SPECIAL REPORT

PERS 107

G. V. KASS

OPERATING MANUAL FOR THE ACCOUNTING PROGRAM (VERSION 1968) OF THE NATIONAL INSTITUTE FOR PERSONNEL RESEARCH

NATIONAL INSTITUTE FOR PERSONNEL RESEARCH
COUNCIL FOR SCIENTIFIC AND INDUSTRIAL RESEARCH

CSIR SPECIAL REPORT NO.107 UDC 681.3:657
Johannesburg, South Africa. February 1969.

An Accounting Program for the N.I.P.R.

Operation Procedures

G.V. KASS.

Summary : This report is intended to supplement "An Accounting program for the N.I.P.R." Special Report No. 98.

Whereas the above mentioned report gave details of the inputs to, and outputs from the system together with a format of the files used; this report is intended to guide the user of the system to the use of the programs.

-----oOo-----

C O N T E N T S.

DESCRIPTION OF INDIVIDUAL PROGRAMS — PAGE

SUBROUTINES.

OTHER INFORMATION

I. The following is a list of the programs in the system:

		NAME	
17.	Initialise the Table files	SETUP	
10.	Terminal Input Program	IMPUT	
18.	Initialise the Monthly file	SETUPM	
4.	Transfer Well to Monthly Re-initialise Well	CHECK1	
5.	Display Project Names	DISP1	
7.	Display Monthly file	DISP3	
2.	Accumulate times	ACCUM2	
15.	Print Time Spent reports	OUTRUT2	
9.	Reset Acc. Times file for next month	EDIT2	
12.	Initialise the Acc. Times file	INIT2	
1.	Accumulate expenditure	ACCUM1	
14.	Print Project Expenditure reports	OUTRUT1	
8.	Reset Acc. Expenditure file for next month	EDIT1	
11.	Initialise the Acc. Exp. file	INIT1	
3.	Accumulate Suppliers' Record file	ACCUM3	+
16.	Display Suppliers' Record file	OUTPUT3	+
13.	Initialise the Suppliers' Record file	INIT3	+
20.	Create backup for Project/Names file	SPEC2	
21.	Retrieve backup for Project/Names file	SPEC3	
22.	Create backup for Accumulated Times file	SPEC4	
23.	Retrieve backup for Accumulated Times file	SPEC5	
24.	Create backup for Accumulated Expenditure file	SPEC6	
25.	Retrieve backup for Accumulated Expenditure file	SPEC7	
34.	Print Reconciliation Analysis of Project Cost	YELLOW	
30.	Suppliers' Accounts Returns : Commitments	WEEK1	+
31.	Suppliers' Accounts Returns : Payments (Monthly)	WEEK2	*

/4 .. .. ..

		NAME	
32.	Suppliers' Accounts Returns : Cancellations	WEEK3	*
33.	Suppliers' Accounts Returns : Payments (Immediate)	WEEK4	*
26.	Create backup for Suppliers Record file	SPEC8	*
27.	Retrieve backup for Suppliers Record file	SPEC9	*
6.	Display Ranks-Salaries-Votes Vehicles	DISP2	
28.	Print Project Costing report	SUPP1	*
29.	Print Suppliers Records report	SUPP2	*
19.	Select and Sort on the Monthly file	SORT1	

\+ Written but not tested

* Not written.

/5 .. .. ..

BASIC INFORMATION ON PROGRAMS

PROGRAM	SIZE IN DECIMAL	AVERAGE RUNNING TIME	REMARKS
ACCUM1	20,000	4 - 6 Mins	Uses the SORT feature.
ACCUM2	26,000	4 - 5 Mins	
ACCUM3	-	-	Not written
CHECK1	6,000	2 Mins	
DISP1	4,000	2 Mins	
DISP2	6,000	2 Mins	
DISP3	5,000	2 Mins	
EDIT1	4,000	2 Mins	
EDIT2	4,000	2 Mins	
IMPUT	68,000		Depends on how long the user is at the terminal.
INIT1	4,000	½ Min	
INIT2	4,000	½ Min	
INIT3	4,000	½ Min	
OUTRUT1	22,000	3 Mins	
OUTRUT2	14,000	3 Mins	
OUTPUT3	-		Not written
SETUP	4,000	½ Min	
SORT1	24,000	4 Mins	± 250 documents
SPEC2	4,000	1 Min	Replaced by IEBISAM
SPEC3	4,000	1 Min	ditto
SPEC4	4,000	1 Min	
SPEC5	4,000	1 Min	
SPEC6	4,000	1 Min	
SPEC7	4,000	1 Min	
SPEC8	-	-	Not written
SPEC9	-	-	Not written
SUPP1	-	-	Not written
SUPP2	-	-	Not written
WEEK1	6,000	-	Not used at present
WEEK2	-	-	Not written
WEEK3	-	-	Not written
WEEK4	-	-	Not written
YELLOW	8,000	½ Min	

/6 .. .. ..

Program Sizes and Timing

Size

The size given is the size of the linkage edited module rounded up to 200 bytes.

To calculate how much core space is required to run the program, the following must be added together:-

a. The basic program size,

b. The amount of space needed for the I/O buffers. (Note that unless otherwise spacified on the job-control-cards, each data set that is accessed sequentially will have two buffers. Data sets accessed randomly have one buffer.)

c. Overhead for using Indexed Sequential files.

d. Extra core space needed for the Sort program.

Timing

Note that times dealing with indexed sequential files depend heavily on how well or how badly the file is organised. Runs of programs that have to create records in an I.S. file (especially at the beginning of the year) take longer than those runs which merely update existing records.

PROCEDURES.

r of procedures have been drawn up for the more common operations of the system. These are

1. NACCO1P - Call IMPUT for a terminal session.
2. NACCO2P - Call SETUP to initialise the tables - RANKS, CONAMS (INDEX, PRIME, OVERFLOW) and the input buffer INBUF
3. NACCO3P - Call SETUPM to initialise a monthly file MONTH1
4. NACCO4P - Call CHECK1 to update the monthly file MONTH1 from the Well
5. NACCO5P - Call INIT2 to initialise an Accumulated Times file, ACTM1 from the Monthly file MONTH1
7. NACCO7P - Call OUTRUT2 to print the Times reports
8. NACCO8P - Call EDIT2 to advance the label of the Accumulated Times file to the next month.
9. NACCO9P - Call DISP1 to display Project-Names table
10. NACC1DP - Call DISP3 to display the Monthly file MONTH1
11. NACC11P - Call OUTRUT1 to print the Expenditure report.
12. NACC12P - Call ACCUM1 to update the Accumulated Expenditure file ACEX1 from the Monthly file, MONTH1
13. NACC13P - Call INIT1 to initialise an Accumulated Expenditure file ACEX1

/7 .. .. ..

INTERACTION AND ORDER OF RUNNING THE PROGRAMS.

The following flowchart gives the interaction and the order of running the programs.

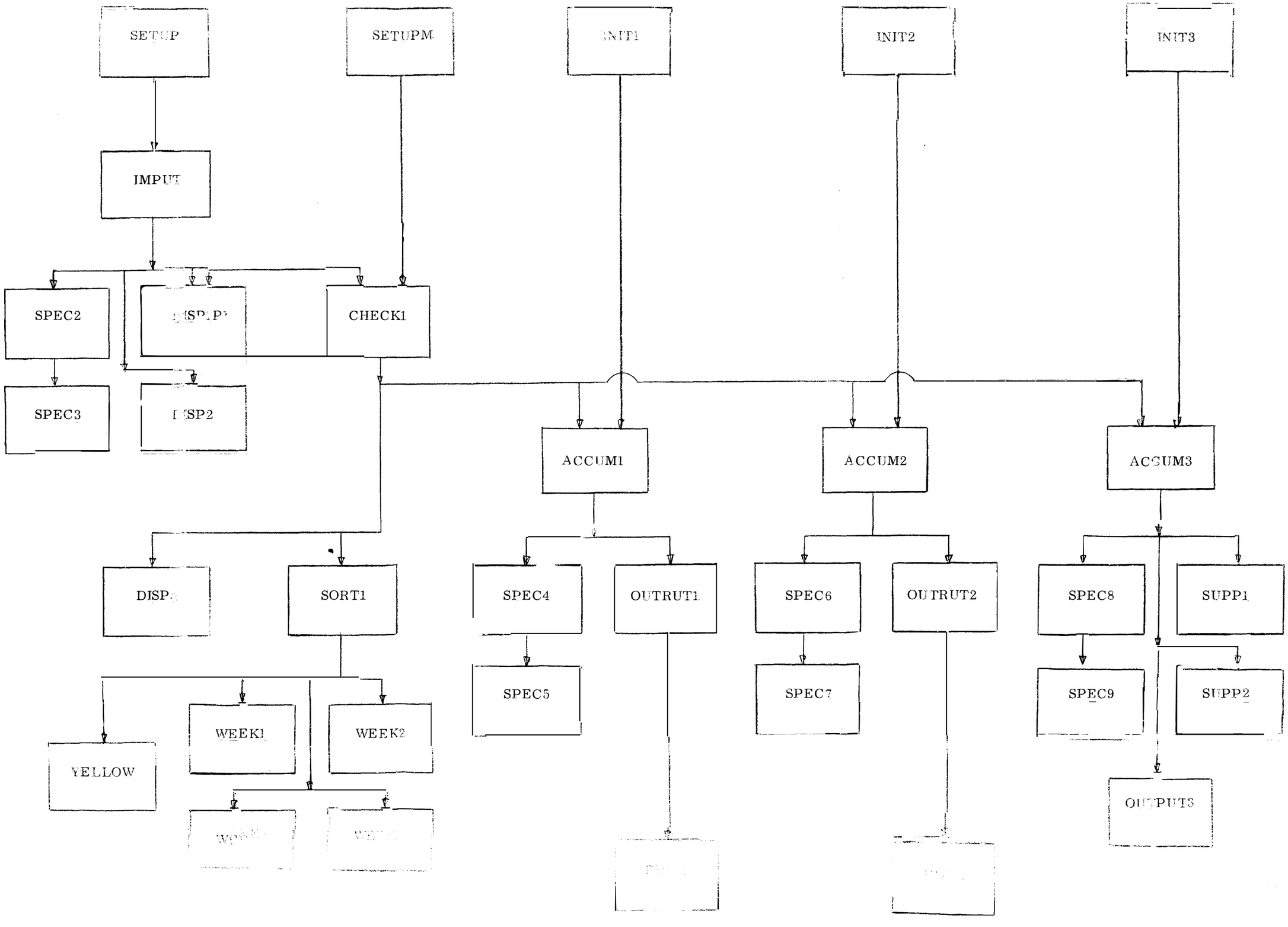
Before a program can be run, it must be preceded by those programs that lead to it.

In some cases, programs could be run in a different order to that displayed on the flowchart. But in these cases, although physically possible (i.e. the programs do not bomb-out), it is logically incorrect or time wasting.

e.g. EDIT1	could be run immediately after INIT1. A different parameter for INIT1 would achieve the same result.
or, CHECK1	need not wait until IMPUT has run (but there would be no entries to clear out of the buffer).

/8 .. .. ..

INTERACTION AND ORDER OF RUNNING N.I.P.R. ACCOUNTING SYSTEM PROGRAMS



II. The following is a list of files used by the system.
(The symbolic name is not fixed, but will be used throughout this report when refering to that file).

RANKS	Seq.(F),1x1128	Table of Vehicles, Votes, Ranks and Salaries.
CONAM	I.S.(FB),17x37	Table of Contracts(Projects) and Names (Staff)
INBUF	Seq.(F),1x260	Input buffer
MONTH	Seq.(VB),10x260	Monthly file
ACTIM	I.S.(FB),12x18	Accumulated Times file.
ACEX	I.S.(FB),171x21	Accumulated Expenditure file
ACSUP	I.S.(FB),34x50	Suppliers' file
SELECT	Seq.(FB),54x67	Temporary file from which to prepare reports etc.
*	Seq.(F),1x80	Parameter Card(s)
SYSOUT	Seq.(F),1x120 or 133	System Output Device

Where Seq. = Sequential file, I.S. = Indexed Sequential
F = Fixed length records, V = Variable length records
B = Blocked records
and MXN = m records of length n in a block

Note: For unblocked records, m = 1

(i) Some programs use workfiles, more specific information will be given where they are used.

(ii) For more information concerning the above files, see Reference 1.

(iii) All references to files within the system are via COBOL I/O routines. Thus, any change made to a file should be reflected in the File Description entries of all programs using that file.

(iv) * Is an unblocked input file from the card reader.

/9 .. .. ..

III. File Usage.

A. When files' back-up copies should be taken.

B. Whether these files should be periodically re-organised from their back-up copies, and under what conditions this should be done.

C. Which programs only read them.

D. Which programs create or modify them in any way.

File	Back-up After:-	Reorganising	Only read by:-	Created or Modified by:
RANKS	IMPUT	No	DISP2	SETUP,IMPUT
CONAM	IMPUT	Yes	DISP1,OUTRUT2,OUTRUT1 SPEC2,SUPP1,SUPP2, SORT1	SETUP,IMPUT, SPEC3
INBUF	IMPUT	No	-	SETUP,CHECK1
MONTH	CHECK1, ACCUM1, ACCUM2, ACCUM3	No	DISP3,SORT1	SETUPM,CHECK1, ACCUM1,ACCUM2, ACCUM3
ACTIM	ACCUM2	Yes	OUTRUT2,SPEC4	INIT2,EDIT2, ACCUM2,SPEC5
ACEX	ACCUM1	Yes	OUTRUT1,SPEC6	INIT,EDIT1, ACCUM1,SPEC7
ACSUP	ACCUM3	Yes	DISP4,SPEC8,SUPP1, SUPP2	INIT3,ACCUM3, SPEC9
SELECT	-	-	YELLOW,WEEK1,WEEK2, WEEK3,WEEK4	SORT1
*	-	-	SETUPM,DISP,ACCUM2, EDIT2,INIT2,ACCUM1, OUTRUT1,EDIT1,INIT1, ACCUM3,INIT3,SORT1	-
SYSOUT	-	-	-	All programs except IMPUT.

/10 .. .. ..

Back-up of files.

Back-up of files should be taken as soon as they have been modified.

A file that has just been initialised need not be saved as it can just as easily be re-initialised.

Three generations of day-to-day back-up files should be kept in case an error takes some time to be discovered.

Retrieval of back-up files.

If any program that modifies a file, halts in the middle of execution or does not complete properly (not due to a program error), those files being modified should be scrapped, and a re-run performed on the latest back-up copy.

Back-up and retrieval of files can be modified as suits system requirements.

Re-organisation of Files.

Indexed Sequential files must be re-organised periodically because:

(i) The overflow areas may become full; and
(ii) Processing becomes inefficient when there are many entries in the overflow area.

While these files are being built up, the files should be re-organised after every run that updates them.

After the files have been built up, and there are not too many additions and deletions to the file, the files need only re-organising periodically (e.g. Once a month).

/11 .. .. ..

The user's attention is drawn to the fact that IBM Operating System automatically keeps statistics pertinent to the I.S. file. Information is obtainable in the Format 2 DSCB as to

(i) No. of records tagged for deletion.
(ii) No. of random references to overflow records other than the first overflow record.
(iii) No. of records in the prime date area.
(iv) Status indicators to (a) Last block full
(b) Last track full
(v) No. of tracks remaining in the independent overflow area.
(vi) No. of records in the overflow area.
(vii) No. of cylinder overflow areas that are full.

amongst other information.

Reference: IBM S/360 O.S. System Control Blocks (C28-6628).

For further information regarding space requirements for the data set both in the prime data area and overflow areas, the user is referred to :-
IBM S/360 O.S. Supervisor and Data Management Services (C28-6646).

/12 .. .. ..

IV. Format of the Program Descriptions.

A. Program Name

This is the program identification.

B. Purpose

An outline of what the program does.

C.I. Parameter Cards

(i) A list of parameter cards required, and rules concerning validity.

(ii) Options available and how to invoke them.

(iii) Future options that could be considered for implementation.

II. Input

Which files are used as Input.

D.I. Messages

(i) A list of all messages and output possible -

(a) Correct and Informative messages:- What they signify

(b) Error messages
What they mean, cross-reference to Recovery Procedures (below) if necessary, position in program from where they originate (Procedure name) if applicable, and what program action is taken.

(ii) Suggestions for expanding or limiting messages as necessary.

(iii) In the case of reports:- what features may be easily modified.

II. Output.

Which files are used for Output and when and how the files are changed.

/13 .. .. ..

E. Recovery

(i) How to recover from specific or known errors (where a message is given).

(ii) Recovery from other errors.

F. Files.

A list of files used by the program, under the headings :-

DDNAME, Organisation, Disposition, Symbolic Name

where DDNAME : Name given on DD card to identify file.

Organisation : Seq. = Sequential, I.S.=Indexed Sequential.
I.S.(Seq) = Indexed Sequential accessed sequentially.

Disposition : NEW = Created by this program
SHR = Only read by this program
OLD = File exists but will be changed (updated) by this program
MOD = Program adds records to the end of this file
* = Input stream

Symbolic Name: Standard Name given to the file.

If workfiles are used, their characteristics will be discussed. If any files need saving after this run, they will be noted.

Also whether files need shaking up at regular intervals.

/14 .. .. ..

Other information.

Other information about a program such as its size, estimating running time and any special features are shown on pages 5 and 6.

/15 .. .. ..

ACCUM1.

A. Name of Program : ACCUM1.

B. Purpose.

Update the Accumulated Expenditure file (ACEX) from the Monthly file by :-

(i) Changing or adding to budgeted values.

(ii) Adding to Accumulated Expenditure To Date.

(iii) Adding to and subtracting from committed figures.

(iv) Changing and transferring projects from one code to another (i.e. cancelling them, renaming them, or changing them from division to division).

(v) Generating total figures within the file for:-

a. Total over each project.

b. Total by account headings within each division.

c. Total over each division.

C. (I)Parameter Card.

The program requires a single parameter card

Format

Col. 1 = a blank; the usual parameter:-
Accumulation will take place from the Monthly file for :-

i. Blocks within the Monthly file that have the same identification month as the Accumulated Expenditure file.

ii. Only blocks which have not yet been used for updating the Accumulated Expenditure file before, will be used.

= A Overrides(i) above
i.e. No check will be made to see if the month of the Accumulated Expenditure file corresponds to the month of the block in the Monthly file used for the update.

/16 .. .. ..

= I Overrides (i) and (ii) above i.e. No check will be made on month, or whether the block has already been accumulated, when considering a block in the Monthly file for updating.

The above are subject to the following options.

Cols. 2 - 4 = nnn Only blocks from nnn to be used for updating [Usually left blank].

5 - 7 = ppp No blocks after ppp must be used for updating.
Those columns left blank show that there is no upper limit.

8 - 80 Ignored.

In the numeric items (Cols. 2 - 4, 5 - 7), the required limits (if specified) must be right-adjusted. Leading zeros may be omitted.

(II) Input

The following files are used for input

1. MONTH
2. ACEX
3. *
4. The Sort Library
5. 5 Workfiles.

D.(I) Messages.

i. List of all messages.

a. Correct messages.

ACCUM1 BEGUN - DATE dd-mm-yy TIME hh.nn.ss

The first message of the program, it shows that program ACCUM1 has started. This run took place at the given date and time.

dd = day
mm = month
yy = year
hh = hour
nn = minutes
ss = seconds

ACCUM1 FINISHED.

This is the last message of the program, and shows that the program successfully reached the last statement. A printout of the Accumulated Expenditure file (ACEX) should now be taken to see if that file has been correctly modified. (Use program OUTRUT1 q.v.)

AND WILL BE PROCESSED.

Ablock that passed the criterion of being in the request range (See message 'PREPARE FOR BLOCK bbbb), has been found suitable for updating. The records that follow in the block will be used for further processing.

AND WILL NOT BE PROCESSED.

The block did not pass all the tests for updating. The records that follow in the block will therefore be ignored.

BLOCK OF MONTH mm FOUND.

A week-month document (i.e. the beginning of a block of entries) has been found in the Monthly file. This block has entries pertaining to the month mm.

END OF SORT 1.

An informative message to show that the first sort has been completed, if this message is not immediately preceded by the message 'WHY ON SORT 1 ..... (qv. in Error messages, below).

ENTRY nnnn DOCUMENT CODE dddd.

The document dddd, which is the nnnnth document from the beginning of the block (this number is thus 1 less than the entry number written in the file), is to be used in updating the Accumulated Expenditure file, if applicable.

NO

or YES

Informative message from the sort procedure to show if FILE1 is being used as an input file for this data pass, or not.

OPEN xx.....xx

The Monthly file has been opened. The label record says xx xx

PREPARE FOR BLOCK bbbb

The block just started, is a block that lies in the range specified in columns 2 - 7 of the parameter card (See C(I))

SORTS IT GOOD EH.

Informative message to say that a pass through the data is beginning for the SORT.
The number of times this message appears, reflects the number of passes made through the data for the Sort.
If it appears many times (e.g. more than 10), either a new sorting procedure is required, or the working space for the present procedure should be enlarged.
(See SORTS-IT-GOOD section in Part J of program SORT1).

UPDATE FILE OF MONTH mm YEAR yy

The Accumulated Expenditure file (ACEX) was found to have a label record with a month of mm in the year yy.

YES

See under 'NO'

N Where N is some number. This is a marker to show that certain stages in the program have been reached. It has only de-bug value.

/19 .. .. ..

b. Error messages.

ACFILE HAS NO CONTROL RECORD.

The Accumulated Expenditure file (ACEX) did not contain its label record. The program terminates after this.

User Response.

i. Check whether the initialisation program (INIT1) worked correctly.

ii. Check whether this program (ACCUM1) or the edit program (EDIT1) failed to work previously, and no corrective action was taken.

iii. Check the job control language cards, for incorrectly name files.

Make appropriate correction and continue as in 'Recovery' (E) below.

ERROR - NO PARAM CARD.

No parameter card (see C(I)) was provided.

User Response. Supply a parameter card, and re-run.

INVALID DOC CODE-IGNORED.

The above mentioned document (See message 'ENTRY nnnnn DOCUMENT CODE dddd), was found, and the program has no coding to handle it. This document is therefore being ignored.

User Response. Supply coding for this new document, and/or appropriately alter the GOTO in paragraph PRO-CESS in section EDIT-EM. Depending on whether or not this document should have been ignored (as it was) accept or reject this program run.

NON-NUMERIC UPDATE LIMITS

The parameter card was found to contain a non-numeric item in columns 2 - 7. These columns should contain the update limits, or be left blank. See 'Parameter Card' C(I).

User Response. Correct the error on the parameter card, and re-run.

TOO MANY ENTRIES.

When assigning a unique key to each entry to be updated in this run, the program maximum of 65,534 keys was exceeded.

ddddddddd WHY ON SORT 1 .....

The first sort in the program was not completed successfully. The code returned was ddddddddd (TALLY). Refer to IBM S/360 OS: COBOL(F) Programmer's Guide (C28-6380) 'Using the Sort Feature' for further information. The program continues.

User Response. If the reason for the error can be found, take the necessary action. Otherwise, continue as for a bomb-out (See Recovery (E) below).

Note: That the program continues after this error, but this is just to see what happens. Unless the program worked properly despite the error, this message should be treated as a bomb-out.

(II) Output.

The following files are used for Output.

1. SYSOUT
2. MONTH(Update)
3. ACEX (Deletion and update)
4. 5 Workfiles.

/21 .. .. ..

E. Recovery.

1. Where the error is due to the parameter card, correct the error and re-run.
Note that in this case, no file has yet been modified, and thus it is unnecessary to use back-up copies here.

2. Program error. Correct the program error and continue as for system errors below.

3. System and other errors.
When the program halts in the middle of execution, or a subsequent printout of the files show that the program did not work properly, the Accumulated Expenditure file (ACEX) becomes garbage, and is no longer usable, and must be scratched.

It is therefore advisable to scratch both the Accumulated Expenditure file (ACEX) and the Monthly file (MONTH) and to restore them from the latest back-up copy, which should reflect how they stood at the beginning of the run.
After retrieving these files, re-run ACCUM1.

If it is not possible or otherwise inconvenient to scratch the Monthly file (MONTH) as well, continue with the old Accumulated Expenditure file (ACEX) using suitable options on the parameter card.
In this case, the option 'I' must be chosen, and delimiting blocks given, so as not to re-use documents that have already been accumulated in the Accumulated Expenditure file.
Examination of the last printout of the Monthly file (before the bomb-out), should assist in

/22 .. .. ..

choosing the required blocks.

F. Files.

DDNAME	Org.	Disp.	Symbolic Name
MONTH	Seq.	Old	MONTH
ACCUM	I.S.	Old	ACEX
ACCUS	I.S.(Seq.)	Old	ACEX
IN	Seq.	*	*
SORTLIB	Seq.	SHR	-
SYSOUT	Seq.	SYSOUT	SYSOUT
SORTIN	Seq.	NEW	-
SORTOUT	Seq.	NEW	-
SORTWK01	Seq.	NEW	-
SORTWK02	Seq.	NEW	-
SORTWK03	Seq.	NEW	-

Where SORTLIB is the standard sort library, and the other SORT xxx are temporary files used by the sort feature and the program.

Reference : "Using the Sort Feature" in IBM S/360 OS: COBOL(F) Programmers' Guide (C28-6380)

Save the following files after a successful run:

ACEX
MONTH

SORTIN and SORTOUT each define a separate file with records with blocksize 650 bytes.

SORTWKxx are files defined by the SORT program. If greater sorting efficiency is required, additional files may be defined (i.e. SORTWK04, SORTWK05, etc.)

ACCUM2.

A. Program Name : ACCUM2.

B. Purpose.

Update the Accumulated Times file (ACTIM) from the Monthly file by :-

i. Changing or adding to budgeted values.
ii. Adding to monthly figures.
iii. Changing and transferring staff from one code to another (i.e. division to division and/or rank to rank and/or name to name):-
iv. Generating total figures within the file for:-
    i. Total over each staff member.
    ii. Total for projects within a division.
    iii. Total for a division.

C.(I) Parameter Cards.

The program requires a single parameter card.

Format:

Col. 1 = a blank; the usual parameter:- Accumulation will take place from the Monthly file for:

i. Blocks within the Monthly file that have the same identifying month as the Accumulated Times file.
ii. Only blocks which have not yet been used for updating the Accumulated Times file before, will be used.

= A Overrides (i) above.

i.e. No check will be made to see if the month of the Accumulated Times file corresponds to the month of the block in the Monthly file used for the update.

/24 .. .. ..

= I Overrides (i) and (ii) above.
i.e. No check will be made on month or whether the block has already been accumulated when considering a block in the Monthly file for updating.

The above are subject to the following options.

Cols. 2 - 4 = nnn Only blocks from nnn to be used for updating.
(Usually left blank)

5 - 7 = ppp No blocks after ppp must used for updating.
Blank means that there is no upper limit.

8 80 Ignored, can be used for comments.

For numeric items (Cols. 2 - 4, 5 - 7), the required limits (if specified) must be right-adjusted.
Leading zeros may be omitted.

(II)Input.

The following files are used for input.

1. MONTH
2. ACTIM
3. *
4. One workfile. (If the Monthly file has documents transferring staff names)

/25 .. .. ..

D.(I) Messages.

i. List of all messages.

a. Correct messages.

EMPTY-M and EMPTY-P and EMPTY-D
For information.
The 'M','P' or 'D' resp. table is going to be emptied.
This occurs either when the table becomes full, or when there is nothing more to update from the Monthly file.

ØØFINISHEDØØ

The Update has been completed. A printout of the Accumulated Times file (ACTIM) should now be taken to see if that file has been correctly modified.
(use program OUTPUT2 q.v.)

FIND-M
For information.
An entry has just been used in an update, and the program is trying to find space in order to prepare totals.

ØLOOP1Ø
For information.
A record has just been read from the Monthly file.

OPEN xxx xx
The Monthly file with label record xxx .... xx has been opened.

PREPARE FOR BLOCK xxxx
Block humber xxxx has just been encountered, and will be used for updating.
(i.e. the block humber falls between those requested on the parameter card).

/26 .. .. ..

b. Error Messages.

ACFILE HAS NO CONTROL RECORD.
The file ACTIM does not contain a Date Control record.

Causes : i. The Accumulated Times file was not initialised correctly.
ii. The DD card for the file points to an incorrect file.
iii. A program error elsewhere caused deletion of this record which should have a key of low-values.

Recovery : Se E. 2 below.

NO MONTHLY LABEL REC.
The file MONTH does not contain a Label record as its first record.

Causes: i. The Monthly file was not initialised correctly.
ii. The DD card for the file points to an incorrect file.
iii. The file does not exist.

Recovery: See E. 2 below.

(II) Output.

The following files are used for output.

1. SYSOUT
2. MONTH (update)
3. ACTIM (deletion of update)
4. One workfile (If the Monthly file contains documents transferring staff names).

/27 .. .. ..

E. Recovery.

1. On a data interrupt (0C7), check whether the parameter card has numeric entries only (0 - 9, or space) in columns 2 - 7. If not, correct card and re-run.

2. On an error in label or control records.
   i. Correct the cause of the error.
   ii. Re-run the program.

Note that when one of the specified error messages are given, no file has been modified, and thus it is unnecessary to use back-up copies.

3. Program error. Correct the program error and continue as for a System bomb-out, below.

4. System error.
   When the program halts in the middle of execution, the Accumulated Times file (ACTIM) becomes garbage and is no longer usable, and must be scratched.

   It is advisable to scratch both the Accumulated Times file (ACTIM) and the Monthly file (MONTH), and to restore them from the latest back-up copy, which should reflect how they stood at the beginning of the run. After retrieving these files, re-run ACCUM2.

   If it is not possible, or otherwise inconvenient, to scratch the Monthly file (MONTH), restart with the old Accumulated Times file (ACTIM) using suitable options on the parameter card.
   In this case, the option 'I' must be chosen, and delimiting weeks given, so as not to re-use documents that have already been accumulated in the Accumulated Times file. Examination of the last printout of the Monthly file (before the bomb-out), should assist in choosing the required weeks.

/28 .. .. ..

F. Files.

DDNAME	Org.	Disp.	Symbolic Name.
MONTH	Seq.	Old	MONTH
ACCUM	I.S.	Old	ACTIM
ACCUS	I.S.(Seq.)	Old	ACTIM
IN	Seq.	*	*
SYSOUT	Seq.	SYSOUT	SYSOUT
WKFILE	Seq.	NEW	-

Save the following files after a successful run:-

1. ACTIM
2. MONTH.

WKFILE defines a file with records with blocksize 180 bytes.

A. Name of program : ACCUM3.

(Note: This program has been written but not yet tested.)

B. Purpose.

Update the Suppliers' Record file (ACSUP).

Updates use:-

(i) Sub-blocks in the Monthly file.

(ii) Corrections as specified in the parameter cards.

C.(I) Parameter Cards

The program uses two kinds of parameter cards:-

Card 1.

Cols. 1	=	blank =	update from all unused sub-blocks in the Monthly file not yet been used.
	=	I	Ignore the markers in the Monthly file which shows whether sub-blocks have been used before.
2 - 80			Reserved for future use.

Note that the program does not as yet include the option of using specific sub-blocks between certain units for the update.

Card 2. (2,3), (4,5), (6,7)

These pairs of cards show what specific records in the Suppliers Record file (ACSUP) should contain.

Whatever is specified in the field of these cards, is transferred directly to the file.

Columns 13 to 28 of the first card in each pair, contain the key of the record to be modified(or created or deleted).

Note : When a record is modified, none of the original figures in the file (except the key) is retained.

/30 .. .. ..

Cards 2,4,6 ..... (The first card of the pair).

Col.		
	1	'1' to identify the card as the 1st of the pair.
	2-7	Date of order (ddmmyy)
	8-12	Supplier's number.
	13-18	Order number.
	19	Extension to the order number.
	20-22	Vote number.
	23-25	The coded division number.
	26-28	The coded project number.
	29-35	Serial number of commitment.
	36-42	Estimated cost (Amount Committed in whole Rands)
	43-48	Date of the Invoice (ddmmyy)
	49-55	Invoice number.
	56-62	Serial number of the payment list.
	63-67	Serial number of the cheque.
	68-76	Amount paid (to two decimal places, do not specify the decimal point.)
	77-88	Ignored.

Cards 3,5,7, ... (The second card of the pair)

Col.		
	1	'2' to identify the card as the 2nd of the pair.
	2	if = 'D' this record is to be deleted. if ≠ 'D' the record is to be updated.
	3-9	Serial number of cancellation.
	10-69	Reserved for future use.
	70-80	Check-sum to two decimal places (do not specify the decimal point.) This is a check-sum of all numeric entries on the pair of cards, excluding the numeric identifiers in column 1 of each card.

Note: That numeric entries should be right adjusted. Leading zeros may be omitted.

/31 .. .. ..

(II) Input.

The following files are used for input.

1. MONTH
2. ACSUP
3. *

D.(I) Messages.

1. List of all messages.

(a) Correct Messages.

1. ¢ ACCUM3 FINISHED. ¢
   The last message of the program. It shows that program ACCUM3 has finished successfully.
2. ACCUM3 STARTED yyyyyyyyy ddddddddd
   The first message of the program. It shows that program ACCUM3 has been entered and is about to begin. This run takes place on day dddddd in the year yyyyyyyy
3. ¢ CLOSING DOWN. ¢
   The parameter card file (PARAMCDS) contains no more updates for the program. The program will now terminate.
4. END OF UPDATE FROM MONTHLY FILE.
   The Monthly file (MONTH) contains no more blocks for updating. The program will now update from the parameter cards.
5. OPEN xx xx
   The Accumulated Supplier's file (ACSUP) with label record xx ..... xx has been successfully opened, and the label record read.
6. ccccc ..... cc HAS BEEN DELETED.
   The pair of parameter cards ccc .... cc specified that a record should be deleted. This has been done.

/32 .. .. ..

7. ccc ..... cc HAS BEEN UPDATED.
The pair of parameter cards ccc .... cc has been used to update (or create) the specified record.

(b) Error Messages.

1. ACCUM3 TERMINATED.
The program (ACCUM3) was terminated because there was NO PARAMETER CARD q.v.

2. ACSUP NOLABEL REC.
The Supplier's Record file (ASCUP) did not contain a label record.

User Response.

(i) Check and correct the job control cards.

(ii) Check that the file was initialised correctly.

3. AMOUNT PAID DUPLICATE - RECORD IGNORED.
A document (displayed beneath this message) in the Monthly file, specified the amount paid, which the Supplier's Record already had a record of. The document is ignored.
See Recovery (E) 1.

4. BAD CHECKSUM - ENTRIES IGNORED. YOURS Y MINE M
The checksum given on a pair of parameter cards Y, did not agree with that calculated the program M.
The pair is ignored.

5. CARD OUT OF ORDER ccc cc
A parameter card that was supposed to be the first of a pair did not have a '+' in column 1.
The card was cc cc
The card is ignored.

/33 .. .. ..

6. COMMITTED AMOUNTS DO NOT MATCH - WARNING xxx ... xx
A document (displayed beneath this message) which specified a payment differing from the amount originally committed xx xx
The document is accepted.

User response.
No further action is required on the Supplier's Record file (ACSUP), but if the payment complete the transaction the difference in the commitments should be entered or cancelled so that the total commitment for this order is now zero.

7. DATE OF INVOICE DUPLICATE - RECORD IGNORED.
A document (displayed beneath this message) in the Monthly file, specified the date of an invoice which the Supplier's Record already had. The document is ignored.
(See Recovery (E) II).

8. DATE OF ORDER DUPLICATE - RECORD IGNORED.
The document in the monthly file (displayed below this message) specified a date of order, and the Supplier's Record already had this information. The record in the Monthly file is ignored.
(See Recovery (E) I).

9. ESTIMATED COST DUPLICATE - RECORD IGNORED.
A document (displayed beneath this message) in the Monthly file specified a commitment, which the Supplier's Record already contained. The record in the Monthly file is ignored.
(See Recovery (E) I).

/34 .. .. ..

10. INVALID CHARACTER ccc cc
The parameter card ccc ... cc contained non-numeric item in what was supposed to be a numeric field.
The card is ignored.

11. NO PARAMETER CARD.
No parameter card was given to the program.
The program terminates.
User response.
Supply card and re-run.

12. NO PART 2 OF CARD.
The last parameter card was the first of a pair, the second part of the pair did not follow.
The first part is ignored and the program terminates.

13. NOT CANCELLING CORRECT AMT - RECORD IGNORED eeeeee
A document (displayed beneath this message) in the Monthly file, tried to cancel a committed amount. However, the amount that it tried to cancel was not the same as that in the Supplier's Record (Shown as eeeeee).
User response.
(i) If eeee is zero then the amount cancelled has not yet been committed. This should be done, unless the cancellation is an error.
(ii) If the amount was incorrect, then correct file with a parameter card. As the system totals reflect the incorrect amount, the actual amount cancelled by this document must be committed, and the cancellation document then re-entered with the correct amount.

/35 .. .. ..

14. ORDER NOT YET COMMITTED - WARNING.
A document (displayed beneath this message) contained a payment on an order which had not been committed. The document was accepted.
User response.
Commit the order.

15. SERIAL OF CANCELLATION DUPLICATE - RECORD IGNORED
A document (displayed beneath this message) in the Monthly file specified a serial number of a cancellation, which the Supplier's Record already contained.
The document is ignored. (See Recovery (E) I)

16. SERIAL OF COMMITMENT DUPLICATE - RECORD IGNORED.
A document (displayed beneath this message) in the Monthly file specified a serial number of a commitment, which the Supplier's Record already contained. The document is ignored.
(See Recovery (E) I)

17. SUPPLIER'S NO. DUPLICATE - RECORD IGNORED.
A document (displayed beneath this message) in the Monthly file, specified a new supplier's number, but the Supplier's Record already had this number.
The document is ignored.
(See Recovery (E) I).

18. THIS IS NOT PART 2 OF CARD - TRY AS PART I, ccc...cc
A parameter card that was supposed to be the second part of a pair did not have a '2' in column 1.
The card was ccc ..... cc.
The first part of the pair is ignored.
The program continues by assuming that part 2 was missing, and the card, just read, is the 1st part of a new pair.

/36 .. .. ..

19. doc., ent.,*div/proj.*,vote, (ord., extn.),
sup., amt., est., inv.
Where doc = 3, 11 or 12.
This is a display of the document for which
the program gave an error message.

Code.

doc	= document number	extn	= order extension number
ent	= entry number	sup	= supplier's code
div	= division	amt	= amount paid
proj	= project	est	= amount committed
vote	= vote	inv	= invoice number
ord	= order number.		

20. doc., ent.,*div./proj.*, vote,(ord., extn.), day/month/year,
rest.
Where doc = 2.
This is a display of the document for which the
program gave an error message.
Abbreviations same as above; and
day/month/year = date of order.

21. doc., ent., * div/proj * vote (ord,extn) ser.rest
Where doc = 8 or 10.
This is a display of the document for which
the program gave an error message.
Abbreviations same as above; and
ser = serial number of the document.

(II) Output.

The following files are used for output.

1. MONTH
2. ACSUP
3. SYSOUT.

E. Recovery.

1. Where an error message has been given to say
that a document has been ignored:-

/37 .. .. ..

If it was ignored because that document was presented a second time to the program (ACCUM3), then no further action is necessary.

If it was ignored because the document occurred twice in the Monthly file, then although the supplier's file (ACSUP) is correct, the system as a whole is 'out-of-balance' and suitable documents should be entered to cancel the superfluous amounts.

2. Program error. Correct the program error, and continue as for System errors, below.

3. System and other errors.
Recovery is the same as for the other ACCUM programs.
Basically this is :-
   1. Scratch the monthly (MONTH) and Supplier's (ACSUP) file.
   2. Retrieve the latest copy of these above files.
   3. Re-run the program.

The notes that apply to the other ACCUM programs, apply here too.

F. Files.

DDNAME	Org.	Disp.	Symbolic Name.
MNTHFL	Seq.	Old	MONTH
ACSUP	I.S.	Old	ACSUP
IN	Seq.	*	*
SYSOUT	Seq.	SYSOUT	SYSOUT

Save the following files after a successful run:-

1. ACSUP
2. MONTH

/38 .. .. ..

CHECK1

A. Program Name : CHECK1

B. Purpose.

Read the Output Well and transfer entries to the Monthly file. Print the document numbers and the entry numbers of documents transferred. Re-initialise the Output Well.

C. (I) Parameter Cards.

None.

(II) Input.

The following file is used for input.

1. INBUF

D. (I) Messages.

i. List of all messages.

(a) Correct messages.

AT THE END.

The Output Well has been completely scanned, and all entries in it have been transferred to the Monthly file (MONTH). The Well will now be initialised.

DOC dddd ENTRY-NO. nnn

Document ddd which has an entry-number of nnn will be transferred from the Well to the Monthly file.

(Note : Document 0254 which has an entry-number of 1 is the Week-Month document which originally had a document number of 52).

FINISHED.
The last message of the program. It shows that the program has finished.

(b) Error Messages.

BAD DOC NO = dddd RUBS = rrrr
Document number dddd was found in the Well, but no coding has been provided in the program to handle it. rrrr indicates the record in the Well in which this document number occurred.

The program terminates, and the Well is not re-initialised. (See Recovery (i)(E) below)

THE WELL REMAINS UNCHANGED
Because of the error 'BAD DOC NO = dddd RUBS = rrrr' q.v. the Output Well is left as it was. and is not re-initialised.
(See Recovery (i)

(II) Output.
The following files are used for output.
1. MONTH
2. SYSOUT

E. Recovery.
(i) If an unrecognised document appeared in the Well, two cases can be distinguished.

(a) It is a new document which has been accepted by IMPUT, but this program cannot handle it. In this case (i) modify the program (See below on how to include a new document), (ii) retrieve the latest copy of the Monthly file, and (iii) re-run this program.

/40 .. .. ..

(b) The document number is an error.
In this case, either

1. a file that is not the output buffer has been given to the program (check the job control cards and re-run);

2. the error has occurred earlier through some program incorrectly modifying the Output Well. In this case, the Output Well should be scratched, and if any documents have been added to the Monthly file, the Monthly file should also be scratched. Re-start by re-entering all documents (displaying the Monthly file will show what is missing).

3. This program (CHECK1) has been incorrectly modified. In this case continue as in (a) above.

(ii) In the case of a system error.

A. If the message 'AT THE END' has not been given, then

1. Scratch the present Monthly file(MONTH)

2. Retrieve the latest back-up copy of the Monthly file (MONTH).

3. Re-run this program.

B. If the message 'AT THE END' has been given, then

1. It is wise to check that the Monthly file (MONTH) really has been modified correctly. Do this by displaying the file (See program DISP3).

2. Re-initialise the Output Well,either
(a) by bringing back the original (or first) back-up copy, which should be the initialised version of this file, or,
(b) by running this program (CHECK1) with a dummy month file, or,
(c) by running the initialising program SETUP with dummy files for RANKS and CONAM.

The above are given in order of preference, but the one that it is most convenient to use, should be used.

C. If in doubt about which procedure should be followed,

1. Scratch the output well (INBUF)
2. Retrieve the latest back-up copy of the output well (INBUF)
3. Continue as in A (the message 'AT THE END' had not been given), above.

F. Files.

DDNAME	Org.	Disp.	Symbolic Name
INBUF	Seq.	Old	INBUF
MNTHFL	Seq.	MOD	MONTH
SYSOUT	Seq.	SYSOUT	SYSOUT

Save the following file after a successful run :

MONTH.

/42 .. .. ..

DISP1.

A. Program name : DISP1

B. Purpose.

Display the Project-names file (CONAM)

C.(I) Parameter Card.

The program requires a single parameter card.

Format.

Col. 1 - 80 Ignored.

This is a dummy included to facilitate subsequent modification of the program.

(II) Input.

The following files are used for input.

1. *
2. CONAM.

D.(I) Messages.

(i) List of all messages.

(a) Correct messages.

ø FINISHED ø

This is the last message of the program. It shows that the program has successfully listed the whole of the Project-names file.

ø START ø

This is the first message of the program. It shows that the program has been entered and is about to begin.

There are two types of records in the file, which are listed in the following manner:-

1. A Project record:

dddd/ppppppp**aaaa*bbbb*ccc.....cc

where

dddd is the encoded form of the division to which this project belongs(it is 7,000 less than the actual division number).

ppppppp is the project number.

aaaa is the same as dddd above. It is the first part of the division project code.

bbbb is the code that has been assigned (by the program IMPUT) to this project within this division.

ccc...cc is the alpha-numeric description of the project.

2. A Staff Name Record.

rrrrr * dddd * kkkk * nnn....nn ** aaaa * bbbb*ccf

where

rrrr is the alpha-numeric rank of the staff member.

dddd is the division to which the staff member belongs.

kkkk is the code given to the rank rrrrr

nnn...nn is the alpha-numeric staff member's name.

aaaa is the same as ddd. It is the first part of the staff name code.

bbbb is the same as kkkk. It is the second part of the staff name code.

cc is the third part of the staff name code. It is the first two characters of the staff member's name (nnn...nn)

f is the last part of the staff name code. It is assigned to this staff name by the program IMPUT. It is not usually a printable character, and thus appears in the listing as a blank.

(b) Error messages.

None.

(II) Output.

The following file is used for output.

1. SYSOUT.

E. Recovery.

1. After a system error:-
   Re-run the program.
2. On a program error due to the program:-
   Correct the program and re-run.
3. On an error due to a damaged file:-
   a). Scratch the file that caused this error.
   b). Retrieve the latest copy of the project-names file (CONAM).
   c). Repeat all updates that are missing from CONAM.
   d). Re-run the program (DISP1)

F. Files.

DDNAME	Org.	Disp.	Symbolic Name.
IN	Seq.	*	*
CONAMS	I.S.(Seq.)	SMR	CONAM
SYSOUT	Seq.	SYSOUT	SYSOUT.

Save the following files : None.

DISP2.

A. Program Name : DISP2.

B. Purpose.

Display the Sequential Table file (RANKS)
The following are printed out :-

(i) The Vehicle-Rate table.

(ii) The Votes table

(iii) The Rank-Salary table.

(iv) The Rank-Randcode table.

C.(I) Parameter Card.

None.

(II) Input.

The following file is

1. RANKS.

D.(I) Messages.

(i) List of all messages

(a) Correct messages.

DISP2 FINISHED.
The last message of the program
DISP2 has successfully listed the file RANKS.

DISP2 STARTED.
The first message of the program. It shows that the program DISP2 has been entered and is about to begin.

The display consists of the following table reports.

(a) The Vehicle-Rate table.

Form:-

(1) (2)
VEHICLE RECORD. RECORD CODE I.S. H
(3) (4)

NO.	RATE	NO.	RATE	NO.	RATE	.....	NO.	RATE	NO.	RAT
1	**.*	2	**.*	3	***		9	**.*	10	**.
11	**.*	12	**.*	13	**.*		19	**.*	20	**.
								(4)		
251	**.*	252	**.*	253	**.*		259	***	260	***
NO.	RATE	NO.	RATE	NO.	RATE		NO.	RATE	NO.	RAT

/46 .. .. ..

Notes:

(1) The heading for the Vehicle Table.

(2) The record code (H). Any character other than 'H' is an error.

(3) The vehicle number (from 1 to 254, though printed to 260) and the rate in cents per mile for that vehicle.
There are 10 such columns, i.e. each row contains 10 such entries.

(4) When a vehicle number is not valid, or has not been entered into the file, then the rate entry contains asterisks.

(b) <u>The Votes Record.</u>

Form:

```
    (1)                                (2)
VOTES RECORD.          RECORD CODE IS V
 (3)              (4)
VOTE  CD  VOTE   CD  VOTE CD VOTE CD .....VOTE CD VOTE CD
  1   20   2     20   3  ***   4  20          9  20  10  25
  '        '         '         '              '       '
  '        '         '         '              '       '
 991 220   992 240  993 240 994 ***          999 ***  0 ***
VOTE  CD  VOTE  CD  VOTE CD VOTE CE .....VOTE CD VOTE CD
```

Notes.

(1) The heading for the Votes Table.

(2) The record code (V). Any code other than a 'V' is an error.

(3) There are 10 such columns, each containing a vote (4) as defined in Appendix B of Reference 1.

(4) A vote that has no account code, cannot be used, and its code is printed out as asterisks. Note that the salaries vote should be given an account code of 1.

(c) The Rank Record.

Form:-

```
     (1)                                        (2)
RANK-SALARY  RECORD.   RECORD  CODE IS  R

     (3)
RNK SALRY  RNK SALRY  RNK SALRY ..... RNK SALRY
rrr *.**   rrr *.**   rrr *.**          rrr *.**
 '    '     '    '     '    '            '    '
 '    '     '    '     '    '            '    '
rrr *.**   rrr *.**   rrr *.**          rrr *.**
   (4)          (5)
RANK TABLE HAS   nnn  ENTRIES.
    (6)
ccccc  rrr  ccccc  rrr   ccccc  rrr ..... ccccc  rrr
     '           '           '                '
     '           '           '                '
ccccc  rrr  ccccc  rrr   ccccc  rrr       ccccc  rrr
```

Notes.

(1) The heading for the Rank Salary record.

(2) The record code (R). Any code other than 'R' is an error.

(3) There are 10 such columns. Each one contains a rank code and the staff rate in *.** rands per hour.

(4) The heading for the Rank to Rank Code Table.

(5) nnn is the number of entries in the table.

(6) A line of entries. Each line contains 8 entries consisting of the Alpha-numeric rank and the rank code which is the same as the rank code in (3)

Note : For further information see Reference 1 (especially Appendix D.)

(b) Error Messages.

None.

(II) Output.

The following file is used for output.

1. SYSOUT.

/48 .. .. ..

E. Recovery.

1. After a system error:-
   Re-run the program.
2. On a program error due to the program:-
   Correct the program and re-run.
3. On an error due to a damaged file:-
   a). Scratch the file that caused the error.
   b). Retrieve the latest copy of the sequential table file (RANKS)
   c). Repeat all updates that are missing from this file.
   d). Re-run this program (DISP2).

F. Files.

DDNAME	Org.	Disp.	Symbolic Name.
RANKS	Seq.	SHR	RANKS
SYSOUT	Seq.	SYSOUT	SYSOUT.

Save the following files : None.

DISP3.

A. Program name : DISP3

B. Purpose.

Display the contents of the Monthly file (MONTH).

C.(I) Parameter Cards.

None.

(II) Input.

The following file is used for input.

1. MONTH.

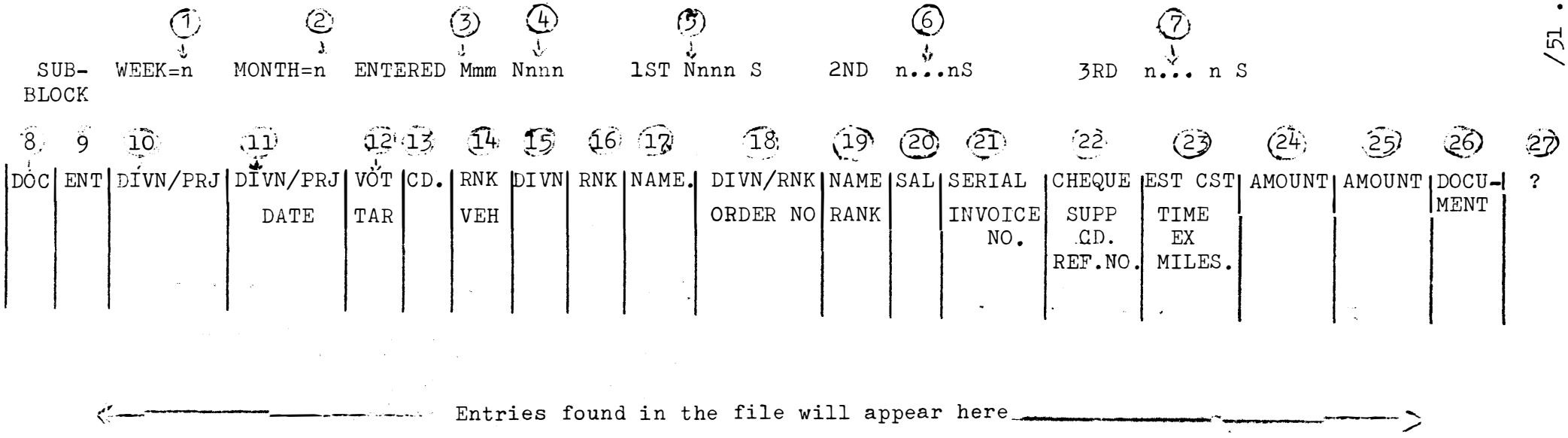
D.(I) Messages.

(i) List of possible messages.

(a) Correct messages.

The printout of the Monthly file.

See following pages.


SUB-BLOCK
WEEK=n
MONTH=n
ENTERED Mmm Nnnn
1ST Nnnn S
2ND n...nS
3RD n... n S
8 DOC
9 ENT
10 DIVN/PRJ
11 DIVN/PRJ DATE
12 VOT TAR
13 CD.
14 RNK VEH
15 DIVN
16 RNK
17 NAME.
18 DIVN/RNK ORDER NO
19 NAME RANK
20 SAL
21 SERIAL INVOICE NO.
22 CHEQUE SUPP CD. REF.NO.
23 EST CST TIME EX MILES.
24 AMOUNT
25 AMOUNT
26 DOCU-MENT
27 ?
Entries found in the file will appear here

18. Either (i) For documents 20, the "new staff name" as in (15), (16) or (ii) the order number for those documents that specify an order number.

19. Either (i) For document 20, the "new staff name" as in (17), or, (ii) the alpha-numeric rank in document 40 (amending ranks). Otherwise blank.

20. For document 40 (amending ranks), this salary in rands per hour if specified (eg. 2.00 rands/hr.). Otherwise blank.

21. The serial number or invoice number if the document specifies one of these. Otherwise blank.

22. The cheque number or Reference number or supplier's code if the document specifies one of these. Otherwise blank.

23. The estimated cost or the time spent or the miles travelled if the document specifies one of these. Otherwise blank.

24. The amount spent (or time or miles converted to rands), if the document specifies one of these.

25. The 'total' amount spent in documents 3, 11 and 12. Otherwise blank.

26. Alpha-numeric description of the document.

27. An asterisk will be placed in this column, if a document is found which the program cannot display.

(b) Error Messages.

* in column headed '?' See (27) above.

/54 .. .. ..

(II) Output.

The following files used for output.

1. SYSOUT.

E. Recovery.

1. After a system error.
   Re-run the program.
2. On a program error or if a new document has been introduced to the system which the program cannot handle :-
   Correct the program and re-run.
3. On an error due to a damaged file:-
   a). Scratch the file that caused the error.
   b). Retrieve the latest copy of the Monthly file (MONTH).
   c). Re-enter all data missing from the file.
   d). Re-run the program (DISP3).

F. Files.

DDNAME	Org.	Disp	Symbolic Name
MNTHFL	Seq.	SHR	MONTH
SYSOUT	Seq.	SYSOUT	SYSOUT

Save the following files : None.

EDIT1

A. Program name : EDIT1

B. Purpose.

Set the current month amounts in the Accumulated Expenditure file (ACEX) to zero.
If the parameter card contains a valid month number and a valid year, this is inserted into the file; otherwise the month is incremented by 1. (The year is incremented by 1 when the month equals 12, the month then being reset to 1).
Any record that is found to contain only zero amounts is deleted from the file. (Note that in a record, if all the fields are zero except the monthly amount, then this deletion does not apply. However, the monthly amount will be reset to zero in this program-run, and thus if no further changes are made to the record, it will be deleted in the next program-run).

C.(I) Parameter Cards.

The program requires a single parameter card.

Format.

Col.	1 - 2	Usually blank (the default option. See B 'Purpose' above) Otherwise the month that should be inserted into the file (and this is done only if cols. 1 - 2 contain a valid month, this field contains the year that should be inserted into the file.
	7 - 80	Ignored.

Note that in order for a number to be valid in this context, each of columns 1 to 6 should contain a numeric punch. (A blank is thus invalid).

/56 .. .. ..

(II) Input.

The following files are used for input.

1. *
2. ACEX.

D.(I) Messages.

(i) List of possible messages.

(a) Correct messages.

ǿ BEGIN EDIT1.ǿ

The first message of the program. It shows that program EDIT1 has been entered and is about to begin.

ǿ END EDIT1. ǿ

The last message of the program. It shows that program EDIT1 has finished.

MONTH CHANGED TO mm, YEAR TO yyyy.

After finding a valid month and year entry on the parameter card (i.e. valid as numbers, and month in range 1 to 12), the program will modify the label record of the Accumulated Expenditure file (ACEX) to reflect a month of mm and a year of yyyy.

MONTH WILL BE INCREMENTED TO SHOW NEXT MONTH.

Because columns 1 - 6 did not contain valid month and year entries (eg. they were left blank), the label record will have its month incremented by 1 to show the next month.

When the month is equal to 12, it is reset to 1, and the year is incremented by 1.

(b) Error messages.

ERR 1ST READ

The Accumulated Expenditure file (ACEX) contained no records. It thus did not contain

a label record. This is an error in the file.

User response.

(i) Check the DD card (especially the DDNAME and (DISP)

(ii) Check the file to see if it was initialised correctly with program INIT1.

(II) Output.

The following files are used for output.

1. ACEX
2. SYSOUT.

E. Recovery.

1. Where an error message is given, take the action advised and continue as for (ii) below.
2. On 'standard' errors:-
   (i) Retrieve the most recent copy of the Accumulated Expenditure file (ACEX); and
   (ii) re-run the program.

Note: Beware of running the program twice on the same file ACEX, and using the default option on the parameter card to obtain automatic incrementing of the month. This multiple usage can result in unplanned multiple incrementing of the month in the label record.

F. Files.

DDNAME	Org.	Disp.	Symbolic Name.
ACCUM	I.S.(Seq.)	Old.	ACEX
IN.	Seq.	*	*
SYSOUT	Seq.	SYSOUT	SYSOUT.

Save the following file after a successful run :

ACEX.

EDIT2.

A. Program name : EDIT2

B. Purpose.

This sets the current month amounts in the Accumulated Times file (ACTIM), to zero.

If the parameter card contains a valid month number and a valid year, this is inserted into the file; otherwise the month is incremented by 1. (The year is incremented by 1 when the month equals 12, the month then being reset to1). Any record containing only zero times, is deleted from the file. If the month figures, **previously contained values, the deletion is** only done the following month).

i.e. A record containing zeros in the 'budgeted' and 'accumulated till now' fields, and non-zero amounts in the monthly field, will first have the monthly field reset to zero, and only on the following run will this record be deleted.

C.(I) Parameter Card.

The program requires a single parameter card.

Format.

Col.	1 - 2	Usually blank, otherwise: the month that should be inserted into the file and Cols. 3 - 6 must contain a valid year.
	3 - 6	Usually blank, otherwise: if Cols. 1 - 2 contain a valid month, this field contains the year that should be inserted into the file.
	7 - 80	Ignored.

/59 .. .. ..

Note that in order for a number to be valid in this context, each of columns 1 to 6 should contain a numeric punch.
(A blank is thus invalid).

(II) Input.

The following files are used for input.

1. *
2. ACTIM.

D.(I) Messages.

(i) List of possible messages.

(a) Correct messages.

ø BEGIN EDIT2. ø

This is the first message of the program, and shows that program EDIT2 has been entered.

øEND EDIT2. ø
This is the last message of the program, and shows that program EDIT2 has finished.

MONTH CHANGED TO mm, YEAR TO yyyy.
Because Columns 1 - 6 of the parameter card contained valid numeric entries (and month between 1 and 12), the month of mm and a year of yyyy.

MONTH WILL BE INCREMENTED TO SHOW NEXT MONTH.
Because columns 1 - 6 of the parameter card did not contain the valid month and year entries (e.g. they were left blank), the label record will have its month incremented by 1 to show the next month. When the month is equal to 12 it is reset to 1, and the year is incremented by 1.

/60 .. .. ..

(b) Error messages.

ERR 1ST READ

The accumulated times file (ACTIM) contained no records, and thus did not contain a label record.

This is an error in the file.

User response:

Check the DD card, and also the file to see if it was initialised properly with program INIT2.

(II) Output.

The following files are used for output.

1. ACTIM
2. SYSOUT.

E. Recovery.

1. Where an error message is given, take the action advised and continue as for (ii) below.
2. On 'Standard' errors -
   (i) retrieve the most recent copy of ACTIM; and
   (ii) re-run the program.

Note : Beware of running the program twice on the same file ACTIM and using the default option on the parameter card to obtain automatic incrementing of the month. This multiple usage can result in unplanned multiple incrementing of the month in the Label record.

F. Files.

DDNAME	Org.	Disp.	Symbolic Name
ACCUM	I.S.(Seq.)	Old	ACTIM
IN	Seq.	*	*
SYSOUT	Seq.	SYSOUT	SYSOUT

Save the following file after a successful run:-

ACTIM.

/61 .. .. ..

IMPUT.

A. Program name : IMPUT.

B. Purpose.

For entering all input that is to be processed by the System.
It receives, checks and edits on-line input, giving completion or error messages where necessary.

Input may be documents for the Monthly file, or for updating the Table files (which is done immediately) or for special instructions to programs.

C.(I) Parameter Cards.

None.

(II) Input.

The following files are used for Input.

1. RANKS
2. CONAM
3. INBUF
4. An on-line terminal (remote)

Description of input from the terminal.

General.

Input takes the form of documents, and is in free-format. A certain amount if editing carried out by the program. (See Reference 1 for full details).

a. General Document Format.

Prompt given by Terminal	Entry.
D	Doc-No. ,document identifying information [,...]]

a. General Document Format (Cont.

Prompt given by Terminal	Entry.
I	information 1 [,information 2...] [=checksum]
I	.. .. ..
I(last entry)	information 1 [,information 2...] = checksum {mandatory on last entry
I	[= document checksum] }

Note : Updates to System tables, carry no checksums.

Prompt - D = Document header entry.
I = Individual item.

(For further prompts, see Section on Output D.I. below).

Note : In "Special Report No.98" the description of the input formats for documents showed certain items within a brace { }

Items within the brace are "individual items" to be entered after prompt 'I' above.
Items before the brace are the document number, and document indentifying information.

b. Additional documents not described in Special Report 98

1. Entering Password.

Terminal Prompt	Entry.
D	

Notes : (i) There are no individual entries.
(ii) This document can only be entered when requested by the program.
(iii) There is no by-passing this document. Once it is requested, a valid password must be given within 3 tries, otherwise the program terminates.

The program recognises 2 passwords (which are set up initially as PASSWORD1 and PASSWORD2, but can be changed by the user with document 51, see below).
When this program is started, a password is requested.

If the 1st password (originally PASSWORD 1) is given, all the 'ordinary' or day-to-day documents can be entered but no updating tables may be performed. If a document for updating the tables is presented, the password document will again be requested. Some password must be given (or the program will terminate). If the '1st level' password is given here, the update document will be rejected, and a new document requested. If the '2nd level' password is given, documents for updating system tables as well as ordinary documents may be presented during the remainder of the run.

If the 2nd password is given at the start, no other password will be requested during the run.

(See Chapter on Security, if the Passwords are irretrievably forgotten).

2. Changing Passwords.

Terminal Prompt	Entry.
D	
I	

Note : (i) No end of document sign ($) is necessary.

(ii) Each password must be less than 30 characters, and should not contain commas (,) nor end with a question mark (?).

(iii) The two passwords should be different.

3. Week-Month Doc.

Terminal Prompt	Entry.
D	52, week, month [, = checksum]
[S	= checksum (if not entered on previous line)]

At the beginning of a run this document is requested after the password document.
If entered in the middle of a run, the entry-number counter for entries in the Monthly file is reset to 1.
This document causes an internal document to be generated with a document number of 254 and a length of 55 bytes:-

Internal entry:-

Byte 1 = Document number = 254
2 - 3 = Entry Number = 1
4 - 5 = Week (as given on document)
6 - 7 = Month(as given on document)
8 - 23= Date and Time of Accumulated Times Update.
34 - 43= Date and Time of Accumulated Expenditure Update.
44 - 53= Date and Time of Supplier's File Update.
54 - 55= Reserved.

/65 .. .. ..

4. Close all Files.

Terminal Prompt	Entry
D	53

Note : (i) There are no indivudual entries.

(ii) When completed the message ⌀ ALL FILES CLOSED.⌀ is given.

This document should be used at intervals to ensure that updates are completed, especially when the computer is not behaving too well.

A. 5. The Job Times Documents.

Terminal Prompt	Entry
D	[1, (or 21, 22, 23, 24 or 25)] { [division number] = [checksum] }
I	[rank] [name] [division], {proj-code / project} time]
	[, division, {Proj-code / project} time] = checksum]
I(last entry)	[rank], [name], [division {proj-code / project} time]
	[, division, {proj-code / project}, time] = checksum
I	= [document checksum]

Note : (i) Each indivudual entry must have at least 2 commas, to be valid.

(ii) A sub-entry of : [division, {proj-code / project}, time] cannot be split from one line to the next.

(iii) The sub-entries (div, proj, time) may be optionally separated by slashes (/) instead of commas(,). Mixed separators may be used.

(iv) The following Form of entry is permissible

I Rank, Name,
I ,, division, project, time, ect.
This allows the rank and name to be checked before continuing with the entry of times.

/66 .. .. ..

The rank and name however, must still be delimited by commas(,) not slashes (/)

(iv) In documents 24 and 25, the project-code is to be used instead of the project. [Where 1 = project-code = 254]

Checksums on Input.

(i) The first checksum on a document must include the document number and any identifying information (unless specified by default).

(ii) No more than one checksum per line is allowed, and it must be the last entry on that line, (except for the end of document symbol (ø))

(iii) No document may end before a checksum has been given, (except the update and program documents which have no checksum i.e. documents 20, 30, 40, 41, 42, 50, 51 and 53).

(iv) A checksum of zero is not recognised by the program.

(v) The numbers entered are to be added to the checksum as they appear. An exception is the date which appears as dd/mm/yy (3 numbers), but is added to the checksum as: ddmmyy (1 number).

Note that : 1/1/69 is added as 10169 (not 1169),i.e. each of the 3 original numbers contribute 2 digits to the original number.

Default Entries.

When entering a document heading (in reply to a prompt of D), the document number and the document indentifying information can be specified by default by leaving the appropriate field between commas blank. (i.e. Although not specified by values, they must still be there positionally). This does not apply to a date or order no., which cannot (at the moment) be specified by default.

'Default' means that the last value of the field will be used.

That is, a default document means :- I want the same document as last time;

or a default cheque number means :- I want the same cheque number as Last time, etc.

Numbers specified by default are not used in checksums.

For individual entries (i.e. in reply to a prompt I) in documents 1, 21, 22, 23, the rank and/or name may be specified by default by leaving the corresponding fields blank.

Special Entries

The following are for cases of emergency.

(i) # delete the previous entry in the input buffer. This does not effect the checksum which will include the deleted entry.

2 main uses are :-

(i) To delete an individual item just entered for a 20 or a 30 document.

(ii) By repeated use, to 'scratch' the input buffer (in core)

/68 .. .. ..

A message will follow this symbol, either to say that the last entry was deleted, or that there were none to be deleted.

(ii) ½£ force end of document, even though a checksum has not been given.
Entries in the buffer not yet transferred to the Output Well are lost.
This is to be used when a document cannot be completed, even with a dummy incorrect checksum.

(iii) % force termination of program, no matter what is being entered.
All files are closed in the normal way.

To be used as a last resort if the program starts producing nonsense although in that case it may be safer to ask the computer operator to cancel the job.

It could also be used for an emergency shut-down, if there is no time to complete what is presently being done.

D.(I) Messages.

(i) List of possible messages.

All messages printed out on the terminal by the program are enclosed by dollar signs '$'.

Prompts.

The prompt is a single character printed out by the program at the beginning of a line, before handing control over to the console user. It indicates what the program is expecting.

D for Document : enter a document number and its identifying information or * for end of session.

I for Individual item or $ for end of document.

R for Repeat : repeat the previous entry. This is preceded by a message indicating why the previous line was rejected.

S for Sum : enter a checksum and only a checksum.
(Not to be confused with the message "PLEASE GIVE CHECKSUM" followed by an 'I' prompt) which occurs when :-
(i) the previous entry needs a checksum;
(ii) the Input buffer is full and no further entries can be accepted until it is emptied to disk, which can only be done when the verifying checksum is given.

for Hash : there was a transmission error. This has same meaning as 'R'.

/70 .. .. ..

(a) Correct Messages.

Messages given on the console for information are :-

(i) Heading and Termination messages.

(ii) Program requests.

(iii) File usage.

(iv) Program action taken under special circumstances.

1. ∅ ALL FILES CLOSED ∅

Given in response to document number 53, when the request has been carried out.

2. ∅ DISC BUFFER POSITIONED.∅

This message is issued when opening the Well on disk, and after giving an indicative dump of the contents of the Well (See ∅ ENTRY NO. xxxx FOUND.∅). If there were entries in the Well, this message shows that the place to enter the next document has been found.

3. ∅ DOC CHECKSUM RESET TO ZERO ∅

When adding item checksums to form a document checksum (this is done, whether or not a document checksum is given), the field containing the total overflowed and was therefore reset to zero. The range of the total is - 9, 999,999,999,999.99 to + 9,999,999,999,999.99 [i.e. $-(10^{13} - 1)$ to $+ (10^{13} - 1)$ to 2 decimal places].

/71 .. .. ..

4. ø DOCUMENT CHECKSUM IS CORRECT. ø
The document checksum at the end of the document agreed with the total calculated by the program.

5. ø ENTER DOCUMENT. ø
The program is ready to accept a new document (in the form of a document number and possibly, other document identification) or an end of run symbol (*).

6. ø ENTER THE INDIVIDUAL ITEMS. ø
The program has accepted a document number and other document identification, and individual items or an end of document symbol (ø) may be given, provided the previous entry contained a checksum.
Note : An end of run symbol (*) will not be accepted until the end of document symbol (ø) has been given first.

7. ø ENTRY NO. xxx FOUND. ø
Whenever the Well or disk is opened an indicative dump of its contents is taken. xxx is the entry number and is the same as the 'xxx' of the message ø LAST ENTRY WAS NUMBERED xxx. ø. The entries printed out in this message, either were at the end of a 260 byte block; or were the last entry in the input well.
The last message of this type is followed by the message ø DISC BUFFER POSITIONED. ø q.v.

8. ø O.K. ø
The checksum just entered is correct, all preceding entries have been accepted. This message is usually accompanied by the message ø ENTRY xxxx ø

/72 .. .. ..

9. ∅ LAST ENTRY DELETED. ∅
A hash sign (//) as the <u>first and only</u> entry on a line, the last entry in the input buffer in core was deleted. Note that no attempt is made to reset the checksum to what it was before the deleted entry.

10. ∅ ENTRY xxxx ∅
Items written to the input buffer, are given a sequence number. The number given to the last entry in a block (or in the case of update entries; to each entry) is printed out in this message as xxx.

Note : (i) The week-month document is always numbered 1. It may be written immediately, or deferred, depending on when the week-month document is given.

(ii) When the last entry-number reaches 999, it is reset to 1 and a new week-month document is written at the beginning of this block. (It will be the same as the one at the beginning of the previous block.

(iii) Document number 6 for Computer Charges, causes additional records to be generated for adjusting the budgeted values for the computing account. Thus one item entry from document 6 causes two (2) records to be written into the Well (the 6 document and a 31 document).

/73 .. .. ..

11. ø NEW-NAME IS IN TABLE - WARNING. ø
When entering the document to amend the staff table, (document 20), a 'new' staff name is already in the table (i.e. in file CONAM). The inclusion of the new staff name will not be carried out, but the 'old' staff name, will be deleted.
The document will be entered in the Monthly file as usual.

12. ø NEW CONTRACT IS IN TABLE - WARNING. ø
When entering the document to amend the division project table (document 30), a 'new' project given is already in the table (i.e. in file CONAM). The inclusion of the new project will not be carried out.
Note that if an 'old' project is included in the entry, it will be deleted).
The document will be entered in the Monthly file as usual.

13. ø NIPR ACCOUNTING INPUT SCHEME DATE dd-mm-yy TIME hh.nn.ss GOOD-BYE! ø
The last message printed out by the program. Date and time are similar to the message ending with WELCOME instead of GOOD-BYE and show the time that the console user signed off.
After this message has been given, records that should be written away, are written, and files opened are closed.

The terminal is then 'closed' and the program stops.

14. ∅ NIPR ACCOUNTING INPUT SCHEME DATE dd-mm-yy
TIME hh-nn-ss WELCOME! ∅

The first message printed out by the program.
The date and time : dd = day
mm = month
yy = year
hh = hour
nn = minute
ss = second
gives the time that the program started according to the computer's clock.

15. ∅ NONE TO BE DELETED - IGNORED. ∅
A hash sign (#) as the first and only entry on a line, is a request to delete the last entry in the input buffer in core (See earlier discussion under "Special entries"). If there are no entries in the buffer in core the message follows, and the request to delete is ignored.

16. password 1
password 2
∅ NOW TRY AGAIN. ∅
This is a display of the valid passwords.
password 1 is the 1st level password.
password 2 is the 2nd level password.
The last message indicates that a suitable password should be chosen, and entered as the password document. The above message only appears after the use of a special key.

17. ∅ PLEASE GIVE CHECKSUM. ∅
This is a request for a checksum.
If it is followed by the prompt 'S', only a checksum may be entered. (See 'S' above).
If it is followed by the prompt 'I', then a complete item, optionally followed by a

/75 .. .. ..

checksum, is required. This happens when a document delimiter is given before a final checksum.
(See 'Checksum on Input' above).

18. ∅ PLEASE GIVE PASSWORD DOCUMENT. ∅
This is a request for the password document, (document number 50).
This request is issued
(i) At the beginning of the program.
(ii) After encountering an update document in the input, and the level of the password is not correct for this update (See discussion under "Entering Password").
A valid password document must be given within 3 tries, otherwise the program terminates.

In the case of (ii) above, if the update document was not required, re-entering the lower-level password will cause rejection of the update document, and processing can continue with another document.

19. ∅ RANDOM UPDATE COMPLETED. ∅
The indexed sequential table (CONAM) (Projects and Staff names) has been successfully updated and the file has now been closed.

Notes : (i). An error following this message cannot lose the update.
(ii) At the end of the run, this file (CONAM) should be saved on a back-up tape.
(iii) The file may need 're-organising.'

/76 .. .. ..

20. ∅ RANK IS IN TABLE - CODE WILL BE CHANGED - WARNING. ∅

When entering the document to amend the Rank table (document 40), a new rank (5 alpha-numeric characters) is already in the table. Program action continues as usual, and the code associated with this rank will be changed to the new rank. This message is only for information.

21. ∅ RANK TABLE FULL - DELETE SOME. ∅
When trying to add a new rank (5 character alpha-numeric) it was found that the table was full (only 102 different Ranks may be entered) It must be emptied by deleting ranks that are no longer used.

22. ∅ RANK WILL BE DELETED. ∅
When entering the document to amend the Rank table (document 40), the new code (2nd positional entry) was omitted (or equal to zero). The rank in the table will be deleted. Any salary given with this entry will be ignored.

23. ∅ THANK YOU - NOW PLEASE GIVE WEEK-MONTH DOC. ∅
This follows a correct password.
A valid week-month document (document number 52) must now be entered.
No further processing is allowed until a valid week-month document has been entered.
(See discussion under 'week-month doc'.)

24. ∅ THANK YOU - NOW WE CAN BEGIN. ∅
This signals that valid password and week-month documents have been received and ordinary documents can now be entered.

25. $ UPDATE COMPLETED. $

The RANKS file on disk has been successfully updated and the file has been closed. Any bomb-out following this message should not affect the file, or cause the update to be lost.

Note : At the end of the run a back-up copy of this file should be saved.

/78 .. .. ..

(b) Error Messages.

1. Error messages printed on the terminal.
General.

It is possible for an error to produce an error message not directly associated with the cause.

If an error message does not seem applicable (e.g. BAD DIVISION NO. for a perfectly good division
check for the following:-

(i) Missing comma or slash
Note that '/' in an alpha field does not mean a delimiter.

(ii) Superfluous comma or slash
Note the ',' in an alpha field does mean a delimiter.

(iii) Omission of a field

(iv) Confusion of O numeric with O alpha.

(v) Confusion of 1 and I and l

(vi) A previous message that was not needed (e.g. a request for a checksum, or the rejection of a document).

(vii) A space where there should be none, or no spaces where there should be some.

(viii) Specifying by default (or equal to zero) where the entry had not been specified before, and thus no default value could be assumed.

Note: When checking a line, do not assume that the program will check the entry from left to right. In many cases, it does not.

/79 .. .. ..

The error messages:-

Unless otherwise stated, the line in error is completely rejected, and the program requests re-entry of the line.

1. ¢ BAD ACCNT CODE. ¢
Documents for adjusting figures or budgets in the Expenditure file (documents 31, 32, 33 contain an account code that

(i) used a decimal place
(ii) was not in the range 1 to 254

2. ¢ BAD AMOUNT ENTRY. ¢
(i) A committed amount was found which used decimal places.

(ii) An amount (whether committed or paid) was not in the range -9,999,999.99 to 9,999,999.99 rands.

3. ¢ BAD CHECKSUM - ENTRIES SINCE LAST CHECKSUM IGNORED. ¢
YOURS YYY MINE MMM
The checksum (yyyy) (other than a document checksum) on the previous line differed from that calculated by the program (mmmm).
All entries since the last checksum are ignored.

Note: If the checksum was given at the end of an 'individual items' line, only the "individual items" line is ignored even if the "document header" line has not yet been included in a checksum.

4. ¢ BAD CHECKSUM - REPEAT WEEK-MONTH DOC AGAIN. ¢
The checksum given with the first week-month document (document 52), does not agree with that calculated by the program.

Note: Checksum = 52 + week+month.
The document with its checksum, is rejected and must be re-entered.

/80 .. .. ..

5. ¢ BAD CHEQUE NO. ¢

Caused by (i) Using decimal places where none are allowed.

(ii) Using a default value where no default cheque number is available.

6. ¢ BAD CONTRACT DESCRIPTION. ¢.

When updating the Project table (document 38), the descriptive title for the new project contains more than the allowed maximum of 30 characters.

7. ¢ BAD CONTRACT NO. ¢

Caused by (i) Using decimal places where none are allowed.

(ii) (a) The given project was not in the range 1 to 9,999,999.

(b) or in the case of the special editing documents 24 and 25, the project was not in the range 1 to 254.

(iii) The division did not 'match' with the given project number.

(iv) The contract has not yet been entered in the Division-project table.

(v) In computer charges (document 6), the computer project number 7031/4238 had not yet been given to the system through document 30, or possibly had been deleted.

8. ¢ BAD DATE. ¢

A bad date on a document

The date should be of the form: dd/mm/yy

where:

dd = the day : the range is 1 to 31
mm = the month: the range is 1 to 12
yy = the year: the range is 1 to 99

Note that no decimal places may be used.

9. ∅ BAD DIVISION. ∅

Caused by (i) Using decimal places where none are allowed.
(ii) Division number is not in the range 7000 to 7199.

10. ∅ BAD DIVISION CODE. ∅

The division number in "document identifying information" (Documents 1, 21, 22, 23, 24, 25):-

(i) Specified decimal places;
(ii) was specified by default (zero or blank) but no similar document had been given previously;
(iii) was not in the range 7000 to 7199.

11. ∅ BAD INVOICE NO. ∅

(i) Using decimal places where none are allowed.
(ii) Using a default which was not available.
(iii) Not in the range 1 to 9,999,999

12. ∅ BAD MILEAGE. ∅

Caused by (i) Using decimal places where none are allowed.
(ii) Not in the range -99,999, to +99,999 miles.

13. ∅ BAD ORDER NO. ∅

Caused by (i) Using decimal places where none are allowed.
(ii) Not in the range 1 to 9,999,999.

14. ∅ BAD PASSWORD - NO UPDATE DONE. ∅

Does not occur. Has been replaced by
∅ INCORRECT PASSWORD FOR THIS DOC - DOC REJECTED. ∅

15. ∅ BAD PROJECT ddd/ppppppp. ONLY USED FOR CHECKSUM.∅
When entering staff times (document 1,21,22,23), the project number (ddd/ppppppp) could not be found in the table of Division-Projects. (Any other reason for it being invalid would give the message ∅ BAD CONTRACT NO.∅).

The program therefore uses this project (dddd/ppppppp) only for calculating the checksum.
No entry is generated in the Monthly file.

16. ∅ BAD RANK CODE.∅
A rank code in document 40.
(i) specified decimal places
(ii) was not in the range 1 to 254.
(zero is allowed but this means that the rank should be deleted.)

17. ∅ BAD REFERENCE NO. ∅
Caused by (i) Using decimal places where none are allowed.
(ii) Entering a number not in the range 1 to 99,999.

18. ∅ BAD SALARY.∅
In updating the Ranks - Salary table (document 40), a salary was specified that
(i) Was negative
(ii) Was not in the range 0.00 to 9.99 rands.

19. ∅ BAD SERIAL NO. ∅
Caused by (i) Using decimal places where none are allowed.
(ii) Using a default or zero value where no default value is available.

20. ∅ BAD STAFF NAME. ∅
The staff name is not in the Staff Names table.
Could be due to :-

(i) The person has not yet been entered in the table.

(ii) The name has been misspelt, or incorrect initials have been given.

(iii) The person has a different rank to that given.

(iv) The person belongs to a different division to that given.

21. ∅ BAD STAFF RANK. ∅

Caused by (i) The given rank was not found in the Ranks table.

22. ∅ BAD SUPPLIER CODE. ∅

(i) Using decimal places where none are allowed.

(ii) Using a default, which was not available.

(iii) Using a code not in the range 1 to 99,999.

23. ∅ BAD TIME-SPENT. ∅

A staff times document (1,21,22,23,24,25), has unacceptable staff times data.

Two cases are distinguishable

A. Documents referring to the budget times (21,23,25)

Possible errors are :- (i) Using decimal places where noe are allowed.

(ii) Using a time not in the range - 99,999 to +99,999 hours.

B. Document referring to time spent (1,22.24)

Possible errors are :- (i) Time not in the range -999.99 to +999.99 hours

24. ∅ BAD VEHICLE NO. ∅

Caused by (i) Using decimal places where none are allowed.

(ii) In the case of the Vehicle mileages document (document 7) using a default where one was not available.

(iii) Using a vehicle number not in the range 1 to 254.

25. ∅ BAD VOTE NO. ∅

A vote number

(i) Using decimal places where none are allowed.

(ii) Using a vote number not in the range 1 to 999.

(iii) Using a vote not present in the table votes.

26. ∅ BAD VOTE OR CODE. ∅

In a document amending the vote (document 41):-

(i) Decimal places were wpecified where none are allowed.

(ii) The vote was not in the range 1 to 999.

(iii) The vote code was not in the range 1 to 254.

(Note : If the vote-code is blank or zero, the associated vote is deleted from the table).

27. ∅ BAD WEEK-MONTH DOC. ∅

A week-month document (document 52), not entered the first time

Might have : (i) Decimal places where none are allowed.

(ii) The <u>week</u> outside the range 1 to 54.

(iii) The <u>month</u> outside the range 1 to 12.

/85 .. .. ..

28. ¢ BAD WEEK-MONTH DOC, TRY AGAIN. ¢
A format error in the first week-month document (document 52), of the session.
Format should be :-

52,{/,} Week{/,} Month = checksum.

where Week = the week of entry, with no decimal places in the range 1 to 54.

and Month = the month of entry, with no decimal places, in the range 1 to 12.

A faulty checksum (if given) will produce a different error message.
Note : The Week and Month in the entry
(i) Are not checked for correction.
(ii) Are not checked for mutual consistency. (i.e. 52, 1, 12 is valid for the program although the 1st week does not usually occur in the 12th month).

29. ¢ COMMENCING SHUT-DOWN. ¢
See under message ¢TOO MANY NAMES-BOMB-OUT OCCURS ON ¢ or under message ¢NEW ADDITION MAKES TOO MANY PROJECTS FOR DIVINS.

30. ¢ DISC BUFFER FULL - COMMENCING SHUT-DOWN.¢
The input well on the disk is full, and no more entries can be accepted.
The program terminates.
User response.
Run the program CHECK1 to empty the well into the monthly file, and to re-initialise the buffer.

31. ¢ DOCUMENT CHECKSUM IS INCORRECT. YOURS YYYY MINE MMMM.¢
The user's document checksum, yyyy the sum of all accepted checksums in the document, differs from that calculated by the program, mmmm. This message is for information only, because no program action is taken.

/86 .. .. ..

The last document has, as far as the program is concerned, been completed. The reason for this message should be determined. It may be due to

(i) omitting items

(ii) entering items more than once.

(iii) because the document checksum overflowed and was set to zero - see DOC CHECKSUM SET TO ZERO under 'Correct' messages.

32. ß EXCESS.DEC.PLACES.ß
A numeric entry contained more than 2 significant digits after the decimal point.

33. ß EXCESS DIGITS IN NO.ß
A numeric entry contained more than 13 significant digits.

34. ß EXTRA.ß
A numeric entry contained more than one decimal point.
(Check for a missing comma, or a full-stop instead of a comma).

35. ß GET HELP - CLOSING DOWN.ß
An incorrect password has been given three times; the program terminates.

36. ß INCORRECT NO. OF ENTRIES.ß
Either too many or too few entries on a line.
Check for missing or misplaced commas or slashes.
Note in documents relating to staff times, a line may not end directly after a division number or a project number, but only after a staff name, or a time entry.

37. ß INCORRECT PASSWORD FOR THIS DOC- DOC REJECTED.ß

A document number referring to table updates (documents 20,30,40,41,42,51) was entered when only the lower level password had been given. The higher level password was re-requested, but was not received and the update document is therefore rejected.
(see "Entering Password" above).

38. ß INCORRECT PLACEMENT OF CHECKSUM.ß

A line contains an equal sign (=) which is not the last entry on the line.

39. ß INV CHAR.ß

In a list of numeric entries contained, a character otherthan:

(i) The digits 0 - 9
(ii) A decimal-point
(iii) A comma or equal sign.
(iv) A backspace or Carriage Return
(v) A blank
(vi) A plus (+) or minus (-) sign.

40. ß INV DOC CODE.ß

The document number either

(i) contained decimal places after the point; or
(ii) was not in the range 1 to 99; or
(iii) was specified by default (equal to blank or zero), but was the 1st document entered other than those requested by the program.

41. ß INV MINUS SIGN.ß

A numeric entry contained either (i) more than one minus sign;
or (ii) a minus sign by itself.

/88 .. .. ..

42. ø LINE IGNORED.ø
The last line is ignored, because it ends with the character '?'.

43. ø NEW ADDITION MAKES TOO MANY PROJECTS FOR DVIN.ø
ppp ..... pp.
øCOMMENCING SHUT-DOWN.ø
There is no space available in the section of the Division-project table for project pp .... pp and so a unique code cannot be assigned to the project.
The program terminates.
The program can handle up to 254 different projects referring to any <u>one</u> division.

<u>User response.</u>
Create a dummy division or an alternate division number for the additional projects.

44. ø NO CODING FOR THIS DOC.ø
This message should only occur if coding is inserted into the program to deal with a document number and "document indentifying information", for a new document, but coding for the "individual item entries" is missing or contains an error.
The program resets itself to accept the next document.

45. ø NO NAMES IN FILE - ERROR - COMMENCE SHUT-DOWN.ø
The staff names file (CONAM) contains no names.
Note : At initialisation a dummy name is written into the file so that there should be at least one name.
This is an error, and the program terminates.
<u>User response.</u>

Check whether the correct file has been offered to the program to read.

46. ∅ OWING TO ERRORS -> DISCONTINUE SERVICE.∅
Abnormal termination of the program. Usually accompanied by some other message. Causes are :-
(i) the special symbol '%' has effected an immediate shut-down;
(ii) the wrong password has been given three times in a row;
(iii) program errors in which the program finds itself at a loss for what to do.

47. ∅ PROGRAM CANNOT HANDLE THIS DOC YET.∅
The document number has not yet been defined to the system, or cannot be entered at this point, (e.g. the password document, number 50, were not requested by the program).

48. ∅ PROGRAM ERR-TABLE TOO SMALL - DELETE SOME PROJECTS.∅
The Division-project table is exceeded. Either some projects must be deleted, or the core-allocation for the table must be increased, (This error may occur in other programs of the system).
See section on "Enlarging the project table in core" below.
The program terminates.

49. ∅ PROGRAM MESS - ATTEMPT SHUT-DOWN.∅
An error, either in the program or in one of the files, has been detected. An accompanying message is printed out on the computer system concole.
See "Messages on Computer's System Console" below.
The program terminates.

50. ∅ RANK TABLE FULL-DELETE SOME.∅
(See "Correct Messages" above).

51. ∅ SORRY -WRONG DOC - RETRY.∅
After requesting a password document (document No.50), the program found that the next entry was not a document 50. The entry is ignored and it is not counted as one of the three allowable tries for the password.

/90 .. .. ..

52. ⊄ THINK CAREFULLY AND TRY AGAIN.⊄
An incorrect password has been given. This message appears twice. The third time the message becomes ⊄GET HELP - CLOSING DOWN.⊄

53. ⊄ TOO MANY ENTRIES.⊄
A line of input contained more than 30 entries.

54. ⊄ TOO MANY NAMES - BOMB-OUT OCCURS ON.⊄
nnn ..... nn

⊄ COMMENCING SHUT-DOWN.⊄
There are no slots available in the section of the staff names table into which nn ..... nn must be placed and so a unique code cannot be assigned to the name.
The program terminates.

User response.
Re-arrange the staff name, so as to have a different 2 characters in the first position of the name. e.g. SMITH E. change to E. SMITH or even possibly 'SMITH E.

Note : Given the same division, rank, and first 2 letters of the name, the program can handle from 8 to 254 entries, depending on the order in which they are presented to the program.

2. Messages on the Computer's System Console.

Whenever a logical or "impossible" error occurs, the program does not attempt to recover. If the error refers to a file, check the DSNAME and the DDNAME of the DD card.
Also display the file with the appropriate DISPx program to see if the contents of the file have been incorrectly modified. Finally, the program

must be checked, for incorrect modification of markers and pointers.

Whenever a message appears on the Computer's Console, the message ⊄PROGRAM MESS - ATTEMPT SHUT-DOWN.⊄ is usually given on the user's terminal.

(See above in "Error Messages)
The message on the Computer's System Console has the form -

NIPR MESSAGE n.

Where 'n' is a number indicating the error and its location.
For n =

1. An end of file mark was encountered in the 'ranks' record in the file RANKS,
   [From paragraph CKR2 in Section CK-RANK]

2. An end of table mark was encountered in the 'votes' record in file RANKS.
   [From paragraph CKV2 in Section CK-VOTE]

3. An incorrect identifier was encountered in the 'votes' record in file RANKS.
   [From paragraph CKV3 in Section CK-VOTE]

4. An incorrect identifier was encountered in the 'ranks' record in file RANKS
   [From paragraph CKR3 in Section CK-RANK]

5. Same as n = 4
   [From the first paragraph in Section UP-RANK]

6. Same as n = 3
   [From paragraph U2 in Section UP-VOTE]

7. An end of file was encountered in the file RANKS.
   (This has the same meaning as in the case n = 1 or n = 2, but there is no indication of which record.)

[From the first paragraph in Section READ-RANR. This section is performed while preparing to check 'vehicles', or while preparing to do any updating in the file RANKS.]

8. Undefined.

9. Undefined.

10. Undefined.

11. An incorrect identifier was encountered in 'vehicles' record in file RANKS.
[From the first paragraph in Section CK-VEHI.]

12. Same as n = 11.
[From the first paragraph in Section UP-VEHI.]

(II) Output.

The following files are used for output.

1. RANKS (only for updates)
2. CONAM (only for updates)
3. INBUF
4. An online terminal (remote)
5. The system console (only for error messages)
No DD card required.

E. Recovery.

1. Any error, other than 'standard' errors (e.g. exceeding time, or core, or I/O system), is serious as the program is supposed to be able to handle bad input data. Such an error shows the presence of an error in the program.

2. With 'standard' errors, not due to the program, proceed as follows:-

(i) When not in the middle of a random update (documents 20 and 30) re-run the program.

/93 .. .. ..

(ii) When in an update while the program is 'doing' the update

(a) if the document is being 'entered' so that the update has not started proceed as in (i) above)

(b) otherwise retrieve the latest back-up copy of CONAM and re-run th program.

Note that if any of the documents being entered specify a cancellatio staff leaving or a transfer, then this entry must be entered in the monthly file (otherwise problems will arise during accumulation.)

In both cases, when the program is re-run, an indicative dump will be made of the input buffer. See message

ØENTRY NO. xxx FOUND.Ø in D.(I) (i) (a) above.

Note that in updates to the sequential file RANKS

(i) The update may not have been done although the entries are in the Well.

(ii) If a message was given that the update was completed however, then the update has been performed, although in this case the entries may not be in the Well.

All errors should be recorded, in case subsequent output shows the recovery procedure was inadequate.

/94 .. .. ..

F. Files.

DDNAME	Org.		Symbolic Name.
RANKS	Seq.	Old	RANKS
CONAM	I.S.	Old	CONAM
CONAMS	I.S.(Seq.)	Old	CONAM
INBUF	Seq.	Old	INBUF
LINEOO1	Seq.	Old	

The file LINEOO1 refers to the remote terminal which the program will use. The DD card for this file should specify DISP=(OLD,DELETE) and UNIT=xxx where xxx is the unit number of the terminal as identified to the computer.

Save the following files after a successful run:-

1. INBUF
2. RANKS (If modified by documents 40, 41 or 42)
3. CONAM (If modified by documents 2- or 30)

G. The Functions of Sections of the Program.

CK-RANK (1) Given a 5 character rank in RANK, return a 1 byte code RNK where

1. RANK-C - Computational
2. Filler x (3) High-val if not there)
2. RNK x

This also sets up the Rank - Salary table.

CK-NAME (2) Given a 27 character name in NAME in the form

1. Name
2. Div-name x. N-DIV
2. Rank-name x. N-RNK
2. Name-name x(25) N-N

return 5 bytes in NM where

1. NM
2. Div name x NM-DV
2. Rank name x NM-RANK(High-vals if not there)
2. Name code x(3) NM-CD

/95 .. .. ..

CK-PROJ (3) Given a 5 character project number in PROJ in the form

1 PROJ
2. Pack-Div x
2. Pack-Proj x(4)

return 2 bytes labelled PRJ

1. PRJ -p-
2. Pack-Div x
2. Code-Proj x (High-val not there)

COMPUTATIONAL-3 PICTURE S9(7)

If n = size of table and t = no of elements already in table $r = \frac{t}{n}$ = % full.

The average no. of trials to place a new element in the Project table = $1\frac{1}{1-r}$

and the average no. of trials to find an element = $\frac{1}{r}(1 - r)$

e.g. if r = 90%: to find an element require 2.57, to insert an element 10 trials.

CK-DIV (4) Given a 4 byte Division no., check if first two digits are 70 or 71 and if so return a 1 byte division code in DIV.

If not return a high value.

CK-VOTE (5) Given a 3 byte Vote No. in VOTE return a 1 byte account code in VOT (High-Val if not accepted.)

BUFF (6) Write NREC records of form $\frac{\text{DOC}}{\text{DIV}}$ to its buffer, writing out and reading in when necessary. (Also opening the file and writing the date.)

CK-VEHI (7) Given 9 figure vehicle code in VEHI, check that code is in table

If not, VEH = High-value

If yes, VEH = Low-value, and

VEH-TARIFF (Computational-3 Picture SV999) = rate per mile.

/96 .. .. ..

UP-RANK (8) Given a 5 character rank in RANK and/or a 1 byte code in RNK and/or a 2 byte salary code in SAL (Computational-3 :-

(i) if RANK is given, find in the table, (or insert in the table) its code RNK;

(ii) if SAL is given, change the salary associated with the given RNK.

Note : (Zeros will be accepted if entered with decimal point.) If H1=zero at the end of the update, the table was too full to be updated.

UP-VOTE (9) Given a 3 byte Vote Number in VOTE and a 1 byte account code in VOT, make the Vote entry in the table refer to the account code.

UP-NAME (10) Given 27 character names in NAME and NAME2, return 5 byte codes in NM and NM2 and substitute NAME2 for NAME in the file.

UP-PROJ (11) Given a 5 character project number in PROJ2 and 35 character project title in PROCT2 in the form

```
1 PROJCT2
  2 PROJ2 (5)
  2 DESC2 (30)
```

return 2 byte codes PRJ and PRJ2 and replace PROJ with PROJ2 in Project-names table.

UP-PSWD (12) Change the password to read as it is set up in core.

/97 .. .. ..

UP-VEHI(13) Given a vehicle code VEHI, and a tariff/mile in VEH-TARIFF, insert the latter in the Vehicle table.

DECODE-SCAN(14) Transfer the I-O-Buffer from IOB-ST to WRITE-COUNT, decoding and placing the values in an array from PT2 till PT2-END with a marker preceding each No. to say how many decimal places it contains.
Numbers preceded by = are placed in CKSM. All blanks are ignored.

STRIP-SCAN (15) This transfers the I-O-Buffer from IOB-ST to WRITE-COUNT eliminating backspacing, stripping off leading and trailing blanks, and reducing embedded blanks to 1 embedded blank.
IOB-ST is altered to reflect the edited buffer.

SHIFT (16) This shifts the items in I-O-Buffer, TALLY+1 places to the left, (suitably changing the value of write-count (=write-count-TALLY-1)
[In fact, move m (Tally+2) m(1)]

Methods of Accessing Tabled Information.

In Input Program.			Core Required.
Staff	→	Indexed Sequentially	-
Ranks	→	In Core : Sequential Search	600
Rank → Salary	→	In Core : Direct	512
Projects	→	In Core : Scatter Table (401 entries)	2,807
Votes	→	In Core : Direct	1,000
			4,919 bytes

(Out-put records → In Core : 50)

/98 .. .. ..

Times Report.			Core Required
Staff	→	Sequentially from Disk	-
Projects	→	In Core : Scatter Table (401 entries)	14,837
			14,837

Expenditure Report

Projects - Sequentially from Disk.

Action of Input table.

1. Set up project table for look-up by project given project code.
2. Given project, give project code (or blank)
3. Set up ranks table.
4. Given rank, give rank code (or blank).
5. Given division, rank, name, give staff code.
6. Given rank and rank code; insert in staff table (check for salary entry).
7. Given rank and rate per hour, insert in Ranks table.
8. Given division rank and name, delete insert (or both) in staff table.
   Assign code.
9. Given project, delete insert (or both)in Projects table.
   Assign code.
10. a. Given a record, insert it in a buffer.
    b. Write records out and get a new buffer, when the old buffer is full.

/99 .. .. ..

Formats of Tables.

		bytes	Size
Staff	Division	1	
	Rank Code	1	200 members require 6,400 bytes
	Name	25	300 " " 9,600 "
	Staff Code	5	
		32	
Rank	Rank	5	100 ranks(Eng. & Afr.) require 600 "
	Code	1	
		6	

Rank-Salary 2 x 256 = 512 bytes if the Code rank is used.

Projects	Division	1	
	Project	4	(Packed)
	Code	2	250 projects require 9,250 bytes
	Title	30	30 projects to a logical record = 90 per track.

Votes 1,000 bytes

/100 .. .. ..

Miscellaneous Points.

Suggested organisation.

Keep Staff and projects in the same I.S. file(1 cylinder)
Staff in order of Division, Rank, Name,
and Projects in order of code.

System Constraints.

1. The "Imput Program" may have to be kept on one of the system disks.
2. The program in core must be as small as possible.
3. As little time as possible should be used (i.e. make execution efficient).

Frequency of use of tables.

Input program. Uses:-

Staff : in pseudo-random sequence. Used less than once a minute.

Ranks : in randomly . Used less than once a minute.

Projects: in randomly . Used possibly once every 10 seconds.

Votes : in randomly . Used possibly once every 10 seconds.

Times Report shows:-

Staff : in order by Name code (The same order as the names.

Projects: almost randomly.

Expenditure Report shows:-

Projects: in order by code. (Not quite in order by the projects.)

/101 .. .. ..

INIT1.

A. Program name : INIT1

B. Purpose:

Initialise the Accumulated Expenditure file by loading a label record carrying the current year and month, and a key of low values.

C.(I) Parameter Cards.

The program requires a single parameter card.

Format.

Col.	1 - 2	The current month of the Accumulated Expenditure file (ACEX). (Usually = 04 i.e. April - the beginning of the financial year.)
	3 - 6	The current year of the Accumulated Expenditure file.
	7 - 80	Ignored.

Note : Columns 1 - 6 must contain numeric values.

(II) Input.

The following file is used for imput:-

1. *

D.(I) Messages.

(i) List of possible messages.

a) Correct Messages.

ØØØØ BEGINNING ØØØØ

The first message of the program.

ØØØØ FINISHED ØØØØ

The last message of the program.

ØØØØ 1st RECORD IN ØØØØ

A valid parameter card was read and the label record of the Accumulated Expenditure file (ACEX) was written.

/102 .. .. ..

b) Error Messages

INV KEY AT LOW-VAL

A label record was encountered when writing a new label record on the Accumulated Expenditure file (ACEX), the program terminates.

User response:-

(i) Scratch the Accumulated Expenditure file (ACEX)

(ii) See that DISP=NEW has been coded or is implied on the DD card.

(iii) Re-run the program.

ØØØØ PARAMETERS NOT NUMERIC - TRY AGAIN. ØØØØ

A non-numeric character was encountered in columns 1 - 6 of the parameter card. The program terminates.

Note : (A blank is non-numeric)

(II) Output

The following files are used for output.

1. ACEX
2. SYSOUT

E. Recovery.

1. Where an error message is given, correct the error and try again.

2. On other errors. Check the job control language and correct if necessary. Scratch the file ACEX and re-run the program.

/103 .. .. ..

F. Files.

DDNAME	Org.	Disp.	Symbolic Name
ACCUM	I.S.(Seq.)	NEW	(ACEX)
IN	Seq.	*	*
SYSOUT	Seq.	SYSOUT	SYSOUT

Save the following files : None.

/104 .. .. ..

INIT2.

A. Program Name : INIT2.

B. Purpose

Initialise the Accumulated Times file by loading a label record carrying the current year and month, and a key of low-values.

C.(I) Parameter Cards.

The program requires a single parameter card.

Format:

Col. 1 - 2 The current month of the Accumulated Times File. (Usually = 04 i.e. April - the beginning of the financial year.)

3 - 6 The current year of the Accumulated Times file. (e.g. 1968)

7 - 80 Ignored.

Note : Columns 1 - 6 must contain numeric values.

(II) Input.

The following file is used for input.

1. *

D.(I) Messages.

i) List of possible messages.

a). Correct messages.

$$$$ BEGINNING $$$$

The first message of the program.

$$$$ FINISHED $$$$

The last message of the program.

/105 .. .. ..

ØØØØ 1st RECORD IN ØØØØ

A valid parameter card was read and the label record of the Accumulated Times file was written.

b). Error messages.

INV KEY AT LOW-VAL

A label record was encountered when writing a new label record on the Accumulated Times file.
The program terminates.

User response.

i) Scratch the Accumulated Times file (ACTIM
ii) See that DISP=NEW has been coded or is implied on the DD card.
iii) Re-run the program.

ØØØØ PARAMETERS NOT NUMERIC - TRY AGAIN. ØØØØ

A non-numeric character was encountered in columns 1 - 6 of the parameter card.
The program terminates.

(Note : A blank is non-numeric.)

(II) Output

The following files are used for output.

1. ACTIM
2. SYSOUT

E. Recovery.

1. Where an error message is given, correct the error and try again. (Note : Scratch the file ACTIM before re-running.)
2. On other errors. Check the job-control language Scratch ACTIM. Re-run.

/106 .. .. ..

F. Files.

DDNAME	Org. I.S. (Seq.)	Disp.	Symbolic Name
ACCUM	Seq.	NEW	ACTIM
IN	Seq.	*	*
SYSOUT	Seq.	SYSOUT	SYSOUT

Save the following files : None.

/107 .. .. ..

INIT3.

A. Program name : INIT3.

B. Purpose.

Initialise the Supplier's Record file (ACSUP).

C.(I) Parameter Cards.

The program requires a single parameter card.

Format.

Col.	1	- 33	Aphanumeric information to be loaded as a label in the Supplier's Record file (ACSUP.)
	34	- 80	Ignored.

(II) Input.

The following file is used for input.

1. * .

D.(I) Messages.

i) List of possible messages.

a) Correct messages.

ØØØØ BEGINNING ØØØØ

The first message of the program.

ØØØØ FINISHED ØØØØ

The last message of the program.

ØØØØ 1st RECORD IN ØØØØ

The label record has been written on the Supplier's Record file (ACSUP).

/108 .. .. ..

b) Error messages.

INV KEY AT LOW-VAL

A label record was encountered when writing a new label on the Supplier's Record file (ACSUP).

The program terminates.

User response:-

(i) Scratch the Supplier's Record file (ACSUP)

(ii) See that DISP=NEW has been coded or is implied on the DD card.

(iii) Re-run the program.

(II) Output

The following files are used for output.

1. ACSUP
2. SYSOUT

E. Recovery.

1. Where an error message is given correct the error and try again.
2. On other errors:-
   (i) Check the job control language.
   (ii) Scratch the Supplier's Record file (ACSUP).
   (iii) Re-run the program.

F. Files.

DDNAME	Org.	Disp.	Symbolic Name
ACSUP	I.S.(Seq.)	NEW	ACSUP
IN	Seq.	*	*
SYSOUT	Seq.	SYSOUT	SYSOUT.

Save the following files : None.

/109 .. .. ..

OUTRUT1.

A. Program name: OUTRUT1

B. Purpose

To prepare the Accumulated Expenditure reports from the Accumulated Expenditure file.

C. I. Parameter Cards

Several parameter cards are usually required, one for each 'heading' (account code) to be printed. These cards may be entered in any sequence but no account code may occur on more than one card.

Format

Col. 1 If numeric (=n) then n lines will be skipped before this account heading is printed.
(Otherwise leave blank)

2 - 4 The Account Code right adjusted with leading zeros inserted.

5 - 23 Alphanumeric description of account.

24 If this column has a minus sign (-), then figures associated with this account code are subtracted from totals (otherwise added).
(should be blank except for account headings dealing with income).

28 - 78 Reserve for future use.
Part of this field could be used for an Afrikaans alpha-numeric description of the account.

79 - 80 If these columns contain 'NA' (for Not Add), then this account code will be printed out after the running expenditure for the project or division.
See under Output D I. (i) below
(should be blank, except for income contract and capital expenditure accounts)

/110 .. .. ..

Examples of parameter cards:-

(i) b001bbbbbbbbbbbSALARIESb .... rest blank

This is a parameter card for account code 1.

Note that the leading zeroes in columns 2 & 3 are mandatory

The description ends in column 23. (See Note (i) below)

No other options are required.

(ii) b160bbbbbRUNNINGbINCOME-bb ..... bbNA

This is a parameter card for account code 160 (Cols 2 - 4).

This account code has been called RUNNING INCOME.

Since income is to be subtracted from total expenditure there is a minus sign (-) in column 24.

Since items with this account heading are to be printed out after Total Running Expenditure the letters 'NA' are punched in columns 79 & 80.

Note (i) The account description in columns 5 - 23, may be placed anywhere in this field, and is printed exactly as it is read

(iii) The correspondence between account code, and vote, is given by Document 41.

The only exceptions are the two system defined account codes:

Account code 1 is the Salaries account code. (i.e. All documents dealing with staff time, are automatically given an account code of 1)

Account code 2 is the Computer Costs account code. (i.e. the document dealing with computer costs, is automatically given an account code of 2)

/111 .. .. ..

II. Input

The following files are used for input

1. *
2. CONAM
3. ACEX

D. I. Messages

(i) List of all Possible messages.

a) Correct messages

The Report

(See Sample report on following pages.)

/112 .. .. ..

OUTPUT

PROJECT EXPENDITURE FOR DIVISION dddd TILL MONTH xx xx

	(3) BUDGETED (RAND) (10)	(4) COMMITTED (RAND)	(5) ACCOUNT FOR THE MONTH (RAND)	(6) ACCOUNT TILL NOW (RAND)	(7) BALANCE NOT BUDGETED (RAND)	BUDGETED (RAND)		
PROJECT dddd/pppp	xx...xx							***
(11) aaaaaaaa	xx	xx	xx	xx		xx		
aaaaaaaa	0.0	xx	xx	xx	(13) xx			
(14) TOTAL RUNNING EXPDT	xx	xx	xx	xx		(15) xx		
(16) aaaaaaaa	xx	xx	xx	xx				
aaaaaaaa	xx	xx	xx	xx				
(17) NET TOT EXPENDITURE		xx	xx	xx				
(21)								
PROJECT dddd/pppp	xxx...xx							
aaaaaaaa	xx	xx	xx	xx		xx		
aaaaaaaa	xx	xx	xx	xx		xx		
(18) TOTAL RUNNING EXPDT	xx	xx	xx	xx	xx			
aaaaaaaa	xx	xx	xx	xx				
NET TOT EXPENDITURE	xx	xx	xx	xx				
.....								
PROJECT DIVN/TOTAL	(9)							
aaaaaaaa	xx	xx	xx	xx		xx	xx	(20)
aaaaaaaa	xx	xx	xx	xx		xx	xx	(21) ***
(19) TOTAL RUNNING EXPDT	xx	xx	xx	xx	xx		xx	
aaaaaaaa	xx	xx	xx	xx				
aaaaaaaa	xx	xx	xx	xx				
**********								
THE C/F TOTALS DO NOT AGREE WITH THE CALCULATED TOTALS								
	(22)	xx	xx	xx				

/113

OUTRUT1

- 113 -

(23)

(24) (25)

THERE WERE nnnn ERRORS IN SUM EXPDT REPORTED FOR THE RUN OF MONTH xx xx

END OF REPORT

/114

Explanation

(1) The division number.
In the case of the Institute Total, this field is blank.

(2) Month and year.

(3) The budget in rands for the project under the account spent.

(4) The amount committed against the project in rands under the account.

(5) The amount spent on the project in rands under the account during the month.

(6) The amount spent on the project in rands under the account from the beginning of the financial year up to and including the month of the table.

(7) The accumulated deficit for the project in rands under the account from the beginning of the financial year to the end of the month of the table for the case where there was no budgeted amount.

The accumulated net balance for the project in rands under the account from the beginning of the financial year until the end of the month of the table.
It is calculated as (6) - (3)
Division and project number, or DIVN/TOTAL, or CANCELLED/FINISHED PROJECTS.
An alpha-numeric description of the project.
A line showing in rands the amount spent, committed or budgeted under the account heading aa .... aa for the project ddd/ppp.... If an account heading has not been provided, the heading takes the form:-

/115 .. .. ..

nnnn *** ... ***, where nnnn is the account code.

Warning asterisks
(Shown against balances for budgeted amounts ((11)),

No asterisk = Money is still available
1 asterisk = Overspent by 0 to 10% of the budget
2 asterisks = Overspent by 10% to 100% of the budget
3 asterisks = Overspent by more than 100% of the budget.

(13) A negative balance since no amount was budgeted under this account for the project.

(14) The total running expenditure spent, committed or budgeted for the project dddd/pppp ...

(15) The total balance calculated from the sum of the figures in columns (7) & (8), or as (6) - (3)

(16) A line showing the amounts spent, committed or budgeted under account headings which are not included in running expenditure (These account headings are specified by 'NA' in columns 79, 80 of the parameter card. See under 'Parameter Cards' C.I. above). No. balances are given for these items.

(17) Net total expenditure budgeted, committed or spent.

(18) Another project in this division.

(19) The divisional totals for the accounts for that division.

(20) The proportional balance.
(assuming a constant monthly rate of expenditure of 1/12 of the budget).

/116 .. .. ..

(21) An error message which appears if the figures calculated by the program do not equal those in the accumulated times file (see under "Error Messages" D.I. (i) b below)

(22) The amount calculated by the program.

(23) The last page of the report.

(24) The number of errors that the program found. This means that the error (21) was reported nnnn times

(25) The month and year to which the report refers.

b) Error Messages

INVALID ACCOUNT CODE FOR xxx xxx
The parameter card with account heading xxx ... xxxx contains an invalid account code (Cols. 2 - 4)
i.e. (i) The account code lies outside the range 1 to 254. The program ignores the parameter card.

THE C/F TOTALS DO NOT AGREE WITH THE CALCULATED TOTALS
See (21) in D.I (i) (a)

Cause of this message

(i) The Accumulated Expenditure file (ACEX) was not initialised correctly (See program INIT1)

(ii) The Accumulated Expenditure file (ACEX) was not updated properly (See program ACCUM1)

(iii) There is not enough space allocated for the Accumulated Expenditure file (ACEX).

User response

(i) Scratch the Accumulated Expenditure file (ACEX) as it is no longer usable.

/117 .. ....

(ii) Retrieve the latest back-up copy that produced correct output (Allocating more space, if necessary).

(iii) If possible, retrieve the Monthly file for the same date (not generally possible)

(iv) Re-run all Accumulated (ACCUM1) and Edit (EDIT1) programs from that point. (If (iii) was not possible, or impracticable, the Accumulated (ACCUM1) program may require selective updating, using the appropriate options)

(v) Re-run this program (OUTRUT1).

Certain errors provide a user completion code before the program terminates.
The code numbers have the following meanings.

User Code

12 The number of accounts that specify the option 'NA' in columns 79, 80 of the parameter card, exceeds the program maximum of 15.

16 The number of accounts contributing towards Total Running Expenditure exceeds the program maximum of 15.

20 The number of accounts to be printed out after Total Running Expenditure, exceeds the program maximum of 15.
This Code should not occur, as there would normally be a user code of 12, before the program could return this code.

User Response

Codes 12, 20

On code 12, first check whether an account code is specified on more than one card (This is an error)

/118 .. ..

If it is necessary to enlarge the program work - areas, increase the following arrays to the desired size:-

1. NO-ADD-CODES-1
2. TEMP-STORE-AREA
3. INST-FIGS-NA

and set NO-OF-SAVUP to be equal to this new maximum.

Code 16.

Enlarge the program- work-areas by increasing the following array to the desired size:-

1. INST-FIGS

and set NO-INST-FIGS to be equal to this new maximum

II Output

The following file is used for Output.

1. SYSOUT.

E. Recovery

1. Where an error message or code is given, follow the instructions given under User response.
2. On all other breakdowns, merely re-run the program.

F. Files

DDNAME	Org.	Disp.	Symbolic Name
INFILE	I.S. (Seq.)	SHR	ACEX
SYSOUT	Seq.	SYSOUT	SYSOUT
CONAMS	I.S. (Seq.)	SHR	CONAM
IN	Seq.	*	*

Save the following files: None

/119 .. .. ..

OUTRUT2

A. Program name: OUTRUT2

B. Purpose

Produce, from the Accumulated Times file, monthly reports of time spent on projects by each member of staff.

C. I. Parameter cards

None

II. Input

The following files are used for input.

1. ACTIM
2. CONAM

D. I. Messages

(i) List of Possible messages

a. Correct messages

The Report. See Sample report on following pages.

/120 .. .. ..

OUTPUT 2

TIME SPENT FOR STAFF IN DIVISION dddd (1) TILL MONTH xx ..... (2) xx (Sample output)

(3) (4) (5) (6) (7) (8)

		TIME BUDGETED	TIME TILL END OF LAST MONTH	TIME SPENT THIS MONTH	TIME SPENT TILL NOW	NOT BUDGETED	BALANCE BUDGETED (HOURS)	
(9) rrrrr	(10) nnn...nn							(15)
(11)	dddd/pppp..	xx	xx	xx	xx		xx	***
	dddd/pppp	0.0	xx	xx	xx	xx		
(13)	TOTAL	xx	xx	xx	xx	xx	(14)	
rrrr	nnn.nn							
(16)	dddd/pppp	xx	xxx	xx	xx		xx	
	dddd/ppp	xx	xx	xx	xx		xx	
	TOTAL	xx	xx	xx	xx	xx		
	(10) DIVISIONAL TOTAL.							
	dddd/pppp	xx	xx	xx	xx		xx	
(17)	TOTAL	xx	xx	xx	xx	xx		

(18) **********

(19) THE C/F TOTALS DO NOT AGREE WITH THE CALCULATED TOTALS.

(20) xx

/12 ...

```
ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø
ø                                                                                 ø
ø                                                                            (23) ø
ø THERE WERE  nnnn  ERRORS IN SUMMATION  REPORTED FOR THE RUN OF MONTH OF xx ...... xx  ø
ø                                                                                 ø
ø                               END  OF  REPORT.                                  ø
ø                                                                                 ø
ø                                                                                 ø
ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø ø
```

/122 ......

1. Divisional number.
   The report is printed in order of divisions.

2. The month and year of the report.

3. The time (in hours) which the member of staff is expected to spend on the project.

4. The accumulated time spent by the member of staff on the project to the beginning of the month.

   Note that (4) = (6)-(5)

5. The time in hours spent during the current month and this month

6. The Accumulated Time spent over all previous months and this month.

7. The Accumulated Time spent as a negative value for projects for which the staff member did not have time budgeted.

8. The time in hours left for the staff member to complete the project. It is calculated as (6)-(3).

9. The rank of the staff member. Staff members appear in order of rank within their division.

10. The name of the staff member of 'STAFF WHO HAVE LEFT' or 'DIVISIONAL TOTAL'. Within a rank, the staff members appear in alphabetical sequence. The last table is the Divisional table and follows the tables for 'STAFF WHO HAVE LEFT'

11. Times spent on project dddd/pppp ... by the person nnn..nn/ (See 10)

12. The total time budgeted for or spent by staff member nnn .... nn

13. The Balance Total is the sum of the amount in columns (7) and (8), or the differences (6) - (3)

14. Warning asterisks

No asterisk	=	Time is still left
1 asterisk	=	Time overspent by 0 to 10% of budgeted.
2 asterisks	=	Time overspent by 10% to 100% of budgeted
3 asterisks	=	Time overspent by more than 100% of budgeted.

15. Another staff member

16. The divisional total. This table shows how much time the division as a whole is spending on each project.

17. An error message printed out if the amounts calculated by the program do not equal those written in the Accumulated Times file. (See under "Error Messages" D.I. (i) b. below)

18. The amount calculated by the program

19. The last page of the report

20. The number of errors that were found. This means that (17) appeared nnnn times.

21. Same as (2).

/124 .. .. ..

b. Error messages

PROJ NO SPACE.

There is insufficient space in the table for all the division-projects and their codes.

See Section on "Enlarging the project table in core" in the program IMPUT.

The program terminates.

THE C/F TOTALS DO NOT AGREE WITH THE CALCULATED TOTALS.

See (19) in D.I. (i) (a)

Cause of this message

(i) The Accumulated Times file (ACTIM) was not initialised correctly (See program INIT2)

(ii) The Accumulated Times file (ACTIM) was not updated properly (See program ACCUM2)

(iii) Not enough space is allocated for the Accumulated Times file (ACTIM)

User response

(i) Scratch the Accumulated Times file (ACTIM)

(ii) Retrieve the latest back-up copy that produced correct output. (Allocating more space, if necessary)

(iii) If possible, retrieve the monthly file current at that time (this is not always possible, as new entries may have been added to the file.)

(iv) Re-run all Accumulated (ACCUM2) and Edit (EDIT2) programs from that point onward. (If (iii) was not possible, or impractical, the Accumulation (ACCUM2) program may require selective

updating by using the proper options available.)

(v) Re-run this program (OUTRUT2)

II. Output

The following file is used for Output.

1. SYSOUT

E. Recovery

1. Where an error message is given, follow the instructions under User response.

2. On all other errors, re-run the program.

F. Files

DDNAME	Org.	Disp.	Symbolic Name
INFILE	I.S. (Seq.)	SHR.	ACTIM
CONAMS	I.S. (Seq.)	SHR.	CONAM
SYSOUT	Seq.	SYSOUT	SYSOUT.

Save the following files : None

OUTPUT3

A. Program name: OUTPUT3

(Note : This program has not yet been tested).

B. Purpose

To display the Supplier's Record file (ACSUP).

C. I. Parameter Cards

A single parameter card is required.

Format

Col. 1 - 80 Ignored

(This is just a dummy card, and at the moment is not used)

II. Input

The following files are used for input.

1. *
2. ACSUP

D. I. Messages

1. List of Possible messages.

a. Correct messages

DATE SUPPL ORDER EXT VTE DIVN PROJ SERIAL COMMIT DTE-INV INVOICE PAYMENT CHECK AMOUNT CANCEL C

A heading printed out before and after listing the file.

DISP4 STARTED.

The first messa e of the program.

/127 .. .. ..

a = Date of order
b = Suppliers number
c = Order number
d = Order extension number
e = Vote
f = Division code
g = Project code

} KEY

h = Serial number of commitment
i = Estimated Cost (Amount committed)
j = Date of invoice
k = Invoice number
l = Serial number of payment list
m = Serial number of cheque
n = Amount paid
o = Serial number of cancellation
p = *, if this record was modified or created by the use of parameter cards instead of the monthly file.

(Other p = blank)

b. Error messages.

None

II Output

The following file is used for output

1. SYSOUT

E. Recovery

1. After a program error correct the program and re-run.
2. After a 'standard' breakdown. Re-run the program.

/128 .. .. ..

F. Files

DDNAME	Org.	Disp.	Symbolic Name
ACSUPS	I.S. (Seq.)	SHR.	ACSUP
IN	Seq.	*	*
SYSOUT	Seq.	SYSOUT	SYSOUT

/129 . . .

SETUP

A. Program name : SETUP

B. Purpose

To set up the following tables and data well

i. The Sequential Table file (RANKS) with High-Low-Values and Blanks as applicable.

ii. The Project Names Table file (CONAM) with label records with keys equal to Low-Value and High-value

iii. The Input buffer (INBUF) full of High-Values and of length, NO-OF-RECS = 60, records.

C. I. Parameter Cards

None

II. Input

No files are used for input.

D. I. Messages

(i) List of Possible messages

a. Correct messages

The following messages should appear in a correct run, and in the following order:-

ØBEGIN.
ØSTART. 1.
Ø1ST REC.
Ø2ND REC.
Ø3RD REC.S
ØEND1. (The Sequential Table is now in
START2.
ØONLY REC.
ØEND2. (The Project-names Table is now in)
ØSTART3.
ØSTART 1ST REC.
ØEND OF LAST REC.
ØEND 3. (The Input buffer is now in)
ØFINISH.

/130 .. .. ..

The meanings of the messages are :-

ØBEGIN.

The first message of the program.

ØEND OF LAST REC.

When initialising the input buffer (INBUF), the last record, of the NO-OF-RECS = 60, has been written.

ØEND 1.

Initialising the Sequential Table (RANKS) has been successful.

ØEND 2.

Initialising the Project-names file (CONAM) has been successful.

ØEND 3.

Initialising the Input buffer (INBUF) has been successful.

ØFINISH.

The last message of the program showing that the program has terminated properly.

ØONLY REC.

The label records with a key of Low-Values and a key of High-Values have been written on the Project-names file (CONAM).

ØSTART1

Initialising the Sequential Table (RANKS) is about to begin.

ØSTART 1ST REC.

The first record in the buffer (INBUF) is about to be written.

ØTART2.

Initialising the Project-name file (CONAM) is about to begin.

/131 .. .. ..

ØSTART3.

Initialising the Input buffer (INBUF) is about to begin.

Ø1ST REC.

The first record in the sequential table file (RANKS) has been written.

Ø2ND. REC.

The second record in the sequential table file (RANKS) has been written.

Ø3RD. REC.S

The third record in the sequential table file (RANKS) has been written.

b. Error Messages.

None.

II. Output

The following files are used for output.

1. SYSOUT
2. RANKS
3. CONAM
4. INBUF

E. Recovery

1. After a system break-down
   a. Scratch the three files INBUF, RANKS, CONAM.
   b. Check and correct the job control cards if necessary.
   c. Re-run this program (SETUP)

F. Files

DDNAME	Org.	Disp.	Symbolic Name
RANKS	Seq.	NEW	RANKS
CONAMS	I.S. (Seq.)	NEW	CONAM
INBUF	Seq.	NEW	INBUF
SYSOUT	Seq.	SYSOUT	SYSOUT

/132 .. .. ..

Save the following files after a successful run:-

1. RANKS
2. CONAM
3. INBUF

Although it is not strictly necessary to save these files at this point, because if necessary they could just be re-initialised with this program (SETUP), but it is easier to take back-ups and recreate the files.

/133 .. .. ..

SETUPM

A. Program name: SETUPM

B. Purpose

Initialising the Monthly file MONTH.

C. I. Parameter Cards.

The program requires a single parameter card.

Format

Cols. 1 - 43 Alpha-numeric information to be placed in the label of the Monthly file.

44 - 80 Ignored

Note : (i) The label in the Monthly file appears as:-
MONTHLYbFILEbFORb
followed by the information or title given on the parameter card.

(ii) The parameter card should contain the date of the Monthly file to be created.

II. Input.

The following file is used for input.

1. *

D. I. Messages

(i) List of possible messages

a. Correct messages

END SETUPM

The last message of the program showing that program SETUPM has finished.

LABEL RECORD IN.

The label record has been written on the Monthly file.

START SETUPM

The first message of the program, and shows that program SETUPM has been started.

/134 .. .. ..

b. Error Messages

ERROR - NO PARAM CARDS

No parameter card was provided for the program.

User response.

Supply a parameter card and continue with recovery described in E(i) below.

II. Output.

The following files are used for output.

1. ACEX
2. SYSOUT

E. Recovery.

(i) After the error message or a system break down,

a. Scratch monthly file MONTH

b. Re-run the program

(ii) On other errors. Check the job control language cards, correct any errors, and continue with (i) above.

F. Files.

DDNAME	Org.	Disp.	Symbolic Name.
MNTHFL	Seq.	NEW	MONTH
IN	Seq.	*	*
SYSOUT	Seq.	SYSOUT	SYSOUT

Save the following files : None.

/135 .. .. ..

SORT1

A. Program name : SORT1

B. Purpose

To put data from a monthly file into a form in which it can be easily tabulated by means of simple reporting programs written in R.P.G.
Specified records from the monthly file are selected from a given week or set of weeks and sorted into alphabetic and numeric sequence on specified fields. The output from the program is written to a file in standard fixed length records with all numeric entries coded in decimal.

Decoding performed:-

i. All binary numbers are changed to packed decimal
ii. The project codes are decoded
iii. The extension to the order number is split from the main body of the order number. e.g. 7015211 becomes 701521/1
iv. The date is 'reversed' to place the 'year' first and the 'day' last in order to give the date a correct collating sequence.

C. I. Parameter Cards

These may appear in any order. They are identified by a key in Column 1.

a. Date Card.

Column 1 = I Ignore the Date records completely.
= D Choose records between the following weeks:-

and 2 - 3 = Lower selected week
4 - 5 = Upper selected week
6 - 80 = Ignored

If the upper limit (Cols 4 - 5) is omitted, it is set equal to the lower selected week.

/136 .. .. ..

Thus D03 is the same as D0303

and means : Select documents from the 3rd week only. The No. of the week must be right adjusted in its field. Leading zeros may be omitted.

b. Document Selection Card.

Column 1 = S This is a Selection card
2 - 4 = Document to be selected e.g.
0006 = Computer charges
5 - 7 = Document to be selected
etc. in three column fields up to Col. 79.

Column 80 or a zero or blank field on the card, terminates the scan of the card.
Valid document nos. (which must be right adjusted, but leading zeros can be omitted) are :-

1,2,3,4,5,6,7,8,9,10,11,12,21,22,23,31,32,33.

Any number of selection cards may be used but each must have the above format.

c. Ordering Sequence Card.

Column 1 = O (alphabetic) This is an Ordering card
2 - 4 = Sort first on this field
4 - 5 = Sort next on this field etc.

Column 80 or a zero or blank field on the card, terminates the scan of the card.
For a definition of the fields see the section on output below (D.II)
Any number of ordering cards may be used but each must have the above format.
The ordering is based on the logical collating sequence of the computer and thus may not place negative numbers before positive numbers.
Present restriction:- The total length of fields chosen must not exceed 33 bytes.

d. Check - Display Card

Column 1 = C, Check that items are numeric and refer to proper records.

II. Input

The following files are used for input:-

1. MONTH
2. CONAM
3. *
4. SELECT (If the Check-Display option is used)
5. Two working-space files

D. I. Messages

i. List of possible messages

a. Correct Messages

ACCEPT DOCUMENT xxxx

Document No. xxxx has been found in the Monthly file, within the specified date limits and will thus be accepted.

BEGINNING SELECT-SORT

The first message of the program. Identifies the system program producing the following output.

DATE RECORDS BE IGNORED

Option 'I' of the date card has been chosen.

DOCUMENT xxxx WILL BE SELECTED

Document No. xxxx was found on a document selection card and will be selected.

d1/d2/d3/ d20/
d21/d22

This is a display of the entries in the output file. The 1st 20 fields are printed on the first line, and fields 21 & 22 on the next line.

/138 .. .. ..

EDITED EM.
The end of the edit section has been reached.
If this message is not preceded by any error message the complete monthly file has been scanned.
If this message is preceded by an error message, then scanning of the monthly file stopped when the error was discovered.

END DISPLAY.
The display requested is now finished.
This message only occurs if a check-display has been requested.

END OF SORT
The selection and sorting of the required documents has been completed.

END SELECT-SORT
The last message of the program. Identifies the system program producing the above output.

NO
or YES

Message from the sort procedure to show whether FILE1 is being used as an input file for this data pass, or not.

SORT ON FIELD xxxx.
The records will be sorted on field xxxx.

SORTS IT GOOD EH.
Message to say that SORT is beginning a pass through the data
The number of times this message appears reflects the number of passes made through the data. If it appears many times (e.g. more than 10), either a new sorting procedure is required, or the working space for the present procedure should be enlarged.
(See SORT-IT-GOOD section in Part J).

STOWED EM AWAY.
The sorted entries have been written to the output file.

UNLESS IGNORE OPTION HAS BEEN SPECIFIED, SELECT BETWEEN WEEKS xx AND yy.
All the parameter cards have been now read in and selection will take place on blocks for weeks between xx and yy enclusively.
If the Ignore option has previously been specified, the 'select between weeks' option is overridden.

YES
See description under NO.

b. Error messages.

DOCUMENT xxxxxxxxxx CANNOT BE SELECTED.

A document selection card (key = S), required the selection of document xxxxxxxxxx which the program does not handle. See Section C.I.b. above.
The program terminates.

Recovery: Check the format of the card for a mis-aligned number, or for a mis-punch. If the specified document is really required, then program modification is necessary.
(Modify Sections :- SELECT-THE-DOCUMENTS and EDIT-EM)

FIELD NOT IN RANGE xxxxxxxxxx
An ordering sequence card (key = O), required sorting on field xxxxxxxxxx which is not an allowable field.
See section C.I.c. above and D. II. below.
The program terminates

Recovery: Check the format of the card for a mis-aligned number, or for a mis-punch.

/140 .. .. ..

INVALID KEY IN COL. 1.
The parameter card (which was printed out on the preceding line), has a key in the 1st column, other than those allowed in Section C.I. above.
The program terminates.

Recovery: Correct the incorrect card, and run again.

MAXIMUM FIELD SPACE EXCEEDED xxxx
In an ordering sequence card (key = 0), the total length of all fields chosen for ordering is greater than the program maximum (33 bytes)
The program terminates.

Recovery: Check to see if the parameters specify what is required, and check that the cards do not specify ordering on a field more than once. (e.g. Ordering 1st on field 4 then on field 2 and finally on field 4, does not need the last ordering option which duplicates the ordering on field 4).

NON NUMERIC ENTRY -- PLEASE CORRECT
The parameter card (which was printed out on the preceding line) contains a non-numeric entry in some column other than the first.
The program terminates.

Recovery: Check the format and punching of the incorrect card, correct the error and run again.

PROGRAM ERROR ONDOC xxxxSUSPEND ACTION.
This message will only occur if the section EDIT-EM in the program is incorrectly modified.
The error occured after reading document xxxx.
No further documents are read from the monthly file, but processing continues normally for the documents read up to this point.

/141 .. .. ..

Note : This message does not appear because of the introduction of a new document into the system, but only through incorrect program modification.

Recovery: Find where program was incorrectly modified
Note that in the EDIT-EM section the GO TO in paragraph PRO-CESS should not allow any document through that cannot be handled by the GO TO in paragraph GP-EXIT.

PROGRAM ERROR TRYING TO ORDER ON FIELD xxxx
The message comes from paragraph ONE-1 in section EDIT-EM
It denotes that ordering on field xxxx was accepted, but now that the program has to do it, there is no coding provided (or no pointer set up to where the coding is).

No further documents are read from the monthly file, but processing continues normally for the documents read up to this point.

Recovery: Find where the program was incorrectly modified, or why it does not work.

PROJ NO SPACE
The scatter storage table in core storage for the projects and their codes has been filled.
See Section E. Recovery 1 on page

The program terminates after this.

TOO MANY ORDERING FIELDS xxxx.
In an ordering sequence card (key = 0), xxxx fields were specified for ordering to take place.
This is more than the program can handle
The program terminates

/142 .. .. ..

Recovery: Check to see if ordering on so many fields is required (maximum number allowed = 40).

In the present version, the maximum field space will be exceeded before the number of ordering fields exceeds the maximum

II. Output

The following files are used for output

1. SYSOUT
2. ESELECT
3. Two working space files

Description of ESELECT.

All records in ESELECT have the following format (Note that the 'Fields' as numbered here, correspond to the fields given in the ordering sequence cards).

Field		Type	Picture	No. of Bytes
1	Document Code	C - 3	S9(3)	2
2	Entry number	C - 3	S9(3)	2
3	Division	C - 3	S9(3)	2
4	Project	C - 3	S9(7)	4
5	Vote	C - 3	S9(3)	2
6	Order Number	C - 3	S9(7)	4
7	Order Number Extension	C - 3	S9(3)	2
8	Serial Number	C - 3	S9(7)	4
9	Account Code	C - 3	S9(3)	2
10	Date (Reversed Year & Day)	D	S9(6)	6
11	Staff Division Code	D	x	1
12	Staff Rank Code	D	x	1
13	Staff Name Code	D	xxx	3
14	Supplier's Code	C - 3	S9(5)	3
15	Cheque Number	C - 3	S9(5)	3
16	Invoice Number	C - 3	S9(7)	4
17	Reference Number	C - 3	S9(5)	3
18	Time or Mileage	C - 3	S9(5), S999V99	3
19	Vehicle Code	C - 3	S9(3)	2
20	Estimated Cost	C - 3	S9(7)	4
21	Amount Paid	C - 3	S9(7)V99	5
22	Total Amount on Invoice	C - 3	S9(7)V99	5
			Total	57

where C - 3 = Computational - 3 = Packed Decimal
D = Display = EBCDIC
S = Sign
V = Decimal point
9 = a digit
(5) = 5 successive digits

/144 .. .. ..

E. Recovery

1. To recover from errors which result from insufficient program working space, either choose options that do not need so much space, or enlarge the core-area specifications. Increasing the number of fields, or the amount of space, must be accompanied by similar changes in the objects of the OCCURS clauses.
   To recover from PROJ NO SPACE., see under 'Recovery' in the description of the program IMPUT

2. For non-program halts, merely repeat the run, checking that the Output data sets will not cause a 'Duplicate name on Volume' error.

F. Files

DDNAME	Org.	Disp.	Symbolic Name
MNTHFL	Seq.	SHR	MONTH
WKFIL2	Seq.	NEW	SELECT
CONAMS	I.S.(Seq.)	SHR	CONAM
IN	Seq.	*	*
SYSOUT	Seq.	SYSOUT	SYSOUT
WKFIE1	Seq.	NEW	-
WKFIE2	Seq.	NEW	-

WKFIE1 & WKFIE2 have blocksize = 3600 bytes
WKFIL2 has blocksize = 3618 bytes

SPEC2

A. Program name: SPEC2

B. Purpose

To create a back-up copy of the Project-Names file (CONAM)

C. I. Parameter Cards

None

II. Input

The following file is used for input

1. CONAM

D. I. Messages

i. List of Possible messages

a. Correct messages

BACK-UP COPY CREATED

A back-up copy of the Project-Names file (CONAM) has been created

SPEC2 FINISHED.

The last message of the program.

SPEC2 STARTED.

The program SPEC2 has been entered and is about to begin.

b. Error messages

None.

II. Output

The following files are used for output.

1. SYSOUT

2. A back-up file of the Project-Names file

E. Recovery

1. After a system error
   a. If the back-up file is on disk, scratch it.
   b. Check and correct the job control cards if necessary.
   c. Re-run the program (SPEC2)

F. Files.

DDNAME	Org.	Disp.	Symbolic Name
CONAMS	I.S.(Seq.)	SHR	CONAM
BACKUP	Seq.	NEW	-
SYSOUT	Seq.	SYSOUT	SYSOUT

The back-up file is an unloaded form of the project-names file CONAM.

The file is retrieved by program SPEC3

Save the following files : None.

SPEC3

A. Program name : SPEC3

B. Purpose

To retrieve and re-organise the Project-Names file (CONAM) from a back-up copy created by program SPEC2

C. I. Parameter Cards

None.

II. Input

1. The back-up of the Project-Names file.

D. I. Messages

i. List of Possible messages

a. Correct messages

FILE CREATED FROM BACK-UP COPY
The Project-Names file (CONAM) has been re-created from the back-up copy.
It has also been re-organised.

SPEC3 FINISHED.
The program SPEC3 has finished.

SPEC3 STARTED
The first message of the program.

b. Error messages

None

II. Output

The following files are used for output.

1. SYSOUT
2. CONAM

/148 .. .. ..

E. Recovery

1. After a system break-down
   a. Scratch the Project-Names file (CONAM)
   b. Check and correct the job control cards if necessary
   c. Re-run the program (SPEC3)

F. Files

DDNAME	Org.	Disp.	Symbolic Name
CONAMS	I.S.(Seq.)	NEW	CONAM
BACKUP	Seq.	SHR	-
SYSOUT	Seq.	SYSOUT	SYSOUT

The back-up file specified by the DD card BACKUP, was created by program SPEC2

Save the following files : None

SPEC4

A. Program name : SPEC4

B. Purpose

To create a back-up copy of the Accumulated Times file (ACTIM).

C. I. Parameter Cards.

None.

II. Input

1. The file ACTIM

D. I. Messages

i. List of Possible messages.

a. Correct Messages

BACK UP FOR ACCUMULATED TIMES FILE CREATED.
A back-up copy of the Accumulated Times file (ACTIM) has been created

SPEC4 FINISHED
The program has finished.

SPEC4 STARTED
The first message of the program.

b. Error Messages

None

II. Output

The following files are used for output.

1. SYSOUT
2. A back-up of the Accumulated Times file

/150 .. .. ..

E. Recovery

1. After a system error.
   a. If the back-up file is on disk, scratch it.
   b. Correct the job control cards if necessary
   c. Re-run the program SPEC4

F. Files

DDNAME	Org.	Disp.	Symbolic Name
ACTIM	I.S.(Seq.)	SHR	ACTIM
BACKUP	Seq.	NEW	-
SYSOUT	Seq.	SYSOUT	SYSOUT

The back-up file is an unloaded form of the Accumulated Times file (ACTIM)

Program SPEC5 will retrieve the file.
Save the following files: None

SPEC5

A. Program name : SPEC5

B. Purpose

To retrieve and re-organise the Accumulated Times file (ACTIM) from the back-up created by program SPEC4.

C. I. Parameter Cards

None

II. Input

1. A back-up of the Accumulated Times file.

D. I. Messages

i. List of possible messages

a. Correct Messages

BACK UP TIMES FILE RETRIEVED.
The Accumulated Times file (ACTIM) has been recreated from the back-up copy. It has also been re-organised.

SPEC5 FINISHED.
The program has finished.

SPEC5 STARTED
The first message of the program.

b. Error Messages.

None

II. Output

The following files are used for output.

1. SYSOUT
2. ACTIM

/152 .. .. ..

E. Recovery

1. After a system break down
   a. Scratch the Accumulated Times file (ACTIM)
   b. Correct the job control cards if necessary.
   c. Re-run the program SPEC5

F. Files

DDNAME	Org.	Disp.	Symbolic Name
ACFILE	I.S.(Seq.)	NEW	ACTIM
BACKUP	Seq.	SHR	-
SYSOUT	Seq.	SYSOUT	SYSOUT

The back-up file specified by the DD card BACKUP, is the file created by program SPEC4.

Save the following files : None.

SPEC6

A. Program name : SPEC6

B. Purpose

To create a back-up for the Accumulated Expenditure file (ACEX)

C. I. Parameter Cards

None

II. Input

1. ACEX - the Accumulated Expenditure file.

D. I. Messages

i. List of Possible messages

a. Correct Message

BACK UP FOR EXPENDITURE FILE CREATED

A back-up for the Accumulated Expenditure file has been created.

SPEC6 FINISHED

The last message of the program.

SPEC6 STARTED

The first message of the program.

b. Error Messages.

None.

II. Output

The following files are used for output.

1. SYSOUT
2. A back-up of the Accumulated Expenditure File.

/154 .. .. ..

E. Recovery

1. After a system break down

   a. If the back-up file is on disk, scratch it.

   b. Correct the job control cards if necessary.

   c. Re-run the program SPEC6

F. Files

DDNAME	Org.	Disp.	Symbolic Name
ACFILE	Seq.	SHR	ACEX
BACKUP	Seq.	NEW	-
SYSOUT	Seq.	SYSOUT	SYSOUT

The back-up file is an unloaded form of the Accumulated Expenditure file ACEX.
The program SPEC7 will retrieve the file
Save the following files : None

SPEC7

A. Program name : SPEC7

B. Purpose

To retrieve and re-organise the Accumulated Expenditure file (ACEX) from the back-up created by program SPEC6

C. I. Parameter Cards

None

II. Input

1. The back-up for the Accumulated Expenditure file.

D. I. Messages

i. List of Possible messages

a. Correct Messages

BACK UP FOR EXPENDITURE FILE RETRIEVED

The Accumulated Expenditure file (ACEX) has been recreated from the back-up. The file has also been re-organised.

SPEC7 FINISHED

The last message of the program.

SPEC7 STARTED

The first message of the program.

b. Error Messages

None.

II. Output

The following files are used for output.

1. SYSOUT
2. ACEX

/156 .. .. ..

E. Recovery

1. After a system breakdown

   a. Scratch the Accumulated Expenditure file (ACEX)

   b. Correct the job control cards if necessary.

   c. Re-run the program SPEC7.

F. Files

DDNAME	Org.	Disp.	Symbolic Name
ACFILE	I.S.(Seq.)	NEW	ACEX
BACKUP	Seq.	SHR	-
SYSOUT	Seq.	SYSOUT	SYSOUT

The backup-file specified by the DD card BACKUP, was the file created by program SPEC6.

Save the following files : None

WEEK1

A. Program name : WEEK1 (Written in RPG)

B. Purpose

To print Suppliers' Accounts Commitments

C. I. Parameter Cards

None

II. Input

1. SELECT

This file is obtained from program SORT1 by using:-

1. A Date Card containing the week of the return to be printed.
2. A Select Document, e.g. an Order form (document 2), or a committed from a list of committed items (document 10)
3. Sort on fields 6 & 7 (order number and its extension)

(See program SORT1 for further details)

D. I. Messages

i. List of possible messages

a. Correct Messages

The Report:-

	(1) ORDER	(2) EXTN	(3) VOTE	(4) DEPT	(5) PROJECT	(6) AMOUNT	(7) DR.
	xx	xx	xx	xx	xx	xx	2
	xx	xx	xx	xx	xx	xx	2
(10)	'	'	'	'	'	'	'
	'	'	'	'	'	'	'
	xxx	xx	xx	xx	xx	xx	2
			(8) → PAGE TOTAL			xx ← (9)	

/158 .. .. ..

Explanation

1. The order which committed the expenditure.
   Entries are in sequence of order number.
2. The order number (1) extension.
3. The vote against which the amount was committed.
4. The division and project debited with the amount
5. The amount committed in rands.
6. A '2' shows the amount (6) is a debit against the project (5), division (4), and vote (3). If the amount is a credit a '1' appears in this column.
7. The page total printed out at the foot of each
8. The total of all amounts (6) on the page. Credits are subtracted from the total figures.
9. A page entries.

b. Error Messages

None

II. Output

The following file is used for output.

1. SYSOUT

E. Recovery

1. After a break down
   a. Check and correct the job control cards if necessary.
   b. Unless SELECT has been given a permanent name and has been kept, re-run the SORT1 program and follow with this program (WEEK1)

Note: If the validity of the data is **doubtful run the** SORT1 **program with the** Check-Display option (See program SORT1)

/159 .. .. ..

F. Files

DDNAME	Org.	Disp.	Symbolic Name
SELECT	Seq.	SHR	SELECT
REPORT	Seq.	SYSOUT	SYSOUT

Save the following files : None

A description of the SELECT file appears in C. II (Input) above.

YELLOW

A. Program name : YELLOW (Written in RPG)

B. Purpose

To print a report similar to the Analysis of Project Costs so that the Institute's figures can easily be reconciled with those produced by headquarters.

C. I. Parameter Cards.

None.

II. Input

The following file is used for input.

1. SELECT

This file is obtained from program SORT by :-

i. Selecting Documents 2,3,4,5,6,7,8,9,10,11,12

ii. Sorting on fields 4 or 5 (in that order).

(See program SORT1)

D. I. Messages

i. List of possible messages

a. Correct Messages

The Report:-

ANALYSIS OF PROJECT COSTS.

(1) DOC	(2) SRCE	(3) SERIAL	(4) CHEQUE	(5) ORDER	(6) VOTE	(7) DEPT	(8) PROJECT	(9) SPENT	(10) COMMIT-TED
xx	xx	xx	xx	xx	xx	xx	xx	xx	
xx	xx	xx		xx	xx	xx	xx		xx
xx	xx	xx	xx	xx	xx	xx	xx	xx	xxCR
								(11) xx T	
xx	xx	xx		xx	xx	xx	xx		xx
xx	xx	xx	xx	xx	xx	xx	xx	xx	
								(11) xx T	
								(12) xx ST	
(13)									
xx	xx	xx	xx	xx	xx	xx	xx	xx	xxCR

Explanation

1. The document number.
2. The origin of the expenditure. (Usually refers to a document).
3. The serial number of the document (where applicable).
4. The number of the cheque settling the account (where applicable).
5. The number of the order which committed the expenditure.
6. The vote against which the expenditure was debited.
7. The division debited with the expenditure.
8. The project debited with the expenditure.
9. The amount spent. If negative it is followed by the letters CR (CREDIT).
10. The amount committed. If negative, it is followed by the letters CR (CREDIT)
11. The accumulated amount spent or committed since the last total.
    This total is printed every time the vote or project number changes.
12. The accumulated money spent or committed since the last such total figure (ST)
    This total is printed out every time a project number changes. It is thus the sum spent on the project (usually for a specific month).
13. Another set of similar figures for the next project.

Note : Amounts spent, committed or totalled, that are negative are followed by the letter CR (for CREDIT).

/162 .. .. ..

b. Error Messages

None

II. Output

The following file is used for output.

1. SYSOUT

E. Recovery

1. After a break-down

a. Correct the job control cards if necessary.

b. Unless SELECT has been given a permanent name and saved, re-run the SORT1 program.

c. Re-run this program (YELLOW)

Note : If the validity of the data to be printed is doubtful, run the SORT1 program with the Check-Display option.
(See program SORT1)

F. Files

DDNAME	Org.	Disp.	Symbolic Name
MONTHLY	Seq.	SHR	SELECT
REPORT	Seq.	SYSOUT	SYSOUT

Save the following files : None

See C. II (Input) above for the description of the SELECT file required by this program.

/163 .. .. ..

Subroutines used by various Programs in the system

Abnormally terminate task	ABEND
Close the terminal line	CLS
Converse on terminal (i.e. give prompt and receive response)	
Supply date	DATE
Supply edited date	GVKSUB1
Open terminal line	OPN
Modify a number	SMOD
Supply time	TIME
Terminal I/O routines	TRMNLIO
Write to terminal	WRT

/164 .. .. ..

Subroutine ABEND

A. Purpose An Assembler language subroutine for abnormally terminating the progran and returning a condition code.

B. Entry name: ABEND

C. Usage:

CALL 'ABEND' USING USER-CONDITION-CODE

where

USER-CONDITION-CODE USAGE IS COMPUTATIONAL PICTURE S9(9), VALUE IS nnnn

and nnnn is the condition code that is returned by the user.

D. Routines called:

None

/165 .. .. ..

Subroutine CLS

A. Purpose:

An assembler language subroutine for closing a terminal line.

B. Entry name is CLS

C. Usage

CALL 'CLS' USING THE-LINE-NUMBER

See discussion in Subroutine OPN

D. Routines called:

TRMNLIO

E. Note: Subroutine OPN must have been called, before this subroutine is called.

/166 .. .. ..

Subroutine CNVRS

A. Purpose

An Assembler language subroutine for:-

(i) typing out a message
(ii) typing out a prompt
(iii) receiving a line of input
(iv) reporting any errors in the above

B. Entry name is CNVRS

C. Usage

```
CALL 'CNVRS' USING THE-LINE-NUMBER,
                   NUMBER-OF-CHARACTERS
                   THE-MESSAGE,
                   THE-PROMPT
```

where

```
THE-LINE-NUMBER USAGE IS COMPUTATIONAL PICTURE S9(9)
NUMBER-OF-CHARACTERS USAGE IS COMPUTATIONAL PICTURE S9(4)
THE-MESSAGE  USAGE IS DISPLAY              PICTURE X(244)
THE-PROMPT   USAGE IS DISPLAY              PICTURE X.
```

(i) nput to the subroutine:-

THE-LINE-NUMBER, see discussion in Subroutine OPN
NUMBER-OF-CHARACTERS is the number of characters in the message to be typed
THE-MESSAGE is the message to be typed
THE PROMPT is the single character prompt that is to be printed after the message.

Note that if NUMBER-OF-CHARACTERS is equal to zero, no message is printed and only the prompt is given. Where a message is to be typed, the subroutine tags 'new-line' and 'carriage-return' and a suitable number of 'idle-characters' to the end of the message.

/167 .. .. ..

(ii) Output from the subroutine:-

THE-LINE-NUMBER : If unchanged, then no error has occured but if set to zero there has been I/O error.
Recovery procedure is left to the calling program.
No assumption should be made as to the contents of either NUMBER-OF-CHARACTERS or of THE-MESSAGE.

NUMBER-OF-CHARACTERS: The number of characters entered by the terminal user, excluding the 'carriage return' and 'new-line' characters.

THE-MESSAGE : The line of input entered by the terminal user.
Only the characters entered are placed in THE-MESSAGE.

THE-PROMPT : Remains unchanged.

Note that no editing is done by the subroutine, (e.g. back space elimination, lower to upper-case conversion), it must be done by the program.

D. Routines called : TRMNLIO

E. Note : This subroutine should only be called for a line-number that has been opened by subroutine OPN.

/168 .. .. ..

Subroutine GVKSUB1

A. Purpose

A Cobol language subroutine for returning, in edited form, the date and time according to the computer's clock.

B. Entry name is DATEG.

C. Usage

CALL 'DATEG' USING THE-DATE-AND-TIME.

where (in the Linkage Section):-

```
1.  THE-DATE-AND-TIME

 2  THE-DATE
    3  THE-DAY-OF-THE-MONTH          PICTURE 99.
    3  EDITING-SYMBOL-1              PICTURE X.
    3  THE-PRESENT-MONTH             PICTURE 99.
    3  EDITING-SYMBOL-2              PICTURE X.
    3  THE-YEAR                      PICTURE 99.
 2  THE-TIME
    3  THE-HOUR-IN-THE-DAY           PICTURE 99.
    3  EDITING-SYMBOL-3              PICTURE X.
    3  THE-MINUTES-PAST-THE-HOUR     PICTURE 99.
    3  EDITING-SYMBOL-4              PICTURE X.
    3  THE-SECONDS-PAST-THE-MINUTE   PICTURE 99.
```

and the items have the meaning described in the data-name.

EDITING-SYMBOL-n is the editing symbol inserted by the user
(e.g. if the first two editing symbols are '/' then THE-DATE will print out as dd/mm/yy)

D. Routines called : DATE

/169 .. .. ..

Subroutine OPN

A. Purpose

An Assembler language su
a terminal line for I/O.

B. Entry name is OPN.

C. Usage

CALL 'OPN' USING THE-LINE-NUMBER

where

THE-LINE-NUMBER USAGE IS COMPUTATIONAL PICTURE IS S9(a)
VALUE IS nnn

nnn is the number of the line to be opened. (A program reference line-number) and appears on a DD card with DDNAME of LINE nnn

(e.g. If nnn is equal to 1, then there will be a DD card :-

//LINE001 DD DISP=(OLD,DELETE), UNIT=xxx

where xxx is the terminal's unit number.

Routines Called : TRMNLIO.

D. Note : Before the program terminates, sub-routine CLS should be called for this line.

/170 .. .. ..

Subroutine SMOD

A. Purpose

A Fortran subroutine for returning a whole number as n mod (401), where n lies in the range 1 to 401.

B. Entry name is SMOD.

C. Usage

CALL 'SMOD' USING THE-INT

where

THE-INTEGER-TO-BE-MODIFIED

USAGE IS COMPUTATIONAL PICTURE IS S9(9)

VALUE is nnnnnnnnn

and nnnnnnnnn is the number that is to be modified lying in the range 1 to 401.

After returning from the subroutine, THE-INTEGER-TO-BE-MODIFIED will have been modified.

D. Routines called.

1. Fortran library subroutine MOD

/171 .. .. ..

Subroutine WRT

A. Purpose

An Assembler language subroutine for

(i) typing out a message

(ii) reporting any errors in the above

B. Entry name is WRT.

C. Usage

CALL 'WRT' USING THE-LINE-NUMBER, NUMBER-OF-CHARACTERS, THE-MESSAGE

where

these data-names are the same as those in subroutine CNVRS.

(i) input to the subroutine

The action is the same as in (i) (input to the subroutine) of subroutine CNVRS, except that there is no prompt typed.

(ii) output from the subroutine

THE-LINE-NUMBER : If unchanged, then no error has occured, but if set to zero there has been I/O error Recovery procedure is left to the calling program.

NUMBER-OF-CHARACTERS: Unchanged

THE-MESSAGE : Unchanged

D. Routines called: TRMNLIO

E. Note : This subroutine should only be called for a line-number that has been opened by subroutine OPN.

/172 .. .. ..

Other Subroutines

Subroutine TIME

See Reference 2.

Subroutine DATE

See Reference 2.

Subroutine TRMNLIO

A University of the Witwatersrand Computing Centre subroutine which connects the Accounting system terminal subroutine, to the terminals.

/173 .. .. ..

Correcting Errors not reported by the System Programs

When data is presented to a system, the system can only check it for validity and reasonability.

Discrepancies and other types of error may be detected when a report has been printed.

Section A.

Errors in committing, cancelling and paying orders, discovered either by the program ACCUM3 or through the 'reconciliation' schedule (See program YELLOW)

The program ACCUM3 will detect discrepancies between commitments and cancellations or payments. The reconciliation report by YELLOW will show the items omitted.

The following are the errors and their remedies:-

1. Payment has been made for an order which has not been committed.
   Remedy Commit the order

2. Payment has been made but the order number quoted differs from that in which the amount was committed.

   Remedy

   a. If payment was made under a correct order number:-
      (i) Cancel the incorrect order number.
      (ii) Commit the amount paid under the correct order number.

   b. If payment was made under an incorrect order number
      (i) Reverse the entry by entering the payment with a minus sign.
      (ii) Enter the payment correctly (use the correct order number)

/174 .. .. ..

3. An incorrect amount has been committed on an order.

Remedy

(i) Cancel the order

(ii) Re-enter the order with the correct amount

(iii) Modify the suppliers' record file.

4. An incorrect amount has been entered when cancelling an order.

Remedy

Cancel the difference in the amounts

e.g. R20 was committed and R15 cancelled.
Continue by cancelling R5 for the same order.
If the amount cancelled was R25, commit an additional R5.

5. An incorrect committed amount has been entered on a payment entry

Remedy

Re-enter the payment with a payment of zero and a committed amount equal to the difference between the two committed amounts

e.g. Committed R20, paid R18 with an incorrect commitment of R15.
Pay R0 with a commitment of R5

Note : The committed amount in a payment automatically offsets the amount previously committed against the order.

6. A Non-existent order, has been cancelled.

Remedy

(i) Enter the order number that was cancelled as a commitment.

(ii) Cancel the order number that was supposed to have been cancelled.

7. An existing order has been incorrectly cancelled.

Remedy

(i) Cancel the order number that was supposed to have been cancelled.

(ii) Re-commit the order number that was incorrectly cancelled.

(iii) Edit the suppliers' record file

Section B

Correcting errors found in a Project Expenditure Report (produced by program OUTRUT1)

Note : Common errors and methods of correcting them are given in the description of the program OUTRUT1.
The errors described here are those that only the user would recognise, not a program.

1. A budget figure is incorrect.
   Remedy
   Enter the difference with document 31 (adjust Budget)

2. An accumulated total is incorrect.
   Remedy
   Enter the difference with document 32 (adjust Accumulated Amount)

3. A committed amount is incorrect.
   Remedy
   Refer to Section A above.

4. Expenditure has been debited to the wrong project.
   Remedy
   Transfer the amount to the proper project using document 30. (Transfer Project)

5. All amounts for one project have been included in those for another

   Remedy

   a. Amend the figures as in (1), (2) and (3) above.
   b. Enter the other project figures in the usual way.

/176 .. .. ..

6. A project number is replaced by asterisks showing that the report program cannot decode the project-code. This is due to the cancellation or transfer of a project which has already taken effect in the tables but the executive document in the monthly file has not yet reached the Accumulated Expenditure file.

Remedy

Run program ACCUM1 past the block containing the adjustment.

7. Altering amounts in CANCELLED/FINISHED projects

Remedy

(a) Create a new project within the same division.
(b) Alter the figures as in (1), (2) and (3) above.
(c) Cancel it. (This will add the figures in this project to those in the CANCELLED/ FINISHED project)

8. To change the alpha-numeric description of the project. (Unfortunately the system does notfacilitate this elementary change).

Remedy

(a) Transfer the amounts for this project to a new dummy project, in the same division.
(b) In a separate sub-block in the monthly file, transfer this dummy project to the project with the required alpha-numeric description using document 30

Note that the entries created by (a) and (b) must be treated in that order and in separate runs of program ACCUM1.

/177 .. .. ..

9. To remove Monthly file records which have only zero entries.

Remedy

Run program EDIT1.

Note that if the date on the file is to remain unchanged, a suitable parameter card must be provided (see program EDIT1)

Section C.

Correcting errors found in the Staff Times Report produced by program OUTRUT2

Note: Common errors and methods of correcting them are given in the description of program OUTRUT2.
The errors described here, are those recognised by a user but not by a program.

The differences between documents 21 and 23

(i) Document 21 adds the given figure to the time budgeted for a staff number on a project.
Document 23 replaces the budgeted time by the new figure

(ii) Document 21 after translating the time to rands, will also modify the accumulated expenditure file.
Document 23 does not affect the accumulated expenditure file.
Where possible document 21 should be used in preference to document 23.

1. A Budget figure is incorrect

Remedy

a. If the project number is known, use document 21 or 23 to correct the figure.

b. If the project is the cancelled or finished category, use document 25 (Document 23, using project code) to correct the figure.

/178 .. .. ..

2. An accumulated time is incorrect.

Remedy

a. If th is known, use document 01 or 22 (Adjust Accumulated Time) to correct the figure.

b. If the project is in the Cancelled or Finished category, use document 24 (A Document 22 using project code) to correct the amounts.

3. A project is printed with asterisks for project number. This results when a project is cancelled or finished or entries have been transfered to another project.

Note: Changing the description of a project, amounts to a transfer since the project number has a different project code, even though it is unchanged.

Remedy

If the project is cancelled or finished, leave it as it stands.

If the project is transfered:-

a. Re-enter the figures for the new project (Using documents 22 and 23 so as not to affect the accumulated expenditure file.)

b. Remove the project now printed out with asterisks, by setting its figures to zero using documents 24 and 25.
(Projects with zero entries are not printed)

4. To change figures in the table: STAFF WHO HAVE LEFT

Remedy

a. Create a dummy person in the same division.

b. Give the dummy person the changes that are required in the table.

c. Enter this person as a resignation. (This has the effect of adding all these figures for this person to the projects in STAFF WHO HAVE LEFT).

5. Changes in descriptions

   a. the division of the staff member
   b. the rank of the staff member.
   c. the name of the staff member (e.g. after getting married)
   d. the initials of a staff member.

   Remedy

   Transfer the person using decument 30 (Transfer project)

6. Time has incorrectly been attributed to a staff member.

   Remedy

   Add the required figures to the staff member who has not got them (As in 1 and 2 above).

   b. Subtract these figures from the one who should no longer have them (see 1 and 2 above).

7. To remove records in the file where all entries are zero.

   Remedy

   Run program EDIT2

   Note that if the date on the file is to remain unchanged, a suitable parameter card must be provided (See program EDIT2).