
C. E. Zipper and W. L. Daniels
The Virginia Agricultural and Mechanical College came into being in 1872 upon acceptance by the Commonwealth of the provisions of the Morrill Act of 1862 "to promote the liberal and practical education of the industrial classes in the several pursuits and professions of life." Research and investigations were first authorized at Virginia's land-grant college when the Virginia Agricultural Experiment Station was established by the Virginia General Assembly in 1886.

The Virginia Agricultural Experiment Station received its first allotment upon passage of the Hatch Act by the United States Congress in 1887. Other related Acts followed, and all were consolidated in 1955 under the Amended Hatch Act which states "It shall be the object and duty of the State agricultural experiment stations ... to conduct original and other researches, investigations and experiments bearing directly on and contributing to the establishment and maintenance of a permanent and effective agricultural industry of the United States, including the researches basic to the problems of agriculture and its broadest aspects and such investigations as have for their purpose the development and improvement of the rural home and rural life and the maximum contributions by agriculture to the welfare of the consumer . . . ."

In 1962, Congress passed the McIntire-Stennis Cooperative Forestry Research Act to encourage and assist the states in carrying on a program of forestry research, including reforestation, land management, watershed management, rangeland management, wildlife habitat improvement, outdoor recreation, harvesting and marketing of forest products, and "such other studies as may be necessary to obtain the fullest and most effective use of forest resources."

In 1966, the Virginia General Assembly "established within the Virginia Polytechnic Institute a division to be known as the Research Division . . . which shall encompass the now existing Virginia Agricultural Experiment Station . . . ."

To simplify terminology, trade names of products or equipment may have been used in this publication, but no endorsement of products or firms mentioned is intended, nor is criticism implied of those not mentioned. Material appearing here may be reprinted provided no endorsement of a commercial product is stated or implied. Please credit the researchers involved and the Virginia Agricultural Experiment Station.

Virginia Tech does not discriminate against employees, students, or applicants on the basis of race, sex, handicap, age, veteran status, national origin, religion, or political affiliation. Anyone having questions concerning discrimination should contact the Equal Employment/Affirmative Action Office.
COSTSUM:  
A System for Analysis of Operational Cost Data  
From Coal Surface Mines  
- A User's Guide -

C. E. Zipper and W. L. Daniels  
Department of Agronomy  
Virginia Polytechnic Institute and State University  
Blacksburg, Virginia 24061

Virginia Agricultural Experiment Station  
Bulletin 86-1  
April, 1986
ABSTRACT

COSTSUM is a series of computer programs designed to provide detailed analysis of the costs of coal surface mining and reclamation. It works with data from haulback operations, where haulers, loaders, and dozers are the primary machines used to handle spoil. It is intended for use by mining engineers, regulatory agency personnel, and researchers with an interest in the effects of spoil handling and reclamation practices upon overall mining cost. The primary data requirements are that the times, types, and locations of all machine operations at the site of interest be recorded on a daily basis. Additional required data include coal production, coal prices, overburden volumes, and machinery operating costs. Data are entered into computer files according to specified formats. The programs calculate cost totals for each mining area, which are compared to returns in order to assess the profitability of mining that area. Cost totals are broken down into cost categories for each mining area and recalculated on a per cubic yard of spoil basis for spoil handling operations and a per ton basis for coal handling operations. Thus, the user is able to identify those areas where spoil was handled most efficiently and those where spoil handling was most costly, and the reasons why.

This bulletin serves as a guide for the use of COSTSUM. Copies of the programs may be obtained free of charge by contacting the authors.
Acknowledgements

COSTSUM was developed as a portion of a research project funded by the Powell River Project, U.S. Office of Surface Mining, and the Virginia Mining and Mineral Resources Research Institute through the U. S. Bureau of Mines.
## CONTENTS

Abstract ........................................ iii
Acknowledgments ................................. iv
Introduction ..................................... 1
Assumptions ..................................... 4
Data Requirements ............................... 7
The Programs .................................... 23
  Level 1 Programs ............................... 27
  Level 2 Programs ............................... 32
    MOVE ......................................... 34
    COST ......................................... 37
Use and Modification ............................ 48
Making Use of the Output ....................... 54
Concluding Remarks ............................. 57
References ..................................... 58
Appendix A: HAULER ......................... 59
Appendix B: LOADER ......................... 66
Appendix C: DOZER ............................. 74
Appendix D: DRILL ............................. 81
Appendix E: OTHER ............................. 84
Appendix F: MOVE .............................. 89
Appendix G: COST .............................. 103
Appendix H: EXEC file and subroutines ....... 126
LIST OF FIGURES

Figure 1. Example of form used for recording daily site data for input to COSTSUM 8

Figure 2. Input requirements, and input and output file names, for the seven programs of COSTSUM 24

Figure 3. Generalized flowchart for the five Level 1 programs 28

Figure 4. Generalized flowchart for the program MOVE 35

Figure 5. Generalized flowchart for the program COST 38
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1.</td>
<td>Column format for file HAULER INDATA, the input to program HAULER FORTRAN</td>
<td>12</td>
</tr>
<tr>
<td>Table 2.</td>
<td>Column format for file LOADER INDATA, the input to program LOADER FORTRAN</td>
<td>13</td>
</tr>
<tr>
<td>Table 3.</td>
<td>Column format for file DOZER INDATA, the input to program DOZER FORTRAN</td>
<td>14</td>
</tr>
<tr>
<td>Table 4.</td>
<td>Column format for file DRILL INDATA, the input to program DRILL FORTRAN</td>
<td>15</td>
</tr>
<tr>
<td>Table 5.</td>
<td>Column format for file OTHER INDATA, the input to program OTHER FORTRAN</td>
<td>16</td>
</tr>
<tr>
<td>Table 6.</td>
<td>Formats for file DATA MOVINPUT, the input file to program MOVE FORTRAN</td>
<td>18</td>
</tr>
<tr>
<td>Table 7.</td>
<td>Formats for file DATA COSINPUT, the input file to program COST FORTRAN</td>
<td>19</td>
</tr>
<tr>
<td>Table 8.</td>
<td>Cost data required by COST FORTRAN, with cost item numbers</td>
<td>20</td>
</tr>
<tr>
<td>Table 9.</td>
<td>Cost categories used by COST FORTRAN, with category numbers</td>
<td>40</td>
</tr>
<tr>
<td>Table 10.</td>
<td>Major internal variables and quantities represented, program COST</td>
<td>41</td>
</tr>
<tr>
<td>Table 11.</td>
<td>Primary operations performed by programming code associated with data READ statements, by GO symbol, program COST</td>
<td>42</td>
</tr>
</tbody>
</table>
INTRODUCTION

COSTSUM (COST of SURface Mining) is a system for collection and analysis of coal surface mine data. The system was developed by the Agronomy Department at Virginia Polytechnic Institute and State University between 1983 and 1985 and is intended for use at haulback operations, where dozers, loaders, and haulers are the primary equipment used to remove the overburden covering the coal. The influence of spoil handling and reclamation practices upon overall mining costs were a primary concern during development of COSTSUM (1,2).

The purpose of this bulletin is to serve as a user's guide to COSTSUM. The system can assist anyone who wishes to conduct detailed cost analyses of haulback surface mining operations. Mining engineers with a desire to improve the cost effectiveness of their mining systems and their ability to estimate future costs will find this data-handling system useful. Regulatory agency personnel can use COSTSUM to assist in preparation of reclamation cost estimates when setting performance bond amounts. Also, COSTSUM is useful for research purposes due to continuing concern over the issue of regulatory costs both within and outside the mining industry. This bulletin is written
with the assumption that the user possesses a basic knowledge of computer operation and FORTRAN programming, including the ability to construct and edit data files, and to edit, compile, load, and run FORTRAN programs.

The COSTSUM user should record machinery operation data at the mine site on a daily basis. Additional data on mining progress should be recorded periodically, and the hourly costs of operating mining machinery must be estimated. Coal production and price figures and an estimate of overhead cost are also required. The data are entered into computer files according to specified formats and analyzed by a series of FORTRAN 77 programs; these programs give a detailed breakdown of the cost of mining and reclamation. Cost totals are prepared and broken down into 15 operational categories for the entire job, each mining block, and each lift of each mining block. When the user can provide volume estimates for the mining blocks and lifts, the above costs are recalculated and output on a per cubic yard of spoil handled basis. In addition, the net and per ton profit or loss is calculated for the entire job and for each mining block. A variety of additional outputs include stripping ratios and spoil movement data.

The time requirements for using COSTSUM are minor on a day to day basis. If a person who is regularly at the site consents to keep daily records, our experience has shown that generally no more than 10 minutes of his
or her time per day are required. Another 10 minutes or so per day are required to enter the daily data into computer files. The time required to obtain block volume, coal production and price, and hourly machine cost figures will vary depending upon access to mining firm records and personnel. Once required data are on the computer, it takes approximately one hour to assemble input files and run the seven data analysis programs of COSTSUM.
ASSUMPTIONS

The primary assumptions of COSTSUM are that the major costs of coal surface mining are machinery operation and labor used to perform specific tasks such as run machinery. Thus, a per hour operating cost is assigned to each machine on the site, and the primary components of the total cost of mining are the total hours of operation x hourly operating cost for each machine on the site. Aside from machinery and labor used for machinery operation and other defined tasks, only three other categories of cost are recognized: coal hauling, supplies, and overhead. It is assumed that some portion of the coal produced is hauled from the site by a contract hauling firm at a per ton rate; however, any portion or all of the coal tonnage may be exempted from incurring this cost. The quantities and costs of seeding and blasting supplies are input by the user and totaled by the computer. Seeding supplies are accounted on an area treated, or on a "batches" of seeding mix applied, basis. Blasting supplies may be accounted on per-lb.-of-explosive, per-hole, and per-foot-of-hole bases. All other costs are lumped into one category: overhead. These are entered as a per-ton-of-coal-produced figure; COSTSUM calculates
the total overhead cost as the product of the per-ton input and the coal tonnage produced over the mining period. However, the user may instruct COSTSUM to consider certain machinery operation costs as overhead, as discussed below. Furthermore, it is assumed that the per-hour machinery operation and other operating costs remain constant over the period of study.

The primary objective of the mining operation is assumed to be mining coal; thus, no other classes of revenue are recognized. Also, the overall mining site is seen as a series of mining "blocks," or areas, and total job profitability is seen as the result of a series of decisions regarding the mining of certain blocks. Those decisions include whether each block should be mined and, if the block is mined, what spoil-handling practices should be used to remove the overburden. The objective of the COSTSUM programs is to assign all costs to the mining blocks, and to break those costs down in detail. Thus, the results of the decisions made for each mining block, in terms of their effect upon profitability, can be assessed.

The assumptions used in assigning costs to mining blocks are fairly straightforward. First, all costs of overburden handling are charged to the block of spoil origin. Thus, all costs of blasting, dozing, carrying, loading, hauling, and reclaiming mined material are charged to the mining block where the spoil originated. Likewise, costs of coal augering, loading, and hauling
are charged to the block of coal origin. Finally, since the primary cost of mining coal is moving overburden, the total overhead cost is distributed to the mining blocks on a per undisturbed or bank cubic yard (bcy) of overburden basis.
DATA REQUIREMENTS

The primary data required are detailed accounts of the operation of each machine at the site (Tables 1-5). We have found that this information is best recorded at the conclusion of each working day by the job foreman on forms developed specifically for that purpose (Fig. 1). Four primary machine types can be handled by COSTSUM: haulers, loaders, dozers, and drills. Currently, the programs are capable of working with data from sites where two types of dozers (A and B; typically, a mining dozer and a reclamation dozer) and three types of loaders (A, B, and C) are used; multiple units of each machine type can be handled. However, it is assumed that the majority of the overburden pushing and carrying will be performed by the A dozers and the A loaders, respectively (Tables 2 and 3). The system also accepts data on the use of water trucks, a coal auger, coal haulers, road graders, backhoes, seeding equipment, and labor. A person with a moderate knowledge of FORTRAN programming should be able to expand the program's ability to handle additional types of machinery in a few hours, based upon the information in this manuscript.
DAILY RECORD OF MACHINERY OPERATION AND MINING PROGRESS - PONY RUN COAL CO.

Date: ________________  Day: M Tu W Th F S (check one)

Person filling out form: ________________________________

Weather: ________________________________

1. HAULERS

<table>
<thead>
<tr>
<th>Machine</th>
<th>NH</th>
<th>Haul From</th>
<th>Haul To</th>
<th>Distribution</th>
<th>Distance</th>
<th>Cycle Time</th>
<th>Loads</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. 1</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>Block #</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Comments:

NH: No. of hours hauling to each location or set of locations.

Haul From: 1 = 992 #1  2 = 992 #2  3 = 992 #3 or 988 or 945B

Haul To: Please write Block # or #'s; if a Block number does not describe the location, please describe as comment.

Distribution: If hauler(s) haul to more than one location during "NH" time period, please indicate proportion hauled to each.

Distance: Please estimate average one-way haul distance, in feet.

Cycle Time: If you estimate an average round trip cycle time (including time to load) please record.

Loads: Number of loads hauled (to be filled in by CZ from data recorded by drivers.)

Figure 1: Example of form used for recording daily site data for input to COSTUM.
2. LOADERS

<table>
<thead>
<tr>
<th>Machine</th>
<th>NH</th>
<th>Block No.</th>
<th>Lift No.</th>
<th>Operation</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>992</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>945 B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>988</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Carry & Dump: Approx. one way carry distance = ____ ft. to Block____

Comments:

**NH:** Number of hours machine was operated, production time only.

**Block No.:** Location of Operation (H1 = first hollow, H2 = second hollow, DS = Dump Site.)

**Bench No.:** (For L1, L2, CD only) 1 = top bench, 2 = second bench, X = To Coal

**Operation:** If "NH" time period is used for more than one operation, please distribute NH hours among operations, or indicate time distribution using fractions or percentages.

- L1 = Load 1 hauler
- L2 = Load 2 haulers
- LC = Load Coal, clean coal, prepare coal for loading
- CD = Carry and Dump
- R = Road work (no need to record routine haul road scraping)
- DS = Dumpsite
- O = Other (please comment)

**Material:**
- TS = Top Soil
- TSS = Top Soil Substitute
- SO = Shot Overburden
- Tx = Toxic material, special handling
- C = Coal
- O = Other

Figure 1. Continued.
3. **DOZER**

<table>
<thead>
<tr>
<th>Machine</th>
<th>NH</th>
<th>Location</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>D9#1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D9#2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FL 14</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PO:** Approx. push distance = _____ ft. from_____ Lift of Block_____ to Block _____. (1 = Top Lift, 2 = second, X = To Coal)

**Comments:**

**NH:** Number of Hours of operation, production time only

**Location:** Please write Block Number or Block Numbers

**Operation:** If "NH" time period is used for more than one operation, please distribute "NH" hours among operations, or indicate time distribution using fractions or percentages. If this is impossible, please explain.

- **ST** = Strip Topsoil
- **PD** = Prepare Drill Bench
- **FO** = Feed Overburden to loader
- **PO** = Push Overburden from one location to another
- **DS** = work Dump Site
- **Rec** = Reclamation; grade or prepare surface for seeding
- **HB** = Hollow fill, work Body
- **HF** = Hollow fill, work Face
- **HD** = Hollow fill, work Drainage
- **HT** = Hollow fill, work Toe
- **R** = Road work within mining area (no need to record routine haul road scraping)
- **O** = Other (please comment)

Figure 1. Continued.
4. DRILLING AND BLASTING

<table>
<thead>
<tr>
<th>Block. No.</th>
<th>Lift No.</th>
<th>X if Drill Hours:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>To Coal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of Holes Drilled: _________ Average Depth: _________ ft.

Diameter (if not 5\(\frac{7}{8}\)): _______ in.

Lb. ANFO: _________ Hrs. Labor to fetch, load, and shoot: _______

5. MISCELLANY

Labor: _________ men, _________ hours each, plus _________

Additional machinery and hours of use

Did anything unusual happen today to disrupt normal operation?

6. AUGER

<table>
<thead>
<tr>
<th>Location</th>
<th>Hours of operation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of man hours to operate or set up

Number of holes hole diameters

(foot) (sections) of average hole depth

7. COAL LOADED TODAY - Please record pit dimensions on map

average thickness of seam (inches)

number of loads stripped from Block

number of loads stripped from Block

number of loads augured from Block

8. LONG TERM RECORDS

Fuel Delivered today: _________ gallons

Long Term Coal Production Figures: _________ Tons produced

From _________ To _________

Figure 1. Continued.
Table 1: Column format for file HAULER INDATA, the input to program HAULER FORTRAN (1).

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>A5</td>
<td>HLDATE</td>
<td>Calendar date.</td>
</tr>
<tr>
<td>9-12</td>
<td>A4</td>
<td>HLBLK</td>
<td>Location of operation; source block for spoil handling operations.</td>
</tr>
<tr>
<td>14</td>
<td>A1</td>
<td>HLLIFT</td>
<td>Lift of source block</td>
</tr>
<tr>
<td>19-22</td>
<td>A4</td>
<td>HLDEST</td>
<td>Spoil disposal area, for spoil movement operation.</td>
</tr>
<tr>
<td>24-26</td>
<td>I3</td>
<td>RECPCT</td>
<td>Percentage of hauling time accounted as reclamation expense of spoil disposal area.</td>
</tr>
<tr>
<td>30-34</td>
<td>F5.2</td>
<td>HLHRS</td>
<td>Hours of operation.</td>
</tr>
<tr>
<td>38-40</td>
<td>I3</td>
<td>HLOADS</td>
<td>Number of loads hauled.</td>
</tr>
<tr>
<td>44-47</td>
<td>I4</td>
<td>HLDIS</td>
<td>Haul distance.</td>
</tr>
<tr>
<td>50-51</td>
<td>A2</td>
<td>HLMATL</td>
<td>Material hauled.</td>
</tr>
<tr>
<td>55-79</td>
<td>A25</td>
<td>HLCOMM</td>
<td>Comment.</td>
</tr>
</tbody>
</table>

1. Variables HLDATE, HLDIS, HLMATL, and HLCOMM for user reference, not processed by HAULER FORTRAN.

For the purposes of data recording, the site is divided into a number of different areas; each is given a four-character identifying symbol. These areas will primarily be two types: mining blocks and reclamation (spoil disposal) areas. The mining blocks are subdivided into lifts; a maximum of 5 lifts can be handled by the program code. It may also prove useful to define other areas where significant machine operation costs might be incurred (haul roads, sediment ponds, etc.). All machinery operations are recorded by
Table 2: Column format for file LOADER INDATA, the input to program LOADER FORTRAN (1).

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>A5</td>
<td>LDDATE</td>
<td>Calendar date.</td>
</tr>
<tr>
<td>9-12</td>
<td>A4</td>
<td>LDBLK</td>
<td>Location of operation; source block for spoil handling operations.</td>
</tr>
<tr>
<td>14</td>
<td>A1</td>
<td>LDLIFT</td>
<td>Lift of source block.</td>
</tr>
<tr>
<td>23-24</td>
<td>A2</td>
<td>LDOP2</td>
<td>Secondary operation, if performed simultaneously with primary.</td>
</tr>
<tr>
<td>29-33</td>
<td>F5.2</td>
<td>LDHRS</td>
<td>Hours of operation.</td>
</tr>
<tr>
<td>34</td>
<td>A1</td>
<td>LDMACH</td>
<td>Machine (2); Loader A B: Loader B C: Coal Loader</td>
</tr>
<tr>
<td>40-42</td>
<td>I3</td>
<td>LDCDIS</td>
<td>Carry Distance (CD only)</td>
</tr>
<tr>
<td>44-47</td>
<td>A4</td>
<td>LDDEST</td>
<td>Spoil disposal area: CD destination, DS location.</td>
</tr>
<tr>
<td>50-51</td>
<td>A2</td>
<td>LDMATL</td>
<td>Material.</td>
</tr>
<tr>
<td>55-79</td>
<td>A25</td>
<td>LDCOMM</td>
<td>Comment.</td>
</tr>
</tbody>
</table>

1. Variables LDDATE, LDCDIS, LDMATL, and LDCOMM for user reference, not processed by LOADER FORTRAN.
2. Blank character designates loader A.
Table 3: Column format for file DOZER INDATA, the input to program DOZER FORTRAN (1).

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>A5</td>
<td>DZDATE</td>
<td>Calendar date.</td>
</tr>
<tr>
<td>9-12</td>
<td>A4</td>
<td>DZBLK</td>
<td>Location of operation; source block for spoil handling operations.</td>
</tr>
<tr>
<td>14</td>
<td>A1</td>
<td>DZLIFT</td>
<td>Lift of source block.</td>
</tr>
<tr>
<td>19-23</td>
<td>F5.2</td>
<td>DZHRS</td>
<td>Hours of operation.</td>
</tr>
<tr>
<td>23-24</td>
<td>A2</td>
<td>DZOP2</td>
<td>Secondary operation, if performed simultaneously with primary.</td>
</tr>
<tr>
<td>29-33</td>
<td>F5.2</td>
<td>DZHRS</td>
<td>Hours of operation.</td>
</tr>
<tr>
<td>34</td>
<td>A1</td>
<td>DZMACH</td>
<td>Machine (2): primary dozer (A) C: other dozer.</td>
</tr>
<tr>
<td>40-42</td>
<td>I3</td>
<td>DZDIST</td>
<td>Push distance (PO only).</td>
</tr>
<tr>
<td>44-47</td>
<td>A4</td>
<td>DZDEST</td>
<td>Spoil disposal area: PO destination, DS location.</td>
</tr>
<tr>
<td>50-51</td>
<td>A2</td>
<td>DZMATL</td>
<td>Material.</td>
</tr>
<tr>
<td>55-79</td>
<td>A25</td>
<td>DZCOMM</td>
<td>Comment.</td>
</tr>
</tbody>
</table>

1. Variables DZDATE, DZDIST, DZMATL, and DZCOMM for user reference, not processed by DOZER FORTRAN.
2. Blank character designates primary dozer.
Table 4: Column format for file DRILL INDATA, the input to program DRILL FORTRAN (1).

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>A5</td>
<td>DRDATE</td>
<td>Calendar date.</td>
</tr>
<tr>
<td>9-12</td>
<td>A4</td>
<td>DRBLK</td>
<td>Location, mining block.</td>
</tr>
<tr>
<td>14</td>
<td>A1</td>
<td>DRLIFT</td>
<td>Lift.</td>
</tr>
<tr>
<td>20-21</td>
<td>I2</td>
<td>DRHOLS</td>
<td>Number of holes drilled.</td>
</tr>
<tr>
<td>30-31</td>
<td>I2</td>
<td>DRDEP</td>
<td>Average hole depth, feet.</td>
</tr>
<tr>
<td>40-43</td>
<td>F4.2</td>
<td>DRHRS</td>
<td>Hours of operation.</td>
</tr>
<tr>
<td>50-53</td>
<td>I4</td>
<td>DRANFO</td>
<td>Pounds of explosive.</td>
</tr>
<tr>
<td>60-62</td>
<td>F3.1</td>
<td>DRLAB</td>
<td>Labor hours required for blasting.</td>
</tr>
<tr>
<td>65-79</td>
<td>A15</td>
<td>DRCOMM</td>
<td>Comment.</td>
</tr>
</tbody>
</table>

1. Variables DRDATE and HLCOMM for user reference, not processed by DRILL FORTRAN.

location and the area to which the cost of operation is to be charged.

For each machine, the hours of operation and location are recorded for each operation performed during the working day. If the purpose of that operation is movement of overburden or soil, the area where that material is taken (the destination, or spoil disposal area) is also recorded. Symbols and operational categories recognized by the programs are listed in Tables 2, 3, and 5. Additional data are recorded for drilling and blasting operations: number of holes, average depth, pounds of explosive, and the labor required to load holes and set off the shot. If
Table 5: Column format for file OTHER INDATA, the input to program OTHER FORTRAN (1).

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>A5</td>
<td>MIDATE</td>
<td>Calendar date.</td>
</tr>
<tr>
<td>10-11</td>
<td>I2</td>
<td>LABHRS</td>
<td>Labor hours.</td>
</tr>
<tr>
<td>13</td>
<td>I1</td>
<td>JOBMEN</td>
<td>Number of employees on job.</td>
</tr>
<tr>
<td>15</td>
<td>I1</td>
<td>SUP</td>
<td>Number of supervisors on job.</td>
</tr>
<tr>
<td>19-22</td>
<td>A4</td>
<td>COLBLK</td>
<td>Coal removal block.</td>
</tr>
<tr>
<td>23</td>
<td>A1</td>
<td>AUGCOL</td>
<td>'A' if coal removed is auger coal.</td>
</tr>
<tr>
<td>25-28</td>
<td>I4</td>
<td>COLTON</td>
<td>Tons of coal removed.</td>
</tr>
<tr>
<td>30-33</td>
<td>I4</td>
<td>FUEL</td>
<td>Gallons of fuel delivered.</td>
</tr>
<tr>
<td>36-38</td>
<td>A3</td>
<td>OMAC</td>
<td>Other machinery used or labor hours to be charged to specific cost category: MAC: coal hauler, GRD: road grader, WAT: water truck, MEN: labor hours, TEM: temporary employee labor hours, BKH: backhoe, AUG: auger, SED: seeder</td>
</tr>
<tr>
<td>41-42</td>
<td>I2</td>
<td>OMACHR</td>
<td>OMAC hours</td>
</tr>
<tr>
<td>44-45</td>
<td>I2</td>
<td>AUGLAB</td>
<td>If OMAC = AUG: labor hours on auger; if OMAC = SED: number of batches or acres seeded; otherwise: cost category operation should be charged to (Table 6).</td>
</tr>
<tr>
<td>48-51</td>
<td>A4</td>
<td>OMACBL</td>
<td>Location of OMAC operation.</td>
</tr>
<tr>
<td>55-79</td>
<td>A25</td>
<td>MICOMM</td>
<td>Comment.</td>
</tr>
</tbody>
</table>

1. Variables DRDATE and HLCOMM for user reference, not processed by OTHER FORTRAN. Variables LABHRS, JOBMEN, SUP, and FUEL processed by OTHER FORTRAN but not processed by COST FORTRAN.
the hauler drivers consent to keep a daily count of the number of loads hauled, this count is also recorded. Additional daily data required by the programs include labor and supervisor hours, and coal production.

Additional data are required to define the characteristics of the mining operation (Tables 6 and 7). In our experience, block and lift volume estimates have proved most difficult to obtain (1,2). Measurement of the empty blocks after overburden removal but before the initiation of backfilling activities can give a rough estimate of block volumes. These estimates can be improved if periodic surveying enables accurate location of the mining blocks on a detailed topographic map, particularly if the user has access to topographic modeling software with volume calculation capabilities (3). Also, the user may provide estimates of the average rates of spoil movement by hauling, loader carry, and dozer push operations; providing such estimates will allow the system's capability for dealing with reclamation costs to be fully realized.

Detailed cost data are also required (Table 8). These should include hourly costs of operating machinery, blasting supply and seeding supply costs, contract coal hauling rates, and overhead costs. Machine operating costs should include the costs of ownership, depreciation, repairs, and operator wage. The only exception is the coal auger, since number of operators can vary; coal auger operator hours are input
Table 6. Formats for file DATA MOVINPUT, the input file to program MOVE FORTRAN (3).

<table>
<thead>
<tr>
<th>Priority (1)</th>
<th>'GO' (2)</th>
<th>Data (3)</th>
<th>Format (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Job average swell factor estimate (e.g.: 0.3)</td>
<td>F10.5(6-15)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Average hauler load, average loader carry volumes (lcy)</td>
<td>2F10.5(6-25)</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Default values, hauler and loader loads per hour, dozer push rate (lcy/hr.)</td>
<td>3F10.5(6-35)</td>
</tr>
<tr>
<td>4</td>
<td>HLA</td>
<td>HRSOURCE HAULER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HLP</td>
<td>LDSPERHR HAULER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDA</td>
<td>CDHOURS LOADER (A loader only)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDP</td>
<td>Route specific loader carry rate (loads/hr.)</td>
<td>CDHOURS LOADER</td>
</tr>
<tr>
<td></td>
<td>DZA</td>
<td>POHOURS DOZER (A dozer only)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DZP</td>
<td>Route specific dozer push rate (lcy/hr.)</td>
<td>POHOURS DOZER</td>
</tr>
<tr>
<td></td>
<td>VOL</td>
<td>Mining block Lift Volume (bcy) Swell (if different from average)</td>
<td>A4(5-8) A1(10) I7(12-18) F6.3(20-25)</td>
</tr>
<tr>
<td>5</td>
<td>END</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Job start date</td>
<td>A15(6-20)</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Job end date</td>
<td>A15(6-20)</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Job name</td>
<td>A15(6-20)</td>
</tr>
</tbody>
</table>

1. No required order of entry for inputs of same priority.
2. Three characters in columns 1-3 of otherwise blank record to indicate data type of records following.
3. File inputs contain data only; all headers, trailers, and records containing word 'Total' removed.
4. Columns in parentheses; formats for files as output.
Table 7. Formats for file DATA COSINPUT, the input file to program COST FORTRAN (1).

<table>
<thead>
<tr>
<th>Priority</th>
<th>'GO'</th>
<th>Data</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>COS</td>
<td>Each record contains:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- cost item number (2)</td>
<td>I2(5-6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- cost</td>
<td>F10.6(8-17)</td>
</tr>
<tr>
<td>2</td>
<td>HLR</td>
<td>HRSOURCE HAULER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDA</td>
<td>OPHOURS, A Loader</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDB</td>
<td>OPHOURS, B Loader</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDC</td>
<td>OPHOURS, C Loader</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DZA</td>
<td>OPHOURS, A Dozer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DZC</td>
<td>OPHOURS, C Loader</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DRL</td>
<td>TOTALS DRILL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MIS</td>
<td>Other machinery data from TOTALS OTHER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>COL</td>
<td>Coal data from TOTALS OTHER, plus:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- tonnage not incurring contract hauler charge</td>
<td>I5(25-29)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- price received for coal, if different from COST(22)</td>
<td>F6.3(32-35)</td>
</tr>
<tr>
<td></td>
<td>VOL</td>
<td>COSINPUT MOVE</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SPB</td>
<td>One record per special block:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'OHD' or 'REC'</td>
<td>A3(5-8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- location symbol</td>
<td>A4(11-15)</td>
</tr>
<tr>
<td>4</td>
<td>RCL</td>
<td>COSINPUT MOVE</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>RED</td>
<td>Reclamation expense redistribution location (follow with additional RCL input)</td>
<td>A4(5-8)</td>
</tr>
<tr>
<td>6</td>
<td>END</td>
<td>COSINPUT MOVE</td>
<td></td>
</tr>
</tbody>
</table>

1. Notes for table entries as for Table 6.
2. See Table 8.
Table 8. Cost data required by COST FORTRAN, with cost item numbers.

<table>
<thead>
<tr>
<th>Item Number</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Haulers, per operating hour</td>
</tr>
<tr>
<td>2</td>
<td>A loader, per operating hour</td>
</tr>
<tr>
<td>3</td>
<td>B loader, per operating hour</td>
</tr>
<tr>
<td>4</td>
<td>C loader, per operating hour</td>
</tr>
<tr>
<td>5</td>
<td>A dozer, per operating hour</td>
</tr>
<tr>
<td>6</td>
<td>C dozer, per operating hour</td>
</tr>
<tr>
<td>7</td>
<td>Drill, per operating hour</td>
</tr>
<tr>
<td>8</td>
<td>Drill and blast cost, per hole</td>
</tr>
<tr>
<td>9</td>
<td>Drill and blast cost, per foot of hole</td>
</tr>
<tr>
<td>10</td>
<td>Explosive, per pound</td>
</tr>
<tr>
<td>11</td>
<td>Labor, per hour</td>
</tr>
<tr>
<td>12</td>
<td>Coal haul truck, per operating hour</td>
</tr>
<tr>
<td>13</td>
<td>Water truck, per operating hour</td>
</tr>
<tr>
<td>14</td>
<td>Grader, per operating hour</td>
</tr>
<tr>
<td>15</td>
<td>Seeder, per operating hour</td>
</tr>
<tr>
<td>16</td>
<td>Backhoe, per operating hour</td>
</tr>
<tr>
<td>17</td>
<td>Coal auger, per operating hour</td>
</tr>
<tr>
<td>18</td>
<td>Job foreman, per labor hour</td>
</tr>
<tr>
<td>19</td>
<td>Overhead, per ton produced</td>
</tr>
<tr>
<td>20</td>
<td>Temporary employee, per labor hour</td>
</tr>
<tr>
<td>21</td>
<td>Contract coal hauling, per ton</td>
</tr>
<tr>
<td>22</td>
<td>Revenue received for coal, per ton</td>
</tr>
<tr>
<td>23</td>
<td>Seeding cost, per batch or acre</td>
</tr>
</tbody>
</table>
separately from the number of auger operating hours (Table 5), so operator wage should not not be considered a component of hourly coal auger operating cost. Contract coal hauling rates are input as a per ton figure. The overhead cost supplied by the user should incorporate all costs not specifically detailed in the above categories; it is estimated on a per ton of mined coal basis.

In order for the total cost of the job to be distributed to the mining blocks, the total cost must be reflected in the inputs (Table 8). It should be emphasized that the total of the daily labor hours recorded in the OTHER INDATA file (Table 5) is not multiplied by the labor per hour input, cost item 11, by any of the COSTSUM programs; the only labor costs calculated in this fashion are the labor hours required for blasting (DRLAB, Table 4), labor required to operate the coal auger (Table 5), and other labor hours listed specifically in columns 36 to 51 of OTHER INDATA (when OMAC is listed as MEN, Table 5). The wages of equipment operators should be included in the per hour machinery operation costs. In our study of mining costs (2), we observed that the majority of labor hours which were not listed on daily forms as spent performing specific tasks were actually spent maintaining machinery. The cost of this "residual" labor was then added to a general machinery ownership, maintainance, repair, and depreciation budget; this
budget was distributed among the machines at the site on the basis of industry average cost figures (4) and operating hours in order to estimate hourly operating costs. Another way of accounting for residual labor hours would be to include their cost in the overhead budget.

To allow calculation of overall and per block profits, the price expected or received per ton of coal must be supplied. Since that price may not be constant over the entire job, individual prices may be supplied for each block for tonnage mined by both strip and auger methods.
THE PROGRAMS

Seven data analysis computer programs form the body of COSTSUM (Figure 2). Five Level 1 programs (HAULER, LOADER, DOZER, DRILL, and OTHER) are designed to analyze the files containing daily records of machine operations. These programs compile totals of the hours spent performing each type of operation which are subtotaled by location, destination (for overburden movement operations), and other criteria. Other quantities, such as hauler loads, feet of drill hole, and pounds of explosives, are also totaled. Two additional programs (MOVE and COST) are defined as Level 2, since their primary inputs are the outputs of the Level 1 programs. If data on overburden volume, hauler capacities, and average dozer and loader material movement rates can be supplied, the program MOVE will estimate the quantities of material moved from each mining block to each spoil disposal area. If mining cost and coal price data are supplied, the program COST will provide a detailed analysis of cost and profitability. Eight subroutines accompany the seven main programs.

The programs are written in FORTRAN 77. The logic of operation of each program is emphasized by using
Figure 2. Input requirements, and input and output file names, for the seven programs of COSTSUM.
embedded comments to define and to describe the functions of integrated blocks of program code.

Similarly, important internal variables are defined with embedded comments. Certain conventions regarding variable names have been followed throughout all programs. Generally, if two or more variables are declared as the same type on the same program line, they are used to represent quantities defined by the same units. For example, in the program HAULER, the variables HLOADS and TOLDS both represent quantities of hauler loads. As an input record is read, the number of hauler loads recorded in that record is stored in the variable HLOADS, while the total number of hauler loads recorded in all the input records read thus far is accumulated in the variable TOLDS. Also, all integer variables whose names end with the three characters "DIM" represent the dimensions of one or more arrays. For example, throughout the programs, the variable BDIM represents the dimension of the BLOCK array, which is the maximum number of location symbols the programs can handle; BDIM serves to regulate performance of arithmetic operations which are repeated for each location. The variables ICOUNT and IMAX are used to protect against read errors which might result in endless loops. ICOUNT keeps a count of the number of input records read; if that number exceeds IMAX (which is set to 1000 in all programs), the program exits.
Formatted input is used throughout. Thus, it is critical that quantities be entered into the input records in the specified columns using the proper formats (Tables 1 - 7). Only three variable types are used in the programs: character, integer, and real. Appendices A through G contain examples of the input files.

Input and output are accomplished by referring to specified files by number in READ and WRITE statements. Under the VM/CMS operating system used at Virginia Tech, these numbered files are assigned names using FILEDEF statements previous to loading the compiled code and starting execution. These are placed in the program COSTSUM EXEC, which also serves to load and start the compiled FORTRAN programs on the Virginia Tech VM/CMS operating system. In this text, input and output files will be discussed by name. File names are associated with file numbers in comments embedded in each program immediately preceding the first READ statement.

The FORTRAN statements which comprise each of the seven programs are listed in the Appendices A through G of this bulletin; the eight subroutines and COSTSUM EXEC are listed in Appendix H. To illustrate use of the programs, a sample data set has been constructed and the seven programs executed. In the appendices, each program listing is preceded by the sample input and followed by selected output files. What follows is a
brief description of the purpose, logic, and output of each of the seven programs.

**Level 1 Programs**

The purpose of the five Level 1 programs is to total the daily operational data recorded on the mine site. Their methods of operation are similar in many respects. Each requires an input file composed of a portion of the daily data. The first four programs total the data on operation of a particular type of machine, while the fifth (OTHER) accepts all additional daily data. The outputs of the Level 1 programs consist of machinery operation totals assembled under various categories. Those output files which are to be input to Level 2 programs also contain three character variables in columns 1 through 3 of selected records, generally following headers but preceding data. The purpose of these three character symbols is to key the Level 2 programs' interpretation of the associated input data.

The logical structures of these five programs are also similar (Fig. 3). To begin, all program variables are declared and arrays are declared and dimensioned. The next step is to give variables and arrays initial values where required. For the most part, this step is accomplished with assignment statements and DO loops. However, initialization of the arrays which are assigned values by subroutines (block and lift symbol arrays, operation symbol arrays in LOADER and DOZER) is
INITIALIZE VARIABLES

READ INPUT RECORD

ASSIGN INTEGERS TO IDENTIFY LOCATION BLOCK AND LIFT, AND DESTINATION (IF APPROPRIATE)

(LOADER AND DOZER ONLY)
ASSIGN INTEGERS TO IDENTIFY MACHINE TYPE AND OPERATION TYPE

ACCUMULATE INPUT DATA QUANTITIES BY LOCATION, AND BY OPERATION AND MACHINE TYPES (LOADER AND DOZER ONLY)

END OF INPUT FILE?

no

yes

PRINT OUTPUT

EXIT

Figure 3. Generalized flowchart for the five Level 1 programs.

accomplished by calling the appropriate initialization subroutines (ABLOCK, ALIFT, ALDOP, and ADZOP, respectively).

Following initialization is a READ statement, which marks the beginning of a series of program lines that are executed for each input record. Each symbol or
quantity of the input record is read into a separate variable for temporary storage. Subroutines KBLOCK and JLIIFT are called to assign integers which identify operation location. Thus, the data are stored in a series of arrays in which one or more dimensions are coded for location using the integer identifiers. Programs LOADER and DOZER use a similar procedure (with the subroutines XLDOP and XDZOP) to store data according to type of operation. The quantities read by the input line are added to the quantities previously stored in the array positions specific to the appropriate location and/or operational type. Upon completion of the calculations associated with the individual input record, the programs return to the READ statement and the process is repeated for the next data record. If the record encountered is the end of the input file, the program goes to the first statement of the block of programming which writes the output files (7000 in LOADER and DOZER, 9000 in the others). Execution of the subsequent series of statements completes the program’s run.

HAULER FORTRAN totals hauler data (hours and loads) by block and lift of spoil origin and by destination. Thus, the output of the program tells the user how much hauling time was spent and how many loads were carried over each haul route on the site. If the variable RECPCT (Table 1) is set equal to zero (i.e. left blank), program logic is straightforward: quantities
are accumulated in three dimensional arrays (source block, source lift, destination) and output accordingly. However, a positive value for RECPCT indicates that a portion of the hauling cost should be charged to reclamation of the destination block rather than as an expense of removing spoil from the source. This variable may be used if, for example, hauling topsoil or topsoil substitute material for use as a surface medium entails a greater expense than would routine disposal. In this case, the user estimates the percentage of total hauling time which constitutes the extra cost; this time is stored in array RRR rather than HRS, the array normally used to accumulate hours due to routine hauling.

Five files are output by HAULER. HRSOURCE HAULER lists the total time spent hauling from each source location to each destination; this information is organized and totaled by source. The symbol 'R' following the destination block symbol indicates operation time to be charged as a destination expense. HOURDEST HAULER contains similar information but with a different organization: hauler hours are organized and totaled by destination rather than by source. Also, only total hauling time is listed; no distinction is made between source and destination accounts. LDSOURCE HAULER and LOADDEST HAULER list hauler load totals organized by source and destination, respectively. The LDSPERHR HAULER output file contains the loads per hour
hauling rate calculated for each hauler route; these rates are calculated only from input records where some positive number of hauler loads is recorded. Thus, a missing hauler load count is not interpreted as 0 loads hauled when these rates are calculated. The HRSOURCE and LDSPERHR files are used as Level 2 inputs.

LOADER FORTRAN and DOZER FORTRAN operate according to identical logic. Their primary activities are to accumulate time of operation data in arrays (OPHRS) dimensioned by machine, location, lift, and operation type. If a data record lists both primary and secondary operations, the hours recorded are split evenly between the two operations, as the program executes statements 2000 through 5000 twice. In addition, hours moving overburden (loader carry and dump, dozer push) are totaled in arrays CDHRS, dimensioned by machine, source block, source lift, and destination. Hours working dumpsites are totaled in arrays WBHRS, which are dimensioned by machine and dumpsite location (destination).

Execution of each program results in three output files; within each, data are totaled separately for each machine. The OPHOURS files list machine operations totaled by source block and lift. Files CDHOURS LOADER and POHOURS DOZER contain hours spent moving material from each source block to each destination. The DSHOURS files list the time spent working dumpsites at each location. The OPHOURS, CDHOURS, and POHOURS files are
required as inputs to Level 2.

**DRILL FORTRAN** is the simplest and shortest of the seven programs. It totals the quantities of each input record in a series of arrays dimensioned by block and lift. In addition, average hole depths and drilling rates are calculated for each block and lift. The output file TOTALS DRILL is used as an input to program COST.

**OTHER FORTRAN** totals all additional data recorded on a daily basis. Labor hours, labor days, supervisor days, and fuel purchases are added into simple totals. Coal tonnage is totaled by block and mining method (auger or strip). The hours worked by other machines (OMAC; Table 5) on the site are totaled by location and by cost category. The labor hours required to operate the auger in each coal block are also totaled. The number of 'batches' of seed and fertilizer applied by the seeder (or the area treated, if seeding is performed by an outside contractor) are totaled by location. All of the above quantities are output in the file TOTALS OTHER; the coal tonnage and machine operation totals are required for input to program COST.

**Level 2 Programs**

The primary inputs to **MOVE FORTRAN** and **COST FORTRAN** are the modified outputs of the Level 1 programs. **MOVE** estimates the quantities of overburden moved from each
source location to each destination, while COST provides a detailed cost analysis of the mining operation. Execution of program MOVE results in an output file that is input to COST, to direct the distribution of reclamation expenses (which have been listed at spoil disposal locations) to the sources of the reclaimed materials, the mining blocks.

The logical structures of the two programs are similar. After variable and array declaration and initialization, the programs execute a series of programming blocks designed to read and interpret the input data (Tables 6, 7, and 11); each is keyed to the GO variable, a three-character symbol which may be found in columns 1-3 of an input record. If any READ statement encounters any non-blank character in columns 1-3, program control is shifted to statement 9700. The code which follows is designed to identify the GO variable; if the symbol read from columns 1 through 3 is recognized, program control is then shifted to the programming block designed to read and interpret the input records which follow. If another non-blank GO variable is encountered, program control goes back to 9700 and the process is repeated. If the GO variable is not recognized, an error message is issued and the program exits.

The end of input is marked by a record where the characters 'END' are placed in columns 1-3 followed by three records containing the starting date of the time
period represented by the data, the ending date, and the job name. Then, the program enters a block of code which uses the stored input quantities to calculate desired totals. Finally, the totals are used to generate output files, and program execution is complete.

MOVE

The purpose of MOVE FORTRAN is to generate a file to direct the distribution of reclamation expenses to the mining blocks on the basis of the relative quantities of material disposed in each reclaimed area originating in each mining block (Figure 4). In the process of generating this information, MOVE performs a series of calculations to estimate the quantities of material moved by each of three modes (hauler, loader carry, dozer push) over each source-to-destination route. These estimates are also passed to program COST, as bank cubic yards (bcy) of material removed from each source block by each of three modes; they are used by COST to calculate the per-cubic-yard-handled costs of moving overburden by dozer and loader, and by hauler. The accuracy of the moved quantity estimates performed by program MOVE will vary, depending upon the importance placed by the user upon the resultant cost data and willingness to spend time developing accurate movement rate estimates.
START
↓
INITIALIZE VARIABLES
↓
READ FIRST THREE INPUT RECORDS
(data priority 1-3)
↓
COMPLETE INITIALIZATION
↓
READ INPUT RECORD
(GO variable, cols. 1-3)
↓
GO = 'END'?

Read 3 input records
(data priority 6-8)
↓
Calculate overburden movement totals
↓
Output "hours move"
output "bysource move"
calculate moved:topo ratios
↓
Adjust overburden movement totals with moved:topo ratios
↓
Output "bydest move"
↓
For each spoil movement destination, calculate fraction of moved material originating in each source
↓
Recalculate lift volumes as bcv
Calculate total bcv hauled, carried, and pushed from each lift
↓
Output "cosinput move"
EXIT

Figure 4. Generalized flowchart for the program MOVE; section of flowchart enclosed in dotted lines represents 8 separate programming blocks each designed to read and interpret data records keyed to individual GO symbols.
Inputs to MOVE include the undisturbed (bank cubic yard, or bcy) volumes of each lift, and the "swell" expected upon disturbance (5). These quantities are used to calculate the amount of material (in loose cubic yards, or lcy) which must be removed from each block in order to expose the coal. The primary inputs used to calculate moved-volume estimates are hours spent hauling, carrying, and pushing material over each transport route (Level 1 output files), and estimates of the rates of material movement. For loaders and haulers, movement rates are estimated as a product of the loads carried or hauled per hour and average load lcy volumes. For the dozer, the rates are estimated directly as an lcy per hour quantity. Average hauler and loader loads per hour, and dozer push rates, are estimated by the user for the entire job as default values; the user may override these defaults by estimating the route specific rates, where that information is available. For haulers, this task is easily done by using the LDSPERHR output file as an input to MOVE.

An initial series of transport quantity estimates is calculated as the product of hours x rate for each transport route. These estimates are refined by comparing the total calculated quantity of material removed from each lift of each mining block to the lcy volume of the lift calculated directly from the inputs and adjusting accordingly.
Four output files are generated by executing the program. **HOURS MOVE** is a direct output of the operating hours spent moving material from each source block and lift to each destination by each of the three modes (haul, carry, push). **BYSOURCE MOVE** is a summary of the hours x rate calculations performed on the input quantities, organized and totaled by mining block; it also lists "moved:topo ratios," the proportion of the input lcy volume represented by the quantity calculated as having been removed from the source lift and block before adjustment. **BYDEST MOVE** contains the moved volume estimates after adjustment with the moved:topo ratios. **COSINPUT MOVE** contains the input block volumes, the volumes calculated as being removed from each block by each of the three modes, and a listing of the proportion of the material disposed in each disposal area originating in each source. Data is written to the COSINPUT file in a format suitable for direct input to COST FORTRAN.

**COST**

The purpose of COST FORTRAN is to provide the user with a detailed analysis of the costs of mining and reclamation so as to allow assessment of the effects of those costs upon mining profitability (Figure 5). The method used in programming to meet this purpose is to provide a system whereby all mining expenses are charged to the mining blocks, or areas where spoil is
Figure 5. Generalized flowchart for the program COST; section of flowchart enclosed with dotted lines represents 10 separate programming blocks, each designed to read and interpret data records keyed to specified GO symbols (Table 11). Variable names defined in Table 10.
removed to expose coal. The expenses are further broken down into 15 cost categories (Table 9); these are calculated on a gross basis and recalculated on a per-yard basis for spoil handling operations and a per-ton basis for coal handling operations. Thus, the user is able to identify the mining areas where spoil was handled in cost effective fashion and those areas
Table 9. Cost categories used by COST FORTRAN, with category numbers.

<table>
<thead>
<tr>
<th>Number</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Clear and bench, strip topsoil</td>
</tr>
<tr>
<td>2</td>
<td>Drill and blast</td>
</tr>
<tr>
<td>3</td>
<td>Loader carry and dozer push</td>
</tr>
<tr>
<td>4</td>
<td>Load and haul, loader</td>
</tr>
<tr>
<td>5</td>
<td>Load and haul, dozer feed to loader</td>
</tr>
<tr>
<td>6</td>
<td>Load and haul, hauling</td>
</tr>
<tr>
<td>7</td>
<td>Load and haul, working dumpsite</td>
</tr>
<tr>
<td>8</td>
<td>Load and haul, total (4 + 5 + 6 + 7)</td>
</tr>
<tr>
<td>9</td>
<td>Prepare and load coal</td>
</tr>
<tr>
<td>10</td>
<td>Auger coal</td>
</tr>
<tr>
<td>11</td>
<td>Haul coal</td>
</tr>
<tr>
<td>12</td>
<td>Reclamation expenses at area of occurrence</td>
</tr>
<tr>
<td>13</td>
<td>Other</td>
</tr>
<tr>
<td>14</td>
<td>Overhead</td>
</tr>
<tr>
<td>15</td>
<td>Reclamation expenses after redistribution to source</td>
</tr>
</tbody>
</table>

where spoil handling was most costly, and (through comparisons of individual block category costs to one another and to site averages) the reasons for the differences.

Costs incurred while removing overburden to expose coal are charged directly to the mining block on the basis of machine operating hours and user-supplied hourly operating costs. However, two costs which cannot
Table 10. Major internal variables and quantities represented, program COST.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Variable Names (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dollar cost per cost item</td>
<td>COST(C)</td>
</tr>
<tr>
<td>Total cost of operations</td>
<td>BLKOST(B,L), CCCOST(CC)</td>
</tr>
<tr>
<td></td>
<td>BLKTOT(B), LIFTOT(B,L), LIKOST(B,L,CC)</td>
</tr>
<tr>
<td>Total overhead cost</td>
<td>OHDTOT</td>
</tr>
<tr>
<td>Price received per ton coal:</td>
<td></td>
</tr>
<tr>
<td>auger</td>
<td>AUGPRI(B)</td>
</tr>
<tr>
<td>strip</td>
<td>STRPRI(B)</td>
</tr>
<tr>
<td>Coal tonnage:</td>
<td></td>
</tr>
<tr>
<td>strip</td>
<td>SCOAL</td>
</tr>
<tr>
<td>auger</td>
<td>ACOAL, AUGCOL(B)</td>
</tr>
<tr>
<td>total</td>
<td>TCOAL, BLKCOL(B)</td>
</tr>
<tr>
<td>Source (block,lift) volumes total</td>
<td></td>
</tr>
<tr>
<td>moved by hauler</td>
<td></td>
</tr>
<tr>
<td>moved by loader carry (LC)</td>
<td></td>
</tr>
<tr>
<td>moved by dozer push (PO)</td>
<td></td>
</tr>
<tr>
<td>moved by LC and PO</td>
<td></td>
</tr>
<tr>
<td>Total hours to operate individual machines (2) (HRSxxx)</td>
<td>HRSHLR, HRSLDA, HRSLDB, HRSLDC, HRSZDA, HRSZC, HRSZRL, HRSMIS</td>
</tr>
<tr>
<td>Total cost to operate machinery types (2) (xxCOST)</td>
<td>HLCOST, LDCOST, DZCOST, DRCOST, MICOST</td>
</tr>
<tr>
<td>Cost of labor to load holes and set of shot</td>
<td>SHCOST</td>
</tr>
</tbody>
</table>

1. Level at which quantities are accumulated represented by array subscripts: B = block, L = lift, C = cost item, CC = cost category. Non-subscripted variables accumulate totals for the entire job.

2. HRSMIS and MICOST represent total hours of operation and cost of operation, all OMAC machines and labor (Table 5).
Table 11. Primary operations performed by programming code associated with data READ statements, by GO symbol, program COST.

<table>
<thead>
<tr>
<th>GO Symbol</th>
<th>Primary Operations (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COS</td>
<td>Read C, COST(C)</td>
</tr>
<tr>
<td></td>
<td>If C = 22, then</td>
</tr>
<tr>
<td></td>
<td>AUGPRI(B) = COST(22)</td>
</tr>
<tr>
<td></td>
<td>STRPRI(B) = COST(22), for all B</td>
</tr>
<tr>
<td>HLR LDA</td>
<td>Read location, hours of operation, type of operation (loaders, dozers, and OMAC only)</td>
</tr>
<tr>
<td>LDB LDC</td>
<td>Assign array subscripts B,L,C,CC</td>
</tr>
<tr>
<td>DZA DZC</td>
<td>LIKOST(B,L,CC) = LIKOST(B,L,CC) + COST(C)*Hours</td>
</tr>
<tr>
<td>DRL MIS</td>
<td>HRSxxx = HRSxxx + Hours</td>
</tr>
<tr>
<td></td>
<td>xxCOST = xxCOST + COST(C)*Hours</td>
</tr>
<tr>
<td>COL</td>
<td>Read location, coal tonnage (COAL), 'AUGER' if auger coal, tonnage exempt from contract hauler charge (SECOHL), price per ton (PRICE)</td>
</tr>
<tr>
<td></td>
<td>Assign array subscript B</td>
</tr>
<tr>
<td></td>
<td>BLKCOL(B) = BLKCOL(B) + COAL; TCOAL = TCOAL + COAL</td>
</tr>
<tr>
<td></td>
<td>If 'AUGER': AUGCOL(B) = AUGCOL(B) + COAL; ACOAL = ACOAL + COAL</td>
</tr>
<tr>
<td></td>
<td>If not 'AUGER': SCOAL = SCOAL + COAL</td>
</tr>
<tr>
<td></td>
<td>COCOHL = COAL - SECOHL</td>
</tr>
<tr>
<td></td>
<td>LIKOST(B,4,11) = LIKOST(B,4,11) + COCOHL*COST(21)</td>
</tr>
<tr>
<td></td>
<td>If PRICE greater than 0.10:</td>
</tr>
<tr>
<td></td>
<td>STRPRI(B) = PRICE, or</td>
</tr>
<tr>
<td></td>
<td>(if 'AUGER') AUGPRI(B) = PRICE</td>
</tr>
<tr>
<td>VOL</td>
<td>Read Block and lift, bcy volume (VOLUME), volumes hauled (H), carried (CA) and pushed (P) from that lift</td>
</tr>
<tr>
<td></td>
<td>Assign array subscripts B,L</td>
</tr>
<tr>
<td></td>
<td>LIFVOL(B,L) = LIFVOL(B,L) + VOLUME</td>
</tr>
<tr>
<td></td>
<td>BLKVOL(B) = BLKVOL(B) + VOLUME</td>
</tr>
<tr>
<td></td>
<td>HAUL(B,L) = H</td>
</tr>
<tr>
<td></td>
<td>CARRY(B,L) = CA</td>
</tr>
<tr>
<td></td>
<td>PUSH(B,L) = P</td>
</tr>
<tr>
<td>RCL</td>
<td>For each block (2)</td>
</tr>
<tr>
<td></td>
<td>Write LIKOST(B,L,12) to file RECLAM COST, all L</td>
</tr>
<tr>
<td></td>
<td>Set RCLEXP(B) = LIKOST(B,L,12) totaled for all L</td>
</tr>
<tr>
<td></td>
<td>Set LIKOST(B,L,12) = 0.0, all L</td>
</tr>
<tr>
<td></td>
<td>Read destination block, fraction of total material disposed at destination originating at source (FRAC), source block, source lift</td>
</tr>
<tr>
<td></td>
<td>Assign array subscripts SB (source block), SL (source lift), DB (destination block)</td>
</tr>
<tr>
<td></td>
<td>X = FRAC*RCLEXP(DB)</td>
</tr>
<tr>
<td></td>
<td>LIKOST(SB,SL,15) = LIKOST(SB,SL,15) + X</td>
</tr>
<tr>
<td></td>
<td>Write BLOCK(SB), LIFT(SL), FRAC, BLOCK(DB), X to file RECLAM COST</td>
</tr>
</tbody>
</table>
Table 11. Continued.

<table>
<thead>
<tr>
<th>GO Symbol</th>
<th>Primary Operations (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPB</td>
<td>Read special block symbol (CHAR), location</td>
</tr>
<tr>
<td></td>
<td>If CHAR = 'REC'</td>
</tr>
<tr>
<td></td>
<td>Assign array subscript B; set T = 0.0</td>
</tr>
<tr>
<td></td>
<td>For each CC:</td>
</tr>
<tr>
<td></td>
<td>X = total LIKOST(B,L,CC), all L</td>
</tr>
<tr>
<td></td>
<td>Write X to file RECLAM COST</td>
</tr>
<tr>
<td></td>
<td>Total T = T + LIKOST(B,L,CC) for all L,CC</td>
</tr>
<tr>
<td></td>
<td>LIKOST(B,1,12) = T</td>
</tr>
<tr>
<td></td>
<td>If CHAR = 'OHD'</td>
</tr>
<tr>
<td></td>
<td>Assign array subscript B; set T = 0.0</td>
</tr>
<tr>
<td></td>
<td>For each CC:</td>
</tr>
<tr>
<td></td>
<td>X = total LIKOST(B,L,CC), all L</td>
</tr>
<tr>
<td></td>
<td>Write X to file RECLAM COST</td>
</tr>
<tr>
<td></td>
<td>Total T = T + LIKOST(B,L,CC) for all L,CC</td>
</tr>
<tr>
<td></td>
<td>OHDTOT = OHDTOT + T</td>
</tr>
<tr>
<td></td>
<td>Set LIKOST(B,L,CC) = 0.0 for all L,CC</td>
</tr>
<tr>
<td></td>
<td>in Block B</td>
</tr>
<tr>
<td>RED</td>
<td>Read location symbol (BLK)</td>
</tr>
<tr>
<td></td>
<td>Assign array subscript B</td>
</tr>
<tr>
<td></td>
<td>For all L:</td>
</tr>
<tr>
<td></td>
<td>LIKOST(B,L,12) = LIKOST(B,L,12) + LIKOST(B,L,15)</td>
</tr>
<tr>
<td></td>
<td>LIKOST(B,L,15) = 0.0</td>
</tr>
<tr>
<td>END</td>
<td>Read job start date (DAYONE)</td>
</tr>
<tr>
<td></td>
<td>Read job end date (DAYEND)</td>
</tr>
<tr>
<td></td>
<td>Read job name (JOB)</td>
</tr>
<tr>
<td></td>
<td>Write JOB, DAYONE, DAYEND to all output files</td>
</tr>
<tr>
<td></td>
<td>Exit data read</td>
</tr>
</tbody>
</table>

1. Array subscripts: B = block, L = lift, C = cost item, CC = cost category; subscripts B and L assigned by subroutines KBLOCK and JLIFT.

be directly charged are reclamation expenses and overhead. As stated previously, reclamation expenses are distributed to mining blocks in amounts proportionate to quantities of reclaimed spoil originating in each block. Overhead costs are also
distributed on a per-bcy basis rather than the per-ton basis commonly used in the coal industry. Our reason for this method is our interest in spoil-handling costs; a per-ton basis for overhead cost distribution will have disproportionate effects upon the calculated per-bcy spoil-handling costs of mining various blocks due to differences in stripping ratios. Since the primary cost of surface mining is the movement of overburden, not the movement of coal, we feel that a per-ton overhead cost distribution system gives a distorted picture of block to block spoil handling cost comparisons. However, since the per-ton basis is more commonly used by industry, overhead cost is input as a per-ton quantity.

The program distributes overhead costs after reading all inputs (statements 8810 +). Costs which are considered as overhead include any OMAC operation listed as cost category 14 in OTHER INDATA regardless of location. The total cost of operating machinery in any location is considered as overhead if it is identified by the user as a special block (after setting GO equal to 'SPB') with the symbol OHD in the DATA COSINPUT file (data priority 5, Table 7). Costs which can be conveniently handled in this fashion may include construction and maintenance of roads to the site or sediment ponds and other environmental control structures. In addition, the input overhead cost per ton (cost item 19) is multiplied by the total tonnage
produced and added to the overhead total; that total is then distributed to the mining blocks and lifts in direct proportion to their undisturbed (bcy) volumes.

The primary operations considered as reclamation are dozer grading, seeding, and hauling identified as reclamation costs by the user with the RECPCT variable (Table 1). In addition, if the user identifies a location as a special block with the symbol 'REC', all operations performed at that location will be considered as reclamation costs (category 12). Thus, no special operation symbols are required to categorize daily dozer and loader operations in these areas as reclamation expenses, and a printout of the costs by category previous to placement of all costs in category 12 is obtained. This option has proved useful for handling the costs of constructing excess spoil disposal facilities such as hollow fills.

If the GO variable 'RCL' is placed in the DATA COSINPUT file and followed by the appropriate inputs, the program will redistribute all spoil disposal area reclamation expenses (category 12) to the source blocks (category 15). That set of appropriate inputs should consist of a listing of the spoil disposal areas; each disposal area symbol is repeated for each mining block which acted as a source of material disposed in that area, and is accompanied by the the fraction of total spoil disposed in that area which originated in that mining block (Appendix G); this set of inputs can be
taken directly from the COSINPUT MOVE output file. However, the above procedure may cause destination reclamation expenses to be distributed to an area which has acted as a source of material but is not itself a mining block. For example, if topsoil removal is required for the stability of hollow fill 2 (HF2), that topsoil may be disposed in hollow fill 1 (HF1), causing a portion of the HF1 construction costs to be distributed to HF2, a non mining area. This problem can be solved by inserting a record in the input file which sets GO equal to 'RED' followed by a record designating HF2 (data priority 5, Table 7). This procedure causes HF2 category 15 expenses to be moved to HF2 category 12. A second use of the RCL option will now redistribute these expenses to the appropriate mining blocks.

Execution of COST FORTRAN results in four output files. SUMMARY COST contains the most general output; included are a listing of the input costs, a profit summary for each block, total job costs by cost category, machine cost and operating hours totals, and block and lift cost summaries which include stripping ratios, and per-bcy and per-ton cost totals. PERBLOCK COST contains a listing of all costs for each mining block, by lift and by cost category. There are two separate listings in PERBLOCK COST for each block, the first containing cost totals and the second containing per-block-bcy overburden-handling costs and per-ton
coal handling costs. PERYARD COST contains the same per-bcy and per-ton costs, but in a format which facilitates cost comparisons among the mining blocks. In addition, the costs of moving material from each source block by load and haul operations (category 8) and by loader carry and dozer push operations (category 3) are listed per bcy of spoil handled by each operation. A record of the distribution of overhead and reclamation costs is placed in file RECLAM COST.
USE AND MODIFICATION

This set of programs has been used extensively on an IBM 3084 mainframe computer with a VM/CMS operating system and a VS FORTRAN compiler. The programs will run on other operating systems making use of other FORTRAN 77 compilers as is, or with minor modifications. On some systems, it may be necessary to define input and output file names with OPEN statements within the programs (6). The only library functions used are those which transform integer variables to real, and vice versa: IFIX, REAL, and FLOAT.

In order to use COSTSUM, the user must first assemble the necessary input data and construct input files for the five Level 1 programs. However, before running any programs, the user may wish to modify subroutines ABLOCK and KBLOCK, to facilitate processing of a set of location symbols appropriate to the site under study. Modification of ALIFT and KLIFT may also be desirable. The present system uses T to designate the top lift, 2 to designate the second, 3 to designate the third, X to designate the lift just above the coal seam (lift number 4), and R to designate the fifth; the R lift symbol is used for rehandled material. If no lift symbol is provided (reclamation
areas and sediment ponds, for example), JLIFT assigns that operation to lift number 5.

The next step is to run the five Level 1 programs. After all five of these programs have been executed successfully, an input file should be assembled for program MOVE (Table 6; Appendix F). Data of priorities 1 through 3 is entered into file DATA MOVINPUT manually. The fourth record should contain a GO variable appropriate to priority 4 data. The minimum priority 4 data set will consist of data keyed to GO symbol 'HLA', 'LDA', or 'DZA', and data keyed to 'VOL'. The data keyed to GO variables 'HLA', 'HLP', 'LDA', and 'DZA' may consist of the modified outputs of Level 1 programs. Required modifications consist of removal of headers and footers, and removal of records containing the word 'TOTAL'; blank records may remain or be removed, at user option. The 'LDP' and 'DZP' data sets are read by MOVE in near identical formats as the 'LDA' and 'DZA' data sets, the only difference being that two decimal places are acceptable in 'LDP' and 'DZP'. Thus, duplicates of the CDHOURS LOADER and POHOURS DOZER file segments used as 'LDA' and 'DZA' inputs can be used as "masks" if the user elects to assemble 'LDP' and 'DZP' inputs; correct column placement is facilitated by overlaying existing F5.1 data with loader and dozer spoil movement rates in F5.2 format. The data keyed to 'VOL' and 'END' are input manually. If the user elects to have the program use the job average swell factor
(data priority 1) for any lift or lifts, a decimal point (or 0.0) entry is required for the swell factor in the corresponding 'VOL' input records, or a data read error may result.

To begin assembly of file DATA COSINPUT (Table 7; Appendix G), the GO symbol 'COS' is manually entered on the first record and followed by 23 records, each containing a a cost item number and corresponding dollar cost figure (Table 8). The 25th data record should consist of a priority 2 GO variable. Data keyed to 'HLR', 'LDA', 'LDC', 'DZA', 'DZC', and 'DRL' are taken directly from Level 1 output files. Again, manual removal of headers, footers, and records containing the word 'TOTAL' is required. Data keyed to 'MIS' and 'COL' come from the TOTALS OTHER output file. To prepare this file for placement in DATA COSINPUT, remove all records from the top of the file to immediately preceding 'MIS'; remove all lines below containing the word 'TOTAL', and then remove the 'COAL PRODUCTION' and 'BLOCK TONS' header records. The GO symbol 'COL' and the coal production figures remain, but the 'TOTL' record and all that follow are removed. Finally, data defining the coal tonnage not incurring the contract hauler charge (COST(21)) and block specific coal prices (if appropriate) are manually entered.

Data keyed to 'VOL', 'RCL', and 'RED' are taken directly from output file COSINPUT MOVE. If no priority 3 or 5 data are required to run COST, no editing of
COSINPUT MOVE is required and file assembly is completed. Otherwise, the data keyed to 'SPB' and/or 'RED' should be entered manually, as indicated in Table 7.

Extensive use of the programs may require that minor modifications be made, if the programs are to fit the user's site-specific data requirements. Currently, COSTSUM is set to handle 22 location symbols. Expansion of this number requires an alteration of the array declaration statements of each program, upgrading the current "22" as appropriate. The statements assigning initial value to variable BDIM must also be altered so as to conform to the enlarged arrays. Expansion of the number of lifts from the current 5 cannot be accomplished easily due to the output formats.

If the user wishes to expand the number of possible dozer and loader operations, similar modifications are required of programs LOADER and DOZER. The dimensions of arrays OP and OPHRS and the initial assignments of value to variables LDIM (in LOADER) and DZDIM (in DOZER) will need to be altered. In addition, program COST requires modification if those operations are to be charged to the appropriate cost categories; the required changes can be made by adding to the blocks of statements (3015 + and 4015 +) that assign a value to integer variable CC. No modification of subroutines is required.
If the program's capability to handle OMAC machines is not sufficient, similar changes are again possible. Program OTHER can handle up to 14 separate OMAC machines in its present configuration. To increase this number, arrays MACHIN, MACHT, and MACHRS require expansion, as does the initial assignment of IDIM. Program COST will also require modification. Arrays COITEM and COST require expansion, in order to enable the per unit cost of operation of the additional OMAC item to be entered. The initial assignment to variable CDIM will need to be enlarged if these additional cost items are to be processed. A program statement which assigns a 20-character identifier to the additional COITEM elements will allow identification of the cost item in the SUMMARY output file. Finally, the series of conditional assignment statements following 6015 will require modification if the program is to assign the proper cost when it encounters the additional OMAC item.

If the cost categories used here are to be changed or expanded, only program COST requires user attention. First, the COCAT assignment statement block should be modified as appropriate; the dimension of array COCAT and assignment of variable CCDIM may require expansion; the appropriate dimensions of arrays LIKOST, BLKOST, and CCCOST also need to be enlarged. Finally, the conditional assignment statements for the integer
variable CC will have to be altered, in order to assign the appropriate costs to the new cost category. These statements are found throughout the program, but they are always associated with the block of statements following the input READ or in the 9700 block. Changes in the cost categories may affect subsequent calculations. Statements 8930 + calculate the total load and haul cost (category 8) by adding the totals in cost categories 4, 5, 6, and 7 (statements 8930 +). Also, categories 9, 10, and 11 deal with coal-handling costs and are recalculated on per-ton (rather than per-block-bcy) bases by statements 9500 +. The per-bcy-handled calculations for costs in categories 3 and 8 are performed by statements following 8950. Calculations specific to categories 12 and 15 (reclamation) and 14 (overhead) have been detailed above.
MAKING USE OF THE OUTPUT

In order to make use of the data output of COSTSUM, a person must have a strong interest in a particular mining operation. Such persons may include mining engineers, regulatory agency personnel, and researchers.

The output data from COSTSUM can help mining engineers to know exactly what their costs are, on a block by block basis (2), and thus what they might be in future situations. Although every mining block is unique, there are certain situations which tend to repeat themselves as the mining operation moves through a particular landscape. By referring to data describing costs incurred during recently past mining activities, engineers and operators will be better able to assess the likely profitability of future mining. Thus, they will be better able to make decisions regarding the advisability of taking additional blocks and to develop more cost effective spoil-handling strategies. Also, accurate cost data on present operations will facilitate preparation of estimates of the costs of mining future sites.

Computer models to estimate mining costs are becoming increasingly available; some have detailed
data requirements (7). Accurate data on the machinery operation times and costs required by existing mining operations can facilitate using such models to prepare accurate mining-cost estimates.

The Surface Mining Control and Reclamation Act of 1977 requires that mining firms post performance bonds in amounts sufficient to cover the cost of reclamation. Maintaining accurate mining-cost data can help firms to prepare accurate reclamation cost estimates, thus eliminating costs associated with excess bond amounts. An accurate record of cost data can help the firm to justify these reclamation cost estimates in case of regulatory inquiries. Similarly, if the regulatory agency can arrange to monitor the costs of representative mining operations within its service district, that agency will have an empirical basis for judging the reclamation cost estimates provided by firms mining in similar conditions.

Within the mining industry and regulatory agencies, there is much concern regarding the costs of the environmental protection standards imposed by the Surface Mining Control and Reclamation Act of 1977. Accurate documentation of costs can assist those who wish to research the cost effects of the Act, and the regulations implementing the Act, upon current surface mining operations. For example, the costs specific to certain provisions of the Act can be interpreted in more meaningful fashion in the context of an accurate
record of the overall costs of mining. In addition, an accurate cost record, in conjunction with mining cost simulation models, can aid the preparation of estimates of cost comparisons between presently legislated spoil-handling and reclamation practices and proposed alternatives. Such research can assist those who wish to develop more cost effective environmental protection laws and regulations for the surface mining industry.
CONCLUDING REMARKS

We have been collecting data from active surface mine sites since July of 1983 (1,2). Since developing COSTSUM, our ability to process that data reliably and efficiently has increased immeasurably. This ability allows us to concentrate attention where it belongs: at the site. Despite the sophistication of any data-processing system, accurate data gathering at the site and accurate interpretation of site-gathered data remain essential to any study of the costs associated with coal surface mining operations.

The programs which constitute COSTSUM can be obtained free of charge by sending an IBM PC compatible double-sided, double-density diskette and a self-addressed envelope to Dr. W. L. Daniels, Dept. of Agronomy, Virginia Tech, Blacksburg, 24061 (703-961-7175).
References


APPENDIX A: HAUER

<table>
<thead>
<tr>
<th>DAY 2</th>
<th>MB1</th>
<th>1</th>
<th>SD1</th>
<th>14.00</th>
<th>100</th>
<th>500</th>
<th>SO SHOT OVERBURDEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAY 3</td>
<td>MB1</td>
<td>1</td>
<td>SD1</td>
<td>14.00</td>
<td>100</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>DAY 4</td>
<td>MB1</td>
<td>1</td>
<td>SD1</td>
<td>14.00</td>
<td>100</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>DAY 5</td>
<td>MB2</td>
<td>1</td>
<td>SD2</td>
<td>10.00</td>
<td>80</td>
<td>300</td>
<td>2 HRS. CHANGE OIL</td>
</tr>
<tr>
<td>DAY 6</td>
<td>MB1</td>
<td>2</td>
<td>SD1</td>
<td>14.00</td>
<td>120</td>
<td>400</td>
<td></td>
</tr>
<tr>
<td>DAY 7</td>
<td>MB1</td>
<td>2</td>
<td>SD1</td>
<td>14.00</td>
<td>130</td>
<td>400</td>
<td></td>
</tr>
<tr>
<td>DAY 8</td>
<td>MB1</td>
<td>2</td>
<td>SD1</td>
<td>14.00</td>
<td>15</td>
<td>500</td>
<td>FINISH BLOCK</td>
</tr>
<tr>
<td>DAY 9</td>
<td>MB2</td>
<td>2</td>
<td>SD2</td>
<td>14.00</td>
<td>120</td>
<td>400</td>
<td></td>
</tr>
<tr>
<td>DAY 10</td>
<td>MB3</td>
<td>1</td>
<td>SD1</td>
<td>14.00</td>
<td>120</td>
<td>400</td>
<td></td>
</tr>
</tbody>
</table>

**HAULER FORTRAN**

<table>
<thead>
<tr>
<th>C O S T S U M</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAMMED BY C.E. ZIPPER, 1985</td>
</tr>
<tr>
<td>DOCUMENTATION: VIRGINIA AGR. EXP. STATION BULLETIN NO. 86-1</td>
</tr>
<tr>
<td>VPI &amp; SU, BLACKSBURG 24061</td>
</tr>
</tbody>
</table>

PROGRAM HAULER

**THIS PROGRAM TOTALS HAUER DATA (HOURS AND LOADS) BY BLOCKS**

**AND LIFTS OF ORIGIN AND BY BLOCKS OF DESTINATION**

**THE FOLLOWING ARRAYS ARE DEFINED TO ACCUMULATE HOUR AND LIFT TOTALS, "HRS" SUFFIX INDICATES HOUR ACCUMULATORS, WHILE "LDS" INDICATES LOADS. PREFIXES INDICATE THE FOLLOWING:**

- "XXX(I,J,K) FROM BLOCK I LIFT J TO BLOCK K"
- "BBXXX(I,K) FROM BLOCK I TO BLOCK K"
- "BLXXX(I,J) FROM BLOCK I, LIFT J"
- "FXXX(I) FROM BLOCK I"
- "TXXX(K) TO BLOCK K"

**ANY HOURS ARRAY NAME PRECEDED BY 'L' ACCUMULATES ONLY IF LOADS>0**

**THUS, LOADS PER HOUR CALCULATIONS DO NOT INCLUDE 0 LOAD ENTRIES, AND ARE NOT DISTORTED BY MISSING DATA**

**PROGRAM HAULER**

**THIS PROGRAM TOTALS HAUER DATA (HOURS AND LOADS) BY BLOCKS**

**AND LIFTS OF ORIGIN AND BY BLOCKS OF DESTINATION**

**THE FOLLOWING ARRAYS ARE DEFINED TO ACCUMULATE HOUR AND LIFT TOTALS, "HRS" SUFFIX INDICATES HOUR ACCUMULATORS, WHILE "LDS" INDICATES LOADS. PREFIXES INDICATE THE FOLLOWING:**

- "XXX(I,J,K) FROM BLOCK I LIFT J TO BLOCK K"
- "BBXXX(I,K) FROM BLOCK I TO BLOCK K"
- "BLXXX(I,J) FROM BLOCK I, LIFT J"
- "FXXX(I) FROM BLOCK I"
- "TXXX(K) TO BLOCK K"

**ANY HOURS ARRAY NAME PRECEDED BY 'L' ACCUMULATES ONLY IF LOADS>0**

**THUS, LOADS PER HOUR CALCULATIONS DO NOT INCLUDE 0 LOAD ENTRIES, AND ARE NOT DISTORTED BY MISSING DATA**

**PROGRAM HAULER**

**THIS PROGRAM TOTALS HAUER DATA (HOURS AND LOADS) BY BLOCKS**

**AND LIFTS OF ORIGIN AND BY BLOCKS OF DESTINATION**

**THE FOLLOWING ARRAYS ARE DEFINED TO ACCUMULATE HOUR AND LIFT TOTALS, "HRS" SUFFIX INDICATES HOUR ACCUMULATORS, WHILE "LDS" INDICATES LOADS. PREFIXES INDICATE THE FOLLOWING:**

- "XXX(I,J,K) FROM BLOCK I LIFT J TO BLOCK K"
- "BBXXX(I,K) FROM BLOCK I TO BLOCK K"
- "BLXXX(I,J) FROM BLOCK I, LIFT J"
- "FXXX(I) FROM BLOCK I"
- "TXXX(K) TO BLOCK K"

**ANY HOURS ARRAY NAME PRECEDED BY 'L' ACCUMULATES ONLY IF LOADS>0**

**THUS, LOADS PER HOUR CALCULATIONS DO NOT INCLUDE 0 LOAD ENTRIES, AND ARE NOT DISTORTED BY MISSING DATA**

**PROGRAM HAULER**

**THIS PROGRAM TOTALS HAUER DATA (HOURS AND LOADS) BY BLOCKS**

**AND LIFTS OF ORIGIN AND BY BLOCKS OF DESTINATION**

**THE FOLLOWING ARRAYS ARE DEFINED TO ACCUMULATE HOUR AND LIFT TOTALS, "HRS" SUFFIX INDICATES HOUR ACCUMULATORS, WHILE "LDS" INDICATES LOADS. PREFIXES INDICATE THE FOLLOWING:**

- "XXX(I,J,K) FROM BLOCK I LIFT J TO BLOCK K"
- "BBXXX(I,K) FROM BLOCK I TO BLOCK K"
- "BLXXX(I,J) FROM BLOCK I, LIFT J"
- "FXXX(I) FROM BLOCK I"
- "TXXX(K) TO BLOCK K"

**ANY HOURS ARRAY NAME PRECEDED BY 'L' ACCUMULATES ONLY IF LOADS>0**

**THUS, LOADS PER HOUR CALCULATIONS DO NOT INCLUDE 0 LOAD ENTRIES, AND ARE NOT DISTORTED BY MISSING DATA**
REAL BBPER(22,22), BLPER(22,5)
REAL FPER(22), TPER(22)

* BDIM IS THE NUMBER OF MINING AREAS, OR BLOCKS, ON THE JOB
* BDIM = 22
* INITIALIZE VARIABLES
TOLDS = 0
TOHRS = 0.0
LTOHRS = 0.0
ICOUNT = 0
BLANK = ' ',
R = 'R'
Q = ' '

* IMAX = ENDLESS LOOP PROTECTION, SET TO MAXIMUM ICOUNT
IMAX = 1000

* INITIALIZE ACCUMULATOR ARRAYS - CALL ABLOCK FOR "BLOCK" ARRAY
DO 40 I = 1, BDIM
FLDSC(I) = 0
TLDS(I) = 0
FHRS(I) = 0.0
THRS(I) = 0.0
LFHRS(I) = 0.0
LTHRS(I) = 0.0
40 CONTINUE
DO 60 J = 1, 5
DO 50 K = 1, BDIM
BLHRSCI(J,K) = 0.0
BLHRS(J,K) = 0.0
BLLDS(J,K) = 0
50 CONTINUE
60 CONTINUE
DO 80 K = 1, BDIM
DO 70 I = 1, BDIM
LDSCI(J,K) = 0
HRSCI(J,K) = 0.0
LLHRSCI(J,K) = 0.0
LHRSCI(J,K) = 0.0
RRRCI(J,K) = 0.0
70 CONTINUE
80 CONTINUE
90 CONTINUE

DO 95 I = 1, BDIM
DO 93 K = 1, BDIM
BBHRSCI(K) = 0.0
BBLDS(K) = 0
93 CONTINUE
95 CONTINUE

CALL ABLOCK (BLOCK,BDIM)

WRITE(06,*) 'BDIM = ', BDIM, ' ENTER READ BLOCK'

READ DATA
FILE 10 HAUER INDATA
FILE 31 HRSOURCE HAUER
FILE 32 LDSOURCE HAUER
FILE 33 LOADDEST HAUER
FILE 34 HOURDEST HAUER
FILE 35 LDSPERHR HAUER

1000 READ (10,1010,ERR=9910,END=9000) HLDATE, HLBLK, HLLIFT, HLDEST,
RECPCT, HLHRS, HLOADS, HLDIS, HLMATL, HLCOMM
1010 FORMAT (A5,3X,A4,1X,A1,4X,A4,1X,I3,3X,F5.2,3X,I3,1X,2X,I4, 1
        2X,A2,3X,A25)

* THIS BLOCK ONLY FOR PARTITION OF HAULER EXPENSE AMONG
   SOURCE AND DESTINATION; 'RRR' ARRAY ACCUMULATES
   HOURS TO BE REDISTRIBUTED, AS INDICATED BY 'RECPCT'
   INPUT VARIABLE; HOURS TO BE REDISTRIBUTED INDICATED
   IN HOURS-BY-SOURCE OUTPUT, BY 'R' AFTER DEST. BLOCK.

GIVE VALUES TO I,J,K
   I: SOURCE BLOCK
   J: LIFT
   K: DESTINATION BLOCK

1070 CALL KBLOCK (BLOCK,BDIM,HLBLK,I)
CALL JLIFT (HLBLIFT,J)
CALL KBLOCK (BLOCK,BDIM,HLDEST,K)

IF RECPCT > 0:
   THE HAULER HOURS TO BE CHARGED TO DESTINATION BLOCK ARE
   INDICATED BY 'R'; THE COST PROGRAM WILL CONSIDER
   'R' HOURS AS AN EXPENSE OF RECLAIMING THE DESTINATION
   'RRR' ARRAY ACCUMULATES 'R' HOURS

IF (RECPCT .NE. 0.0) THEN
   REC = HLHRS*RECPCT/100.0
   RRR(I,J,K) = RRR(I,J,K) + REC
   HRS(I,J,K) = HRS(I,J,K) + HLHRS - REC
ELSE
   HRS(I,J,K) = HRS(I,J,K) + HLHRS
ENDIF

COUNT AND TOTAL

ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9910
BBHRS(I,K) = BBHRS(I,K) + HLHRS
BLHRS(I,J) = BLHRS(I,J) + HLHRS
FRHRS(I) = FRHRS(I) + HLHRS
TFRS(K) = TFRS(K) + HLHRS
LDS(I,J,K) = LDS(I,J,K) + HLOADS
BBLDSC(I,K) = BBLDSC(I,K) + HLOADS
BLDSC(I,J) = BLDSC(I,J) + HLOADS
FBLDSC(I) = FBLDSC(I) + HLOADS
TBLDSC(K) = TBLDSC(K) + HLOADS
TLODS = TLODS + HLOADS
TOHRS = TOHRS + HLHRS

IF LOADS WERE COUNTED, ACCUMULATE L__HRS QUANTITIES

IF (HLOADS .GT. 0) THEN
   LTOHRS = LTOHRS + HLHRS
   LBHRS(I,K) = LBHRS(I,K) + HLHRS
   LBLHRS(I,J) = LBLHRS(I,J) + HLHRS
   LFHRSI = LFHRSI + HLHRS
   LTHRS(K) = LTHRS(K) + HLHRS
   LHRSI,J,K) = LHRSI,J,K) + HLHRS
ENDIF
GOTO 1000

9000 WRITE (06,*) 'COUNTING COMPLETE ENTERED WRITE BLOCK'

FORMAT STATEMENTS FOR WRITE BLOCKS

9010 FORMAT (6X,'FROM',3X,'TO',5(4X,'FROM'),13X,'TO')
9020 FORMAT(5X,'BLOCK',1X,'BLOCK',2X,'LIFT',2X,'LIFT',2X,'LIFT',2X,'LIFT', 1
        2X,'LIFT',2X,'LIFT',2X,'LIFT',5X,'TOTAL',5X,'BLOCK')
9030 FORMAT (5X,A4,2X,A4,3X,F6.1,2X,F7.1,2X,A4)
9031 FORMAT (5X,A4,2X,A4,1X,A1,1X,5(F6.1,2X))
HAULER HOURS BY ORIGIN TO FILE 31

9098 WRITE (31,*) 'HAULER HOURS BY ORIGIN'
WRITE (31,9010)
WRITE (31,9020)
WRITE (31,9099)

9099 FORMAT ('HLR')
9100 DO 9120 I = 1, BDIM
   DO 9110 K = 1, BDIM
      IF (BBHRS(I,K).EQ.0) GOTO 9110
      WRITE (31,9050) BLOCK(I), BLOCK(K), HRS(I,1,K), HRS(I,2,K),
      \ HRS(I,3,K), HRS(I,4,K), HRS(I,5,K), BBHRS(I,K), BLOCK(K)
   1    REC = 0.0
   DO 9105 J = 1, 5
   REC = REC + RRR(I,J,K)
   CONTINUE
   IF (REC .LT. 0.1) GOTO 9110
   WRITE (31,9040) BLOCK(I), BLANK, BLHRS(I,1), BLHRS(I,2), BLHRS(I,3),
   \ BLHRS(I,4), BLHRS(I,5), FHRSCI, BLANK
9110 CONTINUE

9120 CONTINUE
WRITE (31,9610) TOHRS
WRITE (31,*) 'LAST DATE PROCESSED = ', HLDATE
9175 CONTINUE
9176 CONTINUE

HAULER LOADS BY ORIGIN TO FILE 32

9200 DO 9220 I = 1, BDIM
   DO 9210 K = 1, BDIM
      IF (BBLDS(I,K).EQ.0) GOTO 9210
      WRITE (32,9040) BLOCK(I), BLOCK(K), LDS(I,1,K), LDS(I,2,K),
      \ LDS(I,3,K), LDS(I,4,K), LDS(I,5,K), BBLDS(I,K), BLOCK(K)
   1    CONTINUE
   IF (FLDSC(I).EQ.0) GOTO 9220
   WRITE (32,9060) BLOCK(I), BLANK, BLLDS(I,1), BLLDS(I,2), BLLDS(I,3),
   \ BLLDS(I,4), BLLDS(I,5), FLDSCI, BLANK
9220 CONTINUE
WRITE (32,*) 'TOTAL HAULER LOADS = ', TOLDS
WRITE (32,*) 'LAST DATE PROCESSED = ', HLDATE

HAULER LOADS BY DESTINATION TO FILE 33
WRITE (33,*) 'HAULER LOADS BY DESTINATION'
WRITE (33,*)
WRITE (33,9060)
WRITE (33,9020)
WRITE (33,*)
9300 DO 9320 K = 1, BDIM
DO 9310 I = 1, BDIM
IF (BBLDS(I,K) .EQ. 0) GOTO 9310
WRITE (33,9040) BLOCK(K), BLOCK(I), LDS(I,5,K), LDS(I,4,K),
LDS(I,3,K), LDS(I,2,K), LDS(I,1,K), BBLDS(I,K), BLOCK(I)
9310 CONTINUE
IF (TLDS(K) .EQ. 0) GOTO 9320
WRITE (33,9080) BLOCK(K), TLDS(K)
WRITE (33,*)
9320 CONTINUE
WRITE (33,*) 'TOTAL HAULER LOADS = ', TOLDS
WRITE (33,*) 'LAST DATE PROCESSED = ', HLDATE
WRITE (33,*)
* HAULER HOURS BY DESTINATION TO FILE 34 *
WRITE (34,*) 'HAULER HOURS BY DESTINATION'
WRITE (34,*)
WRITE (34,9060)
WRITE (34,9020)
WRITE (34,*)
9400 DO 9420 K = 1, BDIM
DO 9410 I = 1, BDIM
IF (BBHRS(I,K) .LT. 0.1) GOTO 9410
WRITE (34,9030) BLOCK(K), BLOCK(I), HRS(I,1,K), HRS(I,2,K),
HRS(I,3,K), HRS(I,4,K), HRS(I,5,K), BBLHRS(I,K), BLOCK(I)
9410 CONTINUE
IF (THRS(K) .EQ. 0) GOTO 9420
WRITE (34,9070) BLOCK(K), THRS(K)
WRITE (34,*)
9420 CONTINUE
WRITE (34,9430) TOHRS
9430 FORMAT(5X,'TOTAL HAULER HOURS = ',1X,F7.1)
* LOADS PER HOUR CALCULATIONS *
WRITE (35,*) 'LOADS PER HOUR BY HAULER SOURCE'
WRITE (35,*)
WRITE (35,9060)
WRITE (35,9020)
WRITE (35,*)
9500 DO 9520 I = 1, BDIM
DO 9510 K = 1, BDIM
DO 9500 J = 1, 5
IF (LHRS(I,J,K) .LT. 0.1) THEN
PER(I,J,K) = 0.0
ELSE
PER(I,J,K) = LDS(I,J,K)/LHRS(I,J,K)
ENDIF
9510 CONTINUE
IF (LBBHRS(I,K) .LT. 0.1) THEN
BBPER(I,K) = 0.0
ELSE
BBPER(I,K) = BBLDS(I,K)/LBBHRS(I,K)
ENDIF
9520 CONTINUE
IF (LFHRS(I) .LT. 0.1) THEN
FFPER(I) = 0.0
ELSE
FFPER(I) = FLDS(I)/LFHRS(I)
ENDIF
IF (LTHRS(I) .LT. 0.1) THEN
TPER(I) = 0.0
ELSE
TPER(I) = TLDS(I)/LTHRS(I)
ENDIF
CONTINUE
DO 9550 I = 1, BDMIN
DO 9540 J = 1, 5
IF (LBLHRS(I, J) .LT. 0.1) THEN
  BLPER(I, J) = 0.0
ELSE
  BLPER(I, J) = BLLDIS(I, J)/LBLHRS(I, J)
ENDIF
CONTINUE
CONTINUE
WRITEC35, 9020) 1-IRITEC35, 9010) 1-IRITEC35, 9050)
FORMAT (’HLP’)
DO 9620 I = 1, BDMIN
DO 9610 K = 1, BDMIN
IF (BBPER(I, K) .LT. 0.1) GOTO 9610
WRITE (35, 9032) BLOCK(I), BLOCK(K), PER(I, 1, K), PER(I, 2, K),
        PER(I, 3, K), PER(I, 4, K), PER(I, 5, K), BBPER(I, K), BLOCK(K)
CONTINUE
IF (FPER(I) .LT. 0.1) GOTO 9620
WRITE (35, 9032) BLOCK(I), BLANK, BLPER(I, 1), BLPER(I, 2), BLPER(I, 3),
        BLPER(I, 4), BLPER(I, 5), FPER(I), BLANK
WRITE (35, 9X)
CONTINUE
LOADS PER HOUR TO DESTINATION BLOCKS
WRITE (35, *)
WRITE (35, *) ’LOADS-PER-HOUR TO DESTINATION BLOCKS’
WRITE (35, *)
DO 9700 I = 1, BDMIN
IF (TPER(I) .LT. 0.1) GOTO 9700
WRITE (35, 9690) BLOCK(I), TPER(I)
CONTINUE
WRITE (35, *) ’TOTAL JOB AVERAGE’
WRITE (35, *)
WRITE (35, 9710) TOLDS
FORMAT (5X, ’TOTAL LOADS = ’, I8)
WRITE (35, 9720) TLOHRS
FORMAT (5X, ’TOTAL HOURS = ’, F8.1)
TOPER = TOLDS/TLOHRS
WRITE (35, 9730) TOPER
FORMAT (5X, ’TOTAL LOADS PER HOUR = ’, F8.2)
WRITE (35, *)
FINALE
WRITE (35, *) ’LAST DATE PROCESSED = ’, HLDATE
WRITE (35, *) ’FINISHED; TOTAL RECORDS PROCESSED = ’, ICOUNT
GOTO 9999
WRITE (06, *) ’READ ERROR; ICOUNT = ’, ICOUNT
STOP
END

HAULER HOURS BY ORIGIN

FROM TO FROM FROM FROM FROM FROM TO
BLOCK BLOCK 1 LIFT 3 LIFT 2 LIFT X LIFT R LIFT TOTAL BLOCK
### HLR

<table>
<thead>
<tr>
<th>HLR</th>
<th>SD1</th>
<th>SD2</th>
<th>SD3</th>
<th>SD4</th>
<th>SD5</th>
<th>SD6</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1</td>
<td>42.0</td>
<td>10.0</td>
<td>4.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB2</td>
<td>32.0</td>
<td>14.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB6</td>
<td>74.0</td>
<td>24.0</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
<td>11.0</td>
</tr>
</tbody>
</table>

**TOTAL HAULER HOURS =** 109.0

### HAULER HOURS BY DESTINATION

<table>
<thead>
<tr>
<th>TO BLOCK</th>
<th>FROM BLOCK</th>
<th>T LIFT</th>
<th>FROM 3 LIFT</th>
<th>FROM 2 LIFT</th>
<th>FROM X LIFT</th>
<th>FROM R LIFT</th>
<th>TOTAL</th>
<th>FROM BLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD1</td>
<td>MB1</td>
<td>42.0</td>
<td>32.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>74.0</td>
<td>MB1</td>
</tr>
<tr>
<td>SD1</td>
<td>MB3</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>6.0</td>
<td>MB3</td>
</tr>
<tr>
<td>SD2</td>
<td>MB2</td>
<td>10.0</td>
<td>14.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>24.0</td>
<td>MB2</td>
</tr>
<tr>
<td>SD2</td>
<td>MB3</td>
<td>5.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
<td>MB3</td>
</tr>
</tbody>
</table>

### LOADS PER HOUR BY HAULER SOURCE

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>TO BLOCK</th>
<th>T LIFT</th>
<th>FROM 3 LIFT</th>
<th>FROM 2 LIFT</th>
<th>FROM X LIFT</th>
<th>FROM R LIFT</th>
<th>TOTAL</th>
<th>BLOCK TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLP</td>
<td>MB1</td>
<td>7.14</td>
<td>8.28</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.64</td>
<td>SD1</td>
</tr>
<tr>
<td>HLP</td>
<td>MB1</td>
<td>7.14</td>
<td>8.28</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.64</td>
<td>SD1</td>
</tr>
<tr>
<td>HLP</td>
<td>MB2</td>
<td>8.00</td>
<td>8.21</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>8.13</td>
<td>SD2</td>
</tr>
<tr>
<td>HLP</td>
<td>MB2</td>
<td>8.00</td>
<td>8.21</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>8.13</td>
<td>SD2</td>
</tr>
<tr>
<td>HLP</td>
<td>MB3</td>
<td>5.00</td>
<td>0.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.00</td>
<td>SD1</td>
</tr>
<tr>
<td>HLP</td>
<td>MB3</td>
<td>5.60</td>
<td>0.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.60</td>
<td>SD2</td>
</tr>
<tr>
<td>HLP</td>
<td>MB3</td>
<td>5.27</td>
<td>0.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.27</td>
<td>SD2</td>
</tr>
</tbody>
</table>

### LOADS-PER-HOUR TO DESTINATION BLOCKS

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>SD1</th>
<th>SD2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.44</td>
<td>7.69</td>
</tr>
</tbody>
</table>

**TOTAL JOB AVERAGE**

- **TOTAL LOADS =** 818
- **TOTAL HOURS =** 109.0
- **TOTAL LOADS PER HOUR =** 7.50

**LAST DATE PROCESSED =** D 10
APPENDIX B: LOADER

* * * * LOADER INDATA * * * *

DAY 2 MB1 1 L2 7.0
DAY 3 MB1 1 L2 7.0
DAY 4 MB1 1 L2 7.0
DAY 5 MB2 1 L2 5.0
DAY 5 MB1 2 CD 2.0 100 SD1 CARRY TO NEARBY AREA
DAY 6 MB1 2 L2 7.
DAY 7 MB1 2 L2 7.
DAY 8 MB1 2 L2 2.
DAY 8 MB2 2 L2 5.
DAY 9 MB2 2 L1 4.
DAY 9 MB3 T L2 3. TS SLOW LOADING BOTTOM
DAY 9 MB1 X LC 5. C
DAY 9 MB1 X PA 2. C
D 10 MB3 T L2 2.5 TS
D 10 MB2 X LC 7. C

* * * * LOADER FORTRAN * * * *

* * * COSTSUM * * *

DOCUMENTATION: VIRGINIA AGR. EXP. STATION BULLETIN NO. 86-1
VPI & SU, BLACKSBURG 24061

PROGRAM LOADER

THIS PROGRAM WILL TOTAL LOADER HOURS BY OPERATION AND LOCATION
IT ALSO TOTAL CARRY - DUMP HOURS BY DESTINATION AND SOURCE,
AND HOURS WORKING DUMPSITE BY DESTINATION. DATA FROM THREE
LOADER MODELS CAN BE HANDLED BY THE PROGRAM.

DECLARE VARIABLE TYPES

CHARACTER*5 LDDATE
CHARACTER*4 LDBLK, BLK(22)
CHARACTER*1 LDLIFT, LIFT(5)
CHARACTER*2 LDOP1, OPER
CHARACTER*2 LDOP2
REAL LDRHS, HRS, OPHRS(3,15,22,5)
REAL AHRHS, BHRHS, CHRS
REAL CDHRHS(3,22,5,22), BCDHRHS(3,22), SCDHRHS, TCDHRHS
REAL WBAHRS, WBHRHS, MBHRHS(3,22)
REAL WBAHRS, WBHRHS, WBCHRHS
REAL TOTHRS
REAL LHRST, LHRST2, LHRST3, LHRSTX, LHRSC
REAL BHRSOB, BLKHRHS(3,22)
CHARACTER*1 LDMACH
INTEGER LDCDIS
CHARACTER*4 LDEST
CHARACTER*2 LDMLA
CHARACTER*25 LDGRN
CHARACTER*2 OP(15)
INTEGER ONEOP, FIRST
INTEGER IMAX, ICOUNT
INTEGER RCOUNT, LDIM, BDIM, I, J, K, L, LL, D, BCD, KK

ARRAY SUBSCRIPTS

I: MACHINE TYPE
J: LOADER OPERATION
K: SOURCE BLOCK

66
L: SOURCE LIFT
D: DESTINATION BLOCK

INITIALIZE VARIABLES

BDIM = 22
LDIM = 15
LL = 5
IMAX = 1000
AHRS = 0.0
BHRS = 0.0
CHR5 = 0.0
TOTHRS = 0.0
CPHRS = 0.0
TCDHRS = 0.0
WBRHRS = 0.0
WBHRS = 0.0
WBCHRS = 0.0

CALL ABLOCK (BLK,BDIM)
CALL ALDOP (OP,LDIM)
CALL ALIFT (LIFT,LL)

DO 60 I = 1,3
  DO 50 J = 1,LDIM
    DO 40 K = 1,BDIM
      DO 30 L = 1,5
        OPHRS(I,J,K,L) = 0.0
  10 CONTINUE
  20 CONTINUE
  30 CONTINUE
  40 CONTINUE
  50 CONTINUE

DO 140 I = 1,3
  DO 130 D = 1,BDIM
    DO 120 K = 1,BDIM
      CDHRS(I,K,L,D) = 0.0
  110 CONTINUE
  120 CONTINUE
  130 CONTINUE
  140 CONTINUE

RCOUNT = 0

READ DATA AND COUNT

FILE 11 LOADER INDATA
FILE 45 OPHOURS LOADER
FILE 46 DSHOURS LOADER
FILE 47 CDHOURS LOADER

1000 READ (11,1011,ERR=9900,END=7000)
  1 LDDATE,LDBLK,LDLIFT,LDOP1,LDOP2,LDHRS,
  2 LDMACH,LDCDIS,LDDEST,LDMATL,LDCOMM
1011 FORMAT (A5,5X,A4,1X,A1,5X,A2,1X,A2,4X,F5.2,A1,5X,I3,1X,
  1 A4,2X,A2,3X,A25)
  RCOUNT = RCOUNT + 1
  TOTHRS = TOTHRS + LDHRS

ASSIGN VALUE TO I TO INDICATE MACHINE TYPE

I = 4
IF (LDMACH .EQ. ' ') I = 1
IF (LDMACH .EQ. 'B') I = 2
IF (LDMACH .EQ. 'C') I = 3
IF (I .EQ. 4) THEN
  WRITE (45,*) 'WRONG CHARACTER LDMAC ', LDDATE ENDIF

* IF DUAL OPERATIONS PERFORMED:
  - LDHRS SPLIT BETWEEN LDOP1 AND LDOP2
  - ONEOP AND FIRST ACT AS LOGICAL VARIABLES
  1 = .TRUE  0 = .FALSE.
  - PROGRAM ROUTED THROUGH STATEMENT 2000 TWICE
    VIA LOGICAL VARIABLE ACTIVITY STATEMENTS 4900+

1500 IF (LDOP2 .NE. ' ') THEN
  ONEOP = 0
  FIRST = 1
  OPER = LDOP1
  HRS = LDHRS/2.0
ELSE
  ONEOP = 1
  HRS = LDHRS
  OPER = LDOP1
ENDIF

* ASSIGN VALUE TO J TO INDICATE OPERATION TYPE
* ASSIGN VALUE TO K TO INDICATE SOURCE BLOCK
* ASSIGN VALUE TO L TO INDICATE SOURCE LIFT

2000 CALL XLDOP(OP, LDIM, OPER, J)
CALL KBLOCK(BLK, BDIM, LDBLK, K)
CALL JLIFT(LDILT, L)
  IF (LDBLK .EQ. ' ') WRITE(45,*) LDDATE, 'CHECK SOURCE'
  IF (LDOP1 .EQ. ' ') WRITE(45,*) LDDATE, 'CHECK OPERATION'

* CHECK FOR ENDLESS LOOP

ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9900

* TOTAL HOURS BY MACHINE AND OPERATION

3000 IF (I .EQ. 1) AHRS = AHRS + HRS
IF (I .EQ. 2) BHRS = BHRS + HRS
IF (I .EQ. 3) CHRS = CHRS + HRS
OPHRS (I,J,K,L) = OPHRS(I,J,K,L) + HRS
BLKHRGS(I,K) = BLKHRGS(I,K) + HRS

* IF OPERATION = CD ASSIGN D TO CARRY - DUMP DESTINATION
* TOTAL CD HOURS BY DESTINATION

4000 IF (OPER .EQ. 'CD') THEN
  CALL KBLOCK (BLK, BDIM, LDDEST, D)
  IF (LDDEST .EQ. ' ') WRITE(47,*) LDDATE, 'CHECK DESTINATION'
  CDHRS(I,K,L,D) = CDHRS(I,K,L,D) + HRS
  BCDHRS(I,K) = BCDHRS(I,K) + HRS
ENDIF

* IF OPERATION = WB OR DS
  ASSIGN D TO IDENTIFY DESTINATION
  TOTAL WB/DS HOURS
  (WB = WORK BACKFILL, SAME MEANING AS DS)

4200 IF (OPER .EQ. 'WB' .OR. OPER .EQ. 'DS') THEN
  CALL KBLOCK (BLK, BDIM, LDDEST, KK)
  IF (LDDEST .EQ. ' ') WRITE(46,*) LDDATE, 'CHECK DESTINATION'
  WBHRS(I, KK) = WBHRS(I, KK) + HRS
  WBTHRS = WBTHRS + HRS
ENDIF
IF DUAL OPERATIONS FIRST TIME THROUGH
RETURN TO STATEMENT 2000 TO TOTAL LDOP2

4900 IF (ONEOP .EQ. 1) GOTO 6000
IF (FIRST .EQ. 1) THEN
FIRST = 0
OPER = LDOP2
5000 GOTO 2000
ENDIF

6000 GOTO 1000

INPUTS FINISHED WRITE TOTALS

TOTAL LOAD HOURS BY OPERATION FILE 45

7000 WRITE (45,*) 'B LOADER OPERATION HOURS TOTALS BY BLOCK AND LIFT'
WRITE (45,7020)
1 2X, 'X LIFT', 2X, 'R LIFT', 2X, 'TOTAL')
WRITE (45,*)
WRITE (45,7021)
7021 FORMAT (*LDB*)
DO 7070 K = 1, BDIM
LHRST = 0.0
LHRS2 = 0.0
LHRS3 = 0.0
LHRSX = 0.0
LHRSC = 0.0
DO 7060 J = 1, LDIM
BHRSON = OPHRS(2, J, K, 1) + OPHRS(2, J, K, 2) + OPHRS(2, J, K, 3)
+ OPHRS(2, J, K, 4) + OPHRS(2, J, K, 5)
1 IF (BHRSON .LT. 0.1) GOTO 7060
WRITE (45,7050) BLK(K), OP(J), OPHRS(2, J, K, 1), OPHRS(2, J, K, 2),
OPHRS(2, J, K, 3), OPHRS(2, J, K, 4), OPHRS(2, J, K, 5), BHRSON
7050 FORMAT (5X, A4, 6X, A2, 6X, F7.1)
LHRST = LHRST + OPHRS(2, J, K, 1)
LHRS2 = LHRS2 + OPHRS(2, J, K, 2)
LHRS3 = LHRS3 + OPHRS(2, J, K, 3)
LHRSX = LHRSX + OPHRS(2, J, K, 4)
LHRSC = LHRSC + OPHRS(2, J, K, 5)
7060 CONTINUE
IF (BLKHRS(2, K) .LT. 0.1) GOTO 7070
WRITE (45,7065) BLK(K), LHRST, LHRS2, LHRS3, LHRSX, LHRSC, BLKHRS(2, K)
7065 FORMAT (5X, A4, 6X, 6X, F7.1)
WRITE (45,*)
7070 CONTINUE
WRITE (45,*) 'LAST DATE PROCESSED = ', LDDATE
GOTO 8000

8000 WRITE (45,*)
WRITE (45,*) 'A LOADER OPERATION HOURS TOTALS'
WRITE (45,*) 'BLOCKS AND LIFTS ARE MATERIAL SOURCES'
WRITE (45,*)
WRITE (45,7020)
WRITE (45,*)
WRITE (45,8021)
8021 FORMAT (*LDA*)
DO 8070 K = 1, BDIM
LHRST = 0.0
LHRS2 = 0.0
LHRS3 = 0.0
LHRSX = 0.0
LHRSC = 0.0
DO 8060 J = 1, LDIM
  BHRSON = OPHRS(1, J, K, 1) + OPHRS(1, J, K, 2) + OPHRS(1, J, K, 3) + OPHRS(1, J, K, 4) + OPHRS(1, J, K, 5)
  IF (BHRSON .LT. 0.1) GOTO 8060
  WRITE (45, 7050) BLK(K), OP(J), OPHRS(1, J, K, 1), OPHRS(1, J, K, 2), OPHRS(1, J, K, 3), OPHRS(1, J, K, 4), OPHRS(1, J, K, 5), BHRSON
  LHRST = LHRST + OPHRS(1, J, K, 1)
  LHRST2 = LHRST2 + OPHRS(1, J, K, 2)
  LHRST3 = LHRST3 + OPHRS(1, J, K, 3)
  LHRST4 = LHRST4 + OPHRS(1, J, K, 4)
  LHRST5 = LHRST5 + OPHRS(1, J, K, 5)
8060 CONTINUE
IF (BLKRSON(1, K) .LT. 0.1) GOTO 8070
WRITE (45, 7065) BLK(K), LHRST, LHRST2, LHRST3, LHRST4, LHRST5, BLKRSON(1, K)
WRITE (45, *)
8070 CONTINUE
WRITE (45, *) ' LAST DATE PROCESSED = ', LDDATE
GOTO 9000

9000 WRITE (45, *)
WRITE (45, *) 'COAL LOADER OPERATION HOURS TOTALS BY BLOCK AND LIFT'
WRITE (45, 7020)
WRITE (45, *)
WRITE (45, 9021)
9021 FORMAT ('LDL')
DO 9070 K = 1, BDIM
  LHRST = 0.0
  LHRST2 = 0.0
  LHRST3 = 0.0
  LHRST4 = 0.0
  LHRST5 = 0.0
  DO 9060 J = 1, LDIM
    BHRSON = OPHRS(3, J, K, 1) + OPHRS(3, J, K, 2) + OPHRS(3, J, K, 3) + OPHRS(3, J, K, 4) + OPHRS(3, J, K, 5)
    IF (BHRSON .LT. 0.1) GOTO 9060
    WRITE (45, 7050) BLK(K), OP(J), OPHRS(3, J, K, 1), OPHRS(3, J, K, 2), OPHRS(3, J, K, 3), OPHRS(3, J, K, 4), OPHRS(3, J, K, 5), BHRSON
    LHRST = LHRST + OPHRS(3, J, K, 1)
    LHRST2 = LHRST2 + OPHRS(3, J, K, 2)
    LHRST3 = LHRST3 + OPHRS(3, J, K, 3)
    LHRST4 = LHRST4 + OPHRS(3, J, K, 4)
    LHRST5 = LHRST5 + OPHRS(3, J, K, 5)
9060 CONTINUE
IF (BLKRSON(3, K) .LT. 0.1) GOTO 9070
WRITE (45, 7065) BLK(K), LHRST, LHRST2, LHRST3, LHRST4, LHRST5, BLKRSON(3, K)
WRITE (45, *)
9070 CONTINUE
WRITE (45, *)
WRITE (45, 9021)
WRITE (45, *) ' LAST DATE PROCESSED = ', LDDATE
WRITE (45, *) 'NUMBER OF RECORDS READ = ', RCOUNT
WRITE (46, 9347)
9347 FORMAT ('WBD LDD BLK', 6X, 'TO', 8X, 'L O A D E R S ')
WRITE (46, 9350)
9350 FORMAT ('LDBL', 1X, 'LDDL')
LDBL = '1'
LDDL = '1'
WRITE (46, *)
WRITE (46, *) 'LOADER WORK DUMPSITE HOUR TOTALS BY DESTINATION'
WRITE (46, 9347)
DO 9370 KK = 1,BDIM
WBAHRS = WBAHRS + WBHRS(1,KK)
WBBHRS = WBBHRS + WBHRS(2,KK)
WBCHR = WBCHR + WBHRS(3,KK)
WBHRS = WBHRS(1,KK) + WBHRS(2,KK) + WBHRS(3,KK)
IF (WBHRS .LT. 0.1) GOTO 9370
WRITE(46,9360) LDBLK, LLDLFT, BLKCK, WBAHRS(1,KK), WBBHRS(2,KK),
WBHRS(3,KK), WBHRS, KK
9360 CONTINUE
WRITE (46,*)
WRITE (46,9380) LDBLK, LLDLFT, WBAHRS, WBBHRS, WBCHR, WBTHRS
9380 FORMAT (A4,1X,A1,3X,A4,4(2X,F6.1),10X,I2)
CONTINUE
WRITE (46,*) 'PROBLEM WITH WB HOURS TOTAL; WBTHRS ', WBTHRS
ENDIF
WRITE (46,*) 'NUMBER OF RECORDS READ = ', RCOUNT
WRITE (46,*) ' LAST DATE PROCESSED = ', LDDATE
WRITE CD HOURS TOTALS - FILE 47
A LOADERS
WRITE (47,*) ' A LOADER CD HOURS'
WRITE (47,9510)
9510 FORMAT (6X,'FROM',3X,'TO',5(4X,'FROM'))
WRITE (47,9520)
9520 FORMAT (5X,'BLOCK',1X,'BLOCK',2X,'T LIFT',2X,'2 LIFT',2X,'3 LIFT',
1 2X,'X LIFT',2X,'R LIFT',2X,'TOTAL')
WRITE (47,8021)
TCDHRS = 0.0
DO 9580 K = 1,BDIM
IF (BCDHR(1,K) .LT. 0.1) GOTO 9580
LHRST = 0.0
LHRST2 = 0.0
LHRST3 = 0.0
LHRSTX = 0.0
LHRSTC = 0.0
BCD = 0
DO 9550 D = 1,BDIM
SCDHR = SCDHR(1,K,1,D) + SCDHR(1,K,2,D) + SCDHR(1,K,3,D) +
CDHR(1,K,4,D) + CDHR(1,K,5,D)
IF (SCDHR .LT. 0.1) GOTO 9550
LHRST = LHRST + CDHR(1,K,1,D)
LHRST2 = LHRST2 + CDHR(1,K,2,D)
LHRST3 = LHRST3 + CDHR(1,K,3,D)
LHRSTX = LHRSTX + CDHR(1,K,4,D)
LHRSTC = LHRSTC + CDHR(1,K,5,D)
BCD = BCD + 1
WRITE (47,9530) BLK(K), BLK(D), CDHR(1,K,1,D), CDHR(1,K,2,D),
CDHR(1,K,3,D), CDHR(1,K,4,D), CDHR(1,K,5,D), SCDHR
9530 FORMAT (5X,A4,2X,A4,6(F6.1,2X))
9550 CONTINUE
TCDHR = TCDHR + BCDHR(1,K)
IF (BCD .LT. 2) GOTO 9570
WRITE (47,9560) BLK(K), LHRST, LHRST2, LHRST3, LHRSTX, LHRSTC, BCDHR(1,K)
9560 FORMAT (5X,A4,6X,6(F6.1,2X))
9570 WRITE (47,*)
9580 CONTINUE
WRITE (47,9590) TCDHR
9590 FORMAT (5X,'TOTAL A CARRY & DUMP HOURS = ',F6.1)
WRITE (47,*)
WRITE (47,*)
WRITE (47,*) ' LAST DATE PROCESSED = ', LDDATE
B LOADERS

WRITE(47,*)
WRITE(47,*) ' B LOADER CD HOURS'
WRITE(47,*) WRITE(47,9510)
WRITE(47,9520)
TCDHRS = 0.0
DO 9680 K = 1,BDIM
     IF (BCDHRSC2,K) .LT. 0.1) GOTO 9680
     LHRST = 0.0
     LHRS2 = 0.0
     LHRS3 = 0.0
     LHRSX = 0.0
     LHRSC = 0.0
     BCD = 0
     DO 9650 D = 1,BDIM
           SCDHRS = CDHRS(2,K,1,D) + CDHRS(2,K,2,D) + CDHRS(2,K,3,D) +
                    CDHRS(2,K,4,D) + CDHRS(2,K,5,D)
           IF (SCDHRS .LT. 0.1) GOTO 9650
           LHRST = LHRST + CDHRS(2,K,1,D)
           LHRS2 = LHRS2 + CDHRS(2,K,2,D)
           LHRS3 = LHRS3 + CDHRS(2,K,3,D)
           LHRSX = LHRSX + CDHRS(2,K,4,D)
           LHRSC = LHRSC + CDHRS(2,K,5,D)
           BCD = BCD + 1
           WRITE (47,9530) BLK(K),BLK(D),CDHRS(2,K,1,D),CDHRS(2,K,2,D),
                        CDHRS(2,K,3,D),CDHRS(2,K,4,D),CDHRS(2,K,5,D),SCDHRS
9650    CONTINUE
     TCDHRS = TCDHRS + BCDHRSC2,K
8670    WRITE(47,9690) TCDHRS
8690 FORMAT(5X,'TOTAL B CARRY & DUMP HOURS = ',F6.1)
WRITE(47,*)
WRITE(47,*)

COAL LOADER

WRITE(47,*)
WRITE(47,*) ' C LOADER CD HOURS'
WRITE(47,*) WRITE(47,9510)
WRITE(47,9520)
TCDHRS = 0.0
DO 9780 K = 1,BDIM
     IF (BCDHRSC3,K) .LT. 0.1) GOTO 9780
     LHRST = 0.0
     LHRS2 = 0.0
     LHRS3 = 0.0
     LHRSX = 0.0
     LHRSC = 0.0
     BCD = 0
     DO 9750 D = 1,BDIM
           SCDHRS = CDHRS(3,K,1,D) + CDHRS(3,K,2,D) + CDHRS(3,K,3,D) +
                    CDHRS(3,K,4,D) + CDHRS(3,K,5,D)
           IF (SCDHRS .LT. 0.1) GOTO 9750
           LHRST = LHRST + CDHRS(3,K,1,D)
           LHRS2 = LHRS2 + CDHRS(3,K,2,D)
           LHRS3 = LHRS3 + CDHRS(3,K,3,D)
           LHRSX = LHRSX + CDHRS(3,K,4,D)
           LHRSC = LHRSC + CDHRS(3,K,5,D)
           BCD = BCD + 1
           WRITE (47,9530) BLK(K),BLK(D),CDHRS(3,K,1,D),CDHRS(3,K,2,D),
                        CDHRS(3,K,3,D),CDHRS(3,K,4,D),CDHRS(3,K,5,D),SCDHRS
9750    CONTINUE
     TCDHRS = TCDHRS + BCDHRSC3,K
IF (BCD .LT. 2) GOTO 9770
WRITE(47,9560) BLK(K),LHRST,LHRS2,LHRS3,LHRSX,LHRSC,BCDHRSC(3,K)

9770 WRITE(47,*)
9780 WRITE(47,*) BLK(K),LHRST,LHRS2,LHRS3,LHRSX,LHRSC,BCDHRSC(3,K)

WRITE(47,9790) TCDHRS
9790 FORMAT(5X,'TOTAL C CARRY & DUMP HOURS = ',F6.1)
WRITE(47,*)
WRITE(47,*)

9900 WRITE (06,*) 'READ ERROR; LDDATE = ', LDDATE,' IMAX = ',J
GOTO 9999
9911 WRITE (06,*) 'READ ERROR LDDATE = ', LDDATE
9999 STOP
END

* * * OPHOURS LOADER * * *

A LOADER OPERATION HOURS TOTALS
BLOCKS AND LIFTS ARE MATERIAL SOURCES

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>OPER</th>
<th>T LIFT</th>
<th>2 LIFT</th>
<th>3 LIFT</th>
<th>X LIFT</th>
<th>R LIFT</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>MB1</td>
<td>L2</td>
<td>21.0</td>
<td>16.0</td>
<td>0.0</td>
<td>0.0</td>
<td>37.0</td>
</tr>
<tr>
<td></td>
<td>MB1</td>
<td>CD</td>
<td>0.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>MB1</td>
<td>21.0</td>
<td>18.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>39.0</td>
</tr>
<tr>
<td></td>
<td>MB2</td>
<td>L1</td>
<td>0.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td>MB2</td>
<td>L2</td>
<td>5.0</td>
<td>5.0</td>
<td>0.0</td>
<td>0.0</td>
<td>10.0</td>
</tr>
<tr>
<td></td>
<td>MB2</td>
<td>5.0</td>
<td>9.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>14.0</td>
</tr>
<tr>
<td></td>
<td>MB3</td>
<td>L2</td>
<td>5.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>MB3</td>
<td>5.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.5</td>
</tr>
</tbody>
</table>

COAL LOADER OPERATION HOURS TOTALS BY BLOCK AND LIFT

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>OPER</th>
<th>T LIFT</th>
<th>2 LIFT</th>
<th>3 LIFT</th>
<th>X LIFT</th>
<th>R LIFT</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDC</td>
<td>MB1</td>
<td>LC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>MB1</td>
<td>PA</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>MB1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
<td>0.0</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>MB2</td>
<td>LC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>MB2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
<td>0.0</td>
<td>7.0</td>
</tr>
</tbody>
</table>

* * * CDHOURS LOADER * * *

A LOADER CD HOURS

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>FROM</th>
<th>TO</th>
<th>FROM</th>
<th>FROM</th>
<th>FROM</th>
<th>FROM</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>SD1</td>
<td>0.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.0</td>
</tr>
</tbody>
</table>

TOTAL A CARRY & DUMP HOURS = 2.0
### DOZER INDATA

<table>
<thead>
<tr>
<th>DAY</th>
<th>MACHINE</th>
<th>LOCATION</th>
<th>WORKING HOURS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MB1</td>
<td>ST</td>
<td>1.0</td>
</tr>
<tr>
<td>1</td>
<td>MB2</td>
<td>PD</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>MB1</td>
<td>FO</td>
<td>5.0</td>
</tr>
<tr>
<td>2</td>
<td>MB1</td>
<td>DS</td>
<td>2.0</td>
</tr>
<tr>
<td>3</td>
<td>MB1</td>
<td>FO</td>
<td>5.0</td>
</tr>
<tr>
<td>3</td>
<td>MB1</td>
<td>DS</td>
<td>2.0</td>
</tr>
<tr>
<td>4</td>
<td>MB2</td>
<td>FO</td>
<td>7.0</td>
</tr>
<tr>
<td>5</td>
<td>MB2</td>
<td>FO</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>MB2</td>
<td>FO</td>
<td>5.0</td>
</tr>
<tr>
<td>5</td>
<td>MB1</td>
<td>DS</td>
<td>3.0</td>
</tr>
<tr>
<td>5</td>
<td>MB2</td>
<td>FO</td>
<td>2.0</td>
</tr>
<tr>
<td>6</td>
<td>MB1</td>
<td>FO</td>
<td>3.0</td>
</tr>
<tr>
<td>6</td>
<td>MB2</td>
<td>DS</td>
<td>4.0</td>
</tr>
<tr>
<td>7</td>
<td>MB1</td>
<td>FO</td>
<td>5.0</td>
</tr>
<tr>
<td>7</td>
<td>SD1</td>
<td>RR</td>
<td>5.0</td>
</tr>
<tr>
<td>8</td>
<td>SD1</td>
<td>RR</td>
<td>5.0</td>
</tr>
<tr>
<td>8</td>
<td>SD2</td>
<td>RR</td>
<td>5.0</td>
</tr>
<tr>
<td>9</td>
<td>MB3</td>
<td>ST</td>
<td>5.0</td>
</tr>
<tr>
<td>10</td>
<td>SD1</td>
<td>RR</td>
<td>4.0</td>
</tr>
<tr>
<td>10</td>
<td>SD2</td>
<td>RR</td>
<td>3.0</td>
</tr>
</tbody>
</table>

### DOZER FORTRAN

```fortran
PROGRAM DOZER

THIS PROGRAM TOTALS DOZER HOURS BY OPERATION AND LOCATION.
IT ALSO TOTALS PUSH OVERBURDEN HOURS BY SOURCE AND
DESTINATION, AND IT TOTALS WORK DUMPSITE HOURS.

DECLARE VARIABLE TYPES
```

**ARRAY SUBSCRIPTS**

- `I`: MACHINE TYPE
* J: LOADER OPERATION
* K: SOURCE BLOCK
* L: SOURCE LIFT
* D: DESTINATION BLOCK

INITIALIZE VARIABLES

BDIM = 22
DZDIM = 20
IMAX = 1000
LL = 5
AHRS = 0.0
CHRS = 0.0
TOTHRS = 0.0
CDHRS = 0.0
WBHRS = 0.0
WBCHRS = 0.0
WBTHRS = 0.0

CALL ABLOCK (BLK, BDIM)
CALL ADZOP (OP, DZDIM)
CALL ALIFT (LIFT, LL)

DO 40 I = 1, 2
   DO 30 J = 1, DZDIM
      DO 20 K = 1, BDIM
         OPHRS(I, J, K, L) = 0.0
      CONTINUE
   CONTINUE
40 CONTINUE

DO 60 K = 1, BDIM
   DO 50 I = 1, 2
      BLKHRSC(I, K) = 0.0
      BCDHRS(I, K) = 0.0
      WBHRSC(I, K) = 0.0
   CONTINUE
50 CONTINUE

DO 140 I = 1, 2
   DO 130 D = 1, BDIM
      DO 120 K = 1, BDIM
         CDHRS(I, K, L, D) = 0.0
      CONTINUE
   CONTINUE
140 CONTINUE

RCOUNT = 0

READ DATA AND COUNT

FILE 12 DOZER INDATA
FILE 55 OPHOURS DOZER
FILE 56 DSHOURS DOZER
FILE 57 PMHOURS DOZER

1000 READ (12, 1012, ERR=9911, END=7000)
   DZDATE, DZBLK, DZLIFT, DZOP1, DZOP2, DZHRS,
   DZMACH, DZDIST, DZDEST, DZMATL, DZCOMM
1012 FORMAT (A5, 3X, A4, 1X, A1, 5X, A2, 1X, A2, 4X, F5.2, A1, 5X, I3, 1X,
   A4, 2X, A2, 3X, A25)

RCOUNT = RCOUNT + 1
TOTHRS = TOTHRS + DZHRS

ASSIGN VALUE TO I TO INDICATE MACHINE TYPE

I = 4
IF (DZMACH .EQ. ' ') I = 1
IF (DZMACH .EQ. 'C')  I = 2
IF (I .EQ. 4) THEN
    WRITE (55,*) 'WRONG CHARACTER DZMAC ',DZDATE
ENDIF

IF DUAL OPERATIONS PERFORMED:
    - SPLIT DZHRS AMONG LDOP1 AND LDOP2
    - ONEOP AND FIRST ACT AS LOGICAL VARIABLES
    1 = .TRUE  0 = .FALSE.
    - PROGRAM GOES THROUGH STATEMENT 2000 TWICE VIA
      LOGICAL VARIABLE OPERATION IN STATEMENTS 6000 +

1500 IF (DZOP2 .NE. ' ') THEN
    ONEOP = 0
    FIRST = 1
    OPER = DZOP1
    HRS = DZHRS/2.0
ELSE
    ONEOP = 1
    HRS = DZHRS
    OPER = DZOP1
ENDIF

ASSIGN VALUE TO J TO INDICATE OPERATION TYPE
ASSIGN VALUE TO K TO INDICATE SOURCE BLOCK
ASSIGN VALUE TO L TO INDICATE SOURCE LIFT

2000 CALL XDZOP(OP,DZDIM,OPER,J)
CALL KBLOCK(BLK,BDIM,DZBLK,K)
CALL JLIFT(DZLIFT,L)

ENDLESS LOOP PROTECTION
ICOUNT = ICDOUNT + 1
IF (ICOUNT .GT. IMAX) GO TO 9911

TOTAL HOURS BY MACHINE AND OPERATION

3000 IF (I .EQ. 1) AHRS = AHRS + HRS
IF (I .EQ. 2) CHRS = CHRS + HRS
OPHRS (I,J,K,L) = OPHRS(I,J,K,L) + HRS
BLKHRSCI,K) = BLKHRSCI,K) + HRS

IF OPERATION = PO ASSIGN D TO PUSH OVERBURDEN DESTINATION
'CD' SYMBOLS STAND FOR PO' OPERATION VARIABLES

IF (OPER .EQ. 'PO') THEN
4000 CALL KBLOCK(BLK,BDIM,DZDEST,D)
IF (DZDEST .EQ. ' ') WRITE(57,*) DZDATE,'CHECK DESTINATION'
4100 CDHRS(I,K,L,D) = CDHRS(I,K,L,D) + HRS
BCDHRSCI,K) = BCDHRSCI,K) + HRS
ENDIF

IF OPER = WB OR DS
TOTAL WB HOURS BY DESTINATION
ASSIGN D TO IDENTIFY DESTINATION
(WB MEANS WORK BACKFILL, SAME MEANING AS DS)

4200 IF (OPER .EQ. 'WB' .OR. OPER .EQ. 'DS') THEN
CALL KBLOCK(BLK,BDIM,DZDEST,D)
IF (DZDEST .EQ. ' ') WRITE(56,*) DZDATE,'CHECK DESTINATION'
WBHRS(I,D) = WBHRS(I,D) + HRS
WBTHRS = WBTHRS + HRS
ENDIF

IF DUAL OPERATIONS FIRST TIME THROUGH
RETURN TO STATEMENT 2000 TO TOTAL DZOP2
4900 IF (ONEOP .EQ. 1) GOTO 6000
   IF (FIRST .EQ. 1) THEN
      FIRST = 0
      OPER = D2OP2
   GOTO 2000
ENDIF

6000 GOTO 1000
READ FINISHED WRITE TOTALS

TOTAL LOAD HOURS BY OPERATION FILE 55

7000 WRITE (55,*) ' C DOZER OPERATION HOURS TOTALS BY BLOCK AND LIFT'
WRITE (55,7020)
7020 FORMAT (5X,'BLOCK',2X,'OPER',2X,'T LIFT',2X,'2 LIFT',2X,'3 LIFT',
1 2X,'R LIFT',2X,'TOTAL')
WRITE (55,*)
7021 FORMAT ('DZC')
DO 7070 K = 1,BDIM
   LHRST = 0.0
   LHRS2 = 0.0
   LHRS3 = 0.0
   LHRSX = 0.0
   LHRSC = 0.0
   DO 7060 J = 1,DZDIM
      BHRST = OPHRS(2,J,K,1) + OPHRS(2,J,K,2) + OPHRS(2,J,K,3) + OPHRS(2,J,K,4) + OPHRS(2,J,K,5)
   IF (BHRST .LT. 0.1) GOTO 7060
   WRITE (55,7050) BLK(K),OP(J),OPHRS(2,J,K,1),OPHRS(2,J,K,2),
1 OPHRS(2,J,K,3),OPHRS(2,J,K,4),OPHRS(2,J,K,5),BHRST
7050 FORMAT (5X,A4,6X,A2,6X,F6.1))
WRITE (55,*)
7060 CONTINUE
   IF (BHRST .LT. 0.1) GOTO 7070
   WRITE (55,7065) BLK(K),LHRST,LHRS2,LHRS3,LHRSX,LHRSC,BHRST
7065 FORMAT (5X,A4,6X,F7.1))
WRITE (55,*)
7070 CONTINUE
GOTO 8000

8000 WRITE (55,*)
WRITE (55,*) ' A DOZERS OPERATION HOURS TOTALS BY BLOCK AND LIFT'
WRITE (55,7020)
WRITE (55,*)
WRITE (55,8020)
WRITE (55,*)
8021 FORMAT ('DZA')
DO 8070 K = 1,BDIM
   LHRST = 0.0
   LHRS2 = 0.0
   LHRS3 = 0.0
   LHRSX = 0.0
   LHRSC = 0.0
   DO 8060 J = 1,DZDIM
      BHRST = OPHRS(1,J,K,1) + OPHRS(1,J,K,2) + OPHRS(1,J,K,3) + OPHRS(1,J,K,4) + OPHRS(1,J,K,5)
   IF (BHRST .LT. 0.1) GOTO 8060
   WRITE (55,7050) BLK(K),OP(J),OPHRS(1,J,K,1),OPHRS(1,J,K,2),
1 OPHRS(1,J,K,3),OPHRS(1,J,K,4),OPHRS(1,J,K,5),BHRST
8050 FORMAT (5X,A4,6X,A2,6X,F6.1))
WRITE (55,*)
8060 CONTINUE
   IF (BHRST .LT. 0.1) GOTO 8070
   WRITE (55,8065) BLK(K),LHRST,LHRS2,LHRS3,LHRSX,LHRSC,BHRST
8065 FORMAT (5X,A4,6X,F7.1))
WRITE (55,*)
1 OPHRS(1,J,K,3), OPHRS(1,J,K,4), OPHRS(1,J,K,5), BHRSP
LHRST = LHRST + OPHRS(1,J,K,1)
LHRST2 = LHRST2 + OPHRS(1,J,K,2)
LHRST3 = LHRST3 + OPHRS(1,J,K,3)
LHRSTX = LHRSTX + OPHRS(1,J,K,4)
LHRSTC = LHRSTC + OPHRS(1,J,K,5)
8060 CONTINUE
IF (BLKHRSC(1,K).LT.0.1) GOTO 8070
WRITE(55,7065) BLKCK, LHRST, LHRST2, LHRST3, LHRSTX, LHRSTC, BLKHRS(1,K),
WRITE (55,*)
8070 CONTINUE
WRITE (55,*) 'LAST RECORD PROCESSED = ', DZDATE
WRITE (55,*) 'NUMBER OF RECORDS READ = ', RCOUNT
*
WRITE WB/DS HOURS TOTALS
WRITE(56,*)
WRITE(56,*) 'DOZERS WORK DUMPSITE HOUR TOTALS BY DESTINATION'
WRITE (56,9347)
9347 FORMAT (' ',6X,'TO',8X,'D O Z E R S')
WRITE(56,9350)
9350 FORMAT (' ',1X,',', '3X,'BLOCK',4X, 'A ',5X,'C ')
WRITE(56,*)
DO 9370 KK = 1,BDIM
   WBAHRS = WBAHRS + WBHRS(1,KK)
   WBCHR = WBCHR + WBHRS(2,KK)
   WBHRS = WBHRS(1,KK) + WBHRS(2,KK)
IF (WBHRS .LT. 0.1) GOTO 9370
WRITE(56,9360) DZBLK, DZLIFT, BLKCK, WBHRS(1,KK), WBHRS(2,KK),
9360 FORMAT ('A4,1X,A1,3X,A4,3(2X,F6.1))
WRITE (56,9380) DZBLK, DZLIFT, WBHRS, WBCHR, WBTHRS
9380 FORMAT ('A4,1X,A1,3X,'TOTAL',1X,F6.1,2X,F6.1,2X,F6.1)
IF (WBTHRS .NE. WBAHRS + WBCHR) THEN
   WRITE (56,*) 'PROBLEM WITH WB HOURS TOTAL; WBTHRS = ', WBTHRS
ENDIF
WRITE(56,*)
WRITE(56,*) 'NUMBER OF RECORDS READ = ', RCOUNT
WRITE (56,*) 'LAST RECORD PROCESSED = ', DZDATE
*
WRITE PO HOURS TOTALS - FILE 57
D-9 DOZERS
WRITE(57,*)
WRITE(57,*) 'A DOZERS PO HOURS'
WRITE(57,*)
9510 FORMAT (6X,'FROM',3X,'TO',5(4X,'FROM'))
WRITE(57,9510)
9520 FORMAT (5X,'BLOCK',1X,'BLOCK',2X,'T LIFT',2X,'2 LIFT',2X,'3 LIFT',
9520 FORMAT (5X,'BLK',1X,'BLK',2X,'T LIFT',2X,'2 LIFT',2X,'3 LIFT',
1 2X,'X LIFT',2X,'R LIFT',2X,'TOTAL')
WRITE(57,9520)
TCDOHRS = 0.0
DO 9580 K = 1,BDIM
   IF (BCDHRSC(1,K).LT.0.1) GOTO 9580
   LHRST = 0.0
   LHRST2 = 0.0
   LHRST3 = 0.0
   LHRSTX = 0.0
   LHRSTC = 0.0
9580 CONTINUE
BCD = 0
DO 9550 D = 1,BDIM

SCDHRS = CDHRS(1,K,1,D) + CDHRS(1,K,2,D) + CDHRS(1,K,3,D) + CDHRS(1,K,4,D) + CDHRS(1,K,5,D)
1
IF (SCDHRS .LT. 0.1) GOTO 9550
LHRST = LHRST + CDHRS(1,K,1,D)
LHRST2 = LHRST2 + CDHRS(1,K,2,D)
LHRST3 = LHRST3 + CDHRS(1,K,3,D)
LHRSTX = LHRSTX + CDHRS(1,K,4,D)
LHRSTC = LHRSTC + CDHRS(1,K,5,D)

BCD = BCD + 1
WRITE (57,9530) BLK(K),BLK(D),CDHRS(1,K,1,D),CDHRS(1,K,2,D),

CDHRS(1,K,3,D),CDHRS(1,K,4,D),CDHRS(1,K,5,D),SCDHRS
9530 FORMAT (5X,A4,2X,A4,6(F6.1,2X))
9550 CONTINUE

TCDHRS = TCDHRS + BCDHRSC(K)
IF (BCD .LT. 2) GOTO 9570
WRITE(57,9560) BLK(K),LHRST,LHRS2,LHRS3,LHRSX,LHRSC,BCDHRSC(K)
9560 FORMAT(5X,A4,6X,6(F6.1,2X))
9570 WRITE(57,*)
9580 CONTINUE
WRITE(57,9590) TCDHRS
9590 FORMAT(5X,'TOTAL A DOZER PUSH OVERBURDEN HOURS = ',F6.1)
WRITE(57,*)

WRITE(57,*) 'C DOZER PUSH OVERBURDEN HOURS'
WRITE(57,*)
WRITE(57,9510)

TCDHRS = 0.0
DO 9680 K = 1,BDIM

IF (BCDHRSM(2,K) .LT. 0.1) GOTO 9680
LHRST = 0.0
LHRST2 = 0.0
LHRST3 = 0.0
LHRSTX = 0.0
LHRSTC = 0.0

BCD = 0
DO 9650 D = 1,BDIM

SCDHRS = CDHRS(2,K,1,D) + CDHRS(2,K,2,D) + CDHRS(2,K,3,D) + CDHRS(2,K,4,D) + CDHRS(2,K,5,D)
1
IF (SCDHRS .LT. 0.1) GOTO 9650
LHRST = LHRST + CDHRS(2,K,1,D)
LHRST2 = LHRST2 + CDHRS(2,K,2,D)
LHRST3 = LHRST3 + CDHRS(2,K,3,D)
LHRSTX = LHRSTX + CDHRS(2,K,4,D)
LHRSTC = LHRSTC + CDHRS(2,K,5,D)

BCD = BCD + 1
WRITE (57,9530) BLK(K),BLK(D),CDHRS(2,K,1,D),CDHRS(2,K,2,D),

CDHRS(2,K,3,D),CDHRS(2,K,4,D),CDHRS(2,K,5,D),SCDHRS
9650 CONTINUE

TCDHRS = TCDHRS + BCDHRSC(2,K)
IF (BCD .LT. 2) GOTO 9670
WRITE(57,9560) BLK(K),LHRST,LHRS2,LHRS3,LHRSX,LHRSC,BCDHRSC(2,K)
9670 WRITE(57,*)
9680 CONTINUE
WRITE(57,9690) TCDHRS
9690 FORMAT(5X,'TOTAL C DOZER PUSH OVERBURDEN HOURS = ',F6.1)
WRITE(57,*)

WRITE(57,*) 'NUMBER OF RECORDS READ = ',RCOUNT
WRITE (57,*) 'LAST RECORD PROCESSED = ',DZDATE
WRITE(06,*)'FINISHED; ',ICOUNT,' RECORDS PROCESSED'
GOTO 9999
9911 WRITE(06,*)'READ ERROR LDDATE = ', DZDATE, ',ICOUNT = ',ICOUNT
9999 STOP
END

* * * * OPHOURS DOZER * * * *

C DOZER OPERATION HOURS TOTALS BY BLOCK AND LIFT

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>OPER</th>
<th>T LIFT</th>
<th>2 LIFT</th>
<th>3 LIFT</th>
<th>X LIFT</th>
<th>R LIFT</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DZC</td>
<td>SD1</td>
<td>RR</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td>SD2</td>
<td>RR</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>

A DOZERS OPERATION HOURS TOTALS BY BLOCK AND LIFT

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>OPER</th>
<th>T LIFT</th>
<th>2 LIFT</th>
<th>3 LIFT</th>
<th>X LIFT</th>
<th>R LIFT</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DZA</td>
<td>MB1</td>
<td>ST</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>MB1</td>
<td>PD</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>MB1</td>
<td>FO</td>
<td>10.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td>18.0</td>
</tr>
<tr>
<td></td>
<td>MB1</td>
<td>DS</td>
<td>4.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>8.0</td>
</tr>
<tr>
<td></td>
<td>MB1</td>
<td>FC</td>
<td>0.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>MB1</td>
<td></td>
<td>15.0</td>
<td>14.0</td>
<td>0.0</td>
<td>0.0</td>
<td>29.0</td>
</tr>
<tr>
<td></td>
<td>MB2</td>
<td>ST</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>MB2</td>
<td>PD</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>MB2</td>
<td>FO</td>
<td>7.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>MB2</td>
<td>PO</td>
<td>7.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>MB2</td>
<td>DS</td>
<td>3.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>MB2</td>
<td></td>
<td>18.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>18.0</td>
</tr>
<tr>
<td></td>
<td>MB3</td>
<td>ST</td>
<td>5.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>MB3</td>
<td></td>
<td>5.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>SD1</td>
<td>RR</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>SD2</td>
<td>RR</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
<td>5.0</td>
</tr>
</tbody>
</table>

* * * * POHOURS DOZER * * * *

A DOZERS PO HOURS

<table>
<thead>
<tr>
<th>FROM</th>
<th>TO</th>
<th>BLOCK</th>
<th>T LIFT</th>
<th>2 LIFT</th>
<th>3 LIFT</th>
<th>X LIFT</th>
<th>R LIFT</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DZA</td>
<td>MB2</td>
<td>SD2</td>
<td>7.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
</tr>
</tbody>
</table>

TOTAL A DOZER PUSH OVERBURDEN HOURS = 7.0

* * * * DSHOURS DOZER * * * *

DOZERS WORK DUMPSITE HOUR TOTALS BY DESTINATION TO DOZERS

<table>
<thead>
<tr>
<th>TO</th>
<th>BLOCK</th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD1</td>
<td>8.0</td>
<td>0.0</td>
<td>8.0</td>
</tr>
<tr>
<td>SD2</td>
<td>3.0</td>
<td>0.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>

TOTAL 11.0 0.0 11.0
APPENDIX D: DRILL

<table>
<thead>
<tr>
<th>DAY</th>
<th>MB1</th>
<th>T</th>
<th>50</th>
<th>20</th>
<th>7.0</th>
<th>5000</th>
<th>3.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAY 2</td>
<td>MB1</td>
<td>T</td>
<td>50</td>
<td>20</td>
<td>7.0</td>
<td>5000</td>
<td>3.5</td>
</tr>
<tr>
<td>DAY 3</td>
<td>MB2</td>
<td>T</td>
<td>50</td>
<td>20</td>
<td>7.0</td>
<td>5000</td>
<td>3.5</td>
</tr>
<tr>
<td>DAY 4</td>
<td>MB1</td>
<td>X</td>
<td>50</td>
<td>15</td>
<td>7.0</td>
<td>2500</td>
<td>3.5</td>
</tr>
<tr>
<td>DAY 5</td>
<td>MB1</td>
<td>X</td>
<td>50</td>
<td>15</td>
<td>7.0</td>
<td>2500</td>
<td>3.5</td>
</tr>
<tr>
<td>DAY 6</td>
<td>MB2</td>
<td>X</td>
<td>50</td>
<td>15</td>
<td>7.0</td>
<td>2500</td>
<td>3.5</td>
</tr>
</tbody>
</table>

PROGRAM DRILL

THE PURPOSE OF THIS PROGRAM IS TO TOTAL DRILLING HOURS AND MATERIALS BY BLOCK AND LIFT

CHARACTER*5 DRDATE
CHARACTER*4 DRBLK, BLK(22)
CHARACTER*1 DRLIFT, LIFT(5)
INTEGER DRHOLS, HOLES, BHOLES, LHOLES(22,5)
INTEGER DRDEP
INTEGER HOLFT, BHOLFT, LHOLFT(22,5)
REAL AVGDEP, BAVG, LAVG
REAL DRHRS, HRS, BHRS, LHRS(22,5)
INTEGER DRANFO, ANFO, BANFO, LANFO(22,5)
REAL DRLAB, ADRLAB, SHTLAB, BSHTL, LSHTL(22,5)
CHARACTER*15 DRCOMM
INTEGER ICOUNT, BDIM,K,L, II, IMAX
REAL RATE

INITIALIZE VARIABLES

BDIM = 22
IMAX = 1000
HOLES = 0
HOLFT = 0
ANFO = 0
ICOUNT = 0
HRS = 0.0
SHTLAB = 0.0

ADRLAB IS DEFAULT VALUE FOR DRLAB

ADRLAB = 0.0

DO 110 K=1,BDIM
DO 100 L = 1,5
LHOLES(K,L) = 0
LHOLFT(K,L) = 0
LHRS(K,L) = 0.0
LANFO(K,L) = 0
LSHTL(K,L) = 0.0
100 CONTINUE
110 CONTINUE

CALL ABLOCK(BLK,BDIM)
CALL ALIFT(LIFT)
READ DATA AND TOTALS DRILL

1000 READ (13,1013,ERR=9900,END=9000)
   1 DRDATE,DRBLK,DRLIFT,DRHOLS,DRDEP,DRHRS,
   2 DRANFO,DLAB,DRCOMM
1013 FORMAT (A5,3X,A4,1X,A1,5X,I2,8X,I2,7X,F5.2,6X,I4,6X,F3.1,2X,A15)

CALL KBLOCK (BLK,BDIM,DRBLK,K)
CALL JLIFT (DRLIFT,L)

LIFT TOTALS THEN JOB TOTALS

LHOLES(K,L) = LHOLES(K,L) + DRHOLS
LHOLFT(K,L) = LHOLFT(K,L) + (DRHOLS*DRDEP)
LHRS(K,L) = LHRS(K,L) + DRHRS
LANFO(K,L) = LANFO(K,L) + DRANFO
IF (DRLAB .EQ. 0.0)   DRLAB = 3.5
LSHTL(K,L) = LSHTL(K,L) + DRLAB

HOLES = HOLES + DRHOLS
HOLFT = HOLFT + (DRHOLS*DRDEP)
HRS = HRS + DRHRS
ANFO = ANFO + DRANFO
SHTLAB = SHTLAB + DRLAB

ENDLESS LOOP PROTECTION

ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9910

GOTO 1000

FINISHED READ ENTER WRITE; 'B' VARIABLES = BLOCK TOTALS

9000 AVGDEP = REAL(HOLFT)/REAL(HOLES)
WRITE (60,X)
WRITE (60,X) 'DRILL TOTALS '
WRITE (60,X)
WRITE (60,9010)
9010 FORMAT (18X,'DRILL',2X,'AVERAGE',9X,'DRILL',3X,'SHOOT',4X, 1 'DRILLING RATE')
WRITE (60,9020)
9020 FORMAT (5X,'BLK',1X,'!',1X,'!',2X,'HOLES',2X,'FEET',3X,'DEPTH',3X, 1 'ANFO',2X,'HOURS',3X,'LABOR',4X,'FEET PER HOUR')
WRITE (60,9021)
9021 FORMAT ('DRL')
DO 9032 K = 1,BDIM
   BHOLES = 0
   BHOLFT = 0
   BHRS = 0.0
   BANFO = 0
   BSHTL = 0.0
   DO 9026 L = 1,5
      IF (LHOLES(K,L) .EQ. 0) GOTO 9026
      LAVG = REAL(HLHOLFT(K,L))/LHOLES(K,L)
      RATE = LHOLFT(K,L)/LHRS(K,L)
      WRITE (60,9024) BLK(K),LIFT(L),LHOLES(K,L),LHOLFT(K,L), 1 LAVG,LANFO(K,L),LHRS(K,L),LSHTL(K,L),RATE
9024 FORMAT (4X,A4,1X,A1,2X,I4,2X,I6,2X,F5.1,1X,I7,1X,F6.1,1X, 1 F6.1,8X,F6.2)
   BHOLES = BHOLES + LHOLES(K,L)
   BHOLFT = BHOLFT + LHOLFT(K,L)
   BHRS = BHRS + LHRS(K,L)
   BANFO = BANFO + LANFO(K,L)
DO 9032 K = 1,BDIM
BSHTL = BSHTL + LSHTL(K,L)

CONTINUE
IF (BHOLES .EQ. 0) GOTO 9032
BAVG = REAL(BHOLFT)/BHOLES
RATE = BHOLFT/BHRS
* WRITE(60,9030) BLK(K),BHOLES,BHOLFT,BAVG,BANFO,BHRS,BSHTL,
1 FORMAT (4X,A4,4X,I4,2X,I6,2X,F5.1,1X,I7,1X,F6.1,1X,F6.1,8X,
1 F6.2)
WRITE(60,*)
9032 CONTINUE
* FINALE
* * *
RATE = HOLFT/HRS
WRITE (60,*)
WRITE (60,9040) HOLES, HOLFT, AVGDEP, ANFO, HRS, SHTLAB, RATE
9040 FORMAT (5X,'TOTAL',2X,I4,2X,I6,2X,F5.1,1X,I7,1X,F6.1,1X,F6.1,8X,
+ F6.2)
WRITE (60,*)
WRITE (60,*) 'NUMBER OF RECORDS PROCESSED = ', ICOUNT
WRITE (60,*) 'LAST DATE PROCESSED = ', DRDATE
WRITE (06,*) 'FINISHED; ', ICOUNT, ' RECORDS PROCESSED'
GOTO 9999
9900 WRITE (06,*) ' READ ERROR DRDATE = ', DRDATE
9910 WRITE (06,*) 'LOOPING; ICOUNT = ', ICOUNT, 'DRDATE = ', DRDATE
9999 STOP
END
*
**
** DRILLING TOTALS
**

<table>
<thead>
<tr>
<th>BLK L</th>
<th>HOLES</th>
<th>FEET</th>
<th>DEPTH</th>
<th>ANFO</th>
<th>HOURS</th>
<th>LABOR</th>
<th>DRILLING RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>T</td>
<td>100</td>
<td>2000</td>
<td>20.0</td>
<td>10000</td>
<td>14.0</td>
<td>7.0</td>
</tr>
<tr>
<td>MB1</td>
<td>X</td>
<td>100</td>
<td>1500</td>
<td>15.0</td>
<td>5000</td>
<td>14.0</td>
<td>7.0</td>
</tr>
<tr>
<td>MB1</td>
<td></td>
<td>200</td>
<td>3500</td>
<td>17.5</td>
<td>15000</td>
<td>28.0</td>
<td>14.0</td>
</tr>
<tr>
<td>MB2</td>
<td>T</td>
<td>50</td>
<td>1000</td>
<td>20.0</td>
<td>5000</td>
<td>7.0</td>
<td>3.5</td>
</tr>
<tr>
<td>MB2</td>
<td>X</td>
<td>50</td>
<td>750</td>
<td>15.0</td>
<td>2500</td>
<td>7.0</td>
<td>3.5</td>
</tr>
<tr>
<td>MB2</td>
<td></td>
<td>100</td>
<td>1750</td>
<td>17.5</td>
<td>7500</td>
<td>14.0</td>
<td>7.0</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>300</td>
<td>5250</td>
<td>17.5</td>
<td>22500</td>
<td>42.0</td>
<td>21.0</td>
</tr>
</tbody>
</table>
APPENDIX E: OTHER

**OTHER INDATA**

<table>
<thead>
<tr>
<th>DAY</th>
<th>16 2</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAY</td>
<td>40 5 1</td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>40 5 1</td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>40 5 1</td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>40 5 1</td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>40 5 1</td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>40 5 1</td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>40 5 1</td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>40 5 1</td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>40 5 1</td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>1060</td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>570</td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>SED</td>
<td>5 4</td>
<td></td>
</tr>
<tr>
<td>SED</td>
<td>2 3</td>
<td></td>
</tr>
<tr>
<td>MAN</td>
<td>5 12</td>
<td></td>
</tr>
<tr>
<td>MAN</td>
<td>2 12</td>
<td></td>
</tr>
</tbody>
</table>

**OTHER FORTRAN**

```fortran
PROGRAM OTHER

THE PURPOSE OF THIS PROGRAM IS TO TOTAL QUANTITIES IN OTHER INDATA FILE: MANHOURS, OTHER MACHINE HOURS, FUEL, COAL TONNAGE, MANDAYS

CHARACTER*5 MIDATE
INTEGER LABHRS, TOTLAB, MANHRS
INTEGER JOBMAX, X, JOBDAY
INTEGER SUP, TSUP
CHARACTER*4 COLBK, BLOCK(22)
CHARACTER*1 AUGCOL
INTEGER COLTON, COAL, BLKCOL(2,22)
INTEGER FUEL, TOFUEL
CHARACTER*3 OMAC, MACHIN(14)
INTEGER OMACHR, MACTOT, MACHRS(14,22,15), MACTOT(14)
INTEGER AUGLAB, AUGMH(22), AUGMH(22), AUGMHT
INTEGER SEDBAT, SEDBA(22)
CHARACTER*4 OMACBL
CHARACTER*1 OMACLI
CHARACTER*25 MICOMM
INTEGER I, ICOUNT, IMAX, INUM
INTEGER BDIM, K, A, IAUG, CC, CCDIM, IDIM

INITIALIZE VARIABLES

IDIM = 14
CCDIM = 15
BDIM = 22
IMAX = 1000
AUGMHT = 0
COAL = 0
TOFUEL = 0
TOTLAB = 0
MACTOT = 0
ICOUNT = 0
JOBMAX = 0
TSUP = 0
```

84
SEDBAT = 0

* ARRAY CONTENTS
* MACHIN(I): OMAC SYMBOLS
* MACH(T): TOTAL OMAC HOURS BY OMAC TYPE
* MACHRS(I,K,CC): OMAC HOURS BY OMAC TYPE, BLOCK, AND COST CATEGORY
* BLOCK(K): COAL BLOCKS AND MINESITE OPERATION AREAS
* BLKCOL(A,K): COAL Tonnage BY MINING METHOD AND BLOCK
* AUGMH(K): AUGER MAN HOURS BY BLOCK
* SEDBAC: HYDROSEEDER BATCHES PER RECLAMATION BLOCK
* MACHRS ARE TOTALED BY COST CATEGORY (ARRAY SUBSCRIPT CC) AS FOLLOWS:

* COCAT(01) = 'PRE-DRILL'
* COCAT(02) = 'DRILL AND SHOOT'
* COCAT(03) = 'CARRY AND PUSH'
* COCAT(04) = 'LOAD & HAUL: LOADERS'
* COCAT(05) = 'LOAD & HAUL: DOZERS'
* COCAT(06) = 'LOAD & HAUL: HAULERS'
* COCAT(07) = 'LOAD & HAUL: DUMPSITE'
* COCAT(08) = 'LOAD & HAUL: TOTAL'
* COCAT(09) = 'COAL: CLEAN & LOAD'
* COCAT(10) = 'COAL: AUGER'
* COCAT(11) = 'COAL: HAULING'
* COCAT(12) = 'RECLAMATION'
* COCAT(13) = 'OTHER'
* COCAT(14) = 'OVERHEAD'

DO 50 I = 1,IDIM
  MACHIN(I) = ' '
  MACH(T) = 0
DO 40 K = 1,BDIM
  DO 30 CC = 1,CCDIM
    MACHRS(I,K,CC) = 0
  CONTINUE
40 CONTINUE
DO 50 K = 1,BDIM
  BLKCOL(1,K) = 0
  BLKCOL(2,K) = 0
  AUGMH(K) = 0
  SEDBAC(K) = 0
50 CONTINUE
CALL ABLKCOL(BLOCK,BDIM)
WRITE(C06,*,'ENTER READ DATA')
READ DATA
FILE 14 OTHER INDATA
FILE 70 TOTALS OTHER
1000 READ (14,1014,ERR=9910,END=9000)
  1 MIDATE,LABHRS,JOBMEN,SUP,COLBLK,AUGCOL,COLTON,FUEL,
  2 OMAC,OMACHR,AUGLAB,OMACBL,OMICMM
1014 FORMAT (A5,4X,I2,1X,I1,1X,I1,3X,A4,A1,1X,I4,1X,I4,2X,A3,2X,
  1 I2,1X,I2,2X,A4,3X,A25)

CALCULATE JOB TOTALS

TOLAB = TOTLAB + LABHRS
JOBDAY = JOBDAY + JOBMEN
TOFUEL = TOFUEL + FUEL
TSUP = TSUP + SUP

ENDLESS LOOP PROTECT

ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9910

IF 'OMAC' RECORDED: ASSIGN VALUE OF I TO MACHINE TYPE 'OMAC'
LOAD MACHINE TYPE 'OMAC' SYMBOLS INTO ARRAY MACHIN(I)
TOTAL MACHINE HOURS BY OMAC TYPE IN MACHT(I) ARRAY
TOTAL MACHINE HOURS BY OMAC TYPE, COCAT, AND LOCATION
IN MACHRS(I,K,CC) ARRAY
AUGLAB REPRESENTS: AUGER LABOR HOURS WHEN OMAC = 'AUG'
SEEDER BATCHES WHEN OMAC = 'SED'
COST CATEGORY OTHERWISE

IF (OMAC .EQ. 'AUG') GOTO 3000
CALL KBLOCK (BLOCK,BDIM,OMACBL,K)
I = 1
2000 IF (MACHIN(I) .EQ. OMAC) GOTO 2100
IF (MACHIN(I) .EQ. ' ') THEN
MACHIN(I) = OMAC
INUM = I
GOTO 2100
ENDIF
I = I + 1
IF (I .GT. IDIM) GOTO 9900
GOTO 2000
2100 MACHT(I) = MACHT(I) + OMACHR
MACTOT = MACTOT + OMACHR
IF (OMAC .EQ. 'AUG') THEN
CC = 10
AUGMHT = AUGMHT + AUGLAB
AUGMH(K) = AUGMH(K) + AUGLAB
ELSE IF (OMAC .EQ. 'SED') THEN
CC = 12
SEDBAT = SEDBAT + AUGLAB
SEDBA(K) = SEDBA(K) + AUGLAB
ELSE
CC = AUGLAB
ENDIF
MACHRS(I,K,CC) = MACHRS(I,K,CC) + OMACHR

TOTAL COAL TONNAGE BY BLOCK AND MINING METHOD
ASSIGN: A=1 FOR STRIP COAL, A=2 FOR AUGER COAL

3000 IF (COLTON .EQ. 0) GOTO 1000
A = 1
IF (AUGCOL .EQ. 'A') A = 2
CALL KBLOCK (BLOCK,BDIM,COLBLK,K)
BLKCOL(A,K) = BLKCOL(A,K) + COLTON
COAL = COAL + COLTON
GOTO 1000

FINISHED READ ENTER WRITE

9000 WRITE(70,*)
WRITE(70,*) 'TOTALS: OTHER MACHINERY, COAL, FUEL, LABOR'
WRITE(70,*)
MICOMM = 'COAL PRODUCED ='
WRITE(70,9001) MICOMM,COAL
MICOMM = 'FUEL PURCHASED ='
WRITE(70,9002) MICOMM,TOFUEL
MICOMM = 'LABOR DAYS ='
WRITE(70,9003) MICOMM,JOBDAY
MICOMM = 'SUPERVISOR DAYS ='
WRITE(70,9004) MICOMM,TSUP
MICOMM = 'LABOR HOURS ='
WRITE(70,9005) MICOMM,TOTLAB
9001 FORMAT (1X,A17,I7,' TONS')
9002 FORMAT (1X,A17,I7,' GALLONS')
9003 FORMAT (1X,A17,I7)
9004 FORMAT (7X,'OTHER MACHINERY OPERATION')
WRITE(70,9005)
87

9005 FORMAT(5X,'MACH',2X,'BLK',1X,'L',2X,'HOURS',1X,'LAB/BAT',1X,'CC')
WRITE(70,9006)
9006 FORMAT('MIS')
DO 9090 I = 1,INUM
IF (MACHT(I) .EQ. 0) GOTO 9090
DO 9080 K = 1,BDIM
DO 9025 CC = 1,CDDIM
IF (MACHRS(I,K,CC) .EQ. 0) GOTO 9025
IF (MACHIN(I) .EQ. 'AUG') THEN
WRITE(70,9010) MACHIN(I), BLOCK(K), MACHRS(I,K,CC), AUGMHT(K),CC
9010 FORMAT(5X,A3,2X,A4,4X,I5,2X,I5,2X,I2)
ELSE IF (MACHIN(I) .EQ. 'SED') THEN
WRITE(70,9010) MACHIN(I), BLOCK(K), MACHRS(I,K,CC), SEDBA(K),CC
ELSE WRITE(70,9020) MACHIN(I), BLOCK(K), MACHRS(I,K,CC),CC
9020 FORMAT (5X,A3,2X,A4,4X,I5,9X,12)
ENDIF
9025 CONTINUE
9030 IF (K .EQ. BDIM) THEN
WRITE(70,9040) MACHT(I)
9040 FORMAT(11X,'TOTAL',2X,I5)
WRITE(70,*)
ENDIF
9080 CONTINUE
9090 CONTINUE

WRITE(70,*)
WRITE(70,9130) MACTOT
9130 FORMAT(5X,'TOTAL HOURS OTHER MACHINES = ',1X,I7)
WRITE(70,9140) AUMGT
9140 FORMAT(5X,'TOTAL MAN HOURS ON AUGER = ',1X,I7)
IF (AUMGT .GT. 0) THEN
MANHRS = MACTOT + AUMGT - MACHT(IAug)
ELSE
MANHRS = MACTOT
ENDIF
WRITE(70,9144) MANHRS
9144 FORMAT(5X,'TOTAL OTHER MACHINE MAN HOURS = ',1X,I7)

COAL TOTALS - TRAILING 0 AND TO FACILITATE COST INPUTS
WRITE(70,*)
WRITE(70,*) 'COAL PRODUCTION'
WRITE(70,9150)
9150 FORMAT(3X,'BLOCK',3X,'TONS')
WRITE(70,9155)
9155 FORMAT('COL')
9161 DO 9200 K = 1,BDIM
IF (BLKCOL(1,K) .NE. 0) THEN
WRITE(70,9120) BLOCK(K), BLKCOL(1,K)
9120 FORMAT(4X,A4,2X,I5,7X,6X,'0',4X,').')
ENDIF
9200 CONTINUE
DO 9300 K = 1,BDIM
IF (BLKCOL(2,K) .NE. 0) THEN
WRITE(70,9220) BLOCK(K), BLKCOL(2,K)
9220 FORMAT(4X,A4,2X,I5,2X,'AUGER',6X,'0','4X,'').'
ENDIF
9300 CONTINUE
WRITE(70,*)
WRITE(70,9310) COAL
WRITE(70,*)
9310 FORMAT(5X,'TOTAL',1X,I5)

FINALE
WRITE(70,*) 'NUMBER OF RECORDS PROCESSED = ', ICOUNT
WRITE(70,*) 'LAST DATE PROCESSED = ', MIDATE
WRITE(06,*) 'FINISHED ', ICOUNT, ' RECORDS PROCESSED'
GOTO 9999
9900 WRITE(06,*) 'OMAC ASSIGNMENT ERROR; I = ', I, ' OMAC = ', OMAC
GOTO 9999
9910 WRITE(06,*) 'READ PROBLEM; MIDATE = ', MIDATE, ', ICOUNT = ', ICOUNT
9999 STOP

* * * TOTALS OTHER * * *

TOTALS: OTHER MACHINERY, COAL, FUEL, LABOR

COAL PRODUCED = 1930 TONS
FUEL PURCHASED = 2000 GALLONS
LABOR DAYS = 49
SUPERVISOR DAYS = 9
LABOR HOURS = 392

OTHER MACHINERY OPERATION

<table>
<thead>
<tr>
<th>MACH</th>
<th>BLK</th>
<th>L</th>
<th>HOURS</th>
<th>LAB/BAT</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRD</td>
<td>ROAD</td>
<td>2</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td>MB2</td>
<td>7</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td></td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUG</td>
<td>MB2</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SED</td>
<td>SD1</td>
<td>5</td>
<td>4</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>SED</td>
<td>SD2</td>
<td>2</td>
<td>3</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td></td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAN</td>
<td>SD1</td>
<td>5</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAN</td>
<td>SD2</td>
<td>2</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td></td>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TOTAL HOURS OTHER MACHINES = 26
TOTAL MAN HOURS ON AUGER = 6
TOTAL OTHER MACHINE MAN HOURS = 29

COAL PRODUCTION

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>TONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>1060</td>
</tr>
<tr>
<td>MB2</td>
<td>570</td>
</tr>
<tr>
<td>MB2</td>
<td>300</td>
</tr>
<tr>
<td>TOTAL</td>
<td>1930</td>
</tr>
</tbody>
</table>

NUMBER OF RECORDS PROCESSED = 15
LAST DATE PROCESSED = D 11
APPENDIX F: MOVE

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DATA MOVINPUT</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.30</td>
<td>30.0</td>
<td>10.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.0</td>
<td>20.0</td>
<td>200.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HLR</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>SD1</td>
<td>42.0</td>
<td>32.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>SD2</td>
<td>10.0</td>
<td>14.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB3</td>
<td>SD1</td>
<td>4.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB3</td>
<td>SD1 R</td>
<td>1.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB3</td>
<td>SD2</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB3</td>
<td>SD2</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB3</td>
<td></td>
<td>11.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td><strong>HLP</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>SD1</td>
<td>7.14</td>
<td>8.28</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>MB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>SD2</td>
<td>8.00</td>
<td>8.21</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>MB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB3</td>
<td>SD1</td>
<td>5.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>MB3</td>
<td>SD2</td>
<td>5.60</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>MB3</td>
<td></td>
<td>5.27</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td><strong>LDA</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>SD1</td>
<td>0.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DZA</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>SD2</td>
<td>7.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>MB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>VOL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>T</td>
<td>8000</td>
<td>.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>2</td>
<td>5000</td>
<td>.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>T</td>
<td>4000</td>
<td>.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>2</td>
<td>2500</td>
<td>.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB3</td>
<td>T</td>
<td>1200</td>
<td>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>END</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAY 1</td>
<td>1999</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAY 11</td>
<td>1999</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PONY RUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PROGRAM MOVE

* * * COSTSUM * * *

PROGRAMMED BY C.E. ZIPPER, 1985

DOCUMENTATION: VIRGINIA AGR. EXP. STATION BULLETIN NO. 86-1

VPI & SU, BLACKSBURG 24061

PROGRAM MOVE

THIS PROGRAM'S PURPOSE IS GENERATE AN INPUT TO PROGRAM COST WHICH ALLOWS RECLAMATION EXPENSES TO BE DISTRIBUTED FROM DESTINATION TO SOURCE BASED UPON THE PROPORTION OF THE TOTAL MATERIAL RECLAIMED ORIGINATING IN EACH SOURCE BLOCK. PURPOSE IS ACHIEVED THROUGH ESTIMATION OF QUANTITY OF OVERBURDEN MOVED FROM EACH BLOCK AND LIFT TO EACH DESTINATION BY EACH OF THREE MODES OF MOVEMENT: HAULER, LOADER CARRY, DOZER PUSH. INPUTS: MACHINE HOURS FOR EACH MOVEMENT ROUTE, CAPACITY ESTIMATES, AND ESTIMATED RATES OF MOVEMENT. SOURCE BLOCK AND LIFT UNDISTURBED VOLUME AND SWELL ESTIMATES ARE ALSO ENTERED; THE INITIAL ESTIMATES OF QUANTITIES MOVED

89
ARE ADJUSTED SO THAT THE TOTALED ESTIMATE OF MATERIAL MOVED FROM EACH SOURCE BLOCK AND LIFT EQUALS THE UNDISTURBED VOLUME X SHELL INPUT.

DEFAULT VALUES FOR MOVEMENT RATES ARE ALSO ENTERED; MOVED QUANTITY OVER ANY ROUTE WITH NO SPECIFIC MOVEMENT RATE IS CALCULATED USING DEFAULT VALUES.

IT ALSO PRODUCES OUTPUT OF OVERBURDEN BCY VOLUME TOTALS AND QUANTITIES HAULED, CARRIED, AND PUSHED FROM EACH BLOCK-LIFT.

REAL HHRSC22,5,22,HPERC22,5,22,HFRAC
REAL LHRSC22,5,22,LPERC22,5,22,CFRAC
REAL DHRSC22,5,22,DPERC22,5,22,PFRC
REAL HRS(22,5,22),TOTHRS
REAL BHRSC(22),BLHRS(22,5),BTHRSC(22,22),THRS(22)
REAL RATIO(22,5),JRATIO,HFRAC
INTEGER HYDS(22,5,22)
INTEGER LDYS(22,5,22)
INTEGER TDYS(22,5,22)
INTEGER TYYS(22,5,22),TOTYDS
INTEGER HYD(22,5,22),HFRAC
INTEGER LDY(22,5,22),CFRAC
INTEGER TDY(22,5,22),DFRC
INTEGER TYD(22,5,22),PFRC
INTEGER HAUL(22,5,22),PUSH(22,5)
REAL RH,RHLB,RHLT,RHLTOT
REAL RC,RLDB,RLDT,RLDTOT
REAL RP,RADB,RADT,RADTOT
REAL RHL(5),RLD(5),RDL(5)
REAL LIFHRS(5),BLKHRS
REAL LIPERC(5),BLKPER
INTEGER BLKVOL(22),LIFVOL(22,5),TOTVOL,VOLUME
INTEGER NEWVOL(22),NEWTOT
INTEGER B,L,T,BDIM,LDIM,TDIM,MOTES,TT
INTEGER ICOUNT,IMAX
CHARACTER*1 LIF,LIFT(5),R
CHARACTER*4 BLK,DES,BLOCK(22)
CHARACTER*3 GO
CHARACTER*5 CHAR
CHARACTER*15 DAYONE,DAYEND, JOB
REAL HLCAP, LDCAP
REAL SWELL, DSWELL
REAL XSWELL(22,5)
REAL W,X,Y,Z,FLOAT
INTEGER IFIX,IX, IY

** INITIALIZE VARIABLES**

BDIM = 22
LDIM = 5
TDIM = 22
IMAX = 1000
ICOUNT = 0
TOTVOL = 0
TOTYDS = 0
TOTHRS = 0.0

FILE 71 DATA MOVINPUT
FILE 72 BYSOURCE MOVE
FILE 73 BYDEST MOVE
FILE 74 COSINPUT MOVE
FILE 75 HOURS MOVE

READ INITIAL INPUTS:
DSWELL: ESTIMATE OF JOB AVERAGE SWELL (E.G. 1.3)
HLCAP & LDCAP: AVERAGE HAULER LOAD, AVERAGE LOADER CARRY (LCY)
HPERC, LPERC, DPERC: DEFAULT VALUES, HAULER AND LOADER LOADS PER HOUR, DOZER PUSH RATE (LCY/HOUR)

READ (71,5) DSWELL
READ (71,5) HLCAP,LDCAP
READ (71,5) HLPER,LDPER,DZPER
5 FORMAT (5X,3(F10.5))

* INITIALIZE ARRAYS

SUBSCRIPTS:
B = SOURCE BLOCK, L = SOURCE LIFT, T = DESTINATION

NAMES:
__YDS = LCY, __HRS = HOURS, __PER = MOVEMENT RATE
__RATIO = 'MOVED' VOLUME (AS PER INPUTS) DIVIDED BY
__VOLUME X SWELL
B = BY SOURCE BLOCK
B__ = BY SOURCE BLOCK AND LIFT
T = BY DESTINATION
B__T = BY SOURCE BLOCK TO DESTINATION
H = HAULER
L__ = LOADER CARRY
D = DOZER PUSH

DO 30 B = 1,BDIM
  BLKVCOL(B) = 0
  BYDSC(B) = 0
  TYDSC(B) = 0
  BHRSC(B) = 0.0
  THRSC(B) = 0.0
  BRATIO(B) = 0.0
DO 20 L = 1,LDIM
  LIFVOL(B,L) = 0
  BLYDSC(B,L) = 0
  BLHRSC(B,L) = 0
  HAUL(B,L) = 0
  CARRY(B,L) = 0
  PUSH(B,L) = 0
  RATIO(B,L) = 0.0
  XSWELL(B,L) = 0.0
DO 10 T = 1,TDIM
  HHRSC(B,L,T) = 0.0
  LHRSC(B,L,T) = 0.0
  DHRSC(B,L,T) = 0.0
  HPER(B,L,T) = HLPER
  LPER(B,L,T) = LDPER
  DPER(B,L,T) = DZPER
  MYDSC(B,L,T) = 0
  LYDSC(B,L,T) = 0
  DYDSC(B,L,T) = 0
  YDSC(B,L,T) = 0
  HRS(B,L,T) = 0.0
10 CONTINUE
20 CONTINUE
30 CONTINUE
DO 50 B = 1,BDIM
  DO 40 T = 1,TDIM
    BTYDSC(B,T) = 0
    BTHRSC(B,T) = 0
40 CONTINUE
50 CONTINUE
TOTHRS = 0.0
CALL ABLOCK (BLOCK,BDIM)
CALL ALIFT (LIFT,LDIM)

READ INPUT RECORD 4 - 'GO' VARIABLE IDENTIFIES TYPE VIA 9700
READ(71,70,END=9905,ERR=9905) GO
70 FORMAT(A3)
GOTO 9700

HAULER INPUTS: HRSOURCE HAULER FORMAT
1000 READ(71,1010,END=9910,ERR=9910)GO,BLK,DES,R,LIFHRS(1),LIFHRS(2),
LIFHRSC3, LIFHRSC4, LIFHRSC5, BLKHRS
1010 FORMAT(A3, 2X, A4, 2X, A4, 1X, A1, 1X, 6(F7.1, 1X))
ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9990
IF (GO .NE. 9) GOTO 9700
IF (DES .EQ. 9) GOTO 1000
IF (BLK .EQ. 9) GOTO 1000
CALL KBLOCK (BLOCK, BDIM, BLK, B)
CALL KBLOCK (BLOCK, BDIM, DES, T)
DO 1020 L = 1, 5
HHRSC(L,T) = HHRSC(L,T) + LIFHRS(L)
1020 CONTINUE
GOTO 1000
*
* HLP
* LDSPERHR HAULER FILE
*
1500 WRITE(06, *) 'HAULER PER BLOCK READ; ICOUNT = ', ICOUNT
1501 READ(71, 151), END=9915, ERR=9915) GO, BLK, DES, LIFPER(1), LIFPER(2),
+ LIFPER(3), LIFPER(4), LIFPER(5), BLKPER
1510 FORMAT(A3, 2X, A4, 2X, A4, 3X, 6(F7.2, 1X))
ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9990
IF (GO .NE. 9) GOTO 9700
IF (DES .EQ. 9) GOTO 1501
IF (BLK .EQ. 9) GOTO 1501
CALL KBLOCK (BLOCK, BDIM, BLK, B)
CALL KBLOCK (BLOCK, BDIM, DES, T)
DO 1520 L = 1, 5
IF (LIFPER(L) .GT. 0.1) HPER(L, T) = LIFPER(L)
1520 CONTINUE
GOTO 1501
*
* LDA
* LOADER INPUTS: CDHOURS LOADER FORMAT
*
2000 WRITE(06, *) 'LOADER HOURS BLOCK READ; ICOUNT = ', ICOUNT
2001 READ(71, 2010), END=9920, ERR=9920) GO, BLK, DES, LIFHRSC1, LIFHRSC2,
+ LIFHRSC3, LIFHRSC4, LIFHRSC5, BLKHRS
2010 FORMAT(A3, 2X, A4, 2X, A4, 6(F7.2, 1X))
ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9990
IF (GO .NE. 9) GOTO 9700
IF (DES .EQ. 9) GOTO 2001
IF (BLK .EQ. 9) GOTO 2001
CALL KBLOCK (BLOCK, BDIM, BLK, B)
CALL KBLOCK (BLOCK, BDIM, DES, T)
DO 2020 L = 1, 5
LHRS(L, T) = LHRS(L, T) + LIFHRSC(L)
2020 CONTINUE
GOTO 2001
*
* LDP
* LOADER PER BLOCK READ; ICOUNT = ', ICOUNT
2500 WRITE(06, *) 'LOADER HOURS BLOCK READ; ICOUNT = ', ICOUNT
2501 READ(71, 2510), END=9925, ERR=9925) GO, BLK, DES, LIFPER(1), LIFPER(2),
+ LIFPER(3), LIFPER(4), LIFPER(5), BLKPER
ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9990
IF (GO .NE. 9) GOTO 2501
IF (DES .EQ. 9) GOTO 2501
IF (BLK .EQ. 9) GOTO 2501
CALL KBLOCK (BLOCK, BDIM, BLK, B)
CALL KBLOCK (BLOCK, BDIM, DES, T)
DO 2520 L = 1, 5
IF (LIFPER(L) .GT. 0.1) LPER(L, T) = LIFPER(L)
2520 CONTINUE
GOTO 2501
*
* DZA
* DOZER INPUTS: POHOURS DOZER FORMATS
*
3000 WRITE(06, *) 'DOZER HOURS BLOCK READ; ICOUNT = ', ICOUNT
3001 READ(71,3010,END=9930,ERR=9930) GO,BLK,DES,LIFHSR(1),LIFHSR(2),
+ LIFHSR(3),LIFHSR(4),LIFHSR(5),BLKHSR
3010 FORMAT(A3,2X,A4,2X,A4,6(F7.2,1X))
   ICOUNT = ICOUNT + 1
   IF (ICOUNT .GT. IMAX) GOTO 9990
   IF (GO .NE. ' ') GOTO 9700
   IF (DES .EQ. ' ') GOTO 3001
   IF (BLK .EQ. ' ') GOTO 3001
   CALL KBLOCK (BLOCK,BDIM,BLK,B)
   CALL KBLOCK (BLOCK,BDIM,DES,T)
   DO 3020 L = 1,5
      DHRSCB,L,T) = DHRSCB,L,T) + LIFHRSCL)
   3020 CONTINUE
   GOTO 3001
3001 CONTINUE
GOTO 3001

3500 WRITE(06,*) 'DOZER PER BLOCK READ; ICOUNT = ',ICOUNT
3501 READ(71,3010,END=9935,ERR=9935) GO,BLK,DES,LIFPER(1),LIFPER(2),
+ LIFPER(3),LIFPER(4),LIFPER(5),BLKPER
   ICOUNT = ICOUNT + 1
   IF (ICOUNT .GT. IMAX) GOTO 9990
   IF (GO .NE. ' ') GOTO 9700
   IF (DES .EQ. ' ') GOTO 3501
   IF (BLK .EQ. ' ') GOTO 3501
   CALL KBLOCK (BLOCK,BDIM,BLK,B)
   CALL KBLOCK (BLOCK,BDIM,DES,T)
   DO 3520 L = 1,5
      IF (LIFPERCL) .GT. 0.1) DPERB,L,T) = LIFPERCL)
   3520 CONTINUE
   GOTO 3501

7000 WRITE(06,*) 'DOZER PER BLOCK READ; ICOUNT = ',ICOUNT
7001 READ(71,7010,END=9970,ERR=9970) GO,BLK,LIF,VOLUME,SWELL
7010 FORMAT(A3,1X,A4,1X,A1,1X,I7,F8.3)
   ICOUNT = ICOUNT + 1
   IF (ICOUNT .GT. IMAX) GOTO 9990
   IF (GO .NE. ' ') GOTO 9700
   IF (BLK .EQ. ' ') GOTO 7000
   CALL JLIFT (LIF,L)
   IF (SWELL .LT. 0.01) SWELL = DSWELL
   X = VOLUME*(1.0000000 + SWELL) + 0.50000
   XS dwell (B,L) = 1.0000000 + SWELL
   LIFVOL(B,L) = LIFVOL(B,L) + X
   BLK VOL(B) = BLKVOL(B) + X
   TOTVOL = TOTVOL + X
   GOTO 7001

8000 READ (71,8010,END=8020,ERR=8020) DAYONE
READ (71,8010,END=8020,ERR=8020) DAYEND
READ (71,8010,END=8020,ERR=8020) JOB
WRITE(06,*) 'CALCULATING; ICOUNT = ',ICOUNT
8010 FORMAT (5X,A15)

8020 DO 8050 B = 1,BDIM
   DO 8040 L = 1,LDIM
      X = HHRSC(B,L,T)*HPER(B,L,T)*HLCAP
      Y = LHRSC(B,L,T)*LPER(B,L,T)*LDCAP
      Z = DHRSC(B,L,T)*DPER(B,L,T)
      HRS(B,L,T) = HHRSC(B,L,T) + LHRSC(B,L,T) + DHRSC(B,L,T)
   8040 CONTINUE
   GOTO 8050
8050 CONTINUE

93
H = IFIX(X)
C = IFIX(Y)
P = IFIX(Z)
TT = H + C + P
YDSCB,L,T) = TT
HYDSCB,L,T) = H
LYDSCB,L,T) = C
DYDSCB,L,T) = P
TOTYDS = TOTYDS + TT
BYDSCB,L,T) = BYDSCB,L,T) + TT
DYDSCB,L,T) = TYDSCB,L,T) + TT
BTYDSCB,L,T) = BTYDSCB,L,T) + TT
TOTHRS = TOTHRS + HRS(B,L,T)
BHRSCB,L,T) = BHRSCB,L,T) + HRS(B,L,T)
BLHRSCB,L,T) = BLHRSCB,L,T) + HRS(B,L,T)
BTHRSCB,L,T) = BTHRSCB,L,T) + HRS(B,L,T)

8030 CONTINUE
8040 CONTINUE
8050 CONTINUE

OUTPUT BY SOURCE BLOCK - MACHINE HOURS

8710 FORMAT ('MACHINE HOURS ',A15,' THRU ',A15,3X,A15)
WRITE(75,*)
WRITE(75,8720)
8720 FORMAT(5X,'FROM',6X,'TO',5X,'VIA',5X,6C'FROM',4X>'
WRITE(75,8730)
8730 FORMAT(5X,'BLOCK',4X,'BLOCK',2X,7X,'T LIFT 2 LIFT 3 LIFT X LIFT + R LIFT TOTAL')
WRITE(75,*)
RHLTOT = 0.0
RLDTOT = 0.0
RDZTOT = 0.0
DO 8950 B = 1,BDIM
IF (BHRSCB,B) .LT. 0.1) GOTO 8950
RHLB = 0.0
RLDB = 0.0
RDZB = 0.0
DO 8790 L = 1,LDIM
RHL(L) = 0.0
RLD(L) = 0.0
RDZ(L) = 0.0
8790 CONTINUE
DO 8900 T = 1,TDIM
IF (BTHRSCB,B,T) .LT. 0.1) GOTO 8900
RH = 0.0
RC = 0.0
RP = 0.0
MODES = 0
DO 8800 L = 1,LDIM
IF (HRS(B,L,T) .LT. 0.00) GOTO 8800
RH = RH + HRS(B,L,T)
RC = RC + LHRS(B,L,T)
RP = RP + DHRS(B,L,T)
RHL(L) = RHL(L) + HRS(B,L,T)
RLD(L) = RLD(L) + LHRS(B,L,T)
RDZ(L) = RDZ(L) + DHRS(B,L,T)
8800 CONTINUE
RHLB = RHLB + RH
RLDB = RLDB + RC
RDZB = RDZB + RP
WRITE (75,8810) BLOCK(B),BLOCK(T)
8810 FORMAT(5X,A5,1X,'TO',2X,A5)
IF (RH .GT. 0.0) THEN
CHAR = 'HAUL'
WRITE (75,8820) CHAR,HRS(B,1,T),HHRSCB,B,2,T),
+ HHRSCB,B,3,T),HHRSCB,B,4,T),HHRSCB,B,5,T),RH

MODES = MODES + 1
ENDIF
IF (RC .GT. 0.0) THEN
  CHAR = 'CARRY'
  WRITE (75,8820) CHAR,LHRS(B,1,T),LHRS(B,2,T),
  + LHRS(B,3,T),LHRS(B,4,T),LHRS(B,5,T),RC
  MODES = MODES + 1
ENDIF
IF (RP .GT. 0.0) THEN
  CHAR = 'PUSH'
  WRITE (75,8820) CHAR,DRHS(B,1,T),DRHS(B,2,T),
  + DRHS(B,3,T),DRHS(B,4,T),DRHS(B,5,T),RP
  MODES = MODES + 1
ENDIF
IF (MODES .GT. 1) THEN
  CHAR = 'HOURS'
  WRITE (75,8820) CHAR,HRS(B,1,T),HRS(B,2,T),
  + HRS(B,3,T),HRS(B,4,T),HRS(B,5,T),BTHRSCB,T)
ENDIF
FORMATC21X,A5,6ClX,F7.l))
CONTINUE
RHLTOT = RHLTOT + RHLB
RLDTOT = RLDTOT + RLDL
RDZTOT = RDZTOT + RDZB
CHAR = 'TOTAL'
WRITE(75,8810) BLOCKCBJ,CHAR
CHAR = 'HAUL'
WRITE(75,8820) CHAR,RHL(1),RHL(2),RHL(3),RHL(4),RHL(5),RHLB
CHAR = 'CARRY'
WRITE(75,8820) CHAR,RLD(1),RLD(2),RLD(3),RLD(4),RLD(5),RLDB
CHAR = 'PUSH'
WRITE(75,8820) CHAR,DRDZ(1),RDZ(2),RDZ(3),RDZ(4),RDZ(5),RDZB
CHAR = 'HOURS'
WRITE(75,8820) CHAR,BLHRSCB,1J,BLHRSCB,2J,BLHRSCB,3),
  + BLHRSCB,4J,BLHRSCB,5J,BHRSCB)
CONTINUE
WRITE(75,8960)
FORMATC5X,'TOTAL JOB')
WRITE(75,8970) HLRSTOT CHAR
CHAR = 'HAUL'
WRITE(75,8970) CHAR,RHLTOT CHAR = 'CARRY'
WRITE(75,8970) CHAR,RLDTOT CHAR = 'PUSH'
WRITE(75,8970) CHAR,RDZTOT CHAR = 'HOURS'
WRITE(75,8970) CHAR,TOTHRS
FORMATC21X,A5,38X,Fl0.l))
JE
OUTPUT BY SOURCE BLOCK - CALCULATED DIRECTLY FROM INPUTS
COMPARE TO INPUT VOLUME*SWELL TO CALCULATE RATIOS
WRITE(72,9010)
FORMAT(25X,'LOOSE CUBIC YARD VOLUMES')
WRITE(72,9011) DAYONE,DAYEND JOB
FORMAT (10X,A15, ' TRU ',A15,3X,A15)
WRITE(72,9012)
FORMAT(17X,'ESTIMATES BASED ON HOURS AND RATES INPUTS')
WRITE(72,9020)
FORMAT(1X,'FROM',6X,'TO',5X,'VIA',5X,'FROM',4X,'% OF')
WRITE(72,9030)
FORMAT(1X,'BLOCK',4X,'BLOCK',2X,7X,'T LIFT 2 LIFT 3 LIFT X LIFT
+ R LIFT TOTAL TOTAL')
WRITE (72,*)
MLT = 0
LDT = 0
DZT = 0
DO 9250 B = 1,BDIM
  IF (B(HRSCB) .EQ. 0) GOTO 9250
HLB = 0
LDB = 0
DZB = 0
DO 9090 L = 1, LDIM
   HL(L) = 0
   LD(L) = 0
   DZ(L) = 0
9090 CONTINUE
DO 9200 T = 1, TDIM
   IF (BYDSC(B,T) .EQ. 0) GOTO 9200
   H = 0
   C = 0
   P = 0
   MODES = 0
   DO 9100 L = 1, LDIM
      IF (YDS(B,L,T) .EQ. 0) GOTO 9100
      H = H + HYDS(B,L,T)
      C = C + LYDS(B,L,T)
      P = P + DYDSC(B,L,T)
      HL(L) = HL(L) + HYDS(B,L,T)
      LD(L) = LD(L) + LYDS(B,L,T)
      DZ(L) = DZ(L) + DYDSC(B,L,T)
9100 CONTINUE
   HLB = HLB + H
   LDB = LDB + C
   DZB = DZB + P
   WRITE (72,9110) BLOCK(B), BLOCK(T)
9110 FORMAT(1X,A5,1X,'TO',2X,A5)
   IF (H.GT.0) THEN
      HFRAC = (H*100.0)/BYDSC(B)
      CHAR = 'HAUL'
      WRITE (72,9120) CHAR, HYDS(B,1,T), HYDS(B,2,T),
                      HYDS(B,3,T), HYDS(B,4,T), HYDS(B,5,T), H, HFRAC
      MODES = MODES + 1
   ENDIF
   IF (C.GT.0) THEN
      CFRAC = (C*100.0)/BYDSC(B)
      CHAR = 'CARRY'
      WRITE (72,9120) CHAR,LYDS(B,1,T),LYDS(B,2,T),
                      LYDS(B,3,T),LYDS(B,4,T),LYDS(B,5,T),C,CFRAC
      MODES = MODES + 1
   ENDIF
   IF (P.GT.0) THEN
      PFRAC = (P*100.0)/BYDSC(B)
      CHAR = ‘PUSH’
      WRITE (72,9120) CHAR,DYDSC(B,1,T),DYDSC(B,2,T),
                      DYDSC(B,3,T),DYDSC(B,4,T),DYDSC(B,5,T),P,PFRAC
      MODES = MODES + 1
   ENDIF
   IF (MODES.GT.1) THEN
      FRAC = (BYDSC(B,T)*100.0)/BYDSC(B)
      CHAR = ‘MOVED’
      WRITE (72,9120) CHAR,YDS(B,1,T),YDS(B,2,T),
                      YDS(B,3,T),YDS(B,4,T),YDS(B,5,T),FRAC
   ENDIF
9120 CONTINUE
   HTOT = HLB + HLD
   LDTOT = LDB + LDB
   DZTOT = DZB + DZB
   HFRAC = (HLB*100.0)/BYDSC(B)
   CFRAC = (LDB*100.0)/BYDSC(B)
   PFRAC = (DZB*100.0)/BYDSC(B)
   FRAC = HFRAC + CFRAC + PFRAC
   CHAR = ‘TOTAL’
   WRITE(72,9110) BLOCK(B), CHAR
   CHAR = ‘HAUL’
   WRITE(72,9120) CHAR, HL(1), HL(2), HL(3), HL(4), HL(5), HLB, HFRAC
   CHAR = ‘CARRY’
   WRITE(72,9120) CHAR, LD(1), LD(2), LD(3), LD(4), LD(5), LDB, CFRAC
CHAR = 'PUSH'
WRITE(72,9120) CHAR,DZ(1),DZ(2),DZ(3),DZ(4),DZ(5),DZB,PFRAC
CHAR = 'MOVED'
WRITE(72,9120) CHAR,BLYDS(B,1),BLYDS(B,2),BLYDS(B,3),
+ BLYDS(B,4),BLYDS(B,5),BYDSCB,FRAC
CHAR = 'TOPO'
WRITE(72,9120) CHAR,LIFVOL(B,1),LIFVOL(B,2),LIFVOL(B,3),
+ LIFVOL(B,4),LIFVOL(B,5),BLKVOLCB)
DO 9225 L = 1,5
IF (LIFVOL(B,L) .EQ. 0) THEN
  RATIO(L) = 1.0
ELSE
  RATIO(L) = REAL(BLYDS(B,L))/REAL(LIFVOL(B,L))
ENDIF
9225 CONTINUE
IF (BLKVOLCB) .EQ. 0) THEN
  BRATIO = 0.0
ELSE
  BRATIO = REAL(BYDSCB))/REAL(BLK VOLCB)
ENDIF
WRITE(72,9230) RATIO(1),RATIO(2),RATIO(3),RATIO(4),
+ RATIO(5),BRATIO
9230 FORMAT(12X,'MOVED:TOPO:',6(2X,F6.2))
WRITE(72,*)
9250 CONTINUE
WRITE(72,9260)
9260 FORMAT(1X,'TOTAL JOB')
WRITE(72,*)
HFRAC = (HLTOT*100.0)/TOTYDS
CFRAC = (LDTOT*100.0)/TOTYDS
PFRA C = (DZTOT*100.0)/TOTYDS
FRAC = HFRAC + CFRAC + PFRAC
CHAR = 'HAUL'
WRITE(72,9270) CHAR,HLTOT,HFRAC
CHAR = 'CARRY'
WRITE(72,9270) CHAR,LDTOT,CFRAC
CHAR = 'PUSH'
WRITE(72,9270) CHAR,DZTOT,PFRA C
CHAR = 'MOVED'
WRITE(72,9270) CHAR,TOTYDS,FRAC
CHAR = 'TOPO'
WRITE(72,9271) CHAR,TOTVOL
9270 FORMAT(17X,A5,38X,10,F8.1,'%')
9271 FORMAT(17X,A5,38X,10)
JRATIO = REAL(TOTYDS)/REAL(TOTVOL)
WRITE(72,9280) JRATIO
9280 FORMAT(12X,'MOVED:TOPO:',40X,F8.2)
*
*  USE RATIOS TO ADJUST __YDS ARRAYS
*
NEWTOT = 0
DO 9330 B = 1,BDIM
  NEWVOL(B) = 0
DO 9320 T = 1,TDIM
  BTYDS(B,T) = 0
DO 9310 L = 1,LDIM
  IF (RATIO(B,L) .GT. 0.01) THEN
    X = HYDS(B,L,T)/RATIO(B,L)
    Y = LYDS(B,L,T)/RATIO(B,L)
    Z = DYDS(B,L,T)/RATIO(B,L)
    HYDS(B,L,T) = IFIX(X)
    LYDS(B,L,T) = IFIX(Y)
    DYDS(B,L,T) = IFIX(Z)
    YDS(B,L,T) = HYDS(B,L,T) + LYDS(B,L,T) + DYDS(B,L,T)
  NEWVOL(B) = NEWVOL(B) + YDS(B,L,T)
  BTYDS(B,T) = BTYDS(B,T) + YDS(B,L,T)
ENDIF
9310 CONTINUE
9320 CONTINUE
NEWTOT = NEWTOT + NEWVOL(B)
CONTINUE

TOTAL BCY HAUL, CARRY, PUSH VOLUMES BY BLOCK-LIFT OF ORIGIN
RECALCULATE LIFT VOLUMES AS BANK CUBIC YARDS

DO 9370 B = 1, BDIM
   DO 9360 L = 1, LDIM
      IF (LIFVOL(B, L) .EQ. 0) GOTO 9360
      DO 9350 T = 1, TDIM
         Haul(B, L) = Haul(B, L) + HYDSC(B, L, T)
         Carry(B, L) = Carry(B, L) + LYDSC(B, L, T)
         Push(B, L) = Push(B, L) + DYDSC(B, L, T)
      CONTINUE
      IF (XSHELL(B, L) .EQ. 0.000) THEN
         WRITE (73, 9355) B, L, XSHELL(B, L)
         DO 9355 FORMT ('XSHELL(', 'II', ',', 'II', 'I) = ', F6.4)
            GOTO 9360
      ENDIF
      W = LIFVOL(B, L) / XSHELL(B, L) + 0.5
      X = Haul(B, L) / XSHELL(B, L) + 0.5
      Y = Carry(B, L) / XSHELL(B, L) + 0.5
      Z = Push(B, L) / XSHELL(B, L) + 0.5
      LIFVOL(B, L) = IFIX(W)
      Haul(B, L) = IFIX(X)
      Carry(B, L) = IFIX(Y)
      Push(B, L) = IFIX(Z)
   CONTINUE
CONTINUE

OUTPUT BY DESTINATION BLOCK
MOVED VOLUMES ADJUSTED VIA MOVE:TOPO RATIOS

WRITE(73, 9010)
WRITE(73, 9011) DAYONE, DAYEND, JOB
WRITE(73, 9376)

9350 FORMAT('FROM', 'TO', 'VIA', 'FROM', 'TO')
WRITE(73, 9377)

9355 FORMAT('FROM', 'TO', 'VIA', 'FROM', 'TO')
WRITE(73, 9378)

9370 CONTINUE

DO 9370 B = 1, BDIM
   DO 9360 L = 1, LDIM
      IF (LIFVOL(B, L) .EQ. 0) GOTO 9360
      DO 9350 T = 1, TDIM
         HLT = HLT + H
         LDT = LDT + C
         DZT = DZT + P
         WRITE (73, 9110) BLOCK(B), BLOCK(T)
         IF (H .GT. 0) THEN
            CHAR = 'HAUL'
         ENDIF
      CONTINUE
   CONTINUE

CONTINUE
HLT = HLT + H
LDT = LDT + C
DZT = DZT + P
WRITE (73, 9110) BLOCK(B), BLOCK(T)
IF (H .GT. 0) THEN
   CHAR = 'HAUL'
ENDIF
WRITE (73, 9120) CHAR, HYDSCB(1, T), HYDSCB(2, T),
+ HYDSCB(3, T), HYDSCB(4, T), HYDSCB(5, T), P
MODES = MODES + 1
ENDIF
IF (C .GT. 0) THEN
CHAR = 'CARRY'
WRITE (73, 9120) CHAR, LYDSCB(1, T), LYDSCB(2, T),
+ LYDSCB(3, T), LYDSCB(4, T), LYDSCB(5, T), C
MODES = MODES + 1
ENDIF
IF (P .GT. 0) THEN
CHAR = 'PUSH'
WRITE (73, 9120) CHAR, DYDSCB(1, T), DYDSCB(2, T),
+ DYDSCB(3, T), DYDSCB(4, T), DYDSCB(5, T), P
MODES = MODES + 1
ENDIF
IF (MODES .GT. 1) THEN
CHAR = 'TOTAL'
WRITE (73, 9120) CHAR, YDSCB(1, T), YDSCB(2, T),
+ YDSCB(3, T), YDSCB(4, T), YDSCB(5, T), BTYDSCB(T)
ENDIF
9500 CONTINUE
HLTOT = HLT + HLT
LDTOT = LDT + LDT
DZTOT = DZTOT + DZT
CHAR = 'TOTAL'
WRITE (73, 9110) CHAR, BLOCK(T)
CHAR = 'HAUL'
WRITE (73, 9270) CHAR, HLT
CHAR = 'CARRY'
WRITE (73, 92270) CHAR, LDT
CHAR = 'PUSH'
WRITE (73, 9270) CHAR, DZT
CHAR = 'MOVED'
TYDSC(T) = HLT + LDT + DZT
WRITE (73, 9270) CHAR, TYDSC(T)
WRITE (73, *)
9550 CONTINUE
WRITE (73, 9260)
CHAR = 'HAUL'
WRITE (73, 9270) CHAR, HLT
CHAR = 'CARRY'
WRITE (73, 9270) CHAR, LDT
CHAR = 'PUSH'
WRITE (73, 9270) CHAR, DZT
TOTYDS = HLTOT + LDTOT + DZTOT
CHAR = 'MOVED'
WRITE (73, 9270) CHAR, TOTYDS
WRITE (73, *)
*
FILE 74 = BCY VOLUMES OUTPUT: TOTAL, HAUL, CARRY, PUSH
*
WRITE (74, 9555)
9555 FORMAT ('VOL')
DO 9590 B = 1, BDIM
DO 9580 L = 1, LDIM
IF (LIFVOL(B, L) .EQ. 0) GOTO 9580
WRITE (74, 9570) BLOCK(B), LIFT(L), LIFVOL(B, L), HAUL(B, L),
+ CARRY(B, L), PUSH(B, L)
9570 FORMAT (4X, A4, 1X, A1, 4X, 1X, 19))
9580 CONTINUE
9590 CONTINUE
*
FILE 74 = INPUT FOR RECLAMATION COST REDISTRIBUTION: COST FORTRAN
*
9600 WRITE (74, 9601)
9601 FORMAT ('RCL')
DO 9625 T = 1, TDIM
TYDSC(T) = 0
DO 9620 L = 1, LDIM
DO 9615 B = 1, BDIM
  TYDS(T) = TYDS(T) + YDS(B, L, T)
9615 CONTINUE
9620 CONTINUE
9625 CONTINUE

* 9630 DO 9639 T = 1, TDIM
    DO 9630 L = 1, LDIM
      DO 9670 B = 1, BDIM
        IF (TYDS(T) .LE. 0) GOTO 9670
        IX = YDS(B, L, T)
        IY = TYDS(T)
        X = FLOAT(IX)
        Y = FLOAT(IY)
        FRAC = X / Y
        IF (FRAC .GE. .0001) THEN
          WRITE (74, 9640) BLOCK(T), FRAC, BLOCK(B), LIFT(L)
          FORMAT (4X, A4, 1X, F8.6, 1X, A4, 1X, A4)
        ENDIF
      CONTINUE
    CONTINUE
  CONTINUE
9640 WRITE (74, 9692)
9650 CONTINUE
9660 CONTINUE
9670 WRITE (74, 9692)
9680 WRITE (74, 8010) DAYONE
9690 WRITE (74, 8010) DAYEND
9700 WRITE (74, 8010) JOB
9710 GOTO 9999

* 9700 BLOCK
* ANYTIME CHARACTERS ARE ENTERED IN COLUMNS 1-3 OF INPUT
* RECORD, PROGRAM IS ROUTED HERE
*
9700 IF (GO .EQ. 'HLR') GOTO 1000
9710 IF (GO .EQ. 'HLP') GOTO 1500
9720 IF (GO .EQ. 'LDA') GOTO 2000
9730 IF (GO .EQ. 'LDP') GOTO 2500
9740 IF (GO .EQ. 'DZA') GOTO 3000
9750 IF (GO .EQ. 'DZP') GOTO 3500
9760 IF (GO .EQ. 'VOL') GOTO 7000
9770 IF (GO .EQ. 'END') GOTO 8000
9780 WRITE (06, X) 'GO ERROR EXITING 9700 BLOCK'
9790 GOTO 9998

* ERROR MESSAGES
*
9905 WRITE (06, X) 'READ ERROR STATEMENT 70'
9910 GOTO 9999
9915 WRITE (06, X) 'READ ERROR STATEMENT 1000'
9920 GOTO 9999
9925 WRITE (06, X) 'READ ERROR STATEMENT 10000'
9930 GOTO 9999
9935 WRITE (06, X) 'READ ERROR STATEMENT 25000'
9940 GOTO 9999
9945 WRITE (06, X) 'READ ERROR STATEMENT 250000'
9950 GOTO 9999
9955 WRITE (06, X) 'READ ERROR STATEMENT 70000'
9960 GOTO 9999
9965 WRITE (06, X) 'IMAX EXCEEDED; ICOUNT = ', ICOUNT,
9970 GOTO 9999
9975 WRITE (06, X) 'FINISHED ', ICOUNT, ' RECORDS PROCESSED'
9999 STOP
END
## BYSOURCE MOVE

### LOOSE CUBIC YARD VOLUMES

**DAY 1 1999 THRU DAY 11 1999 PONY RUN**

**ESTIMATES BASED ON HOURS AND RATES INPUTS**

<table>
<thead>
<tr>
<th>FROM BLOCK</th>
<th>TO BLOCK</th>
<th>VIA T LIFT</th>
<th>2 LIFT</th>
<th>3 LIFT</th>
<th>X LIFT</th>
<th>R LIFT</th>
<th>TOTAL</th>
<th>% OF TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1 TO SD1</td>
<td>HAUL</td>
<td>8996</td>
<td>7948</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16944</td>
<td>97.7%</td>
</tr>
<tr>
<td></td>
<td>CARRY</td>
<td>0</td>
<td>400</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>2.3%</td>
</tr>
<tr>
<td></td>
<td>MOVED</td>
<td>8996</td>
<td>8348</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17344</td>
<td>100.0%</td>
</tr>
<tr>
<td>MB1 TO TOTAL</td>
<td>HAUL</td>
<td>8996</td>
<td>7948</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16944</td>
<td>97.7%</td>
</tr>
<tr>
<td></td>
<td>CARRY</td>
<td>0</td>
<td>400</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>2.3%</td>
</tr>
<tr>
<td></td>
<td>PUSH</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>MOVED</td>
<td>8996</td>
<td>8348</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17344</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>MOVED:TOPO</td>
<td>0.94</td>
<td>1.19</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.04</td>
<td></td>
</tr>
<tr>
<td>MB2 TO SD2</td>
<td>HAUL</td>
<td>2400</td>
<td>3448</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5848</td>
<td>80.7%</td>
</tr>
<tr>
<td></td>
<td>PUSH</td>
<td>1400</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1400</td>
<td>19.3%</td>
</tr>
<tr>
<td></td>
<td>MOVED</td>
<td>3800</td>
<td>3448</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7248</td>
<td>100.0%</td>
</tr>
<tr>
<td>MB2 TO TOTAL</td>
<td>HAUL</td>
<td>2400</td>
<td>3448</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5848</td>
<td>80.7%</td>
</tr>
<tr>
<td></td>
<td>CARRY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>PUSH</td>
<td>1400</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1400</td>
<td>19.3%</td>
</tr>
<tr>
<td></td>
<td>MOVED</td>
<td>3800</td>
<td>3448</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7248</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>TOPO</td>
<td>4800</td>
<td>3500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8300</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MOVED:TOPO</td>
<td>0.79</td>
<td>0.99</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.87</td>
<td></td>
</tr>
<tr>
<td>MB3 TO SD1</td>
<td>HAUL</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>900</td>
<td>51.7%</td>
</tr>
<tr>
<td>MB3 TO SD2</td>
<td>HAUL</td>
<td>840</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>840</td>
<td>48.3%</td>
</tr>
<tr>
<td>MB3 TO TOTAL</td>
<td>HAUL</td>
<td>1740</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1740</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>CARRY</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>PUSH</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>MOVED</td>
<td>1740</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1740</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>TOPO</td>
<td>1560</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1560</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MOVED:TOPO</td>
<td>1.12</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.12</td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL JOB**

<table>
<thead>
<tr>
<th>HAUL</th>
<th>24532</th>
<th>93.2%</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARRY</td>
<td>400</td>
<td>1.5%</td>
</tr>
<tr>
<td>PUSH</td>
<td>1400</td>
<td>5.3%</td>
</tr>
<tr>
<td>MOVED</td>
<td>26332</td>
<td>100.0%</td>
</tr>
<tr>
<td>TOPO</td>
<td>26460</td>
<td></td>
</tr>
<tr>
<td>MOVED:TOPO</td>
<td></td>
<td>1.00</td>
</tr>
</tbody>
</table>

## BYDEST MOVE

### LOOSE CUBIC YARD VOLUMES

**DAY 1 1999 THRU DAY 11 1999 PONY RUN**

**MOVED VOLUMES ADJUSTED VIA MOVE:TOPO RATIOS**

<table>
<thead>
<tr>
<th>FROM BLOCK</th>
<th>TO BLOCK</th>
<th>VIA T LIFT</th>
<th>2 LIFT</th>
<th>3 LIFT</th>
<th>X LIFT</th>
<th>R LIFT</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1 TO SD1</td>
<td>HAUL</td>
<td>9600</td>
<td>6664</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16264</td>
</tr>
<tr>
<td></td>
<td>CARRY</td>
<td>0</td>
<td>335</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>335</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>9600</td>
<td>6999</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16599</td>
</tr>
<tr>
<td>MB3 TO SD1</td>
<td>HAUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CARRY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SD1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>TOTAL TO</td>
<td></td>
<td>HAUL 806</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HAUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17070</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CARRY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>335</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PUSH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MOVED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18244</td>
</tr>
<tr>
<td>MB2 TO</td>
<td></td>
<td>HAUL 3031</td>
<td>3500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SD2</td>
<td></td>
<td>PUSH 1768</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1768</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TOTAL 4799</td>
<td>3500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>MB3 TO</td>
<td></td>
<td>HAUL 753</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SD2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL TO</td>
<td></td>
<td>HAUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7284</td>
</tr>
<tr>
<td>SD2</td>
<td></td>
<td>CARRY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PUSH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1768</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MOVED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8088</td>
</tr>
<tr>
<td>TOTAL JOB</td>
<td></td>
<td>HAUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24354</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CARRY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>335</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PUSH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1768</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MOVED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26457</td>
</tr>
</tbody>
</table>

```
* * * * COSINPUT MOVE * * * *

| VOL | MB1 T | 8000  | 8000  | 0      | 0      |
|     | MB1 2 | 5000  | 4760  | 239    | 0      |
|     | MB2 T | 4000  | 2526  | 0      | 1473   |
|     | MB2 2 | 2500  | 2500  | 0      | 0      |
|     | MB3 T | 1200  | 1199  | 0      | 0      |

| RCL | SD1 0.551566 MB1 T |
|     | SD1 0.046309 MB3 T |
|     | SD1 0.402126 MB1 2 |
|     | SD2 0.530159 MB2 T |
|     | SD2 0.083186 MB3 T |
|     | SD2 0.386655 MB2 2 |

END

DAY 1 1999
DAY 11 1999
PONY RUN
APPENDIX G: COST

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th>DATA</th>
<th>COS</th>
<th>INPUT</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>75.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>100.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>80.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>70.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>70.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>40.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>70.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>20.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>50.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>50.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>40.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>30.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>35.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>70.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>8.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>3.001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>8.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>1.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>20.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>HLR</strong></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1</td>
<td>SD1</td>
<td>42.0</td>
<td>32.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>74.0</td>
<td>SD1</td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>SD2</td>
<td>10.0</td>
<td>14.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>24.0</td>
<td>SD2</td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>SD1</td>
<td>4.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>6.0</td>
<td>SD1</td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>SD2</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
<td>SD2</td>
<td></td>
</tr>
<tr>
<td>MB3</td>
<td>SD1</td>
<td>11.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>11.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>LDA</strong></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1</td>
<td>L2</td>
<td>21.0</td>
<td>16.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>37.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>CD</td>
<td>0.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td></td>
<td>21.0</td>
<td>18.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>39.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>L1</td>
<td>0.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>L2</td>
<td>5.0</td>
<td>5.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>10.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td></td>
<td>5.0</td>
<td>9.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>14.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB3</td>
<td>L2</td>
<td>5.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB3</td>
<td></td>
<td>5.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>LDC</strong></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1</td>
<td>LC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
<td>0.0</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>PA</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.0</td>
<td>0.0</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
<td>0.0</td>
<td>7.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>LC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
<td>0.0</td>
<td>7.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
<td>0.0</td>
<td>7.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DZC</strong></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SD1</td>
<td>RR</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
<td>4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD1</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.0</td>
<td>4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD2</td>
<td>RR</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.0</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD2</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.0</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>DZA</strong></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1</td>
<td>ST</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

103
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1</td>
<td>PD</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>MB1</td>
<td>FO</td>
<td>10.0</td>
<td>8.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>18.0</td>
</tr>
<tr>
<td>MB1</td>
<td>DS</td>
<td>4.0</td>
<td>4.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>8.0</td>
</tr>
<tr>
<td>MB1</td>
<td>FC</td>
<td>0.0</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.0</td>
</tr>
<tr>
<td>MB1</td>
<td></td>
<td>15.0</td>
<td>14.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>29.0</td>
</tr>
<tr>
<td>MB2</td>
<td>ST</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>MB2</td>
<td>PD</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>MB2</td>
<td>FO</td>
<td>7.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
</tr>
<tr>
<td>MB2</td>
<td>PO</td>
<td>7.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.0</td>
</tr>
<tr>
<td>MB2</td>
<td>DS</td>
<td>3.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.0</td>
</tr>
<tr>
<td>MB2</td>
<td></td>
<td>18.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>18.0</td>
</tr>
<tr>
<td>MB3</td>
<td>ST</td>
<td>5.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
</tr>
<tr>
<td>MB3</td>
<td></td>
<td>5.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
</tr>
<tr>
<td>SD1</td>
<td>RR</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
</tr>
<tr>
<td>SD1</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
</tr>
<tr>
<td>SD2</td>
<td>RR</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
</tr>
<tr>
<td>SD2</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.0</td>
</tr>
<tr>
<td>DRL</td>
<td>MB1</td>
<td>T</td>
<td>100</td>
<td>2000</td>
<td>20.0</td>
<td>10000</td>
<td>14.0</td>
</tr>
<tr>
<td>DRL</td>
<td>MB1</td>
<td>X</td>
<td>100</td>
<td>1500</td>
<td>15.0</td>
<td>5000</td>
<td>14.0</td>
</tr>
<tr>
<td>DRL</td>
<td>MB1</td>
<td>200</td>
<td>3500</td>
<td>17.5</td>
<td>15000</td>
<td>28.0</td>
<td>14.0</td>
</tr>
<tr>
<td>DRL</td>
<td>MB2</td>
<td>T</td>
<td>50</td>
<td>1000</td>
<td>20.0</td>
<td>5000</td>
<td>7.0</td>
</tr>
<tr>
<td>DRL</td>
<td>MB2</td>
<td>X</td>
<td>50</td>
<td>750</td>
<td>15.0</td>
<td>2500</td>
<td>7.0</td>
</tr>
<tr>
<td>DRL</td>
<td>MB2</td>
<td>100</td>
<td>1750</td>
<td>17.5</td>
<td>7500</td>
<td>14.0</td>
<td>7.0</td>
</tr>
<tr>
<td>MIS</td>
<td>GRD ROAD</td>
<td>2</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIS</td>
<td>MAC MB2</td>
<td>7</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIS</td>
<td>AUG MB2</td>
<td>3</td>
<td>6 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIS</td>
<td>SED SD1</td>
<td>5</td>
<td>4 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIS</td>
<td>SED SD2</td>
<td>2</td>
<td>3 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIS</td>
<td>MAN SD1</td>
<td>5</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIS</td>
<td>MAN SD2</td>
<td>2</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL</td>
<td>MB1</td>
<td>1060</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL</td>
<td>MB2</td>
<td>570</td>
<td>200</td>
<td>22.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL</td>
<td>MB2</td>
<td>300</td>
<td>AUGER</td>
<td>0</td>
<td>18.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOL</td>
<td>MB1</td>
<td>T</td>
<td>8000</td>
<td>8000</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>VOL</td>
<td>MB1</td>
<td>2</td>
<td>5000</td>
<td>4760</td>
<td>239</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>VOL</td>
<td>MB2</td>
<td>T</td>
<td>4000</td>
<td>2526</td>
<td>0</td>
<td>1473</td>
<td></td>
</tr>
<tr>
<td>VOL</td>
<td>MB2</td>
<td>2</td>
<td>2500</td>
<td>2500</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>VOL</td>
<td>MB3</td>
<td>T</td>
<td>1200</td>
<td>1199</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SPB</td>
<td>OHD ROAD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCL</td>
<td>SD1</td>
<td>0.561566</td>
<td>MB1</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCL</td>
<td>SD1</td>
<td>0.046309</td>
<td>MB3</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCL</td>
<td>SD1</td>
<td>0.042126</td>
<td>MB1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCL</td>
<td>SD2</td>
<td>0.30159</td>
<td>MB2</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCL</td>
<td>SD2</td>
<td>0.083186</td>
<td>MB3</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCL</td>
<td>SD2</td>
<td>0.386655</td>
<td>MB2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

END

DAY 1 1999
DAY 11 1999
PONY RUN

* * * * COST FORTRAN * * * *

PROGRAMMED BY C. E. ZIPPER, 1985

DOCUMENTATION: VIRGINIA AGR. EXP. STATION BULLETIN NO. 86-1
VPI & SU, BLACKSBURG 24061
PROGRAM COST

THE PURPOSE OF THIS PROGRAM IS TO TOTAL COSTS FOR SURFACE MINING OPERATIONS - OVERHEAD DISTRIBUTED PER CUBIC YARD

INTEGER C,B,L,CC,I,J
INTEGER SB,SL,DB,DL
INTEGER BDIM,CDIM,LDIM,CCDIM
INTEGER ICOUNT, IMAX
INTEGER HOLES, FEET, ANFO
INTEGER H, HTOT, HAUL(22,5)
INTEGER CA, CTOT, CARRY(22,5)
INTEGER P, PTOT, PUSH(22,5)
INTEGER IHERS, AUGHERS
INTEGER COAL, BLKCOL(22), AUGCOL(22), TCOAL, ACOAL, SCOAL
INTEGER COCOHL, TOCOHL, SECOHL
INTEGER VOLUME, LIFVOL(22,5), BLKVOL(22), TVOL, V
REAL COST(23), PREOHD, STRPRI(22), PRICE, AUGPRI(22)
REAL LIFHRS(5), BLKHRS, HRS, SHTHRS, HRSSHT
REAL HRSHLR, HRSLDA, HRSDB, HRSLD, HRSDBA, HRSDBC, HRSDBD, HRSDBE
REAL LIKOST(22,5,15), BLKOST(22,15), CCCOST(15), BLKTOT(22)
REAL LIFTOT(22,5), RCLEXP(22), RCLDPX(22)
REAL HLCOST, LDCOST, DZCOST, DRCOST, MICOST, SMCOST
REAL LICOST, BLCOST, X, Y, T, TOT, W, WO
REAL OHDTOT, OHDPKY
REAL BLHAUL(22,5), BHAUL(22)
REAL BLCARP(22,5), BCARP(22)

LOGICAL FIRST
CHARACTER*1 LIFT(5), LIF, SOLIFT, DELIFT, R
CHARACTER*2 OPER
CHARACTER*3 GO, MACHIN
CHARACTER*4 BLOCK(22), BLK, DES, LOC, DEBLK, SOBLK
CHARACTER*5 AUGER
CHARACTER*15 DAYONE, DAYEND
CHARACTER*20 COITEM(23), COCATE(15), CHAR, JOB

This program accumulates costs in arrays per item costs are read by 1000 block and put into array "COST". Subscript 'C' corresponds to costs in 'COST' array:

COITEM(01) = 'HAULER PER HOUR'
COITEM(02) = 'A LOADER PER HOUR'
COITEM(03) = 'B LOADER PER HOUR'
COITEM(04) = 'COAL LOADER PER HOUR'
COITEM(05) = 'A DOZER PER HOUR'
COITEM(06) = 'C DOZER PER HOUR'
COITEM(07) = 'DRILL PER HOUR'
COITEM(08) = 'ANFO PER LB.'
COITEM(09) = 'DRILL/BLAST PER HOLE'
COITEM(11) = 'LABOR PER HOUR'
COITEM(12) = 'COAL TRUCK PER HOUR'
COITEM(13) = 'WATER TRUCK PER HOUR'
COITEM(14) = 'GRADER PER HOUR'
COITEM(15) = 'SEEDER PER HOUR'
COITEM(16) = 'BACKHOE PER HOUR'
COITEM(17) = 'AUGER PER HOUR'
COITEM(18) = 'FOREMAN LABOR HOUR'
COITEM(19) = 'OVERHEAD PER TON'
COITEM(20) = 'TEMPORARY LABOR HOUR'
COITEM(21) = 'CONTRACT COAL HAUL'
COITEM(22) = 'COAL REVENUE PER TON'
COITEM(23) = 'SEEDING / BATCH, ACRE'

Costs are totaled by cost category (array subscript 'C') as follows:

105
COCAT(01) = 'CLEAR AND BENCH'
COCAT(02) = 'DRILL AND SHOOT'
COCAT(03) = 'CARRY AND PUSH'
COCAT(04) = 'LOAD & HAUL: LOADERS'
COCAT(05) = 'LOAD & HAUL: DOZERS'
COCAT(06) = 'LOAD & HAUL: HAULERS'
COCAT(07) = 'LOAD & HAUL: DUMPsite'
COCAT(08) = 'LOAD & HAUL: TOTAL'
COCAT(09) = 'COAL: CLEAN & LOAD'
COCAT(10) = 'COAL: AUGER'
COCAT(11) = 'COAL: HAULING'
COCAT(12) = 'RECLAMATION - DESTIN'
COCAT(13) = 'OTHER'
COCAT(14) = 'OVERHEAD PER TON'
COCAT(15) = 'RECLAMATION - SOURCE'

* * *

COSTS ARE ALSO TOTALED BY BLOCK (SUBSCRIPT 'B') AND LIFT (SUB. 'L')

* * *

EACH INPUT BLOCK IS PRECEDED BY A CHARACTER×3 VARIABLE

'GO' IN COLS 1-3. IF A READ STATEMENT ENCOUNTERS A NON-BLANK

'GO', PROGRAM CONTROL GOES TO 9700 BLOCK. FINAL 'GO' IS 'END',

* * *

INITIALIZE VARIABLES

BDIM = 22
CDIM = 25
LDIM = 5
CCDIM = 15
CALL ABLOCK (BLOCK,BDIM)
CALL ALIFT (LIFT,LDIM)
DO 30 B = 1,BDIM
BLKCOL(B) = 0
BLKVOL(B) = 0
AUGCOL(B) = 0
BLKTOT(B) = 0.0
RCLEXP(B) = 0.0
RCLDXP(B) = 0.0
DO 20 L = 1,LDIM
HAUL(B,L) = 0
CARRY(B,L) = 0
PUSH(B,L) = 0
LIFVOL(B,L) = 0
LIFTOT(B,L) = 0.0
DO 10 CC = 1,CCDIM
LIKOST(B,L,CC) = 0.0
CONTINUE 20
CONTINUE 30
CONTINUE
DO 50 B = 1,BDIM
DO 40 CC = 1,CCDIM
BLKOST(B,CC) = 0.0
CONTINUE 40
CONTINUE 50
CONTINUE
DO 60 CC = 1,CCDIM
CCCOST(CC) = 0.0
CONTINUE 60
CONTINUE
HLHST = 0.0
LDHST = 0.0
DZHST = 0.0
DHST = 0.0
MICST = 0.0
SHST = 0.0
TCOAL = 0
ACOAL = 0
CCOHL = 0
TCOHL = 0
HTOT = 0
CTOT = 0
PTOT = 0
TVOL = 0
HRSHLR = 0.0
HRSLDA = 0.0
HRSLDB = 0.0
HRSLDC = 0.0
HRSDZA = 0.0
HRSDZC = 0.0
HRSDRL = 0.0
HRSDRC = 0.0
HRSDRC = 0.0
OHDTOT = 0.0
OHDPCY = 0.0

"I_COUNT" AND "IMAX" FUNCTION AS ENDLESS LOOP PROTECTORS

I_COUNT = 0
IMAX = 1800

FILE 03 DATA COSINPUT
FILE 92 REDIST COST
FILE 93 SUMMARY COST
FILE 94 PERBLOCK COST
FILE 95 PERYARD COST

READ FIRST INPUT RECORD - SHOULD BE COS, COLS. 1 - 3

800 READ(03,990) GO
900 FORMAT (A5)
   IF (GO .EQ. 'COS') GOTO 1000
goto 9900

1000 READ (03,1010,ERR=9901,END=9901) GO, C, COST(C)
   IF (GO .NE. ' ') GOTO 1000
   I_COUNT = I_COUNT + 1
   IF (I_COUNT .GT. IMAX) GOTO 9901
   IF (C .EQ. 19) PREOHD = COST(C)
   IF (C .EQ. 22) THEN
     DO 1020 B = 1,BDIM
        AUGPRI(B) = COST(22)
     STRPRI(B) = COST(22)
   1020 CONTINUE
   ENDIF
   GOTO 1000

FILE 06 DATA IN HLR BLOCK I_COUNT = ',I_COUNT

2000 C = 1
   WRITE (06,*) 'IN HLR BLOCK I_COUNT = ',I_COUNT

2005 READ (03,2010,ERR=9902,END=9902) GO, BLK, DES, R, LIFHRSC1, LIFHRSC2, LIFHRSC3, LIFHRSC4, LIFHRSC5, BLKHS
2010 FORMAT (A5,2X,A4,2X,A4,1X,A1,1X,F6.1,2X,F7.1)

SETUP
   I_COUNT = I_COUNT + 1
   IF (I_COUNT .GT. IMAX) GOTO 9902
   IF (GO .NE. ' ', '') GOTO 9700
   IF (DES .EQ. ' ') GOTO 2005
   IF (BLK .EQ. ' ') GOTO 2005
   CALL KBLOCK (BLOCK,BDIM,BLK,B)
2005 CALL KBLOCK (BLOCK,BDIM,DES,B)
   CC = 12
   ELSE CALL KBLOCK (BLOCK,BDIM,BLK,B)
   CC = 6
   ENDIF

TOTAL
DO 2020 L = 1, LDIM
  LIKOST(B,L,CC) = LIKOST(B,L,CC) + LIFHRS(L) * COST(C)
   HLCOST             = HLCOST + LIFHRS(L) * COST(C)
   HRSHLR             = HRSHLR + LIFHRS(L)
2020 CONTINUE
GOTO 2005

LDA, LDB, OR LDC    OPHOURS LOADER

3000 READ(03,3010,ERR=9903,END=9903) GO,BLK,OPER,LIFHRS(1),LIFHRS(2),
   + LIFHRS(3), LIFHRS(4), LIFHRS(5), BLKHIRS
3010 FORMAT (A3,2X,A4,4X,A2,6C2X,F6.1))    SETUP

   ICONT = ICOUNT + 1
   IF (ICOUNT .GT. IMAX) GOTO 9903
   IF (GO .NE. ' ') GOTO 9700
   IF (BLK .EQ. ' ') GOTO 3000
   IF (OPER .EQ. ' ') GOTO 3000
   CC = 13
3015 IF (OPER .EQ. 'ST') CC = 1
   IF (OPER .EQ. 'PD') CC = 1
   IF (OPER .EQ. 'LI') CC = 4
   IF (OPER .EQ. 'L2') CC = 4
   IF (OPER .EQ. 'CD') CC = 3
   IF (OPER .EQ. 'LC') CC = 9
   IF (OPER .EQ. 'WB') CC = 7
   IF (OPER .EQ. 'DS') CC = 7
   IF (OPER .EQ. 'HP') CC = 01
   IF (OPER .EQ. 'HB') CC = 03
   IF (OPER .EQ. 'HD') CC = 13
   IF (OPER .EQ. 'HT') CC = 13
   IF (OPER .EQ. 'HF') CC = 13
   IF (OPER .EQ. 'CT') CC = 01
   IF (OPER .EQ. 'RR') CC = 12
   IF (OPER .EQ. 'PA') CC = 10
   CALL KBLOCK (BLOCK,BDIM,BLK,B)

CALL KBLOCK (BLOCK,BDIM,BLK,B)    TOTAL

DO 3020 L = 1, LDIM
   LIKOST(B,L,CC) = LIKOST(B,L,CC) + LIFHRS(L) * COST(C)
   LDCOST           = LDCOST + LIFHRS(L) * COST(C)
   IF (CC .EQ. 2) HRSLDA = HRSLDA + LIFHRS(L)
   IF (CC .EQ. 3) HRSLDB = HRSLDB + LIFHRS(L)
   IF (CC .EQ. 4) HRSLDC = HRSLDC + LIFHRS(L)
3020 CONTINUE
GOTO 3000

DZA OR DZC    OPHOURS DOZER

4000 READ(03,4010,ERR=9904,END=9904) GO,BLK,OPER,LIFHRS(1),LIFHRS(2),
   + LIFHRS(3), LIFHRS(4), LIFHRS(5), BLKHIRS
4010 FORMAT (A3,2X,A4,4X,A2,6C2X,F6.1))    SETUP

   ICONT = ICOUNT + 1
   IF (ICOUNT .GT. IMAX) GOTO 9904
   IF (GO .NE. ' ') GOTO 9700
   IF (BLK .EQ. ' ') GOTO 4000
   IF (OPER .EQ. ' ') GOTO 4000
   CC = 13
4015 IF (OPER .EQ. 'ST') CC = 1
   IF (OPER .EQ. 'PD') CC = 1
   IF (OPER .EQ. 'FD') CC = 1
   IF (OPER .EQ. 'FL') CC = 5
   IF (OPER .EQ. 'FC') CC = 3
   IF (OPER .EQ. 'PO') CC = 3
   IF (OPER .EQ. 'LC') CC = 9
   IF (OPER .EQ. 'MB') CC = 7
   IF (OPER .EQ. 'DS') CC = 7
   IF (OPER .EQ. 'HP') CC = 1
   IF (OPER .EQ. 'HB') CC = 3
   IF (OPER .EQ. 'HD') CC = 13
IF (OPER .EQ. 'HT') CC = 13
IF (OPER .EQ. 'HF') CC = 12
IF (OPER .EQ. 'CT') CC = 1
IF (OPER .EQ. 'RR') CC = 12
IF (OPER .EQ. 'PA') CC = 10
CALL KBLOCK (BLOCK,BDIM,BLK,B)

DO 4020 L = 1,LDIM
LIKOST(B,L,CC) = LIKOST(B,L,CC) + LIFHRS(L)*COST(CC)
DZCOST = DZCOST + LIFHRS(L)*COST(CC)
IF (C .EQ. 5) HRSDZA = HRSDZA + LIFHRS(L)
IF (C .EQ. 6) HRSDZC = HRSDZC + LIFHRS(L)
CONTINUE
GOTO 4000

CALL KBLOCK CBLOCK,BDIM,BLK,B)
CALL JLIFT CLIF,L)

* DRL TOTALS

* SETUP

5000 CC = 2
5005 READ (03,5010,ERR=9905,END=9905) GO, BLK, LIF, HOLES, FEET, AVG,
      + ANFO, HRS, SHTHRS
5010 FORMAT (A5,1X,A4,1X,A1,1X,I5,3X,I5,3X,F4.1,2X,I6,2C2X,F5.1))

ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9905
IF (GO .NE. ' ') GOTO 9700
IF (LIF .EQ. ' ') GOTO 5005
IF (BLK .EQ. ' ') GOTO 5005
X = HRS*COST(7) + ANFO*COST(10) + HOLES*COST(8) + SHTHRS*COST(11)
   + FEET*COST(9)
CALL KBLOCK (BLOCK,BDIM,BLK,B)
CALL JLIFT (LIF,L)

LIKOST(B,L,CC) = LIKOST(B,L,CC) + X
DRCOST = DRCOST + HRS*COST(7)
HRSDRL = HRSDRL + HRS
SHCOST = SHCOST + SHTHRS*COST(11)
HRSSHT = HRSSHT + SHTHRS
GOTO 5005

* MIS

READ MISC MACHINE HOURS FROM OTHER TOTALS
* IF MACHIN = 'SED' AUGHRS REPRESENTS SEEDER BATCHES
* OTHER WISE AUGHRS SHOULD EQUAL ZERO
*
6000 READ (03,6010,ERR=9906,END=9906) GO,MACHIN,BLK,IHRS,AUGHRS,CC
6010 FORMAT (A5,2X,A3,2X,A4,4X,I5,2X,I5,2X,I2)

ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9906
IF (GO .NE. ' ') GOTO 9700
IF (MACHIN .EQ. ' ') GOTO 6000
C = 0

6015 IF (MACHIN .EQ. 'MAC') C = 12
IF (MACHIN .EQ. 'MEN') C = 11
IF (MACHIN .EQ. 'MAN') C = 11
IF (MACHIN .EQ. 'WAT') C = 13
IF (MACHIN .EQ. 'GRD') C = 13
IF (MACHIN .EQ. 'SED') C = 15
IF (MACHIN .EQ. 'BAC') C = 16
IF (MACHIN .EQ. 'AUG') C = 17
IF (MACHIN .EQ. 'SUP') C = 18
IF (MACHIN .EQ. 'TEM') C = 20
IF (C .EQ. 0) GOTO 9916
IF (CC .EQ. 9 .OR. CC .EQ. 10 .OR. CC .EQ. 11) THEN
   CC = 15
   WRITE (92,*) 'NO CC GIVEN, MACHINE = ',MACHIN,' GO = MIS'
ENDIF
IF (CC .EQ. 9 .OR. CC .EQ. 10 .OR. CC .EQ. 11) THEN
   L = 4
ELSE
L = 1
ENDIF
Y = 0.0
IF (MACHIN .EQ. 'AUG') Y = AUGHRSLCOST(11)
IF (MACHIN .EQ. 'SED') Y = AUGHRSLCOST(23)
X = IHRS*ICOST(C) + Y
HRSMIS = HRSMIS + IHRS
CALL KBLOCK (BLOCK,BDIM,BLK,B)
*LIKOST(B,L,CC) = LIKOST(B,L,CC) + X
MICOST = MICOST + X
GOTO 6000

* COL
READ PER BLOCK TONNAGE FROM OTHER TOTALS
READ TONS OF COAL SELF-HAULED, COLUMNS 25-29
READ PRICE PER TON, COLUMNS 34-38, IF DIFFERENT FROM COST(22)
TOTAL
TCOAL = TOTAL TONNAGE
SCOAL = STRIP TONNAGE (NON-AUGER)
SECOHL = TONNAGE SELF HAULED, NOT HAULED BY CONTRACT
COCHL = TONNAGE HAULED BY CONTRACT HAULER & COST(21)
7000 READ (03,7010,ERR=9907,END=9907) GO,BLK,COAL,AUGER,SECOHL,PRICE
7010 FORMAT (A5,1X,A4,2X,IS,2X,AS,2X,I5,2X,F6.3)
SETUP !COUNT = !COUNT + 1
WRITE (06,*) !COUNT, 'COAL ', COCHL, ' CONT HAUL $',COST(21)
IF (ICOUNT .GT. IMAX) GOTO 9907
IF (GO .NE. ' ') GOTO 9700
IF (BLK .EQ. ' ') GOTO 7000
CALL KBLOCK (BLOCK,BDIM,BLK,B)
*BLKCOL(B) = BLKCOL(B) + COAL
IF (AUGER .EQ. 'AUGER') THEN
AUGCOL(B) = AUGCOL(B) + COAL
ACOAL = ACOAL + COAL
ENDIF
TCOAL = TCOAL + COAL
SCOAL = SCOAL - ACOAL
COCHL = COAL - SECOHL
LIKOST(B,4,11) = LIKOST(B,4,11) + COCHL*COST(21)
TOCOHL = TOCOHL + COCHL
IF (PRICE .GT. 0.5) THEN
IF (AUGER .EQ. 'AUGER') THEN
AUGPRI(B) = PRICE
ELSE
STRPRI(B) = PRICE
ENDIF
ENDIF
GOTO 7000
*XVOL
READ PER BLOCK-LIFT VOLUME (BCY)
HAULED, CARRIED, AND PUSHED PORTIONS
7200 READ (03,7210,ERR=9907,END=9907) GO,BLK,LIF,VOLUME,H,CA,P
7210 FORMAT (A5,1X,A4,1X,A1,4(1X,I9))
SETUP !COUNT = !COUNT + 1
WRITE (06,*) !COUNT, 'VOLUME'
IF (ICOUNT .GT. IMAX) GOTO 9907
IF (GO .NE. ' ') GOTO 9700
IF (BLK .EQ. ' ') GOTO 7200
CALL KBLOCK (BLOCK,BDIM,BLK,B)
CALL JLIFT (LIF,L)
*BLKVOL(B) = BLKVOL(B) + VOLUME
LIFVOL(B,L) = LIFVOL(B,L) + VOLUME
TVOL = TVOL + VOLUME
HAUL(B,L) = H
HTOT = HTOT + H
CARRY(B, L) = CA
CTOT = CTOT + CA
PUSH(B, L) = P
PTOT = PTOT + P
GOTO 7200

THIS INPUT FollowS DATA INPUTS (ABOVE) But
PRECEDES RCL INPUTS

SPECIAL BLOCKS: MINING AREAS TO BE TREATED TOTALLY AS OVERHEAD
(Road to SITE, Sediment Ponds) OR AS RECLAMATION EXPENSES

8000 WRITE (92,*) 'SPECIAL BLOCKS'
WRITE (92,*)
8005 READ (05,8010,ERR=9908,END=9908) GO,CHAR,BLK
8100 FORMAT(A3,2X,A3,2X,A4)
IF (GO .NE. ' ') GOTO 9700
ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9908
IF (CHAR .EQ. ' ') GOTO 8005
IF (CHAR .EQ. 'REC') GOTO 8100
IF (CHAR .EQ. 'OHD') GOTO 8200
WRITE (06,*) 'SPECIAL BLOCK PROBLEM CHAR = ', CHAR
GOTO 9908

IF SPECIAL BLOCK CHAR = 'REC', TOTAL ALL EXPENSES AS RECLAMATION
AN EXAMPLE FOR USE OF THIS BLOCK: ALL COSTS OF HOLLOW FILL CONSTRUCTION
MUST BE TREATED AS RECLAMATION EXPENSE IF THEY ARE TO BE REDISTRIBUTED TO SOURCE BY 8500 BLOCK BELOW

8100 WRITE (92,*) ' ',BLK, ' = SPECIAL BLOCK'
WRITE (92,*) ALL EXPENSES TREATED AS RECLAMATION'
WRITE(92,*)
WRITE(06,*) 'INTO 8100'
T = 0.0
CALL KBLOCK(BLOCK,BDIM,BLK,B)
DO 8150 CC = 1,CCDIM
X = 0.0
DO 8140 L=1,LDIM X = X + LIKOST(B,L,CC)
LIKOST(B,L,CC) = 0.0
CONTINUE
8140 IF (X .EQ. 0.0) GOTO 8150
WRITE (92,8145)CCAT(CC),X
8145 FORMAT(5X,A20,8X,F8.2)
T = T + X
CONTINUE
8150 WRITE (92,8155)T
8155 FORMAT (5X,'TOTAL = RECLAMATION = ',6X,F8.2)
WRITE (92,*)
LIKOST(B,1,12) = T
GOTO 8005

IF SPECIAL BLOCK CHAR = 'OHD', TOTAL ALL EXPENSES AS OVERHEAD
TO BE REDISTRIBUTED TO SOURCE AREAS ON PER BCY BASIS VIA OHDTOT VARIABLE AND STATEMENTS 8810 +

8200 WRITE (92,*) ' ',BLK, ' = SPECIAL BLOCK'
WRITE (92,*) ALL EXPENSES TREATED AS OVERHEAD'
WRITE (92,*)
WRITE (06,*) '8200'
T = 0.0
CALL KBLOCK(BLOCK,BDIM,BLK,B)
DO 8250 CC = 1,CCDIM
X = 0.0
DO 8240 L=1,LDIM X = X + LIKOST(B,L,CC)
LIKOST(CB,L,CC) = 0.0

CONTINUE
IF (X .EQ. 0.0) GOTO 8250
WRITE(92,8245) COCAT(CC),X
8245 FORMAT (5X,A20,8X,F8.2)
T = T + X
8250 CONTINUE
WRITE (92,8255) T
8255 FORMAT (5X,'TOTAL',23X,F8.2)
WRITE (92,8257) OHDTOT
8257 FORMAT (5X,'OVERHEAD ACCOUNT =' ,F10.2)
WRITE (92,92) GOTO 8005

*RCL AND RED: SHOULD FOLLOW ALL INPUT - EXCEPT END

*RCL
*DISTRIBUTE RECLAMATION COSTS
*FROM DESTINATION TO SOURCE BASED UPON FRACTION
*OF TOTAL MATERIAL AT DESTINATION ORIGINATING AT EACH SOURCE
"FRAC" = PROPORTION OF DESTINATION MATERIAL ORIGINATING AT SOURCE
= PROPORTION OF DEST. RECLAMATION COST ATTRIBUTED TO SOURC
SINCE PROGRAM READS IN RECLAMATION COSTS AT DESTINATION,
THIS PROGRAM BLOCK REDISTRIBUTES THOSE COSTS TO SOURCE
FOR EXAMPLE: ALL COSTS OF HOLLOW FILL RECLAMATION SHOULD BE CHARGED
TO COAL BLOCKS WHERE THE SPOIL PUT IN THE FILL ORIGINATES

INPUT COLUMN FORMAT
COLS. 1-3: GO
5-8: DESTINATION DATA COS input
9-18: FRAC (DECIMAL) FOR THIS PURPOSE
19-22: SOURCE BLOCK
24: SOURCE LIFT

RCLEXP(B) CONTAINS ORIGINAL RECLAMATION EXPENSES BY BLOCK
RCLDXP(B) KEEPS TRACK OF UNDISTRIBUTED RECLAMATION EXPENSES
X = DISTRIBUTED FRACTION
T = TOTAL; TOTALS DISTRIBUTED FRACTIONS THROUGH 8550, THEN
ADDS QUANTITIES REMAINING IN RCLDXP(B) VARIABLE
RECLAMATION EXPENSES REDISTRIBUTED TO COCAT(15)
TO PUT THEM BACK INTO COCAT(12) GO THROUGH 8600 BLOCK

8500 WRITE (92,8499)
8499 FORMAT (7(9X,'*'))
WRITE (92,X)
WRITE (92,X) 'DESTINATION RECLAMATION TOTALS FROM INPUT'
WRITE (92,X)
WRITE (92,X) 'REDISTRIBUTE RECLAMATION EXPENSE'
T = 0.0
DO 8503 B = 1,BDIM
DO 8502 L = 1,LDIM
IF (LIKOST(B,L,12) .GT. 0.01) THEN
WRITE (92,8501) BLOCK(B), LIFT(L), LIKOST(B,L,12)
8501 FORMAT (5X,31X,A4,1X,A1,1X,F10.2)
RCLEXP(B) = RCLEXP(B) + LIKOST(B,L,12)
LIKOST(B,L,12) = 0.0
ENDIF
8502 CONTINUE
T = T + RCLEXP(B)
RCLDXP(B) = RCLEXP(B)
8503 CONTINUE
WRITE(92,8504) T
8504 FORMAT (5X,'TOTAL BEFORE REDISTRIBUTION',11X,F10.2)
WRITE(92,X)
WRITE(92,8505)
8505 FORMAT(' DISTRIBUTION OF RECLAMATION EXPENSES FROM DESTINATION + TO SOURCE')
WRITE(92,*)
WRITE(92,8507)
T = 0.0
8507 FORMAT (5X,'SOURCE',3X,'FRAC',1X,'DESTIN',5X,'DESTIN EXP',3X,'SOUR + CE EXP',3X,'UNDISTRIBUTED')
*
8515 READ(03,8520) GO, DEBLK, FRAC, SOBLK, SOLIFT
8520 FORMAT (A3,1X,A4,F10.8,A4,1X,A1)
IF (GO .NE. ' ') GOTO 8570
ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9908
IF (SOBLK .EQ. ' ') GOTO 8515
CALL KBLOCK (BLOCK,BDIM,SOBLK,SB)
CALL JLIFT (SOLIFT,SL)
CALL KBLOCK (BLOCK,BDIM,DEBLK,DB)
X = FRAC*RCLDXP(DB)
LIKOST(SB,SL,15) = LIKOST(SB,SL,15) + X
RCLDXP(DB) = RCLDXP(DB) - X
WRITE(92,8550) BLOCKCSB>,LIFTCSL>,FRAC,BLOCKCDB>,RCLDXPCDB>,X,RCLDXPCDB)
8550 FORMAT(5X,A4,1X,A1,1X,F8.5,1X,A4,5X,F10.2,3X,F10.2,3X,F10.2)
T = T + X
GOTO 8515
*
8570 WRITE (92,8580) T
8580 FORMAT (5X,'TOTAL REDISTRIBUTION',18X,F10.2)
WRITE(92,*)
DO 8590 B = 1,BDIM
LIKOST(B,5,12) = LIKOST(B,5,12) + RCLDXP(B)
T = T + RCLDXP(B)
RCLDXP(B) = 0.0
RCLDXP(B) = 0.0
8590 CONTINUE
WRITE (92,8595) T
8595 FORMAT (5X,'TOTAL AFTER REDISTRIBUTION',12X,F10.2)
WRITE(92,*)
8597 FORMAT(' * * * * * * * *' )
WRITE (92,8597) GOTO 9700
*
* RED
* PREPARE FOR SECOND REDISTRIBUTION BY PLACING SELECTED
* BLOCK RECLAMATION EXPENSES FROM COCAT(15) TO COCAT(12)
* AN EXAMPLE: IF RCL BLOCK ABOVE REDISTIBUTES HOLLOW FILL 1 EXPENSE
* BACK TO HOLLOW FILL 2, GO THROUGH THIS BLOCK BEFORE
* TO ALLOW NEW HF2 EXPENSE TO GO BACK TO COAL BLOCKS
*
8600 READ(03,8610) GO, BLK
8610 FORMAT (A3,1X,A4)
IF (GO .NE. ' ') GOTO 9700
ICOUNT = ICOUNT + 1
IF (ICOUNT .GT. IMAX) GOTO 9908
IF (BLK .EQ. ' ') GOTO 8600
CALL KBLOCK (BLOCK,BDIM,BLK,B)
WRITE (92,8611) BLK
8611 FORMAT (5X,'IN RED ; BLOCK = ',A4)
DO 8620 L = 1,LDIM
LIKOST(B,L,12) = LIKOST(B,L,12) + LIKOST(B,L,15)
LIKOST(B,L,15) = 0.0
8620 CONTINUE
GOTO 8600
*
* END
* FINAL INPUT
*
8800 READ (03,8805,ERR=9919,END=9919) DAYONE
READ (03,8805,ERR=9919,END=9919) DAYEND
READ (03,8805,ERR=9919,END=9919) JOB
8805 FORMAT (5X,A15)
*
WRITE HEADERS ALL FILES
*
WRITE (92,8807) JOB, DAYONE, DAYEND
WRITE (93,8807) JOB, DAYONE, DAYEND
WRITE (94,8807) JOB, DAYONE, DAYEND
WRITE (95,8807) JOB, DAYONE, DAYEND
8807 FORMAT (A20,A15,' THRU ',A15)
*
CALCULATE AND TOTAL FROM HERE TO 8950
*
OHDTOT (FROM SPB, 8000 & 8200) + COST(19)*TOTAL COAL PRODUCED
* + TOTAL, ALL LOCATIONS, COST CATEGORY 14
*
= TOTAL OVERHEAD, REDISTRIBUTED TO SOURCE BLOCKS PER BCY
*
8810 WRITE (92,*) 'DISTRIBUTE OVERHEAD COSTS'
WRITE (92,*)
WRITE (92,8830) OHDTOT
8830 FORMAT (5X,'OVERHEAD TOTAL BEFORE PER TON TOTAL ADDED = ',F10.2)
OHDTOT = OHDTOT + COST(19)*TFCOAL
WRITE (92,8832) OHDTOT
8832 FORMAT (5X,'OVERHEAD TOTAL AFTER PER TON TOTAL ADDED = ',F10.2)
X = 0.0
DO 8839 B = 1,BDIM
8839 CONTINUE

CONTINUE

OHDTOT = OHDTOT + X
WRITE (92,8840) OHDTOT
8840 FORMAT(5X,'OVERHEAD TOTAL AFTER ADDING LIFT OVERHEAD = ',F10.2)
OHDPYC = OHDTOT/TVOL
WRITE (92,8842) OHDPCY
8842 FORMAT(5X,'OVERHEAD PER BCY = ',F10.2)
*
TOTAL AGGREGATE COST CATEGORIES
*
8900 T = 0.0
DO 8909 B = 1,BDIM
8909 CONTINUE

CONTINUE

T = T + BLKTOT(B)
8909 CONTINUE
*
TOTAL COCAT(08) - LOAD AND HAUL, ALL MACHINES
*
8930 DO 8950 B = 1,BDIM
8950 CONTINUE

CONTINUE

WRITE (06,*) '8950'

CCCOST(08) = CCCOST(4) + CCCOST(5) + CCCOST(6) + CCCOST(7)
*
CALCULATE HAUL AND CARRY-PUSH COSTS, PER BCY HANDLED
* DO 8956 B = 1, BDIM
  H = 0
  C = 0
  DO 8954 L = 1, LDIM
     IF (HAUL(B, L) .EQ. 0) THEN
        BHAUL(B, L) = 0.0
     ELSE
        BHAUL(B, L) = LIKOST(B, L, 8) / HAUL(B, L)
     ENDIF
     V = CARRY(B, L) + PUSH(B, L)
     IF (V .EQ. 0) THEN
        BLKAR(B, L) = 0.0
     ELSE
        BLKAR(B, L) = LIKOST(B, L, 3) / V
     ENDIF
     H = H + HAUL(B, L)
     C = C + V
  8954 CONTINUE
  IF (H .EQ. 0) THEN
     BHAUL(B) = 0.0
  ELSE
     BHAUL(B) = BLKAR(B, 8) / H
  ENDIF
  IF (C .EQ. 0) THEN
     BLKAR(B) = 0.0
  ELSE
     BLKAR(B) = BLKAR(B, 3) / C
  ENDIF
  8956 CONTINUE

* **********   * **********
* ENTER WRITE BLOCK; FIRST OUTPUT THE INPUT PER UNIT COSTS   *
* **********   * **********
* WRITE(93,*)
* WRITE(93,*) 'INPUT COSTS '
* WRITE(93,*)
* DO 8980 C = 1, CDIM
* WRITE (93,8970) C, COITEM(C), COST(C)
* 8970 FORMAT(5X, I2, 2X, A20, 2X, F6.2)
* 8980 CONTINUE
* *
* WRITE (93,*)
* WRITE (93,*) 'TOTAL JOB COSTS'
* WRITE (93,8989) T
* 8989 FORMAT(5X, F11.2)
* *
* PROFIT-LOSS SUMMARY
* WRITE (93,*)
* WRITE (93,8999)
* 8999 FORMAT('PROFIT LOSS SUMMARY ')
* WRITE (93,*)
* WRITE (93,9001)
* 9001 FORMAT(59X, 'PROFIT', 8X, 'PROFIT', 6X, 'PROFIT', 3X, 'COAL')
* WRITE (93,9002)
* 9002 FORMAT (5X, 'BLOCK', 8X, 'COST', 5X, 'RETURNS', 4X, 'W/OVERHEAD', 3X,
* + 'W/OVERHEAD', 2X, 'PER TON', 2X, 'PRICE')
* T = 0.0
* Z = 0.0

* X = RETURNS PER BLOCK
* BLKTOT(B) = COST OF MINING BLOCK, WITH OVERHEAD
* W = RETURNS - COSTS, PER BLOCK PROFIT
* ND = RETURNS - VARIABLE COSTS (NO OVERHEAD)
* Z = TOTAL RETURNS
* Y = PROFIT PER TON
T = TOTAL PROFITS
BLCOST = TOTAL COSTS

DO 9050 B = 1, BDIM
   IF (BLKTOT(B) .LT. 0.5) GOTO 9050
   X = AUGCOL(B) * STRPRI(B) + (BLKCOL(B) - AUGCOL(B)) * STRPRI(B)
   W = X - BLKTOT(B)
   IF (BLKCOL(B) .GT. 0) THEN
      PRICE = X / BLKCOL(B)
      Y = W / BLKCOL(B)
   ELSE
      PRICE = 0.0
      Y = 0.0
   ENDIF
   W0 = X - (BLKTOT(B) - BLKVOL(B) * MDPCY)
   T = T + W
   BLCOST = BLCOST + BLKTOT(B)
   Z = Z + X
   WRITE (93, 9040) BLOCK(B), BLKTOT(B), X, W, W0, Y, PRICE
9050 CONTINUE
WRITE (93, 9055) BLCOST, Z, T
9055 FORMAT (5X, 'TOTAL JOB COSTS - GROSS, PER BLOCK BCY, AND PER TON')

WRITE (93,*) 'TOTAL JOB COSTS - GROSS, PER BLOCK BCY, AND PER TON'
WRITE (93,*)
T = 0.0
IF (TVOL .GT. 0 .AND. TCOAL .GT. 0) THEN
   DO 9070 CC = 1, CCDIM
      YDCOST = CCCOST(CC) / TVOL
      TNCOST = CCCOST(CC) / TCOAL
      IF (CC .EQ. 10 .AND. ACOAL .GT. 0) THEN
         TNCOST = CCCOST(CC) / ACOAL
      ENDIF
      WRITE (93, 9065) COCAT(CC), CCCOST(CC), YDCOST, TNCOST
9065 FORMAT (5X, A20, 1X, F10.2, 1X, F8.3, 1X, F8.2)
   T = T + CCCOST(CC)
9070 CONTINUE
ENDIF
WRITE (06,*) '9070'
WRITE (93, 9073) TCOAL, SCOAL, ACOAL
9073 FORMAT (5X, 'COAL PRODUCTION, TONS, TOTAL STRIP & AUGER - ', 3(2X, I6))
WRITE (93, 9074) TVOL
9074 FORMAT (5X, 'TOTAL VOLUME, BANK CUBIC YARDS - ', I9)
SRATIO = REAL(TVOL) / TCOAL
CHAR = 'TOTAL COAL'
WRITE (93, 9075) CHAR, SRATIO
9075 FORMAT (5X, 'TOTAL STRIPPING RATIO YARDS PER TON', 1X, A10, F7.2)
SRATIO = REAL(TVOL) / (TCOAL - ACOAL)
CHAR = 'STRIP COAL'
WRITE (93, 9075) CHAR, SRATIO
X = T / TCOAL
WRITE (93, 9078) X
9078 FORMAT (5X, 'COST PER TON = ', F7.2)
X = (T - CCCOST(10) - CCCOST(9) - CCCOST(11)) / TVOL
WRITEC93,9081) X
9081 FORMAT ('5X,'COST PER YARD EXCLUSIVE OF COAL HANDLING ' ,F8.3)
X = TOCOHL*COSTC21)
WRITEC93,9082) X,TOCOHL
9082 FORMAT ('5X,'CONTRACT HAULER COST AND TONS ' ,F10.2,2X,I6)
X = TVOL
WRITEC93,9083) X
9083 FORMAT ('5X,'TOTAL COST PER OVERBURDEN BANK CUBIC YARD ',F8.3)
X = OHDTOT/TCOAL
WRITEC93,9084) X, OHDTOT
9084 FORMAT ('5X,'PER TON AND TOTAL OVERHEAD COSTS ',F5.2,F10.2)
X = PREOHD*TCOAL
WRITEC93,9085) PREOHD,X
9085 FORMAT ('5X,'PRE SPECIAL BLOCK PER TON OVERHEAD COSTS ',F5.2,F10.2)
X = CCCOST(9)/ACOAL
WRITEC93,9086) CCCOST(9), X
9086 IF (ACOAL .GT. 0) THEN
IF (CACOAL .GT. 0) THEN
WRITEC93,9086) CCCOSTClO)/ACOAL
ENDIF
9086 FORMAT ('5X,'GROSS AND PER TON COSTS - ',A20,F8.2,1X,F8.2)
X = CCCOSTClO)/ACOAL
WRITEC93,9086) CCCOSTClO), X
ENDIF

MACHINE COST TOTALS AND MACHINE HOURS
*
WRITEC93,9088) CHAR,HLCOST,HRSHLR
WRITEC93,9088) CHAR,LDCOST,HRSLDA,HRSDB,HRSDC
WRITEC93,9088) CHAR,DZCOST,HRSDZA,HRSDZC
WRITEC93,9088) CHAR,DRCOST,HRSDLR
WRITEC93,9088) CHAR,MICOST,HRSMIS
WRITEC93,9088) CHAR,SHCOST,HRSSHT
CHAR = 'TOTALS
WRITEC93,9088) CHAR,T,HRS
9088 FORMAT ('5X,A15,F11.2,5X,3(F9.1))
WRITE (06,*) '9088'
WRITE (93,*)

BLOCK AND LIFT COST SUMMARIES - BRIEF VERSION - FILE 93
*
WRITE (93,*) 'BLOCK AND LIFT COST SUMMARIES'
WRITE (93,*) ' FIRST $ PER YARD DOES NOT INCLUDE COAL HANDLING +COSTS'
WRITE (93,*)
TOT = 0.0
WRITE (93,9140)
9140 FORMAT ('12X,'YARDS',4X,'TONS',2X,'YARDS',3X,'STRIPPING',2X,':F8.3' ,+ 2X,'$ PER',2X,'$ PER',2X,'$ PER',2X,'$ PER',2X,'$ COAL COST')
WRITE (93,9150)
9150 FORMAT ('5X,'BLK L',2X,'VOLUME',3X,'COAL',3X,'TON',4X,'$ COST',5X ,+ 3X,'YARD',5X,'YARD',3X,'TON',4X,'$ TON ')
WRITE (93,*)
WRITE (06,*) '9150'
DO 9190 B = 1,BDIM
IF (BLKTOT(B) .LT. 0.5) GOTO 9190
DO 9180 L = 1,LDIM
IF (LIFVOL(B,L) .EQ. 0.0) GOTO 9180
IF (LIFVOL(B,L) .GT. 0) THEN
YDCOST = LIFTOT(B,L)/LIFVOL(B,L)
T = (LIFTOT(B,L) - LIKOST(B,L,9) - LIKOST(B,L,10)
- LIKOST(B,L,11))/LIFVOL(B,L)
ELSE
  YDCOST = 0.0
  T = 0.0
ENDIF
WRITE(93,9176) BLOCK(B), LIFT(L), LIFVOL(B,L), LIFTOT(B,L),
         T, YDCOST
9176 FORMAT(4X,A4,1X,A1,1X,I7,16X,F9.2,1X,F7.2,F7.2)
TOT = TOT + LIFTOT(B,L)
9180 CONTINUE
IF (BLKCOL(B) .EQ. 0) GOTO 9189
IF (BLKVOL(B) .EQ. 0) GOTO 9189
SRATIO = REAL(BLKVOL(B))/BLKCOL(B)
YDCOST = BLKTOT(B)/BLKVOL(B)
* Z = COAL COST TOTAL; Y = COAL COST PER TON; T = SHORT $ PER BCY
X = BLKOST(B,9)/BLKCOL(B)
Z = X*BLKOST(B,9) + BLKOST(B,10) + BLKOST(B,11)
IF (BLKCOL(B) .GT. 0) THEN
  Y = Z/BLKCOL(B)
  TNCOST = BLKTOT(B)/BLKVOL(B)
  ELSE
  Y = 0.0
  TNCOST = 0.0
ENDIF
IF (BLKVOL(B) .GT. 0) THEN
  T = (BLKTOT(B) - Z)/BLKVOL(B)
  ELSE
  T = 0.0
ENDIF
WRITE(93,9187) BLOCK(B), BLKVOL(B), BLKCOL(B), SRATIO, BLKTOT(B),
         T, YDCOST, TNCOST, Y
9187 FORMAT(4X,A4,3X,I7,2X,I5,2X,F5.1,1X,F10.2,3X,F5.2,2X,F5.2,
           1X,F7.2,1X,F7.2)
9189 WRITE(93,*)
9190 CONTINUE
WRITE (93,9192) TOT
9192 FORMAT (5X,'TOTAL =',25X,F11.2)
WRITE (93,*) '9190'
WRITE(93,*) 'TO CHECK OUTPUT: TOTAL JOB COSTS = MACHINERY COST'
WRITE(93,*) 'TOTALS + CONTRACT HAULING + DRILL AND SHOOT'
WRITE(93,*) 'TOTALS - DRILL COST TOTAL - SHOOT LABOR TOTALS'
WRITE(93,*) '+ INPUT OVERHEAD + SEEDING'
* BLOCK AND LIFT COST DETAILS - FILE 94
* WRITE(94,*) 'BLOCK AND LIFT COST DETAILS'
* WRITE(94,*) FIRST = .TRUE.
* AFTER RE-CALCULATING BLKOST AND LIKOST VARIABLES ON PER BCY(TON) BASES
* AND SETTING FIRST = .FALSE.
* PROGRAM RETURNS TO 9205
9205 DO 9466 B = 1, BDIM
  IF (BLKTOT(B) .LT. 0.01) GOTO 9466
  IF (BLKCOL(B) .EQ. 0) GOTO 9400
  WRITE (94,9210) BLOCK(B)
9210 FORMAT('BLOCK = ',A4,12X,'TOTAL',3X,'T LIFT',3X,'2 LIFT',
           3X,'3 LIFT',3X,'X LIFT',3X,'R LIFT')
  DO 9240 CC = 1, CCDIM
    IF (BLKOST(B,CC) .EQ. 0.00) GOTO 9240
    WRITE (94,9230) CCCAT(CC), BLKOST(B,CC), LIKOST(B,1,CC),
               LIKOST(B,2,CC), LIKOST(B,3,CC), LIKOST(B,4,CC),
               LIKOST(B,5,CC)
9230 FORMAT (A20,6(F9.2))
9240 CONTINUE
WRITE (94,9235) BLKTOT(B),LIFTOT(B,1),LIFTOT(B,2),
LIFTOT(B,3),LIFTOT(B,4),LIFTOT(B,5)
9235  FORMAT ('TOTAL',15X,6(F9.2))
WRITE (94,*)
*
9400 IF (.NOT. FIRST) GOTO 9466
*
9410 IF (BLKCOL(B) .EQ. 0) THEN
WRITE (94,9410) BLOCK(B)
9410  FORMAT('BLOCK = ',A4,12X,'COST')
IF (BLKTOT(B) .EQ. 0.00) GOTO 9450
DO 9440 CC = 1,CCDIM
IF (BLKOST(B,CC) .EQ. 0.0) GOTO 9440
WRITE (94,9430) COCAT(CC),BLKOST(B,CC)
9430  FORMAT (A20,F9.2)
CONTINUE WRITE (94,9445) BLKTOT(B)
9445  FORMAT ('TOTAL',15X,F9.2)
WRITE (94,*)
ENDIF
9466 CONTINUE
IF (.NOT. FIRST) GOTO 9600
*
9500 DO 9550 9500
WRITE (94,9500) '9500'
9550  B = 1,BDIM IF (BLKVOL(B) .NE. 0) THEN
BLKTOT(B) = BLKTOT(B)/BLKVOL(B)
ELSE
BLKTOT(B) = 0.0
ENDIF
DO 9540 CC = 1,CCDIM
IF (CC .EQ. 9 .AND. BLKCOL(B) .GT. 0) THEN
BLKOST(B,CC) = BLKOST(B,CC)/BLKCOL(B)
ELSEIF (CC .EQ. 10 .AND. AUGCOL(B) .GT. 0) THEN
BLKOST(B,CC) = BLKOST(B,CC)/AUGCOL(B)
ELSEIF (CC .EQ. 11 .AND. BLKCOL(B) .GT. 0) THEN
BLKOST(B,CC) = BLKOST(B,CC)/BLKCOL(B)
ELSEIF (CC .NE. 9 .AND. CC .NE. 10 .AND. CC .NE. 11 .AND.
LIFVOL(B) .NE. 0) THEN
BLKOST(B,CC) = BLKOST(B,CC)/LIFVOL(B)
ELSE
BLKOST(B,CC) = 0.0
ENDIF
DO 9530 L = 1,LDIM
IF (CC .EQ. 9 .AND. BLKCOL(B) .GT. 0) THEN
LIKOST(B,L,CC) = LIKOST(B,L,CC)/BLKCOL(B)
ELSEIF (CC .EQ. 10 .AND. AUGCOL(B) .GT. 0) THEN
LIKOST(B,L,CC) = LIKOST(B,L,CC)/AUGCOL(B)
ELSEIF (CC .EQ. 11 .AND. BLKCOL(B) .GT. 0) THEN
LIKOST(B,L,CC) = LIKOST(B,L,CC)/BLKCOL(B)
ELSEIF (CC .NE. 9 .AND. CC .NE. 10 .AND. CC .NE. 11 .AND.
LIFVOL(B,L) .NE. 0) THEN
LIKOST(B,L,CC) = LIKOST(B,L,CC)/LIFVOL(B,L)
ELSE
LIKOST(B,L,CC) = LIKOST(B,L,CC)/LIFVOL(B,L)
ENDIF
9500 CONTINUE
IF FIRST = .FALSE.
*
RECALCULATE GROSS COSTS ON PER YARD AND PER TON BASIS
*
WRITE (94,*)
LIKOST\(B,L,CC\) = 0.0

ENDIF

9530 CONTINUE
9540 CONTINUE
9550 CONTINUE
WRITE (06,\*) '9550'
DO 9570 B = 1, BDIM
   DO 9560 L = 1, LDIM IF(LIFVOL(B,L) .GT. 0) THEN
      LIFTOT(B,L) = LIFTOT(B,L)/LIFVOL(B,L)
   ELSE
      LIFTOT(B,L) = 0.0
   ENDIF
9560 CONTINUE
9570 CONTINUE
WRITE (06,\*) '9570'
GOTO 9205

9600 WRITE(95,\*) '9600'
WRITE(95,\*) 'PER BLOCK BCY OVERBURDEN HANDLING COST COMPARISONS'
WRITE(95,\*) 'PER TON COAL HANDLING COST COMPARISONS'
WRITE(95,\*)
DO 9609 CC = 1, CCDIM
   WRITE(95,9606) CCATCCC) FORMAT(2X,A20)
   DO 9608 B = 1, BDIM IF(CBLKOSTCB,CC).EQ. 0.00) GOTO 9608
      WRITE(95,9607) BLOCKCB>, BHAULCB>, BLHAULCB,1>, BLHAULCB,2>, BLHAULCB,3>, BLHAULCB,4>, BLHAULCB,5>
   FORMAT(A4,16X,6(F9.2))
9607 CONTINUE
9608 CONTINUE
WRITE(95,\*)
9609 CONTINUE

WRITE COSTS PER BCY HAULED AND PER BCY CARRIED & PUSHED

9610 FORMAT (7X,'FOLLOWING COSTS DEPEND UPON ACCURACY OF MOVED VOLUME +ESTIMATES')
WRITE(95,\*)
WRITE(95,\*) 'HAULING COSTS PER BCY HAULED'
WRITE(95,\*) BLK = ' '
WRITE(95,9610) BLK
DO 9615 B = 1, BDIM IF (BLKOST(B,8) .LT. 0.005) GOTO 9615
   WRITE (95,9611) BLOCK(B), BHAUL(B), BLHAUL(B,1), BLHAUL(B,2),
      BLHAUL(B,3), BLHAUL(B,4), BLHAUL(B,5)
9611 FORMAT(A4,16X,6(F9.2))
9615 CONTINUE
   IF (HTOT .EQ. 0) GOTO 9620
   X = CCCOST(8)/HTOT
   BLK = 'JOB'
   WRITE (95,9611) BLK, X
9620 CONTINUE
WRITE(95,\*) 'CARRY & PUSH COSTS PER BCY CARRIED-PUSHED'
WRITE(95,\*) BLK = ' '
WRITE(95,9620) BLK
DO 9625 B = 1, BDIM
   IF (BLKOST(B,3) .LT. 0.005) GOTO 9625
      WRITE (95,9611) BLOCK(B), BCARP(B), BLCARPB,1), BLCARPB,2),
         BLCARPB,3), BLCARPB,4), BLCARPB,5)
9625 FORMAT(A4,16X,6(F9.2))
9620 CONTINUE
9625 CONTINUE
IF (CTOT .EQ. 0 .AND. PTOT .EQ. 0) GOTO 9670
X = CCCOSTC3/(CTOT + PTOT)
BLK = 'JOB'
WRITE (95,9611) BLK, X
*
WRITE (06,*) 'EXECUTION COMPLETE ICOUNT = ',ICOUNT
WRITE(95,*)
9670 WRITE(95,*) 'LOCATION BLOCKS AND INTERNAL BLOCK NUMBERS'
DO 9680 B = 1,BDIM
   IF (BLOCK(B) .NE. '') WRITE(95,9675) BLOCK(B),B
9675 FORMAT(5X,A4,2X,I2)
9680 CONTINUE
GOTO 9990
*
9700 BLOCK
9700 IF(GO .EQ. 'COS') GOTO 1000
IF(GO .EQ. 'HLR') GOTO 2000
IF(GO .EQ. 'LDA') C = 2
IF(GO .EQ. 'LDA') GOTO 3000
IF(GO .EQ. 'LDB') C = 3
IF(GO .EQ. 'LDB') GOTO 3000
IF(GO .EQ. 'LDC') C = 4
IF(GO .EQ. 'LDC') GOTO 3000
IF(GO .EQ. 'DZA') C = 5
IF(GO .EQ. 'DZA') GOTO 4000
IF(GO .EQ. 'DZC') C = 6
IF(GO .EQ. 'DZC') GOTO 4000
IF(GO .EQ. 'DRL') GOTO 5000
IF(GO .EQ. 'MIS') GOTO 6000
IF(GO .EQ. 'COL') GOTO 7000
IF(GO .EQ. 'VOL') GOTO 7200
IF(GO .EQ. 'SPB') GOTO 8000
IF(GO .EQ. 'RCL') GOTO 8500
IF(GO .EQ. 'RED') GOTO 8600
IF(GO .EQ. 'END') GOTO 8800
WRITE(06,*) '"GO" ERROR 9700 BLOCK; GO = ',GO,'ICOUNT = ',ICOUNT
GOTO 9990
*
9900 WRITE (06,*) 'ERROR INITIAL "GO" NOT READ AS "COS"'
GOTO 9990
9901 WRITE (06,*) 'ERROR READ BLOCK 1000 ICOUNT = ',ICOUNT
GOTO 9990
9902 WRITE (06,*) 'ERROR READ BLOCK 2000 ICOUNT = ',ICOUNT
GOTO 9990
9903 WRITE (06,*) 'ERROR READ BLOCK 3000 ICOUNT = ',ICOUNT
GOTO 9990
9904 WRITE (06,*) 'ERROR READ BLOCK 4000 ICOUNT = ',ICOUNT
GOTO 9990
9905 WRITE (06,*) 'ERROR READ BLOCK 5000 ICOUNT = ',ICOUNT
GOTO 9990
9906 WRITE (06,*) 'ERROR READ BLOCK 6000 ICOUNT = ',ICOUNT
GOTO 9990
9907 WRITE (06,*) 'ERROR READ BLOCK 7000 ICOUNT = ',ICOUNT
GOTO 9990
9908 WRITE (06,*) 'ERROR READ BLOCK 8000 ICOUNT = ',ICOUNT
GOTO 9990
9909 WRITE (06,*) 'ERROR READ BLOCK 9000 ICOUNT = ',ICOUNT
GOTO 9990
9910 WRITE (06,*) 'MACHIN SYMBOL NOT LISTED MACHIN = ',MACHIN
GOTO 9990
9919 WRITE (06,*) 'ERROR READING 8800 END BLOCK ',ICOUNT
GOTO 8810
9990 STOP
9991 END
SUMMARY COST

PONY RUN  DAY 1 1999  THRU  DAY 11 1999

INPUT COSTS

1 HAULER PER HOUR  75.00
2 A LOADER PER HOUR  100.00
3 B LOADER PER HOUR  80.00
4 COAL LOADER PER HOUR  70.00
5 A DOZER PER HOUR  70.00
6 C DOZER PER HOUR  40.00
7 DRILL PER HOUR  70.00
8 DRILL/BLAST PER HOLE  2.50
9 DRILL HOLE PER FOOT  0.10
10 ANFO PER LB.  0.10
11 LABOR PER HOUR  20.00
12 COAL TRUCK PER HOUR  50.00
13 WATER TRUCK PER HOUR  50.00
14 GRADER PER HOUR  40.00
15 SEEDER PER HOUR  30.00
16 BACKHOE PER HOUR  35.00
17 AUGER PER HOUR  70.00
18 FOREMAN LABOR HOUR  8.00
19 OVERHEAD PER TON  3.00
20 TEMPORARY LABOR HOUR  8.00
21 CONTRACT COAL HAUL  1.50
22 COAL REVENUE PER TON  20.00
23 SEEDING / BATCH, ACRE  100.00

TOTAL JOB COSTS  36726.91

PROFIT LOSS SUMMARY

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>COST</th>
<th>RETURNS</th>
<th>PROFIT W/ OVERHEAD</th>
<th>PROFIT WD/ OVERHEAD</th>
<th>PROFIT PER TON</th>
<th>COST COAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1</td>
<td>23085.27</td>
<td>21200.00</td>
<td>-1885.27</td>
<td>1814.98</td>
<td>-1.78</td>
<td>20.00</td>
</tr>
<tr>
<td>MB2</td>
<td>11646.50</td>
<td>18090.00</td>
<td>6445.50</td>
<td>8293.62</td>
<td>7.41</td>
<td>20.79</td>
</tr>
<tr>
<td>MB3</td>
<td>1995.14</td>
<td>0.00</td>
<td>-1995.14</td>
<td>-1653.58</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>TOTAL</td>
<td>36726.91</td>
<td>39290.00</td>
<td>0.285</td>
<td>3.05</td>
<td>1.774</td>
<td>19.03</td>
</tr>
</tbody>
</table>

TOTAL JOB COSTS - GROSS, PER BCY, AND PER TON

| CLEAR AND BENCH | 490.00 | 0.024 | 0.25 |
| DRILL AND SHOOT | 685.00 | 0.033 | 3.57 |
| CARRY AND PUSH  | 830.00 | 0.040 | 0.43 |
| LOAD & HAUL: LOADERS | 5650.00 | 0.275 | 2.93 |
| LOAD & HAUL: DOZERS | 1750.00 | 0.085 | 0.91 |
| LOAD & HAUL: HAULERS | 7965.00 | 0.385 | 4.13 |
| LOAD & HAUL: DUMPSITE | 770.00 | 0.037 | 0.40 |
| LOAD & HAUL: TOTAL | 16135.00 | 0.779 | 8.36 |
| COAL: CLEAN & LOAD | 840.00 | 0.041 | 0.44 |
| COAL: AUGER | 470.00 | 0.023 | 9.82 |
| COAL: HAULING | 2945.00 | 0.142 | 1.53 |
| RECLAMATION - DESTIN | 0.00 | 0.000 | 0.00 |
| OTHER | 0.00 | 0.000 | 0.00 |
| OVERHEAD PER TON | 5891.92 | 0.285 | 3.05 |
| RECLAMATION - SOURCE | 2240.00 | 0.108 | 1.16 |
| TOTAL | 36726.91 | 1.774 | 19.03 |

JOB SUMMARY

COAL PRODUCTION, TONS, TOTAL STRIP & AUGER - 1930 1630 300
TOTAL VOLUME, BANK CUBIC YARDS - 20700
STRIPPING RATIO YARDS PER TON TOTAL COAL 10.73
STRIPPING RATIO YARDS PER TON STRIP COAL 12.70
COST PER TON = 19.03
COST PER YARD EXCLUSIVE OF COAL HANDLING 1.569
CONTRACT HAULER COST AND TONS 2595.00 1750
TOTAL COST PER OVERBURDEN BANK CUBIC YARD 1.774
PER TON AND TOTAL OVERHEAD COSTS 3.05 5891.93
PRE SPECIAL BLOCK PER TON OVERHEAD COSTS 3.00 5791.93
GROSS AND PER TON COSTS - COAL: CLEAN & LOAD 840.00 0.44
GROSS AND PER TON COSTS - COAL: AUGER 470.00 1.57

MACHINE COST TOTALS (INCLUDING OPERATOR WAGE) AND MACHINE HOURS

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HAULERS</td>
<td>LOADERS</td>
<td>DOZERS</td>
<td>DRILL</td>
<td>OTHER MACHINES</td>
<td>SHOT LABOR</td>
<td>TOTALS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8175.00</td>
<td>6830.00</td>
<td>4620.00</td>
<td>2940.00</td>
<td>1830.00</td>
<td>420.00</td>
<td>24815.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>109.0</td>
<td>58.5</td>
<td>62.0</td>
<td>42.0</td>
<td>26.0</td>
<td>21.0</td>
<td>339.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GROSS AND PER TON COSTS - COAL: AUGER 470.00 1.57

BLOCK AND LIFT COST SUMMARIES
FIRST $ PER YARD DOES NOT INCLUDE COAL HANDLING COSTS

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLK L</td>
<td>VOLUME</td>
<td>YARDS</td>
<td>TONS</td>
<td>STRIPPING</td>
<td>$ PER</td>
<td>$ PER</td>
<td>$ PER</td>
<td>COAL COST</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1 T</td>
<td>8000</td>
<td>11861.35</td>
<td>1.48</td>
<td>1.48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1 2</td>
<td>5000</td>
<td>7123.92</td>
<td>1.42</td>
<td>1.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1  X</td>
<td>0</td>
<td>4100.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB1</td>
<td>13000</td>
<td>1060</td>
<td>12.3</td>
<td>23085.27</td>
<td>1.62</td>
<td>1.78</td>
<td>21.78</td>
<td>1.96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>4000</td>
<td>5434.56</td>
<td>1.36</td>
<td>1.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2  2</td>
<td>2500</td>
<td>3026.97</td>
<td>1.21</td>
<td>1.21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2  X</td>
<td>0</td>
<td>3185.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2</td>
<td>6500</td>
<td>870</td>
<td>7.5</td>
<td>11646.50</td>
<td>1.46</td>
<td>1.79</td>
<td>13.59</td>
<td>2.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB3 T</td>
<td>1200</td>
<td>1995.14</td>
<td>1.66</td>
<td>1.66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>36726.91</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TO CHECK OUTPUT: TOTAL JOB COSTS = MACHINERY COST TOTALS + CONTRACT HAULING + DRILL AND SHOOT TOTALS - DRILL COST TOTAL - SHOOT LABOR TOTALS + INPUT OVERHEAD + SEEDING

* * * PERBLOCK COST * * *

<table>
<thead>
<tr>
<th>PONY RUN</th>
<th>DAY 1 1999 THRU DAY 11 1999</th>
</tr>
</thead>
</table>

BLOCK AND LIFT COST DETAILS

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLOCK = MB1 TOTAL</td>
<td>T LIFT</td>
<td>2 LIFT</td>
<td>3 LIFT</td>
<td>X LIFT</td>
<td>R LIFT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLEAR AND BENCH</td>
<td>70.00</td>
<td>70.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRILL AND SHOOT</td>
<td>4590.00</td>
<td>2570.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2020.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CARRY AND PUSH</td>
<td>340.00</td>
<td>0.00</td>
<td>340.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD &amp; HAUL: LOADERS</td>
<td>3700.00</td>
<td>2100.00</td>
<td>1600.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD &amp; HAUL: DOZERS</td>
<td>1260.00</td>
<td>700.00</td>
<td>560.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD &amp; HAUL: HAULERS</td>
<td>5550.00</td>
<td>3150.00</td>
<td>2400.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD &amp; HAUL: DUMPSITE</td>
<td>560.00</td>
<td>280.00</td>
<td>280.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD &amp; HAUL: TOTAL</td>
<td>11070.00</td>
<td>6230.00</td>
<td>4840.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COAL: CLEAN &amp; LOAD</td>
<td>350.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>350.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COAL: AUGER</td>
<td>140.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>140.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COAL: HAULING</td>
<td>1590.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1590.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVERHEAD PER TON</td>
<td>3700.24</td>
<td>2277.07</td>
<td>1423.17</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECLAMATION - SOURCE</td>
<td>1235.03</td>
<td>714.28</td>
<td>520.75</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>23085.27</td>
<td>11861.35</td>
<td>7123.92</td>
<td>0.00</td>
<td>4100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLOCK = MB2 TOTAL</td>
<td>T LIFT</td>
<td>2 LIFT</td>
<td>3 LIFT</td>
<td>X LIFT</td>
<td>R LIFT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLEAR AND BENCH</td>
<td>70.00</td>
<td>70.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRILL AND SHOOT</td>
<td>2295.00</td>
<td>1285.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1010.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CARRY AND PUSH</td>
<td>490.00</td>
<td>490.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD &amp; HAUL: LOADERS</td>
<td>1400.00</td>
<td>500.00</td>
<td>900.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD &amp; HAUL: DOZERS</td>
<td>490.00</td>
<td>490.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD &amp; HAUL: HAULERS</td>
<td>1800.00</td>
<td>750.00</td>
<td>1050.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD &amp; HAUL: DUMPSITE</td>
<td>210.00</td>
<td>210.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD &amp; HAUL: TOTAL</td>
<td>3900.00</td>
<td>1950.00</td>
<td>1950.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COAL: CLEAN &amp; LOAD</td>
<td>490.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>490.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COAL: AUGER</td>
<td>330.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>330.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COAL: HAULING</td>
<td>1355.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1355.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVERHEAD PER TON</td>
<td>1850.12</td>
<td>1138.54</td>
<td>711.59</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECLAMATION - SOURCE</td>
<td>866.39</td>
<td>501.00</td>
<td>365.39</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>11646.50</td>
<td>5434.54</td>
<td>3026.97</td>
<td>0.00</td>
<td>3185.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| LOAD & HAUL: LOADERS | 550.00 |
| LOAD & HAUL: HAULERS | 615.00 |
| LOAD & HAUL: TOTAL | 1165.00 |
| OVERHEAD PER TON | 341.56 |
| RECLAMATION - SOURCE | 138.58 |
| TOTAL | 1995.14 |

**PER BANK CUBIC YARD OVERBURDEN HANDLING COSTS**

**PER TON COAL HANDLING COSTS**

**TOTALS PER BANK CUBIC YARD**

<table>
<thead>
<tr>
<th>BLOCK = MB1</th>
<th>TOTAL</th>
<th>T LIFT</th>
<th>2 LIFT</th>
<th>3 LIFT</th>
<th>X LIFT</th>
<th>R LIFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEAR AND BENCH</td>
<td>0.01</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>DRILL AND SHOOT</td>
<td>0.35</td>
<td>0.32</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>CARRY AND PUSH</td>
<td>0.03</td>
<td>0.00</td>
<td>0.07</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LOAD &amp; HAUL: LOADERS</td>
<td>0.28</td>
<td>0.26</td>
<td>0.32</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LOAD &amp; HAUL: DOZERS</td>
<td>0.10</td>
<td>0.09</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LOAD &amp; HAUL: HAULERS</td>
<td>0.43</td>
<td>0.39</td>
<td>0.48</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LOAD &amp; HAUL: DUMPSITE</td>
<td>0.04</td>
<td>0.03</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LOAD &amp; HAUL: TOTAL</td>
<td>0.85</td>
<td>0.78</td>
<td>0.97</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>COAL: CLEAN &amp; LOAD</td>
<td>0.33</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.33</td>
<td>0.00</td>
</tr>
<tr>
<td>COAL: HAULING</td>
<td>1.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.50</td>
<td>0.00</td>
</tr>
<tr>
<td>OVERHEAD PER TON</td>
<td>0.28</td>
<td>0.28</td>
<td>0.28</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>RECLAMATION - SOURCE</td>
<td>0.10</td>
<td>0.09</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>TOTAL</td>
<td>1.78</td>
<td>1.48</td>
<td>1.42</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BLOCK = MB2</th>
<th>TOTAL</th>
<th>T LIFT</th>
<th>2 LIFT</th>
<th>3 LIFT</th>
<th>X LIFT</th>
<th>R LIFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEAR AND BENCH</td>
<td>0.01</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>DRILL AND SHOOT</td>
<td>0.35</td>
<td>0.32</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>CARRY AND PUSH</td>
<td>0.08</td>
<td>0.12</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LOAD &amp; HAUL: LOADERS</td>
<td>0.22</td>
<td>0.13</td>
<td>0.36</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LOAD &amp; HAUL: DOZERS</td>
<td>0.08</td>
<td>0.12</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LOAD &amp; HAUL: HAULERS</td>
<td>0.28</td>
<td>0.19</td>
<td>0.42</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LOAD &amp; HAUL: DUMPSITE</td>
<td>0.03</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LOAD &amp; HAUL: TOTAL</td>
<td>0.60</td>
<td>0.49</td>
<td>0.78</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>COAL: CLEAN &amp; LOAD</td>
<td>0.56</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.56</td>
<td>0.00</td>
</tr>
<tr>
<td>COAL: AUGER</td>
<td>1.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.10</td>
<td>0.00</td>
</tr>
<tr>
<td>OVERHEAD PER TON</td>
<td>1.56</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.56</td>
<td>0.00</td>
</tr>
<tr>
<td>RECLAMATION - SOURCE</td>
<td>0.13</td>
<td>0.13</td>
<td>0.15</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>TOTAL</td>
<td>1.79</td>
<td>1.36</td>
<td>1.21</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**PONY RUN**

**DAY 1 1999 THRU DAY 11 1999**

**FOLLOWING COSTS DEPEND UPON ACCURACY OF MOVED VOLUME ESTIMATES**

**HAULING COSTS PER BCY HAULED**

| BLOCK = | TOTAL | T LIFT | 2 LIFT | 3 LIFT | X LIFT | R LIFT |

---

*Note: The text contains a table with various cost calculations for different operations such as loading, hauling, and other handling costs. The table is divided into sections based on different blocks (MB1, MB2) and includes costs for clear and bench, drilling and shooting, carry and push, and various types of hauling and other associated costs. The text also mentions the period covered by the costs and notes that the following costs depend upon the accuracy of the moved volume estimates.*
<table>
<thead>
<tr>
<th>Block</th>
<th>Total</th>
<th>1 Lift</th>
<th>2 Lift</th>
<th>3 Lift</th>
<th>X Lift</th>
<th>R Lift</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1</td>
<td>1.42</td>
<td>0.00</td>
<td>1.42</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>MB2</td>
<td>0.33</td>
<td>0.33</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>JB</td>
<td>0.48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LOCATION BLOCKS AND INTERNAL BLOCK NUMBERS**

- MB1 1
- MB2 2
- MB3 3
- MB4 4
- MB5 5
- SD1 6
- SD2 7
- SD3 8
- SD4 9
- SD5 10
- Road 11

**SPECIAL BLOCKS**

- Road = Special Block
- All expenses treated as overhead

**OTHER TOTAL = 100.00**

**OVERHEAD ACCOUNT = 100.00**

**DESTINATION RECLAMATION TOTALS FROM INPUT**

<table>
<thead>
<tr>
<th>Source</th>
<th>Destin Exp</th>
<th>Source Exp</th>
<th>Undistributed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD1 T</td>
<td>785.00</td>
<td>714.28</td>
<td>580.72</td>
</tr>
<tr>
<td>SD1 R</td>
<td>510.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD2 T</td>
<td>475.00</td>
<td>78.61</td>
<td>365.39</td>
</tr>
<tr>
<td>SD2 R</td>
<td>470.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL BEFORE REDISTRIBUTION = 2240.00**

**DISTRIBUTION OF RECLAMATION EXPENSES FROM DESTINATION TO SOURCE**

<table>
<thead>
<tr>
<th>Source</th>
<th>Frac Destin</th>
<th>Destin Exp</th>
<th>Source Exp</th>
<th>Undistributed</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB1 T</td>
<td>0.55157 SD1</td>
<td>1295.00</td>
<td>714.28</td>
<td>580.72</td>
</tr>
<tr>
<td>MB2 T</td>
<td>0.55016 SD2</td>
<td>945.00</td>
<td>78.61</td>
<td>365.39</td>
</tr>
<tr>
<td>MB3 T</td>
<td>0.08319 SD2</td>
<td>945.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MB2 T</td>
<td>0.38665 SD2</td>
<td>945.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL REDISTRIBUTION = 2240.00**

**TOTAL AFTER REDISTRIBUTION = 2240.00**

**PONY RUN DAY 1 1999 THRU DAY 11 1999**

**DISTRIBUTE OVERHEAD COSTS**

- Overhead total before per ton total added = 100.00
- Overhead total after per ton total added = 5891.93
- Overhead total after adding lift overhead = 5891.93
- Overhead per BCY = 0.28
APPENDIX H: EXEC FILE AND SUBROUTINES

*  *  *  COSTSUM EXEC  *  *

&TRACE ON
&ERROR &EXIT
FILEDEF 03 DISK DATA     COSINPUT
FILEDEF 06  TERM
FILEDEF 10 DISK HAULER   INDATA
FILEDEF 11 DISK LOADER   INDATA
FILEDEF 12 DISK DOZER    INDATA
FILEDEF 13 DISK DRILL    INDATA
FILEDEF 14 DISK OTHER    INDATA
FILEDEF 31 DISK HRSOURCE HAULER
FILEDEF 32 DISK LDSOURCE HAULER
FILEDEF 33 DISK LOADDEST HAULER
FILEDEF 34 DISK HOURDEST HAULER
FILEDEF 35 DISK LDSPERHR HAULER
FILEDEF 45 DISK OPHOURS LOADER
FILEDEF 46 DISK DSHPERHR LOADER
FILEDEF 47 DISK CDHPERHR LOADER
FILEDEF 55 DISK OPHOURS DOZER
FILEDEF 56 DISK DHPERHR DOZER
FILEDEF 57 DISK PDPERHR DOZER
FILEDEF 60 DISK TOTALS DRILL
FILEDEF 70 DISK TOTALS OTHER
FILEDEF 71 DISK DATA     MOVINPUT
FILEDEF 72 DISK BYSOURCE MOVE
FILEDEF 73 DISK BYDEST    MOVE
FILEDEF 74 DISK COSINPUT  MOVE
FILEDEF 75 DISK HRSOURCE MOVE
FILEDEF 92 DISK REDIST   COST
FILEDEF 93 DISK SUMMARY  COST
FILEDEF 94 DISK PERBLOCK COST
FILEDEF 95 DISK PERYARD  COST
GLOBAL TXTLIB VFORLIB
GLOBAL LOADLIB VFLODLIB
LOAD XXXXX
START

*  *  *  SUBROUTINES  *  *

SUBROUTINE ABLOCK (BLK,BDIM)
*  *
*  THIS SUBROUTINE ASSIGNS SYMBOLS TO THE ARRAY "BLK" AND RETURNS
*  THEM TO THE MAIN PROGRAM. THE SYMBOLS ARE REPRESENTATIONS OF
*  MINING BLOCKS AND OTHER MINE SITE AREAS. "BDIM" IS THE DIMENSION
*  OF THE "BLK" ARRAY.
*  *
*  POSITIONS IN "BLK" NOT LOADED WITH SPECIFIC SYMBOLS ARE LOADED
*  WITH BLANK CHARACTER STRINGS.
*  *
INTEGER BDIM
CHARACTER*4 BLK(BDIM)
INTEGER KKK
BLK(01) = 'MB1'
BLK(02) = 'MB2'
BLK(03) = 'MB3'
BLK(04) = 'MB4'
BLK(05) = 'MB5'
BLK(06) = 'SD1'
BLK(07) = 'SD2'
BLK(08) = 'SD3'
BLK(09) = 'SD4'

126
SUBROUTINE KBLOCK (BLK, BDIM, KBLK, KK)

* THIS SUBROUTINE Assigns A Value To "KK" Corresponding To The
  Position Held In Array "BLK" By The Mining Block Symbol "KBLK".
  "BDIM" Is The Dimension Of The Array "BLK". The Array Must Be
  initialized Before This Subroutine Is Used, With ' ' Being
  Loaded Into Non-Specific Positions. If The Subroutine Encounters
  A Symbol Not Held In Array "BLK", It Will Assign That Symbol To
  First Available ' ' Position. If The Array Overloads, An Error
  Message Is Issued To Output Unit "06" And It Returns To The Main
  Program.

INTEGER BDIM
CHARACTER*4 BLK(BDIM), KBLK
INTEGER KK

10 IF (BLK(KK) .EQ. KBLK) GOTO 97
IF (KBLK .EQ. BLK(KK)) GOTO 97
IF (BLK(KK) .EQ. ' ') THEN
  BLK(KK) = KBLK
  GOTO 97
ENDIF
KK = KK + 1
IF (KK .GT. BDIM) THEN
  WRITE (06,*) 'KBLOCK ERROR KK = ', KK, 'KBLK = ', KBLK
  GOTO 97
ENDIF
GOTO 10
97 RETURN
98 STOP
99 END

SUBROUTINE ALIFT (LIF, LL)

* THIS SUBROUTINE Assigns Lift Designating Characters
  To The Array (LIF)

INTEGER LL
CHARACTER*1 LIF(LL)
LIF(1) = 'T'
LIF(2) = '2'
LIF(3) = 'X'
LIF(4) = 'R'
LIF(5) = '4'
12 RETURN
13 STOP
END

SUBROUTINE JLIFT (LIF, L)

* THIS SUBROUTINE Assigns A Value To "J" For Use By The Main
  Program, Which Corresponds To The Lift Symbol "LIF".

CHARACTER*1 LIF
INTEGER L
IF (CLIF .EQ. 'X') THEN
  L = 4
  GOTO 15
ENDIF
IF (CLIF .EQ. 'T') THEN
  L = 1
  GOTO 15
ENDIF
IF (CLIF .EQ. 'C') THEN
  L = 4
  GOTO 15
ENDIF
IF (CLIF .EQ. '2') THEN
  L = 2
  GOTO 15
ENDIF
IF (CLIF .EQ. '3') THEN
  L = 3
  GOTO 15
ENDIF
IF (CLIF .EQ. 'I') THEN
  L = 5
  GOTO 15
ENDIF
IF (CLIF .EQ. 'l') THEN
  L = 1
  GOTO 15
ENDIF
IF (CLIF .EQ. 'R') THEN
  L = 5
  GOTO 15
ENDIF
IF (CLIF .EQ. '4') THEN
  L = 4
  GOTO 15
ENDIF
WRITE(06,*) 'JLIFT ERROR J=', L, 'LIF=', LIF
L = 5
15 RETURN
17 STOP
END

SUBROUTINE ALDOP (LDOP, LDIM)

THE PURPOSE OF THIS SUBROUTINE IS TO LOAD SYMBOLS FOR THE LOAD
OPERATIONS INTO THE ARRAY "LDOP" AND TO PASS THAT ARRAY
TO THE MAIN PROGRAM. BLANKS ARE LOADED INTO ARRAY ELEMENTS
NOT LOADED WITH SPECIFIC SYMBOLS. "LDIM" IS THE DIMENSION OF
THE "LDOP" ARRAY.

INTEGER LDIM, I
CHARACTER*2 LDOP(LDIM)
LDOP(1) = 'ST'
LDOP(2) = 'PD'
LDOP(5) = 'L1'
LDOP(4) = 'L2'
LDOP(5) = 'CD'
LDOP(6) = 'DS'
LDOP(7) = 'LC'
LDOP(8) = 'RD'
LDOP(9) = '0'
I = 9
15 I = I + 1
LDOP(I) = ' ' IF (I .LT. LDIM) GOTO 15
16 RETURN
17 STOP
END
SUBROUTINE XLDOP(LDOP,LDIM,OPER,X)
* THE PURPOSE OF THIS SUBROUTINE IS TO PASS VALUES FOR "X"
* BACK TO THE MAIN PROGRAM. "X" REPRESENTS THE POSITION
* IN THE ARRAY "LDOP" HELD BY THE SYMBOL "OPER". "LDIM" IS THE
* SIZE OF THE "LDOP" ARRAY. IF "OPER" IS NOT PRESENT AS A SYMBOL
* IN "LDOP", IT IS LOADED INTO THE FIRST BLANK POSITION.

INTEGER LDIM,I,X
CHARACTER*2 LDOP(LDIM),OPER
X = 0
37 X = X + 1
IF (OPER .EQ. LDOP(X)) GOTO 77
IF (LDOP(X) .EQ. ' ') THEN
   LDOP(X) = OPER
   GOTO 77
ENDIF
IF (X .LT. LDIM) GOTO 37
WRITE (06,*) 'LDOP ERROR; X',X,'OPER = ',OPER
77 RETURN
78 STOP
END

SUBROUTINE ADZOP(DZOP,DZDIM)
* THE PURPOSE OF THIS SUBROUTINE IS TO LOAD SYMBOLS FOR THE DOZER
* OPERATIONS INTO THE ARRAY "DZOP" AND TO PASS THAT ARRAY
* TO THE MAIN PROGRAM. BLANKS ARE LOADED INTO ARRAY ELEMENTS
* NOT LOADED WITH SPECIFIC SYMBOLS. "DZDIM" IS THE DIMENSION OF
* THE "DZOP" ARRAY.

INTEGER DZDIM, I
CHARACTER*2 DZOP(DZDIM)
DZOP(1) = 'ST'
DZOP(2) = 'PD'
DZOP(3) = 'FD'
DZOP(4) = 'PO'
DZOP(5) = 'DS'
DZOP(6) = 'RR'
DZOP(7) = 'HB'
DZOP(8) = 'HF'
DZOP(9) = 'HT'
DZOP(10) = 'HD'
DZOP(11) = 'CT'
DZOP(12) = 'O'
I = 12
15 I = I + 1
   DZOP(I) = '
IF (I .LT. DZDIM) GOTO 15
16 RETURN
17 STOP
END

SUBROUTINE XDZOP(DZOP,DZDIM,OPER,X)
* THE PURPOSE OF THIS SUBROUTINE IS TO PASS VALUES FOR "X"
* BACK TO THE MAIN PROGRAM. "X" REPRESENTS THE POSITION
* IN THE ARRAY "DZOP" HELD BY THE SYMBOL "OPER". "DZDIM" IS THE
* SIZE OF THE "DZOP" ARRAY. IF "OPER" IS NOT PRESENT AS A SYMBOL
* IN "LDOP", IT IS LOADED INTO THE FIRST BLANK POSITION.

INTEGER DZDIM,I,X
CHARACTER*2 DZOP(DZDIM), OPER
    X = 0
37   X = X + 1
    IF (OPER .EQ. DZOP(X)) GOTO 77
    IF (DZOP(X) .EQ. ',') THEN
        DZOP(X) = OPER
        GOTO 77
    ENDIF
    IF (X .LT. DZDIM) GOTO 37
    WRITE (06,*) 'DZOP ERROR; X = ',X, 'OPER = ',OPER
77   RETURN
78   STOP
END
Virginia's Agricultural Experiment Stations

1 —— Blacksburg
   Virginia Tech
   Main Station

2 —— Steeles Tavern
   Shenandoah Valley Research Station
   Beef, Sheep, Fruit, Forages, Insects

3 —— Orange
   Piedmont Research Station
   Small Grains, Corn, Alfalfa, Crops

4 —— Winchester
   Winchester Fruit Research Laboratory
   Fruit, Insect Control

5 —— Middleburg
   Virginia Forage Research Station
   Forages, Beef

6 —— Warsaw
   Eastern Virginia Research Station
   Field Crops

7 —— Suffolk
   Tidewater Research and Continuing Education Center
   Peanuts, Swine, Soybeans, Corn, Small Grains

8 —— Blackstone
   Southern Piedmont Research and Continuing Education Center
   Tobacco, Horticulture Crops, Turfgrass, Small Grains, Forages

9 —— Critz
   Reynolds Homestead Research Center
   Forestry, Wildlife

10 —— Glade Spring
   Southwest Virginia Research Station
   Burley Tobacco, Beef, Sheep

11 —— Hampton
   Seafood Processing Research
   and Extension Unit
   Seafood