TREP(I)> TIME TO PREPARE BLOCK I FOR COAL
+ / *<NST(I)> PRIMARY BACKFILL CELL NUMBER
+ / *(LOAD AND HAUL MODE).
790 FORMAT(************ PRIMARY BACKFILL AREA NUMBER************)
+ / *<LNST(I)> PRIMARY BACKFILL AREA NUMBER
+ / *(LOAD AND HAUL MODE).
+ / *<LNSTH(I)> PRIMARY BACKFILL AREA NUMBER
+ / *(LOADER CARRY MODE).
+ / *(DOZER PUSH MODE).
WRITE(*,1790)
1790 FORMAT(************ PRIMARY BACKFILL AREA NUMBER************)
+ / *<BNST(I)> PRIMARY BACKFILL AREA NUMBER
+ / *(DOZER PUSH MODE).
+ / *(DOZER PUSH MODE).
+ / *(DOZER PUSH MODE).
+ / ************ PRIMARY DISPOSAL AREA NUMBER************
+ / *<NSTH(I)> PRIMARY DISPOSAL AREA NUMBER
+ / *(LOADER AND HAUL MODE).
+ / *(LOADER AND HAUL MODE).
+ / *(DOZER PUSH MODE).
+ / *(DOZER PUSH MODE).
+ / ************ PRIMARY DISPOSAL AREA NUMBER************
+ / *<BNSTH(I)> PRIMARY DISPOSAL AREA NUMBER
+ / *(DOZER PUSH MODE).
+ / *(DOZER PUSH MODE).
+ / ************ PRIMARY BACKFILL CELL NUMBER************
+ / *<BNST(I)> PRIMARY BACKFILL CELL NUMBER
+ / *(DOZER PUSH MODE).
+ / *(DOZER PUSH MODE).
+ / ************ PRIMARY BACKFILL CELL NUMBER************
+ / *<BNSTH(I)> PRIMARY BACKFILL CELL NUMBER
+ / *(DOZER PUSH MODE).
+ / *(DOZER PUSH MODE).
+ / ************ PRIMARY BACKFILL CELL NUMBER************
+ / *<BNSTH(I)> PRIMARY BACKFILL CELL NUMBER
+ / *(DOZER PUSH MODE).
+ / *(DOZER PUSH MODE).
+ / ************ PRIMARY BACKFILL CELL NUMBER************
+ / *<BNSTH(I)> PRIMARY BACKFILL CELL NUMBER
+ / *(DOZER PUSH MODE).
+ / *(DOZER PUSH MODE).
+ / + / NOTE: I=1,NBLOCK /
+ / PRIMARY AREAS ARE AREAS FROM WHICH PROGRAM BEGINS ITS SEARCH
+ / TO FIND DESTINATION OF OVERBURDEN /
IF(NOPT.EQ.1) THEN
WRITE(*,*) ENTER DATA IN THE FOLLOWING FORMAT:
WRITE(*,*)
+ TREP(I),NST(I),NSTH(I),LNST(I),LNSTH(I),BNST(I),BNSTH(I)
WRITE(*,*)
+ ####### #### #### #### #### ####
READ(*,*) (TREP(I),NST(I),NSTH(I),LNST(I),LNSTH(I),BNST(I),
+BNSTH(I),I=1,NBLOCK)
ELSE
WRITE(*,*) HERE IS YOUR EXISTING DATA
WRITE(*,*)
+ TREP(I),NST(I),NSTH(I),LNST(I),LNSTH(I),BNST(I),BNSTH(I)
DO 1901 I=1,NBLOCK
1901 WRITE(*,11) TREP(I),NST(I),NSTH(I),LNST(I),LNSTH(I),BNST(I),
+BNSTH(I)
11 FORMAT(2X,F6.1,6(2X,I2))
WRITE(*,*) ENTER NEW DATA:
READ(*,*) (TREP(I),NST(I),NSTH(I),LNST(I),LNSTH(I),BNST(I),
+BNSTH(I),I=1,NBLOCK)
END IF
RETURN
END
SUBROUTINE HBCARD(NOPT,PRBLK,NROCK)
COMMON/HCOM/PRBLK(10,5),PRLT(10,5),ADEP(10,5),FBLAS(10,5),
+PDOZ(10,5),FPROCK(10,5),FCF10,5),NFALT(10,5),BCF(10,5),
+FRCO(10,5),FTIME(10,5),FTIME2(10,5)
COMMON/GEN/NSPACE(7),NBLOCK,NLIFT(10)
INTEGER PRDR(7),PRBLK,PRLT
WRITE(*,91)
WRITE(*,191)
WRITE(*,92)
WRITE(*,192)
WRITE(*,93)
91 FORMAT(//,*INPUT DATA ABOUT LIFTS*,'*','*<VARIABLE> DEFINITION*')
91 FORMAT(/','*<NCLFT(I,J)> CODE TO INDICATE IF LIFT J OF BLOCK I IS TO BE TREATED SPECIALLY FOR OVERBURDEN REMOVAL OPERATION*','*0-NO, 1-YES.)*')
92 FORMAT(',*<PRBLK(I,J)> BLOCK NUMBER THAT HAS TO BE STRIPPED BEFORE WORK ON LIFT J OF BLOCK I CAN BE STARTED.*')
92 FORMAT(',*<PRBLT(I,J)> LIFT NUMBER THAT HAS TO BE STRIPPED BEFORE WORK ON LIFT J OF BLOCK I CAN BE STARTED.*')
93 FORMAT(',*<ADEP(I,J)> DEPTH OF THE LIFT J OF BLOCK I (FT.)*')
93 FORMAT(//,'NOTE: THERE ARE NLIFT(I) NUMBER OF REPETITION OF DATA FOR BLOCK I AND THERE ARE NBLOCK NUMBER OF BLOCKS') IF(NOPT.EQ.1) THEN WRITE(*,'')\# NCLFT(I,J) PRBLK(I,J) PRLFT(I,J) ADEP(I,J) \# HERE IS YOUR EXISTING DATA\# DO 97 I=1,NBLOCK KK=NLIFT(I) DO 97 J=1,KK 97 READ(*,*)NCLFT(I,J),PRBLK(I,J),PRLFT(I,J),ADEP(I,J) ELSE WRITE(*,'')\# ENTER DATA IN THE FOLLOWING FORMAT:* WRITE(*,')\# NCLFT(I,J) PRBLK(I,J) PRLFT(I,J) ADEP(I,J) \# ENTER NEW DATA* DO 197 = BLOCK JJ=NLIFT(I,J) DO 197 J=1,KK 197 READ(*,*)NCLFT(I,J),PRBLK(I,J),PRLFT(I,J),ADEP(I,J) END IF WRITE(*,891)
891 FORMAT(//,'*<FTIME(I,J)> FRACTION OF LOADER TIME DOZER IS NEEDED FOR FEEDING OVERBURDEN TO LOADER (LOADER CARRY MODE).*')
891 FORMAT(//,'*<FTIME2(I,J)> FRACTION OF LOADER TIME DOZER IS NEEDED FOR FEEDING OVERBURDEN TO LOADER (LOAD AND HAUL MODE).*')
WRITE(*,*)   *        "NOTE: I=1,NBLOCK, J=1,NLIFT(I)"

158
IF(NOPT.EQ.1) THEN
  WRITE(*,*) : ENTER DATA IN THE FOLLOWING FORMAT: 
  WRITE(*,*) : FTIME(I,J) FTIME2(I,J) 
  DO 1898 I=1,NBLOCK 
  JJ=NLIFT(I)
  1898 READ(*,*) (FTIME(I,J),FTIME2(I,J),J=1,JJ)
  ELSE 
  WRITE(*,*) : HERE IS YOUR EXISTING DATA 
  WRITE(*,*) : FTIME(I,J) FTIME2(I,J) 
  DO 3898 I=1,NBLOCK 
  JJ=NLIFT(I)
  DO 3898 J=1,JJ
  3898 WRITE(*,4007) FTIME(I,J),FTIME2(I,J)
  4007 FORMAT(2(X,F10.2))
  WRITE(*,*) : ENTER NEW DATA: 
  DO 2898 I=1,NBLOCK 
  JJ=NLIFT(I)
  2898 READ(*,*) (FTIME(I,J),FTIME2(I,J),J=1,JJ)
  END IF
  WRITE(*,95)
  WRITE(*,96)
  WRITE(*,295)
  WRITE(*,296)

95 FORMAT(/,': * INPUT DATA ABOUT LIFTS 
+ /: * VARIABLE> DEFINITION 
+ /: * FABLAS(I,J) > FRACTION OF OVERBURDEN THAT HAS TO 
+ /: * BE BLASTED FOR BLOCK I LIFT J 
+ /: * FDOZ(I,J)  > FRACTION OF OVERBURDEN TO BE DOZED 
+ /: * FRCD(I,J) > FRACTION TO BE LOADED AND CARRIED 
+ /: * FLNH(I,J) > FRACTION TO BE LOADED AND HAULED 
+ /: * FROCK(I,J,K) > FRACTION OF KTH TYPE ROCK PRESENT 
+ /: * IN LIFT J AND BLOCK I. 
+ /: * THERE ARE NLIFT(I) NUMBER OF REPETITION OF DATA 
+ /: FOR BLOCK I AND THERE ARE NBLOCK NUMBER OF BLOCKS. FOR 
+ /: EACH LIFT THERE ARE NROCK NUMBER OF FROCK VALUE"

96 FORMAT(/,': * VARIABLE> DEFINITION 
+ /: * FABLAS(I,J) > FRACTION OF OVERBURDEN THAT HAS TO 
+ /: * BE BLASTED FOR BLOCK I LIFT J 
+ /: * FDOZ(I,J)  > FRACTION OF OVERBURDEN TO BE DOZED 
+ /: * FRCD(I,J) > FRACTION TO BE LOADED AND CARRIED 
+ /: * FLNH(I,J) > FRACTION TO BE LOADED AND HAULED 
+ /: * FROCK(I,J,K) > FRACTION OF KTH TYPE ROCK PRESENT 
+ /: * IN LIFT J AND BLOCK I. 
+ /: * THERE ARE NLIFT(I) NUMBER OF REPETITION OF DATA 
+ /: FOR BLOCK I AND THERE ARE NBLOCK NUMBER OF BLOCKS. FOR 
+ /: EACH LIFT THERE ARE NROCK NUMBER OF FROCK VALUE"

99 FORMAT(/,': ENTER DATA IN THE FOLLOWING FORMAT 
+ /: FABLAS(I,J) FDOZ(I,J) FRCD(I,J) FLNH(I,J) FROCK(I,J,K) 
+ /: * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 
DO 199 I=1,NBLOCK 
  KK=NLIFT(I)
  DO 199 J=1, KK
  199 READ(*,*) FBLAS(I,J),FDOZ(I,J),FRCD(I,J), 
  + FLNH(I,J),FROCK(I,J,K),K=1,NROCK)
  ELSE 
  WRITE(*,*) : HERE IS YOUR EXISTING DATA 
  WRITE(*,*)
1 5 9
+· FBLAS(I,J) FDOZ(I,J) FRCD(I,J) FLNH(I,J) FROCK(I,J,K) ....
DO 299 I=1,NBLOCK
J=NLIFT(I)
DO 299 J=1,JK
299 WRITE(*,200) FBLAS(I,J),FDOZ(I,J),FRCD(I,J),
+FLNH(I,J),(FROCK(I,J,K),K=1,NROCK)
DO 399 'I'=1,NBLOCK
KK=NLIFT(I)
DO 399 J=1,KK
399 READ(*,*) FBLAS(I,J),FDOZ(I,J),FRCD(I,J),
+FLNH(I,J),(FROCK(I,J,K),K=1,NROCK)
200 FORMAT(1X,5(1X,F9.4))
END IF
WRITE(*,9001)
9001 FORMAT(1X,5(1X,F9.4))
9000 READ(*,*)(BCF(I,J),J=1,JK)
ELSE
WRITE(*,*)' HERE IS YOUR EXISTING DATA: '
WRITE(*,*)' BCF(I,J) '
DO 9003 I=1,NBLOCK
JJ=NLIFT(I)
9003 WRITE(*,9004)(BCF(I,J),J=1,JK)
9004 FORMAT(5(1X,F8.2))
DO 9006 I=1,NBLOCK
JJ=NLIFT(I)
9006 READ(*,*)(BCF(I,J),J=1,JK)
END IF
END
SUBROUTINE ICARD(NOPT,PRDR,NCLFT,NHAUL,NPREV2)
COMMON/ICOM/NAPFL(10),NBFIL(10,5),FILL(10,10,5),
+NSECFL(10,10),GFLIL(10,10,5),CFILCL(10),NFILCL,BTIME(10)
COMMON/GEN/NSPACE(7),NBLOCK,NLIFT(10)
INTEGER PRDR,NSPACE(7),NCLFT(10,5)
IF(NFILCL.EQ.0.AND.NOPT.EQ.2) THEN
9000 WRITE(*,1000)
1000 FORMAT(1X,'NUMBER OF BACKFILLING AREAS IS ZERO. SO, BEFORE '
+ ' CHANGING OR CREATING THIS CARD YOU HAVE TO CHANGE NFILCL IN '
+ ' ACARD.')
RETURN
ELSE
IF(NFILCL.EQ.0) RETURN
END IF
WRITE(*,210)
WRITE(*,121)
WRITE(*,211)
210 FORMAT(/,,/ : **INPUT DATA FOR BACKFILL CELLS**
 : **DESCRIPTION**
 : **---------------------------------------------------------------*/
 *<CFILCL(I)> CAPACITY OF BACKFILL CELL I (CU. YD.)**
 *<BTIME(I)> TIME NEEDED TO GRADE BACKFILL CELL**
 *<NAFIL(I)> NUMBER OF BLOCK THAT HAS TO BE MINED**
 *<NBFIL(I,J)> BLOCK NUMBER OF THE BLOCKS THAT HAS**
 *TO BE MINED BEFORE BACKFILL CELL I**
 *IS AVAILABLE**

211 FORMAT(/,,/ : **NOTE: J=1,NAFIL(I)'**
 IF (NOT.P EQ .1) THEN
 WRITE(*,*): ENTER DATA IN THE FOLLOWING FORMAT:
 WRITE(*,*): CFILCL(I), BTIME(I), NAFIL(I), NBFIL(I,J)
 WRITE(*,*): ############. ###. ##. ##
 DO 212 I=1,NFILCL
 READ(*,*) CFILCL(I),BTIME(I),NA,(NBFIL(I,J)),J=1,NA
 212 NAFIL(I) = NA
 ELSE
 WRITE(*,*): HERE IS YOUR EXISTING DATA:
 WRITE(*,*): CFILCL(I), BTIME(I), NAFIL(I), NBFIL(I,1)
 DO 9911 I=1,NFILCL
 NA=NAFIL(I)
 9911 WRITE(*,213)CFILCL(I),BTIME(I),NAFIL(I),(NBFIL(I,J)),J=1,NA
 DO 1211 I=1,NFILCL
 READ(*,*) CFILCL(I),BTIME(I),NA,(NBFIL(I,J)),J=1,NA
 1211 NAFIL(I) = NA
 213 FORMAT(2(2X,F9.1),2X,I8,6(2X,I8))
 END IF
 IF(NHAUL EQ .1) THEN
 WRITE(*,220)
 WRITE(*,321)
 WRITE(*,221)
 ELSE
 WRITE(*,*): \*
 WRITE(*,220)
 WRITE(*,321)
 WRITE(*,221)
 END IF

220 FORMAT(/,,/ : **INPUT DATA ABOUT HAUL TIME**
 : **---------------------------------------------------------------*/
 *<GFIL(JJ,K,1)> HAUL TIME FROM BLOCK JJ TO**
 *<FILL(JJ,K,1)> DISTANCE OF BLOCK JJ TO BACKFILL**
 *<CELL K (FT.)> BACKFILL CELL K (MINUTE)**

221 FORMAT(/,*): \*
 NF=NFILCL
 IF(NHAUL EQ .1) THEN
 WRITE(*,*): ENTER DATA IN THE FOLLOWING FORMAT:
 WRITE(*,*): GFIL(JJ,K,1), FILL(JJ,K,1)
 WRITE(*,*): ############. ####
 DO 222 JJ=1,NBLOCK

READ(*,*) (GFIL(JJ, K, 1), FILL(JJ, K, 1), K=1, NF)
ELSE IF (NPREV2 .EQ. 0) THEN
   WRITE(*,*) 'HERE IS YOUR EXISTING DATA.'
   WRITE(*,*) GFIL(JJ, K, 1) FILL(JJ, K, 1)
   DO 997 JJ=1, NBLOCK
   WRITE(*,223) (GFIL(JJ, K, 1), FILL(JJ, K, 1), K=1, NF)
223   FORMAT(2(I1X,F7.2,5X))
   ELSE
      WRITE(*,*) 'NO DATA EXISTS. ENTER NEW DATA.'
      WRITE(*,*) GFILL(JJ, K, 1) FILL(JJ, 1)
      WRITE(*,*) '##.## ####.##'
   END IF
   DO 422 JJ=1, NBLOCK
   NF=NFILCL
422   READ(*,*) (GFIL(JJ, K, 1), FILL(JJ, K, 1), K=1, NF)
   END IF
ELSE
   WRITE(*, 231)
   WRITE(*, 351)
   WRITE(*, 232)
231   FORMAT(// '******************************************************************************')
      ' /: INPUT DATA FOR HAUL ROAD CHARACTERISTICS
      + /: NUMBER OF SECTION IN HAUL ROAD
      + /: FROM BLOCK JJ TO BACKFILL CELL K
531   FORMAT( /: GRADE OF SEGMENT NUMBER L OF BLOCK
      + /: JJ AND BACKFILL CELL K (%)
      + /: LENGTH OF SEGMENT L OF BLOCK
      + /: JJ AND BACKFILL CELL K (FEET)
      + /:******************************************************************************')
   ELSE
      WRITE(*, 631)
      WRITE(*, 831)
531   WRITE(*,233)NSECFL(JJ,K), THERE ARE NBLOCK *NFILCL NUMBER OF
      /: THIS DATA GROUP. REPEAT LAST TWO DATA NS TIMES.' /
      /: THIS DATA WILL NOT APPEAR IF NHAUL = 1')
   IF (NOPT .EQ. 1 .OR. NPREV2 .EQ. 0) THEN
      WRITE(*,731)
      WRITE(*,732)
      WRITE(*,733)
      WRITE(*,734)
      WRITE(*,735)
531   FORMAT( /: ENTER DATA IN THE FOLLOWING FORMAT: '
      + /: NSECFL(JJ, K) GFIL(JJ, K, L) FILL(JJ, K, L) '
      + /: #### ####
      DO 233 JJ=1, NBLOCK
      DO 233 K=1, NFILCL
      READ(*,*) NS, (GFIL(JJ, K, L), FILL(JJ, K, L), L=1, NS)
   233   NSECFL(JJ, K) = NS
      ELSE
      WRITE(*, 631)
531   WRITE(*,233)NSECFL(JJ,K),(GFIL(JJ,K,L),FILL(JJ,K,L),L=1,NS)
      WRITE(*, 234) NSECFL(JJ, K), (GFIL(JJ, K, L), FILL(JJ, K, L), L=1, NS)
531   FORMAT(1X,I2,(1X,F7.2))
   WRITE(*,*) 'ENTER NEW DATA.'
   DO 933 JJ=1, NBLOCK
933   READ(*,*) (GFIL(JJ, K, 1), FILL(JJ, K, 1), K=1, NF)
   END IF
DO 3222 JJ=1,NBLOCK
NF=NLIFT(JJ)
DO 3222 K=1,NF
3222 IF(NCLFT(JJ,K).EQ.1) WRITE(*,1222) GALT(JJ,K,1),HALT(JJ,K,1)
WRITE(*,*) ENTER NEW DATA
DO 4223 JJ=1,NBLOCK
NF=NLIFT(JJ)
DO 4223 K=1,NF
4223 IF(NCLFT(JJ,K).EQ.1) READ(*,*) GALT(JJ,K,1),HALT(JJ,K,1)
1222 FORM(2(2X,F7.2,5X)) END IF
ELSE
WRITE(*,231)
WRITE(*,561)
WRITE(*,232)
231 FORM(//' ******************************************************************************
+ //  * INPUT DATA FOR HAUL ROAD CHARACTERISTICS  *
+ //  ******************************************************************************
+ //  * <NSECAL(JJ,K)> NUMBER OF SECTION IN HAUL ROAD  *
+ //  * FROM BLOCK JJ AND LIFT K TO ITS  *
+ //  * DESTINATION.  *
+ //  * <GALT(JJ,K,L)> GRADE OF SEGMENT NUMBER L OF BLOCK*'
561 FORM(// JJ AND LIFT K (%).  *
+ //  * <HALT(JJ,K,L)> LENGTH OF SEGMENT NUMBER L OF  *
+ //  * BLOCK JJ AND LIFT K (FEET)  *
+ ******************************************************************************
232 FORM(// L = 1,NSECAL(JJ,K). REPEAT DATA GROUP FOR EACH BLOCK
+ AND FOR EACH BLOCK I NLIFT(I) TIMES
+ \ REPEAT LAST TWO DATA NSECAL(JJ,K) TIMES.'
+ \ THIS TABLE WILL NOT APPEAR IF NHAUL= 1. )
IF(NOPT.EQ.1 OR NPREV3.EQ.0) THEN WRITE(*,562)
562 FORM(//' ENTER DATA IN THE FOLLOWING FORMAT:  '/
+ / NSECAL(JJ,K),GALT(JJ,K,L),HALT(JJ,K,L)  '
+ / # # # # # # # # #  '
DO 233 JJ=1,NBLOCK
  KK=NLIFT(JJ)
DO 233 K=1, KK
IF(NCLFT(JJ,K).EQ.1) THEN
READ(*,*) NS,(GALT(JJ,K,L),HALT(JJ,K,L),L=1,NS)
NSECAL(JJ,K) = NS
END IF
233 CONTINUE
ELSE
WRITE(*,*) 'HERE IS YOUR EXISTING DATA'
WRITE(*,*) NSECAL GALT(JJ,K,L),HALT(JJ,K,L)  
DO 433 JJ=1,NBLOCK
KK=NLIFT(JJ)
DO 433 K=1, KK
IF(NCLFT(JJ,K).EQ.1) THEN
NS=NSECAL(JJ,K)
WRITE(*,234)NSECAL(JJ,K),(GALT(JJ,K,L),HALT(JJ,K,L),L=1,NS)
234 FORM(1X,12,8(1X,F7.2))
END IF
433 CONTINUE
DO 333 JJ=1,NBLOCK
   KK= NLIFT(JJ)
DO 333 K=1, KK
IF(NCLFT(JJ,K).EQ.1) THEN
   READ(*,*)NS,(GALT(JJ,K,L),HALT(JJ,K,L),L=1,NS)
   NSECAL(JJ,K)=NS
END IF
333 CONTINUE
NPREV3=1
END IF
END IF
RETURN
SUBROUTINE ACARD(NOPT,PRDR,NHOLLW,NEX,NFILCL)
COMMON/ACOM/I SEED,ISTOP,CTIM,CDEN,
+NSDY,NDMH,NROCK,PFAC(5),DENS(5),SELF(5),CREVR,EFFECT(5)
COMMON/GEN/NSPACE(7),NBLOCK,NLIFT(10)
INTEGER PRDR(7)
WRITE(*,7)
7 FORMAT(/'/','')
WRITE(*,71)
WRITE(*,72)
WRITE(*,73)
WRITE(*,74)
WRITE(*,75)
71 FORMAT(/'/','GENERAL INPUT DATA','/','** ', 'VARIABLE', ' DEFINITION', ' )
+ '/','**', 'VARIABLE', ' DEFINITION', ' )
+ '/','**', 'VARIABLE', ' DEFINITION', ' )
72 FORMAT(/'/','**', 'I SEED', ' INITIAL SEED VALUE (ANY PRIME NUMBER, ', ' )
+ '/','**', 'ISTOP', ' CODE TO INDICATE IF ANTHETIC SAMPLING ', ' )
+ '/','**', 'SHOULD BE USED. (0 IF NO, 1 IF YES) ', ' )
73 FORMAT(/'/','**', 'NBLOCK', ' NUMBER OF BLOCKS ', ' )
+ '/','**', 'NFILCL', ' NUMBER OF BACKFILL CELLS ', ' )
74 FORMAT(/'/','**', 'NHOLLW', ' NUMBER OF EXTRA SPOIL DISPOSAL AREAS ', ' )
+ '/','**', 'NEX', ' NUMBER OF EXTRA SPOIL STORAGE AREAS ', ' )
+ '/','**', 'CHT', ' SHIFITIME (MIN.) ', ' )
+ '/','**', 'CDEN', ' DENSITY OF COAL (TONS/CU FT. ) ', ' )
+ '/','**', 'NSDY', ' NUMBER OF SHIFITS PER DAY', ' )
75 FORMAT(/'/','**', 'NDMH', ' NUMBER OF DAYS PER MONTH ', ' )
+ '/','**', 'CREVR', ' PERCENTAGE OF COAL RECOVERABLE ', ' )
IF(NOPT.EQ.1) THEN
   WRITE(*,76)
76 FORMAT(/'/','ENTER THE DATA IN THE FOLLOWING FORMAT: ', ' )
+ '/','## ## ## ## ## ##.## ## ## ### ', ' )
+ '/','## ## ## ## ## ##.## ## ## ### ', ' )
   READ(*,*)ISEED,ISTOP,NBLOCK,NFILCL,NHOLLW,NEX,CTIM,CDEN,NSDY,
+NDMH,CREVR
ELSE
   WRITE(*,77)
165

77 FORMAT(/' HERE IS YOUR EXISTING DATA:
+ I SEED, ISTOP, NBLOCK, NFILCL, NHOLLW, NEX, CTIM, CDEN, NSDY, NDMH,
+ CREVR')
WRITE(*,78) SEED, ISTOP, NBLOCK, NFILCL, NHOLLW, NEX, CTIM, CDEN, NSDY,
+ NDMH, CREVR
78 FORMAT(2(Ix,I5),3(Ix,I6),1x,1x,F4.0,1x,F4.3,2(Ix,I4),2x,F5.3)
WRITE(*,78) SEED, ISTOP, NBLOCK, NFILCL, NHOLLW, NEX, CTIM, CDEN, NSDY,
+ NDMH, CREVR
END IF
WRITE(*,41)
WRITE(*,42)
41 FORMAT(/'***************************************************************
+ /* INPUT DATA A (CONTINUED) */
+ ***************************************************************
42 FORMAT(/' * <NROCK> NUMBER OF TYPES OF ROCK. */
+ * <PFAC(I)> POWDER FACTOR OF ITH TYPE OF ROCK. */
+ * <DENS(I)> DENSITY OF ITH TYPE ROCK (TONS/ CU. YD).*/
+ * <SWELF(I)> SWELLING FACTOR OF ITH TYPE OF ROCK. */
+ * <EFFECT(I)> FACTOR FOR DRILLING RATE OF ITH TYPE */
+ */
+ ***************************************************************
+ /* NOTE: I=1,NROCK*/
IF(NOPT.EQ.1) THEN
WRITE(*,43)
43 FORMAT(/NROCK PFAC(I) DENS(I) SWELF(I) EFFECT(I);I=1,NROCK)
ELSE
WRITE(*,46)
46 FORMAT(/' HERE IS YOUR EXISTING DATA: ')
WRITE(*,43) NROCK, (PFAC(I), DENS(I), SWELF(I), EFFECT(I)), I=1,NROCK
END IF
RETURN
END SUBROUTINE BCARD(NOPT, PRDR)
COMMON/BCOM/NPDR, ISDR, DS, PRM(20), HIST(20), TMOVE
COMMON/GEN/NSPACE(7), NBLOCK, NLIFT(10)
INTEGER PRDR(7)
WRITE(*,7)
WRITE(*,81)
WRITE(*,82)
WRITE(*,83)
81 FORMAT(/'***************************************************************
+ /* INPUT DATA FOR DRILLS */
+ ***************************************************************
+ /* <VARIABLE> DEFINITION */
+ /* ***************************************************************
82 FORMAT(/' * <NPDR> NUMBER OF DRILLS */
166

+ * <ISDR> CODE FOR THE PROBABILITY DISTRIBUTION*
+ * OF DRILLING RATE *
+ * <DB> BURDEN (FT.) *
+ * <DS> SPACING (FT.) *
+ * <NAMESPACE(1)> NUMBER OF DATA POINTS IF EMPIRICAL *
+ * DISTRIBUTION INDICATED. OTHERWISE *
+ * ENTER 0 *
+ * <TMOVE> AVERAGE TIME FOR DRILL TO MOVE FROM *
+ * ONE HOLE TO ANOTHER (MIN). *
+ * ********************************************************************

83 FORMAT( *DISTRIBUTION INDICATED. OTHERWISE *
I |
+ * ENTER O *
+ * * TMOVE AVERAGE TIME FOR DRILL TO MOVE FROM *
+ * ONE HOLE TO ANOTHER (MIN). *
+ *
+ *) IF(NOPT.EQ.1) THEN
WRITE(*,41)
41 FORMAT(" ENTER THE DATA IN THE FOLLOWING FORMAT: '/
+ NPDR ISDR DB DS NAMESPACE(1) TMOVE '/
+ ## ## ##.## ##.## # # # # # # # # ")
READ(*,*)NPDR,ISDR,DB,DS,NAMESPACE(1),TMOVE
ELSE
WRITE(*,38)NPDR,ISDR,DB,DS,NAMESPACE(1),TMOVE
38 FORMAT( HERE IS YOUR EXISTING DATA: '/
+ NPDR ISDR DB DS NAMESPACE(1) TMOVE '/
+2(2X,I4),2(2X,F4.1),5X,I2,4X,F5.2/ 'ENTER NEW VALUE')
READ(*,*)NPDR,ISDR,DB,DS,NAMESPACE(1),TMOVE
END IF
WRITE(*,85)
WRITE(*,86)
WRITE(*,87)
WRITE(*,89)
85 FORMAT( ********************************************************************
+ * PROBABILITY DISTRIBUTION DATA FOR DRILLING RATE *
+ * ********************************************************************
+ * DEFINITION *
+ *)
86 FORMAT( ********************************************************************
+ * <PRM(J)> IF ISDR = 2 THEN VALUE OF Jth POINT *
+ * FOR DRILLING RATE. IF ISDR IS NOT 2 *
+ * THEN Jth PARAMETER OF THE DISTRIBUTION *
+ * INDICATED. *
+ *)
87 FORMAT( ********************************************************************
+ * <HIST(J)> CORRESPONDING CUMULATIVE PROBABILITY. *
+ * IF ISDR IS NOT 2 THEN ENTER 0 *
+ * ********************************************************************
+ *)
89 FORMAT( ********************************************************************
+ * NOTE: J = 1,MNM. MNM = NAMESPACE(1) IF ISDR = 2 *
+ * OTHERWISE MNM = NUMBER OF PARAMETERS *
+ * OF THE DISTRIBUTION INDICATED BY ISDR')
IF (ISDR .EQ. 2) THEN
MNM = NAMESPACE(1)
ELSE
MNM = PRDR(ISDR+1)
END IF
IF (NOPT.EQ.1) THEN
WRITE(*,91)
91 FORMAT( ********************************************************************
+ * ENTER THE DATA IN THE FOLLOWING FORMAT: ' *
+ * <PRM(J)> HIST(J) '
+ * # # # # # # #
READ(*,*) (PRM(J),HIST(J),J=1,MNM)
ELSE
WRITE(*,187)
187 FORMAT( ' HERE IS YOUR EXISTING DATA ')
WRITE(*,188)(PRM(J),HIST(J),J=1,MNM)
188 FORMAT( ' PRM(J) HIST(J)
+/2(3X,F6.2))
WRITE(*,*) ENTER NEW DATA:
READ(*,*)(PRM(J),HIST(J),J=1,MNM)
END IF
RETURN
END
SUBROUTINE CCARD(NOPT,PRDR)
COMMON/CCOM/NPBL,TRATE1,TRATE2,LTIME,DTIME,CONST,CAPB
COMMON/GEN/NSPACE(7),NBLOCK,NLIFT(10)
REAL LTIME
INTEGER PRDR(7)
WRITE(*,*)
7 FORMAT(//)/--------/'
WRITE(*,92)
WRITE(*,93)
91 FORMAT(//)
92 FORMAT( '-----------------------------------------------------------------------------------'
+ / * DATA FOR BULDOZERS *'
+ / * <VARIABLE> DEFINITION *'
93 FORMAT( + / * <NPBL> NUMBER OF DOZERS *'
+ / * <TRATE1> TRAVELLING RATE FOR LOADED DOZER (FT/MIN). *'
+ / * <TRATE2> TRAVELLING RATE FOR EMPTY DOZER *'
+ / * <LTIME> LOADING TIME (MIN). *'
+ / * <DTIME> DUMPING TIME (MIN). *'
+ / * <CONST> TRAM CONSTANT (MIN). *'
+ / * <CAPB> BUCKET CAPACITY (CU.YD). *'
+ / *-----------------------------------------------------------------------------------'
IF(NOPT.EQ.1) THEN
WRITE(*,39)
39 FORMAT( ' ENTER DATA IN THE FOLLOWING FORMAT: '/
+ * NPBL TRATE1 TRATE2 LTIME DTIME CONST CAPB' /
+ *----------------------------------------------------------------------------------' )
READ(*,*)NPBL,TRATE1,TRATE2,LTIME,DTIME,CONST,CAPB
ELSE
WRITE(*,94)
94 FORMAT( ' HERE IS YOUR EXISTING DATA ')
WRITE(*,192)NPBL,TRATE1,TRATE2,LTIME,DTIME,CONST,CAPB
+ '/3X,14,6(3X,F5.2)/ ENTER NEW VALUE '
READ(*,*) NPBL,TRATE1,TRATE2,LTIME,DTIME,CONST,CAPB
END IF
RETURN
END
SUBROUTINE DCARD(NOPT,PRDR,NPREV)
COMMON/DCOM/N1,NG1,ISTR,TEX,DMF,HCTR,GVC,TCTR,THAUL
+ ,PTR(20),HITR(20),NGEAR,DA,ROI,DGR,AMAX,VMAX,DHILL(20)
+ ,SPEED(10),RP(10)
COMMON/GEN/NSPACE(7),NBLOCK,NLIFT(10)
INTEGER DGR,PRDR(7),DUMMY(10)
WRITE(*,7)
7 FORMAT(/'//://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://://:////
111 FORMAT(//:'******************************************************
+://: * PROBABILITY DISTRIBUTION DATA FOR DUMPING TIME *
+://: ******************************************************
+://: * <VARIABLE> DEFINITION *
+://: ******************************************************
+://:
112 FORMAT(:*: <PTR(J)> VALUE OF JTH POINT IF EMPERICAL *
+://: * DISTRIBUTION INDICATED OTHERWISE JTH *
+://: * PARAMETER OF THE DISTRIBUTION INDICATED *
+://: * FOR DUMPING TIMES. *
+://: * <HITR(J)> CORRESPONDING CUMULATIVE PROBABILITY *
+://: * IF EMPIRICAL DISTRIBUTION INDICATED, *
+://: * OTHERWISE ENTER 0. *
+://: ******************************************************
+://:
113 FORMAT(://: NOTE: J = 1,NS WHERE NS=NSPACE(3) IF DISTRIBUTION IS EMPERICAL'
+://: OTHERWISE 'NS = NUMBER OF PARAMETER OF THE DISTRIBUTION’/’ INDICATED. ’
+://: IF (ISTR.EQ.2) THEN
+://: NS = NSPACE(3)
+://: ELSE
+://: NS = PRDR(ISTR+1)
+://: END IF
+://: IF (NOPT.EQ.1) THEN
+://: WRITE(*,46)
+://: 46 FORMAT( //: ' ENTER DATA IN THE FOLLOWING FORMAT: '/
+://: ' PTR(J) HITR(J) '/
+://: )
+://: READ(*,*) (PTR(J),HITR(J),J=1,NS)
+://: ELSE
+://: WRITE(*,47)
+://: 47 FORMAT( ' HERE IS YOUR EXISTING DATA ')
+://: WRITE(*,1192)(PTR(J),HITR(J),J=1,NS)
+://:
+://: 1192 FORMAT( //: ' <NGEAR> NUMBER OF GEARS IN OVERBURDEN TRUCK *
+://: ' <DA> DEACCELERATION WHEN BRAKE USED *
+://: ' <BROI> ROLLING RESISTANCE OF THE MINE (%) *
+://: ' <DGR> MAXIMUM DOWNHILL GRADE (DEGREES) *
+://: ' <AMAX> MAXIMUM ACCELERATION ALLOWED *
+://: ' <VMAX> MAXIMUM VELOCITY ALLOWED (M.P.H) *
+://: WRITE(*,3114)
1114 FORMAT(’NOTE: THIS TABLE WILL NOT APPEAR IF NHAUL = 1.’)
IF (NOPT.EQ. 1) THEN
WRITE(*,51)
51 FORMAT(’ENTER DATA IN THE FOLLOWING FORMAT: ’/
+’ NGEAR DA RORI DGR AMAX VMAX’/
+’## #.## ##.# ##.# #.## ##.##’)
READ(*,*) NGEAR, DA, RORI, DGR, AMAX, VMAX
ELSE
IF (NPREV.EQ. 1) THEN
WRITE(*,*)’HERE IS YOUR EXISTING DATA’
WRITE(*,1112)(NGEAR, DA, RORI, DGR, AMAX, VMAX)
1112 FORMAT(’NGEAR DA RORI DGR AMAX VMAX’/
+’/1X,1X,F5.2,1X,F4.1,1X,F5.2,1X,F5.2/’
+’ENTER NEW DATA’)
ELSE
WRITE(*,*)’NO DATA EXISTS. ENTER DATA’
WRITE(*,1112)(NGEAR, DA, RORI, DGR, AMAX, VMAX)
END IF
WRITE(*,1114)
1114 FORMAT(’*/‘******************************************************************************’
+’/’*<VARIABLE> DEFINITION’*/’
+’/’*<VARIABLE> MAXIMUM DOWNHILL SPEED PERMISSIBLE’*/’
+’/’**********’)
DO 1110 J=1,DGR
1110 DUMMY(J)=J
1115 FORMAT(’/1X,’DHILL(‘,I2,’)’))
IF (NOPT.EQ. 1) THEN
WRITE(*,51)
61 FORMAT(’ENTER DATA IN THE FOLLOWING FORMAT: ’)
WRITE(*,1115)(DHILL(J), J=1,DGR)
WRITE(*,62)
62 FORMAT(’## #.## ##.# ##.# #.## ##.##’)
READ(*,*) (DHILL(J), J=1,DGR)
ELSE
IF (NPREV.EQ. 1) THEN
WRITE(*,*)’HERE IS YOUR EXISTING DATA’
WRITE(*,1115)(DHILL(J), J=1,DGR)
WRITE(*,1115)(DHILL(I), I=1,DGR)
1115 FORMAT(’/1X,F9.1)’/’ENTER NEW DATA’)
ELSE
WRITE(*,*)’NO DATA EXISTS. ENTER DATA.’
WRITE(*,1115)(DHILL(J), J=1,DGR)
END IF
READ(*,*) (DHILL(J), J=1,DGR)
END IF
WRITE(*,1118)
1118 FORMAT(’*/‘******************************************************************************’
+’/’*<SPEED(J)> SPEED LIMIT AT JTH GEAR (M.P.H.)’*/’
+’/’*<RP(J)> CORRESPONDING RIMPULL (LBS).’*/’
+’/’NOTE: J=1,NGEAR’/)

IF(NOPT.EQ.1) THEN
  DO 229 J=1,NGEAR
  DUMMY(J) = J
  117 FORMAT(1X,'SPEED('I2',' RP('I2'))
  WRITE(*,31)
  31 FORMAT(' ENTER DATA IN THE FOLLOWING FORMAT: ')
  WRITE(*,117)(DUMMY(J),DUMMY(J),J=1,NGEAR)
  WRITE(*,32)
  32 FORMAT(##.# ######.##.# ######.#### ....)
  READ(*,*)(SPEED(J),RP(J),J=1,NGEAR)
ELSE
  IF(NPREV.EQ.1) THEN
    WRITE(*,*)'HERE IS YOUR EXISTING DATA'
    WRITE(*,117)(DUMMY(J),DUMMY(J),J=1,NGEAR)
  WRITE(*,116)(SPEED(J),RP(J),J=1,NGEAR)
  ELSE WRITE(*,*)'NO DATA EXISTS. ENTER DATA.'
  END IF
  WRITE(*,*)(DUMMY(J),DUMMY(J),J=1,NGEAR)
  END IF
  WRITE(*,*)READ(*,*)(SPEED(J),RP(J),J=1,NGEAR)
  NPREV=1
  NPREV=1
  END IF
END
SUBROUTINE KCARD(NOPT,PRDR,NCLFT,NHAUL,NPREV4)
COMMON/KCOH/HEXST(10,5),GEXST(10,5),NSECX(10,5),CEXST(5),NEX
COMMON/GEN/NSPACE(7),NBLOCK,NLIFT(10)
INTEGER PRDR(7),NCLFT(10)
IF(NEX.EQ.0.AND.NOPT.EQ.2) THEN
  WRITE(*,1000)
  1000 FORMAT(//'
  ** IT IS NECESSARY TO CHANGE NEX IN AGARD BEFORE CHANGING THIS CARD')
  RETURN
ELSE
  END IF
  WRITE(*,1210)
  WRITE(*,1211)
  1210 FORMAT(//'INPUT DATA FOR EXTRA SPOIL STORAGE AREA /*
  ** '/'<VARIABLE> DESCRIPTION */
  ** <CEXST(I)> CAPACITY OF SPOIL STORAGE AREA I */
  ** <CU.YD.> */
  ** NOTE: I = 1,NEX /*
  IF(NOPT.EQ.1) THEN
  WRITE(*,2222)
  2222 FORMAT('ENTER DATA IN THE FOLLOWING FORMAT: ')
  CEXST(1) CEXST(2) .... CEXST(NEX)
  READ(*,*)(CEXST(I),I=1,NEX)
ELSE
WRITE( *, *, ' HERE IS YOUR EXISTING DATA ' )
WRITE( *, *, ' CEXST(1) CEXST(2) ......... CEXST( NEX ) ' )
WRITE( *, *, ' ENTER NEW DATA ' )
READ( *, * ) ( CEXST( I ), I = 1, NEX )
ENDIF

1213 FORMAT( 2X, F8.1 )
IF ( NHAUL . GT . 1 ) THEN
WRITE( *, 3220 )
WRITE( *, 3221 )
WRITE( *, 3222 )
3220 FORMAT( ' *********************************************** ' )
+ / * INPUT DATA ABOUT HAUL TIME * /
+ / * <GEXST(JJ,K,1)> HAUL TIME FROM BLOCK JJ TO *
+ / * <HEXST(JJ,K,1)> DISTANCE OF BLOCK JJ TO SPOIL *
+ / * STORAGE AREA K ( FEET ) *
+ / ***********************************************
3221 FORMAT( ' NOTE: THERE ARE NBLOCK * NEX NUMBER OF DATA PAIRS .' )
IF ( NOPT . EQ . 1 . OR . NPREV . EQ . 1 ) THEN
WRITE( *, 3221 )
3222 FORMAT( ' ENTER DATA IN THE FOLLOWING FORMAT: ' )
+ / ' GEXST( JJ,K,1 ) HEXST( JJ,K,1 ) ' + / ' ******* ******* ' )
DO 3222 JJ = 1, NBLOCK
3222 READ( *, * ) ( GEXST( JJ,K,1 ), HEXST( JJ,K,1 ), K = 1, NEX )
ELSE
WRITE( *, *, ' HERE IS YOUR EXISTING DATA ' )
WRITE( *, *, ' GEXST( JJ,K,1 ) HEXST( JJ,K,1 ) ' )
DO 3222 JJ = 1, NBLOCK
3222 WRITE( *, 3223 ) ( GEXST( JJ,K,1 ), HEXST( JJ,K,1 ), K = 1, NEX )
DO 3223 JJ = 1, NBLOCK
3223 READ( *, * ) ( GEXST( JJ,K,1 ), HEXST( JJ,K,1 ), K = 1, NEX )
3223 FORMAT( 2( 1X, F7.2, 6A ) )
ENDIF
ELSE
WRITE( *, 31 )
WRITE( *, 41 )
WRITE( *, 32 )
31 FORMAT( ' *********************************************** ' )
+ / * INPUT DATA FOR HAUL ROAD CHARACTERISTICS * /
+ / * *********************************************** ' + / * <NSECEX(JJ,K)> NUMBER OF SECTION IN HAUL ROAD *
+ / * FROM BLOCK JJ TO STORAGE AREA K. * )
41 FORMAT( ' * <GEXST(JJ,K,L)> GRADE OF SEGMENT NUMBER L OF BLOCK* ' + / * JJ AND SPOIL STORAGE AREA K ( % ) * ' + / * <HEXST(JJ,K,L)> LENGTH OF SEGMENT NUMBER L OF *
+ / * BLOCK JJ AND SPOIL STORAGE AREA K * ' + / * ( FEET ) * ' + / *********************************************** )
32 FORMAT( ' L=1, NSECEX(JJ,K). THERE ARE NBLOCK * NEX NUMBER OF ' + / ' THIS DATA GROUP. REPEAT LAST TWO DATA NSECEX(JJ,K)' ,
+ / ' TIMES ' )
IF(NOPT.EQ. 1.0 .OR. NPREV4.EQ.0) THEN
WRITE(*,*): ENTER DATA IN THE FOLLOWING FORMAT:
WRITE(*,*) : NSECEX(JJ,K) GEXST(JJ,K,L) HEXST(JJ,K,L)
WRITE(*,*) : # # # # #
DO 33 JJ=1,NBLOCK
DO 33 K=1,NEX
READ(*,NS,(GEXST(JJ,K,L),HEXST(JJ,K,L),L=1,NS)
33 NSECEX(JJ,K) = NS
ELSE
WRITE(*,*) : HERE IS YOUR EXISTING DATA
WRITE(*,*) : NSECEX(JJ,K) GEXST(JJ,K,L) HEXST(JJ,K,L)
DO 3233 JJ=1,NBLOCK
DO 3233 K=1,NEX
WRITE(*,3234)NSECEX(JJ,K),(GEXST(JJ,K,L),HEXST(JJ,K,L),L=1,NS)
3234 FORMAT( 1X,I6,6X,2(1X,F7.2,6X))
3233 CONTINUE
WRITE(*,*) : ENTER NEW DATA
DO 4233 JJ=1,NBLOCK
DO 4233 K=1,NEX
READ(*,*)NS,(GEXST(JJ,K,L),HEXST(JJ,K,L),L=1,NS)
4233 NSECEX(JJ,K)=NS
NPREV4=1
END IF
END IF
RETURN
END

SUBROUTINE LCARD(NOPT,PRDR,NCLEFT,NHAUL,NPREVS)
COMMON/LCOM/CHOLLW(S),GHOLLW(I0,.5,5),HOLLW(I0,5,5),
+NSECHW(I0,5),NHOLLW,COMMON/GEN/SPACE(7),NBLOCK,NLIFT(I0)
INTEGER PRDR(7),NCLEFT(I0,5)
IF(NHOLLW.EQ.0 .AND. NOPT.EQ.2) THEN
WRITE(*,1000)
1000 FORMAT( //': NUMBER OF EXTRA SPOIL DISPOSAL AREA IS ZERO.'
+': CHANGE NHOLLW IN ACARD BEFORE CHANGING THIS CARD.' )
RETURN
ELSE
WRITE(*,2210) RETURN
END IF
WRITE(*,2211)
2210 FORMAT( //':******************************************************************************
+ ': INPUT DATA FOR EXTRA SPOIL DISPOSAL AREA *
+ ':******************************************************************************')
2211 FORMAT( : 'DESCRIPTION **
+ ': ' <CHOLLW(I)> CAPACITY OF SPOIL DISPOSAL AREA I **
+ ': (CU.YD.). **
+ ': ***************************************************************************/
+ ': NOTE: I=1,NHOLLW **
IF(NOPT.EQ.1) THEN
WRITE(*,3232)
3232 FORMAT( //': ENTER DATA IN THE FOLLOWING FORMAT:
+ ': CHOLLW(1) CHOLLW(2) .......... CHOLLW(I) '
WRITE(*,*) 'HERE IS YOUR EXISTING DATA'
WRITE(*,*) CHOLLW(I)
WRITE(*,2213)(CHOLLW(I),I=1,NHOLLW)

2213 FORMAT(IX,F9.1)
WRITE(*,*) ENTER NEW DATA'
READ(*,*)CHOLLW(I),I=1,NHOLLW
END IF

IF(NHAUL.EQ.1) THEN
WRITE(*,2220)
WRITE(*,2221)

2220 FORMAT(/'INPUT DATA ABOUT HAUL TIME

+ /'HAIL TIME FROM BLOCK JJ TO
+ /'SPOIL DISPOSAL AREA K (MIN.)
+ /'DISTANCE OF BLOCK JJ TO SPOIL DISPOSAL AREA K (FEET).
+ /'NOTE: THERE ARE NBLOCK*NHOLLW NUMBER OF DATA FOR EACH
+ /'VARIABLE. ')

3444 FORMAT(/'ENTER DATA IN THE FOLLOWING FORMAT: '
+ /'GHOLLW(JJ,K,1) HHOLLW(JJ,K,1)
+ /#'#.###

DO 2222 JJ=1,NBLOCK

2222 READ(*,*)GHOLLW(JJ,K,1),HHOLLW(JJ,K,1),K=1,NF
ELSE
WRITE(*,*) 'HERE IS YOUR EXISTING DATA'
WRITE(*,*) GHOLLW(JJ,K,1) HHOLLW(JJ,K,1)
DO 2777 JJ=1,NBLOCK

2777 WRITE(*,2223)(GHOLLW(JJ,K,1),HHOLLW(JJ,K,1),K=1,NF)
WRITE(*,*) ENTER NEW DATA'
END IF
ELSE
WRITE(*,2231)
WRITE(*,444)
WRITE(*,2232)

231 FORMAT(/'INPUT DATA FOR HAUL ROAD CHARACTERISTICS

444 FORMAT(/'NUMBER OF SECTION IN HAUL ROAD
+ /'FROM BLOCK JJ TO DISPOSAL AREA K *
+ /'GRADE OF SEGMENT NUMBER L OF *
+ /'LENGTH OF SEGMENT NUMBER L OF *
+ /'BLOCK JJ AND DISPOSAL AREA K (FEET).')
175

+ FORMAT( '/ ', L=1,NSECW(JJ,K). THERE ARE NBLOCK *NHOLLW NUMBER OF'
+ / ' THIS DATA GROUP. REPEAT LAST TWO DATA'
+ / ') NSECW(JJ,K) TIMES
+ / ') IF(NOPT.EQ.1.OR.NPREV5.EQ.0) THEN
  WRITE(*,445)
  DO 2233 JJ=1,NBLOCK
  DO 2233 K=1,NHOLLW
  READ(*,*) NS,(GHOLLW(JJ,K,L),HHOLLW(JJ,K,L),L=1,NS)
  2233 NSECW(JJ,K) = NS
ELSE
WRITE(*,*)' HERE IS YOUR EXISTING DATA'
WRITE(*,*)' NSECW(JJ,K) GHOLLW(JJ,K,L) HHOLLW(JJ,K,L)
DO 2244 JJ=1,NBLOCK
DO 2244 K=1,NHOLLW
WRITE(*,2234)NSECW(JJ,K),(GHOLLW(JJ,K,L),HHOLLW(JJ,K,L),L=1,NS)
WRITE(*,*)' ENTER NEW DATA'
DO 2255 JJ=1,NBLOCK
DO 2255 K=1,NHOLLW
READ(*,*) NS,(GHOLLW(JJ,K,L),HHOLLW(JJ,K,L),L=1,NS)
2255 NSECW(JJ,K)=NS
NPREV5=1
2234 FORMAT(1X,I6,(6X,2(1X,F7.2,6X))) END IF END IF RETURN END

88~/~A~M,~~~if~b;s~~~F~~trfi~i5~fj~~~~~tfta:~ILCL) COMMON/GEN/NSPACE(7),NHOLLW
INTEGE~ PR0R(7),NCLF't(10
5) IF(NEX.EQ.O.AND.NOPT.EQ.2) THEN
WRITE(*,*)' NUMBER OF EXTRA SPOIL STORAGE AREA IS ZERO'
WRITE(*,*)' CHANGE NEX IN ACARD BEFORE CHANGING THIS CARD'
RETURN ELSE IF(NEX.EQ.O) RETURN END IF
IF(NHAUL.EQ.1) THEN
WRITE(*,3220)
WRITE(*,8220)
WRITE(*,3221)
3220 FORMAT( '/ ' 'INPUT DATA TYPE M'
+ / ' INPUT DATA ABOUT HAUL TIME'
+ / ') <FGFIL(JJ,K,1)> HAUL TIME FROM STORAGE AREA JJ
+ / ') <FFILL(JJ,K,1)> DISTANCE OF STORAGE AREA JJ TO
+ / ') BACKFILL CELL K (FEET).
8220 FORMAT( / *
+ /*:----------------------------------------------------------------------*/
3221 FORMAT( /: NOTE: THERE ARE NEX*NFILCL NUMBER OF DATA FOR EACH' 
+ /: VARIABLE' 
+ IF(NOPT.EQ.1.OR.NPREV8.EQ.1) THEN
WRITE(*,8221)
8221 FORMAT(/: ENTER DATA IN THE FOLLOWING FORMAT: 
+ /: *** FGFIL(JJ,K,1) FFILL(JJ,K,1)' 
+ /: ****#.#"# "# # "# #"# 
DO 3222 JJ=1,NEX
READ(*,*)(FGFIL(JJ,K,1),FFILL(JJ,K,1),K=1,NFILCL)
3222 ELSE WRITE(*,*): HERE IS YOUR EXISTING DATA' 
WRITE(*,*)' FGFIL(JJ,K,1) FFILL(JJ,K,1)' 
DO 3222 JJ=1,NEX
READ(*,*)(FGFIL(JJ,K,1),FFILL(JJ,K,1),K=1,NFILCL)
3223 FORMAT(2(1X,F7.2,6X))
END IF
ELSE WRITE(*,31) 
WRITE(*,41)
WRITE(*,32)
31 FORMAT(/: * INPUT DATA FOR HAUL ROAD CHARACTERISTICS
+ /: *----------------------------------------------------------------------*/
+ /: * <NSECEF(JJ,K)> NUMBER OF SECTION IN HAUL ROAD OF
+ /: * STORAGE AREA JJ TO BACKFILL CELL K
41 FORMAT( /: * FGFIL(JJ,K,L)> GRADE OF SEGMENT NUMBER L OF
+ /: * STORAGE AREA JJ AND BACKFILL CELL K (%)
+ /: * FFILL(JJ,K,L)> LENGTH OF SEGMENT NUMBER L OF
+ /: * STORAGE AREA JJ AND BACKFILL CELL K (FEET)
+ /: *-----------------------------------------------------------------------*/
32 FORMAT( /: NOTE: L=1 NSECEF(JJ,K). THERE ARE NEX*NFILCL NUMBER OF
+ /: THIS DATA GROUP. REPEAT LAST TWO DATA NSECEF(JJ,K) TIMES
IF(NOPT.EQ.1.OR.NPREV8.EQ.0) THEN
WRITE(*,*): ENTER DATA IN THE FOLLOWING FORMAT.'
WRITE(*,*)' NSECEF(JJ,K) FGFIL(JJ,K,L) FFILL(JJ,K,L) ' 
WRITE(*,*) '# # # # # # # # # #
DO 33 JJ=1,NEX
DO 33 K=1,NFILCL
READ(*,*) NS,(FGFIL(JJ,K,L),FFILL(JJ,K,L),L=1,NS)
NSECEF(JJ,K) = NS
33 ELSE WRITE(*,*): HERE IS YOUR EXISTING DATA.'
WRITE(*,*)' NSECEF(JJ,K) FGFIL(JJ,K,L) FFILL(JJ,K,L) ' 
DO 33 JJ=1,NEX
DO 33 K=1,NFILCL
WRITE(*,3234)NSECEF(JJ,K),(FGFIL(JJ,K,L),FFILL(JJ,K,L),L=1,NS)
3234 FORMAT(1X,16,6X,2(1X,F7.2,6X))
3233 CONTINUE
WRITE(*,*): ENTER NEW DATA'
DO 4233 JJ=1,NEX
DO 4233 K=1,NFILCL
READ(*,*)NS,(FGFIL(JJ,K,L),FFILL(JJ,K,L),L=1,NS)
NSECEF(JJ,K)=NS
NPREV8=1
END IF
END IF
RETURN
END
SUBROUTINE NCARD(NOPT,PRDR)
COMMON/NCOM/CHCOST,ZNHP,ZSAL,NHP,NSP,WELP,OBFT,EXRECL,TAREA,CSEED
+DI,WHT,PRMST,AFCST,SLCOST,DETCST,BLCST,PRMT,COPT
INTEGER PRDR(7),COPT
WRITE(*,101)
WRITE(*,108)
WRITE(*,102)
WRITE(*,4102)
101 FORMAT('******************************************************************
+  / *  INPUT COST DATA
+  / *  <VARIABLE>  DEFINITION
+  / *  ******************************************************************
108 FORMAT(+/ *  <CHCOST>  COAL HAULING COST ($/TON)
+  / *  <ZNHP>  HOURLY WAGE FOR HOURLY PERSONNEL($)
+  / *  <ZSAL>  SALARY OF SALARIED PERSONNEL ($/YEAR).
+  / *  <NSP>  NUMBER OF HOURLY PERSONNEL.
+  / *  <WELP>  WELFARE TRUST (%).
+  / *  <OBFT>  BENIFIT TRUST ($/TON).
+  / *  <EXRECL>  GRADING COST ($/ACRE).
+  / *  <TAREA>  TOTAL AREA OF MINE (ACRES).
  +  / *  <CSEED>  SEEDING COST ($/ACRE).
+******************************************************************
102 FORMAT(/ *  ENTER DATA IN THE FOLLOWING FORMAT
  / */  CHCOST ZNHP ZSAL NHP NSP WELP OBFT EXRECL TAREA CSEED
  / +*/
READ(*,*)CHCOST,ZNHP,ZSAL,NHP,NSP,WELP,OBFT,EXRECL,TAREA,CSEED ELSE
WRITE(*,*)' HERE IS YOUR EXISTING DATA'
WRITE(*,*)' CHCOST ZNHP ZSAL NHP NSP WELP OBFT EXRECL TAREA CSEE
+D'
WRITE(*,908)CHCOST,ZNHP,ZSAL,NHP,NSP,WELP,OBFT,EXRECL,TAREA,CSEED
908 FORMAT(1X,F6.2,1X,F4.2,1X,F6.0,2(1X,F13.6),2(1X,F4.2),1X,3(1X,F5.1))
WRITE(*,*)' ENTER NEW DATA'
READ(*,*)CHCOST,ZNHP,ZSAL,NHP,NSP,WELP,OBFT,EXRECL,TAREA,CSEED
ENDIF
WRITE(*,14)
WRITE(*,15)
WRITE(*,16)
4102 FORMAT(/' /'/) DATA TYPE M (CONTINUED)
+  / */
178

+ //  **********************************************************************
+ //  * <DDI>  DRILLING DIAMETER (FT.).  *
+ //  * <WHFT>  PERCENTAGE OF DRILL HOLE WET.  *
+ //  * <PRMCST>  PRIMER COST ($/100 CARTRIDGE).  *
+ //  * <AFGST>  AMMONIUM FUEL COST ($).  *
+ //  * <SLCST>  SLURRY COST ($/LB).  *
+ //  * <DETCST>  DETONATOR COST ($).  *
+ //  * <BLCST>  BLASTING AGENT COST ($).  *
+ //  * <PRMRT>  PRIMER TO BLASTING AGENT RATIO  *
+ //  * <COPT>  ENTER 1 IF COST DATA FOR EQUIPMENT WILL  *
+ //  * BE PROVIDED. ENTER 0 IF OWNING AND  *
+ //  * OPERATING COST WILL BE PROVIDED.  *
+ //  **********************************************************************

IF(NOPT.EQ.1) THEN
WRITE(*,111) ENTER DATA IN THE FOLLOWING FORMAT: '
WRITE(*,111)

111 FORMAT('DDI,WHFT,PRMCST,AFCST,SLCST,DETCST,BLCST,PRMRT,COPT')

WRITE(*,11)

ELSE
WRITE(*,111) HERE IS EXISTING DATA
WRITE(*,111)

READ(*,17)DDI,WHFT,PRMCST,AFCST,SLCST,DETCST,BLCST,PRMRT,COPT
WRITE(*,11) ENTER NEW DATA

READ(*,17)DDI,WHFT,PRMCST,AFCST,SLCST,DETCST,BLCST,PRMRT,COPT

END IF
RETURN

END

SUBROUTINE OCARD(NOPT,PRDR,COPT,NPREV9,NPREV0)
COMMON/OCOM,NCTIRE(6),BPRICE(6),RESVAL(6),DEPPER(6)
+TIIIFAC(6),FUELUP(6),FUELCS(6),CLGFPH(6),EXUMUL(6),BFR(6),SPL(6)
+NOPER(6),OPRHRW(6),I1(6),A(6),Z(6),BF(6),TRCOST(8),ESTL1H(6)
+OWCOST(6),OPOCST(6)

INTEGER PRDR,COPT
REAL I
DO 1 I=1,5
WRITE(*,7)
7 FORMAT('///////

IF(I.EQ.1) THEN
WRITE(*,7)
ELSE IF(I.EQ.2) THEN
WRITE(*,7)
ELSE IF(I.EQ.3) THEN
WRITE(*,7)
ELSE IF(I.EQ.4) THEN
WRITE(*,7)
ELSE WRITE(*,7) FOLLOWING INPUT FOR DOZER 

ELSE IF(I.EQ.2) THEN
WRITE(*,7)
ELSE IF(I.EQ.3) THEN
WRITE(*,7)
ELSE IF(I.EQ.4) THEN
WRITE(*,7)
ELSE WRITE(*,7) FOLLOWING INPUT FOR TRUCK (OVERBURDEN)
ELSE WRITE(*,7)
ELSE WRITE(*,7) FOLLOWING INPUT FOR LOADER(OVERBURDEN)
ELSE WRITE(*,7) FOLLOWING INPUT FOR LOADER(COAL)

RETURN
IF (CLOPT .EQ. 1) THEN
WRITE(*,111)
WRITE(*,311)
WRITE(*,113)
END IF
IF (NOPT .EQ. 0) THEN
WRITE(*,112)
ELSE
WRITE(*,122)
ENDIF
ENDIF
ENDIF
ENDIF
END IF

111 FORMAT(/)** INPUT COST DATA FOR EQUIPMENT AND SUPPLY */
+/* +/* <VARIABLE> DEFINITION */
311 FORMAT(/)
+/* <NCITIRE> CODE TO INDICATE IF TIRE MOUNTED */
+/* (0 - NO, 1-YES) */
+/* <DPRICE> DELIVERED PRICE ($/UNIT). */
+/* <RESVAL> RESALE VALUE ($/UNIT). */
+/* <DEPPER> DEPRECIATION PERIOD (HOURS) */
+/* <TIIFAC> INTEREST, INSURANCE AND TAX FACTOR */
+/* <FUELUP> FUEL PRICE ($/GALLON) */
+/* <FUELCS> FUEL CONSUMPTION (GALLON/HR.). */

IF (NOPT .EQ. 1 .OR. NPREV9.EQ.0) THEN
WRITE(*,114)
READ(*,*) NCTIRE(I),DPRICE(I),RESVAL(I),DEPPER(I),
+TIIFAC(I),FUELUP(I),FUELCS(I)
ELSE
WRITE(*,*) 'HERE IS YOUR EXISTING DATA'
WRITE(*,114)
READ(*,*) NCTIRE(I),DPRICE(I),RESVAL(I),
+DEPPER(I),TIIFAC(I),FUELUP(I),FUELCS(I)
WRITE(*,*) 'ENTER NEW DATA'
READ(*,*) NCTIRE(I),DPRICE(I),RESVAL(I),DEPPER(I),
+TIIFAC(I),FUELUP(I),FUELCS(I)
ENDIF

114 FORMAT(1X,17,2X,3(1X,F7.0),3(1X,F6.2))
END IF

121 FORMAT(/)** COST OF LUBRICANT, GREASE, ETC. ($/HOUR) */
+/* <CLFGPH> COST OF LUBRICANT, GREASE, ETC. ($/HOUR) */
+/* <EXUMUL> EXTENDED USE MULTIPLIER */
+/* <BRF> BASIC REPAIR FACTOR */
+/* <SPL> SPECIAL ITEM ($/HOUR) ($/HOUR) */
+/* <NOPER> NUMBER OF OPERATORS */
+/* <OPERHW> OPERATORS' WAGE ($/HOUR) */

IF (NOPT .EQ. 1 .OR. NPREV9.EQ.0) THEN
WRITE(*,*) 'ENTER DATA IN THE FOLLOWING FORMAT:
WRITE(*,122)
ELSE
WRITE(*,*) 'HERE IS YOUR EXISTING DATA'
WRITE(*,122)
WRITE(*,122)
WRITE(*,123) CLFGPH(I),EXUMUL(I),BRF(I),SPL(I),NOPER(I),OPERHW(I)
WRITE(*,124) ENTER NEW DATA
READ(*,*)CLFGPH(I),EXUMUL(I),BRF(I),SPL(I),NOPER(I),OPERHW(I)
END IF
123 FORMAT(2(1X,F6.2),2(1X,F5.2),1X,I3,3X,F6.2)
IF(NCTIRE(I).EQ.0) THEN
WRITE(*,126)
126 FORMAT(/'**************************************************************************'/
+ '*<VARIABLE> DEFINITION '*
+ '*<I>' IMPACT FACTOR '*
+ '*<A>' ABRASIVENESS FACTOR '*
+ '*<Z>' Z FACTOR '*
+ '*<BF>' BASIC FACTOR '*
+ '**************************************************************************'/
+ '/ NOTE: THIS TABLE WILL NOT APPEAR IF NCTIRE = 1 ')
IF(NOPT.EQ.1) THEN
WRITE(*,*) ENTER DATA IN THE FOLLOWING FORMAT:
 WRITE(*,127)
127 FORMAT(' I A Z BF ')
WRITE(*,127) I1(I),A(I),Z(I),BF(I)
ELSE
 WRITE(*,*)' HERE IS YOUR EXISTING DATA'
 WRITE(*,127)
 WRITE(*,128) I1(I),A(I),Z(I),BF(I)
 WRITE(*,*) ENTER NEW DATA
 READ(*,*)I1(I),A(I),Z(I),BF(I)
END IF
128 FORMAT(4(2X,F4.1)) ELSE
 WRITE(*,131)
131 FORMAT(/'**************************************************************************'/
+ '*<TRCOST> TIRE COST ($/TIRE) '*
+ '*<ESTLIH> ESTIMATED LIFE OF TIRES (HOUR) '*
+ '**************************************************************************'/
+ '/ IF(NOPT.EQ.1 OR NPREV9.EQ.0) THEN
 IF(NOPT.EQ.1 OR NPREV9.EQ.0) THEN
 WRITE(*,*) ENTER DATA IN THE FOLLOWING FORMAT:
 WRITE(*,*) TRCOST ESTLIH
 WRITE(*,*) TRCOST(I),ESTLIH(I)
 ELSE
 WRITE(*,*)' HERE IS YOUR EXISTING DATA'
 WRITE(*,*) TRCOST(I),ESTLIH(I)
 WRITE(*,*) TRCOST(I),ESTLIH(I)
 END IF
 END IF
132 FORMAT(2(1X,F6.0)) ELSE
 WRITE(*,194)
194 FORMAT('**************************************************************************')
OWNING AND OPERATING COST

* Operating Cost ($/HR) *

* Owning Cost ($/HR) *

IF(NOPT.EQ.1.OR.NPREV.EQ.0) THEN
WRITE('ENTER DATA IN THE FOLLOWING FORMAT:','OPCOST OWCAST')
WRITE('OPCOST OWCAST ','OPCOST OWCAST')
READ('OPCOST OWCAST') ELSE
WRITE('HERE IS YOUR EXISTING DATA','OPCOST OWCAST')
WRITE('OPCOST OWCAST')
119 FORMAT('IX,F5.1,2X,F5.1')
READ('OPCOST OWCAST') END IF

1 CONTINUE
IF(COPT.EQ.1) THEN
NPREV=1
ELSE
NPREV=1
END IF
RETURN
END
APPENDIX C

Program Output
**OPSIM SIMULATION SUMMARY**

**INPUT FILE: OPSIME.DAT**

**SIMULATION # 1 OF 1**

**OPERATION AND COST SUMMARY, BY BLOCK (OPERATING COST ONLY)**

<table>
<thead>
<tr>
<th>BLOCK NO.</th>
<th>LIFT NUMBER</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCK NO. 1</td>
<td>1</td>
<td>2</td>
<td>TOTAL</td>
<td></td>
</tr>
<tr>
<td><strong>BLOCK VOLUME (BCY)</strong></td>
<td>= 38300</td>
<td>= 49790</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AVERAGE SWELL</strong></td>
<td>= 1.30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STRIPING RATIO (BCY/TON)</strong></td>
<td>= 14.87</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DRILLING AND BLASTING**

<table>
<thead>
<tr>
<th>LIFT NUMBER</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Blasted (T)</strong></td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td><strong>Drilled (FT)</strong></td>
<td>2336</td>
<td>3399</td>
<td>5735</td>
</tr>
<tr>
<td><strong>Started at (HR)</strong></td>
<td>6.0</td>
<td>102.7</td>
<td></td>
</tr>
<tr>
<td><strong>Finished at (HR)</strong></td>
<td>41.9</td>
<td>153.6</td>
<td></td>
</tr>
<tr>
<td><strong>Time taken (HR)</strong></td>
<td>35.9</td>
<td>50.8</td>
<td>86.7</td>
</tr>
<tr>
<td><strong>Cost ($)</strong></td>
<td>975.2</td>
<td>1379.6</td>
<td>2354.7</td>
</tr>
<tr>
<td><strong>Cost ($/LCY)</strong></td>
<td>.06</td>
<td>.06</td>
<td>.06</td>
</tr>
</tbody>
</table>

**DOZER PUSH**

<table>
<thead>
<tr>
<th>LIFT NUMBER</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dozed (%)</strong></td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td><strong>Dozed (LCY)</strong></td>
<td>8111</td>
<td>11804</td>
<td>19916</td>
</tr>
<tr>
<td><strong>Started at (HR)</strong></td>
<td>28.6</td>
<td>41.7</td>
<td>70.3</td>
</tr>
<tr>
<td><strong>Finished at (HR)</strong></td>
<td>41.9</td>
<td>153.6</td>
<td></td>
</tr>
<tr>
<td><strong>Time taken (HR)</strong></td>
<td>1591.9</td>
<td>2316.4</td>
<td>3908.3</td>
</tr>
<tr>
<td><strong>Cost ($)</strong></td>
<td>.06</td>
<td>.06</td>
<td>.06</td>
</tr>
<tr>
<td><strong>Cost ($/LCY)</strong></td>
<td>.20</td>
<td>.20</td>
<td>.20</td>
</tr>
</tbody>
</table>

**LOADER CARRYING OVERBURDEN**

<table>
<thead>
<tr>
<th>LIFT NUMBER</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cared %</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Cared (LCY)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Started at (HR)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Finished at (HR)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Cycle time (MIN)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Travel time (MIN)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Cost ($)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Cost ($/LCY)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**DOZER FEEDING OVERBURDEN TO LOADER (LOADER CARRY)**

<table>
<thead>
<tr>
<th>LIFT NUMBER</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Started at (HR)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Finished at (HR)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Time taken (HR)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Cost ($)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Cost ($/LCY)</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**OVERBURDEN LOADING AND HAULING**

<table>
<thead>
<tr>
<th>LIFT NUMBER</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hauled (%)</strong></td>
<td>60</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td><strong>Hauled (LCY)</strong></td>
<td>12168</td>
<td>17706</td>
<td>29874</td>
</tr>
<tr>
<td><strong>Started at (HR)</strong></td>
<td>70.6</td>
<td>195.2</td>
<td></td>
</tr>
<tr>
<td><strong>Finished at (HR)</strong></td>
<td>96.7</td>
<td>233.3</td>
<td></td>
</tr>
<tr>
<td><strong>Time taken (HR)</strong></td>
<td>26.2</td>
<td>38.1</td>
<td>64.2</td>
</tr>
<tr>
<td><strong>Loader cycle time (MIN)</strong></td>
<td>.3</td>
<td>.3</td>
<td></td>
</tr>
<tr>
<td><strong>Truck cycle time (MIN)</strong></td>
<td>1.9</td>
<td>.9</td>
<td></td>
</tr>
<tr>
<td><strong>Loading time (MIN)</strong></td>
<td>.0</td>
<td>.0</td>
<td></td>
</tr>
<tr>
<td><strong>Forward haul time (MIN)</strong></td>
<td>.0</td>
<td>.0</td>
<td></td>
</tr>
<tr>
<td><strong>Waiting time (MIN)</strong></td>
<td>2.1</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td><strong>Dumping time (MIN)</strong></td>
<td>.6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td><strong>Cycle time (MIN)</strong></td>
<td>5.8</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td><strong>Cost ($)</strong></td>
<td>2068.7</td>
<td>3010.2</td>
<td>5078.9</td>
</tr>
<tr>
<td><strong>Cost ($/LCY)</strong></td>
<td>.17</td>
<td>.17</td>
<td>.17</td>
</tr>
</tbody>
</table>

**DOZER FEEDING OVERBURDEN TO LOADER**
<table>
<thead>
<tr>
<th>Project</th>
<th>Time Taken (HR)</th>
<th>Started at (HR)</th>
<th>Finished at (HR)</th>
<th>Cost ($)</th>
<th>Cost ($/LCY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coal Loading and Hauling</td>
<td>14.0</td>
<td>233.3</td>
<td>352.4</td>
<td>45.6</td>
<td>9433</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Goal</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Lifts in Block No. 2</td>
<td>2</td>
</tr>
<tr>
<td>Block Volume (BCY)</td>
<td>27196</td>
</tr>
<tr>
<td>Average Swell</td>
<td>1.30</td>
</tr>
<tr>
<td>Block Volume (LCY)</td>
<td>35355</td>
</tr>
<tr>
<td>Stripping Ratio (BCY/Ton)</td>
<td>17.22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lift Number</th>
<th>Block No. 2</th>
<th>Total</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Drilling and Blasting</th>
<th>Blasted (%)</th>
<th>Drilled (ft)</th>
<th>Started at (HR)</th>
<th>Finished at (HR)</th>
<th>Time Taken (HR)</th>
<th>Cost ($)</th>
<th>Cost ($/LCY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dozer Push</td>
<td>Dozed (%)</td>
<td>Dozed (LCY)</td>
<td>Started at (HR)</td>
<td>Finished at (HR)</td>
<td>Time Taken (HR)</td>
<td>Cost ($)</td>
<td>Cost ($/LCY)</td>
</tr>
<tr>
<td>Loader Carrying Overburden</td>
<td>Carried (%)</td>
<td>Carried (LCY)</td>
<td>Started at (HR)</td>
<td>Finished at (HR)</td>
<td>Time Taken (HR)</td>
<td>Cycle Time (MIN)</td>
<td>Travel Time (MIN)</td>
</tr>
<tr>
<td>Dozer Feeding Overburden to Loader (Loader Carry)</td>
<td>Started at (HR)</td>
<td>Finished at (HR)</td>
<td>Time Taken (HR)</td>
<td>Cost ($)</td>
<td>Cost ($/LCY)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overburden Loading and Hauling</td>
<td>Started at (HR)</td>
<td>Finished at (HR)</td>
<td>Time Taken (HR)</td>
<td>Loader Cycle Time</td>
<td>Truck Cycle Time</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LOADING TIME 1.9 1.9
FORWARD HAUL TIME 1.1 1.1
RETURN HAUL TIME 0.0 0.0
WAITING TIME 2.2 2.2
DUMPING TIME .6 .6

COST ($) 1603.8 2002.6 3606.4
COST ($/LCY) .17 .17 .17

DOZER FEEDING OVERBURDEN TO LOADER
STARTED AT (HR) .0 .0
FINISHED AT (HR) .0 .0
TIME TAKEN (HR) .0 .0
COST ($) .0 .0 .0
COST ($/LCY) .00 .00 .00

COAL LOADING AND HAULING
PRODUCED (TON) 1579
TIME TAKEN (HR) 10.2
STARTED AT (HR) 377.7
FINISHED AT (HR) 387.9
LOADER CYCLE TIME .3
TRUCK CYCLE TIME
LOADING TIME 1.7
HAUL TIME 12.0

COST ($/LCY) .00 .00 .00

TOTAL NUMBER OF LIFTS IN BLOCK NO. 3 = 2
BLOCK VOLUME (BCY) = 30887.
AVERAGE SWELL = 1.30
BLOCK VOLUME (LCY) = 40153.
STRIPPING RATIO (BCY/TON) = 14.18

LIFT NUMBER

BLOCK NO. 3

1 2

TOTAL

DRILLING AND BLASTING
BLASTED (%) 80 80 80
DRILLED (FT) 1945 2679 4625
FINISHED AT (HR) 289.7 382.2 70.9
TIME TAKEN (HR) 30.4 40.5 1164.6
COST ($) 826.0 1099.6 1925.6
COST ($/LCY) .06 .06 .06

DOZER PUSH
DOZED (%) 40 40 40
DOZED (LCY) 6756 9304 16061
FINISHED AT (HR) 289.7 382.2 70.9
TIME TAKEN (HR) 15.2 21.0 36.2
COST ($) 845.6 1164.6 2010.1
COST ($/LCY) .13 .13 .13

LOADER CARRYING OVERBURDEN
CARRIED (%) 0 0 0
CARRIED (LCY) 0 0 0
FINISHED AT (HR) .0 .0 .0
TIME TAKEN (HR) .0 .0 .0
COST ($) 0 0 0
COST ($/LCY) 0.00 0.00 0.00

DOZER FEEDING OVERBURDEN TO LOADER (LOADER CARRY)
STARTED AT (HR) .0 .0 .0
FINISHED AT (HR) .0 .0 .0
TIME TAKEN (HR) .0 .0 .0

COST ($) 826.0 1099.6 1925.6
COST ($/LCY) .06 .06 .06
COST ($) 0.0 0.0 0.0
COST ($/LCY) 0.00 0.00 0.00

OVERBURDEN LOADING AND HAULING

<table>
<thead>
<tr>
<th>HAULED (%)</th>
<th>HAULED (LCY)</th>
<th>10134</th>
<th>13957</th>
<th>24091</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>304.9</td>
<td>403.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>326.7</td>
<td>433.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21.8</td>
<td>30.9</td>
<td>51.8</td>
<td>51.8</td>
<td>51.8</td>
</tr>
<tr>
<td>0.3</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TRUCK CYCLE TIME

| LOADING TIME | 1.9 |
| RETURN HAUL TIME | 0.0 |
| WAITING TIME  | 2.7 |
| DUMPING TIME  | 0.6 |
| CYCLE TIME (MIN) | 5.8 |

| COST ($) | 1723.0 |
| COST ($/LCY) | 0.17 |

DOZER FEEDING OVERBURDEN TO LOADER

| STARTED AT (HR) | 0.0 |
| TIME TAKEN (HR) | 0.0 |
| COST ($)        | 0.0 |
| COST ($/LCY)    | 0.00 |

coal loading and hauling

| PRODUCED (TON) | 2178 |
| TIME TAKEN (HR) | 12.5 |
| 433.1 | 445.6 |
| 0.3 |
| 1.7 |
| 12.0 |
| CYCLE TIME (MIN) | 13.7 |

| TOTAL NUMBER OF LiftS IN BLOCK NO. 4 = 2 |
| BLOCK VOLUME (BCY) = 40384. |
| AVERAGE SWELL = 1.30 |
| BLOCK VOLUME (LCY) = 52500. |
| STRIPPING RATIO (BCY/TON) = 14.49 |

<table>
<thead>
<tr>
<th>LIFT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCK NO. 4</td>
</tr>
</tbody>
</table>

| BLOCK NO. | 4 |
| BLOCK VOLUME (BCY) | 40384. |
| AVERAGE SWELL | 1.30 |
| BLOCK VOLUME (LCY) | 52500. |
| STRIPPING RATIO (BCY/TON) | 14.49 |

<table>
<thead>
<tr>
<th>DRILLING AND BLASTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLASTED (%)</td>
</tr>
<tr>
<td>DRILLED (FT)</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOZER PUSHER</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOZED (%)</td>
</tr>
<tr>
<td>DOZED (LCY)</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOADER CARRYING OVERBURDEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARRIED (%)</td>
</tr>
<tr>
<td>CARRIED (LCY)</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

**DOZER FEEDING OVERBURDEN TO LOADER (LOADER CARRY)**

<table>
<thead>
<tr>
<th>STARTED AT (HR)</th>
<th>FINISHED AT (HR)</th>
<th>TIME TAKEN (HR)</th>
<th>COST ($)</th>
<th>COST ($/LCY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**OVERBURDEN LOADING AND HAULING**

<table>
<thead>
<tr>
<th>HAULED (%)</th>
<th>HAULED (LCY)</th>
<th>STARTED AT (HR)</th>
<th>FINISHED AT (HR)</th>
<th>TIME TAKEN (HR)</th>
<th>LOADER CYCLE TIME</th>
<th>TRUCK CYCLE TIME</th>
<th>LOADING TIME</th>
<th>FORWARD HAUL TIME</th>
<th>RETURN HAUL TIME</th>
<th>WAITING TIME</th>
<th>DUMPING TIME</th>
<th>CYCLE TIME (MIN)</th>
<th>COST ($)</th>
<th>COST ($/LCY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>502.4</td>
<td>665.1</td>
<td>30.5</td>
<td>3.3</td>
<td>3.3</td>
<td>1.9</td>
<td>1.9</td>
<td>0.6</td>
<td>2.9</td>
<td>0.6</td>
<td>5.8</td>
<td>2413.0</td>
<td>17</td>
</tr>
</tbody>
</table>

**DOZER FEEDING OVERBURDEN TO LOADER**

<table>
<thead>
<tr>
<th>STARTED AT (HR)</th>
<th>FINISHED AT (HR)</th>
<th>TIME TAKEN (HR)</th>
<th>COST ($)</th>
<th>COST ($/LCY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**COAL LOADING AND HAULING**

<table>
<thead>
<tr>
<th>PRODUCED (TON)</th>
<th>TIME TAKEN (HR)</th>
<th>STARTED AT (HR)</th>
<th>FINISHED AT (HR)</th>
<th>LOADER CYCLE TIME</th>
<th>TRUCK CYCLE TIME</th>
<th>LOADING TIME</th>
<th>HAUL TIME</th>
<th>CYCLE TIME (MIN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2786</td>
<td>16.8</td>
<td>665.1</td>
<td>679.9</td>
<td>0.3</td>
<td>1.7</td>
<td>12.0</td>
<td>13.7</td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL NUMBER OF LIFTS IN BLOCK NO. 5 = 1**

**BLOCK VOLUME (BCY) = 6067.**

**AVERAGE SWELL = 1.30**

**BLOCK VOLUME (LCY) = 7887.**

**STRIPPING RATIO (BCY/TON) = 6.81**

<table>
<thead>
<tr>
<th>BLOCK NO.</th>
<th>TOTAL LIFT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

**DRILLING AND BLASTING**

<table>
<thead>
<tr>
<th>BLASTED (%)</th>
<th>DRILLED (FT)</th>
<th>STARTED AT (HR)</th>
<th>FINISHED AT (HR)</th>
<th>TIME TAKEN (HR)</th>
<th>COST ($)</th>
<th>COST ($/LCY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>89</td>
<td>1023</td>
<td>388.2</td>
<td>408.1</td>
<td>15.9</td>
<td>430.9</td>
<td>0.06</td>
</tr>
</tbody>
</table>

**DOZER PUSH**

<table>
<thead>
<tr>
<th>DOZED (%)</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DOZER (LCY)</strong></td>
<td>3154</td>
</tr>
<tr>
<td><strong>TIME TAKEN (HR)</strong></td>
<td>12.2</td>
</tr>
<tr>
<td><strong>STARTED AT (HR)</strong></td>
<td>404.1</td>
</tr>
<tr>
<td><strong>FINISHED AT (HR)</strong></td>
<td>416.3</td>
</tr>
<tr>
<td><strong>COST ($)</strong></td>
<td>677.6</td>
</tr>
<tr>
<td><strong>COST ($/LCY)</strong></td>
<td>.21</td>
</tr>
</tbody>
</table>

| **LOADER CARRYING OVERBURDEN** |
| **CARRIED (%)** | 0 | 0 |
| **CARRIED (LCY)** | 0 | 0 |
| **STARTED AT (HR)** | .0 |
| **FINISHED AT (HR)** | .0 |
| **TIME TAKEN (HR)** | .0 |
| **CYCLE TIME (MIN)** | .0 |
| **TRAVEL TIME (MIN)** | .0 |
| **COST ($)** | .0 |
| **COST ($/LCY)** | .00 | .00 |

| **DOZER FEEDING OVERBURDEN TO LOADER (LOADER CARRY)** |
| **STARTED AT (HR)** | .0 |
| **FINISHED AT (HR)** | .0 |
| **TIME TAKEN (HR)** | .0 |
| **COST ($)** | .0 |
| **COST ($/LCY)** | .00 | .00 |

| **OVERBURDEN LOADING AND HAULING** |
| **HAULED (%)** | 69 | 70 |
| **HAULED (LCY)** | 5521 | 5521 |
| **STARTED AT (HR)** | 433.1 |
| **FINISHED AT (HR)** | 445.0 |
| **TIME TAKEN (HR)** | 11.9 |
| **LOADER CYCLE TIME** | .3 |
| **TRUCK CYCLE TIME** | |
| **LOADING TIME** | 1.9 |
| **FORWARD HAUL TIME** | 1.3 |
| **RETURN HAUL TIME** | .0 |
| **WAITING TIME** | 2.0 |
| **DUMPING TIME** | .6 |
| **CYCLE TIME (MIN)** | 5.8 |
| **COST ($)** | 938.7 |
| **COST ($/LCY)** | .17 | .17 |

| **DOZER FEEDING OVERBURDEN TO LOADER** |
| **STARTED AT (HR)** | .0 |
| **FINISHED AT (HR)** | .0 |
| **TIME TAKEN (HR)** | .0 |
| **COST ($)** | .0 |
| **COST ($/LCY)** | .00 | .00 |

| **COAL LOADING AND HAULING** |
| **PRODUCED (TON)** | 891 |
| **TIME TAKEN (HR)** | 11.0 |
| **STARTED AT (HR)** | 445.0 |
| **FINISHED AT (HR)** | 456.0 |
| **LOADER CYCLE TIME** | .3 |
| **TRUCK CYCLE TIME** | |
| **LOADING TIME** | 1.7 |
| **HAUL TIME** | 12.0 |
| **CYCLE TIME (MIN)** | 13.7 |

TOTAL NUMBER OF LIFTS IN BLOCK NO. 6 = 2
BLOCK VOLUME (BCY) = 33896.
AVERAGE SWELL = 1.30
BLOCK VOLUME (LCY) = 44065.
STRIPPING RATIO (BCY/TON) = 17.66

<p>| <strong>LIFT NUMBER</strong> |</p>
<table>
<thead>
<tr>
<th><strong>BLOCK NO.</strong></th>
<th>6</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
</table>
### Drilling and Blasting

<table>
<thead>
<tr>
<th>Drilled (ft)</th>
<th>2412</th>
<th>2934</th>
<th>5346</th>
</tr>
</thead>
<tbody>
<tr>
<td>Started at (hr)</td>
<td>594.3</td>
<td>707.6</td>
<td></td>
</tr>
<tr>
<td>Finished at (hr)</td>
<td>631.8</td>
<td>752.5</td>
<td></td>
</tr>
<tr>
<td>Time taken (hr)</td>
<td>37.6</td>
<td>44.8</td>
<td>82.4</td>
</tr>
<tr>
<td>Cost ($)</td>
<td>1019.8</td>
<td>1216.5</td>
<td>2236.3</td>
</tr>
<tr>
<td>Cost ($/LCY)</td>
<td>.06</td>
<td>.06</td>
<td>.06</td>
</tr>
</tbody>
</table>

### Dozer Push

| Dozed (%) | 30 | 30 | 30 |
| Dozed (LCY) | 5580 | 7638 | 13219 |
| Time taken (hr) | 9.8 | 30.6 | 40.4 |
| Started at (hr) | 631.8 | 752.5 | |
| Finished at (hr) | 641.6 | 783.1 | |
| Cost ($) | 543.8 | 1700.2 | 2244.0 |
| Cost ($/LCY) | .10 | .22 | .17 |

### Loader Carrying Overburden

| Carried (%) | 0 | 0 | 0 |
| Carried (LCY) | 0 | 0 | 0 |
| Started at (hr) | 0 | 0 | 0 |
| Finished at (hr) | 0 | 0 | 0 |
| Time taken (hr) | 0 | 0 | 0 |
| Cycle time (min) | 0 | 0 | 0 |
| Travel time (min) | 0 | 0 | 0 |
| Cost ($) | 0 | 0 | 0 |
| Cost ($/LCY) | 0.00 | 0.00 | 0.00 |

### Dozer Feeding Overburden to Loader (Loader Carry)

| Started at (hr) | 0 | 0 | 0 |
| Finished at (hr) | 0 | 0 | 0 |
| Time taken (hr) | 0 | 0 | 0 |
| Cost ($) | 0 | 0 | 0 |
| Cost ($/LCY) | 0.00 | 0.00 | 0.00 |

### Overburden Loading and Hauling

| Hauled (%) | 69 | 69 | 70 |
| Hauled (LCY) | 13022 | 17823 | 30845 |
| Started at (hr) | 665.1 | 783.1 | |
| Finished at (hr) | 693.1 | 821.4 | |
| Time taken (hr) | 28.0 | 38.3 | 66.3 |
| Loader cycle time | 3 | 3 |
| Truck cycle time | 1.9 | 1.9 |
| Loading time | 1.9 | 1.9 |
| Forward haul time | 1.1 | 1.1 |
| Return haul time | 0.0 | 0.0 |
| Waiting time | 2.2 | 2.2 |
| Dumping time | 6.6 | 6.6 |
| Cycle time (min) | 5.8 | 5.8 |
| Cost ($) | 2213.9 | 3030.1 | 5244.1 |
| Cost ($/LCY) | .17 | .17 | .17 |

### Dozer Feeding Overburden to Loader

| Started at (hr) | 0 | 0 | 0 |
| Finished at (hr) | 0 | 0 | 0 |
| Time taken (hr) | 0 | 0 | 0 |
| Cost ($) | 0 | 0 | 0 |
| Cost ($/LCY) | 0.00 | 0.00 | 0.00 |

### Coal Loading and Hauling

<p>| Produced (ton) | 1919 |
| Time taken (hr) | 11.5 |
| Started at (hr) | 821.4 |
| Finished at (hr) | 832.9 |
| Loader cycle time | 3 |
| Truck cycle time | 1.7 |
| Loading time | 1.7 |
| Haul time | 12.0 |
| Cycle time (min) | 13.7 |</p>
<table>
<thead>
<tr>
<th>LIFT NUMBER</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL NUMBER OF LIFTS IN BLOCK NO. 7</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLOCK VOLUME (BCY)</td>
<td>22815</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVERAGE SWELL</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLOCK VOLUME (LCY)</td>
<td>29659</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRIPPING RATIO (BCY/TON)</td>
<td>10.64</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DRILLING AND BLASTING**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLASTED (%)</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>DRILLED (FT)</td>
<td>1703</td>
<td>1703</td>
<td>3407</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
<td>761.5</td>
<td>866.7</td>
<td></td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
<td>787.0</td>
<td>892.5</td>
<td></td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
<td>25.6</td>
<td>29.8</td>
<td>51.4</td>
</tr>
<tr>
<td>COST ($)</td>
<td>694.4</td>
<td>700.2</td>
<td>1394.6</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
</tbody>
</table>

**DOZER PUSH**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOZED (%)</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>DOZED (LCY)</td>
<td>4448</td>
<td>4448</td>
<td>8897</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
<td>10.0</td>
<td>10.0</td>
<td>20.0</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
<td>767.0</td>
<td>892.5</td>
<td></td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
<td>797.1</td>
<td>902.5</td>
<td></td>
</tr>
<tr>
<td>COST ($)</td>
<td>556.8</td>
<td>556.8</td>
<td>1113.6</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
</tr>
</tbody>
</table>

**LOADER CARRYING OVERBURDEN**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARRIED (%)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CARRIED (LCY)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CYCLE TIME (MIN)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TRAVEL TIME (MIN)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COST ($)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**DOZER FEEDING OVERBURDEN TO LOADER (LOADER CARRY)**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTED AT (HR)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COST ($)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**OVERBURDEN LOADING AND HAULING**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAULED ($)</td>
<td>69</td>
<td>69</td>
<td>69</td>
</tr>
<tr>
<td>HAULED (LCY)</td>
<td>10380</td>
<td>10380</td>
<td>20761</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
<td>821.4</td>
<td>902.5</td>
<td></td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
<td>843.7</td>
<td>924.8</td>
<td></td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
<td>22.3</td>
<td>22.3</td>
<td>44.6</td>
</tr>
<tr>
<td>LOADER CYCLE TIME</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>TRUCK CYCLE TIME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOADING TIME</td>
<td>1.9</td>
<td>1.9</td>
<td></td>
</tr>
<tr>
<td>FORWARD HAUL TIME</td>
<td>1.2</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>RETURN HAUL TIME</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>WAITING TIME</td>
<td>2.1</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td>DUMPING TIME</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>CYCLE TIME (MIN)</td>
<td>5.8</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td>COST ($)</td>
<td>1764.9</td>
<td>1764.9</td>
<td>3529.7</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
<td>0.17</td>
<td>0.17</td>
<td>0.17</td>
</tr>
</tbody>
</table>

**DOZER FEEDING OVERBURDEN TO LOADER**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTED AT (HR)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COST ($)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**COAL LOADING AND HAULING**

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>1</th>
<th>2</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODUCED (TON)</td>
<td>2143</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
<td>12.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
<td>924.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
<td>937.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOADER CYCLE TIME</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRUCK CYCLE TIME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOADING TIME</td>
<td>1.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HAUL TIME 12.0
CYCLE TIME(MIN) 13.7

<table>
<thead>
<tr>
<th>TOTAL NUMBER OF Lifts IN BLOCK NO.</th>
<th>8</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCK VOLUME (BCY)</td>
<td>=</td>
<td>67477</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVERAGE SWELL</td>
<td>=</td>
<td>1.30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLOCK VOLUME (LCY)</td>
<td>=</td>
<td>87721</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRIPPING RATIO (BCY/TON)</td>
<td>=</td>
<td>15.66</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BLOCK NO.</th>
<th>8</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>TOTAL</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DRILLING AND BLASTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLASTED (%)</td>
</tr>
<tr>
<td>DRILLED (FT)</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOZER PULL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOZED (%)</td>
</tr>
<tr>
<td>DOZED (LCY)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOADER CARRYING OVERBURDEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARRIED (%)</td>
</tr>
<tr>
<td>CARRIED (LCY)</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>CYCLE TIME (MIN)</td>
</tr>
<tr>
<td>TRAVEL TIME (MIN)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOZER FEEDING OVERBURDENT TO LOADER (LOADER CARRY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OVERBURDEN LOADING AND HAULING</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAULED (%)</td>
</tr>
<tr>
<td>HAULED (LCY)</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>LOADER CYCLE TIME</td>
</tr>
<tr>
<td>TRUCK CYCLE TIME</td>
</tr>
<tr>
<td>LOADING TIME</td>
</tr>
<tr>
<td>FORWARD HAUL TIME</td>
</tr>
<tr>
<td>RETURN HAUL TIME</td>
</tr>
<tr>
<td>WAITING TIME</td>
</tr>
<tr>
<td>DUMPING TIME</td>
</tr>
<tr>
<td>CYCLE TIME (MIN)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOZER FEEDING OVERBURDENT TO LOADER</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
</tbody>
</table>

191
C O A L L O A D I N G A N D H A U L I N G

PRODUCED (TON) 4309
TIME TAKEN (HR) 20.5
STARTED AT (HR) 881.4
FINISHED AT (HR) 901.9
LOADER CYCLE TIME .3
TRUCK CYCLE TIME
LOADING TIME 1.7
HAUL TIME 12.0

C O S T ($)
C O S T ($/LCY)
.00 .00 .00 .00

C O S T ($)
C O S T ($/LCY)
.00 .00 .00 .00

L I T T L E N U M B E R

B L O C K N U M B E R

1 2 T O T A L

D R I L L I N G A N D B L A S T I N G

B L A S T E D (%) 89 89 90
D R I L L E D (FT) 1090 1090 2181
S T A R T E D AT (HR) 839.9 899.5
F I N I S H E D AT (HR) 857.7 917.1
T I M E T A K E N (HR) 17.8 17.7 35.5
C O S T ($) 483.0 479.6 962.6
C O S T ($/LCY) .06 .06 .06

D O Z E R P U S H

D O Z E D (%) 50 50 50
D O Z E D (LCY) 4225 4225 8450
S T A R T E D AT (HR) 857.7 917.1
F I N I S H E D AT (HR) 867.2 926.7
C O S T ($) 528.8 528.8 1057.7
C O S T ($/LCY) .13 .13 .13

L O A D E R C A R R Y I N G O V E R B U R D E N

C A R R I E D (%) 0 0 0
C A R R I E D (LCY) 0 0 0
S T A R T E D AT (HR) .0 .0 .0
F I N I S H E D AT (HR) .0 .0 .0
T I M E T A K E N (HR) .0 .0 .0
C Y C L E T I M E (MIN) .0 .0 .0
T R A V E L T I M E (MIN) .0 .0 .0
C O S T ($) .0 .0 .0
C O S T ($/LCY) .00 .00 .00


S T A R T E D AT (HR) .0 .0 .0
F I N I S H E D AT (HR) .0 .0 .0
T I M E T A K E N (HR) .0 .0 .0
C O S T ($) .0 .0 .0
C O S T ($/LCY) .00 .00 .00

O V E R B U R D E N L O A D I N G A N D H A U L I N G

H A U L E D (%) 50 50 50
H A U L E D (LCY) 4225 4225 8450
S T A R T E D AT (HR) 881.4 926.7
F I N I S H E D AT (HR) 890.5 935.7
T I M E T A K E N (HR) 9.1 9.1 18.2
L O A D E R C Y C L E T I M E .3 .3 .3
T R U C K C Y C L E T I M E
L O A D I N G T I M E 1.9 1.9
F O R W A R D H A U L T I M E 1.2 1.2
R E T U R N H A U L T I M E .0 .0
<table>
<thead>
<tr>
<th>WAITING TIME</th>
<th>3.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMPING TIME</td>
<td>3.1</td>
</tr>
<tr>
<td>CYCLE TIME (MIN)</td>
<td>5.8</td>
</tr>
<tr>
<td>COST ($)</td>
<td>718.4</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
<td>.17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOZER FEEDING OVERBURDEN TO LOADER</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COAL LOADING AND HAULING</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODUCED (TON)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>LOADER CYCLE TIME</td>
</tr>
<tr>
<td>TRUCK CYCLE TIME</td>
</tr>
<tr>
<td>HAUL TIME</td>
</tr>
<tr>
<td>CYCLE TIME (MIN)</td>
</tr>
</tbody>
</table>

| TOTAL NUMBER OF LIFTS IN BLOCK NO. 10 | 2 |
| BLOCK VOLUME (BCY) | 22911.0 |
| AVERAGE SWELL | 1.30 |
| BLOCK VOLUME (LCY) | 29784.0 |
| STRIPPING RATIO (BCY/TON) | 22.10 |

<table>
<thead>
<tr>
<th>DRILLING AND BLASTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLASTED (%)</td>
</tr>
<tr>
<td>UNDRILLED (FT)</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOZER PUSH</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOZED (%)</td>
</tr>
<tr>
<td>DOZED (LCY)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOADER CARRYING OVERBURDEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARRIED (%)</td>
</tr>
<tr>
<td>CARRIED (LCY)</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>CYCLE TIME (MIN)</td>
</tr>
<tr>
<td>TRAVEL TIME (MIN)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOZER FEEDING OVERBURDEN TO LOADER (LOADER CARRY)</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTED AT (HR)</td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
</tr>
<tr>
<td>COST ($)</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
</tr>
</tbody>
</table>
## Overburden Loading and Hauling

<table>
<thead>
<tr>
<th>HAULED (%)</th>
<th>50</th>
<th>50</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAULED (LCY)</td>
<td>722.2</td>
<td>7669</td>
<td>14892</td>
</tr>
<tr>
<td>STARTED AT (HR)</td>
<td>991.9</td>
<td>1085.8</td>
<td></td>
</tr>
<tr>
<td>FINISHED AT (HR)</td>
<td>1007.5</td>
<td>1102.3</td>
<td></td>
</tr>
<tr>
<td>TIME TAKEN (HR)</td>
<td>15.5</td>
<td>16.5</td>
<td>32.0</td>
</tr>
<tr>
<td>LOADER CYCLE TIME</td>
<td>.3</td>
<td>.3</td>
<td></td>
</tr>
<tr>
<td>TRUCK CYCLE TIME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOADING TIME</td>
<td>1.9</td>
<td>1.9</td>
<td></td>
</tr>
<tr>
<td>FORWARD HAUL TIME</td>
<td>1.1</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>RETURN HAUL TIME</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>WAITING TIME</td>
<td>2.2</td>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td>DUMPING TIME</td>
<td>.6</td>
<td>.6</td>
<td></td>
</tr>
<tr>
<td>DUMPING TIME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCLE TIME (MIN)</td>
<td>5.8</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td>COST ($)</td>
<td>1227.9</td>
<td>1304.0</td>
<td>2531.8</td>
</tr>
<tr>
<td>COST ($/LCY)</td>
<td>.17</td>
<td>.17</td>
<td>.17</td>
</tr>
</tbody>
</table>

## Dozer Feeding Overburden to Loader

| STARTED AT (HR) | .0 | .0 | .0 |
| FINISHED AT (HR) | .0 | .0 | .0 |
| TIME TAKEN (HR) | .0 | .0 | .0 |
| COST ($) | .0 | .0 | .0 |
| COST ($/LCY) | .00 | .00 | .00 |

## Coal Loading and Hauling

| PRODUCED (TON) | 1036 |
| STARTED AT (HR) | 1102.3 |
| FINISHED AT (HR) | 1110.5 |
| LOADER CYCLE TIME | .3 |
| TRUCK CYCLE TIME | | |
| LOADING TIME | 1.7 |
| HAUL TIME | 12.0 |
| CYCLE TIME (MIN) | 13.7 |

## Grading of Backfill Cells

### Backfill Cell 1

| STARTED AT (HR) | 416.3 |
| FINISHED AT (HR) | 417.3 |
| TIME TAKEN | 1.0 |

### Backfill Cell 2

| STARTED AT (HR) | 752.5 |
| FINISHED AT (HR) | 754.5 |
| TIME TAKEN | 2.0 |

### Backfill Cell 3

| STARTED AT (HR) | 754.5 |
| FINISHED AT (HR) | 757.5 |
| TIME TAKEN | 3.0 |

### Backfill Cell 4

| STARTED AT (HR) | 857.7 |
| FINISHED AT (HR) | 860.7 |
| TIME TAKEN | 3.0 |
# BACKFILL CELLS SUMMARY

<table>
<thead>
<tr>
<th>NO.</th>
<th>CAPACITY (LCY)</th>
<th>VOLUME LEFT UNFILLED (LCY)</th>
<th>VOLUME RECEIVED FROM BLOCK LIFT</th>
<th>AMOUNT (LCY)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>BY LOAD AND HAUL</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>15300.</td>
<td>0.</td>
<td>3</td>
<td>13957.</td>
</tr>
<tr>
<td>2</td>
<td>20900.</td>
<td>0.</td>
<td>8</td>
<td>10373.</td>
</tr>
<tr>
<td>3</td>
<td>14300.</td>
<td>0.</td>
<td>8</td>
<td>14191.</td>
</tr>
<tr>
<td>4</td>
<td>15850.</td>
<td>0.</td>
<td>8</td>
<td>15850.</td>
</tr>
</tbody>
</table>

|     |                |                             | BY LOADER CARRY                |               |
|     |                |                             | BY DOZER PUSH                  |               |
| 1   |                |                             | 5                              | 1343.         |
| 2   |                |                             | 8                              | 10527.        |
| 3   |                |                             | 6                              | 109.          |
| 4   |                |                             | EXTRA SPOIL DISPOSAL AREA      |               |

|     |                |                             | BY LOAD AND HAUL               |               |
| 1   | 653500.        | 322361.                     | 1                              | 12168.        |
| 2   | 653500.        | 322361.                     | 2                              | 9434.         |
| 3   |                |                             | 3                              | 10135.        |
| 4   |                |                             | 4                              | 11779.        |
| 5   |                |                             | 5                              | 5521.         |
| 6   |                |                             | 6                              | 14193.        |
| 7   |                |                             | 7                              | 26564.        |
| 8   |                |                             | 8                              | 20191.        |
| 9   |                |                             | 9                              | 13022.        |
| 10  |                |                             | 10                             | 17823.        |
|     |                |                             | EXTRA SPOIL DISPOSAL AREA      |               |
| 1   |                |                             | BY LOAD AND HAUL               |               |

|     |                |                             | BY LOADER CARRY                |               |

|     |                |                             | BY DOZER PUSH                  |               |
| 1   |                |                             | 1                              | 8112.         |
| 2   |                |                             | 2                              | 11804.        |
| 3   |                |                             | 3                              | 6289.         |
| 4   |                |                             | 4                              | 6756.         |
| 5   |                |                             | 5                              | 7853.         |
| 6   |                |                             | 6                              | 9305.         |
| 7   |                |                             | 7                              | 1812.         |
| 8   |                |                             | 8                              | 9462.         |
### SCHEDULING OF EQUIPMENT

#### DRILL

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>NUMBER OF MACHINE</th>
<th>START</th>
<th>HOURS</th>
<th>WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>B 1 L1</td>
<td>1</td>
<td>96.73</td>
<td>50.82</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 2 L1</td>
<td>1</td>
<td>153.35</td>
<td>27.94</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 3 L1</td>
<td>1</td>
<td>247.28</td>
<td>30.43</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 2 L2</td>
<td>1</td>
<td>289.71</td>
<td>33.97</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 3 L2</td>
<td>1</td>
<td>329.68</td>
<td>40.51</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 4 L1</td>
<td>1</td>
<td>382.19</td>
<td>15.87</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 5 L1</td>
<td>1</td>
<td>404.06</td>
<td>42.57</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 6 L1</td>
<td>1</td>
<td>456.63</td>
<td>60.35</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 6 L2</td>
<td>1</td>
<td>524.98</td>
<td>31.27</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 7 L1</td>
<td>1</td>
<td>586.25</td>
<td>37.57</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 7 L2</td>
<td>1</td>
<td>631.82</td>
<td>59.83</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 8 L1</td>
<td>1</td>
<td>699.65</td>
<td>44.81</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 9 L1</td>
<td>1</td>
<td>752.46</td>
<td>25.58</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 9 L2</td>
<td>1</td>
<td>787.04</td>
<td>32.26</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 9 L3</td>
<td>1</td>
<td>832.88</td>
<td>17.79</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 10 L1</td>
<td>1</td>
<td>892.47</td>
<td>17.67</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 10 L2</td>
<td>1</td>
<td>917.14</td>
<td>28.05</td>
<td>DRILLING</td>
</tr>
<tr>
<td>B 10 L2</td>
<td>1</td>
<td>1007.46</td>
<td>29.48</td>
<td>DRILLING</td>
</tr>
</tbody>
</table>

#### DOZER

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>NUMBER OF MACHINE</th>
<th>START</th>
<th>HOURS</th>
<th>WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>B 1 L1</td>
<td>2</td>
<td>41.92</td>
<td>28.65</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 2 L1</td>
<td>2</td>
<td>133.55</td>
<td>41.69</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 3 L1</td>
<td>2</td>
<td>187.50</td>
<td>18.19</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 3 L2</td>
<td>2</td>
<td>289.71</td>
<td>15.22</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 4 L1</td>
<td>2</td>
<td>329.68</td>
<td>22.71</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 5 L1</td>
<td>2</td>
<td>382.19</td>
<td>20.96</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 5 L2</td>
<td>2</td>
<td>404.06</td>
<td>12.19</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 6 L1</td>
<td>2</td>
<td>416.25</td>
<td>1.00</td>
<td>GRADING BACKFILL CELL</td>
</tr>
<tr>
<td>B 7 L1</td>
<td>2</td>
<td>456.63</td>
<td>15.26</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 8 L1</td>
<td>2</td>
<td>524.98</td>
<td>43.91</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 8 L2</td>
<td>2</td>
<td>586.25</td>
<td>18.61</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 9 L1</td>
<td>2</td>
<td>631.82</td>
<td>9.79</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 9 L2</td>
<td>2</td>
<td>699.65</td>
<td>30.44</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 10 L1</td>
<td>2</td>
<td>752.46</td>
<td>30.60</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 10 L2</td>
<td>2</td>
<td>752.46</td>
<td>2.00</td>
<td>GRADING BACKFILL CELL</td>
</tr>
<tr>
<td>B 11 L1</td>
<td>2</td>
<td>754.46</td>
<td>3.00</td>
<td>GRADING BACKFILL CELL</td>
</tr>
<tr>
<td>B 12 L1</td>
<td>2</td>
<td>787.04</td>
<td>10.02</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 13 L1</td>
<td>2</td>
<td>832.88</td>
<td>12.60</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 14 L1</td>
<td>2</td>
<td>892.47</td>
<td>17.67</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 15 L1</td>
<td>2</td>
<td>917.14</td>
<td>28.05</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 16 L1</td>
<td>2</td>
<td>957.19</td>
<td>34.74</td>
<td>PUSHING OVERBURDEN</td>
</tr>
<tr>
<td>B 17 L1</td>
<td>2</td>
<td>1048.94</td>
<td>36.90</td>
<td>PUSHING OVERBURDEN</td>
</tr>
</tbody>
</table>

#### TRUCK
<table>
<thead>
<tr>
<th>LOCATION</th>
<th>NUMBER OF MACHINE</th>
<th>START HOURS</th>
<th>WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>B 1 L1</td>
<td>2</td>
<td>70.57</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 1 L2</td>
<td>2</td>
<td>195.24</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 2 L1</td>
<td>2</td>
<td>233.30</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 2 L2</td>
<td>2</td>
<td>304.92</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 3 L1</td>
<td>2</td>
<td>352.39</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 3 L2</td>
<td>2</td>
<td>403.14</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 4 L1</td>
<td>2</td>
<td>433.15</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 4 L2</td>
<td>2</td>
<td>471.89</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 5 L1</td>
<td>2</td>
<td>568.89</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 5 L2</td>
<td>2</td>
<td>621.70</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 6 L1</td>
<td>2</td>
<td>665.11</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 6 L2</td>
<td>2</td>
<td>730.09</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 7 L1</td>
<td>2</td>
<td>783.06</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 7 L2</td>
<td>2</td>
<td>821.37</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 8 L1</td>
<td>2</td>
<td>843.69</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 8 L2</td>
<td>2</td>
<td>881.40</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 9 L1</td>
<td>2</td>
<td>902.49</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 9 L2</td>
<td>2</td>
<td>926.66</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 10 L1</td>
<td>2</td>
<td>991.93</td>
<td>HAULING</td>
</tr>
<tr>
<td>B 10 L2</td>
<td>2</td>
<td>1085.84</td>
<td>HAULING</td>
</tr>
</tbody>
</table>

### LOADER

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>NUMBER OF MACHINE</th>
<th>START HOURS</th>
<th>WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>B 1 L1</td>
<td>1</td>
<td>70.57</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 2 L1</td>
<td>1</td>
<td>195.24</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 2 L2</td>
<td>1</td>
<td>233.30</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 3 L1</td>
<td>1</td>
<td>304.92</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 3 L2</td>
<td>1</td>
<td>352.39</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 4 L1</td>
<td>1</td>
<td>403.14</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 4 L2</td>
<td>1</td>
<td>471.89</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 5 L1</td>
<td>1</td>
<td>568.89</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 5 L2</td>
<td>1</td>
<td>621.70</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 6 L1</td>
<td>1</td>
<td>665.11</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 6 L2</td>
<td>1</td>
<td>730.09</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 7 L1</td>
<td>1</td>
<td>783.06</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 7 L2</td>
<td>1</td>
<td>821.37</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 8 L1</td>
<td>1</td>
<td>843.69</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 8 L2</td>
<td>1</td>
<td>881.40</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 9 L1</td>
<td>1</td>
<td>902.49</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 9 L2</td>
<td>1</td>
<td>926.66</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 10 L1</td>
<td>1</td>
<td>991.93</td>
<td>OVERBURDEN</td>
</tr>
<tr>
<td>B 10 L2</td>
<td>1</td>
<td>1085.84</td>
<td>OVERBURDEN</td>
</tr>
</tbody>
</table>

### COAL LOADER

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>NUMBER OF MACHINE</th>
<th>START HOURS</th>
<th>WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>B 1 L1</td>
<td>2</td>
<td>233.30</td>
<td>COAL</td>
</tr>
<tr>
<td>B 2 L1</td>
<td>2</td>
<td>377.71</td>
<td>COAL</td>
</tr>
<tr>
<td>B 2 L2</td>
<td>2</td>
<td>433.15</td>
<td>COAL</td>
</tr>
<tr>
<td>B 3 L1</td>
<td>2</td>
<td>445.02</td>
<td>COAL</td>
</tr>
<tr>
<td>B 3 L2</td>
<td>2</td>
<td>665.11</td>
<td>COAL</td>
</tr>
<tr>
<td>B 4 L1</td>
<td>2</td>
<td>821.37</td>
<td>COAL</td>
</tr>
<tr>
<td>B 4 L2</td>
<td>2</td>
<td>881.40</td>
<td>COAL</td>
</tr>
<tr>
<td>B 5 L1</td>
<td>2</td>
<td>935.94</td>
<td>COAL</td>
</tr>
<tr>
<td>B 5 L2</td>
<td>2</td>
<td>1102.33</td>
<td>COAL</td>
</tr>
</tbody>
</table>

### SUMMARY OF OPERATIONS AND COSTS

<table>
<thead>
<tr>
<th>OPERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>B 1 L1</td>
</tr>
<tr>
<td>B 2 L1</td>
</tr>
<tr>
<td>B 2 L2</td>
</tr>
<tr>
<td>B 3 L1</td>
</tr>
<tr>
<td>B 3 L2</td>
</tr>
<tr>
<td>B 4 L1</td>
</tr>
<tr>
<td>B 4 L2</td>
</tr>
<tr>
<td>B 5 L1</td>
</tr>
<tr>
<td>B 5 L2</td>
</tr>
<tr>
<td>B 6 L1</td>
</tr>
<tr>
<td>B 6 L2</td>
</tr>
<tr>
<td>B 7 L1</td>
</tr>
<tr>
<td>B 7 L2</td>
</tr>
<tr>
<td>B 8 L1</td>
</tr>
<tr>
<td>B 8 L2</td>
</tr>
<tr>
<td>B 9 L1</td>
</tr>
<tr>
<td>B 9 L2</td>
</tr>
<tr>
<td>B 10 L1</td>
</tr>
<tr>
<td>B 10 L2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>NUMBER OF MACHINE</th>
<th>START HOURS</th>
<th>WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>B 1 L1</td>
<td>2</td>
<td>233.30</td>
<td>COAL</td>
</tr>
<tr>
<td>B 2 L1</td>
<td>2</td>
<td>377.71</td>
<td>COAL</td>
</tr>
<tr>
<td>B 2 L2</td>
<td>2</td>
<td>433.15</td>
<td>COAL</td>
</tr>
<tr>
<td>B 3 L1</td>
<td>2</td>
<td>445.02</td>
<td>COAL</td>
</tr>
<tr>
<td>B 3 L2</td>
<td>2</td>
<td>665.11</td>
<td>COAL</td>
</tr>
<tr>
<td>B 4 L1</td>
<td>2</td>
<td>821.37</td>
<td>COAL</td>
</tr>
<tr>
<td>B 4 L2</td>
<td>2</td>
<td>881.40</td>
<td>COAL</td>
</tr>
<tr>
<td>B 5 L1</td>
<td>2</td>
<td>935.94</td>
<td>COAL</td>
</tr>
<tr>
<td>B 5 L2</td>
<td>2</td>
<td>1102.33</td>
<td>COAL</td>
</tr>
<tr>
<td>Category</td>
<td>Details</td>
<td>Value</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>----------------------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>TOTAL COAL MINED (TONS)</td>
<td></td>
<td>22612</td>
<td></td>
</tr>
<tr>
<td>TOTAL OVERBURDEN REMOVED (LCY)</td>
<td></td>
<td>393815</td>
<td></td>
</tr>
<tr>
<td>TOTAL OVERBURDEN REMOVED (BCY)</td>
<td></td>
<td>302935</td>
<td></td>
</tr>
<tr>
<td>AVERAGE SWELL</td>
<td></td>
<td>1.30</td>
<td></td>
</tr>
<tr>
<td>STRIPPING RATIO (BCY/TON)</td>
<td></td>
<td>13.40</td>
<td></td>
</tr>
<tr>
<td>DRILLING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL NO. OF DRILLING MACHINE</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TOTAL LENGTH DRILLED (FT)</td>
<td></td>
<td>46393</td>
<td></td>
</tr>
<tr>
<td>TIME REQUIRED TO DRILL (HR)</td>
<td></td>
<td>708.5</td>
<td></td>
</tr>
<tr>
<td>DOZING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL NO. OF DOZERS</td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>TOTAL AMOUNT OF OVERBURDEN HANDLED (LCY)</td>
<td></td>
<td>151582</td>
<td></td>
</tr>
<tr>
<td>TOTAL TIME TAKEN TO REMOVE OVERBURDEN (HR)</td>
<td></td>
<td>288.1</td>
<td></td>
</tr>
<tr>
<td>OVERBURDEN HAULING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL NO. OF TRUCKS</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>TOTAL AMOUNT OF OVERBURDEN HANDLED (LCY)</td>
<td></td>
<td>256700</td>
<td></td>
</tr>
<tr>
<td>TOTAL DISTANCE HAULED (FT)</td>
<td></td>
<td>8209403</td>
<td></td>
</tr>
<tr>
<td>TOTAL TIME TAKEN TO REMOVE OVERBURDEN (HR)</td>
<td></td>
<td>351.9</td>
<td></td>
</tr>
<tr>
<td>TOTAL AMOUNT OF OVERBURDEN REHANDLED (LCY)</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>MATCH FACTOR</td>
<td></td>
<td>1.7</td>
<td></td>
</tr>
<tr>
<td>COAL LOADING AND HAULING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL NO. OF LOADER</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>TOTAL AMOUNT OF COAL HAULED (TONS)</td>
<td></td>
<td>22612</td>
<td></td>
</tr>
<tr>
<td>TOTAL TIME TAKEN TO REMOVE COAL (HR)</td>
<td></td>
<td>71.6</td>
<td></td>
</tr>
</tbody>
</table>

EQUIPMENT COST (OPERATORS WAGES ARE NOT INCLUDED)

<table>
<thead>
<tr>
<th>Category</th>
<th>Details</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>COST OF DRILL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPERATING COST PER HOUR</td>
<td></td>
<td>27.15</td>
</tr>
<tr>
<td>OWNING COST PER HOUR</td>
<td></td>
<td>35.40</td>
</tr>
<tr>
<td>TOTAL COST PER HOUR</td>
<td></td>
<td>62.54</td>
</tr>
<tr>
<td>TOTAL HOURS WORKED</td>
<td></td>
<td>704.34</td>
</tr>
<tr>
<td>TOTAL HOURS IDLE</td>
<td></td>
<td>399.66</td>
</tr>
<tr>
<td>TOTAL OPERATING COST</td>
<td></td>
<td>44052.90</td>
</tr>
<tr>
<td>TOTAL IDLE COST</td>
<td></td>
<td>14147.99</td>
</tr>
<tr>
<td>COST OF DRILLING ($/LCY)</td>
<td></td>
<td>.18</td>
</tr>
<tr>
<td>COST OF DOZING</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPERATING COST PER HOUR</td>
<td></td>
<td>27.78</td>
</tr>
<tr>
<td>OWNING COST PER HOUR</td>
<td></td>
<td>20.73</td>
</tr>
<tr>
<td>TOTAL COST PER HOUR</td>
<td></td>
<td>48.51</td>
</tr>
<tr>
<td>TOTAL HOURS WORKED</td>
<td></td>
<td>289.69</td>
</tr>
<tr>
<td>TOTAL HOURS IDLE</td>
<td></td>
<td>814.31</td>
</tr>
<tr>
<td>TOTAL OPERATING COST</td>
<td></td>
<td>42160.99</td>
</tr>
<tr>
<td>TOTAL IDLE COST</td>
<td></td>
<td>50636.51</td>
</tr>
</tbody>
</table>
### COST OF DOZING ($/LCY)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COST OF TRUCK</strong></td>
<td></td>
</tr>
<tr>
<td>Operating Cost Per Hour</td>
<td>27.25</td>
</tr>
<tr>
<td>Owning Cost Per Hour</td>
<td>10.66</td>
</tr>
<tr>
<td>Total Cost Per Hour</td>
<td>37.91</td>
</tr>
<tr>
<td>Total Hours Worked</td>
<td>568.63</td>
</tr>
<tr>
<td>Total Hours Idle</td>
<td>555.37</td>
</tr>
<tr>
<td>Total Operating Cost</td>
<td>41601.00</td>
</tr>
<tr>
<td>Total Idle Cost</td>
<td>11840.43</td>
</tr>
<tr>
<td>Cost of Overburden Truck ($/LCY)</td>
<td>0.21</td>
</tr>
</tbody>
</table>

### COST OF OVERBURDEN LOADER

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Cost Per Hour</td>
<td>24.57</td>
</tr>
<tr>
<td>Owning Cost Per Hour</td>
<td>16.30</td>
</tr>
<tr>
<td>Total Cost Per Hour</td>
<td>41.07</td>
</tr>
<tr>
<td>Total Hours Worked</td>
<td>182.88</td>
</tr>
<tr>
<td>Total Hours Idle</td>
<td>921.12</td>
</tr>
<tr>
<td>Total Operating Cost</td>
<td>22533.85</td>
</tr>
<tr>
<td>Total Idle Cost</td>
<td>43593.36</td>
</tr>
<tr>
<td>Cost of Overburden Loader ($/LCY)</td>
<td>0.27</td>
</tr>
</tbody>
</table>

### COST OF COAL LOADER

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Cost Per Hour</td>
<td>23.22</td>
</tr>
<tr>
<td>Owning Cost Per Hour</td>
<td>16.50</td>
</tr>
<tr>
<td>Total Cost Per Hour</td>
<td>39.72</td>
</tr>
<tr>
<td>Total Hours Worked</td>
<td>142.39</td>
</tr>
<tr>
<td>Total Hours Idle</td>
<td>961.61</td>
</tr>
<tr>
<td>Total Operating Cost</td>
<td>11312.49</td>
</tr>
<tr>
<td>Total Idle Cost</td>
<td>31732.97</td>
</tr>
<tr>
<td>Cost of Coal Loader ($/TON)</td>
<td>1.90</td>
</tr>
</tbody>
</table>

### COSTS

**Completion Time (Shifts)**

138

**Payroll Costs**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Hourly Payroll (Non Operators = 4)</td>
<td>37315</td>
</tr>
<tr>
<td>Total Hourly Payroll (Operators = 5)</td>
<td>66129</td>
</tr>
<tr>
<td>Salaried Personnel Payroll (1 Persons)</td>
<td>6747</td>
</tr>
</tbody>
</table>

**Total Payroll Costs**

110192

**Labor Related Overhead Cost**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pension Trust</td>
<td>10432</td>
</tr>
<tr>
<td>Benefit Trust</td>
<td>23290</td>
</tr>
</tbody>
</table>

**Owning Cost of Equipment ($)**
OWNING COST FOR DRILLING MACHINE
  DEPRECIATION COST  17102
  INSURANCE, TAXES ETC.  22210

OWNING COST FOR DOZER.
  DEPRECIATION COST  28910
  INSURANCE, TAXES ETC.  40145

OWNING COST FOR TRUCK (OVERBURDEN)
  DEPRECIATION COST  8129
  INSURANCE, TAXES ETC.  15547

OWNING COST FOR LOADER (OVERBURDEN)
  DEPRECIATION COST  32316
  INSURANCE, TAXES ETC.  22654

OWNING COST FOR LOADER (COAL)
  DEPRECIATION COST  21544
  INSURANCE, TAXES ETC.  15103

TOTAL EQUIPMENT OWNING COST  223663

SUPPLY COST ($)

TOTAL TIRE REPLACEMENT COST  18542
TOTAL FUEL COST  46010
TOTAL REPAIR COST  16657
MISCELENEOUS SUPPLY (LUBRICANT ETC.) COST  9279
TOTAL BLASTING COST  26151
TOTAL SUPPLY COST  116641

COAL HAULING COST  29396
GRADING AND SEEDING COST  1250

TOTAL COST ($)  514867

COST OF COAL ($/TON)  22.77

END
The vita has been removed from the scanned document
AN INTEGRATED COMPUTER SIMULATOR
FOR SURFACE MINE PLANNING AND
DESIGN
by
Amal Chakraborty
Committee Chairman: Dr. Ertugrul Topuz
Department of Mining and Minerals Engineering

(ABSTRACT)

In the increasingly competitive coal market, it is becoming more important for coal operators to develop mathematical models for surface mining which can estimate mining costs before the actual mining begins. The problem becomes even more acute with the new reclamation laws, as they affect surface coal mining methods, productivity, and costs.

This study presents a computer simulator for a mountaintop removal type of surface mining operation. It will permit users to compare the costs associated with different overburden handling and reclamation plans. It may be used to minimize productivity losses, and, perhaps, to increase productivity and consequently to reduce operating costs through design and implementation of modified mountain top removal methods.