# TABLE OF CONTENTS

1. INTRODUCTION  
   WHAT IS CASHII  
   WHY USE CASHII  
   SAMPLE CASHII RUN  
   HOW TO USE THE REST OF THE BOOK  

2. CASHII BASICS  
   GETTING STARTED  
   CREATING A DATA FILE  
   SECTION 1  
   SECTION 2 (COST ELEMENTS)  
   SECTION 3 (TRANSACTIONS)  
   EXECUTING CASHII  

3. CASHII OUTPUT  
   TRANSACTION REPORT  
   ANALYTICAL REPORT  
   RECAP OF COST STATEMENT REPORT  

4. CASHII TERMS, DEFINITIONS & SYNTAX  
   TERMS AND DEFINITIONS  
   ERROR MESSAGES  

5. APPENDICES  
   SAMPLE RUN USING GEISCO  
   SAMPLE RUN USING BCS  
   REFERENCE MATERIAL
1. INTRODUCTION

WHAT IS CASHII

CASHII is a FORTRAN program which is available to be run on contractors' own computers. Of course, even though the program is written as generically as possible, it is possible that some conversion effort would be required in order to make it execute correctly. Therefore, working versions of the program are maintained on two commercial computer service companies, General Electric Information Services and Boeing Computer Services. Contractors may elect to use the program on either of these companies. Contracting Officers use the program on the Copper Impact System (BCS). Contractors are not permitted to access the Copper Impact System. However, every effort is made to insure that the program is the same in all places.

Each vendor has identified a contact to assist in initial setup of the CASHII system. For BCS the name and address is:

M. Rosenberg
BCS
7980 Gallows Road
Vienna, Va 22180
(703) 821-6200

For GEISCo contact:
B. Clarke
GEISCo
1300 N. 17th Street - Suite 1100
Arlington, Va. 22209
(703) 276-4046

CASHII is the name of a computer program developed by the Department of Defense. Contractors who request flexible progress payments must use CASHII to provide the relevant calculations on flexible payment rate and contractor investment ratio. This data must also be supplied to contracting officers who use CASHII to verify the contractors data.
WHY USE CASHII

The following is excerpted from the Defense Acquisition Regulations Appendix E Part 5: E-530.1 Background and Policy. Paying progress payments assists in financing a contractor’s performance and reduces the contractor’s investment in its work in process inventory. The actual investment held by a contractor in work in process inventory is influenced by a number of factors in addition to progress payments, such as delivery schedules, cash management practices, and Government payment practices. Progress payment amounts that are determined by using uniform, standard progress payment rates (i.e., 90% or 95% under E-503) are insensitive to these other factors influencing investment and, as a consequence, result in investments by contractors in work in process inventory that vary among contractors and across contracts; on the other hand, flexible progress payment rates (expressed as a percentage that will be applied to costs to determine the amount payable as a progress payment rate) are designed to tailor more closely the progress payment rate to the cash needs for financing performance of a particular contractor for a given contract.

For flexible progress payments, cash needs are measured and projected in relation to investment underlying the work in process inventory over the life of the contract. Total investment is measured by a weighted average of total costs paid by the contractor to complete performance of the contract, and the contractor’s investment is the weighted average of the amount not paid by the Government. The DOD, as a matter of policy, has concluded that a contractor should retain at least a 5% investment in work in process inventory over the life of the contract. Accordingly, the DOD will make progress payments at a rate (expressed as a whole number) that is the highest rate which yields a corresponding investment by the contractor in work in process inventory of not less than 5%. This progress payment rate is to be determined by the DOD Cash Flow Computer Model. In no event will the progress payment rate be greater than 100% or less than the uniform, standard progress payment rate that would have been applied to the contract absent flexible progress payments.
1. INTRODUCTION CONT’D

WHY USE CASHII

Contracting officers shall use a flexible progress payment rate in lieu of uniform, standard progress payment rates for the purpose of computing monthly progress payments if requested by the contractor, if the criteria in E-530.3 are met, and if the contractor agrees to the requirements of this section. Flexible progress payments shall be regarded as customary progress payments, as defined in E-503.

E-530.2 DOD Cash Flow Computer Model. The flexible progress payment rate shall be determined through application of the DOD Cash Flow Computer Model, available to contracting officers on the COPPER IMPACT computer time sharing network under the computer file name "CASHII". The model takes into account key cash flow factors, such as contract cost profile, delivery schedules, subcontractor progress payments, liquidation rates, and payment/reimbursement cycles. Operating instructions and cash flow data requirements are retrievable within the model in a conversational mode. Contractors may obtain copies of the DOD Cash Flow Computer Model User’s Guide (including the related computer program) from the Defense Technical Information Center, Building 5 Cameron Station, Alexandria, Va 22314. Contracting officers may not grant contractor access to Government leased COPPER IMPACT time sharing computer network.

For negotiated fixed price type contracts over $1 million, flexible progress payments may be authorized if requested by the contractor. The flexible progress payments may range from 90% to 100% as determined by the cash flow model which is the subject of this document.
Figures 1, 2 and 3 illustrate a simple run of the program. Figure 1 shows that the program always starts by asking you to answer six questions. These questions are asked at the terminal in order to provide you maximum flexibility. The rest of the data required by CASHII is in a file which the program reads. You are asked to supply the filename even before the six questions shown here. The first four questions are asked at the terminal so that the information contained in the data file can be kept constant while different answers are supplied here. This capability provides some "what if" analysis through CASHII.

The last two questions are asking how much detail you wish to see for output. A recap of the cost report data reformats and prints the data found in the input file. Printing this data can be very useful in verifying that the system is reading the data you think you typed.

A transaction report is a detailed report showing one line for every transaction that takes place. The system inserts transactions for planned progress payments. This enables you to analyze exactly how cash flow is effected by different assumptions.

Both of these reports are discussed in detail in section 3 of this document. For now, however let's continue with the sample output shown in Figure 2. It is the Analytical Report. The Analytical Report is always produced, no matter how you answer the preceding two questions.
THE FOLLOWING DATA IS REQUESTED OF THE USER AT THE TERMINAL SO DIFFERENT CALCULATIONS CAN BE MADE USING THE SAME DATA FILE.

WHAT IS THE PROGRESS PAYMENT LAG TIME IN DAYS? (TIME FROM WHEN STATEMENT IS ISSUED UNTIL PAYMENT IS RECEIVED?)

>5

WHAT IS THE DELIVERY PAYMENT LAG TIME IN DAYS? (TIME FROM WHEN DELIVERY IS MADE UNTIL PAYMENT IS RECEIVED.)

>20

WOULD YOU LIKE TO SPECIFY A CERTAIN PROGRESS PAYMENT RATE FOR THESE CALCULATIONS? A NO ANSWER IMPLIES YOU WANT THE COMPUTER TO CALCULATE THE FLEXIBLE PROGRESS RATE USING THE CONTRACTOR INVESTMENT RATIO. (YES / NO)

>NO

AFTER HOW MANY DAYS WOULD YOU LIKE THE ALTERNATE LIQUIDATION RATE USED IN CALCULATIONS? VALUE MUST BE GREATER THAN OR EQUAL TO 365.

>365

WOULD YOU LIKE TO SEE A TRANSACTION REPORT? (YES / NO)

>NO

WOULD YOU LIKE TO SEE A RECAP OF THE COST REPORT DATA? (YES / NO)

>NO

FIGURE 1

-5-
PROJECT TITLE: CASH EXAMPLE WITH PROJECTED DATA

************************************************************

CONTRACT PRICE: 7700.00
TOTAL CONTRACTOR COST: 7000.00
PROFIT PAID TO CONTRACTOR: 700.00
START UP COST OF CONTRACT: 0.00
CONTRACT START DATE: 820101
DATA FILE PROFIT PERCENTAGE: 10.00
COMPUTED PROFIT PERCENTAGE: 10.00

************************************************************

**** PROGRESS PAYMENT RATE DATA****

PROGRESS PAYMENT RATE: 98.00
ORDINARY LIQUIDATION RATE: 98.00
ALTERNATE LIQUIDATION RATE: 89.09
CONTRACTOR INVESTMENT RATIO: 5.75

************************************************************

COST ELEMENT DATA

# NAME UNPAID COST PAID PAID COST ELIGIBILITY DATES FLOAT DATES FLOAT PROGRESS PAYMENT

********************************************************************************

1 DIRECT LABOR -2 INCURRED COST
2 OVERHEAD 15 INCURRED COST
3 SUBCONTRACT PROG PAY 20 98.0 -2 SUBCONTRACT PROG PAY
4 MATERIAL 25 98.0 -2 PAID COST
5 PURCHASED PARTS 20 98.0 -1 PAID COST
6 INTERDIVISIONAL 15 98.0 -4 PAID COST

********************************************************************************

PROGRESS PAYMENT LAG TIME IN DAYS: 5
DELIVERY PAYMENT LAG TIME IN DAYS: 20
DATE OF FINAL DELIVERY: 830525

PROGRESS PAYMENT RATE COMPUTED BY PROGRAM: 98.00

********************************************************************************

FIGURE 2

-6-
PROJECT TITLE CASH EXAMPLE WITH PROJECTED DATA

*******************************************************************************
CONTRACT PRICE 7700.00
TOTAL CONTRACTOR COST 7100.00
PROFIT PAID TO CONTRACTOR 700.00
START UP COST OF CONTRACT 0.00
CONTRACT START DATE 829191
DATA FILE PROFIT PERCENTAGE 10.00
COMPUTED PROFIT PERCENTAGE 10.00
*******************************************************************************

### PROGRESS PAYMENT RATE DATA###

PROGRESS PAYMENT RATE # 98.00 #
ORDINARY LIQUIDATION RATE # 98.00 #
ALTERNATE LIQUIDATION RATE # 89.09 #
CONTRACTOR INVESTMENT RATIO # 5.42 #

*******************************************************************************

COST ELEMENT DATA

<table>
<thead>
<tr>
<th>#</th>
<th>NAME</th>
<th>UNPAID COST</th>
<th>PAID COST</th>
<th>ELIGIBILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DIRECT LABOR</td>
<td>-2</td>
<td>INCURRED COST</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>OVERHEAD</td>
<td>15</td>
<td>INCURRED COST</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SUBCONT PROC PAY</td>
<td>28</td>
<td>80.0</td>
<td>-2</td>
</tr>
<tr>
<td>4</td>
<td>MATERIAL</td>
<td>28</td>
<td>80.0</td>
<td>-2</td>
</tr>
<tr>
<td>5</td>
<td>PURCHASED PARTS</td>
<td>28</td>
<td>90.0</td>
<td>-1</td>
</tr>
<tr>
<td>6</td>
<td>INTERDIVISIONAL</td>
<td>15</td>
<td>90.0</td>
<td>-4</td>
</tr>
</tbody>
</table>

*******************************************************************************

PROGRESS PAYMENT RATE COMPUTED BY PROGRAM 98.00

*******************************************************************************
The first block of information in Figure 2 is a combination of program calculations and facts contained in the input file. Total contractor cost is the sum of all cost elements contained in the file. Profit paid is total cost subtracted from contract price. Contract price is derived by adding all the deliverables. Profit paid and contractor cost are then used to calculate profit percentage. The calculated profit percentage is then compared to the percentage specified in the datafile. If the two percentages are not the same, an error message is printed and the program stops. Start up cost and contract start time are simply read in from the input file.

Now the program calculates the progress payment rate, the alternate liquidation rate and the contractor investment ratio. Ordinary liquidation rate is always the same as the progress payment rate. These four numbers are the real meat of this report. The "what if" analysis is performed to see the effect on them.

The rest of the numbers shown in Figure 2 are simply being reported from the input file. You should verify that these numbers are what you intended. If not, the input file should be changed. Also included here are the values that you input to be used as the progress payment and delivery payment time lags.

To complete our initial look at the CASHII program let's turn to Figure 3. It shows the input file that was used to generate Figure 2.
CASH EXAMPLE WITH PROJECTED DATA

SECTION 1

DIRECT LABOR: INCMR: $20,000
OVERHEAD: INCMR: $15,000
SUBCMPl. PROC PAT SUBC: $20,000
MATERIAL PAID: $35,000
PURCHASED PARTS: PAID: $20,000
INTERDIVISIONAL: PAID: $15,000

SECTION 2

3,820,100
3,820,200
3,820,300
4,820,400
3,820,500
3,820,600
3,820,700
3,820,800
4,820,900
3,821,000
3,821,100
4,821,200
3,821,300
3,821,400
4,821,500
3,821,600
3,821,700
4,821,800
3,821,900
4,821,900
3,830,000
3,830,100
4,830,200
3,830,300
3,830,400
4,830,500

SECTION 3

3,812,000
3,812,100
4,812,200
3,812,300
3,812,400
4,812,500
3,812,600
3,812,700
4,812,800
3,812,900
4,812,900
3,831,000
3,831,100
4,831,200
3,831,300
3,831,400
4,831,500
3,831,600
3,831,700
4,831,800
3,831,900
4,831,900

FIGURE 3
The input data file is actually made up of three sections. A detailed discussion of the information contained in each is found in the next section of this document. For now, the first two lines contain information about the rest of the file and the title of the project. In this example, the next six lines contain information about the individual cost elements that will make up the contract expenses. (You may have up to 10 lines of cost elements but you have to specify how many you have.)

The rest of the file contains the transactions that take place over the life of the contract. A transaction type of 3 means a monthly cost statement (either actual or estimated), the number after the 3 is the date and the following six numbers are the actual (or estimated) costs for each of the cost elements specified above. A transaction type 4 is a planned delivery. The number after the 4 is the date and the next number is the value of the deliverable.

Now you know nearly everything necessary to use the CASH11 program. If your contract used the six cost elements illustrated in this example, the only thing we haven't discussed is the rest of the numbers on the cost element line. Each of them is discussed in the next section.
HOW TO USE THE REST OF THIS BOOK

If you have read this far you have a good fundamental understanding of how CASHII works. The next step depends on how you will be using it. If you plan to make extensive use of the system you should study sections 2, 3 and the appropriate appendix. If you feel you have already seen enough, you may be able to squeak by with section 2 and the appropriate appendix. Section 2 is needed by everyone since it describes in detail what the items in the data file are. You may find that you can skip much of it simply by referring to section 4, the short guide to syntax for specific questions or rules and limitations.

Section 4 will probably prove useful to browse through. However, it is not intended to be read from start to finish. Its function is that of a dictionary for the CASHII system. The last part of it addresses error messages. If when you execute CASHII, unexpected conditions occur you may be able to obtain help by looking there.

Finally, the appendix which is most relevant will depend on exactly how you are using CASHII. The program itself is maintained on two commercial computer service vendors. One appendix demonstrates how CASHII works on each. CASHII is also available for your own in-house computer. (CASHII is written in FORTRAN) If that's where you are using it most of this book is useful. However, the appendices of sample runs may not be of any value.
2. CASHII BASICS

GETTING STARTED

First you must decide where you will be using CASHII. One important consideration is who will be using it (actually entering data and executing the program). If it is someone with no previous computer experience you should look for a "user friendly" system. That means a computer system which is as simple as possible to operate and spends as little time as possible with computerese. You should also plan on at least some introductory training in the particular computer system and how it works. (This book should be all you need on the application, if it's not, there is an on-line teaching tool which should help.)

If the typical users will be computer professionals none of the above is of particular concern. You will probably want to get your own copy of the program and run it on your own in-house computer. Some regular system should be set up to insure that you are kept up-to-date with the latest version. If you elect to use your own in-house system, skip to Creating the Data File. The rest of this discussion is for those who will be running the program on an outside vendor and who have no particular computer experience.

The next step is access to one of the commercial vendors. DOD does not provide this access. You will have to make arrangements directly with the vendor to become a customer. It's important that you inform the vendor that you wish to use the CASHII system. They may have some special arrangements that must be made.

GOVERNMENT Note for government users: Your access to CASHII is through Copper Impact on Boeing Computer Services only.

USERS NOTE
2. CASHII BASICS CONT'D

---

GETTING STARTED

---

After picking your vendor, you will want to discuss with them the type of terminal device necessary to access their computer. The key consideration is how much time you will spend using CASHII. If you expect to spend a substantial amount of time using CASHII then a higher speed terminal is a good investment. For your purposes terminals operate at 30 or 120 characters per second (CPS). A full set of output for the sample problem presented in section 2 takes about 10 minutes to print at 30 CPS. If you anticipate doing one similar run per week then 30 CPS should be adequate. For one a day or for contracts that have much more data than that illustrated 120 CPS is preferable and possibly even more economical. Your vendor can help you determine more accurately what volume will make the higher speed devices the most economical.

Of course, if you already have a terminal chances are good that it will be compatible with the computer you select. If you elect to attend an introductory training session on the vendor's computer service, you should ask that specific discussion be devoted to your terminal and how it works.

That should be all it takes to get started. Now you are ready to enter data. That's the next topic.
CREATING A DATA FILE

We have already had a brief discussion of the data file. You know what it looks like and that it is divided into three sections. Now we will discuss each of the sections in sufficient detail to enable you to write down your data in the proper format. Note that when you are preparing to run the CASHII program, you must have entered (and saved) the appropriate data into the data file. That's the first step. Then CASHII reads the information from the file. The actual steps of entering the data file are shown in the appropriate appendix. You will see as we go along that it makes sense to write the data on paper in the proper format first.

The first line in our example is:

820101,0,6,22,.1

A A A A

Planned profit percent as a decimal.
This example plans on 10% profit.
The number of transaction lines in this data file (more on this in the discussion of section 3).
The number of cost elements associated with this contract. Maximum of 10 allowed.
Start up costs.
Start up date (year, month, day format)

The second line is:

CASHII EXAMPLE WITH PROJECTED DATA

This is simply the title that we wish to print at the top of each of the reports. You can have any title you wish up to 50 characters long.
CREATING A DATA FILE

That completes the first section of the file. The second section is where you specify each of the cost elements which will be involved with the contract. Each cost element requires that you specify the type of cost (as specified in the progress payment clause of the contract) followed by three items of numeric data.

The type of cost must be INCUR, PAIDC or SUBCT.

A typical cost element line (as shown in our earlier example) is:

DIRECT LABOR, INCUR, -2, 0, 0

| Dummy elements for INCUR must be 0. |
| Days float |
| Type of cost |
| Title (maximum of 20 characters) |

INCUR means that expenses associated with this cost element are eligible for progress payment as they are incurred.

PAIDC means that expenses are eligible for progress payments only as they are actually paid.

SUBCT means that this cost element represents progress payments paid to a subcontractor and are eligible for 100% reimbursement to the prime contractor.

Important note for large business firms: Some contract costs such as material or purchased parts costs incurred are not eligible for progress payments until actually paid. Therefore, these elements must be coded as PAIDC.
2. CASHII BASICS CONT'D
------------------------

CREATING A DATA FILE
------------------------

FLOAT is the next item on the line. Slightly different considerations are necessary depending on whether the cost element is INCUR, PAIDC or SUBCT. Following the example, let's look at INCUR first.

FLOAT is the number of days between the cost statement date and the time the cost is paid on a cash basis. Bank lag times must be included in float. Both positive and negative numbers may be entered.

Float for labor should be based on a weighted average over the cost accounting period. For example, if the cost accounting period is 4 weeks and the contractor pays every 2 weeks, and it takes an average of 8 days from the time the payroll is closed until the payroll checks clear the contractor's bank account, the average float for the 4 week period would be 1 day, computed as follows:

<table>
<thead>
<tr>
<th>PAYROLL PERIOD</th>
<th>FIRST</th>
<th>SECOND</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAY PAYROLL CLOSES</td>
<td>14</td>
<td>28</td>
</tr>
<tr>
<td>PLUS: PAYMENT LAG</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>PAYROLL PAYMENT DATE</td>
<td>22</td>
<td>36</td>
</tr>
<tr>
<td>LESS: DAY OF COST STATEMENT</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>FLOAT</td>
<td>-6</td>
<td>+8</td>
</tr>
<tr>
<td>TIMES: PAYROLL WEIGHT*</td>
<td>.5</td>
<td>.5</td>
</tr>
<tr>
<td>WEIGHTED FLOAT</td>
<td>-3</td>
<td>+4</td>
</tr>
</tbody>
</table>

AVERAGE FLOAT (-3+4) = 1 DAY

*Expected dollar amount is the same for each payroll period so they are equally weighted. The average float is then the sum of the two individual float amounts.
2. CASHII BASICS CONT’D

CREATING A DATA FILE

The next cost element is also type INCUR. It is:

OVERHEAD, INCUR, 15, 0, 0

This follows exactly the same format as we just discussed. However, calculating float for overhead items is intricate enough that we should step through a specific example.

Overhead will include things like indirect labor, vacation, depreciation, materials and supplies. These things will typically have very different float times so it is necessary to calculate a weighted average. The example below illustrates the calculation.

<table>
<thead>
<tr>
<th>ELEMENT OF EXPENSE</th>
<th>AMOUNT</th>
<th>PERCENTAGE</th>
<th>FLOAT DAYS</th>
<th>WEIGHTED FLOAT DAYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indirect labor</td>
<td>$1,700</td>
<td>17.0%</td>
<td>-2</td>
<td>-.34</td>
</tr>
<tr>
<td>Payroll taxes</td>
<td>1,500</td>
<td>15.0</td>
<td>-6</td>
<td>-.90</td>
</tr>
<tr>
<td>Holidays</td>
<td>600</td>
<td>6.0</td>
<td>29</td>
<td>1.74</td>
</tr>
<tr>
<td>Sick Pay</td>
<td>400</td>
<td>4.0</td>
<td>30</td>
<td>1.20</td>
</tr>
<tr>
<td>Accrued vacation</td>
<td>1,000</td>
<td>10.0</td>
<td>60</td>
<td>6.00</td>
</tr>
<tr>
<td>Retirement</td>
<td>1,500</td>
<td>15.0</td>
<td>42</td>
<td>6.30</td>
</tr>
<tr>
<td>Insurance</td>
<td>500</td>
<td>5.0</td>
<td>-15</td>
<td>-.75</td>
</tr>
<tr>
<td>Depreciation-Plt&amp;Equip</td>
<td>1,200</td>
<td>12.0</td>
<td>0</td>
<td>.00</td>
</tr>
<tr>
<td>Travel</td>
<td>300</td>
<td>3.0</td>
<td>9</td>
<td>.27</td>
</tr>
<tr>
<td>Materials &amp; Supplies</td>
<td>600</td>
<td>6.0</td>
<td>20</td>
<td>1.20</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>700</td>
<td>7.0</td>
<td>4</td>
<td>.28</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>$10,000</td>
<td><strong>100.0%</strong></td>
<td><strong>15.00</strong></td>
<td></td>
</tr>
</tbody>
</table>
CREATING A DATA FILE

Note the significant number of float days associated with vacation. Vacation expense is accrued long before it is actually paid. Since we are calculating a weighted average the overall effect is consistent with its importance to total expenses. Also notice that depreciation float is shown as zero. Facilities capital cost of money should also be treated as having no float.

Weighted float days is percent of total times float days. The weighted float days are then added and the total used as the float time for this cost element.

Since only ten cost elements are allowed, it makes sense to combine similar cost elements in this manner whenever necessary.

The next line in our data file illustrates a SUBCT expense. The format for SUBCT and PAIDC cost elements is the same so we will discuss them together.
DAIDC and SUBCT COST ELEMENTS

Let's start by showing the example we are working with:

SUBCONT PROG PAY, SUBCT, 20, .8,-2

A A A A A

<table>
<thead>
<tr>
<th>Float for paid costs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent of cost paid in current cost accounting period.</td>
</tr>
<tr>
<td>Float for unpaid costs.</td>
</tr>
<tr>
<td>Type of cost element.</td>
</tr>
<tr>
<td>Cost element.</td>
</tr>
</tbody>
</table>

We can deal with the two float numbers with an example (which will be equally applicable to PAIDC cost elements) so let's discuss the percent of cost paid first. This number is the portion of costs incurred during the accounting period for which the contractor has already issued checks. This cost is eligible for progress payments. Note, the number must be expressed as a decimal. In this case, the contractor has paid 80% of the expenses during the accounting period.

The first float item is specified as float for unpaid costs. This is the number of days after the cost statement is made that the expense is paid. The second float item is float for paid costs. This is the number of days before the cost statement is made that this category of expenses is paid. Calculations for both these numbers can be illustrated in one example.
2. CASHII BASICS CONT'D

CREATING A DATA FILE

The contractor made a review of invoices received during April to develop these data. Material costs are recorded on an accrual basis. To keep the example simple, assume that invoices are paid 20 days after receipt. Invoices selected for this computation should cover a period of time sufficient to represent the contractor’s normal dollar volume and payment cycle for such purchases. Cost elements involving an insignificant amount in relation to the contractor’s normal volume need not be analyzed.

SAMPLE MONTH - APRIL

<table>
<thead>
<tr>
<th>DATE</th>
<th>INVOICE RECEIVED</th>
<th>INVOICE CHECK</th>
<th>BANK START OF APR</th>
<th>DAYS FROM (DAYS FROM START OF APR)</th>
<th>TOTAL DOLLARS</th>
</tr>
</thead>
<tbody>
<tr>
<td>APR 1</td>
<td>$100</td>
<td>APR 21</td>
<td>3</td>
<td>4</td>
<td>$2,400</td>
</tr>
<tr>
<td>2</td>
<td>150</td>
<td>22</td>
<td>3</td>
<td>25</td>
<td>$3,750</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
<td>23</td>
<td>3</td>
<td>26</td>
<td>$5,200</td>
</tr>
<tr>
<td>7</td>
<td>175</td>
<td>27</td>
<td>3</td>
<td>30</td>
<td>$5,875</td>
</tr>
<tr>
<td>8</td>
<td>125</td>
<td>28</td>
<td>3</td>
<td>31</td>
<td>$1,600</td>
</tr>
<tr>
<td>9</td>
<td>50</td>
<td>29</td>
<td>3</td>
<td>32</td>
<td>$1,100</td>
</tr>
<tr>
<td>21</td>
<td>25</td>
<td>MAY 11</td>
<td>3</td>
<td>44</td>
<td>$2,300</td>
</tr>
<tr>
<td>23</td>
<td>50</td>
<td>13</td>
<td>3</td>
<td>46</td>
<td>$5,200</td>
</tr>
<tr>
<td>29</td>
<td>100</td>
<td>19</td>
<td>3</td>
<td>52</td>
<td>$1,325</td>
</tr>
<tr>
<td>30</td>
<td>25</td>
<td>20</td>
<td>3</td>
<td>53</td>
<td></td>
</tr>
</tbody>
</table>

TOTAL $1,000

(MAY TOTAL $200.) TOTAL DAY DOLLARS $22,075 $9,925

Divided by: Total amount paid each month 800 200
Weighted average payment days 28 50
Less: Number of days in April -30 -30
Float Days -2 -2

LIMITATION ON FLOAT FOR PAIDC & SUBCT

In view of the requirements of the Prompt Payment Act, to encourage prompt payment of subcontractors, the float on the unpaid portion of PAIDC or SUBCT categories of cost shall not exceed 28 days, notwithstanding any data the contractor may have which would indicate a historically longer float period.
CREATING A DATA FILE

Now to find the weighted average float for paid expenses (those paid for in April) simply divide the total day dollars by the April total invoices paid amount. So --

\[ \frac{22075}{800} = 27.5 \text{ days} \]

Now subtract the number of days in April 28-30 = -2 and arrive at -2 as the float for paid expenses. In other words you paid expenses on average 2 days before the cost statement was issued.

The calculation for unpaid expenses works the same way:

\[ \frac{9925}{200} = 49.6 \text{ days} \]

50-30 days in April = 20 days float.

These expenses are not paid until 20 days after the cost statement is issued.

It is important that the items used in this analysis be similar. For instance, if subcontractor progress payments are made much quicker than other payments, a separate analysis should be performed for each.

You can also see that the April total of $800 represents 80% of the total invoices received. That's where we got the percent of cost paid number.

PAIDC cost elements can be viewed in exactly the same way, so a separate example will not be given.

The rest of the lines in this part of the data file follow the formats just discussed. Now let's move on to the third section.
2. CASHII BASICS CONT'D

CREATING THE DATA FILE

TRANSACTIIONS - You can see from looking at the example that several different formats are followed in the third section of the data file. The format varies according to the type of transaction. The transactions are identified by the numbers 1, 2, 3, 4, and 5. Here is what each means:

1) Actual progress payment received. These will be entered into the input data file as appropriate to reflect receipt of payment. Contains three items, date, amount, progress payment rate.

2) Planned progress payment. These transactions show up in the transaction report. They are calculated by and used only by the CASHII program.

3) Monthly cost statement (actual or estimated). Usually make up the bulk of the transactions entered in the data file. Contains date plus one entry for each cost element, zero if no value.

4) Planned delivery. Contains date and value.

5) Actual delivery. Contains date, value, liquidation rate.

All transactions must be entered in order by date. The total number of transaction lines must be equal to the number of transactions specified in the first line of the data file. You use transaction types 1, 3, 4 and 5. The program uses transaction type 2. The total number of transactions allowed in any one run of the program is 325. Since the type 2 transactions used by the program count toward this limit your limit is somewhat less. You can see if you are within the limit by trying the calculation:

\[ 2 \times (\text{# of type 3's}) + (\text{# of 1's} + \text{# of 4's} + \text{# of 5's}) \]

If the answer you get is less than 325 you should be OK.

Now let's look at each type of transaction.
CREATING THE DATA FILE

Each line in the third section of the data file must specify which type of transaction it is. In our example, the next line is:

```
3,820130,0,0,0,100,0,0
```

The transaction illustrated is a type 3 or Monthly cost statement transaction. This portion of the file must be in order by the transaction dates. So, the next line, no matter what type of transaction it represents must have a date greater than January 30, 1982. Transactions of type 3 may be either actual or estimated expenses.

The six data items following the date correspond to the six cost elements specified in section 2 of the file. Since we specified a 6 on the first line of the file, each type 3 line must have six cost items or the program will not work. Can you tell which cost element incurred the $100 expense shown above?

Right, it's the fourth one or in our example, Material expense. We are saying here that for the transaction dated 820130 only material expenses were planned or actually incurred. If labor expense had been incurred, it would replace the first zero after the date. If interdivisional expenses were incurred, it would replace the last zero on the line and so forth. Note: that commas are used to separate the individual items of data on the line. Therefore, the system will be confused if you insert commas within the numbers. Remember, as shown in our example the number one thousand should be entered as 1000 NOT as 1,000.
2. CASHII BASICS CONT'D
-----------------------

CREATING THE DATA FILE
-----------------------

Our example continues with two more type 3 transactions. You should agree that the second of them shows a date of March 30, 1982 and specifies $150 direct labor expense and $50 overhead expense. If you don't agree, study the preceding page of explanation on the layout of this line.

The next line is a transaction type 4 or planned delivery transaction. In our example it looks like this:

```
4,820425,250
\hline
1 1 1 dollar value of planned delivery
1 1 date of planned delivery
1 1 transaction type
```

Transaction type 4's require only two items on the line after the transaction type.

So far we have discussed transaction types 2 (used by the program only) 3 (Monthly Cost Statements) and 4 (Planned Delivery). That leaves 1 and 5. Both of them represent Actuals -- a 1 is an actual progress payment, and a 5 is an actual delivery payment.

At the beginning of a contract, your file may well look like the example we have used so far. That is, agreement was reached as part of the original negotiations to request flexible progress payments. Sometimes, however, a contract is already underway before the decision is made to request flexible payments. In this situation the type 1 transaction is used to reflect any actual progress payments received.

Let's modify our sample data file to reflect some actual payments received.
2. CASH II BASICS CONT'D

CREATING THE DATA FILE

more transaction data

820101,0,6,25,.1
cash example with actual and projected data
820102,0,6,25,.0
new title just for
820103,0,6,25,.0
direct labor, incur,-2,0,0
820104,0,6,25,.0
illustration
820105,0,6,25,.0
purposes
820106,0,6,25,.0
subcont prog pay, subc,-20,9,-2
820107,0,6,25,.0
material, paide,-28,8,-2
820108,0,6,25,.0
purchased parts, paide,-20,9,-1
820109,0,6,25,.0
interdivisional, paide,-15,9,-4
820110,0,6,25,.0
3,820130,0,0,100,0,0
820111,0,6,25,.0
1,820204,0,9 =
820112,0,6,25,.0
3,820207,0,0,100,0,0
820113,0,6,25,.0
1,820304,90,9 =
820114,0,6,25,.0
3,820330,150,50,0,0,0,0
820115,0,6,25,.0
1,820404,270,9 =
820116,0,6,25,.0
4,820425,250
820117,0,6,25,.0
3,820430,150,50,100,100,100,100
820118,0,6,25,.0
1,820530,150,50,100,100,100,100
820119,0,6,25,.0
3,820630,150,50,100,100,100,100
820120,0,6,25,.0
3,820730,150,50,100,100,100,100
820121,0,6,25,.0
3,820830,150,50,100,100,100,100
820122,0,6,25,.0
4,820925,500
820123,0,6,25,.0
3,820930,150,50,-100,300,100,100
820124,0,6,25,.0
3,821030,150,50,-100,300,100,100
820125,0,6,25,.0
3,821130,150,50,-100,300,100,100
820126,0,6,25,.0
4,821225,1000
820127,0,6,25,.0
3,830130,150,50,-100,300,0,0
820128,0,6,25,.0
4,830219,1000
820129,0,6,25,.0
3,830228,150,50,0,200,0,0
820130,0,6,25,.0
3,830330,150,50,0,0,0,100
820131,0,6,25,.0
3,830430,150,50,0,0,0,0
820132,0,6,25,.0
4,830525,3950

actual progress payments received
CREATING THE DATA FILE

The changes between this example and the ones we have worked with so far are flagged. You will see that we added three transactions so the item in line one that tells the program how many transactions to expect was also changed to 25 from 22. The three new transactions are type 1 transactions. The type 1 transaction (or actual progress payment) format is:

```
1,820204,0, .9
```

- Progress payment rate (90% here)
- Progress payment actually received (see below for why this one is zero)
- Date payment received
- Transaction type

Type 1 transactions show receipt of actual progress payments.

In this case, since the payment rate is 90%, you can tell the example is for a large business concern. These actual progress payments are always at the uniform or standard progress payment rate. And they occurred before flexible payment rate was requested.

This payment was 0 simply because the payment value would have been so small it was not requested by the contractor. Since the contract illustrated was already in progress and there must be a one to one match between cost statements and actual (or planned) progress payments, this transaction was inserted.
The final transaction type is type 5, actual delivery payment received. Here is just a piece of our latest example file to show a type 5 transaction:

820101,0,6,25,.1
cash example with actual and projected data
direct labor, incur,-2,0,0
overhead, incur,15,0,0
subcont prog pay, subct,30,.9,-3
material, paidc,.8,-2
purchased parts, paidc,20,.9,-1
interdivisional, paidc,15,.9,-4
3,820130,0,0,0,100,0,0
1,820204,0,0,0,0,0
3,820238,0,0,100,0,0
1,820304,.9,0
3,820330,150,50,0,0,0,0
1,820404,270,.9
5,820425,250,.9 actual delivery payment received
3,820-30,150,50,100,100,100,100
3,8209930,100,50,100,100,100,100

Comparing to the earlier example you will see that the transaction 5 replaced a type 4 transaction. The delivery has actually taken place and payment was received. Since the type 5 replaced a 4, the number of transactions in the file remains the same and no change is required in the first line. The format for a type 5 transaction, actual delivery payment received is:

5,820425,250,.9

<table>
<thead>
<tr>
<th>Liquidation rate (90% here)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value of delivery</td>
</tr>
<tr>
<td>Date delivery made</td>
</tr>
<tr>
<td>Transaction type</td>
</tr>
</tbody>
</table>
2. CASHII BASICS CONT'D

CREATING THE DATA FILE

The liquidation rate is part of a delivery transaction. There are two liquidation rates, ordinary and alternate. For further discussion see DAR Appendix E Part 5.

We didn't discuss a type 2 transaction's format in the data file. That's because type 2 or planned progress payment transactions are used only by the program. You are not allowed to have any of them in your data file.

If you specify when you run CASHII that you want to see a transaction report, CASHII will print type 2 transactions in the appropriate spots. The transaction report is your best tool for seeing the planned cash flow and timing of the contract. It is discussed in section 3, CASHII output.

Now that we have completed our input file, let's talk about modifying it.
2. CASHII BASICS CONT'D

CHANGING/CORRECTING YOUR DATAFILE

Simple, Type it over.

Actually, you will probably want to change a specific item on one or a few lines rather than the whole file. In that case, it probably is simplest to just type the lines in question over again.

One of the strengths of computers is the ability to let the machine do the work for you. There is undoubtedly a way to instruct whichever computer you are using to change some specific data without retyping the entire line. There are some examples of doing this in the appendix of sample runs. If you don't happen to be using one of those vendors, try looking in a reference manual on System Commands or Editing Commands or try typing the line over.

Now you are ready to execute CASHII.
EXECUTING CASHII

Once again, the precise mechanism for executing CASHII depends on the computer you are using. Some specific examples are given in the appendix. However, in every case you must go through the following steps:

Access and sign on to the computer of your choice.

Enter the data required into a data file and save it.

Execute the program.

Sign off the computer.

The balance of this section is devoted to answering the questions asked by CASHII.

Once you have begun execution of CASHII, the first question will be "enter input file name". Supply the name of the file that contains your data in response to this question.

Next the program asks you to

Enter the progress payment lag time.

and

What is the delivery payment lag time?
EXECUTING CASHII

These two numbers are very significant in terms of the overall analysis. Progress payment lag tends to be the more significant. Two rules must be followed or CASHII will not process your data. Progress payment lag time may not exceed 56 days. Delivery payment lag time may not exceed 60 days.

Longer lag times tend to increase the contractor investment ratio and if significant enough, increase the progress payment rate.

The next question enables you to specify a particular progress payment rate. If you answer yes, CASHII asks you to enter the rate. Answering no allows the program to compute the rate.

The next question is: After how many days would you like the alternate liquidation rate used?

The alternate liquidation rate cannot be used until at least 365 days into the contract.

The next question is: Would you like to see a transaction report?

Answering yes provides the output discussed in the next section. Note that the report is 125 characters wide. The transaction report is useful for determining the actual cash flow of the contract. If you are interested only in the progress payment rate and the contractors investment ratio, it can be skipped.

The final question is: Would you like to see a recap of the cost report data?

Answering yes provides the output discussed in the next section. This report is 132 characters wide. The recap of cost statements can be very useful in detecting a typing mistake on the input data file. It simply formats nicely and prints all the type 3 transaction data from the file, with totals. You may routinely answer no unless you expect a problem in the data.
EXECUTING CASHII

Now CASHII produces the output you requested and the Analytical Report. On completion of all the output, one final question is asked:

Would you like to run the model again using the same datafile but changing the terminal input?

Answering yes starts the run over again at the question "Enter progress payment lag time". Answering no stops the program.

If you have another data file that you wish to process, you should answer no, let the program stop and then execute it again.

See the appendix for complete annotated examples of executing CASHII.

Now we are ready to discuss the output in detail. That's the subject of the next section.
CASHII offers to produce three reports each time it is executed. One of them, the Analytical Report, is always produced. An example of it was shown in section 1, A Sample CASHII Session. It answers the question; What will the flexible progress payment percentage be? The other two reports are both optional and are produced only if you request them. They are the Transaction Report and the Recap Cost Statement Report. Both of them round to the nearest whole number.

Let's address the reports in the same order in which they are produced.

TRANSACTION REPORT

If you answered yes for the transaction report, that will be the first one generated. Figure 4 shows what it looks like:
Working across the report from left to right you see that each transaction has generated 10 columns of information. This includes the type 2 transactions, Planned Progress payments which are inserted by CASHII. Let's look at those columns which may not be self-explanatory:

Column 3 is calculated as column 1 minus column 2.

Column 5 involves several calculations so it is best explained using our example. Note, that the delivery amount of the first planned delivery is $250, the ordinary liquidation rate is .98 and the profit rate is .10. The calculations are:

\[
\begin{align*}
250 \times .98 &= 245 = \text{liquidation amount} \\
250/(1 + .10) &= 227.3 = \text{cost in this delivery} \\
227.3 \times .98 &= 222.7 = \text{cost previously reimbursed as progress payments} \\
227.3 - 222.7 &= 4.6 = \text{cost reimbursed at delivery (rounds to 5)} \text{ displayed in column 5.}
\end{align*}
\]

Column 7 also involves some calculating using some of the same numbers as column 5:

\[
\begin{align*}
250 - 245 &= 5 = \text{net payment at delivery} \\
5 - 4.6 &= .4 = \text{profit paid at delivery (rounds to 0)} \text{ displayed in column 7.}
\end{align*}
\]

Column 10 is calculated by subtracting the liquidation amount (calculated in column 5) from cumulative progress payments.
3. CASHII OUTPUT CONT'D

-------------------------

ANALYTICAL REPORT

-------------------------

The analytical report is always produced. It contains the key items of information that CASHII is all about. Namely, the progress payment rate and the contractor investment ratio. Recall from our earlier discussion that the progress payment rate may range up to 100% while the contractor investment ratio is held at approximately 5%. Details on the method of calculating the contractor investment percentage are found in section 4, short guide to syntax.

Figure 5 is a repeat of the sample Analytical Report shown earlier:


### Project Title: CASA Example with Projected Data

<table>
<thead>
<tr>
<th>COST ELEMENT DATA</th>
<th>UNPAID COST</th>
<th>PAID COST</th>
<th>ELIGIBILITY</th>
<th>DAYS FLOAT</th>
<th>DAYS FLOAT</th>
<th>PROGRESS PAYMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 DIRECT LABOR</td>
<td></td>
<td>-2</td>
<td>INCURRED COST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 OVERHEAD</td>
<td></td>
<td></td>
<td>INCURRED COST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 SUBCONTRACT PAY</td>
<td>28</td>
<td>88.0</td>
<td>-2</td>
<td>SUBCONTRACT PROC PAY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 MATERIAL</td>
<td>28</td>
<td>88.0</td>
<td>-2</td>
<td>PAID COST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 PURCHASED PARTS</td>
<td>28</td>
<td>98.0</td>
<td>-1</td>
<td>PAID COST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 INTERDIVISIONAL</td>
<td>15</td>
<td>28.0</td>
<td>-4</td>
<td>PAID COST</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Progress Payment Rate Data**

- **Progress Payment Rate**: 98.00
- **Ordinary Liquidation Rate**: 98.00
- **Alternate Liquidation Rate**: 89.00

- **Contractor Investment Ratio**: 5.42

**Cost Element Data**

<table>
<thead>
<tr>
<th># NAME</th>
<th>UNPAID COST</th>
<th>PAID COST</th>
<th>ELIGIBILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 DIRECT LABOR</td>
<td></td>
<td>-2</td>
<td>INCURRED COST</td>
</tr>
<tr>
<td>2 OVERHEAD</td>
<td></td>
<td></td>
<td>INCURRED COST</td>
</tr>
<tr>
<td>3 SUBCONTRACT PAY</td>
<td>28</td>
<td>88.0</td>
<td>-2</td>
</tr>
<tr>
<td>4 MATERIAL</td>
<td>28</td>
<td>88.0</td>
<td>-2</td>
</tr>
<tr>
<td>5 PURCHASED PARTS</td>
<td>28</td>
<td>98.0</td>
<td>-1</td>
</tr>
<tr>
<td>6 INTERDIVISIONAL</td>
<td>15</td>
<td>28.0</td>
<td>-4</td>
</tr>
</tbody>
</table>

**Progress Payment Lag Time in Days**: 6

**Delivery Payment Lag Time in Days**: 20

**Date of Final Delivery**: 830525

**Progress Payment Rate Computed by Program**: 98.00

Figure 5
The first block of information consists of a summary of data from your data file and results of CASHII's calculation of profit percentage. The calculated profit percentage and the percent specified in the first line of your data file must be the same or the program will stop.

Contract price is the sum of all the deliverables in the data file. Total contractor cost is the sum of all the cost elements in the data file. The difference between these two numbers is the profit. Start up cost and start date are simply read in from the data file.

Next is a block showing the result of the calculations which have taken place. In this example the progress payment rate was calculated to be 98%. The ordinary liquidation rate is always the same as the progress payment rate. Liquidation rate is the portion of any deliverable amount which is assumed to have already been reimbursed. The alternate liquidation rate is calculated by CASHII and can be used, at your option, after 365 days of the contract. Finally, the contractor investment ratio is calculated. This number should not be less than 5%.

The next block of data on the Analytical Report is a formatted printout of your cost element data from the input data file. You should verify that these numbers are what you expected. If not, it will be necessary to change the data file.

Finally, the Analytical Report shows you how you answered the two lag time questions asked at the terminal and the date of final delivery as specified in the data file.
3. CASHII OUTPUT CONT'D

RECAP OF COST STATEMENT REPORT

The Recap Report is produced last, if you requested it. Here's a sample of what it looks like:

<table>
<thead>
<tr>
<th>DATE</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
<th>05</th>
<th>06</th>
<th>07</th>
<th>08</th>
<th>09</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0202</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0303</td>
<td>150</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0404</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0505</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0606</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0707</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0808</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0909</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1010</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1111</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1212</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1313</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1414</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1515</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1616</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1717</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1818</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1919</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2020</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2121</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2222</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2323</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2424</td>
<td>150</td>
<td>50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

TOTAL: 2100 700 0 2400 900 900 0 0 0 0 7000

Careful study will show that this report pulls all the transaction type 3's from your data file, formats them and prints them for your inspection.

As mentioned previously, this is a good way to check the validity of your input data. If CASHII is giving unexpected results, check here to see that the computer actually has the data you think it does.

One other point of interest is the negative numbers in cost element #3, Sub-contractor Progress Payments. These simply indicate liquidation of earlier entries. The total for subcontractor cost elements must add to zero over the life of the contract.
4. CASHII TERMS, DEFINITIONS AND SYNTAX

CASHII is a very simple program to run. The questions it asks and the error messages it provides are as clear as possible and provide as much information as possible.

Syntax refers to exactly how you must provide information so that it will be understood by the program. For instance, items in the datafile must be separated by commas and percentages must be expressed as a decimal. In fact, those are the two most notable syntax requirements. Most of the other rules are imposed by the program itself (such as maximum delivery payment lag time of 60 days) and so will be discussed with the appropriate definitions of error messages.

Syntax for the data file:

Initial Line:

820:01,0,6,25,.1

A A A A A

| Profit rate as decimal
| Number of transactions in file
| Number of cost elements (max of 10)
| Start up cost (max 6 digits)
| Date contract begins

Title line (must be second)

CASH RUN PROJECT J-327

A

| Run title (max 50 characters)
4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT’D

Cost element lines

Direct labor, INCUR, 2, 0, 0
Material, PAIDC, 28, 8, 4

Float for paid costs (must be 0 for INCUR)
Cost paid during accounting period (0 for INCUR)
Float for unpaid items
Type of cost element (must be INCUR, PAIDC or SUBCT)
Cost of element title (max of 20 characters)

You must have as many cost element lines as you specified in the initial line of the file.

Transaction lines

Type 1 (Actual progress payment received)

1,820404,270, 9

Progress payment rate (must be decimal number)
Actual payment received (max of 6 digits)
Date payment received.

Type 2 (Planned Progress Payment)

Computed by the CASHII program and is not to be in the data file.

Type 3 (Monthly cost statement, Actual or estimated)

3,820830, 150, 50, 100, 100, 100, 100
3,820930, 200, 0, -100, 0, 100, 0

Individual cost, one entry per cost element (max of 6 digits each)

Date
Transaction Lines Cont'd

Type 4 (Planned delivery)

4,830201,1000
     A      A
       Value of delivery (max of 6 digits)
       Planned date of delivery

Type 5 (Actual delivery)

5,820425,250,.9
     A      A
       Liquidation rate (must be a decimal number)
       Value of actual delivery (max 6 digits)
       Date of actual delivery

Most terms were defined as part of section 2, creating the data file. The definitions here supplement the ones given earlier. These definitions are in alphabetical order.

ACCOUNTING PERIOD - The period over which costs are accumulated to be billed in a cost statement. The program is designed to compute progress payments no more frequently than monthly; however, a monthly cost accounting period may be a calendar month, 30 days, 4 weeks, or 5 weeks. Cost accounting periods must be at least 28 days in length and there may be no more than twelve per year. Used in type 3 transactions in the input data file.
4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

ALTERNATE LIQUIDATION RATE - That portion of an actual delivery value which is assumed to have already been reimbursed. The alternate rate may only be used after 365 days of the contract has elapsed. See Ordinary Liquidation Rate, see DAR Appendix E, Part 5.

CASHII PROGRAM - The CASHII computer program calculates the highest whole number progress payment rate that is possible while keeping the contractor weighted cumulative work-in-process inventory from going below a level of 5.0%. The program is written in ANSI FORTRAN x3.9-1978.

CONTRACT START AND END DATES - The contract start date is the date, in year, month, day (YYMMDD) format, of the first day of the accounting period of the first cost statement. If actuals are input, this will be the day of incurred cost for the proposal. If there is no start-up cost, the contract start date will be the first day of the month of definitization. The contract end date is the date of the final delivery. Used in the input data file.

CONTRACTOR INVESTMENT RATIO (CIR) - Is calculated with weighted day dollars. Cumulative day dollars of costs reimbursed (CR) are subtracted from cumulative paid costs to get unreimbursed cost. CIR is then unreimbursed costs divided by cumulative work in process inventory. Work-in-process (WIP) is calculated as weighted cumulative paid cost (CP) less the weighted cumulative cost of deliveries made (COD). COD is weighted on the day of delivery payment and then removed from the weighted CP to arrive at a weighted WIP amount. Thus:

\[ WIP = CP - COD \]
\[ CIR = (CP - CR) / WIP \]

CASHII seeks to hold CIR to not less than 5%. CASHII performs successive calculations to determine the highest progress rate (up to 100%) possible while keeping the weighted Contractor Investment Ratio equal to or greater than 5%. The result of the calculation is printed in the Analytical Report.
4. CASHI TERMS, DEFINITIONS AND SYNTAX CONT’D

COST AS INCURRED - Cost as incurred is a type of cost which is eligible for progress payments when it is incurred. Incurred costs are those costs identified through the use of the accrual method of accounting and reporting. Facilities capital cost of money, which is recognized as an allowable cost, is also an incurred cost for progress payment purposes. Used in the input data file for appropriate cost elements. Use the code INCUR.

COST AS PAID - Cost as paid is a type of cost which is not eligible for progress payments until it is paid. Used in the input data file for appropriate cost elements. Use the code PAIDC.

COST ELEMENTS - Cost elements are those groupings of cost which are of the same progress payment eligibility and of similar float times. The model will accept up to 10 cost elements. Contractors are encouraged to develop computer programs to reformat DD Form 633 data to a format consistent with the DOD Cash Flow computer model input file requirements. All cost elements must be "typed" as one of INCUR, PAIDC or SUBCT. Must be specified in the input data file.

COST ELIGIBLE FOR CUSTOMARY PROGRESS PAYMENT - Includes only those recorded or estimated costs which result, at the time of the cost statement request, from payment made by cash, check, or other form of actual payment for items or services purchased directly for the contract, together with cost incurred, but not necessarily paid, for materials which have been issued from the Contractor's stores inventory and placed in the production process for use on the contract, for direct labor, for direct travel, for other direct in-house costs, and for properly allocable and allowable indirect costs. The Cash Flow Model converts incurred cost to paid cost by adding the cost element payment float to the incurred cost date (i.e., month-end date plus float).
4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT’D

COST ELIGIBLE FOR 100% PROGRESS PAYMENT - The amount of progress payments which have been paid to contractors' subcontractors and other divisions are eligible for 100% reimbursement. Used in the input data file for appropriate cost elements. Use the code SUBCT.

COST STATEMENT OF MONTHLY COST - The actual or projected monthly cost (by up to 10 cost elements) estimated to be incurred or paid during the cost accounting period. Specified in the input data file as transaction type 3. Must be entered in order by date.

COST STATEMENT DATE - The last day of the cost accounting period. Entered in the input data file as year, month, day, i.e., 820228.

CUSTOMARY PROGRESS PAYMENT RATE - The uniform standard percent (90% or 95%) or the flexible percent (if requested by the contractor) of the amount of the contractor's total costs. Calculated by CASHII or may be entered in response to the proper question when CASHII is executing.

DATA FILE - A collection of information stored in the computer. For the CASHII system the contract information discussed in section 2, necessary to perform the flexible progress payment calculations.

DELIVERY PAYMENT LAG - is the number of days between the delivery date and the receipt of the actual delivery payment. Specified in response to the question asked when CASHII is executed. May not exceed 60 days.

ERROR MESSAGE - A message supplied by CASHII indicating some problem exists which prevents normal program functions. For instance, entering a lag time which is beyond the maximum allowed by the program. See the listing of error messages which concludes this section.
FLEXIBLE PROGRESS PAYMENT RATE - The flexible progress payment rate is the progress payment rate calculated by the CASHII program. Once the contract is definitized this rate is maintained throughout the life of the contract. However, if actual and projected cash flow data generated during performance of the contract reveal that the progress payment rate will result in a weighted average investment in work-in-process inventory by the contractor in excess of 7% or less than 3% the progress payment rate shall be re-determined by using the DOD Cash Flow Computer model. In no event will the progress payment rate be less than the uniform standard progress payment rate or greater than 100%.

FLOAT - Float is the number of days between the cost statement date and the time the cost is paid on a cash basis. Bank lag times are included in the float number entered in the data file. The program will accept both positive and negative whole numbers for float entries. Specified for each cost element. See the examples of calculations in section 2.

MONTH OF DEFINITIZATION - The month of contract definitization is the month in which a contract is signed or definitized. This will be the first cost statement period.

ORDINARY LIQUIDATION RATE - That portion of an actual delivery value which is assumed to have been already reimbursed. Always equal to the progress payment rate. Used in the input data file, transaction type 5. See also Alternate Liquidation Rate, see DAR, Appendix E, Part 5.

PERCENT OF MATERIAL COST PAID - The percent of material cost that is paid as of the cost statement date. An estimate of the percentage of dollars paid on invoices per accounting period should be made and included in cost element description lines in the data file.
PROGRESS PAYMENT ELIGIBILITY CODES - There are three types of costs with different progress payment eligibility codes. These are: cost as incurred (INCUR), cost as paid (PAIDC) and cost eligible for 100% progress payment (SUBCT).

PROGRESS PAYMENT LAG - Progress payment lag is the number of days between the end of the cost statement date and the progress payment receipt. This lag should not normally include more than 5 working days from the cost statement date for the contractor to prepare and submit a progress payment request. Specified when executing CASHII. May not exceed 56 days.

START UP COST - Start-up cost is the cost authorized and incurred by the contractor prior to definitization, i.e., approval by the government of the signed contract. If progress payments have been received during this time start-up cost is calculated by totaling the cost to date of definitization and subtracting any progress payments received plus applicable profit. Start-up cost is input in the first line of the data file.

SUBCONTRACTOR PROGRESS PAYMENTS - Subcontractor progress payments are eligible for 100% reimbursement to the contractor. This is the only cost element that accepts negative whole numbers in a cost statement. These negative numbers represent subcontractor liquidations to accumulated progress payments when the subcontractor starts deliveries. The sum of subcontractor progress payments must equal 0 at the time of the last transaction.
TRANSACTION TYPE - Each type of transaction used in the third section of the input data file must be identified uniquely. There are five types:

1) Actual progress payment received. These will be entered into the input data file as appropriate to reflect receipt of payment. Contains three items, date, amount, progress payment rate.

2) Planned progress payment. These transactions show up in the transaction report. They are calculated by and used only by the CASHII program.

3) Monthly cost statement (actual or estimated). Usually make up the bulk of the transactions entered in the input datafile. Contains date plus one entry for each cost element, zero if no value.

4) Planned delivery contains date and value.

5) Actual delivery. Contains date, value, liquidation rate.

All transactions must be entered in order by date. The total number of transaction lines must be equal to the number of transactions specified in the first line of the data file. The total number of transactions (including type 2's) may not exceed 325. If yours do, try combining earlier historical ones.
UNLIQUIDATED PROGRESS PAYMENT - Column 10 on the Transaction report. It is that portion of cumulative progress payments which have not been liquidated.

WORK-IN-PROGRESS - A calculation performed by the program in order to calculate contractor investment ratio. WIP is calculated as cumulative day dollar cost of deliveries made subtracted from cumulative day dollar paid cost.
As you might expect with computers, any number of things can keep it from working exactly right. CASHII tries hard to give you as much information as possible about the error. If the error is on a specific line, that line will be printed at the terminal. For instance, the following line has an error. Can you see it?

```
3,820330,150,50,100100,100,100
```

Right, the problem is the missing comma between the first two 100s. In this case the error message is:

```
Error in the following input line - then the line prints
```

then a second message prints: The program was stopped because of an input file error.

and the program stops. Now you have to fix the problem and execute CASHII again. Sometimes when you get the error message above, there is nothing obviously wrong. The first thing to try is typing the entire line over again and saving the new version. Most times when an error occurs, CASHII continues checking the input data file for other errors.
ERROR MESSAGES Cont'd

The following is the list of error messages with some comments on corrective action where it isn't obvious.

Actual progress payment must have a value greater than 0. The progress payment dated 'date' is less than 0.

'date' will be the date of the transaction.
Fix the line and try again.

Cost and delivery transactions must not have value larger than six digits. The cost statement dated 'date' has a value that is too large.

'date' is the date of the type 3, 4 or 5 transaction in error. Could be a missing comma. If not divide all your input numbers by 10, 100, 1000, etc. to get smaller numbers.

Cost reimbursement cannot be greater than incurred cost. The transaction dated 'date' violates this rule. The delivery payments to date exceed the cost to date.

Total cost at this point from cost statements XXXXX
Total deliveries at this point XXXXX
Cost reimbursed at this point based on delivery prices XXXXX

'date' identifies the transaction where this error was noticed. Possibly a missing type 3 transaction. Could also be a type 4 or 5 transaction with too large a value.
4. CASHII, TERMS, DEFINITIONS AND SYNTAX CONT'D
-----------------------------------------

ERROR MESSAGES Cont'd
---------------------

Cost element 'X' does not have a valid cost element code.

'X' will be a number. The program starts counting at the first cost element. The number 6 would mean the sixth one down doesn't specify INCUR, SUBCT or PAIDC.

Data is not in chronological order. Check transactions dated 'date' and 'date' to correct.

'date' indicates transactions where the problem was noticed. Possibly a typo or you just don't have all the transactions in order by date.

Data file contains more actual progress payments than cost statements. The number of actual progress payments should equal the number of cost statements before the last actual progress payment.

Could it be you accidently wiped out a cost statement? Count them. Or maybe you didn't mean to enter an actual progress payment.

Data file indicated 'X' transactions were included in the data file. 'Y' were actually included.

'X' and 'Y' will be different numbers. They need to be the same. Your file said (in the first line) that 13 transactions were included, the program only found 12. Possibly a typo or maybe you forgot one. Fix and re-run.

NOTE: If 'X' is smaller than 'Y', the program simply ignores the excess data.
4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT’D

ERROR MESSAGES Cont’d

Delivery payment must have a value larger than 0 but not contain more than 6 digits. The transaction dated ‘date’ violates this rule.

fix it. If the value is too big divide by 10,100, etc. Remember, the rest of the numbers in the file must be consistent.

Do not input any planned progress payments into the data file. The transaction dated ‘date’ must be removed.

‘date’ indicates the transaction in error. You have a type 2 transaction in the data file. You can’t do that, only the program uses type 2 transactions.

First transaction must be a cost report.

you have to have a type 3 transaction as the first transaction in your data file.

Float days on costs paid in the present accounting period should be negative. This is not the case for cost element ‘X’.

‘X’ identifies the cost element in question.

Liquidation rate associated with an actual delivery must be between 0 and 1. The transaction dated ‘date’ violates this rule.

‘date’ indicates the transaction in error. Liquidation rate must be a decimal number. A 90% rate is specified as .9.
4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT’D

ERROR MESSAGES Cont’d

----------------------------------

Last transaction must be a delivery.

yours isn’t, it should be. If it’s there check the total number of transactions in the file and make sure you have the proper number in the first line of the file.

Negative cost is permitted only for subcontractor type progress payments. The cost statements dated ’date’ has a negative value and is not a subcontractor cost element.

probably a typo.

Number of cost elements must be between 1 and 10.

combine similar costs into one category.

Percent paid in the present accounting period should be between 0 and 1 for cost element ‘X’.

’X’ identifies the cost element in question, counting down from the first cost element. This value must be a decimal number. 80% would be .8.

Planned progress payments cannot occur less than 28 days apart. The transaction dated ’date’ violates this rule.

check input file to insure cost statements are at least 28 days apart.

Progress payment rate must be between 0 and 1. The transaction dated ’date’ violates this rule.

date identifies the transaction in error. You must specify the rate as a decimal. 90% would be .9.
The profit rate computed from the data file (XX.XX) does not match the profit rate (XX..'X) in the first line of the data file.

Computed profit based on the following values computed from data inputs:

- Total price from delivery prices: XXXXXX
- Total cost from cost statement: XXXXXX
- Profit dollars: XXXXX

Profit rate must be between 0 and 1.

- a mistake on the first line of your data file. The last number is the profit rate and it has to be a decimal number. 10% profit is specified as .1.

Start-up cost cannot be a negative number.

- fix it.

Subcontractor progress payments should add to 0 across the life of the contract for cost element 'X'.

- subcontractor costs must be fully liquidated. 'X' identifies which cost element has the problem. Fully liquidated means add to zero. You don't have enough negative numbers to equal the total value of subcontractor expenses.

Transaction dated 'date' has an illegal transaction code. The code must be a 1, 2, 3, 4, or 5.

- actually, it can't be a 2 either.

The transaction date 'date' has an illegal month 'XX'.

- The month is less than zero or greater than 14.

The transaction date 'date' has an illegal day 'XX'.

- The day is greater than the number of days in a particular month.

The float days for unpaid costs cannot exceed 28 days. This is not the case for cost element 'X'.

- See note at bottom of page 19.
SAMPLE USING GEISCO

IDENTIFY TERMINAL TYPE AND SPEED
ENTER YOUR USER NUMBER
ENTER YOUR PASSWORD

NEW IS THE COMMAND NECESSARY TO ENTER
NEW DATA. SAMPI IS THE FILE NAME.
F77 MEAN: FORTRAN

ACCIDENTALLY TYPED 2 R'S
→ IS THE CORRECTION KEY.

1,8204 IS A MISTAKE. HOLD DOWN CONTROL
KEY AND TYPE AN X. THE COMPUTER PRINTS
DELETED. START THE LINE OVER AGAIN.

FINISH ENTERING YOUR DATA
WHEN FINISHED ENTERING YOUR DATA,
REMOVE LINE NUMBERS FROM THE FILE
SAVE (REPLACE) THE DATA
EXECUTE THE PROGRAM

PAPER MOVES TO TOP OF PAGE (IF YOUR
TERMINAL IS EQUIPPED PROPERLY)

NOTE: COMMAND SYSTEM REFERENCE MANUAL
PUBLICATION #3501.01 FURTHER
EXPLAINS COMMANDS.

CAUTION--THE CASH MODEL VERSION IS NOT COMPATIBLE
WITH THE USER GUIDE DATED AUGUST 28, 1981. PLEASE CONSULT THE
USER GUIDE DATED FEBRUARY 1, 1983 FOR GUIDANCE

INPUT FILE NAME? SAMPI ← ENTER THE NAME OF YOUR DATA FILE

APPENDIX 1
SAMPLE USING BCS

ENTER YOUR USER NUMBER
ENTER YOUR PASSWORD

ENTER YOUR PROJECT ID

EDIT IS A COMMAND, NECESSARY TO ENTER DATA
SAMP1 IS THE FILE NAME
DATA IS THE FILE TYPE
I MEANS WE WANT TO INSERT DATA

ENTERING THE DATA

HIT THE M INSTEAD OF @ IS THE CORRECTION KEY 1,820404,2 WAS A MISTAKE, HOLD DOWN THE CONTROL KEY, TYPE AN X AND CONTINUE WITH THE CORRECT DATA.

ENTERING A CARRIAGE RETURN WHEN NO MORE DATA
SAVES THE FILE
EXECUTE THE PROGRAM
ENTER THE NAME OF YOUR DATA FILE

NOTE: MAINSTREAM - CTS COMMAND LANGUAGE REFERENCE MANUAL FURTHERExplains THE COMMANDS.

APPENDIX 2
REFERENCE MATERIAL FOR THIS DOCUMENT IS CONTAINED IN DEFENSE ACQUISITION REGULATION, APPENDIX E, PART 5.
**SECTION ABOVE Establishes Attributes of Variables Passed Between Main Program and Subroutines.**

**SECTION IMMEDIATELY BELOW Writes Model Headings.**

**CALLS subroutine TO READ INPUT FILE.**

**CALLS subroutine TO EDIT INPUT DATA.**

**FILE READ ERROR INTERRUPTED. THE PROGRAM WAS STOPPED BECAUSE OF AN INPUT FILE ERROR.**
**Routine Allows User to Check Input Data When Edit Routine Detects Errors.**

```
WRITE(6,210)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

FORMAT(' WOULD YOU LIKE TO CHECK YOUR INPUT DATA? (YES OR NO) ')
WRITE(6,210)
READ(5,FMT=') DATCHK.
IF (DATCHK(1:1).NE.'Y'.AND.DATCHK(1:1).NE.'N') THEN
  GO TO 205
END IF
IF(DATCHK(1:1).EQ.'Y') THEN
  IF(DATCHK(1:1).EQ.'YES') THEN
    FORMAT(' DO YOU WANT TO CHECK: 1-COST INPUT DATA; 2-DELIVERY INPUT?')
    WRITE(6,70)
  WRITE(6,220)
  READ(5,FMT=') DATCHK.
  IF(DATCHK.LT.1.OR.DATCHK.GT.2) THEN
    WRITE(6,70)
    GO TO 225
  END IF
END IF
IF(DATCHK.GT.0) THEN
  GO TO 305
ELSE
  GO TO 2000
END IF
GO TO 2000
END IF

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CALLS SUBROUTINE TO GET USER INPUTS FROM TERMINAL.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CALL DATAIN(FRGLAG,DELLAG,FLXFLG,FRGRTE,FRGFLG,
&ALTLDY,TRNFLG,CSFILG,DRRUN,TWDFLG)
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CALLS SUBROUTINE TO CREATE PLANNED PROGRESS PAYMENTS AND
ADD DELIVERY LAG TO DELIVERY DATES.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CALL CREPAY(TRNSAC,NMTRAN,ACPFRG,CCOST,FRGLAG,FRGFLG,STDAY,DELLAG,
&ERROR)
  IF (ERROR.EQ.'YES') THEN
    GO TO 205
  END IF

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INITIALIZES (OR RESETS) VARIABLES TO ZERO.
INITIALIZES FIRST TIME THROUGH; RESETS AS FLEX RATE STEPS
DOWN AND WHEN ALTERNATIVES ARE RUN WITH THE SAME DATA
FILE.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
320 DO 250 I=1,NMTRAN
  DO 240 J=1,10
    OUTPUT(I,J)=0
  CONTINUE
  CONTINUE
  DO 270 I=1,1000
    DO 260 J=1,2
      CSILT(I,J)=0
      UFILT(I,J)=0
      UFCSUB(I,J)=0
  CONTINUE
```
CONTINUE
DO 290 I = 1, 100
   DO 290 J = 14, 17
      TRNSAC(I, J) = 0
CONTINUE
DO 295 I = 1, 170
   CSTDAY(I) = 0
CONTINUE
   LFLG = 0
   FPCNT = 0
   CNTST = 0
   ALTFRF = 0
   FDCAST = 0
   INFLOW = 0
   COUNT = 0
   FLOATD = 0
   ALOFLG = 0
   SUBCNT = 0
   UPDCNT = 0
   SUMCST = 0
   TOTCHG = 0

******************************************************************************
calls subroutines to handle each transaction type.
******************************************************************************
DO 300 I = 1, NMTRAN
   IF (TRNSAC(I, 1), .EQ. 1) THEN
      CALL ACTPFY(TRNSAC, I, OUTPUt, FLOATD, CSTFL, FRFRAT, ALTLDY, STCOST)
   END IF
   IF (TRNSAC(I, 1), .EQ. 2) THEN
      CALL FRGFAY(TRNSAC, I, FRGRT, CSTFL, FLOATD, OUTPUt, TOTCHG, SUMCST, TOTP, ALTLDY, FRFRAT, STCOST, FPCNT, CSTDAY, TOTCHG)
   END IF
   IF (TRNSAC(I, 1), .EQ. 3) THEN
      CALL COSTST(TRNSAC, I, NMCSL, ELMNT, FLOATD, CSTFL, OUTPUt, SUMCST, UPDCNT, UFDL, UFDSUB, FRGRT, STCOST, CNTST, CSTDAY, TOTCHG)
   END IF
   IF (TRNSAC(I, 1), .EQ. 4) THEN
      CALL FLDKELD(TRNSAC, I, ALTLDY, FRFRAT, ALOFLG, CSTFL, FLOATD, TOTCHG, SUMCST, OUTPUt, FRGRT, LSTDEL, DELLAG, STCOST, TODAYS, STDAY, ALTFRF, LFLG)
   END IF
   IF (TRNSAC(I, 1), .EQ. 5) THEN
      CALL ACTDEL(TRNSAC, I, FRFRAT, ALTLDY, ALOFLG, CSTFL, FLOATD, TOTCHG, SUMCST, OUTPUt, LSTDEL, DELLAG, STCOST, TODAYS, STDAY, ALTFRF)
   END IF
CONTINUE
******************************************************************************
calls subroutine to compute contractor investment ratio.
******************************************************************************
CALL INVEST(TRNSAC, NMTRAN, TODAYS, CSTFL, FLOATD, RATIO, OUTPUt, DELLAG)
& STCOST, FRFRAT)
IF (FLDLG(1:1), .EQ. 'N') THEN
   IF (RATIO, LT, (.05, AND, FRGRT, GT, (.90))) THEN
      FRGRT = FRGRT - 0.01
      GO TO 290
   END IF
END
CALLS SUBROUTINE TO PRINT RESULTS.

CALL FINTOUT(INFLG,CI,FLGLG,TRAN,ELMENT,FLXFLG,NMTRAN, 
NMCSTL,STDATE,COSTL,FFKAT,STCOST,TOIFD,TUCOS,EXITOUT,TOFROF, 
&FLKLS,FRKTE,RATIO,FLGLG,DELLAG,LSTSTL,TITLE,DATE,Y,FLGLG)

ALLOWS USER TO CHANGE TERMINAL INPUTS AND RE-RUN USING 
THE SAME DATA FILE.

Would you like to run the model again using the same 
DATA file but changing the variables at the terminal? 

1. Format: 'YES / NO')
2. Format: Input? 'YES / NO')

WRITE(6,FMT='(A1)') CC
WRITE(6,FMT='(A11)')
WRITE(6,FMT='(L5)')
READ(5,FMT='(A3)') FLGIT
IF (FLGIT(1:1) .NE. 'Y'.AND.FLGIT(1:1) .NE. 'N') THEN
   GO TO 320
END IF
IF (FLGIT.EQ. 'YES') THEN
   REFUN=1
   GO TO 100
END IF
END CONTINUE
STOP

********** END OF MAIN PROGRAM **********

********** START OF SUBROUTINE FILEIN **********

SUBROUTINE FILEIN(STDATE,STCOST,NMCSTL,NMTRAN,FFRAT,TITLE,COSTL,CAS155)
REAL FFRAT,ELMENT(10,5),TRANAC(100,20),STCOST
INTEGER A,TRNACNT,STDATE,NMCSTL,NMTRAN,ACFG,STDAY,IPR,IER
CHARACTER*20 COSTL(10,2)
CHARACTER*50 TITLE
CHARACTER*50 HOLD,SAVE
CHARACTER*5 ERROR
DIMENSION F(1,5)
DATA IF(1/4,3,8,7,4)/
ERROR='NO'
OPEN(UNIT=1,STATUS='OLD',ACCESS='SEQUENTIAL')

READ FIRST LINE OF INPUT FILE.

READ(UNIT=1,FMT=*) STDATE,STCOST,NMCSTL,NMTRAN,FFRAT
READ(UNIT=1,FMT=*) TITLE
READ(UNIT=1,FMT=(A50))HOLD
SAVE=HOLD
CALL GETSTR(HOLD,*,SHOLD,IPR)
IF (IPR.EQ.6 OR IPR.GT.9) GOTO 135
STDATE=INTSTD(HOLD,IER)
IF (IER.EQ.1) GOTO 135
CALL GETSTR(HOLD, SHOLD, IFOS)
IF(IFOS.LT.2, OR. IFOS.GT.9) GOTO 170
COST = REALSV(SHOLD, IERR)
IF(IERR.EQ.1) GOTO 175
CALL GETSTR(HOLD, SHOLD, IFOS)
IF(IFOS.LT.2, OR. IFOS.GT.9) GOTO 175
NMCSTL = INTSV(SHOLD, IERR)
IF(IERR.EQ.1) GOTO 175
CALL GETSTR(HOLD, SHOLD, IFOS)
IF(IFOS.LT.2, OR. IFOS.GT.9) GOTO 175
NFMT = REALSV(SHOLD, IERR)
IF(IERR.EQ.1) GOTO 175
GOTO 140
WRITE(*, 390) SAVE
ERROR = YES
GOTO 600
CONTINUE
*************** READS TITLE FROM INPUT FILE. ***************
READ(UNIT=1, FMT=(A80)) HOLD
TITLE = HOLD
DO 120 I=1, NMCSTL
  DO 110 J = 1, 5
    ELEMENT(I, J) = 0
  CONTINUE
110 CONTINUE
*************** READS COST ELEMENT DATA FROM INPUT FILE. ***************
READ(UNIT=1, FMT=*) COSTEL(I, 1), COSTEL(I, 2), ELEMENT(I, 2), ELEMENT(I, 4)
DO 190 I=1, NMCSTL
  READ(UNIT=1, FMT=*) COSTEL(I, 1), COSTEL(I, 2), ELEMENT(I, 2), ELEMENT(I, 4)
ENDDO
SAVE = HOLD
CALL GETSTR(HOLD, SHOLD, IFOS)
IF(IFOS.LT.2) GOTO 185
CALL GETSTR(HOLD, SHOLD, IFOS)
IF(IFOS.LT.2, OR. IFOS.GT.9) GOTO 185
COSTEL(I, 2) = SHOLD
CALL GETSTR(HOLD, SHOLD, IFOS)
IF(IFOS.LT.2, OR. IFOS.GT.9) GOTO 185
ELEMENT(I, 2) = REALSV(SHOLD, IERR)
IF(IERR.EQ.1) GOTO 185
CALL GETSTR(HOLD, SHOLD, IFOS)
IF(IFOS.LT.2, OR. IFOS.GT.9) GOTO 185
ELEMENT(I, 2) = REALSV(SHOLD, IERR)
IF(IERR.EQ.1) GOTO 185
CALL GETSTR(HOLD, SHOLD, IFOS)
IF(IFOS.GT.9) GOTO 185
ELEMENT(I, 4) = REALSV(SHOLD, IERR)
IF(IERR.EQ.1) GOTO 185
IF (COSTEL(I, 2).EQ. "INCUR") THEN
  ELEMENT(I, 1) = 1
END IF
IF (COSTEL(I, 2).EQ. "PAID") THEN
ELEMENT (1,1) = 2
END IF

IF (COSTEL (1,2) .EQ. 'SUBC') THEN
  ELEMENT (1,1) = 3
END IF

GO TO 190

185 WRITE (6, 390) SAVE
ERROR = 'YES'

190 CONTINUE

***** NEXT LINE ESTABLISHES RELATIVE REFERENCE POINT FOR CONTRACT

START DATE. ALL TRANSACTIONS WILL USE THIS POINT

TO DETERMINE DAYS AFTER CONTRACT AWARD.

SDAY = FDAY (STDATE)
FF1 (I) = NMCTSL + 2
TRANCNT = NMTRAN + 2
DO 270 I = 1, TRANCNT
  DO 260 J = 1, 20
    TRNSAC (I, J) = 0
  CONTINUE
260 CONTINUE
270 CONTINUE

READ TRANSACTION DATA FROM INPUT FILE.

READ (UNIT = 1, FMT = (AB8, C4)), (END = 500) HOLD
SAVE = HOLD
CALL GETSTR (HOLD, ' ', SHOLD, IPOS)
IF (IPOS .NE. 2) GOTO 290
ENDIVH (SHOLD, IERR)
IF (IERR .EQ. 1) GOTO 290
IF (A.LT.1000 AND A.GT.5) GOTO 290
TRNSAC (1, 1) = A
DO 290 J = 2, 111 (A)
  CALL GETSTR (HOLD, ' ', SHOLD, IPOS)
  IF (IPOS .EQ. 0 AND J .NE. FF1 (A)) GOTO 290
  IF (IPOS .GT. 7 OR (IPOS .LT. 2 AND J .NE. FF1 (A))) GOTO 290
  TRNSAC (I, J) = REALSV (SHOLD, IERR)
  IF (IERR .EQ. 1) GOTO 290
  IF (J .EQ. 2) THEN
    TRANV = INT (TRNSAC (I, 2))
    TRNSAC (I, 13) = (FDAY (I) - SDAY)
  ENDIF
GOTO 280
290 WRITE (6, 390) SAVE
ERROR = 'YES'
GOTO 300

300 FORMAT (' ERROR IN FOLLOWING INPUT LINE\'/, A80)
300 CONTINUE
300 CONTINUE

CHECKS TO SEE IF PROGRAM READ THE NUMBER OF TRANSACTIONS
STATED IN THE FIRST LINE OF THE INPUT FILE.

IF 1 .LE. NMTRAN THEN
  WRITE (6, 400) NMTRAN, 1-1
ERROR = 'YES'
END IF

THE DATA FILE INDICATED 1,17. TRANSACTIONS WERE INCLUDED IN THE FILE. ONLY 1,17 TRANSACTIONS ARE ACTUALLY INCLUDED. PLEASE CHECK INPUT FILE. (}
END OF SUBROUTINE FILEIN

START OF SUBROUTINE EDIT

EACH EDIT CHECK IS UNDERLINED WITH '*'S.

SUBROUTINE EDIT(TNSAC, ELEMENT, NMCSLT, STDATE, STCOST, FRFRAT, NMTRAN, COSTEL, ERROR, CCOST, HCGRG, T01FYD, T01PROF, PROFIT, LSTDEL, TODAYS, T01COS)

REAL TNSAC(325,20), ELEMENT(10,5), FRFRAT, CCOST, T01PROF, PROFIT

INTEGER NMCSLT, STDATE, NMTRAN, LSTDEL, DAPMO(12)

* CHARACTER*7 COSTEL(10,2)
* CHARACTER*8 ERROR
* INTEGER HCGRG, HLCGRG
* REAL COSTD, .F., .FL, .TM


ERROR = 'NO'
T01FYD = 0
CCOST = 0
HCGRG = 0
T01COS = STCOST

 IF (STCOST.LT.1.0 OR NMCSLT.LT.1.) THEN

 FORMAT( 'THE START-UP COST CANNOT BE A NEGATIVE NUMBER. ' )
 WRITE(6,10)
 ERROR = 'YES'
 WRITE(6,40)
 END IF

 IF (NMCSLT.GT.10.0 OR NMCSLT.LT.1.) THEN

 FORMAT( 'THE NUMBER OF COST ELEMENTS MUST BE BETWEEN 1 AND 10 ' )
 WRITE(6,20)
 ERROR = 'YES'
 WRITE(6,40)
 END IF

 IF (FRFRAT.LT.0.0 OR FRFRAT.GT.1.0 ) THEN

 FORMAT( 'THE PROFIT RATE MUST BE BETWEEN 0 AND 1. ' )
 WRITE(6,30)
 ERROR = 'YES'
 WRITE(6,40)
 END IF

 DO 1 = 1, NMTRAN

 IF (TNSAC(1,2) .GT. 100000)

 FORMAT( 'THE TRANSACTION DATE ' )
 WRITE(6,30) TNSAC(1,2), IMOPC
ERROR = YES
WRITE(6, 4000)
GOTO 50
END IF

FORMAT 1: THE TRANSACTION DATE, FB.0, HAS AN ILLEGAL DAY, 13,
WHITE(6, 4000) TRANSAC(1, 2), IDAPC
ERROR = YES
WRITE(6, 4000)
END IF

************************************************************

CHECK 3 COST STATEMENT DATA.

************************************************************

IF TRANSAC(1, 1), ED.1) THEN
DO 700 J=1, NMCSTL
IF (TRANSAC(1, J+2), GT, 99999 OR TRANSAC(1, J+2), LT, -99999) THEN
CAS&4500
END IF

 FORMAT 6: COST AND DELIVERY TRANSACTIONS MUST NOT HAVE A VALUE
 (a) VALUES LARGER THAN SIX DIGITS OR A VALUE THAT IS TOO LARGE.
 WRITE(6, 4000) TRANSAC(1, 2)
ERROR = YES
WRITE(6, 4000)
END IF

************************************************************

NEXT LINE SUMS COST DATA TO DETERMINE TOTAL COST.
TLCOST = TLCOST + TRANSAC(1, J+2)
IF (TRANSAC(1, J+2), LT, 0) AND (ELEMENT(J, 1), NE. 0) THEN
CAS&4500
END IF

 FORMAT 7: A NEGATIVE COST IS PERMITTED ONLY FOR SUBCONTRACTOR
 AS A NEGATIVE VALUE AND IS NOT A SUBCONTRACTOR COST ELEMENT.
 WRITE(6, 4000) TRANSAC(1, 2)
ERROR = YES
WRITE(6, 4000)
END IF

************************************************************

NEXT LINE SUMS COST DATA BY COST ELEMENT.
ELEMENT(J, 5) = ELEMENT(J, 5) + TRANSAC(1, J+2)
CONTINUE
END IF

************************************************************

CHECKS ACTUAL PROGRESS PAYMENT DATA.

************************************************************

IF (TRANSAC(1, 1), ED.1) THEN
ACPP=1
IF (TRANSAC(1, 3), LT, 0) THEN
CAS&4500
END IF

 FORMAT 10: AN ACTUAL PROGRESS PAYMENT MUST HAVE A VALUE, 
 & GREATER THAN 0. THE PROGRESS PAYMENT DATED (FB.0), IS LESS THAN COST.
 WRITE(6, 4000) TRANSAC(1, 2)
ERROR = YES
WRITE(6, 4000)
END IF

IF (TRANSAC(1, 4), LT, 0 OR TRANSAC(1, 4), GT, 1) THEN
CAS&4500
END IF

 FORMAT 11: THE TRANSACTION DATED (FB.0) VIOLATES THIS RULE.
 FORMAT: DO NOT INPUT ANY PLANNED PROGRESS PAYMENTS INTO THE
CHARGE FILE. THE TRANSACTION DATED "F.B.W." MUST BE REMOVED.
WRITE(5,'(F8.0)') TNSAC(1,2)
ERROR= YES
WRITE(6,4000)
END IF
IF (TNSAC(1,4),GT.), THEN

 FORMAT: THE DATA IS NOT IN CHRONOLOGICAL ORDER. CHECK TNSAC
4,5,6,7 DATED "F.B.W." AND "F.B.W." TO CONNECT, /
WRITE(5,'(F8.0)') TNSAC(1,2), TNSAC(1-1,2)
ERROR= YES
WRITE(6,4000)
END IF
IF (TNSAC(1,1),EQ.4, OR, TNSAC(1,1),EQ.5) THEN

 FORMAT: A DELIVERY PAYMENT MUST HAVE A VALUE LARGER THAN 1.
2, 3, 4, 5, 6 BUT NOT CONTAIN MORE THAN 6 DIGITS. THE TRANSACTION DATED ,
1, ERROR, VIOLATES THIS RULE. /
WRITE(6,900) TNSAC(1,2)
ERROR= YES
WRITE(6,4000)
END IF
IF (TNSAC(1,1),EQ.5) THEN

 FORMAT: THE LIQUIDATION RATE ASSOCIATED WITH AN ACTUAL , (C800)150
0. DELIVERY MUST BE BETWEEN 0 AND 1. THE TRANSACTION DATED "F.B.W., CHAS04160
8 VIOLATES THIS RULE. /
WRITE(6,100) TNSAC(1,2)
ERROR= YES
WRITE(6,4000)
END IF
END IF
END IF
IF (TNSAC(1,1),LT.1, OR, TNSAC(1,1),GT.5) THEN

 FORMAT: THE TRANSACTION DATED "F.B.W." HAS AN ILLEGAL , /
0. TRANSACTION CODE. THE CODE MUST BE A 1, 2, 3, 4, OR 5. /
WRITE(6,110) TNSAC(1,2)
ERROR= YES
WRITE(6,4000)
END IF
COSTFD=TOTFD / (1+FRMAT)-1
TH=TOTCO$+1.00000
IF (COSTFD,GT.THI) THEN

 FORMAT: COST REIMBURSEMENT CANNOT BE GREATER THAN INCREASED
0. COST. THE TRANSACTION DATED "F.B.W." VIOLATES THIS RULE. /
WRITE(6,120) TNSAC(1,2)
ERROR= YES
WRITE(6,4000)
END IF
END IF
IF (TNSAC(1,1),LT.1, OR, TNSAC(1,1),GT.5) THEN


FORMAT THE TOTAL COST AT THIS POINT FROM COST STATEMENTS, TOTALi.

FORMAT THE TOTAL DELIVERIES AT THIS POINT, TOTALd.

FORMAT COST REIMBURSED AT THIS POINT BASED ON DELIVERY PRICES, CHS 850.0.

WRITE*, "1, 0.0 THREA'I, 0.0"
WRITE*, "1, 0.0 TOTCOS"
WRITE*, "1, 100.0 TOTPR
WRITE*, "1, 100.0 COSTED
ERROR= YES
WRITE*, 4000
END IF
CONTINUE
CONTINUE
DO 1000 I=1, N6.
IF (COSTELM 1, 2).NE. INCOR ) AND (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
1000 WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
**************************************************************************

WRITE*, 554
**************************************************************************

IF (COSTELM 1, 2).NE. INCOR ) OR (COSTELM 1, 2).NE. PNC ) AND (COSTELM 5, 5).NE. 0)
CONTINUE
200 FORMAT ('THE PROFIT RATE COMPUTED FROM THE DATA FILE ('',FS,2,') DOES NOT MATCH THE PROFIT RATES (--,FS,2,') IN THE FIRST LINE OF THE DATA FILE.')
210 FORMAT ('',T6,'COMPUTED PROFIT BASED ON THE FOLLOWING VALUES COMPARED TO THOSE FROM DATA INPUTS:')
220 FORMAT ('',T11,TOTAL PRICE FROM DELIVERY PRICES ,T4B,110,T,111, TOTAL COST FROM COST STATEMENTS ,T4B,110,,T,111, PROFIT DOLLARS ,T4B,110)
223 223 WRITE (6,200) PROFIT*100,ERRAT*100
223 223 WRITE (6,210)
223 223 WRITE (6,220) INT(TOTFYD), INT(TOTCOS), INT(TOPROF)
223 223 ERROR= YES
223 223 WRITE (6,400)
223 223 END IF

************************************************************************************

SUMS COST STATEMENT DATA WHEN ERRORS ARE DETECTED IN
EDIT SUBROUTINE.

**************************************************************

IF (ERROR,EO,YES) THEN
   DO 100 I=1,NUMTRAN
      IF (TRANS(1,1).EQ.,) THEN
         DO 10 J=1,NUMSTL
            TRANS(1,14)=TRANS(1,14)+TRANS(1,1,2)
      10   CONTINUE
   END IF
100  CONTINUE
END IF
RETURN
END

***********  END OF SUBROUTINE EDIT  ***********

***********  START OF SUBROUTINE DATHIN  ***********

SUBROUTINE DATHIN,FRLAG,DELLAG,FLXFLG,FGRATE,FGRF,FLG,
  INTLDY,TTYFLG,CSTFLG,FERUN,TWOF,L)
  INTEGER FRLAG,DELLAG,FGRALG,INTLDY,TWOF)
  CHARACTER*2 TTYFLG,CSTFLG,FLXFLG
  REAL FGRF
10  FORMAT( THE FOLLOWING DATA IS REQUESTED OF THE USER AT THE )
20  FORMAT( TERMINAL SO DIFFERENT CALCULATIONS CAN BE MADE USING )
30  FORMAT( THE SAME DATA FILE )
40  FORMAT( WHAT IS THE PROGRESS PAYMENT LAG TIME IN DAYS? (TIME )
50  FORMAT( FROM WHEN STATEMENT IS ISSUED UNTIL PAYMENT IS RECEIVED )
60  FORMAT( ELED) )
70  FORMAT( WHAT IS THE DELIVERY PAYMENT LAG TIME IN DAYS? (TIME )
80  FORMAT( FROM WHEN DELIVERY IS MADE UNTIL PAYMENT IS RECEIVED )
90  FORMAT( WOULD YOU LIKE TO SPECIFY A CERTAIN PROGRESS PAYMENT )
100 FORMAT( RAGE FOR THESE CALCULATIONS? A NO ANSWER IMPLIES YOU )
110 FORMAT( WANT THE COMPUTER TO CALCULATE THE FLEXIBLE PROGRESS )
120 FORMAT( RATE USING THE CONTRACTOR INVESTMENT RATIO. (YES / NO )
130 FORMAT( WHAT PROGRESS PAYMENT RATE WOULD YOU LIKE TO SPECIFY? )
140 FORMAT( PLEASE USE THE FORMAT 0,XX )
150 FORMAT( AFTER HOW MANY DAYS WOULD YOU LIKE THE ALTERNATE )
160 FORMAT( LIQUIDATION RAGE USED IN CALCULATIONS? VALUE MUST BE )
170 FORMAT( GREATER THAN OR EQUAL TO .65 )
180 FORMAT( WOULD YOU LIKE TO SEE A TRANSACTION REPORT? (YES / NO )
190 FORMAT( WOULD YOU LIKE TO SEE A RECAP OF THE COST REPORT DATA )
200 FORMAT( THE PROGRESS PAYMENT LAG MUST BE POSITIVE AND CANNOT EXCEED )
210 FORMAT( THE DELIVERY PAYMENT LAG MUST BE POSITIVE AND CANNOT EXCEED )
220 FORMAT( THE ALTERNATE LIQUIDATION RATE CANNOT BE USED UNTIL )
230 FORMAT( ONE YEAR HAS ELAPSED ON THE CONTRACT. (365 DAYS )
240 IF (FERUN,EO,YES) THEN
250   WRITE(6,50)
260   WRITE(6,60)
270   WRITE(6,70)
270   RETURN
280 END

***********  THE END OF THE PROGRAM  ***********
C IMPORTANT HIGHLIGHTS TWO COST STATEMENTS MAY OCCUR BETWEEN PROGRESS PAYMENTS.
C
IF (FRGLAG.GE.28) THEN
  TWOFLG=1
ELSE
  TWOFLG=0
END IF
C
C ASKS FOR DELIVERY PAYMENT LAG.
C
410 WRITE(6,290)
WRITE(6,80)
WRITE(6,81)
READ(S,FMT='*'), DELLAG
IF (DELLAG.GT.60) THEN
  WRITE(6,270)
  GO TO 410
END IF
C
C ASKS WHETHER FLEX RATE IS DESIRED.
C
480 WRITE(6,290)
WRITE(6,130)
WRITE(6,131)
WRITE(6,132)
WRITE(6,133)
READ(S,FMT='(A3)'), FLXFLG
IF (FLXFLG(1:1),NE.'Y',AND.FLXFLG(1:1),NE.'N') THEN
  GO TO 480
END IF
C
C ALLOWS SPECIFIC PROGRESS PAYMENT RATE TO BE USED IN LIEU OF FLEX RATE.
C
550 WRITE(6,290)
WRITE(6,170)
WRITE(6,171)
READ(S,FMT='*'), FRGRTE
IF (FRGRTE.GT.1.0) THEN
  FORMAT(' THE PROGRESS PAYMENT RATE MUST BE BETWEEN 1 AND 0."
WRITE(6,570)
GO TO 550
END IF
FRGRTE=1.0
ELSE
FRGRTE=FRGRTE
WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.

C

***********************************************************************

C

WHEN TO SWIST TO ALTERNATE LIQUIDATION RATE.
ACFRG=I
CNTFLG=I
END IF
I=0
CONTINUE
END IF

******************************************************************************
ADD DELIVERY LAG TO EACH DELIVERY DATE INPUT TO
ESTABLISH DATE ON WHICH PAYMENT IS RECEIVED.
******************************************************************************
DO 150 I=1,NMTRAN
IF (TRANSAC(I,1).EQ.4 .OR. TRANSAC(I,1).EQ.5) THEN
TRANSAC(I,13)=TRANSAC(I,13)+DELLAG
Z=INT(TRANSAC(I,13))+STDAY
TRANSAC(I,2)=D.DATE(I)
END IF
150 CONTINUE

******************************************************************************
CREATE A PLANNED PROGRESS PAYMENT TRANSACTION FOR EACH
COST STATEMENT NOT COVERED BY AN ACTUAL PROGRESS PAYMENT.
******************************************************************************
DO 200 I=ACFRG,NMTRAN
IF (TRANSAC(I,1).EQ.3 .AND. (INT(TRANSAC(I,13))+FRSLAG+STDAY).LT.2) .OR. TRANSAC(I,1).EQ.6) THEN
FRLOG=FRLOG+1
NMTRAN=NMTRAN+1
TRANSAC(NMTRAN,1)=2
TRANSAC(NMTRAN,13)=TRANSAC(I,13)+FRSLAG
TRANSAC(NMTRAN,2)=D.DATE(I)
FRLOG=FRLOG+1
ELSE
GO TO 200
END IF

******************************************************************************
CHECKS TO SEE IF PLANNED PROGRESS PAYMENTS ARE AT LEAST
28 DAYS APART. MAKES SURE PROGRESS PAYMENTS ARE HELD ONCE PER MONTH.
******************************************************************************
IF (FRLOG.GT.1) THEN
DAYS=TRANSAC(NMTRAN,13)-TRANSAC(NMTRAN-1,13)
END IF
IF (DAYS.LT.28 .AND. FRLOG.GT.1) THEN
WRITE(6,300) TRANSAC(NMTRAN,2)
ERROR= YES
WRITE(6,400)
END IF
200 CONTINUE

***** NEXT LINE SORTS TRANSACTIONS AFTER ALL NEW TRANSACTIONS

CALL SORT (TRANSAC,NMTRAN,2,13)

Print: PLANDED PROGRESS PAYMENTS CANNOT OCCUR LESS THAN 28 DAYS APART. THE TRANSACTION DATED ,/F7.0, VIOLATES THIS RULE. CHANGE INPUT FILE ,/F7.0, TO INSURE COST STATEMENTS ARE AT LEAST 28 DAYS APART.
SUBROUTINE ACTIFY(TRANSAC,A,OUTPUT,FLOATD,CSTFLT,FRFRA,ALTLDY,STCOST,STCOST)
REAL TRANSAC(25,20),OUTPUT(25,10),CSTFLT(1000,2),FRFRA,STCOST
INTEGER A,FLOATD,ALTLDY,STCOST

**DETERMINES PAID INCURRED COST AT THE TIME OF THE TRANSACTION.**

DO 1=1,FLOATD
  IF (CSTFLT(I,I).LE.TRNSAC(I,1)) THEN
    TRANSAC(I,1)=CSTFLT(I,1)+TRANSAC(I,1)
  END IF
  CONTINUE
  IF (TRANSAC(A,17).LT.ALTLDY) THEN
    TRANSAC(A,19)=TRANSAC(A,4) / (1+FRFRA)
  ELSE
    TRANSAC(A,19)=TRANSAC(A,4)
  END IF

OUTUT(A,1)=OUTPUT(A-1,1)
OUTUT(A,2)=OUTPUT(A-1,1)-TRANSAC(A,17)
OUTUT(A,3)=TRANSAC(A,17)
OUTUT(A,4)=OUTPUT(A-1,4)+TRANSAC(A,3)
OUTUT(A,5)=OUTPUT(A-1,5)
OUTUT(A,6)=OUTPUT(A,4)+OUTPUT(A,5)
OUTUT(A,7)=OUTPUT(A-1,7)
OUTUT(A,8)=OUTPUT(A,6)+OUTPUT(A,7)
OUTUT(A,9)=OUTPUT(A,3)-OUTPUT(A,6)
OUTUT(A,10)=OUTPUT(A-1,10)+TRANSAC(A,3)

**NEXT LINE IS A COUNTER FOR PROGRESS PAYMENTS. USED IN SUBROUTINE FRGPAY IF PROGRESS PAYMENT LAG EXCEEDS COST.**

**STATEMENT PERIOD. USED TO MATCH PROGRESS PAYMENTS WITH PROPER COST STATEMENT DATA.**

END

**END OF SUBROUTINE ACTIFY**

**START OF SUBROUTINE FRGPAY**
SUBROUTINE FRGRAY (TRNSAC, A, FRGRT, CSTFLT, FLOATD, OUTPUT, TOTCHG, SUMCST, TOTFYD, ALTLDY, FRGRT, STCOST, FLINT, USFLD, TWOFLG)

REAL FRM15, FRM19, FRM20, FRM21E, FRM25, PAYMT
REAL TRANSAC(5, 20), FRGRTE, CSTFLT(1000, 2), OUTPUT(525, 111, TOTCHG, SUMCST, TOTFYD, STCOST

IN ENDF

C ***** COUNTER DESCRIBED ABOVE AT END OF SUBROUTINE ACFPY.
FFCNY=FFCNT+1
C ***** NEXT 14 LINES DO SAME THING AS DESCRIBED ABOVE IN SUBROUTINE ACFPY.
C ***** HLFFY.

TRNSAC(A, 15) = TRNSAC(A-1, 15)
TRNSAC(A, 16) = TRNSAC(A-1, 16)
DO 100 I = 1, FLOATD
   IF (CSTFLT(I, 1) .LE. TRNSAC(A, 17)) THEN
      TRNSAC(A, 17) = TRNSAC(A, 17) + CSTFLT(I, 2)
   END IF
100 CONTINUE

TRNSAC(A, 17) = TRNSAC(A, 17) + STCOST
TRNSAC(A, 18) = FRGRTE
IF (TRNSAC(A, 17) .LT. ALTLDY) THEN
   TRNSAC(A, 19) = TRNSAC(A, 18)
ELSE
   TRNSAC(A, 19) = TRNSAC(A, 18) / (1 + FRFRT)
END IF

******************************************************************************
MATHES PROGRESS PAYMENT TO APPENDED COST STATEMENT DATA
WHEN PROGRESS PAYMENT LAG EXCEEDS COST STATEMENT PERIOD.

NOTE: ALL VARIABLES FROM EquATE TO LINE NUMBERS ON
DD FORT 1195 (PROGRESS PAYMENT FORM).
******************************************************************************

IF (TWOFLG.EQ. 1) THEN
   FFM15 = FRM15 * CSTDAY(FFCNT), 15 * TRNSAC(A, 18) + TRNSAC(CSTDAY(FFCNT)
   5, 16)
   GO TO 110
END IF

FRM16 = TRNSAC(A, 15) * TRNSAC(A, 18) + TRNSAC(A, 16)
110 FRM16 = TOTFYD * TransAC(A, 19)
IF (FRM16.EQ. LT, FRM15) THEN
   FFM15 = FRM16
   END IF

FRM19 = FRM15 - OUTFUT(A-1, 4)
IF (TWOFLG.EQ. 1) THEN
   FFM20E = TRNSAC(CSTDAY(FFCNT), 15) - SUMCST) * TRNSAC(A, 18) + TRNSAC(CSTDAY(FFCNT)
   PAY(FFCNT), 16)
   GO TO 120
END IF

FRM20E = (TRNSAC(A, 15) - SUMCST) * TRNSAC(A, 18) + TRNSAC(A, 16)
120 FRM21E = (TOTFYD - TOTCHG) * TRNSAC(A, 19)
IF (FRM21E .LT. FRM20E) THEN
   FFM20E = FRM21E
   END IF

FM20E = FRM20E - OUTFUT(A-1, 10)
IF (FRM5 .LT. FRM19) THEN
   FRM19 = FRM25
   END IF

FMMT = FRM19

******************************************************************************
FILLS IN OUTPUT FOR TRANSACTION REPORT.
******************************************************************************
**OUTPUT(A,1) = OUTPUT(A-1,1)**
**OUTPUT(A,2) = OUTPUT(A,1) + TRANSA(A,17)**
**OUTPUT(A,3) = TRANSA(A,17)**
**OUTPUT(A,4) = OUTPUT(A-1,4) + PAYMENT**
**OUTPUT(A,5) = OUTPUT(A-1,5)**
**OUTPUT(A,6) = OUTPUT(A,4) + OUTPUT(A,5)**
**OUTPUT(A,7) = OUTPUT(A-1,7)**
**OUTPUT(A,8) = OUTPUT(A,6) + OUTPUT(A,7)**
**OUTPUT(A,9) = OUTPUT(A,3) + OUTPUT(A,6)**
**OUTPUT(A,10) = OUTPUT(A-1,10) + PAYMENT**
RETURN
END

******* END OF SUBROUTINE PRGFAY *******

******* START OF SUBROUTINE COSTST *******

SUBROUTINE COSTST: TRANSA, A, NMCSTL, ELEMENT, FLOATD, CSTFLT, OUTPUT,
SUBCN, UDFCN, UDFLT, UFDSUB, FGRT, STCOST, CSTT, CSTDAY, TWOFLG
REAL TRANSA(329, 20), ELEMENT(10, 5), CSTFLT(1000, 2), OUTPUT(329, 10),
UDFHT(1000, 2), UFDSUB(1000, 2), FGRT, STCOST
INTEGER A, NMCSTL, FLOATD, SUBCN, UDFCN, CSTT, CSTDAY (170), TWOFLG
REAL PAID, UNPAID

DETERMINES PAID INCURRED COST ELIGIBILITY FOR PROGRESS PAYMENTS.

************ COST ELIGIBILITY CODE 'INCUR'

IF: ELEMENT(I,1),ED,1) THEN
  CSTFLT(FLOATD,1) = TRANSA(A,13) + ELEMENT(I,2)
  CSTFLT(FLOATD,2) = TRANSA(A,11) + TRANSA(A,12)
  TRANSA(A,15) = TRANSA(A,15) + TRANSA(A,11)
END IF

************ COST ELIGIBILITY CODE 'PAIDC'

IF: ELEMENT(I,1),ED,2) THEN
  PAID = TRANSA(A,12) + ELEMENT(I,3)
  UNPAID = TRANSA(A,12) + (1 - ELEMENT(I,3))
  CSTFLT(FLOATD,1) = TRANSA(A,11) + ELEMENT(I,2)
  CSTFLT(FLOATD,2) = UNPAID
  UDFCN = UDFCN + 1
  UDFLT(UDFCNT,1) = CSTFLT(FLOATD,1)
  UDFLT(UDFCNT,2) = UNPAID
  FLOATD = FLOATD + 1
  CSTFLT(FLOATD,1) = TRANSA(A,13) + ELEMENT(I,4)
  CSTFLT(FLOATD,2) = PAID
  TRANSA(A,15) = TRANSA(A,15) + PAID
END IF

************ COST ELIGIBILITY CODE 'SUBCT'

IF: ELEMENT(I,1),ED,3) THEN
  PAID = TRANSA(A,12) + ELEMENT(I,3)
  UNPAID = TRANSA(A,12) + (1 - ELEMENT(I,3))
  CSTFLT(FLOATD,1) = TRANSA(A,11) + ELEMENT(I,2)
  CSTFLT(FLOATD,2) = UNPAID
  UDFCN = UDFCN + 1
  UFDSUB = UDFCN + 1
  CSTFLT(FLOATD,1) = CSTFLT(FLOATD,1)
END IF
**580 CONTINUE**

*---------------------------------------------*
C Puts unaided floated costs for cost eligibility code C 'PAID' into cost eligible for progress payments after C floated period expires.
C
C DO 450 I=1,UPDCNT
C IF(UPDFLT(I,1).LE.TRNSAC(A,15)) THEN
C TRNSAC(A,15)=TRNSAC(A,15)+UPDFLT(I,2)
C UPDFLT(I,1)=1000000
C UPDFLT(I,2)=0
C END IF
C
C 450 CONTINUE

**520 CONTINUE**

*---------------------------------------------*
C Puts unaided floated costs for cost eligibility code C 'SUBCT' into cost eligible for 100% progress payments C after floated period expires.
C
C DO 520 I=1,SUBCNT
C IF(UPDFSU(I,1).LE.TRNSAC(A,15)) THEN
C TRNSAC(A,16)=TRNSAC(A,16)+UPDFSU(I,2)
C UPDFSU(I,1)=1000000
C UPDFSU(I,2)=0
C END IF
C
C 520 CONTINUE

*---------------------------------------------*
C Determines paid incurred cost at the time of the C transaction.
C
C DO 570 I=1,FLOATD
C IF(CSTFLT(I,1).LE.TRNSAC(A,15)) THEN
C TRNSAC(A,17)=TRNSAC(A,17)+CSTFLT(I,2)
C END IF
C
C 570 CONTINUE

TRNSAC(A,17)=TRNSAC(A,17)+STCOST

*---------------------------------------------*
C Fills in output for transaction report. Also rolls C costs eligible for progress payments forward and C keeps track of progress payment and liquidation rates.
C
IF(A.GT.1) THEN
TRNSAC(A,15)=TRNSAC(A,15)+TRNSAC(A-1,15)
TRNSAC(A,16)=TRNSAC(A,16)+TRNSAC(A-1,16)
TRNSAC(A,18)=TRNSAC(A-1,18)
TRNSAC(A,19)=TRNSAC(A-1,19)
OUTFUT(A,1) =OUTFUT(A-1,1)+TRNSAC(A,14)
OUTFUT(A,4) =OUTFUT(A-1,4)
OUTFUT(A,5) =OUTFUT(A-1,5)
OUTFUT(A,6) =OUTFUT(A-1,6)
OUTFUT(A,7) =OUTFUT(A-1,7)
OUTFUT(A,8) =OUTFUT(A-1,8)
OUTFUT(A,11) =OUTFUT(A-1,10)
ELSE
TRNSAC(A,15)=TRNSAC(A,15)+STCOST
ENDIF
**PEND OF SUBROUTINE COSTST **

**START OF SUBROUTINE FUNDDE **

**SUBROUTINE FUNDDE(TNAME,A,ALTLO,FFRAT,ALLOFLG,CSFTLT,FLGAD,FTBCH,SUMST,OSTF,PROFT,LSTDEL,DELLAG,STCOST,TODAYS,STDAY,**

**NEXT THREE LINES PERFORM SAME OPERATIONS AS DESCRIBED IN**

**SUBROUTINE ACIFEF.**

**NEXT LINE DETERMINES DATE OF LAST DELIVERY TRANSACTION IN**

**TERMS OF DAYS AFTER AWARD.**

**NEXT SIX LINES DETERMINE PAID INCURRED COST AT THE TIME**

**OF THE TRANSACTION AS PREVIOUSLY DESCRIBED.**

**DO 110 I=1,FGAD**

**IF (CSFTLT(1,1)**

**TNAME(A,17)=TNAME(A,17)+CSFTLT(1,2)**

**END IF**

**CONTINUE**

**TNAME(A,17)=TNAME(A,17)+STCOST**

**CONVERTS TO ALTERNATE LIQUIDATION RATE AND MAKES PROFIT**

**CATCH-UP ADJUSTMENT.**

**IF (TNAMEDA,16,ST,ALTLO) THEN**

**TNAME(A,19)=TNAME(A,18)+(1+FFRAT)**

**IF (ALLOFLG,EQ.0.) THEN**

**ALLOFLG=ALOFLG-OUTPUT(A-1,7)**

**CONTINUE**

**ENDIF**

**END IF**

**END SUBROUTINE FUNDDE **

**END OF SUBROUTINE COSTST **
TRANSA(C(A,19)) = TRANSA(C(A,18))
END IF

********************************************
CHECKS TO SEE IF LIQUIDATION AMOUNT EXCEEDS ULIQUID PROGRESS PAYMENT BALANCE.
********************************************

LIQUIDAT = TRANSA(C(A,2)) - TRANSA(C(A,19))
IF (LIQUIDAT .GT. OUTFUT(A-1,10)) THEN
  LIQUIDAT = OUTFUT(A-1,10)
ENDIF

IF (INT(TRANSA(C(A,2))).NE. (1+DATE(1))) THEN
  LFLG = 1
ENDIF

ENDIF

****** NEXT 4 LINES DETERMINE DELIVERY PAYMENT AND THE SPLIT*******

****** BETWEEN COST AND PROFIT,*******

NETFAY = TRANSA(C(A,3)) - LIQUIDAT
COST = TRANSA(C(A,3)) / (1+PREMRAT)
COSTFD = COST - TRANSA(C(A,12)) * COST
PREMRAT = NETFAY / COSTFD

*************************************************************************
COMPUTES WHAT PROFIT WOULD BE AT ALTERNATE RATE PRIOR TO SWITCHOVER. USED TO MAKE ONE-TIME PROFIT ADJUSTMENT.
*************************************************************************

IF (LFLG.EQ.1) THEN
  ALTFFR = ALTFFR + TRANSA(C(A,3)) / (1+PREMRAT)
ENDIF

****** FILLS IN OUTPUT FOR TRANSACTION REPORT.*******

OUTFUT(A,1) = OUTFUT(A-1,1)
OUTFUT(A,2) = OUTFUT(A-1,4) + OUTFUT(A,1) - TRANSA(C(A,12))
OUTFUT(A,3) = TRANSA(C(A,13))
OUTFUT(A,4) = OUTFUT(A-1,4)
OUTFUT(A,5) = OUTFUT(A-1,5) + COSTFD
OUTFUT(A,6) = OUTFUT(A-1,6) + OUTFUT(A,5)
OUTFUT(A,7) = OUTFUT(A-1,7) + PREMRAT * CUTFUT
OUTFUT(A,8) = OUTFUT(A-1,8) + OUTFUT(A,7)
OUTFUT(A,9) = OUTFUT(A-1,9) + OUTFUT(A,8)
OUTFUT(A,10) = OUTFUT(A-1,10) - LIQUIDAT + CUTFUT
TOTCHG = TOTCHG + TRANSA(C(A,3))
SUMCST = SUMCST + COST

*************************************************************************
FILLS UP ALL DATA FOR LAST DELIVERY TRANSACTION, ZEROGES OUT FLOAT AND UNLIQUIDATED PROGRESS PAYMENTS. FILLS IN OUTPUT FOR TRANSACTION REPORT.
*************************************************************************

IF (INT(TRANSA(C(A,2))).EQ. (1+DATE(1))) THEN
  OUTFUT(A,1) = OUTFUT(A-1,1)
  OUTFUT(A,2) = 0
  OUTFUT(A,3) = OUTFUT(A,1)
  OUTFUT(A,4) = OUTFUT(A-1,4)
  OUTFUT(A,5) = OUTFUT(A-1,5) - OUTFUT(A,4)
  OUTFUT(A,6) = OUTFUT(A-1,6) + OUTFUT(A,4) + OUTFUT(A,5)
  OUTFUT(A,7) = OUTFUT(A-1,7) + PREMRAT * OUTFUT(A,6)
  OUTFUT(A,8) = OUTFUT(A-1,8) + OUTFUT(A,7)
  OUTFUT(A,9) = OUTFUT(A-1,9) - OUTFUT(A,8)
  OUTFUT(A,10) = 0
ENDIF

END
RETURN
END

END OF SUBROUTINE FINEEL
****** CALCULATIONS ARE THE SAME AS DESCRIBED IN SUBROUTINE FLDEL ******

SUBROUTINE ACTDEL (TRANSAC, A, FRAT, ALTLDY, ALOFLG, CSTFLT, FLOATD, STOTCHG, SUMCST, OUTPUT, LSTDEL, DELLAG, STCOST, TODAY, STDAY, ALTFFR)
REAL TRANSAC(25, 20), FRAT, CSTFLT(1000, 2), STOTCHG, SUMCST, OUTPUT(1000, 20)
STCOST, TODAY, ALTFFR
INTEGER A, ALTLDY, ALOFLG, FLOATD, STDAY, DELLAG
REAL FRFATY, LIOCAT, COST, NETFAT
TRANSAC(A, 15)=TRANSAC(A-1, 15)
TRANSAC(A, 16)=TRANSAC(A-1, 16)
TRANSAC(A, 18)=TRANSAC(A-1, 18)
TRANSAC(A, 19)=TRANSAC(A, 19)
DO 89 = 1, FLOATD
IF (CSTFLT(I, 1), LE, TRANSAC(A, 17)) THEN
TRANSAC(A, 17)=TRANSAC(A, 17)+CSTFLT(I, 1)
END IF
89 CONTINUE
TRANSAC(A, 17)=TRANSAC(A, 17)+STCOST
OUTPUT(A, 1)=OUTPUT(A-1, 1)
LIOCAT=TRANSAC(A, 3)*TRANSAC(A, 4)
IF (LIOCAT.GT.OUTPUT(A-1, 10)) THEN
LIOCAT=OUTPUT(A-1, 10)
END IF
NETFAT=TRANSAC(A, 17)-LIOCAT
IF (LIOCAT.GT.OUTPUT(A, 3)) THEN
ALTCUP=0
IF (ALOFLG.EQ.0) THEN
ALOFLG=1
ALTCUP=ALTFRR=OUTPUT(A-1, 7)
END IF
ELSE
ALTCUP=0
END IF
COST=TRANSAC(A, 3) / (1+FRFATY)
COSTFD=COST-TRANSAC(A, 18)*COST
FRFATY=NETFAT-COSTFD
IF (ALOFLG.EQ.0) THEN
ALTFRR=ALTFRR+(TRANSAC(A, 3)-TRANSAC(A, 3)/(1+FRFATY))
END IF
OUTPUT(A, 2)=OUTPUT(A, 1)-TRANSAC(A, 17)
OUTPUT(A, 3)=TRANSAC(A, 17)
OUTPUT(A, 4)=OUTPUT(A-1, 4)
OUTPUT(A, 5)=COSTFD+OUTPUT(A-1, 5)
OUTPUT(A, 6)=OUTPUT(A, 4)+OUTPUT(A, 5)
OUTPUT(A, 7)=OUTPUT(A-1, 7)+FRFATY+ALTCUP
OUTPUT(A, 8)=OUTPUT(A, 6)+OUTPUT(A, 7)
OUTPUT(A, 9)=OUTPUT(A, 3)-OUTPUT(A, 6)
OUTPUT(A, 10)=OUTPUT(A-1, 10)-LIOCAT+ALTCUP
TSTOCHG=TSTOCHG+TRANSAC(A, 3)
SUMCST=SUMCST+COST
=INT(TODAY+STDAY)+DELLAG

******************************************************************************

ROLLS UP ALL DATA FOR LAST TRANSACTION. (IT IS EXTREMELY UNLIKELY THAT AN ACTUAL DELIVERY WOULD BE THE LAST TRANSACTION. HOWEVER, IT MAY HAPPEN IF SOMEONE Chooses TO RUN A COMPLETED CONTRACT TO SEE WHAT THE CONTRACT

******************************************************************************
INVEST=

INPUT I 

OUTPUT 0.4*OUTPUT(1,4)

OUTPUT(5)*OUTPUT(1,5) = OUTPUT(1,6)

INPUT I + 1 = OUTPUT(1,7)

INPUT I - OUTPUT(1,8)

END IF

END

SUBROUTINE INVEST

INPUT I

INPUT I = OUTPUT(1,1) + OUTPUT(1,2)

INPUT I = INVEST/INPUT(I) + OUTPUT(1,3)

END

RETURN

END SUBROUTINE INVEST

STOP

END
**COMPUTES CONTRACTOR INVESTMENT RATIO.**

**END OF SUBROUTINE INVEST**

**START OF SUBROUTINE PNTOUT**

SUBROUTINE PNTOUT(HNF,FLG,FLXFLG,TRANSL,ELMEN,FLXFLG,HNF,HNF,FLG,FLXFLG,TRANSL,ELMEN,FLXFLG)

INTEGER FLG,FLXFLG,TRANSL,ELMEN,FLXFLG

CHARACTER *30 TITLE

LOGICAL CC

INTEGER NUM,1000

DATA CC / 200 /

FORMAT (/ )

FORMAT ( / )

**ESTABLISHES HEADINGS FOR TRANSACTION REPORT**

ROW(1) = DATE /
ROW(2) = TOTAL
ROW(3) = PAID
ROW(4) = COST
ROW(5) = TOTAL
ROW(6) = PROFIT
ROW(7) = REIM
ROW(8) = UNLID
ROW(9) = TRNSACT
ROW(10) = INCURRED
ROW(11) = INCURRED
ROW(12) = PROG FAY
ROW(13) = PAID AT
ROW(14) = 'COST
ROW(15) = 'PAID AT
ROW(16) = 'TOTAL
ROW(17) = 'PAID
ROW(18) = 'FRWDUSS
ROW(19) = 'CODE
ROW(20) = 'LOST
ROW(21) = FLOAT
ROW(22) = 'COST
ROW(23) = 'RECO
ROW(24) = 'DELY
ROW(25) = 'RETURN
ROW(26) = 'ORDER
ROW(27) = 'REVENUE
F- WCT
Li 05 1
WC 11 F
fr
N
TS
L WD
k6+7)

DO 550 I=1,10
FUME(I)= (DULLAKS)
NUM[I]=I

CONTINUE

FORMAT(T43, DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL )
FORMAT(TS1, VERSION DATED FEBRUARY 1, 1981)
FORMAT(TS7, TRANSACTION REPORT)
FORMAT(T41,A50)

1.2 FORMAT( 'T13', *** UNLIQUIDATED PROGRESS PAYMENTS WOULD HAVE BEEN***
IN NEGATIVE WHERE 0 APPEARS IN COLUMN 10. ***)
1.3 FORMAT( 'T13', *** CHECK PRICES OF DELIVERABLES TO END***
SIE THERE IS NO -FRONT LOADING-. ***)
COUNT=34

*******************************************************************************

** CONTROLS PRINT ROUTINE WHEN ERRORS ARE DETECTED IN THE **
** EDIT SUBROUTINE. **
*******************************************************************************

IF(EMT.EQ.ST.0) THEN
GO TO 1950
END IF

*******************************************************************************

** PRINTS TRANSACTION REPORT. **
*******************************************************************************

IF (TRANFLG.EQ. 'YES') THEN
DO 1070 I=1,NUMTRAN
IF (COUNT3.EQ.34) THEN
FG=FG+1
WRITE(7,FMT=(A1)) CC
WRITE(2,480)FG
WRITE(2,1000)
WRITE(2,1100)
WRITE(2,1200)
WRITE(2,350)
WRITE(2,1212) TITLE

WRITE(2,520) (NUM(I),I=1,10)
WRITE(2,620) (ROWC,ROWK(1),I=1,9)
WRITE(2,840) (ROWK(I),I=1,8)
WRITE(2,850) (ROWK(I),I=1,8)
WRITE(2,860) (ROWK(I),I=1,8)
WRITE(2,870) (ROWK(I),I=1,8)
WRITE(2,880) (ROWK(I),I=1,8)
WRITE(2,890) (ROWK(I),I=1,8)
WRITE(2,900) (ROWK(I),I=1,8)
WRITE(2,910) (ROWK(I),I=1,8)
WRITE(2,920) (ROWK(I),I=1,8)
WRITE(2,930) (ROWK(I),I=1,8)
WRITE(2,940) (ROWK(I),I=1,8)
COUNT=0

END
*** ESTABLISHES HEADINGS FOR ANALYTICAL REPORT. ***

120 FORMAT(157, 'ANALYTICAL REPORT')
130 FORMAT(' CONTRACT PRICE ', T31, F10.2)
140 FORMAT(' CONTRACT START DATE ', T35, 15)
150 FORMAT(' CUST OF CONTRACT ', T31, F10.2)
160 FORMAT(' DATA FILE PROFIT PERCENTAGE ', T36, F5.2)
170 FORMAT(' COMPounded PROFIT PERCENTAGE ', T36, F5.2)
180 FORMAT(' PROJECT TITLE ', T18, 450)
190 FORMAT(' TOTAL CONTRACTOR COST ', T31, F10.2)
200 FORMAT(' PROGRESS PAYMENT RATE DATA****')
230 FORMAT(' THESE VALUES REFLECT COMPUTATIONS *')
250 FORMAT(' BASED ON OVERLIQUIDATION OF IN ') 
280 FORMAT(' PROGRESS PAYMENTS. SEE NOTE AT *')
310 FORMAT(' END OF TRANSACTION REPORT. *')
320 FORMAT( 'PROGRESS PAYMENT LAG TIME IN DAYS', T41, T13)
330 FORMAT(' DELIVERY PAYMENT LAG TIME IN DAYS ', T41, T13)
340 FORMAT( 'DATE OF FINAL DELIVERY ', T41, T13)
350 FORMAT(' PROGRESS PAYMENT RATE COMPUTED BY PROGRAM ', T54, F7.2)
360 FORMAT(' PROGRESS PAYMENT RATE INPUT BY USER INTO PROGRAM ', T54, F7.2)
370 FORMAT( 'COST ELEMENT DATA')
380 FORMAT( 'NAME ', T8, F8.2, T51, T12, T41, F5.1, T51, T12, T41, SUBCONTRACT 'F
390 FORMAT( 'NAME ', T8, F8.2, T51, T12, T41, F5.1, T51, T12, T41, SUPPLIER ')
**PRINTS SUMMARY DATA IN ANALYTICAL REPORT.**

**PRINTS PROGRESS PAYMENT RATE, ORDINARY AND ALTERNATE LIQUIDATION RATES, AND CONTRACTOR INVESTMENT RATIO.**

**ALSO PRINTS WARNING IF FLEX RATE IS COMPUTED BASED ON OVERLIQUIDATION OF PROGRESS PAYMENTS.**

**IF (FLG.EQ.1) THEN**

WRITE (2, 475)
WRITE (2, 295) FRGRTE
WRITE (2, 505)
WRITE (2, 395) FRGRTE
WRITE (2, 605)
WRITE (2, 715) FRGRTE
WRITE (2, 625) RATIO
WRITE (2, 475)
ELSE
WRITE (2, 475)
WRITE (2, 295) FRGRTE
WRITE (2, 505)
WRITE (2, 305) FRGRTE
WRITE (2, 605)
WRITE (2, 715) FRGRTE
WRITE (2, 625) RATIO
WRITE (2, 475)
ENDIF
WRITE(2,500)  
WRITE(2,510)  TRNSC(NMTRAN,19)*100  
WRITE(2,500)  
WRITE(2,520)  RATIO  
WRITE(2,470)  
END IF  
WRITE(2,500)  
WRITE(2,530)  IF(HHFLG.EQ.0) THEN  
510 FORMAT(' ALL PROGRESS PAYMENTS INPUT ARE ACTUAL.')  
WRITE(2,510)  
END IF  

******************************************************************************

PRINTS COST ELEMENT DATA.  
******************************************************************************

WRITE(2,330)  
WRITE(2,230)  
WRITE(2,400)  
WRITE(2,410)  
WRITE(2,420)  
WRITE(2,530)  
DO 1320 I=1,NMCEL  
  IF(ELEMENT(I,1).EQ.1 AND ELEMENT(I,2).GE.0) THEN  
    WRITE(2,450) I,COSTEL(I,1),INT(ELEMENT(I,2))  
  END IF  
  IF(ELEMENT(I,1).EQ.1 AND ELEMENT(I,2).LT.0) THEN  
    WRITE(2,460) I,COSTEL(I,1),INT(ELEMENT(I,2))  
  END IF  
  IF(ELEMENT(I,1).EQ.0 AND ELEMENT(I,2).GT.0) THEN  
    WRITE(2,440) I,COSTEL(I,1),INT(ELEMENT(I,2)),ELEMENT(I,3),  
    DIST(ELEMENT(I,4))  
  END IF  
  IF(ELEMENT(I,1).EQ.0 AND ELEMENT(I,2).GT.0) THEN  
    WRITE(2,400) I,COSTEL(I,1),INT(ELEMENT(I,2)),ELEMENT(I,3),  
    DIST(ELEMENT(I,4))  
  END IF  
1320 CONTINUE  
WRITE(2,330)  
WRITE(2,340)弗quéLâg  
WRITE(2,350)DELLâg  
WRITE(2,360)LSTDEL  
WRITE(2,220)  
IF(HXFLG.EQ.'YES') THEN  
  WRITE(2,390)FRGRTE  
ELSE  
  WRITE(2,380)FRGRTE  
END IF  
WRITE(2,330)  

******************************************************************************

PRINTS COST RECAP REPORT.  
******************************************************************************

150 IF(DATHEY.EQ.0 OR DATHEY.EQ.1) THEN  
  COSTFLG= 'YES'  
  END IF  
  IF(COSTFLG.EQ. 'YES') THEN  
    TRNSC(1,14)=TRNSC(1,14)+STCOST  
    COUNT=36  
    DO 2190 I=1,NMTRAN  
    IF(COUNT.EQ.16) THEN  


COUNTS=0
WRITE(2,FMT=('(A1)') CC
FG=FG+1
WRITE(2,480)PG
1900 FORMAT('T$4, "RECAP OF COST STATEMENT"
105 FORMAT('T$5, "NOTE: TOTAL FOR FIRST COST STATEMENT INCLUDES START"
$F$COST"
WRITE(2,100)
WRITE(2,110)
WRITE(2,1900)
WRITE(2,1905)
WRITE(2,30)
WRITE(2,121) TITLE
WRITE(2,30)
2060 FORMAT('',T2,' DATE',T17,'#1','T28','#2','T39','#3','T50','#4','Tol.CHS16730
& #5','T72','#6','T83','#7','T95','#8','T105','#9','T114','#10','T122,
WRITE(2,2060)
END IF
2110 FORMAT('',1X,I6,1X,11(I10,1X))
2120 FORMAT('',1X,' TOTAL',1X,11(I10,1X))
IF(TRANSAC(I,1),E0.3) THEN
WRITE(2,2110)INT(TRANSAC(I,2)),INT(TRANSAC(I,3)),INT(TRANSAC(I,CHS16730
& 4)),INT(TRANSAC(I,5)),INT(TRANSAC(I,6)),INT(TRANSAC(I,7)),INT(TRANSAC(CHS16730
& 8)),INT(TRANSAC(I,9)),INT(TRANSAC(I,10)),INT(TRANSAC(I,11)),INT(TRANSAC(CHS16730
& TCOS)
COUNTS=COUNTS+1
END IF
2150 CONTINUE
WRITE(2,200)
WRITE(2,2120)INT(ELMENT(I,5)),INT(ELMENT(2,5)),INT(ELMENT(CHS16730
& 7,5)),INT(ELMENT(4,5)),INT(ELMENT(5,5)),INT(ELMENT(6,5)),INT(ELMEN(CHS16730
&T(7,5)),INT(ELMENT(B,5)),INT(ELMENT(9,5)),INT(ELMENT(10,5)),INT(TCOS)
END IF
C***********************************************************************
C PRINTS DELIVERY SUMMARY.
C***********************************************************************
IF(DATEY.EQ.2.OR.DATEY.EQ.3) THEN
COUNTS=36
DO 2200 I=1,NTRAN
IF(COUNTS.EQ.36) THEN
COUNTS=0
WRITE(2,FMT=('(A1)') CC
PG=PG+1
WRITE(2,480)PG
WRITE(2,100)
WRITE(2,110)
WRITE(2,2300)
WRITE(2,30)
WRITE(2,121) TITLE
WRITE(2,30)
WRITE(2,2310)
WRITE(2,2320)
END IF
IF(TRANSAC(I,1),E0.4.OR.TRANSAC(I,1),E0.5) THEN
WRITE(2,2350)INT(TRANSAC(I,2)),INT(TRANSAC(I,3))
COUNTS=COUNTS+1
END IF
2200 CONTINUE
C
**START OF SUBROUTINE SORT**

****** SORTS TRANSACTIONS AFTER PLANNED PROGRESS PAYMENT TRANSACTIONS/****
**DATES AND DATES FOR RECEIPT OF DELIVERY PAYMENTS ARE CREATED.**

SUBROUTINE SORT(ARNAME, ROWS, COLMNS, SORTCMS)
REAL TEMP
INTEGER ROWS, COLMNS, SORTCMS
DIMENSION ARNAME(325, 20)
DO 140 I=1, ROWS
   DO 120 J=1, (ROWS-1)
      IF (ARNAME(J+1, SORTCMS).LT. ARNAME(J, SORTCMS)) THEN
         DO 100 K=1, COLMNS
            TEMP=ARNAME(J, K)
            ARNAME(J, K)=ARNAME(J+1, K)
            ARNAME(J+1, K)=TEMP
         END DO
      END IF
      120 CONTINUE
   100 CONTINUE
140 CONTINUE
RETURN
END

**END OF SUBROUTINE SORT**

**START OF SUBROUTINE YRMODA**

****** USED WITH I DAY TO CONVERT YMMDD DATES TO DAYS AFTER AWARD**

SUBROUTINE YRMODA(YR, MO, DA, J)
INTEGER YR, DA
IF (MO.GE.3) GO TO 1
M=M+9
Y-YR-1961
GO TO 2
1 M=M-1
Y-YR-1960
2 J=J+1461
IF (J.LT.0) J=J+775
J=4*(195*M+2) / 5 +DA
RETURN
END

**END OF SUBROUTINE YRMODA**
******* START OF SUBROUTINE ROUND *******

SUBROUTINE ROUND(IRD)
REAL IRD, IFL
II = INT(IRD)
IFL = IRD - II
IF(IFL .GE. (.5)) THEN
  IRD = IRD + 1
  GOTO 13520
END IF
IF(IFL*(-1) .GE. (.5)) THEN
  IRD = IRD + (-1)
END IF
13520 RETURN
END

******* END OF SUBROUTINE ROUND *******

******* START OF INTEGER FUNCTION FDAY *******

INTEGER FUNCTION FDAY(IN)
INTEGER IYR,MO,IDA,IN
IN = IN / 10000
I = IYR * 10000
J = IN - I
MO = J / 100
I = MO * 100
IDA = J - I
CALL YRMODA(1900+IYR,MO,IDA,1)
F = I
RETURN

******* END OF INTEGER FUNCTION FDAY *******

******* START OF SUBROUTINE JYMD *******

SUBROUTINE JYMD(IYR,MO,IDA)
INTEGER IYR,MO,IDA
II = IYR / 1461 - 1
J = MOD(IYR*7,1461) / 4
1461
END
**SUBROUTINE GETSTR(STRNG,DELIM,SUBSTR,IPOS)**

**INTEGER** IPOS,LENDLM,LENSTR

**IF (IPOS.LE.1) THEN**

**LENSTR=LEN(STRNG)**

**IF (IPOS.EQ.1) THEN**

**SUBSTR=STRNG(1)**

**GOTO 100**

**ENDIF**

**SUBSTR=STRNG(1:IPOS-1)**

**ELSE**

**SUBSTR=STRNG(IPOS.LENSTR)**

**ENDIF**

**RETURN**

**END**

**ENDLESS OPT(3)**

**FUNCTION REALS(V(STRING1,IN))**

**CHARACTER** STRING1,STRING2

**CHARACTER** STRING1 STRING2

**INTEGER** I,J,IDEALL,IN,ISIGN,ISTRAN,IN,IDEISNO,IPWR

**INTEGER** IBLN,IBLN2

**REAL** REALSV

**EQUIVALENCE (STRING2,STRAY)**

**STRING2=STRING1**

**J=1**

**IBLN=0**

**IBLN2=0**

**IN=0**

**IDEALL=0**

**ISIGN=1**

**REALSV=0**

**DO 100 I=1,8,1**

**IDEISNO=ICHAR(STRAY(I))**

**IF ((IDEISNO.LT.240.OR.IMEISNO.LT.61.OR.IMEISNO.LT.85).AN.IEISNO.LT.0) THEN**

**IN=I**

**GOTO 400**

**ENDIF**

**IF (IMEISNO.EQ.75.AND.IMEISNO.EQ.0) THEN**

**IN=1**

**GOTO 400**

**ENDIF**

**IF (IMEISNO.EQ.75) IDEALL=1**
I ILECD01.EQ.95.AND.I.NE.1) THEN
IN=1
GOTO 400
ENDIF
IF(I1ECD01.EQ.96.AND.ISIGN=-1
IF(I1ECD01.EQ.64.AND.IBLN.EQ.1.AND.IBLN2.EQ.1) GOTO 100
IF(I1ECD01.EQ.64.AND.IBLN.EQ.1.AND.IBLN2.EQ.0) THEN
IN=1
GOTO 400
ENDIF
IF(I1ECD01.EQ.64.AND.IBLN.EQ.0) THEN
IN=1
IBLN2=1
GOTO 100
ENDIF
IBLN2=0
IN=IN+1
100 CONTINUE
IF(ISIGN.EQ.-1.AND.IN.EQ.2) THEN
IN=1
GOTO 400
ENDIF
IF(I1DECDL.GT.0) THEN
INLN=I1DECDL-1
1STRLN=IN-1
ELSE
INLN=IN-1
1STRLN=INLN
ENDIF
IF(ISIGN.EQ.-1) THEN
J=2
INLN=INLN-1
ENDIF
IF(IFWR=INLN-1
DO 300 I=J,1STRLN,1
IF(I1ECD01.EQ.75) THEN
GOTO 300
ENDIF
REALSV=(I1ECD01.-240.)*10.**IFWR)+REALSV
IF(IFWR=-1
300 CONTINUE
REALSV=REALSV*ISIGN
RETURN
END
FUNCTION INTSV STRING1,IN
CHARACTER*1 STRING1
CHARACTER*8 STRING2
CHARACTER*1 STRING3(8)
INTEGER 1,1J,1N,I1DECD01,IFWR,ISIGN,IBLN,IBLN2
EQUIV,ELSE=(STRING3,STRING)
STRING1=STRING1
STRING2=STRING2
STRING3=STRING3
STRING=STRING
ENDIF
IBLN=0
IBLN2=0
IN=0
1N=1
IN+1,1
END SUBROUTINE OPTC
IF(1.LE.EQ.-1 .. AND. IN.EQ.2) THEN
    IN=1
    GOTO 400
ENDIF
ISTRLN=IN-1
IF(ISTRLN.EQ.-1) THEN
    J=2
    IFWR=ISTRLN-2
ELSE
    IFWR=ISTRLN-1
ENDIF
DO 300 I=J,ISTRLN,1
    INTSTV=(IDECD(I)-240)*(10**IFWR)+INTSTV
300    IFWR=IFWR+1
    INTSTV=INTSTV*ISIGN
DATE
ILME