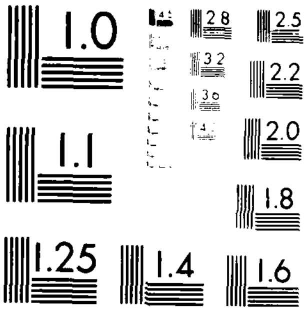
AD-A148 247   CASH FLOW COMPUTER MODEL USERS GUIDE REVISION 1(U)   1/1
OFFICE OF THE DEPUTY UNDER SECRETARY OF DEFENSE
RESEARCH AND ENGINEERING (ACQUISITION MGMT) WASHINGTON
UNCLASSIFIED   DC   01 FEB 83   F/G 9/2   NL

END
DATE
FILMED
[illegible]
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

1

DEPARTMENT OF DEFENSE

CASH FLOW COMPUTER MODEL

USERS GUIDE

REVISION 1

Superseded
AD A104091
AD A129476

CHAIRMAN, DOD CONTRACT FINANCE COMMITTEE
OFFICE OF COST, PRICING & FINANCE
OUSDRE (ACQUISITION MANAGEMENT)
PENTAGON, WASHINGTON, D. C. 20301

FEBRUARY 1, 1983

TABLE OF CONTENTS

A 1

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.

COMMERCIAL VENDORS

VENDOR CONTACTS

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

M. Rosenberg
BCS
7980 Gallows Road
Vienna, Va 22180
(703) 821-~~62100~~ 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.

1. INTRODUCTION CONT'D

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.

FLEXIBLE PAYMENT ELIGIBILITY

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.

1. INTRODUCTION CONT'D

A SAMPLE CASHII RUN

DATA FILE REQUIRED

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.

RECAP OF COST STATEMENTS

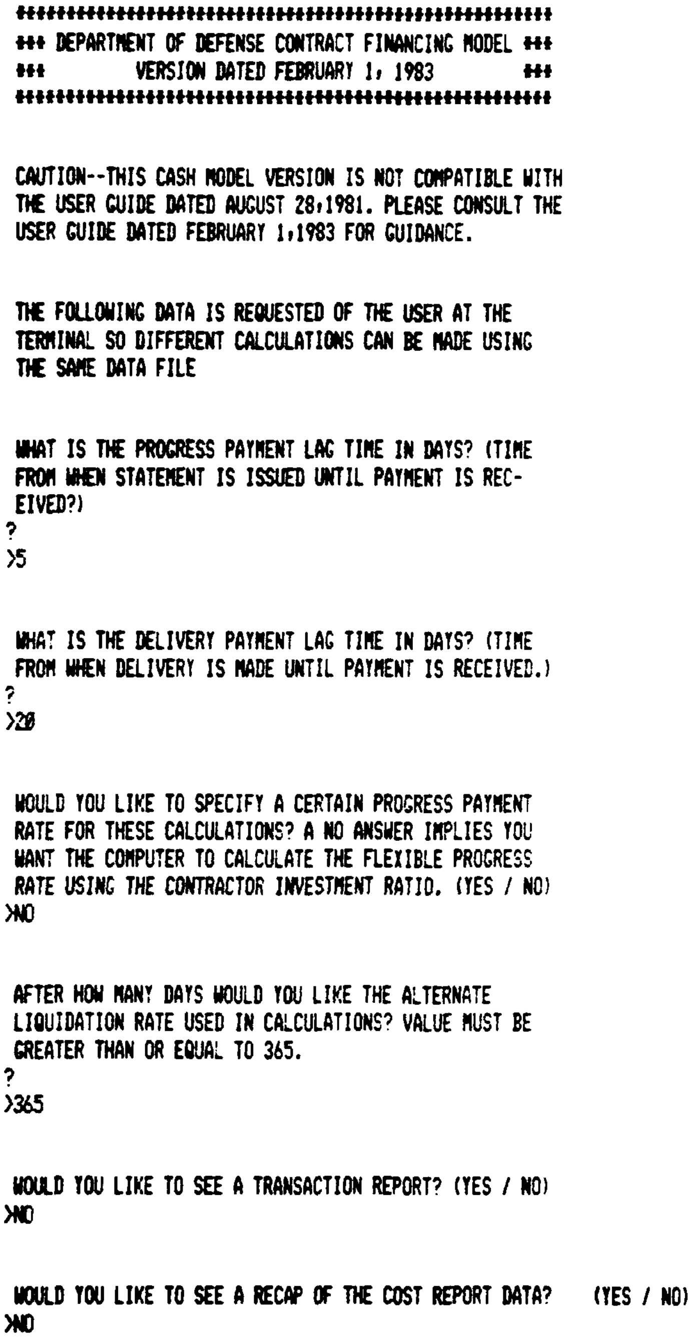
The last two questions are asking how much detail you wish to see for output. A recap of the cost report data reformats and prints the data found in the input file. Printing this data can be very useful in verifying that the system is reading the data you think you typed.

TRANSACTION REPORT

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.

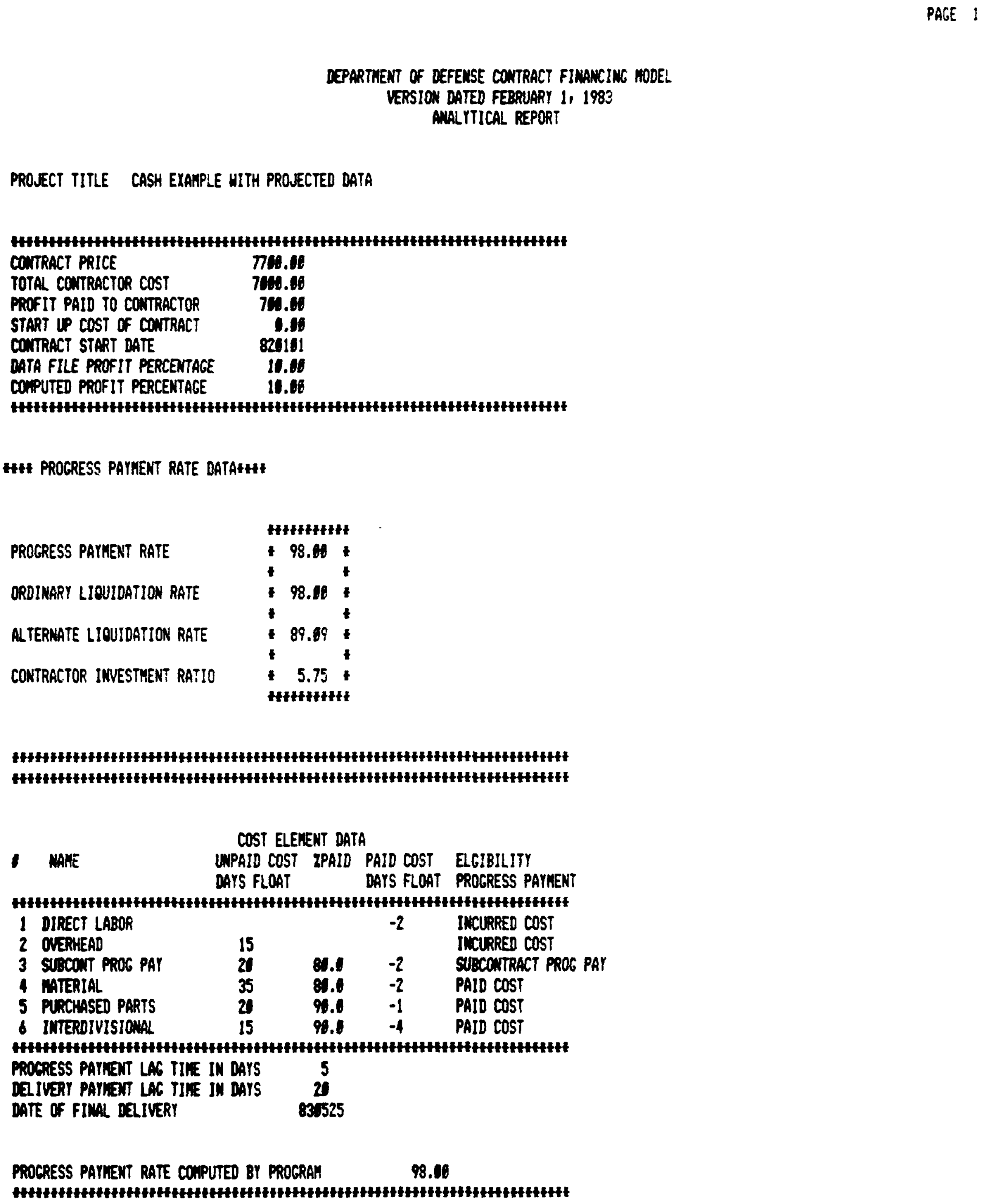
ANALYTICAL REPORT

Both of these reports are discussd 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 preceeding two questions.



```
######################################################
### DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL ###
###        VERSION DATED FEBRUARY 1, 1983          ###
######################################################


CAUTION--THIS 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.


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 REC-
EIVED?)
?
>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



```
                                                                                   PAGE 1

                                     DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL
                                           VERSION DATED FEBRUARY 1, 1983
                                                  ANALYTICAL REPORT


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  ELGIBILITY
                           DAYS FLOAT          DAYS FLOAT PROGRESS PAYMENT
##########################################################################
 1  DIRECT LABOR                                  -2      INCURRED COST
 2  OVERHEAD                  15                          INCURRED COST
 3  SUBCONT PROC PAY          20        80.0      -2      SUBCONTRACT PROC PAY
 4  MATERIAL                  35        80.0      -2      PAID COST
 5  PURCHASED PARTS           20        90.0      -1      PAID COST
 6  INTERDIVISIONAL           15        90.0      -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

```
                         DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL
                                VERSION DATED FEBRUARY 1, 1983
                                      ANALYTICAL REPORT


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             #  93.00 #
                                  #        #
ORDINARY LIQUIDATION RATE         #  98.00 #
                                  #        #
ALTERNATE LIQUIDATION RATE        #  89.09 #
                                  #        #
CONTRACTOR INVESTMENT RATIO       #   5.42#
                                  ##########


######################################################################
######################################################################


                              COST ELEMENT DATA
#   NAME                    UNPAID COST  %PAID  PAID COST  ELGIBILITY
                            DAYS FLOAT          DAYS FLOAT PROGRESS PAYMENT
######################################################################
 1  DIRECT LABOR                                   -2      INCURRED COST
 2  OVERHEAD                   15                          INCURRED COST
 3  SUBCONT PROC PAY           20        80.0      -2      SUBCONTRACT PROG PAY
 4  MATERIAL                   28        80.0      -2      PAID COST
 5  PURCHASED PARTS            20        90.0      -1      PAID COST
 6  INTERDIVISIONAL            15        90.0      -4      PAID COST
######################################################################
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           93.00
######################################################################
```

FIGURE 2

1. INTRODUCTION CONT'D

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.

```
82Ø1Ø1,Ø,6,22,.1
CASH EXAMPLE WIT  PROJECTED DATA
DIRECT LABOR,INCUR,-2,Ø,Ø                          SECTION 1
OVERHEAD,INCUR,15,Ø,Ø
SUBCONT PROG PAY,SUBCT,2Ø,.8,-2
MATERIAL,PAIDC,35,.8,-2
PURCHASED PARTS,PAIDC,2Ø,.9,-1
INTERDIVISIONAL,PAIDC,15,.9,-4                     SECTION 2
3,82Ø13Ø,Ø,Ø,Ø,1ØØ,Ø,Ø
3,82Ø228,Ø,Ø,Ø,1ØØ,Ø,Ø
3,82Ø33Ø,15Ø,5Ø,Ø,Ø,Ø,Ø
4,82Ø425,25Ø
3,82Ø43Ø,15Ø,5Ø,1ØØ,1ØØ,1ØØ,1ØØ
3,82Ø53Ø,15Ø,5Ø,1ØØ,1ØØ,1ØØ,1ØØ
3,82Ø63Ø,15Ø,5Ø,1ØØ,1ØØ,1ØØ,1ØØ
3,82Ø73Ø,15Ø,5Ø,1ØØ,1ØØ,1ØØ,1ØØ
3,82Ø83Ø,15Ø,5Ø,1ØØ,1ØØ,1ØØ,1ØØ
4,82Ø925,5ØØ
3,82Ø93Ø,15Ø,5Ø,-1ØØ,3ØØ,1ØØ,1ØØ                    SECTION 3
3,821Ø3Ø,15Ø,5Ø,-1ØØ,3ØØ,1ØØ,1ØØ
3,82113Ø,15Ø,5Ø,-1ØØ,3ØØ,1ØØ,1ØØ
4,821224,1ØØØ
3,821231,15Ø,5Ø,-1ØØ,3ØØ,1ØØ,Ø
4,83Ø125,1ØØØ
3,83Ø13Ø,15Ø,5Ø,-1ØØ,3ØØ,Ø,Ø
4,83Ø219,1ØØØ
3,83Ø228,15Ø,5Ø,Ø,2ØØ,Ø,Ø
3,83Ø33Ø,15Ø,5Ø,Ø,Ø,Ø,1ØØ
3,83Ø43Ø,15Ø,5Ø,Ø,Ø,Ø,Ø
4,83Ø525,395Ø
```

FIGURE 3

1. INTRODUCTION CONT'D

INPUT DATA FILE

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 CASHII 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.

1. INTRODUCTION CONT'D

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

2. CASHII BASICS CONT'D

GETTING STARTED

SELECTING A TERMINAL

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 vendors 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.

2. CASHII BASICS CONT'D

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
   ^    ^ ^ ^  ^
   |    | | |  |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 asociated 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.

2. CASHII BASICS CONT'D

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)
```

COST
ELEMENT
TYPES

INCUR means that expense 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:

FLOAT EXAMPLE

	PAYROLL PERIOD	
	FIRST	SECOND
DAY PAYROLL CLOSES	14	28
PLUS: PAYMENT LAG	8	8
PAYROLL PAYMENT DATE	22	36
LESS: DAY OF COST STATEMENT	28	28
FLOAT	-6	+8
TIMES: PAYROLL WEIGHT*	.5	.5
WEIGHTED FLOAT	-3	+4

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.

FLOAT EXAMPLE

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

2. CASHII BASIC CONT'D

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.

PAIDC and SUBCT COST ELEMENTS

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

```
SUBCONT PROG PAY, SUBCT, 20, .8,-2
          ^         ^    ^    ^ ^
          |         |    |    | |Float for paid costs.
          |         |    |    | Percent of cost paid in
          |         |    |      current cost accountng period.
          |         |    |Float for unpaid costs.
          |         |Type of cost element.
          |Cost element.
```

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 numer 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

FLOAT DAY DOLLARS

				TOTAL	DAY DOLLARS		
DATE		DATE	DAYS	DAYS FROM	(DAYS FROM START OF APR)		
INVOICE	INVOICE	CHECK	BANK	START OF	TIMES INVOICE AMOUNT)		
RECEIVED	AMOUNT	WRITTEN	FLOAT	APRIL	APRIL	MAY	
(1)	(2)	(3)	(4)	(5=3+4)	(6=2*5)		
APR 1	$ 100	APR 21	3	24	$ 2,400		paid costs
2	150	22	3	25	3,750		
3	200	23	3	26	5,200		
7	175	27	3	30	5,250		
8	125	28	3	31	3,875		
9	50	29	3	32	1,600		
21	25	MAY 11	3	44		$1,100	un-paid costs
23	50	13	3	46		2,300	
29	100	19	3	52		5,200	
30	25	20	3	53		1,325	
TOTAL	$1,000						

(APR TOTAL $800.

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	-20

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.

2. CASHII BASICS CONT'D

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 --

$$22075/800 = 27.5 = 28 \text{ days}$$

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:

$$9925/200 = 49.6 = 50 \text{ days}$$

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

TRANSACTIONS - 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 estmated). 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*(#of type 3's) + (#of 1's + #of 4's + #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.

2. CASHII BASICS CONT'D

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
^    ^    \_________/
|    |         ^
|    |         | cost element data
|    |date of transaction
|transaction type
```

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.

COMMAS
REQUIRED

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 preceeding 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
^   ^    ^
|   |    | dollar value of planned delivery
|   | date of planned delivery
| 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. CASHII BASICS CONT'D

CREATING THE DATA FILE

more transaction data

```
820101,0,6,25,.1
cash example with actual and projected data <----- new title just for
direct labor,incur,-2,0,0                           illustration
overhead,incur,15,0,0                               purposes
subcont prog pay,subct,20,.9,-2
material,paidc,28,.8,-2
purchased parts,paidc,20,.9,-1
interdivisional,paidc,15,.9,-4
3,820130,0,0,0,100,0,0
1,820204,0,.9 <----------
3,820228,0,0,0,100,0,0
1,820304,90,.9 <----------      actual progress payments received
3,820330,150,50,0,0,0,0
1,820404,270,.9 <----------
4,820425,250
3,820430,150,50,100,100,100,100
3,820530,150,50,100,100,100,100
3,820630,150,50,100,100,100,100
3,820730,150,50,100,100,100,100
3,820830,150,50,100,100,100,100
4,820925,500
3,820930,150,50,-100,300,100,100
3,821030,150,50,-100,300,100,100
3,821130,150,50,-100,300,100,100
4,821224,1000
3,830130,150,50,-100,300,0,0
4,830219,1000
3,830228,150,50,0,200,0,0
3,830330,150,50,0,0,0,100
3,830430,150,50,0,0,0,0
4,830525,3950
```

2. CASHII BASICS CONT'D

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.

2. CASHII BASICS CONT'D

CREATING THE DATA FILE

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,28,.8,-2
purchased parts,paidc,20,.9,-1
interdivisional,paidc,15,.9,-4
3,820130,0,0,0,100.0,0
1,820204,0,.9
3,820228,0,0,0,100,0,0
1,820304,90,.9
3,820330,150,50,0,0,0,0
1,820404,270,.9
5,820425,250,.9 <-----------  actual delivery payment received
3,820430,150,50,100,100,100,100
3,820530,150,50,100,100,100,100
```

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 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
^   ^     ^  ^
|   |     |  |
|   |     |  | Liquidation rate (90% here)
|   |     |  Value of delivery
|   |     Date delivery made
|   Transaction type
```

2. CASHII BASICS CONT'D

CREATING THE DATA FILE

LIQUIDATION RATES

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.

2. CASHII BASICS CONT'D

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?

2. CASHII BASICS CONT'D

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.

LIMIT ON LAG TIME

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?

REPORT WIDTH

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.

2. CASHII BASICS CONT'D

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.

3. CASHII OUTPUT

AVAILABLE REPORTS

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:

DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL
VERSION DATED FEBRUARY 1, 1983
TRANSACTION REPORT

CASH EXAMPLE WITH PROJECTED DATA

DATE / TRANSACT CODE	1 TOTAL INCURRED COST	2 FLOAT	3 PAID INCURRED COST	4 PROG PAY RECD	5 COST PAID AT DELIVERY	6 TOTAL COST REIMBURS (4+5)	7 PROFIT PAID AT DELIVERY	8 TOTAL REVENUE (6+7)	9 UNREIM PAID COST (3-6)	10 UNLIQ PROGRESS PAYMENTS
	(DOLLARS)	'DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)	(DOLLARS)
820130 / 3	100	20	80	0	0	0	0	0	80	0
820205 / 2	100	20	80	78	0	78	0	78	2	78
820228 / 3	200	20	180	78	0	78	0	78	102	78
820306 / 2	200	20	180	176	0	176	0	176	4	176
820330 / 3	400	50	350	176	0	176	0	176	174	176
820405 / 2	400	50	350	392	0	392	0	392	-42	392
820430 / 3	1000	110	870	392	0	392	0	392	478	392
820506 / 2	1000	110	870	923	0	923	0	923	-53	923
820515 / 4	1000	50	950	923	5	927	0	928	23	678
820530 / 3	1600	110	1490	923	5	927	0	928	563	678
820605 / 2	1600	110	1490	1513	5	1517	0	1518	-27	1268
820630 / 3	2200	110	2090	1513	5	1517	0	1518	573	1268
820706 / 2	2200	110	2090	2103	5	2107	0	2108	-17	1858
820730 / 3	2800	110	2690	2103	5	2107	0	2108	583	1858
820805 / 2	2800	110	2690	2693	5	2697	0	2698	-7	2448
820830 / 3	3400	110	3290	2693	5	2697	0	2698	593	2448
820905 / 2	3400	110	3290	3283	5	3287	0	3288	3	3038
820930 / 3	4000	110	3890	3283	5	3287	0	3288	603	3038
821006 / 2	4000	110	3890	3870	5	3874	0	3875	16	3625
821015 / 4	4000	50	3950	3870	14	3883	1	3885	67	3135
821030 / 3	4600	110	4490	3870	14	3883	1	3885	607	3135
821105 / 2	4600	110	4490	4456	14	4469	1	4471	21	3721
821130 / 3	5200	110	5090	4456	14	4469	1	4471	621	3721
821206 / 2	5200	110	5090	5042	14	5055	1	5057	35	4307
821231 / 3	5700	100	5600	5042	14	5055	1	5057	545	4307
830106 / 2	5700	100	5600	5539	14	5553	1	5554	47	4804
830113 / 4	5700	100	5600	5539	32	5571	159	5730	29	3980
830130 / 3	6100	90	6010	5539	32	5571	159	5730	439	3980
830205 / 2	6100	90	6010	5939	32	5971	159	6130	39	4380
830214 / 4	6100	40	6060	5939	50	5989	250	6239	71	3489
830228 / 3	6500	90	6410	5939	50	5989	250	6239	421	3489
830306 / 2	6500	90	6410	6331	50	6381	250	6631	29	3881
830311 / 4	6500	90	6410	6331	68	6399	341	6740	11	2990
830330 / 3	6800	60	6740	6331	68	6399	341	6740	341	2990
830405 / 2	6800	60	6740	6654	68	6722	341	7063	18	3313
830430 / 3	7000	50	6950	6654	68	6722	341	7063	228	3313
830506 / 2	7000	50	6950	6860	68	6928	341	7269	22	3519
830614 / 4	7000	0	7000	6860	140	7000	700	7700	0	0

FIGURE 4

3. CASHII OUTPUT CONT'D

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:

250 x .98 = 245 = liquidation amount

250/(1 + .10) = 227.3 = cost in this delivery

227.3 x .98 = 222.7 = cost previously reimbursed as progress payments

227.3 - 222.7 = 4.6 cost reimbursed at delivery (rounds to 5) displayed in column 5.

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

250 - 245 = 5 = net payment at delivery

5 - 4.6 = .4 = profit paid at delivery (rounds to 0) displayed in column 7.

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 dicussion 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:

```
                                         DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL
                                             VERSION DATED FEBRUARY 1, 1983
                                                   ANALYTICAL REPORT

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      #  5.42  #
                                 ##########


##########################################################################
##########################################################################


                              COST ELEMENT DATA
#    NAME                    UNPAID COST  %PAID  PAID COST   ELGIBILITY
                             DAYS FLOAT          DAYS FLOAT  PROGRESS PAYMENT
##########################################################################
 1  DIRECT LABOR                                    -2       INCURRED COST
 2  OVERHEAD                    15                           INCURRED COST
 3  SUBCONT PROG PAY            20        80.0      -2       SUBCONTRACT PROG PAY
 4  MATERIAL                    28        80.0      -2       PAID COST
 5  PURCHASED PARTS             20        90.0      -1       PAID COST
 6  INTERDIVISIONAL             15        90.0      -4       PAID COST
##########################################################################
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

3. CASHII OUTPUT CONT'D

ANALYTICAL REPORT

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 datafile. The difference between those 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%.

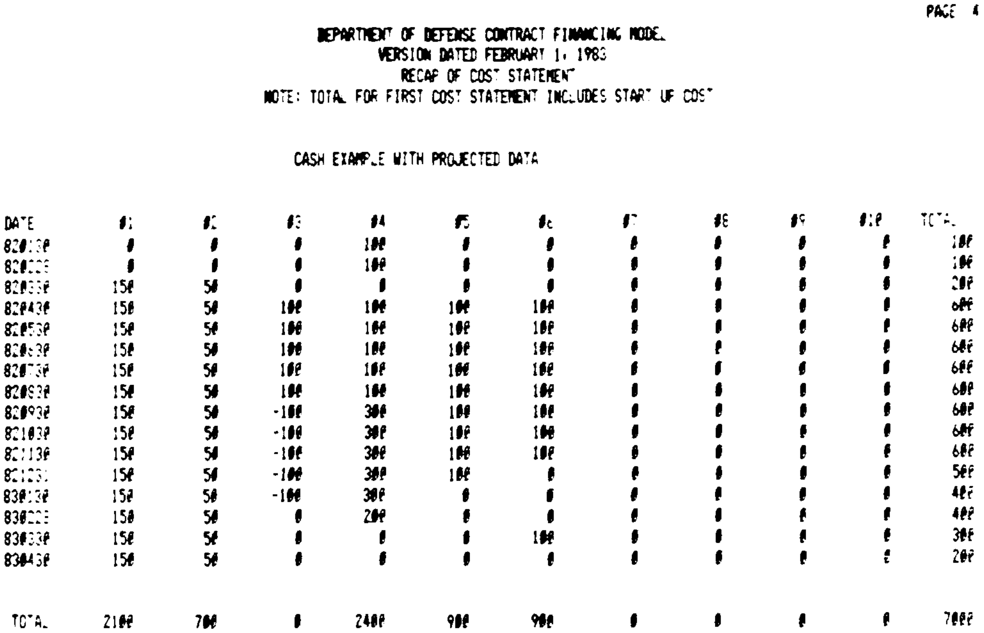
The next block of data on the Analytical Report is a formatted printout of your cost element data from the input data file. You should verify that these numbers are what you expected. If not, it will be necessary to change the data file.

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:



```
                                                                              PAGE 4
                        DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL
                               VERSION DATED FEBRUARY 1, 1983
                                  RECAP OF COST STATEMENT
                   NOTE: TOTAL FOR FIRST COST STATEMENT INCLUDES START UP COST


                           CASH EXAMPLE WITH PROJECTED DATA


DATE        #1      #2      #3      #4      #5      #6      #7      #8      #9     #1Ø   TOTAL
82Ø13Ø       Ø       Ø       Ø     1ØØ       Ø       Ø       Ø       Ø       Ø       Ø     1ØØ
82Ø228       Ø       Ø       Ø     1ØØ       Ø       Ø       Ø       Ø       Ø       Ø     1ØØ
82Ø33Ø     15Ø      5Ø       Ø       Ø       Ø       Ø       Ø       Ø       Ø       Ø     2ØØ
82Ø43Ø     15Ø      5Ø     1ØØ     1ØØ     1ØØ     1ØØ       Ø       Ø       Ø       Ø     6ØØ
82Ø53Ø     15Ø      5Ø     1ØØ     1ØØ     1ØØ     1ØØ       Ø       Ø       Ø       Ø     6ØØ
82Ø63Ø     15Ø      5Ø     1ØØ     1ØØ     1ØØ     1ØØ       Ø       Ø       Ø       Ø     6ØØ
82Ø73Ø     15Ø      5Ø     1ØØ     1ØØ     1ØØ     1ØØ       Ø       Ø       Ø       Ø     6ØØ
82Ø83Ø     15Ø      5Ø     1ØØ     1ØØ     1ØØ     1ØØ       Ø       Ø       Ø       Ø     6ØØ
82Ø93Ø     15Ø      5Ø    -1ØØ     3ØØ     1ØØ     1ØØ       Ø       Ø       Ø       Ø     6ØØ
821Ø3Ø     15Ø      5Ø    -1ØØ     3ØØ     1ØØ     1ØØ       Ø       Ø       Ø       Ø     6ØØ
82113Ø     15Ø      5Ø    -1ØØ     3ØØ     1ØØ     1ØØ       Ø       Ø       Ø       Ø     6ØØ
821231     15Ø      5Ø    -1ØØ     3ØØ     1ØØ       Ø       Ø       Ø       Ø       Ø     5ØØ
83Ø13Ø     15Ø      5Ø    -1ØØ     3ØØ       Ø       Ø       Ø       Ø       Ø       Ø     4ØØ
83Ø228     15Ø      5Ø       Ø     2ØØ       Ø       Ø       Ø       Ø       Ø       Ø     4ØØ
83Ø33Ø     15Ø      5Ø       Ø       Ø       Ø     1ØØ       Ø       Ø       Ø       Ø     3ØØ
83Ø43Ø     15Ø      5Ø       Ø       Ø       Ø       Ø       Ø       Ø       Ø       Ø     2ØØ


 TOTAL    21ØØ     7ØØ       Ø    24ØØ     9ØØ     9ØØ       Ø       Ø       Ø       Ø    7ØØØ
```

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:

```
820101,0,6,25,.1
  ^    ^ ^ ^   ^
  |    | | |   |
  |    | | |   |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
         ^
         |
         |  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,-2
   ^       ^    ^  ^  ^
   |       |    |  |  |
   |       |    |  |  |
   |       |    |  |  |
   |       |    |  |  | 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
```

4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

Transaction Lines Cont'd

Type 4 (Planned delivery)

```
4,830201,1000
    ^      ^
    |      |
    |      |Value of delivery (max of 6 digits)
    |  Planned date of delivery
```

Type 5 (Actual delivery)

```
5,820425,250,.9
    ^      ^    ^
    |      |    |
    |      |    |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. CASHII 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 CashII 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.

4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

FLEXIBLE PROGRESS PAYMENT RATE - The flexible progress payment rate is the progres 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-proces inventory by the contractor in excess of 7% or less than 3% the progress payment rate shall be redetermined 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.

4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

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.

4. CASHII, TERMS, DEFINITIONS AND SYNTAX CONT'D

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.

4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

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.

4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

ERROR MESSAGES

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.

4. CASH TERMS, DEFINITIONS AND SYNTAX CONT'D

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.

4. CASHII TERMS, DEFINITIONS AND SYNTAX CONT'D

ERROR MESSAGES CONT'D

The profit rate computed from the data file (XX.XX) does not match the profit rate (XX.XX) 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 ajd 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

```
HH
U#=MC0360001,
PASSWORD
[redacted]
```

IDENTIFY TERMINAL TYPE AND SPEED

ENTER YOUR USER NUMBER

ENTER YOUR PASSWORD

```
READY
NEW SAMP1
SYSTEM- F77
```

NEW IS THE COMMAND NECESSARY TO ENTER NEW DATA. SAMP1 IS THE FILE NAME.

F77 MEANS FORTRAN

```
READY
100 820101,0,6,22,.1
110 CASH EXAMPLE WITH PROJECTED DATA
120 DIRECT LABOR,INCURR←,-2,0,0
130 OVERHEAD,INCUR,15,0,0
140 SUBCONT PROG PAY,20,.8,-2
150 MATERIAL,PAIDC,28,.8,-2
160 PURCHASED PARTS,PAIDC,20,.9,-1
170 INTERDIVISIONAL,PAIDC,15,.9,-4
180 3,820130,0,0,0,100,0,0
190 3,820225,0,0,0,100,0,0
200 3,820330,150,50,0,0,0,0
210 1,8204DELETED
210 4,820425,250
220 3,820430,150,50,100,100,100,100
230 3,820530,150,50,100,100,100,100
EDI DES
```

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

```
READY
REP
```

SAVE (REPLACE) THE DATA

```
READY
RUN CASHII
```

EXECUTE THE PROGRAM

```
CASHII     TIME   DATE
```

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

```
################################################
### DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL ###
###         VERSION DATED FEBRUARY 1, 1983         ###
################################################

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

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

```
INPUT FILE NAME?SAMP1
```

ENTER THE NAME OF YOUR DATA FILE

SAMPLE USING BCS

```
    to the bcs network
    access port is bcs 0ed

select desired service:  CTS

mainstream-cts online

CP>LOGON AFS00001
ENTER PASSWORD:
CP>[redacted]
```

ENTER YOUR USER NUMBER

ENTER YOUR PASSWORD

```
LOGMSG - 09:40:29 EST WEDNESDAY 01/26/83
** PLEASE TYPE NEWS MEMO 428 - RAMIS II - NEW DOCUMENTS AND INCREASED *
**                             MACHINE SIZE REQUIREMENTS **
** PLEASE TYPE NEWS MEMO 417 - INTELLECT AVAILABLE **
** PLEASE TYPE NEWS MEMO 435 - PLOTTING SERVICES AVAILABLE **
** PLEASE TYPE NEWS MEMO 436 - PHONE NUMBER ADDED - SIOUX FALLS, SD **
** PLEASE TYPE NEWS MEMO 421 - TELL-A-GRAF VERSION 4.1 AVAILABLE **
** PLEASE TYPE NEWS MEMO 437 - PHONE NUMBER OUTAGE - PHOENIX,AZ. 1/28 **
** PLEASE TYPE NEWS MEMO 434 - IMPROVED LOCAL DIAL FOR RJE SERVICES **
LOGON AT 07:00:09 EST FRIDAY 01/28/83 LINE 6D6 (2-1-0ED)
CMS/SP REL 1 01/25/83 V008
D (292) R/O
ENTER PROJECT ID:
>I00
Z (199) R/O
R;

R;

C>EDIT SAMP1 DATA  <-
E>I                <-
```

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

```
I>820101,0,6,22,.1
I>CASH EXAMPLE WITH PROJECTED DATA
I>DIRECT LABOR,INCUR,-2,0,0
I>OVERHEAD,INCUR,15,0,0
I>SUBCONT PROG PAY,SUBCT,20,.8,-2
I>MATERIAL,PAIDC,28,.8,-2
I>PURCHASED PARTS,PAIDC,20,.9,-1
I>INTERDIVISIONAL,PAIDC,15,.9,-4
I>3,820130,0,0,0,100,0,0
I>3,820228,0,0,0,100,0,0
I>3,820330,150,50,0,0,0M@,0  <-
I>1,820404,24,820425,250     <-
I>3,820430,150,50,100,100,100,100
I>3,820530,150,50,100,100,100,100
I>3,820630,150,50,100,100,100,100
I>      <-
E>FILE  <-
R;

C>CASHII  <-
ENTER INPUT FILENAME
>SAMP1  <-
```

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 FURTHER EXPLAINS THE COMMANDS.

APPENDIX 2

REFERENCE MATERIAL FOR THIS DOCUMENT IS CONTAINED IN DEFENSE ACQUISITION REGULATION, APPENDIX E, PART 5.

APPENDIX 3

```
      INTEGER STDATE,NMCSTL,NMTRAN,ACFRG,LSTDEL,
     &FRGLAG,DELLAG,FRGFLG,ALTLDY,FLOATD,ALOFLG,SUBCNT,
     &UFDCNT,STDAY,FFCNT,CNTST,TWOFLG,CSTDAY(170),DATKEY
      REAL TRNSAC(325,20),FRFRAT,FROFIT,ELMENT(10,5),OUTFUT(325,10),
     &FRGRTE,RATIO,TOTFYD,TODAYS,STCOST
      REAL CSTFLT(1000,2),UFDFLT(1000,2),UPDSUB(1000,2)
      REAL CCOST,TOFROF,TOTCOS,SUMCST,TOTCHG
      CHARACTER*20 COSTEL(10,2)
      CHARACTER*50 TITLE
      CHARACTER*3 TRNFLG,CSTFLG,FLGIT,FLXFLG,ERROR,DATCHK
      LOGICAL*1 CC
      DATA CC / Z0C /
C     ****************************************************************
C             SECTION ABOVE ESTABLISHES ATTRIBUTES OF VARIABLES FASSED
C             BETWEEN MAIN FROGRAM AND SUBROUTINES.
C             SECTION IMMEDIATELY BELOW WRITES MODEL HEADINGS.
C      ************************************************************
10    FORMAT(1H1)
20    FORMAT(' ***************************************************** ')
30    FORMAT(' *** DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL *** ')
40    FORMAT(' ***             VERSION DATED FEBRUARY 1, 1983        ***')
  50  FORMAT(' CAUTION--THIS CASH MODEL VERSION IS NOT COMPATIBLE WITH ')
  51  FORMAT(' THE USER GUIDE DATED AUGUST 28,1981. FLEASE CONSULT THE ')
  52  FORMAT(' USER GUIDE DATED FEBRUARY 1,1983 FOR GUIDANCE.')
60    FORMAT( / )
70    FORMAT( / )
      WRITE(6,FMT='(A1)') CC
      WRITE(6,20)
      WRITE(6,30)
      WRITE(6,40)
      WRITE(6,20)
      WRITE(6,70)
      WRITE(6,50)
      WRITE(6,51)
      WRITE(6,52)
      WRITE(6,70)
      RERUN=0
100   ACFRG=0
C      ************************************************************
C             CALLS SUBROUTINE TO READ INPUT FILE.
C      ************************************************************
      CALL FILEIN(STDATE,STCOST,NMCSTL,NMTRAN,FRFRAT,TITLE,COSTEL,
     &ELMENT,TRNSAC,ACFRG,STDAY,TRNCNT,ERROR)
       IF (ERROR.EQ.'YES') THEN
         WRITE(6,200)
         GO TO 2000
      END IF
C      ************************************************************
C             CALLS SUBROUTINE TO EDIT INFUT DATA.
C      ************************************************************
      DATCHK='NO'
      DATKEY=0
      CALL EDIT(TRNSAC,ELMENT,NMCSTL,STDATE,STCOST,FRFRAT,NMTRAN
     &,COSTEL,ERROR,CCOST,ACFRG,TOTFYD,TOFROF,PROFIT,LSTDEL,TODAYS,
     &TOTCOS)
       IF (ERROR.EQ.'YES') THEN
200      FORMAT(' THE FROGRAM WAS STOFFED BECAUSE OF AN INFUT FILE ERROR.
     &')
```

```
        WRITE(6,200)                                                    CAS00590
  5     WRITE(6,70)                                                     CAS00600
      ************************************************************     CAS00610
           ROUTINE ALLOWS USER TO CHECK INPUT DATA WHEN EDIT ROUTINE   CAS00620
           DETECTS ERRORS.                                              CAS00630
      ************************************************************     CAS00640
  0   FORMAT(' WOULD YOU LIKE TO CHECK YOUR INPUT DATA? (YES OR NO)')  CAS00650
      WRITE(6,210)                                                      CAS00660
      READ(5,FMT='(A3)') DATCHK                                         CAS00670
      IF(DATCHK(1:1).NE.'Y'.AND.DATCHK(1:1).NE.'N') THEN                CAS00680
        GO TO 205                                                       CAS00690
      END IF                                                            CAS00700
      IF(DATCHK.EQ.'YES') THEN                                          CAS00710
  0   FORMAT(' DO YOU WANT TO CHECK: 1-COST INPUT DATA; 2-DELIVERY INPUTCAS00720
     & DATA; OR 3-BOTH?')                                               CAS00730
        WRITE(6,70)                                                     CAS00740
  5     WRITE(6,220)                                                    CAS00750
        READ(5,FMT=*) DATKEY                                            CAS00760
          IF(DATKEY.LT.1.OR.DATKEY.GT.3) THEN                           CAS00770
            WRITE(6,70)                                                 CAS00780
            GO TO 225                                                   CAS00790
          END IF                                                        CAS00800
      END IF                                                            CAS00810
      IF(DATKEY.GT.0) THEN                                              CAS00820
        GO TO 305                                                       CAS00830
      ELSE                                                              CAS00840
        GO TO 2000                                                      CAS00850
      END IF                                                            CAS00860
        GO TO 2000                                                      CAS00870
      END IF                                                            CAS00880
      ************************************************************     CAS00890
           CALLS SUBROUTINE TO GET USER INPUTS FROM TERMINAL.           CAS00900
      ************************************************************     CAS00910
      CALL DATAIN(PRGLAG,DELLAG,FLXFLG,PRGRTE,PRGFLG,                   CAS00920
     &ALTLDY,TRNFLG,CSTFLG,RERUN,TWOFLG)                                CAS00930
      ************************************************************     CAS00940
           CALLS SUBROUTINE TO CREATE PLANNED PROGRESS PAYMENTS AND     CAS00950
           ADD DELIVERY LAG TO DELIVERY DATES.                          CAS00960
      ************************************************************     CAS00970
      CALL CREPAY(TRNSAC,NMTRAN,ACPRG,CCOST,PRGLAG,PRGFLG,STDAY,DELLAG, CAS00980
     &ERROR)                                                            CAS00990
      IF (ERROR.EQ.'YES') THEN                                          CAS01000
        GO TO 205                                                       CAS01010
      END IF                                                            CAS01020
      ************************************************************     CAS01030
           INITIALIZES (OR RESETS) VARIABLES TO ZERO.                   CAS01040
           INITIALIZES FIRST TIME THROUGH; RESETS AS FLEX RATE STEPS    CAS01050
           DOWN AND WHEN ALTERNATIVES ARE RUN WITH THE SAME DATA        CAS01060
           FILE.                                                        CAS01070
      ************************************************************     CAS01080
  230 DO 250 I=1,NMTRAN                                                 CAS01090
        DO 240 J=1,10                                                   CAS01100
          OUTPUT(I,J)=0                                                 CAS01110
  0     CONTINUE                                                        CAS01120
  0   CONTINUE                                                          CAS01130
      DO 270 I=1,1000                                                   CAS01140
        DO 260 J=1,2                                                    CAS01150
          CSTFLT(I,J)=0                                                 CAS01160
          UFDFLT(I,J)=0                                                 CAS01170
          UFDSUB(I,J)=0                                                 CAS01180
          CONTINUE
       CONTINUE
       DO 290 I =1,300
            DO 280 J=14,17
              TRNSAC(I,J)=0
80          CONTINUE
90     CONTINUE
       DO 295 I=1,170
         CSTDAY(I)=0
95     CONTINUE
       LFLG=0
       FPCNT=0
       CNTST=0
       ALTPRF=0
       COSDEL=0
       FDCOST=0
       INFLOW=0
       COUNT3=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 ACTPPY(TRNSAC,I,OUTPUT,FLOATD,CSTFLT,PRPRAT,ALTLDY,STCOST
      &,PPCNT)
         END IF
         IF (TRNSAC(I,1).EQ.2) THEN
           CALL PRGPAY(TRNSAC,I,PRGRTE,CSTFLT,FLOATD,OUTPUT,TOTCHG,
      &SUMCST,TOTPYD,ALTLDY,PRPRAT,STCOST,PPCNT,CSTDAY,TWOFLG)
         END IF
         IF (TRNSAC(I,1).EQ.3) THEN
           CALL COSTST(TRNSAC,I,NMCSTL,ELMENT,FLOATD,CSTFLT,OUTPUT,
      &SUBCNT,UPDCNT,UPDFLT,UPDSUB,PRGRTE,STCOST,CNTST,CSTDAY,
      &TWOFLG)
         END IF
         IF (TRNSAC(I,1).EQ.4) THEN
           CALL PLNDEL(TRNSAC,I,ALTLDY,PRPRAT,ALOFLG,CSTFLT,FLOATD,
      &TOTCHG,SUMCST,OUTPUT,PRGRTE,LSTDEL,DELLAG,STCOST,TODAYS,STDAY,
      &ALTPRF,LFLG)
         END IF
         IF (TRNSAC(I,1).EQ.5) THEN
           CALL ACTDEL(TRNSAC,I,PRPRAT,ALTLDY,ALOFLG,CSTFLT,FLOATD,
      &TOTCHG,SUMCST,OUTPUT,LSTDEL,DELLAG,STCOST,TODAYS,STDAY,ALTPRF)
         END IF
0      CONTINUE
        ***********************************************************
            CALLS SUBROUTINE TO COMPUTE CONTRACTOR INVESTMENT RATIO.
        ***********************************************************
       CALL INVEST(TRNSAC,NMTRAN,TODAYS,CSTFLT,FLOATD,RATIO,OUTPUT,DELLAG
      &,STCOST,PRPRAT)
       IF (FLXFLG(1:1).EQ.'N') THEN
         IF (RATIO.LT.(.05).AND.PRGRTE.GT.(.90)) THEN
           PRGRTE=PRGRTE-0.01
           GO TO 230
        END IF
      END IF
       ****************************************************************
             CALLS SUBROUTINE TO FRINT REFORTS.
       ****************************************************************
    5 CALL FNTOUT(IFNFLG,CSTFLG,FRGFLG,TRNSAC,ELMENT,FLXFLG,NMTRAN,
     &NMCSTL,STDATE,COSTEL,FRFRAT,STCOST,TOTFYD,TOTCOS,OUTFUT,TOFROF,
     &FROFIT,FRGRTE,RATIO,FRGLAG,DELLAG,LSTDEL,TITLE,DATKEY,LFLG)
       ****************************************************************
             ALLOWS USER TO CHANGE TERMINAL INFUTS AND RE-RUN USING
             THE SAME DATA FILE.
       ****************************************************************
  310 FORMAT(' WOULD YOU LIKE TO RUN THE MODEL AGAIN USING THE SAME )
  311 FORMAT(' DATA FILE BUT CHANGING THE VARIABLES AT THE TERMINAL' )
  312 FORMAT(' INFUT? (YES / NO) )
  320 WRITE(6,FMT='(A1)') CC
      WRITE(6,310)
      WRITE(6,311)
      WRITE(6,312)
      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.EO.'YES') THEN
        RERUN=1
        GO TO 100
      END IF
[illegible] CONTINUE
      STOF
      END
       **********        END OF MAIN FROGRAM       **********



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

      SUBROUTINE FILEIN(STDATE,STCOST,NMCSTL,NMTRAN,FRFRAT,TITLE,COSTEL,
     &ELMENT,TRNSAC,ACFRG,STDAY,TRNCNT,ERROR)
      REAL FRFRAT,ELMENT(10,5),TRNSAC(325,20),STCOST
      INTEGER A,TRNCNT,STDATE,NMCSTL,NMTRAN,ACFRG,STDAY,IFOS,IERR
      CHARACTER*20 COSTEL(10,2)
      CHARACTER*50 TITLE
      CHARACTER*80 HOLD,SAVE
      CHARACTER*8 SHOLD
      CHARACTER*3 ERROR
      DIMENSION KK1(5)
      DATA KK1/4,3,8,3,4/
      ERROR='NO'
      OFEN(UNIT=1,STATUS='OLD',ACCESS='SEOUENTIAL')
       ****************************************************************
             READS FIRST LINE OF INFUT FILE.
       ****************************************************************
      READ(UNIT=1,FMT=*) STDATE,STCOST,NMCSTL,NMTRAN,FRFRAT
      READ(UNIT=1,FMT=*) TITLE
      READ(UNIT=1,FMT='(A80)')HOLD
      SAVE=HOLD
      CALL GETSTR(HOLD,' ',SHOLD,IFOS)
       IF(IFOS.LT.2.OR.IFOS.GT.9) GOTO 135
      STDATE=INTSTV(SHOLD,IERR)
       IF(IERR.EO.1) GOTO 135
      CALL GETSTR(HOLD,',',SHOLD,IPOS)
       IF(IPOS.LT.2.OR.IPOS.GT.9) GOTO 135
      SICOST=REALSV(SHOLD,IERR)
       IF(IERR.EQ.1) GOTO 135
      CALL GETSTR(HOLD,',',SHOLD,IPOS)
       IF(IPOS.LT.2.OR.IPOS.GT.9) GOTO 135
      NMCSTL=INTSTV(SHOLD,IERR)
       IF(IERR.EQ.1) GOTO 135
      CALL GETSTR(HOLD,',',SHOLD,IPOS)
       IF(IPOS.LT.2.OR.IPOS.GT.9) GOTO 135
      NMTRAN=INTSTV(SHOLD,IERR)
       IF(IERR.EQ.1) GOTO 135
      CALL GETSTR(HOLD,',',SHOLD,IPOS)
       IF(IPOS.GT.9) GOTO 135
      PRFRAT=REALSV(SHOLD,IERR)
       IF(IERR.EQ.1) GOTO 135
      GOTO 140
135   WRITE(6,390)SAVE
      ERROR='YES'
      GOTO 600
140   CONTINUE
C      ************************************************************
C           READS TITLE FROM INPUT FILE.
C      ************************************************************
      READ(UNIT=1,FMT='(A80)')HOLD
      TITLE=HOLD
      DO 120 I=1,NMCSTL
        DO 110 J = 1,5
          ELMENT(I,J)=0
110     CONTINUE
120   CONTINUE
C      ************************************************************
C           READS COST ELEMENT DATA FROM INPUT FILE.
C      ************************************************************
      DO 190 I=1,NMCSTL
        READ(UNIT=1,FMT=*) COSTEL(I,1),COSTEL(I,2),ELMENT(I,2),ELMENT(I,
C    &3),ELMENT(I,4)
      READ(UNIT=1,FMT='(A80)')HOLD
      SAVE=HOLD
      CALL GETSTR(HOLD,',',COSTEL(I,1),IPOS)
       IF(IPOS.LT.2) GOTO 185
      CALL GETSTR(HOLD,',',SHOLD,IPOS)
       IF(IPOS.LT.2.OR.IPOS.GT.9) GOTO 185
      COSTEL(I,2)=SHOLD
      CALL GETSTR(HOLD,',',SHOLD,IPOS)
       IF(IPOS.LT.2.OR.IPOS.GT.9) GOTO 185
      ELMENT(I,2)=REALSV(SHOLD,IERR)
       IF(IERR.EQ.1) GOTO 185
      CALL GETSTR(HOLD,',',SHOLD,IPOS)
       IF(IPOS.LT.2.OR.IPOS.GT.9) GOTO 185
      ELMENT(I,3)=REALSV(SHOLD,IERR)
       IF(IERR.EQ.1) GOTO 185
      CALL GETSTR(HOLD,',',SHOLD,IPOS)
       IF(IPOS.GT.9) GOTO 185
      ELMENT(I,4)=REALSV(SHOLD,IERR)
       IF(IERR.EQ.1) GOTO 185
        IF (COSTEL(I,2).EQ.'INCUR') THEN
          ELMENT(I,1)=1
        END IF
        IF(COSTEL(I,2).EQ.'PAIDC') THEN
```

```
          ELMENT(I,1)=2
        END IF
        IF(COSTEL(I,2).EQ.'SUBCT') THEN
          ELMENT(I,1)=3
        END IF
      GO TO 190
185   WRITE(6,390) SAVE
      ERROR='YES'
190   CONTINUE
C ***** NEXT LINE ESTABLISHES RELATIVE REFERENCE POINT FOR CONTRACT
C       START DATE.  ALL TRANSACTIONS WILL USE THIS POINT
C       TO DETERMINE DAYS AFTER CONTRACT AWARD.
      STDAY= KDAY(STDATE)
      KK1(3)= NMCSTL + 2
      TRNCNT=NMTRAN*2
      DO 270 I=1,TRNCNT
        DO 260 J=1,20
          TRNSAC(I,J)=0
260     CONTINUE
270   CONTINUE
C      **************************************************************
C            READS TRANSACTION DATA FROM INPUT FILE.
C      **************************************************************
      DO 330 I=1,NMTRAN+1
        READ(UNIT=1,FMT='(A80)',END=500)HOLD
        SAVE=HOLD
         CALL GETSTR(HOLD,',',SHOLD,IPOS)
          IF(IPOS.NE.2) GOTO 290
         A=INTSTV(SHOLD,IERR)
         IF(IERR.EQ.1) GOTO 290
         IF(A.LT.1.OR.A.GT.5) GOTO 290
         TRNSAC(I,1) = A
      DO 280 J=2,KK1(A)
        CALL GETSTR(HOLD,',',SHOLD,IPOS)
         IF(IPOS.EQ.0.AND.(J.NE.KK1(A))) GOTO 290
         IF(IPOS.GT.7.OR.(IPOS.LT.2.AND.J.NE.KK1(A))) GOTO 290
       TRNSAC(I,J)=REALSV(SHOLD,IERR)
        IF(IERR.EQ.1) GOTO 290
       IF(J.EQ.2) THEN
         K=INT(TRNSAC(I,2))
         TRNSAC(I,13) = (KDAY(K)-STDAY)
       ENDIF
       GOTO 280
290    WRITE(6,390)SAVE
       ERROR='YES'
       GOTO 330
390    FORMAT(' ERROR IN FOLLOWING INPUT LINE',/,A80)
280   CONTINUE
330   CONTINUE
C      **************************************************************
C            CHECKS TO SEE IF PROGRAM READ THE NUMBER OF TRANSACTIONS
C            STATED IN THE FIRST LINE OF THE INPUT FILE.
C      **************************************************************
500   IF(I.LE.NMTRAN) THEN
      WRITE(6,400) NMTRAN,I-1
      ERROR='YES'
      END IF
400   FORMAT(' THE DATA FILE INDICATED ',I3,' TRANSACTIONS WERE INCLUDED
     * IN THE FILE.  ONLY ',I3,/,' TRANSACTIONS ARE ACTUALLY INCLUDED.  P
     *LEASE CHECK INPUT FILE.')
```

```
      CLOSE(UNIT=1)
      RETURN
      END
C      **********      END OF SUBROUTINE FILEIN      **********
C
C
C
C      **********      START OF SUBROUTINE EDIT      **********
C
C      ******      EACH EDIT CHECK IS UNDERLINED WITH *'S.      ******
      SUBROUTINE EDIT(TRNSAC,ELMENT,NMCSTL,STDATE,STCOST,PRFRAT,NMTRAN,
     &COSTEL,ERROR,CCOST,ACPRG,TOTPYD,TOPROF,PROFIT,LSTDEL,TODAYS,
     &TOTCOS)
      REAL TRNSAC(325,20),ELMENT(10,5),PRFRAT,CCOST,TOPROF,PROFI
     &T,TOTCOS,TOTPYD,STCOST,TODAYS
      INTEGER NMCSTL,STDATE,NMTRAN,LSTDEL,DAPMO(12)
      CHARACTER*20  COSTEL(10,2)
      CHARACTER*3 ERROR
      INTEGER CACPRG,ACPRG
      REAL COSTPD,P,PL,TH
      DATA DAPMO/31,28,31,30,31,30,31,31,30,31,30,31/
      ERROR= NO
      TOTPYD=0
      CCOST=0
      CACPRG=0
      TOTCOS=STCOST
      IF (STCOST.LT.0) THEN
C     ************************************************
10      FORMAT(  THE START-UP COST CANNOT BE A NEGATIVE NUMBER.   / )
        WRITE(6,10)
        ERROR= YES
        WRITE(6,400)
      END IF
      IF (NMCSTL.GT.10.OR.NMCSTL.LT.1) THEN
C     ************************************************
20      FORMAT(  THE NUMBER OF COST ELEMENTS MUST BE BETWEEN 1 AND 10
     &.   / )
        WRITE(6,20)
        ERROR= YES
        WRITE(6,400)
      END IF
      IF (PRFRAT.LT.0.OR.PRFRAT.GT.1)THEN
C     ************************************************
30      FORMAT(  THE PROFIT RATE MUST BE BETWEEN 0 AND 1.   / )
        WRITE(6,30)
        ERROR= YES
        WRITE(6,400)
      END IF
      DO 800 I=1,NMTRAN
        IYRPC=INT(TRNSAC(I,2))/10000
        ISPLTY=IYRPC * 10000
        ISPLTM=INT(TRNSAC(I,2)) - ISPLTY
        IMOPC=ISPLTM/100
        ISPLTD=IMOPC * 100
        IDAPC=ISPLTM - ISPLTD
        IF(IMOPC.LT.1.OR.IMOPC.GT.12) THEN
C     ************************************************
32         FORMAT(  THE TRANSACTION DATE ,F8.0,  HAS AN ILLEGAL MONTH  ,
     &I2,  . )
           WRITE(6,32) TRNSAC(I,2), IMOPC
```

```
          ERROR='YES'
          WRITE(6,400)
          GO TO 36
        END IF
        IF (IDAPC.GT.DAPMO(IMOPC)) THEN
C*************************************************
34        FORMAT(' THE TRANSACTION DATE ',F8.0,' HAS AN ILLEGAL DAY',I3,
     & '.')
          WRITE(6,34) TRNSAC(I,2), IDAPC
          ERROR='YES'
          WRITE(6,400)
        END IF
C     ************************************************************
C            CHECKS COST STATEMENT DATA.
C     ************************************************************
36      IF (TRNSAC(I,1).EQ.3) THEN
          DO 700 J=1,NMCSTL
            IF (TRNSAC(I,J+2).GT.999999.OR.TRNSAC(I,J+2).LT.-999999) THE
     &N
C*************************************************
40              FORMAT(' COST AND DELIVERY TRANSACTIONS MUST NOT HAVE', /
     &  VALUES LARGER THAN SIX DIGITS. THE COST STATEMENT DATED ',F8.0,
     &/  ' HAS A VALUE THAT IS TOO LARGE.')
              WRITE(6,40) TRNSAC(I,2)
              ERROR='YES'
              WRITE(6,400)
            END IF
C ***** NEXT LINE SUMS COST DATA TO DETERMINE TOTAL COST.
            TOTCOS=TOTCOS+TRNSAC(I,J+2)
            IF (TRNSAC(I,J+2).LT.0.AND.ELMENT(J,1).NE.3) THEN
C*************************************************
50              FORMAT(' NEGATIVE COST IS PERMITTED ONLY FOR SUBCONTRACOR
     & ',/,' TYPE PROGRESS PAYMENTS. THE COST STATEMENT DATED ',F8.0,' HA
     &S ',/,' A NEGATIVE VALUE AND IS NOT A SUBCONTRACTOR COST ELEMENT.')
              WRITE(6,50) TRNSAC(I,2)
              ERROR='YES
              WRITE(6,400)
            END IF
C ***** NEXT LINE SUMS COST DATA BY COST ELEMENT.
            ELMENT(J,5)=ELMENT(J,5)+TRNSAC(I,J+2)
700       CONTINUE
        END IF
C     ************************************************************
C            CHECKS ACTUAL PROGRESS PAYMENT DATA.
C     ************************************************************
        IF (TRNSAC(I,1).EQ.1) THEN
        ACPPG=I
          IF (TRNSAC(I,3).LT.0) THEN
C*************************************************
60            FORMAT(' AN ACTUAL PROGRESS PAYMENT MUST HAVE A VALUE ', / ,
     & GREATER THAN 0. THE PROGRESS PAYMENT DATED',F8.0,' IS LESS THAN
     & '0.')
              WRITE(6,60) TRNSAC(I,2)
              ERROR='YES
              WRITE(6,400)
          END IF
          IF (TRNSAC(I,4).LT.0.OR.TRNSAC(I,4).GT.1) THEN
C*************************************************
70            FORMAT(' A PROGRESS PAYMENT RATE MUST BE BETWEEN 0 AND 1.
     & ',/,' THE TRANSACTION DATED ',F8.0,' VIOLATES THIS RULE.')
```

```
            WRITE(6,70) TRNSAC(I,2)
            ERROR= YES
            WRITE(6,400)
          END IF
        END IF
        IF (TRNSAC(I,1).EQ.2) THEN
     ****************************************************
          FORMAT(' DO NOT INPUT ANY PLANNED PROGRESS PAYMENTS INTO THE ,
     &    ,  DATA FILE. THE TRANSACTION DATED  ,F8.0,' MUST BE REMOVED.')
          WRITE(6,75) TRNSAC(I,2)
          ERROR= YES
          WRITE(6,400)
        END IF
        IF (I.GT.1.AND.TRNSAC(I,2).LT.TRNSAC(I-1,2)) THEN
     ****************************************************
30        FORMAT(  THE DATA IS NOT IN CHRONOLOGICAL ORDER.  CHECK  TRANS
     &ACTIONS , / ,  DATED  ,F8.0,  AND  ,F8.0,  TO CORRECT. )
          WRITE(6,80) TRNSAC(I,2), TRNSAC(I-1,2)
          ERROR= YES
          WRITE(6,400)
        END IF
        IF (TRNSAC(I,1).EQ.4.OR.TRNSAC(I,1).EQ.5) THEN
C ***** NEXT LINE SUMS DELIVERY PRICES TO DETERMINE TOTAL PRICE.
          TOTPYD=TOTPYD+TRNSAC(I,3)
          IF (TRNSAC(I,3).GT.999999.OR.TRNSAC(I,3).LE.0) THEN
C    ****************************************************
90          FORMAT(  A DELIVERY PAYMENT MUST HAVE A VALUE LARGER THAN',
     &,   0 BUT NOT CONTAIN MORE THAN 6 DIGITS. THE TRANSACTION DATED  ,
     &,    ,F7.0,  VIOLATES THIS RULE. )
            WRITE(6,90) TRNSAC(I,2)
            ERROR= YES
            WRITE(6,400)
          END IF
          IF (TRNSAC(I,1).EQ.5) THEN
            IF (TRNSAC(I,4).LT.0.OR.TRNSAC(I,4).GT.1) THEN
C    ****************************************************
100             FORMAT(  THE LIQUIDATION RATE ASSOCIATED WITH AN ACTUAL ,/
     &,  DELIVERY MUST BE BETWEEN 0 AND 1. THE TRANSACTION DATED  ,F8.0,
     & VIOLATES THIS RULE.')
              WRITE(6,100) TRNSAC(I,2)
              ERROR= YES
              WRITE(6,400)
            END IF
          END IF
        END IF
        IF (TRNSAC(I,1).LT.1.OR.TRNSAC(I,1).GT.5) THEN
C    ****************************************************
110       FORMAT(' THE TRANSACTION DATED' ,F8.0,' HAS AN ILLEGAL' , / ,
     & TRANSACTION CODE. THE CODE MUST BE A 1, 2, 3, 4, OR 5. )
          WRITE(6,110) TRNSAC(I.2)
          ERROR= YES
          WRITE(6,400)
        END IF
        COSTPD=TOTPYD / (1+PRPRAT)-1
        TH=TOTCOS*1.0005
        IF (COSTPD.GT.TH) THEN
     ****************************************************
          FORMAT(' COST REIMBURSEMENT CANNOT BE GREATER THAN INCURRED
     &, / ,  COST. THE TRANSACTION DATED  ,F8.0,  VIOLATES THIS RULE. )
        FORMAT(  THE DELIVERY PAYMENTS TO DATE EXCEED THE COST TO DATE. )
```

```
126     FORMAT(' TOTAL COST AT THIS POINT FROM COST STATEMENTS',T62,F8.0
     %)
127     FORMAT(' TOTAL DELIVERIES AT THIS POINT',T62,F8.0)
129     FORMAT(' COST REIMBURSED AT THIS POINT BASED ON DELIVERY PRICES'
     %,T62,F8.0)
          WRITE(6,120) TRNSAC(I,2)
          WRITE(6,125)
          WRITE(6,126) TOTCOS
          WRITE(6,127) TOTPYD
          WRITE(6,129) COSTPD
          ERROR='YES'
          WRITE(6,400)
        END IF
      CONTINUE
      DO 500 I=1,NMCSTL
        IF (COSTEL(I,2).NE. INCUR .AND.COSTEL(I,2).NE. PAIDC .AND.COSTEL
     %(I,2).NE. SUBCT ) THEN
C*****************************************************
130          FORMAT(' THE COST ELEMENT NUMBER ',I3,' DOES NOT HAVE A ', / ,
     %  VALID COST ELMENT CODE. ')
          WRITE(6,130) I
          ERROR='YES'
          WRITE(6,400)
        END IF
C     ***********************************************************
C           CHECKS PAID COST AND SUBCONTRACTOR PROGRESS PAYMENT COST
C           ELEMENT DATA
C     ***********************************************************
        IF (ELMENT(I,1).EQ.1) THEN
          IF (ELMENT(I,4).GT.0.0) THEN
C*****************************************************
140            FORMAT( ' THE FLOAT DAYS ON COSTS PAID IN THE PRESENT', / ,
     %ACCOUNTING PERIOD SHOULD BE NEGATIVE. THIS IS NOT THE CASE FOR ', /
     %,' COST ELEMENT ',I3,'. ')
            WRITE(6,140) I
            ERROR='YES
            WRITE(6,400)
          END IF
          IF (ELMENT(I,2).GT.28) THEN
C*****************************************************
145            FORMAT( ' THE FLOAT DAYS FOR UNPAID COSTS CANNOT EXCEED ',/,
     %  28 DAYS.  THIS IS NOT THE CASE FOR COST ELEMENT ',I3,'. ')
            WRITE(6,145) I
            ERROR='YES
            WRITE(6,400)
          END IF
          IF (ELMENT(I,3).LT.0.OR.ELMENT(I,3).GT.1) THEN
C*****************************************************
150            FORMAT(  THE PERCENT PAID IN THE PRESENT ACCOUNTING PERIOD'
     %, / ,  SHOULD BE BETWEEN 0 AND 1 FOR COST ELEMENT ',I3,'. ')
            WRITE(6,150) I
            ERROR='YES
            WRITE(6,400)
          END IF
          IF (ELMENT(I,1).EQ.3.AND.ELMENT(I,5).NE.0) THEN
C*****************************************************
160            FORMAT( ' THE SUBCONTRACTOR PROGRESS PAYMENTS SHOULD ADD',/,
     % TO 0 ACROSS THE LIFE OF THE CONTRACT FOR COST ELEMENT ',I3,'. ')
            WRITE(6,160) I
            ERROR='YES
            WRITE(6,400)
          END IF
        END IF
      CONTINUE
      DO 450 I=1,NCPRG
        IF (TRNSAC(I,1).EQ.1) THEN
          CACPRG=CACPRG+1
        END IF
        IF (TRNSAC(I,1).EQ.3) THEN
          CCOST=CCOST+1
        END IF
450   CONTINUE
      CCOST=CCOST-CACPRG
      TOPROF=TOTPYD-TOTCOS
      PROFIT=TOTPYD / TOTCOS-1
      TODAYS=INT(TRNSAC(NMTRAN,13))
      LSTDEL=INT(TRNSAC(NMTRAN,2))
      IF (TRNSAC(1,1).NE.3) THEN
C     ************************************************
170     FORMAT(' THE FIRST TRANSACTION MUST BE A COST REPORT.')
        WRITE(6,170)
        ERROR='YES'
        WRITE(6,400)
      END IF
      IF (TRNSAC(NMTRAN,1).NE.4.AND.TRNSAC(NMTRAN.1).NE.5) THEN
C     ************************************************
180     FORMAT(' THE LAST TRANSACTION MUST BE A DELIVERY.')
        WRITE(6,180)
        ERROR='YES'
        WRITE(6,400)
      END IF
      IF (CCOST.LT.0) THEN
190     FORMAT(' THE 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 ACTU
     &AL PROGRESS PAYMENT.')
        WRITE(6,190)
        ERROR='YES'
        WRITE(6,400)
      END IF
      P=PRFRAT+0.0005
      PL=PRFRAT-0.0005
      IF (PROFIT.GT.P.OR.PROFIT.LT.PL) THEN
C     ************************************************
200   FORMAT(' THE PROFIT RATE COMPUTED FROM THE DATA FILE (',F5.2,')
     &DOES NOT MATCH THE PROFIT',/,' RATE (',F5.2,') IN THE FIRST LINE O
     &F THE DATA FILE.')
210   FORMAT(' ',T6,'COMPUTED PROFIT BASED ON THE FOLLOWING VALUES COMPU
     &TED FROM DATA INPUTS:')
220   FORMAT(' ',T11,'TOTAL PRICE FROM DELIVERY PRICES',T48,I10,/,T11,
     &'TOTAL COST FROM COST STATEMENTS',T48,I10,/,T11,'PROFIT DOLLARS',
     &T48,I10)
        WRITE(6,200) PROFIT*100,PRFRAT*100
        WRITE(6,210)
        WRITE(6,220) INT(TOTPYD), INT(TOTCOS), INT(TOPROF)
        ERROR='YES'
        WRITE(6,400)
      END IF
       ***********************************************************
            SUMS COST STATEMENT DATA WHEN ERRORS ARE DETECTED IN
```

```
C        EDIT SUBROUTINE.
C    ***************************************************************
      IF (ERROR.EQ.'YES ) THEN
        DO 300 I=1,NMTRAN
          IF (TRNSAC(I,1).EQ.3) THEN
            DO 310 J=1,NMCSTL
              TRNSAC(I,14)=TRNSAC(I,14)+TRNSAC(I,J+2)
 310        CONTINUE
          END IF
 300    CONTINUE
      END IF
 400  FORMAT( / )
      RETURN
      END
C    **********      END OF SUBROUTINE EDIT      **********
C
C
C
C
C    **********      START OF SUBROUTINE DATAIN      **********
C
      SUBROUTINE DATAIN(PRGLAG,DELLAG,FLXFLG,PRGRTE,PRGFLG,
     &ALTLDY,TRNFLG,CSTFLG,RERUN,TWOFLG)
      INTEGER PRGLAG,DELLAG,PRGFLG,ALTLDY,TWOFLG
      CHARACTER*3 TRNFLG,CSTFLG,FLXFLG
      REAL PRGRTE
 30   FORMAT(' THE FOLLOWING DATA IS REQUESTED OF THE USER AT THE')
 31   FORMAT(' TERMINAL SO DIFFERENT CALCULATIONS CAN BE MADE USING )
 32   FORMAT(' THE SAME DATA FILE')
 50   FORMAT(' WHAT IS THE PROGRESS PAYMENT LAG TIME IN DAYS? (TIME )
 51   FORMAT(' FROM WHEN STATEMENT IS ISSUED UNTIL PAYMENT IS REC-')
 52   FORMAT(' EIVED?) )
 80   FORMAT(' WHAT IS THE DELIVERY PAYMENT LAG TIME IN DAYS? (TIME')
 81   FORMAT(' FROM WHEN DELIVERY IS MADE UNTIL PAYMENT IS RECEIVED.)')
 130  FORMAT(' WOULD YOU LIKE TO SPECIFY A CERTAIN PROGRESS PAYMENT')
 131  FORMAT(' RATE FOR THESE CALCULATIONS? A NO ANSWER IMPLIES YOU')
 132  FORMAT(' WANT THE COMPUTER TO CALCULATE THE FLEXIBLE PROGRESS' )
 133  FORMAT(' RATE USING THE CONTRACTOR INVESTMENT RATIO. (YES / NO)')
 170  FORMAT(' WHAT PROGRESS PAYMENT RATE WOULD YOU LIKE TO SPECIFY?')
 171  FORMAT(' (PLEASE USE THE FORMAT 0.XX) )
 190  FORMAT(' AFTER HOW MANY DAYS WOULD YOU LIKE THE ALTERNATE')
 191  FORMAT(' LIQUIDATION RATE USED IN CALCULATIONS? VALUE MUST BE' )
 192  FORMAT(' GREATER THAN OR EQUAL TO 365.')
 210  FORMAT(' WOULD YOU LIKE TO SEE A TRANSACTION REPORT? (YES / NO)')
 230  FORMAT(' WOULD YOU LIKE TO SEE A RECAP OF THE COST REPORT DATA?
     & (YES / NO) )
 250  FORMAT(' THE PROGRESS PAYMENT LAG MUST BE POSITIVE AND CANNOT EXCE
     &ED 56 DAYS.')
 270  FORMAT(  THE DELIVERY PAYMENT LAG MUST BE POSITIVE AND CANNOT EXCE
     &ED 60 DAYS. )
 290  FORMAT( / )
 300  FORMAT( / )
 310  FORMAT(' THE ALTERNATE LIQUIDATION RATE CANNOT BE USED UNTIL )
 311  FORMAT(' ONE YEAR HAS ELASPED ON THE CONTRACT. (365 DAYS) )
      IF (RERUN.EQ.0) THEN
        WRITE(6,30)
        WRITE (6,31)
        WRITE(6,32)
      END IF
C    ***************************************************************
C        ASKS FOR PROGRESS PAYMENT LAG.
```

```
       ****************************************************************
340   WRITE(6,300)
      WRITE(6,60)
      WRITE(6,61)
      WRITE(6,62)
      READ(5,FMT=*), PRGLAG
      IF (PRGLAG.LT.0.OR.PRGLAG.GT.56) THEN
        WRITE(6,250)
        GO TO 340
      END IF
C      ****************************************************************
C           HIGHLIGHTS TWO COST STATEMENTS MAY OCCUR BETWEEN PROGRESS
C           PAYMENTS.
C      ****************************************************************
      IF (PRGLAG.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(5,FMT=*), DELLAG
      IF (DELLAG.LT.0.OR.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(5,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
C           LIEU OF FLEX RATE.
C      ****************************************************************
      IF (FLXFLG(1:1).EQ.'Y') THEN
530     WRITE(6,290)
        WRITE(6,170)
        WRITE(6,171)
        READ(5,FMT=*), PRGRTE
        IF (PRGRTE.LT.0.OR.PRGRTE.GT.1) THEN
570       FORMAT(' THE PROGRESS PAYMENT RATE MUST BE BETWEEN 1 AND 0.
     &/)
          WRITE(6,570)
          GO TO 530
        END IF
        PRGFLG=0
      ELSE
        PRGRTE=1.0
```

```
          PRGFLG=1
        END IF
C        ****************************************************************
C              ASKS WHEN TO SWITCH TO ALTERNATE LIQUIDATION RATE.
C        ****************************************************************
        WRITE(6,290)
        WRITE(6,190)
        WRITE(6,191)
        WRITE(6,192) p V
        READ(5,FMT=*), ALTLDY
        IF (ALTLDY.LT.365) THEN
          WRITE(6,310)
          WRITE(6,311)
          GO TO 750
        END IF
C        ****************************************************************
C              ASKS IF TRANSACTION REPORT IS DESIRED.
C        ****************************************************************
820     WRITE(6,290)
        WRITE(6,220)
        READ(5,FMT='(A3)') TRNFLG
        IF (TRNFLG(1:1).NE.'Y'.AND.TRNFLG(1:1).NE.'N') THEN
          GO TO 820
        END IF
C        ****************************************************************
C              ASKS IF COST RECAP REPORT IS DESIRED.
C        ****************************************************************
860     WRITE(6,290)
        WRITE(6,230)
        READ(5,FMT='(A3)') CSTFLG
        IF (CSTFLG(1:1).NE.'Y'.AND.CSTFLG(1:1).NE.'N') THEN
          GO TO 860
        END IF
        WRITE(6,300)
        RETURN
        END
C        **********       END OF SUBROUTINE DATAIN      **********
C
C
C
C        **********       START OF SUBROUTINE CREPAY      **********
C
        SUBROUTINE CREPAY(TRNSAC,NMTRAN,ACPRG,CCOST,PRGLAG,PRGFLG,STDAY,
       &DELLAG,ERROR)
        REAL TRNSAC(325,20),CCOST
        INTEGER NMTRAN,ACPRG,PRGLAG,PRGFLG,STDAY,DELLAG,Z
        INTEGER CNTFLG,PROG1,DAYS
        CHARACTER*3 ERROR
        PROG1=0
        PRGFLG=0
C        ****************************************************************
C              USED TO MATCH COST STATEMENTS TO PROGRESS PAYMENTS
C              WHEN TWO COST STATEMENTS APPEAR BETWEEN ACTUAL PROGRESS
C              PAYMENTS.
C        ****************************************************************
        IF (CCOST.GT.0) THEN
          CNTFLG=0
          DO 100 I=ACPRG,1,-1
            IF(TRNSAC(I,1).EQ.3.AND.CNTFLG.EQ.0) THEN
              CCOST=CCOST-1
            ACPRG=I
            CNTFLG=1
          END IF
 100    CONTINUE
      END IF
C       *****************************************************************
C           ADDS DELIVERY LAG TO EACH DELIVERY DATE INPUT TO
C           ESTABLISH DATE ON WHICH PAYMENT IS RECEIVED.
C       *****************************************************************
      DO 150 I = 1,NMTRAN
        IF(TRNSAC(I,1).EQ.4 .OR. TRNSAC(I,1).EQ.5) THEN
          TRNSAC(I,13)=TRNSAC(I,13)+DELLAG
          K=INT(TRNSAC(I,13))+STDAY
          Z=K
          TRNSAC(I,2)=KDATE(K)
        END IF
 150  CONTINUE
C       *****************************************************************
C           CREATES A PLANNED PROGRESS PAYMENT TRANSACTION FOR EACH
C           COST STATEMENT NOT COVERED BY AN ACTUAL PROGRESS PAYMENT.
C       *****************************************************************
      DO 200 I=ACPRG,NMTRAN
        IF(TRNSAC(I,1).EQ.3.AND.((INT(TRNSAC(I,13))+PRGLAG+STDAY).LT.Z))
     &  THEN
          PROG1=PROG1+1
          NMTRAN=NMTRAN+1
          TRNSAC(NMTRAN,1)=2
          TRNSAC(NMTRAN,13)=TRNSAC(I,13)+PRGLAG
          K=INT(TRNSAC(NMTRAN,13))+STDAY
          TRNSAC(NMTRAN,2)=KDATE(K)
          PRGFLG=1
        ELSE
          GO TO 200
        END IF
C       *****************************************************************
C           CHECKS TO SEE IF PLANNED PROGRESS PAYMENTS ARE AT LEAST
C           28 DAYS APART.  MAKES SURE PROGRESS PAYMENTS ARE BILLED
C           ONLY ONCE PER MONTH.
C       *****************************************************************
        IF (PROG1.GT.1) THEN
          DAYS=TRNSAC(NMTRAN,13)-TRNSAC(NMTRAN-1,13)
        END IF
        IF (DAYS.LT.28.AND.PROG1.GT.1) THEN
          WRITE(6,300) TRNSAC(NMTRAN,2)
          ERROR='YES'
          WRITE(6,400)
        END IF
 200  CONTINUE
C ***** NEXT LINE SORTS TRANSACTIONS AFTER ALL NEW TRANSACTIONS
C ***** ARE CREATED.
      CALL SORT(TRNSAC,NMTRAN,20,13)
 300  FORMAT(' PLANNED PROGRESS PAYMENTS CANNOT OCCUR LESS',/, THAN 28 D
     &AYS APART.  THE TRANSACTION DATED',/,F7.0,' VIOLATES THIS RULE.  C
     &HECK INPUT FILE',/, TO INSURE COST STATEMENTS ARE AT LEAST 28 DAYS
     & ,/, APART.')
 400  FORMAT(/)
      RETURN
      END
        **********       END OF SUBROUTINE CREPAY       **********
```

```
C
C
C     **********     START OF SUBROUTINE ACTPPY     **********
C
      SUBROUTINE ACTPPY(TRNSAC,A,OUTPUT,FLOATD,CSTFLT,PRFRAT,ALTLDY,STCO
     $ST,PPCNT)
      REAL TRNSAC(325,20),OUTPUT(325,10),CSTFLT(1000,2),PRFRAT,STCOST
      INTEGER A,FLOATD,ALTLDY,PPCNT
C     ************************************************************
C           DETERMINES PAID INCURRED COST AT THE TIME OF THE
C           TRANSACTION.
C     ************************************************************
      DO 60 I=1,FLOATD
        IF (CSTFLT(I,1).LE.TRNSAC(A,13)) THEN
          TRNSAC(A,17)=CSTFLT(I,2)+TRNSAC(A,17)
        END IF
60    CONTINUE
      TRNSAC(A,17)=TRNSAC(A,17)+STCOST
C     ************************************************************
C           NEXT THREE LINES DO THE FOLLOWING:  ROLL COST ELIGIBLE
C           FOR PROGRESS PAYMENT FORWARD; ROLL COST ELIGIBLE FOR 100 0/0
C           PROGRESS PAYMENT FORWARD; AND, KEEP TRACK OF THE CURRENT
C           PROGRESS PAYMENT RATE.
C     ************************************************************
      TRNSAC(A,15)=TRNSAC(A-1,15)
      TRNSAC(A,16)=TRNSAC(A-1,16)
      TRNSAC(A,18)=TRNSAC(A,4)
C     ************************************************************
C           KEEPS TRACK OF LIQUIDATION RATE.
C     ************************************************************
      IF (TRNSAC(A,13).LT.ALTLDY) THEN
        TRNSAC(A,19)=TRNSAC(A,4)
      ELSE
        TRNSAC(A,19)=TRNSAC(A,4) / (1+PRFRAT)
      END IF
C     ************************************************************
C           FILLS IN OUTPUT FOR TRANSACTION REPORT.
C     ************************************************************
      OUTPUT(A,1)=OUTPUT(A-1,1)
      OUTPUT(A,2)=OUTPUT(A,1)-TRNSAC(A,17)
      OUTPUT(A,3)=TRNSAC(A,17)
      OUTPUT(A,4)=OUTPUT(A-1,4)+TRNSAC(A,3)
      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)+TRNSAC(A,3)
C ***** NEXT LINE IS A COUNTER FOR PROGRESS PAYMENTS.  USED IN
C ***** SUBROUTINE PRGPAY IF PROGRESS PAYMENT LAG EXCEEDS COST
C ***** STATEMENT PERIOD.  USED TO MATCH PROGRESS PAYMENTS WITH
C ***** PROPER COST STATEMENT DATA.
      PPCNT=PPCNT+1
      RETURN
      END
C     **********     END OF SUBROUTINE ACTPPY     **********
C
C
C
C     **********     START OF SUBROUTINE PRGPAY     **********
```

```
C
      SUBROUTINE PRGPAY(TRNSAC,A,PRGRTE,CSTFLT,FLOATD,OUTPUT,TOTCHG,
     &SUMCST,TOTPYD,ALTLDY,PRPRAT,STCOST,PPCNT,CSTDAY,TWOFLG)
      REAL FRM15, FRM16, FRM19, FRM20E, FRM21E, FRM25,PAYMNT
      REAL TRNSAC(325,20),PRGRTE,CSTFLT(1000,2),OUTPUT(325,10),TOTCHG,
     &SUMCST,PRPRAT,TOTPYD,STCOST
      INTEGER A,FLOATD,ALTLDY,PPCNT,CSTDAY(170),TWOFLG
C ***** COUNTER DESCRIBED ABOVE AT END OF SUBROUTINE ACTPPY.
      PPCNT=PPCNT+1
C ***** NEXT 14 LINES DO SAME THING AS DESCRIBED ABOVE IN SUBROUTINE
C ***** ACTPPY.
      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,13)) THEN
          TRNSAC(A,17)=TRNSAC(A,17)+CSTFLT(I,2)
        END IF
100   CONTINUE
      TRNSAC(A,17)=TRNSAC(A,17)+STCOST
      TRNSAC(A,18)=PRGRTE
      IF (TRNSAC(A,13).LT.ALTLDY) THEN
        TRNSAC(A,19)=TRNSAC(A,18)
      ELSE
        TRNSAC(A,19)=TRNSAC(A,18) / (1+PRPRAT)
      END IF
C      ***********************************************************
C           MATCHES PROGRESS PAYMENT TO PROPER COST STATEMENT DATA
C           WHEN PROGRESS PAYMENT LAG EXCEEDS COST STATEMENT PERIOD.
C
C           NOTE:  ALL VARIABLES FRMxx EQUATE TO LINE NUMBERS ON
C           DD FORM 1195 (PROGRESS PAYMENT FORM).
C      ***********************************************************
      IF(TWOFLG.EQ.1) THEN
        FRM15=TRNSAC(CSTDAY(PPCNT),15)*TRNSAC(A,18)+TRNSAC(CSTDAY(PPCNT)
     &,16)
        GO TO 110
      END IF
      FRM15=TRNSAC(A,15)*TRNSAC(A,18)+TRNSAC(A,16)
110   FRM16=TOTPYD*TRNSAC(A,19)
      IF (FRM16.LT.FRM15) THEN
        FRM15=FRM16
      END IF
      FRM19=FRM15-OUTPUT(A-1,4)
      IF (TWOFLG.EQ.1) THEN
       FRM20E=(TRNSAC(CSTDAY(PPCNT),15)-SUMCST)*TRNSAC(A,18)+TRNSAC(CSTD
     &AY(PPCNT),16)
       GO TO 120
      END IF
      FRM20E=(TRNSAC(A,15)-SUMCST)*TRNSAC(A,18)+TRNSAC(A,16)
120   FRM21E=(TOTPYD-TOTCHG)*TRNSAC(A,19)
      IF (FRM21E.LT.FRM20E) THEN
        FRM20E=FRM21E
      END IF
      FRM25=FRM20E-OUTPUT(A-1,10)
      IF (FRM25.LT.FRM19) THEN
        FRM19=FRM25
      END IF
      PAYMNT=FRM19
C      ***********************************************************
C           FILLS IN OUTPUT FOR TRANSACTION REPORT.
```

```
C      ***********************************************************
      OUTPUT(A,1)=OUTPUT(A-1,1)
      OUTPUT(A,2)=OUTPUT(A,1)-TRNSAC(A,17)
      OUTPUT(A,3)=TRNSAC(A,17)
      OUTPUT(A,4)=OUTPUT(A-1,4)+PAYMNT
      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)+PAYMNT
      RETURN
      END
C      **********      END OF SUBROUTINE PRGPAY      **********
C
C
C
C      **********      START OF SUBROUTINE COSTST   B**********
C
      SUBROUTINE COSTST(TRNSAC,A,NMCSTL,ELMENT,FLOATD,CSTFLT,OUTPUT,
     &SUBCNT,UPDCNT,UPDFLT,UPDSUB,PRGRTE,STCOST,CNTST,CSTDAY,TWOFLG)
      REAL TRNSAC(325,20),ELMENT(10,5),CSTFLT(1000,2),OUTPUT(325,10),
     &UPDFLT(1000,2),UPDSUB(1000,2),PRGRTE,STCOST
      INTEGER A,NMCSTL,FLOATD,SUBCNT,UPDCNT,CNTST,CSTDAY(170),TWOFLG
      REAL PAID,UNPAID
C      ***********************************************************
C           DETERMINES PAID INCURRED COST ELIGIBILITY FOR PROGRESS
C           PAYMENTS.
C      ***********************************************************
      DO 380 I=1,NMCSTL
        TRNSAC(A,14)=TRNSAC(A,14)+TRNSAC(A,I+2)
        FLOATD=FLOATD+1
C ***** COST ELIGIBILITY CODE 'INCUR'
        IF(ELMENT(I,1).EQ.1) THEN
          CSTFLT(FLOATD,1)=TRNSAC(A,13)+ELMENT(I,2)
          CSTFLT(FLOATD,2)=TRNSAC(A,I+2)
          TRNSAC(A,15)=TRNSAC(A,15)+TRNSAC(A,I+2)
        END IF
C ***** COST ELIGIBILITY CODE 'PAIDC'
        IF(ELMENT(I,1).EQ.2) THEN
          PAID=TRNSAC(A,I+2)*ELMENT(I,3)
          UNPAID=TRNSAC(A,I+2)*(1-ELMENT(I,3))
          CSTFLT(FLOATD,1)=TRNSAC(A,13)+ELMENT(I,2)
          CSTFLT(FLOATD,2)=UNPAID
          UPDCNT=UPDCNT+1
          UPDFLT(UPDCNT,1)=CSTFLT(FLOATD,1)
          UPDFLT(UPDCNT,2)=UNPAID
          FLOATD=FLOATD+1
          CSTFLT(FLOATD,1)=TRNSAC(A,13)+ELMENT(I,4)
          CSTFLT(FLOATD,2)=PAID
          TRNSAC(A,15)=TRNSAC(A,15)+PAID
        END IF
C ***** COST ELIGIBILITY CODE 'SUBCT
        IF(ELMENT(I,1).EQ.3) THEN
          PAID=TRNSAC(A,I+2)*ELMENT(I,3)
          UNPAID=TRNSAC(A,I+2)*(1-ELMENT(I,3))
          CSTFLT(FLOATD,1)=TRNSAC(A,13)+ELMENT(I,2)
          CSTFLT(FLOATD,2)=UNPAID
          SUBCNT=SUBCNT+1
          UPDSUB(SUBCNT,1)=CSTFLT(FLOATD,1)
```

```
            UPDSUB(SUBCNT,2)=UNPAID
            FLOATD=FLOATD+1
            CSTFLT(FLOATD,1)=TRNSAC(A,13)+ELMENT(I,4)
            CSTFLT(FLOATD,2)=PAID
            TRNSAC(A,16)=TRNSAC(A,16)+PAID
          END IF
  380   CONTINUE
C       **************************************************************
C            PUTS UNPAID FLOATED COSTS FOR COST ELIGIBILITY CODE
C            'PAIDC' INTO COST ELIGIBLE FOR PROGRESS PAYMENTS AFTER
C            FLOAT PERIOD EXPIRES.
C       **************************************************************
        DO 450 I=1,UPDCNT
          IF(UPDFLT(I,1).LE.TRNSAC(A,13)) THEN
            TRNSAC(A,15)=TRNSAC(A,15)+UPDFLT(I,2)
            UPDFLT(I,1)=100000
            UPDFLT(I,2)=0
          END IF
  450   CONTINUE
C       **************************************************************
C            PUTS UNPAID FLOATED COSTS FOR COST ELIGIBILITY CODE
C            'SUBCT' INTO COST ELIGIBLE FOR 100 0/0 PROGRESS PAYMENTS
C            AFTER FLOAT PERIOD EXPIRES.
C       **************************************************************
        DO 520 I=1,SUBCNT
          IF(UPDSUB(I,1).LE.TRNSAC(A,13)) THEN
            TRNSAC(A,16)=TRNSAC(A,16)+UPDSUB(I.2)
            UPDSUB(I,1)=100000
            UPDSUB(I,2)=0
          END IF
  520   CONTINUE
C       **************************************************************
C            DETERMINES PAID INCURRED COST AT THE TIME OF THE
C            TRANSACTION.
C       **************************************************************
        DO 570 I=1,FLOATD
          IF(CSTFLT(I,1).LE.TRNSAC(A,13)) THEN
            TRNSAC(A,17)=TRNSAC(A,17)+CSTFLT(I,2)
          END IF
  570   CONTINUE
        TRNSAC(A,17)=TRNSAC(A,17)+STCOST
C       **************************************************************
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)
          OUTPUT(A,1)=OUTPUT(A-1,1)+TRNSAC(A,14)
          OUTPUT(A,4)=OUTPUT(A-1,4)
          OUTPUT(A,5)=OUTPUT(A-1,5)
          OUTPUT(A,6)=OUTPUT(A-1,6)
          OUTPUT(A,7)=OUTPUT(A-1,7)
          OUTPUT(A,8)=OUTPUT(A-1,8)
          OUTPUT(A,10)=OUTPUT(A-1,10)
        ELSE
          TRNSAC(A,15)=TRNSAC(A,15)+STCOST
          OUTPUT(A,1)=TRNSAC(A,14)+STCOST
          TRNSAC(A,18)=PRGRTE
          TRNSAC(A,19)=PRGRTE
        END IF
        OUTPUT(A,2)=OUTPUT(A,1)-TRNSAC(A,17)
        OUTPUT(A,3)=TRNSAC(A,17)
        OUTPUT(A,9)=OUTPUT(A,3)-OUTPUT(A,6)
C        ***************************************************************
C              USED WHEN PROGRESS PAYMENT LAG EXCEEDS COST STATEMENT
C              PERIOD.  MATCHES PROGRESS PAYMENTS WITH PROPER COST
C              STATEMENT DATA.
C        ***************************************************************
        IF (TWOFLG.EQ.1) THEN
          CNTST=CNTST+1
          CSTDAY(CNTST)=A
        END IF
        RETURN
        END
C        **********      END OF SUBROUTINE COSTST      **********
C
C
C
C        **********      START OF SUBROUTINE PLNDEL      **********
C
        SUBROUTINE PLNDEL(TRNSAC,A.ALTLDY,PRPRAT,ALOFLG,CSTFLT,FLOATD,
       *TOTCHG,SUMCST,OUTPUT,PRGRTE,LSTDEL,DELLAG,STCOST,TODAYS,STDAY,
       *ALTPRF,LFLG)
        REAL PRPPAY, LIQDAT, COST, NETPAY.ALTPRF
        REAL TRNSAC(225,20),PRPRAT,CSTFLT(1000,2),TOTCHG,SUMCST,OUTPUT(225
       *,10),STCOST,TODAYS
        INTEGERA,ALTLDY,ALOFLG,FLOATD,STDAY,DELLAG
C ***** NEXT THREE LINES PERFORM SAME OPERATIONS AS DESCRIBED IN
C ***** SUBROUTINE ACTPPY.
        TRNSAC(A,15)=TRNSAC(A-1,15)
        TRNSAC(A,16)=TRNSAC(A-1,16)
        TRNSAC(A,18)=TRNSAC(A-1,18)
C ***** NEXT LINE DETERMINES DATE OF LAST DELIVERY TRANSACTION IN
C ***** TERMS OF DAYS AFTER AWARD.
        K=INT(TODAYS+STDAY)+DELLAG
C ***** NEXT SIX LINES DETERMINE PAID INCURRED COST AT THE TIME
C ***** OF THE TRANSACTION AS PREVIOUSLY DESCRIBED.
        DO 110 I=1,FLOATD
          IF (CSTFLT(I,1).LE.TRNSAC(A,13)) THEN
            TRNSAC(A,17)=TRNSAC(A,17)+CSTFLT(I,2)
          END IF
110     CONTINUE
        TRNSAC(A,17)=TRNSAC(A,17)+STCOST
C        ***************************************************************
C              CONVERTS TO ALTERNATE LIQUIDATION RATE AND MAKES PROFIT
C              CATCH-UP ADJUSTMENT.
C        ***************************************************************
        IF (TRNSAC(A,13).GT.ALTLDY) THEN
          TRNSAC(A,19)=TRNSAC(A,18) / (1+PRPRAT)
          CATCUP=0
          IF (ALOFLG.EQ.0) THEN
            CATCUP=ALTPRF-OUTPUT(A-1,7)
            ALOFLG=1
          END IF
        ELSE
          CATCUP=0
          TRNSAC(A,19)=TRNSAC(A,18)
        END IF
C        ***********************************************************
C           CHECKS TO SEE IF LIQUIDATION AMOUNT EXCEEDS UNLIQUIDATED
C           PROGRESS PAYMENT BALANCE.
C        ***********************************************************
        LIQDAT=TRNSAC(A,3)*TRNSAC(A,19)
        IF(LIQDAT.GT.OUTPUT(A-1,10)) THEN
          LIQDAT=OUTPUT(A-1,10)
            IF(INT(TRNSAC(A,2)).NE.(FDATE(K))) THEN
              LFLG=1
            END IF
        END IF
C ***** NEXT 4 LINES DETERMINE DELIVERY PAYMENT AND THE SPLIT
C ***** BETWEEN COST AND PROFIT.
        NETPAY=TRNSAC(A,3)-LIQDAT
        COST=TRNSAC(A,3) / (1+PRFRAT)
        COSTPD=COST-TRNSAC(A,18)*COST
        PRFPAY=NETPAY-COSTPD
C        ***********************************************************
C           COMPUTES WHAT PROFIT WOULD BE AT ALTERNATE RATE PRIOR
C           TO SWITCHOVER.  USED TO MAKE ONE-TIME PROFIT ADJUSTMENT.
C        ***********************************************************
        IF(ALDFLG.EQ.0) THEN
          ALTPRF=ALTPRF+(TRNSAC(A,3)-(TRNSAC(A,3)/(1+PRFRAT)))
        END IF
C ***** FILLS IN OUTPUT FOR TRANSACTION REPORT.
        OUTPUT(A,1)=OUTPUT(A-1,1)
        OUTPUT(A,2)=OUTPUT(A,1)-TRNSAC(A,17)
        OUTPUT(A,3)=TRNSAC(A,17)
        OUTPUT(A,4)=OUTPUT(A-1,4)
        OUTPUT(A,5)=OUTPUT(A-1,5)+COSTPD
        OUTPUT(A,6)=OUTPUT(A,4)+OUTPUT(A,5)
        OUTPUT(A,7)=OUTPUT(A-1,7)+PRFPAY+CATCUP
        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)-LIQDAT+CATCUP
        TOTCHG=TOTCHG+TRNSAC(A,3)
        SUMCST=SUMCST+COST
C        ***********************************************************
C           ROLLS UP ALL DATA FOR LAST DELIVERY TRANSACTION.  ZEROES
C           OUT FLOAT AND UNLIQUIDATED PROGRESS PAYMENTS.  FILLS IN
C           OUTPUT FOR TRANSACTION REPORT.
C        ***********************************************************
        IF(INT(TRNSAC(A,2)).EQ.(FDATE(K))) THEN
          OUTPUT(A,1)=OUTPUT(A-1,1)
          OUTPUT(A,2)=0
          OUTPUT(A,3)=OUTPUT(A,1)
          OUTPUT(A,4)=OUTPUT(A-1,4)
          OUTPUT(A,5)=OUTPUT(A,3)-OUTPUT(A,4)
          OUTPUT(A,6)=OUTPUT(A,4)+OUTPUT(A,5)
          OUTPUT(A,7)=(OUTPUT(A,3)*(1+PRFRAT))-OUTPUT(A,6)
          OUTPUT(A,8)=OUTPUT(A,6)+OUTPUT(A,7)
          OUTPUT(A,9)=OUTPUT(A,3)-OUTPUT(A,6)
          OUTPUT(A,10)=0
        END IF
        RETURN
        END
         ***********      END OF SUBROUTINE PENDEL      ***********
```

```
C
C
C      **********       START OF SUBROUTINE ACTDEL       **********
C
C ***** CALCULATIONS ARE THE SAME AS DESCRIBED IN SUBROUTINE PLNDEL
C
      SUBROUTINE ACTDEL(TRNSAC,A,PRFRAT,ALTLDY,ALOFLG,CSTFLT,FLOATD,
     &TOTCHG,SUMCST,OUTPUT,LSTDEL,DELLAG,STCOST,TODAYS,STDAY,ALTPRF)
      REAL TRNSAC(325,20),PRFRAT,CSTFLT(1000,2),TOTCHG,SUMCST,OUTPUT(325
     &,10),STCOST,TODAYS,ALTPRF
      INTEGER A,ALTLDY,ALOFLG,FLOATD,STDAY,DELLAG
      REAL PRFPAY,LIQDAT,COST,NETPAY
      TRNSAC(A,15)=TRNSAC(A-1,15)
      TRNSAC(A,16)=TRNSAC(A-1,16)
      TRNSAC(A,18)=TRNSAC(A-1,18)
      TRNSAC(A,19)=TRNSAC(A,4)
      DO 90  I=1,FLOATD
        IF(CSTFLT(I,1).LE.TRNSAC(A,13)) THEN
            TRNSAC(A,17)=TRNSAC(A,17)+CSTFLT(I,2)
        END IF
   90 CONTINUE
      TRNSAC(A,17)=TRNSAC(A,17)+STCOST
      OUTPUT(A,1)=OUTPUT(A-1,1)
      LIQDAT=TRNSAC(A,3)*TRNSAC(A,4)
      IF(LIQDAT.GT.OUTPUT(A-1,10)) THEN
        LIQDAT=OUTPUT(A-1,10)
      END IF
      NETPAY=TRNSAC(A,3)-LIQDAT
      IF(TRNSAC(A,13).GT.ALTLDY) THEN
        CATCUP=0
        IF(ALOFLG.EQ.0) THEN
          ALOFLG=1
          CATCUP=ALTPRF-OUTPUT(A-1,7)
        END IF
      ELSE
        CATCUP=0
      END IF
      COST=TRNSAC(A,3) / (1+PRFRAT)
      COSTPD=COST-TRNSAC(A,18)*COST
      PRFPAY=NETPAY-COSTPD
      IF(ALOFLG.EQ.0) THEN
        ALTPRF=ALTPRF+(TRNSAC(A,3)-(TRNSAC(A,3)/(1+PRFRAT)))
      END IF
      OUTPUT(A,2)=OUTPUT(A,1)-TRNSAC(A,17)
      OUTPUT(A,3)=TRNSAC(A,17)
      OUTPUT(A,4)=OUTPUT(A-1,4)
      OUTPUT(A,5)=COSTPD+OUTPUT(A-1,5)
      OUTPUT(A,6)=OUTPUT(A,4)+OUTPUT(A,5)
      OUTPUT(A,7)=OUTPUT(A-1,7)+PRFPAY+CATCUP
      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)-LIQDAT+CATCUP
      TOTCHG=TOTCHG+TRNSAC(A,3)
      SUMCST=SUMCST+COST
      K=INT(TODAYS+STDAY)+DELLAG
C      ****************************************************************
C           ROLLS UP ALL DATA FOR LAST TRANSACTION.  (IT IS EXTREMELY
C           UNLIKELY THAT AN ACTUAL DELIVERY WOULD BE THE LAST
C           TRANSACTION.  HOWEVER, IT MAY HAPPEN IF SOMEONE CHOOSES
C           TO RUN A COMPLETED CONTRACT TO SEE WHAT THE CONTRACTOR
```

```
         INVESTMENT WAS BASED ON ACTUALS.)
    ************************************************************
  IF (INT(TRNSAC(A,1)).EQ.(FDATE))   THEN
    OUTPUT(A,1)=OUTPUT(A-1,1)
    OUTPUT(A,2)=0
    OUTPUT(A,3)=OUTPUT(A,1)
    OUTPUT(A,4)=OUTPUT(A-1,4)
    OUTPUT(A,5)=OUTPUT(A,3)-OUTPUT(A,4)
    OUTPUT(A,6)=OUTPUT(A,4)+OUTPUT(A,5)
    OUTPUT(A,7)=(OUTPUT(A,3)*(1+PRFRAT))-OUTPUT(A,6)
    OUTPUT(A,8)=OUTPUT(A,6)+OUTPUT(A,7)
    OUTPUT(A,9)=OUTPUT(A,3)-OUTPUT(A,6)
    OUTPUT(A,10)=0
  END IF
  RETURN
  END
    ***********      END OF SUBROUTINE ACTUAL      ***********



    ***********      START OF SUBROUTINE INVEST      ***********
  SUBROUTINE INVEST(TRNSAC,NMTRAN,TODAYS,CSTFLT,FLOATD,RATIO,OUTPUT,
 &DELLAG,STCOST,PRFRAT)
  REAL TRNSAC(325,20),CSTFLT(100,2),RATIO,OUTPUT(325,10)
  INTEGER NMTRAN,FLOATD,DELLAG
  REAL INVSTD,FDCOST,INFLOW,STCOST,TODAYS,COSDEL,PRFRAT
  FDCOST=0
  INFLOW=0
  COSDEL=0
    ************************************************************
         COMPUTES DAY DOLLARS OF COST PAID OUT BY CONTRACTOR.
    ************************************************************
  DO 100 I=1,FLOATD
    FDCOST=FDCOST+((TODAYS-INT(CSTFLT(I,1)))*CSTFLT(I,2))
  CONTINUE
  FDCOST=FDCOST+(TODAYS*STCOST)
    ************************************************************
         COMPUTES DAY DOLLARS OF COST REIMBURSEMENT.
    ************************************************************
  DO 110 I=1,NMTRAN
***** FROM ACTUAL PROGRESS PAYMENTS
    IF (TRNSAC(I,1).EQ.1) THEN
      INFLOW=INFLOW+((TODAYS-INT(TRNSAC(I,13)))*TRNSAC(I,3))
    END IF
***** FROM PLANNED PROGRESS PAYMENTS
    IF (TRNSAC(I,1).EQ.2) THEN
      INFLOW=INFLOW+((TODAYS-INT(TRNSAC(I,13)))*(OUTPUT(I,4)-OUTPUT(
 &I-1,4)))
    END IF
***** FROM DELIVERY PAYMENTS
    IF (TRNSAC(I,1).EQ.4.OR.TRNSAC(I,1).EQ.5) THEN
      INFLOW=INFLOW+((TODAYS-INT(TRNSAC(I,13)))*(OUTPUT(I,5)-
 &OUTPUT(I-1,5)))
    ************************************************************
         COMPUTES DAY DOLLARS OF COST OF DELIVERIES.
    ************************************************************
      COSDEL=COSDEL+((TODAYS-INT(TRNSAC(I,13)))*(TRNSAC(I,3)/(1+PRFR
 &AT)))
    END IF
    CONTINUE
```

```
       ***************************************************************  CAS13790
            COMPUTES CONTRACTOR INVESTMENT RATIO.                       CAS13800
       ***************************************************************  CAS13810
      INVSTD=PDCOST-INFLOW                                              CAS13820
      RATIO=(INVSTD / (PDCOST-COSDEL))                                  CAS13830
      RETURN                                                            CAS13840
      END                                                               CAS13850
       **********       END OF SUBROUTINE INVEST      **********        CAS13860
                                                                        CAS13870
                                                                        CAS13880
                                                                        CAS13890
       **********       START OF SUBROUTINE PNTOUT      **********      CAS13900
                                                                        CAS13910
      SUBROUTINE PNTOUT(TRNFLG,CSTFLG,PRGFLG,TRNSAC,ELMENT,FLXFLG,NMTRANCAS13920
     & ,NMCSTL,STDATE,COSTEL,PRPRAT,STCOST,TOTPYD,TOTCOS,OUTPUT,TOPROF,CAS13930
     &   PROFIT,PRGRTE,RATIO,PRGLAG,DELLAG,LSTDEL,TITLE,DATKEY,LFLG)    CAS13940
      REAL TRNSAC(325,20),ELMENT(10,5),PRPRAT,TOTCOS,OUTPUT(325,10),    CAS13950
     &TOPROF,PROFIT,PRGRTE,RATIO,TOTPYD,STCOST                          CAS13960
      INTEGER PRGFLG,NMCSTL,STDATE,PRGLAG,DELLAG,LSTDEL,PG,NMTRAN       CAS13970
      CHARACTER*3 TRNFLG,CSTFLG,FLXFLG                                  CAS13980
      CHARACTER*20 COSTEL(10,3)                                         CAS13990
      CHARACTER*10 ROWA(8),ROWB(10),ROWC(11),ROWD(3),ROWE(10)           CAS14000
      CHARACTER*50 TITLE                                                CAS14010
      LOGICAL*1 CC                                                      CAS14020
      INTEGER NUM(10),DATKEY                                            CAS14030
      DATA CC / Z0C /                                                   CAS14040
      OPEN(UNIT=3)                                                      CAS14050
   21 FORMAT( / )                                                       CAS14060
   31 FORMAT( / )                                                       CAS14070
C      ***************************************************************  CAS14080
C           ESTABLISHES HEADINGS FOR TRANSACTION REPORT                 CAS14090
C      ***************************************************************  CAS14100
      PG=0                                                              CAS14110
        ROWA(1)='DATE /'                                                CAS14120
        ROWA(2)='TOTAL'                                                 CAS14130
        ROWA(3)='PAID'                                                  CAS14140
        ROWA(4)='COST'                                                  CAS14150
        ROWA(5)='TOTAL'                                                 CAS14160
        ROWA(6)='PROFIT'                                                CAS14170
        ROWA(7)='UNREIM'                                                CAS14180
        ROWA(8)='UNLIQ'                                                 CAS14190
        ROWB(1)='TRANSACT'                                              CAS14200
        ROWB(2)='INCURRED'                                              CAS14210
        ROWB(3)='INCURRED'                                              CAS14220
        ROWB(4)='PROG PAY'                                              CAS14230
        ROWB(5)='PAID AT'                                               CAS14240
        ROWB(6)='COST'                                                  CAS14250
        ROWB(7)='PAID AT'                                               CAS14260
        ROWB(8)='TOTAL'                                                 CAS14270
        ROWB(9)='PAID'                                                  CAS14280
        ROWB(10)='PROGRESS'                                             CAS14290
        ROWC(1)='CODE'                                                  CAS14300
        ROWC(2)='COST'                                                  CAS14310
        ROWC(3)='FLOAT'                                                 CAS14320
        ROWC(4)='COST'                                                  CAS14330
        ROWC(5)='RECD'                                                  CAS14340
        ROWC(6)='DELIVERY'                                              CAS14350
        ROWC(7)='REIMBURS'                                              CAS14360
        ROWC(8)='DELIVERY'                                              CAS14370
        ROWC(9)='REVENUE'                                               CAS14380
```

```
         ROWC(10)=' COST'
         ROWC(11)=' PAYMENTS'
         ROWD(1)=' (4+5)'
         ROWD(2)=' (6+7)'
         ROWD(3)=' (3-6)'
         DO 650 I=1,10
           ROWE(I)=' (DOLLARS)'
           NUM(I)=I
  650    CONTINUE
  100    FORMAT(T43,' DEPARTMENT OF DEFENSE CONTRACT FINANCING MODEL ')
  110    FORMAT(T51,' VERSION DATED FEBRUARY 1, 1983')
  120    FORMAT(T57,' TRANSACTION REPORT ')
  121    FORMAT(T41,A50)
  122  FORMAT(' ',T13,' *** UNLIQUIDATED PROGRESS PAYMENTS WOULD HAVE BEE
      &N NEGATIVE WHERE 0 APPEARS IN COLUMN 10. *** ')
  123  FORMAT(' ',T13,' ***            CHECK PRICES OF DELIVERABLES TO ENSU
      &RE THERE IS NO -FRONT LOADING-.          ***')
         COUNT3=34
C      ***************************************************************
C            CONTROLS PRINT ROUTINE WHEN ERRORS ARE DETECTED IN THE
C            EDIT SUBROUTINE.
C      ***************************************************************
       IF(DATKEY.GT.0) THEN
         GO TO 1850
       END IF
C      ***************************************************************
C            PRINTS TRANSACTION REPORT.
C      ***************************************************************
       IF (TRNFLG.EQ.'YES') THEN
         DO 1070    I=1,NMTRAN
           IF(COUNT3.EQ.34) THEN
             PG=PG+1
             WRITE(7,FMT='(A1)') CC
  740        FORMAT( / ,T112,A,T118,I2)
  750        FORMAT( / ,A)
            WRITE(2,480)PG
            WRITE(2,100)
            WRITE(2,110)
            WRITE(2,120)
            WRITE(2,30)
            WRITE(2,121) TITLE
             WRITE(2,*)' '
  820  FORMAT(' ',T18,I1,T29,I1,T40,I1,T51,I1,T62,I1,T73,I1,T84,I1,T95,I1
      &,T106,I1,T117,I2)
             WRITE(2,820) (NUM(K),K=1,10)
  840        FORMAT(' ',T2,A,T16,A,T38,A,T62,A,T71,A,T81,A,T104,A,T115,A
      &)
             WRITE(2,840) (ROWA(K),K=1,8)
  860        FORMAT(' ',T1,A,T14,A,T36,A,T47,A,T59,A,T71,A,T81,A,T93,A,
      &        T104,A,T113,A)
  890        FORMAT(' ',T2,A,T16,A,T27,A,T38,A,T49,A,T58,A,T69,A,T80,A,
      &        T92,A,T104,A,T113,A)
             WRITE(2,860) (ROWB(K),K=1,10)
             WRITE(2,890) (ROWC(K),K=1,11)
  920        FORMAT(' ',T71,A,T93,A,T104,A)
             WRITE(2,920) ROWD(1),ROWD(2),ROWD(3)
  940        FORMAT(' ',T14,A,T25,A,T36,A,T47,A,T58,A,T69,A,T80,A,T91,A,
      &        T102,A,T113,A)
             WRITE(2,940) (ROWE(K),K=1,10)
             COUNT3=0
```

```
          WRITE(2,20)
        END IF
990     FORMAT(' ',I6,' ',I1,1X,I10,1X,I10,1X,I10,1X,I10,1X,
     *   I10,1X,I10,1X,I10,1X,I10,1X,I10,1X,I10)
        DO 1040 J=1,10
          CALL ROUND(OUTPUT(I,J))
1040    CONTINUE
      WRITE(2,990)INT(TRNSAC(I,2)),INT(TRNSAC(I,1)),INT(OUTPUT(I,1)),INT
     *(OUTPUT(I,
     *2)),INT(OUTPUT(I,3)),INT(OUTPUT(I,4)),INT(OUTPUT(I,5)),INT(OUTPUT(
     &I,6)),INT(OUTPUT(I,7)),INT(OUTPUT(I,8)),INT(OUTPUT(I,9)),INT(OUTPU
     *T(I,10))
        COUNT3=COUNT3+1
       CONTINUE
       IF(LFLG.EQ.1) THEN
        WRITE(2,20)
        WRITE(2,122)
        WRITE(2,123)
       END IF
      END IF
C     *************************************************************
C          ESTABLISHES HEADINGS FOR ANALYTICAL REPORT.
C     *************************************************************
200   FORMAT(T57,' ANALYTICAL REPORT')
210   FORMAT(' CONTRACT PRICE',T31,F10.2)
220   FORMAT(' CONTRACT START DATE',T35,I6)
230   FORMAT(' START UP COST OF CONTRACT',T31,F10.2)
240   FORMAT(' DATA FILE PROFIT PERCENTAGE',T36,F5.2)
[illegible]   FORMAT(' COMPUTED PROFIT PERCENTAGE',T36,F5.2)
250   FORMAT(' PROJECT TITLE',T18,A50)
270   FORMAT(' TOTAL CONTRACTOR COST',T31,F10.2)
280   FORMAT('**** PROGRESS PAYMENT RATE DATA****')
290   FORMAT(' PROGRESS PAYMENT RATE',T35,' *',T38,F6.2,T45,
     &' *')
295   FORMAT(' PROGRESS PAYMENT RATE',T35,' *',T38,F6.2,T45,' *',T68,
     &' *    THESE VALUES REFLECT COMPUTATIONS  *')
300   FORMAT(' ORDINARY LIQUIDATION RATE',T35,' *',T38,F6.2,T45,' *')
305   FORMAT(' ORDINARY LIQUIDATION RATE',T35,' *',T38,F6.2,T45,' *',T68
     &,' *    BASED ON OVERLIQUIDATION OF         *')
310   FORMAT(' ALTERNATE LIQUIDATION RATE',T35,' *',T38,F6.2,T45,' *')
315   FORMAT(' ALTERNATE LIQUIDATION RATE',T35,' *',T38,F6.2,T45,' *',
     &T68,' *    PROGRESS PAYMENTS.  SEE NOTE AT      *')
320   FORMAT(' CONTRACTOR INVESTMENT RATIO',T35,' *',T39,F5.2,T45,' *')
325   FORMAT(' CONTRACTOR INVESTMENT RATIO',T35,' *',T39,F5.2,T45,' *',
     &T68,' *    END OF TRANSACTION REPORT.        *')
330   FORMAT(' ***********************************************************
     &*****************')
340   FORMAT(' PROGRESS PAYMENT LAG TIME IN DAYS',T41,I3)
350   FORMAT(' DELIVERY PAYMENT LAG TIME IN DAYS',T41,I3)
360   FORMAT(' DATE OF FINAL DELIVERY',T40,I6)
      FORMAT(' PROGRESS PAYMENT RATE COMPUTED BY PROGRAM',T54,F6.2)
      FORMAT(' PROGRESS PAYMENT RATE INPUT BY USER INTO PROGRAM',T54,
     &F6.2)
      FORMAT(T31,'COST ELEMENT DATA')
410   FORMAT(' #',T6,'NAME',T28,'UNPAID COST',T41,'%PAID',T48,'PAID
     &FLOAT',T60,'ELGIBILITY')
      FORMAT(T28,'DAYS FLOAT',T48,'DAYS FLOAT',T60,'PROGRESS PAYMEN

      FORMAT(' ',I2,T6,A20,T31,I3,T41,F5.1,T51,I3,T5 ,'SUBCONTRACT PROGR
     &PAY')
```

```
440   FORMAT(' ',I2,T6,A20,T31,I3,T41,F5.1,T51,I3,T60,' PAID COST')
450   FORMAT(' ',I2,T6,A20,T31,I3,T60,  INCURRED COST')
460   FORMAT(' ',I2,T6,A20,T51,I3,T60,' INCURRED COST')
470   FORMAT(T35,' ************')
475   FORMAT(T35,' ************',T68,' *********************************
     &********')
480   FORMAT(T123,' PAGE ,T128,I3)
490   FORMAT(' PROFIT PAID TO CONTRACTOR',T31,F10.2)
500   FORMAT(T35,' *',T45,' *')
505   FORMAT(T35,' *',T45,' *',T68,' *',T108,' *')
      WRITE(2,FMT='(A1)') CC
C      ***************************************************************
C           PRINTS SUMMARY DATA IN ANALYTICAL REPORT.
C      ***************************************************************
      PG=PG+1
      WRITE(2,480) PG
      WRITE(2,30)
      WRITE(2,100)
      WRITE(2,110)
      WRITE(2,200)
      WRITE(2,30)
      WRITE(2,260) TITLE
      WRITE(2,30)
      WRITE(2,330)
      WRITE(2,210) TOTPYD
      WRITE(2,270) TOTCOS
      WRITE(2,490) TOPROF
      WRITE(2,230) STCOST
      WRITE(2,220) STDATE
      PRFRAT=PRFRAT*100
      PROFIT=PROFIT*100
      WRITE(2,240) PRFRAT
      WRITE(2,250) PROFIT
      WRITE(2,330)
      WRITE(2,20)
      WRITE(2,280)
      WRITE(2,20)
      PRGRTE=PRGRTE*100
      RATIO=RATIO*100
C      ***************************************************************
C           PRINTS PROGRESS PAYMENT RATE, ORDINARY AND ALTERNATE
C           LIQUIDATION RATES, AND CONTRACTOR INVESTMENT RATIO.
C           ALSO PRINTS WARNING IF FLEX RATE IS COMPUTED BASED ON
C           OVERLIQUIDATION OF PROGRESS PAYMENTS.
C      ***************************************************************
      IF(LFLG.EQ.1) THEN
        WRITE(2,475)
        WRITE(2,295) PRGRTE
        WRITE(2,505)
        WRITE(2,305) PRGRTE
        WRITE(2,505)
        WRITE(2,315) TRNSAC(NMTRAN,19)*100
        WRITE(2,505)
        WRITE(2,325) RATIO
        WRITE(2,475)
      ELSE
        WRITE(2,470)
        WRITE(2,290) PRGRTE
        WRITE(2,500)
        WRITE(2,300) PRGRTE
        WRITE(2,500)
        WRITE(2,310) TRNSAC(NMTRAN,19)*100
        WRITE(2,500)
        WRITE(2,320) RATIO
        WRITE(2,470)
      END IF
      WRITE(2,20)
      WRITE(2,330)
      IF(PRGFLG.EQ.0) THEN
 510    FORMAT(' ALL PROGRESS PAYMENTS INPUT ARE ACTUAL.')
        WRITE(2,510)
      END IF
C      ********************************************************
C             PRINTS COST ELEMENT DATA.
C      ********************************************************
      WRITE(2,330)
      WRITE(2,20)
      WRITE(2,400)
      WRITE(2,410)
      WRITE(2,420)
      WRITE(2,330)
      DO 1320 I=1,NMCSTL
        IF(ELMENT(I,1).EQ.1.AND.ELMENT(I,2).GE.0) THEN
          WRITE(2,450) I,COSTEL(I,1),INT(ELMENT(I,2))
        END IF
        IF(ELMENT(I,1).EQ.1.AND.ELMENT(I,2).LT.0) THEN
          WRITE(2,460) I,COSTEL(I,1),INT(ELMENT(I,2))
        END IF
        ELMENT(I,3)=ELMENT(I,3)*100
        IF(ELMENT(I,1).EQ.2) THEN
          WRITE(2,440) I,COSTEL(I,1),INT(ELMENT(I,2)),ELMENT(I,3),
     &INT(ELMENT(I,4))
        END IF
        IF(ELMENT(I,1).EQ.3) THEN
          WRITE(2,430) I,COSTEL(I,1),INT(ELMENT(I,2)),ELMENT(I,3),
     &INT(ELMENT(I,4))
        END IF
 1320 CONTINUE
      WRITE(2,330)
      WRITE(2,340) PRGLAG
      WRITE(2,350) DELLAG
      WRITE(2,360) LSTDEL
      WRITE(2,20)
      IF(FLXFLG.EQ.'YES') THEN
        WRITE(2,390)PRGRTE
      ELSE
        WRITE(2,380)PRGRTE
      END IF
      WRITE(2,330)
C      ********************************************************
C             PRINTS COST RECAP REPORT.
C      ********************************************************
 2150 IF(DATKEY.EQ.1.OR.DATKEY.EQ.3) THEN
        CSTFLG='YES'
      END IF
      IF(CSTFLG.EQ.'YES') THEN
      TRNSAC(1,14)=TRNSAC(1,14)+STCOST
      COUNT3=36
      DO 2190 I=1,NMTRAN
        IF(COUNT3.EQ.36) THEN
```

```
         COUNT3=0                                                       CAS16790
       WRITE(2,FMT='(A1)') CC                                           CAS16800
       PG=PG+1                                                          CAS16810
       WRITE(2,480)PG                                                   CAS16820
1900 FORMAT(T54,' RECAP OF COST STATEMENT')                             CAS16830
1905 FORMAT(T36,' NOTE: TOTAL FOR FIRST COST STATEMENT INCLUDES START UCAS16840
    &P COST')                                                           CAS16850
       WRITE(2,100)                                                     CAS16860
       WRITE(2,110)                                                     CAS16870
       WRITE(2,1900)                                                    CAS16880
       WRITE(2,1905)                                                    CAS16890
       WRITE(2,30)                                                      CAS16900
       WRITE(2,121) TITLE                                               CAS16910
       WRITE(2,30)                                                      CAS16920
2060 FORMAT(' ',T2,' DATE',T17,' #1',T28,' #2',T39,' #3',T50,' #4',T61,CAS16930
    &' #5',T72,' #6',T83,' #7',T95,' #8',T105,' #9',T114,' #10',T122,  CAS16940
    &' TOTAL')                                                          CAS16950
       WRITE(2,2060)                                                    CAS16960
       END IF                                                           CAS16970
2110 FORMAT(' ',1X,I6,1X,11(I10,1X))                                    CAS16980
2120 FORMAT(' ',1X,' TOTAL',1X,11(I10,1X))                              CAS16990
         IF(TRNSAC(I,1).EQ.3) THEN                                      CAS17000
           WRITE(2,2110)INT(TRNSAC(I,2)),INT(TRNSAC(I,3)),INT(TRNSAC(I,CAS17010
    &4)),INT(TRNSAC(I,5)),INT(TRNSAC(I,6)),INT(TRNSAC(1,7)),INT(TRNSAC(CAS17020
    &I,8)),INT(TRNSAC(I,9)),INT(TRNSAC(I,10)),INT(TRNSAC(I,11)),INT(TRNCAS17030
    &SAC(I,12)),INT(TRNSAC(I,14))                                       CAS17040
           COUNT3=COUNT3+1                                              CAS17050
         END IF                                                         CAS17060
2190 CONTINUE                                                           CAS17070
     WRITE(2,20)                                                        CAS17080
     WRITE(2,2120)INT(ELMENT(1,5)),INT(ELMENT(2,5)),INT(ELMENT(         CAS17090
    &3,5)),INT(ELMENT(4,5)),INT(ELMENT(5,5)),INT(ELMENT(6,5)),INT(ELMENCAS17100
    &T(7,5)),INT(ELMENT(8,5)),INT(ELMENT(9,5)),INT(ELMENT(10,5)),INT(TOCAS17110
    &TCOS)                                                              CAS17120
     END IF                                                             CAS17130
C     *****************************************************************  CAS17140
C          PRINTS DELIVERY SUMMARY.                                     CAS17150
C     *****************************************************************  CAS17160
     IF(DATKEY.EQ.2.OR.DATKEY.EQ.3) THEN                                CAS17170
       COUNT3=36                                                        CAS17180
       DO 2200 I=1,NMTRAN                                               CAS17190
         IF(COUNT3.EQ.36) THEN                                          CAS17200
           COUNT3=0                                                     CAS17210
           WRITE(2,FMT='(A1)') CC                                       CAS17220
           PG=PG+1                                                      CAS17230
           WRITE(2,480) PG                                              CAS17240
           WRITE(2,100)                                                 CAS17250
           WRITE(2,110)                                                 CAS17260
           WRITE(2,2300)                                                CAS17270
           WRITE(2,30)                                                  CAS17280
           WRITE(2,121) TITLE                                           CAS17290
           WRITE(2,30)                                                  CAS17300
           WRITE(2,2310)                                                CAS17310
           WRITE(2,2320)                                                CAS17320
         END IF                                                         CAS17330
         IF(TRNSAC(I,1).EQ.4.OR.TRNSAC(I,1).EQ.5) THEN                  CAS17340
           WRITE(2,2330) INT(TRNSAC(I,2)), INT(TRNSAC(I,3))             CAS17350
           COUNT3=COUNT3+1                                              CAS17360
         END IF                                                         CAS17370
2200   CONTINUE                                                         CAS17380
```

```
         WRITE(2,20)                                                            CAS17390
         WRITE(2,2350) INT(TOTPYD)                                              CAS17400
      END IF                                                                    CAS17410
 200  FORMAT(T58,'DELIVERY SUMMARY')                                            CAS17420
 210  FORMAT(T49,'TRANSACTION', T75, 'DELIVERY')                                CAS17430
 220  FORMAT(T52,'DATE', T76, 'AMOUNT')                                         CAS17440
 230  FORMAT(' ',T52, I6, T76, I6)                                              CAS17450
 250  FORMAT(T52, 'TOTAL', T74, I8)                                             CAS17460
      CLOSE(UNIT=2)                                                             CAS17470
      RETURN                                                                    CAS17480
      END                                                                       CAS17490
C      ***********      END OF SUBROUTINE PNTOUT      ***********               CAS17500
C                                                                               CAS17510
C                                                                               CAS17520
C                                                                               CAS17530
C      ***********      START OF SUBROUTINE SORT      ***********               CAS17540
C                                                                               CAS17550
C ***** SORTS TRANSACTIONS AFTER PLANNED PROGRESS PAYMENT TRANSACTIONS/ CAS17560
C ***** DATES AND DATES FOR RECEIPT OF DELIVERY PAYMENTS ARE CREATED.   CAS17570
C                                                                               CAS17580
      SUBROUTINE SORT(ARNAME,ROWS,COLMNS,SORTCM)                                CAS17590
      REAL TEMP                                                                 CAS17600
      INTEGER ROWS,COLMNS,SORTCM                                                CAS17610
      DIMENSION ARNAME(325,20)                                                  CAS17620
      DO 140 I=1,ROWS                                                           CAS17630
        DO 120 J=1,(ROWS-1)                                                     CAS17640
          IF (ARNAME(J+1,SORTCM).LT.ARNAME(J,SORTCM)) THEN                      CAS17650
            DO 100 K=1,COLMNS                                                   CAS17660
              TEMP=ARNAME(J,K)                                                  CAS17670
              ARNAME(J,K)=ARNAME(J+1,K)                                         CAS17680
              ARNAME(J+1,K)=TEMP                                                CAS17690
100         CONTINUE                                                            CAS17700
          END IF                                                                CAS17710
120     CONTINUE                                                                CAS17720
140   CONTINUE                                                                  CAS17730
      RETURN                                                                    CAS17740
      END                                                                       CAS17750
C      ***********      END OF SUBROUTINE SORT      ***********                 CAS17760
C                                                                               CAS17770
C                                                                               CAS17780
C                                                                               CAS17790
C      ***********      START OF SUBROUTINE YRMODA      ***********             CAS17800
C                                                                               CAS17810
C ***** USED WITH PDAY TO CONVERT YYMMDD DATES TO DAYS AFTER AWARD              CAS17820
C                                                                               CAS17830
      SUBROUTINE YRMODA(YR,MO,DA,J)                                             CAS17840
      INTEGER YR,DA                                                             CAS17850
      IF(MO.GE.3) GO TO 1                                                       CAS17860
      M=MO+9                                                                    CAS17870
      J=YR-1961                                                                 CAS17880
      GO TO 2                                                                   CAS17890
 1    M=MO-3                                                                    CAS17900
      J=YR-1960                                                                 CAS17910
 2    J=J*1461                                                                  CAS17920
      IF(J.LT.0) J=J-3                                                          CAS17930
      J=J / 4+(153*M+2) / 5+DA                                                  CAS17940
      RETURN                                                                    CAS17950
      END                                                                       CAS17960
       ***********      END OF SUBROUTINE YRMODA      ***********               CAS17970
                                                                                CAS17980
```

```
C
C      **********      START OF SUBROUTINE ROUND      **********
C
C ***** ROUNDS OUTPUT MATRIX PRIOR TO PRINTING TRANSACTION REPORT
C
      SUBROUTINE ROUND(IRD)
      REAL IRD,IFL
        IIT=INT(IRD)
        IFL=IRD-IIT
        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
C      **********      END OF SUBROUTINE ROUND      **********
C
C
C
C      **********      START OF INTEGER FUNCTION FDAY      **********
C
C ***** USED WITH YRMODA TO CONVERT YYMMDD DATES TO DAYS AFTER AWARD
C
      INTEGER FUNCTION FDAY(IN)
      INTEGER IYR,MO,IDA,IN
      IYR=IN / 10000
      I=IYR*10000
      J=IN-I
      MO=J / 100
      I=MO*100
      IDA=J-I
      CALL YRMODA(1900+IYR,MO,IDA,I)
      FDAY=I
      RETURN
C ***** USED WITH JYMD TO CONVERT DAYS AFTER AWARD TO YYMMDD DATES
      ENTRY FDATE(IN)
        CALL JYMD(IN,IYR,MO,IDA)
        IYR=(IYR-1900)*10000
        MO=MO*100
        FDATE=IYR+MO+IDA
        RETURN
        END
C      **********      END OF INTEGER FUNCTION FDAY      **********
C
C
C
C      **********      START OF SUBROUTINE JYMD      **********
C
C ***** USED WITH FDATE TO CONVERT DAYS AFTER AWARD TO YYMMDD DATES
C
      SUBROUTINE JYMD(J,YR,MO,DA)
      INTEGER YR,DA
      IF(J.GT.0) GOTO 4
      YR=4*J / 1461-1
      DA=J-(YR*1461-3) / 4
      GO TO 3
```

```
4     YR=(4*J-1) / 1461
      DA=J-(YR*1461) / 4
      MO=(5*DA-3) / 153
      DA=DA-(153*MO+2) / 5
      IF(MO-9) 2,2,1
2     MO=MO+3
      YR=YR+1960
      RETURN
1     MO=MO-9
      YR=YR+1961
      RETURN
      END
*ERROLESS OPT(3)
      SUBROUTINE GETSTR(STRNG,DELIM,SUBSTR,IPOS)
      CHARACTER*(*) STRNG,SUBSTR,DELIM
      INTEGER*4 IPOS,LENDLM,LENSTR
      IPOS=INDEX(STRNG,DELIM)
      IF (IPOS.GT.0) THEN
            LENDLM=LEN(DELIM)
            LENSTR=LEN(STRNG)
            IF(IPOS-1.LE.0) THEN
                  SUBSTR = ' '
                  GOTO 100
            ENDIF
            SUBSTR=STRNG(1:IPOS-1)
100         STRNG=STRNG(IPOS+LENDLM:LENSTR)
      ELSE
            SUBSTR=STRNG
            STRNG=' '
      ENDIF
      RETURN
      END
*ERROLESS OPT(3)
      FUNCTION REALSV(STRNG1,IN)
      CHARACTER*8 STRNG1,STRNG2
      CHARACTER*1 STRARY(8)
      INTEGER*4 I,J,IDECPL,INTLN,ISIGN,ISTRLN,IN,IDECEO(8),IPWR
      INTEGER*4 IBLNK,IBLNK2
      REAL REALSV
      EQUIVALENCE (STRNG2,STRARY)
      STRNG2=STRNG1
      J=1
      IBLNK=0
      IBLNK2=0
      IN=0
      IDECPL=0
      ISIGN =1
      REALSV=0
      DO 100 I=1,8,1
      IDECEO(I)= ICHAR(STRARY(I))
      IF((IDECEO(I).LT.240.OR.IDECEO(I).GT.249).AND.(IDECEO(I).NE.75.AN
     1D.IDECEO(I).NE.96.AND.IDECEO(I).NE.64)) THEN
            IN=1
            GOTO 400
      ENDIF
      IF(IDECEO(I).EO.75.AND.IDECPL.NE.0) THEN
         IN=1
         GOTO 400
      ENDIF
      IF(IDECEO(I).EO.75) IDECPL=I
      IF(IDECED(I).EQ.96.AND.I.NE.1) THEN
        IN=1
        GOTO 400
      ENDIF
      IF(IDECED(I).EQ.96) ISIGN=-1
      IF(IDECED(I).EQ.64.AND.IBLNK.EQ.1.AND.IBLNK2.EQ.1) GOTO 100
      IF(IDECED(I).EQ.64.AND.IBLNK.EQ.1.AND.IBLNK2.EQ.0) THEN
           IN=1
           GOTO 400
      ENDIF
      IF(IDECED(I).EQ.64.AND.IBLNK.EQ.0) THEN
            IBLNK=1
            IN=I
            IBLNK2=1
            GOTO 100
      ENDIF
      IBLNK2=0
      IN=I+1
100   CONTINUE
      IF(ISIGN.EQ.-1.AND.IN.EQ.2) THEN
         IN=1
         GOTO 400
      ENDIF
      IF(IDECPL.GT.0) THEN
           INTLN=IDECPL-1
           ISTRLN=IN-1
      ELSE
           INTLN=IN-1
           ISTRLN=INTLN
      ENDIF
      IF(ISIGN.EQ.-1) THEN
           J=2
           INTLN=INTLN-1
      ENDIF
      IPWR=INTLN-1
      DO 300 I=J,ISTRLN,1
      IF(IDECED(I).EQ.75) THEN
           GOTO 300
      ENDIF
      REALSV=(IDECED(I)-240.)*(10.**(IPWR))+REALSV
      IPWR=IPWR-1
300   CONTINUE
      REALSV=REALSV*ISIGN
400   RETURN
      END
*PROCESS OPT(3)
      FUNCTION INTSTV(STRNG1,IN)
      CHARACTER*(*) STRNG1
      CHARACTER*8   STRNG2
      CHARACTER*1   STRARY(8)
      INTEGER*4     I,J,IN,IDECED(8),IPWR,ISIGN,IBLNK,IBLNK2
      EQUIVALENCE (STRNG2,STRARY)
      STRNG2=STRNG1
      J=1
      IBLNK=0
      IBLNK2=0
      IN=0
      ISIGN=1
      INTSTV=0
      DO 100 I=1,8,1
      IDECEQ(I)= ICHAR(STRARY(I))
      IF((IDECEQ(I).LT.240.OR.IDECEQ(I).GT.249).AND.IDECEQ(I).NE.96.AND
     1.IDECEQ(I).NE.64) THEN
           IN=1
           GOTO 400
      ENDIF
      IF(IDECEQ(I).EQ.96.AND.I.NE.1) THEN
          IN=1
          GOTO 400
      ENDIF
      IF(IDECEQ(I).EQ.96) ISIGN= -1
      IF(IDECEQ(I).EQ.64.AND.IBLNK.EQ.1.AND.IBLNK2.EQ.1) GOTO 100
      IF(IDECEQ(I).EQ.64.AND.IBLNK.EQ.1.AND.IBLNK2.EQ.0)  THEN
          IN=1
          GOTO 400
      ENDIF
      IF(IDECEQ(I).EQ.64.AND.IBLNK.EQ.0) THEN
          IBLNK=1
          IN=I
          IBLNK2=1
        GOTO 100
      ENDIF
      IBLNK2=0
      IN=I+1
100   CONTINUE
      IF(ISIGN.EQ.-1.AND.IN.EQ.2) THEN
        IN=1
        GOTO 400
      ENDIF
      ISTRLN=IN-1
      IF(ISIGN.EQ.-1) THEN
           J=2
           IPWR=ISTRLN-2
      ELSE
           IPWR=ISTRLN-1
      ENDIF
      DO 300 I=J,ISTRLN,1
      INTSTV=(IDECEQ(I)-240)*(10**IPWR)+INTSTV
300   IPWR=IPWR-1
      INTSTV=INTSTV*ISIGN
400   RETURN
      END

F;
```

DATE

ILME