COMPUTERIZATION OF INTERBRANCH RECONCILIATION

(A CASE STUDY OF WEMA BANK PLC, AGBARA-LAGOS).

BY

ABDUL FATAI ADEWALE ALARAN

PGD/MCS /96/97/308

DEPARTMENT OF MATHEMATICS/COMPUTER SCIENCE,
FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA.

SEPTEMBER, 2001

APPROVAL PAGE

This project work has been read and approved by the undersigned, as meeting the requirements of the Department of mathematics/Computer Science, Federal University of Technology, Minna.

.....	.....
PRINCE R BADMUS	DATE
PROJECT SUPERVISOR	
.....	.....
DR. S.A REJU	DATE
HEAD OF DEPARTMENT	
.....	.....
EXTERNAL EXAMINER	DATE

DEDICATION

This Piece of work is dedicated to the entire Alaran's Family particularly my immediate family who stood by me in thin and thick.

May the almighty God continues to shower his blessings and mercies over you all, particularly my darling wife and the children. You own it all.

DEDICATION

This Piece of work is dedicated to the entire Alaran's Family particularly my immediate family who stood by me in thin and thick.

May the almighty God continues to shower his blessings and mercies over you all, particularly my darling wife and the children. You own it all.

ACKNOWLEDGEMENT

The God almighty, all glory for the Successful Completion of this programme. Who else can one call upon except the almighty.

With deep sense of joy and appreciation, I wish to place on record the special rote played by my project-Supervisor-Prince R.O Badmus. (OMO OBA) you have been very close all the time from the beginning of the program to the end despite your tight schedule, your constructive criticisms and corrections saved me from project pitfalls. I owe a special gratitude to you - Prince. May Allah continues to bless, guide, guide you in all your Understanding.

My unreserved gratitude to all my Lecturers and the current Head of department, Dr S.A. Reju. I wish to acknowledge the special roles of my parents for their proper up bringing and moral values inculcated on us. Also, special Space to my wish and the children. Thanks for your care, love and concern. I had to leave You for most time to see to my lectures and tests. Your Support and Concern are highly appreciated- To my colleagues in the banking Sector, you have all been very wonderful and cooperative. May the lord bless us all.

To others not mentioned here by names, may the almighty Allah continues to remember you also with abundant blessing, love to you all a special thanks.

ABSTRACT

Inter branch Accounting seeks to record transactions between two branches of a given bank or a branch and Head Office. Inter branch Accounting evolved as a result of increase in branch network of banks coupled with explosion in economic activities in the country.

The need for co-ordination of the activities of branches, and a mark of improvement in the services rendered to customers, banks introduced innovations to out do the others to win more customers.

Consequently, most banks introduced system whereby a customer in branch A could transact business in other branches of the bank without necessarily maintaining an account in that branch.

It is expected that at the end of this work, the reconciliation that hitherto is manually done would be fully automated thereby leading to greater efficiency, accuracy and timely disposition of this tedious task.

TABLE OF CONTENTS

credited to his account in Abuja or as the case may be by way of inter branch transaction. It is also possible for a particular branch o render services such as payment of bills and then transfer to the branch concerned.

1.2 BACKGROUND OF STUDY

This study was spurred by the advent of computer technology and its subsequent application in the business-world which necessitated a great change in banking industry, the prevailing problems of inter branch reconciliation in Wema Bank plc is that it is manually done and the need to proffer a lasting solution by automating this process to ensure greater efficiency, accuracy and timely disposition of this tedious task.

1.3 PURPOSE OF STUDY

The aim of this study is to critically examine the problem of inter branch accounting and reconciliation to evolve a program that would eliminate boredom and tedious tasks associated with aspect of banking operation.

1.4 SCOPE AND LIMITATIONS

This study is limited to Wema Bank Plc and will be confined to computerizing inter branch reconciliation. As most documents to be used in

this project are confidential, efforts will be to ensure that date used do not in any way adversely affect the system or the bank.

1.5 SIGNIFICANCE OF STUDY.

At the end of this project, it is expected that the study will achieve the set objectives of enhancing efficiency and guarantee accuracy of inter branch accounts.

Also it is expected that hour lost in reconciliation of inter branch would be used in other operations areas that need attention.

CHAPTER TWO

2.0 MISSION STATEMENT

Wema bank mission is to provide at an added value and using the most modern means, efficient commercial banking services for the benefit of the customer shareholders and society at large. This laudable objective is expected to be achieved through the development of a new are of efficient banking services through the re-orientation and development of state-of the-art computer based systems and the practice of professionalism.

2.1 BUSINESS STRATEGY

As small, greasing institution, the bank has been building its competitive strength through investments in people systems and operational resources. Business-wise, a generic strategy of differentiation in service quality and cost control is being pursued. A corollary of this strategy the bank adopted the development of a strong , branch network as a key strategic option in furtherance of its mission of providing efficient commercial banking services for the benefit of all.

2.2 INTER BRANCH ACCOUNTING

Accounting can be described as a business language used to establish, summarize, present and inner prate both economic and financial values as an

(c) Enhance promptness in reconciliation of interbranch statements.

(d) It involves branches reconciling outstanding with other branches and copies of each branch reconciliation is sent to Head Office Accounts.

(e) It makes it easier to identify a braches that are yet to be responded to.

(f) It avoids long outstanding items

(g) Help to case problems encountered when reconciling.

DISADVANTAGES

(a) There are delays in communications

(b) Instrument and covers for draft etc do not easily get to the receiving branch as all originating entries are passed through Head Office.

(c) Not cost effective in terms of paper work involved.

(d) Misplacement of documents. Because of the number of people and documents involved, it may be a bit difficult to control the movement of documents.

(e) Also, the problem of inappropriate description of transaction may be noticed, that is, as two originating vouchers will be raised, the possibility of one misrepresentation the other can not be ruled our

(b) DECENTRALIZED SYSTEM

Another method of interbranch reconciliation system is the decentralized method. This system allows branches to deal directly with each other, interbranch vouchers are indiscriminately raised against another branch whether for transfer or for claims. Unlike centralized system where interbranch vouchers have to be first sent to the Head Office Operations for further process, Head Office is in the picture of transactions between a branch and another branch only when the reconciliation copies of the interbranch vouchers get to them. This is the method currently being used in Wema Bank Plc.

Interbranch voucher is normally made up of four copies, one copy is retained by the originating branch, one copy from the originating branch is sent to Head Office reconciliation unit, one copy is retained by the responding branch and the last copy is sent to Head Office to complete the transaction between the two branches.

Under the current system, a register is maintained for each branch. This register is called the originating Register. It records the branch transaction with other branches. For example, Head Office register is sub-divided to show transactions with all branches

ADVANTAGE OF DECENTRALIZED SYSTEM

(a) This system enhances communication between branches

(b) Instruments and covers for drafts easily get to the responding branch.

DISADVANTAGES

(a) Gives room for suppression of interbranch vouchers

(b) Causes great bottlenecks in the reconciliation

(c) Creates room for fraud

(d) Allows uncleared vouchers to be long outstanding.

2.5 RECORDING EXTRACTS INTO THE REGISTER

When extracts are received from the branch from the branches, they are first recorded into Interbranch Receipt Register. This enables us to know at a glance which extract was received and the date of receipt. These are cross checked and signed by the Head of the Reconciliation Unit. They are then sorted to their respective branches and then originating vouchers will be separated from responding extracts and then arranged in order of dates.

Before recording into the reconciliation register, the extracts are thoroughly checked to ensure accuracy in respect of interbranch voucher number, originating date, originating branch, the responding branch, and the amount in figures and in works. The vouchers are then entered into their

DEBIT (DR) Branch concerned (through inter branch)

CREDIT (CR) Interest on Bank Funds (Income)

At the Branch, respond to the inter branch thus:

DEBIT (DR) Interest on Bank Funds (Expense)

CREDIT (CR) Head Office Operations (inter branch)

For (b)

At Head Office Operations:

DEBIT (DR) Interest on Bank Funds (Expense)

CREDIT (CR) Head Office Operations (inter branch)

Interest on bank funds is the interest earned on funds invested with Head Office. All branches, CBN Account balances are transferred to Head Office Account with CBN in Lagos. Since head Office used their funds.

2.6 CALCULATIONS OF INTEREST ON BANK FUNDS

Interest payable or receivable by the branches (for debit\credit balances on their inter branch accounts) shall be originated by Head Office

Operations. The branches are advised on a monthly basic of their inter branch status. The applicable rate of interest used in either paying or charging branches is usually tried to the Bank's prime lending rate prevailing at the time of calculation.

The current rates are:

- 2.5% below prime lending rate for net credit balances and
- 1 % below prime lending rate for debit balances.

THE Procedure for the interest calculation is an following:

Interest = PxR

Where: N

P = Product (made up of daily balances added up together from the

Balances added up together from the branches' Mirror Account)

R = Rate of Interest

N = Number of days in a year

At the Branch, respond to the inter branch thus:

DEBIT (DR) Head Office Operations (inter branch)

CREDIT (CR) Branch concerned (income)

General Rules

It is the responsibility of signing Officers to ensure that the vouchers as well as the extracts are properly and fully completed with correct dates, transfer codes, branch numbers and test number (where applicable). Transaction should be concise and must adequately describe the underlying transaction.

USES OF INTEREST ON BANK FUNDS

Interest earned on inter branch is an expense to Head Office, while it is an income to the branches. In the branch, the interest is credited to their profit & Loss Account while to complete the double entry.

Interest on bank fund is used to monitor the performance of the branches with respect to amount invested with Head Office.

Recording into the Inter branch Registers (Central Accounts)

The Central Account is maintained for each branch. This is used to record all inter branch transactions with other branches including Head Office Operations All originating extracts are recorded on the left hands side (debit side while all responding extracts

Are recorded on the credit side against the originating branch on the same line. When the two have been recorded, it confirms the transactions between the branches and so it is cleared

2.7 LISTING OF OUTSTANDING ITEMS

At the end of each of each month, a list of outstanding items are drawn for both the debits and the credits. This list is cross-checked again to ensure that all outstanding. Items have been taken and also ensure that they are actually outstanding. This is the list used for the reconciliation.

Where one side has been received and the other is not received for up to a week, it is necessary to find out from the branch the reason for such delay.

2.8 RECONCILIATION

The reconciliation exercise is the climax of activities relating to inter branch transaction at the Head Office. Inter branch reconciliation is done exactly the same way as a conventional bank reconciliation. At the end of reconciliation exercise there ought to be nil balance, but most of the time, it is not so because of items in transit and human error in recording but the difference must be negligible for it to be accepted.

It is the function of the Reconciliation Unit to follow up an non receipt of original vouchers with originating branch. The originating branch is to be requested to issue duplicate vouchers in case of lose of the original When preparing duplicate vouchers, the originating branch will strike off printed serial number and put the number of the original voucher on the duplicate and mark the voucher in bold letters DUPLICATE VOUCHERS". The number and date of issue of the original voucher must always be identical as otherwise both entries with be reflected as outstanding on the Central/Reconciliation Account.

DOCUMENTS USED

Documents used for inter branch transaction include the Inter branch vouchers, Extract Sheets, Inter branch debit or credit Vouchers, Reimbursement tickets etc.

USE OF INTER BRANCH

Inter branch has several uses which include but not limited to:

(a) Funds Transfer

This is used where funds are being transferred from one branch to the other. For example, it is possible for Mr Ajala an account holder in Ibadan Branch to pay in one Million Naira cheque into his account at Enugu Branch where he does not have an account and the money will be transferred to his Ibadan Branch Account through inter branch transaction

This will involve raising credit inter branch. On the other hand. It is also possible for Abuja branch to pay Five million Naira to Mr Ajala who is an account holder in Ibadan where he maintains an account and raise a debit inter branch against Ibadan Branch.

(b) DRAFTS

It is also possible for a customer to purchase a draft from Lagos payable in Abuja The cover for this draft is sent along with an inter branch to enable the responding branch pay the draft.

(c) Other uses of inter branch vouchers include:

Telegraphic Transfers

All debit transactions involving profit & Loss

All credit transactions involving profit & Loss

Transfer of CBN balances

Transfer of an end of year profit & Loss

Transfer of Fixed Assets etc.

All forms of transfers from one branch to the other is done by means of inter branch.

Coding of Inter branch

For operational efficiency and security all inter branch transactions are coded. To authenticate the genuity of inter branch transactions involving external persons, such transaction must be tested and a test number quoted on the space provided on the inter branch vouchers. The transaction that need to be tested are drafts. Mail and telegraphic transfers. Testing usually is as per the agreed testing procedure using secret test lay number which are in the control of the branch Head/Manager. Vouchers that contain incorrect or incomplete test number are returned unapplied by the responding branch to the originating branch

2.9 ANALYSIS REGISTERS

Branches and Head Office Operations maintain a register each for all originating and responding inter branch vouchers this services as memorandum register.

(a) Out ward register:

To record all outward (originating) inter branch vouchers

(b) Inward Register:

To record details of all inward (responding) inter branch vouchers, correct and timely completion of these registers would allow for easy reconciliation and show at a glance those items still outstanding. All originating extracts from a branch are recorded on the left side of individual registers while the responding extracts are registered on the right side

2.10 PROCEDURE ON COMPLETION OF REGISTERS

The following generally are observed while making an entry

In the registers;

Date:

The date entered in the register is that date on the inter branch

Voucher NOT the date making and entry in the register or

The date received at Head Office.

Inter branch Number (Voucher Number):

The voucher number confers on each inter branch transaction a

Unique identity and this must be captured correctly in the register.

Branch:

Name of branch on whom an inter branch is drawn (out ward inter branch) or

To whom a response is given (inward inter branch)

Transaction Code:

This should be the code describing the nature of transaction e.g. . 0001, 0002, etc.

Amount:

Care should be taken to ensure that amount is properly entered in the right column according to its nature, Debit or Credit.

Initials:

Initials of Officers signing inter branch vouchers (either origination or responding)

Remark:

This column is to indicate the date of response to an inter branch vouchers by a responding branch.

Procedures for Completing Branches' Analysis Register:

(A) Out ward Register:

(a) Pick up a carbonised inter branch voucher

(b) Write transaction date on voucher

(c) Ensure the pre-printed voucher number is

(d) Write a narration that properly describes the transaction

(e) Identify and enter the transaction code number

(f) Enter inter branch details in register

(g) The officers authorised to sign shall check the vouchers, compare the details on the voucher with those on the register before signing same.

(h) The initials column of the register is then completed

(B) Inward Register

On receipt of inter branch vouchers from an originating branch. the responding branch shall verify signatures on the vouchers and then take the following steps:

- Enter the following details into inward analysis register:
  (1) Voucher Number
  (2) Date on the Voucher

(3) Transaction Details

(4) Amount (whether Debits or Credit)

(5) Transaction code

(6) Remarks

Authorized Signatories check and compare entries in register with those on voucher.

Initial register and sign voucher as evidence of response.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGNS

3.1 ANALYSIS OF THE EXISTING SYSTEM

The present system of interbranch reconciliation is essentially manual in nature.

But suffice to mention that, interbranch reconciliation is the process of ascertaining at a given period the state of affairs of interbranch transaction. This take place at the head office operation department. The department acts as the clearing house for all the branches. With ever increasing branch network of the bank, the task of interbranch reconciliation has become tedious and time consuming venture.

3.2 PROBLEMS OF THE EXISTING SYSTEM

The following are the major problems with the existing inerbranch reconciliation system.

(i) Computations are always fill of errors

(ii) A lot of time and energy is wasted in preparation of interbranch reconciliation.

(iii) Manpower is wasted as more staff who could have been engaged in performing more profit oriented ventures for the bank are engaged in preparing the interbranch.

(iv) Data security is not ensured, interbranch system is suppose to be strictly confidential and should not be so exposed for all to see, but this cannot be guaranteed as many people have to be involved in interbranch preparation and reconciliation exercises.

(v) The manually produced interbranch vouchers are not readily legible due to corrections continuously made on them as mistakes are made.

(vi) The clerical work involved is also very enormous and cold jeopardise proper record keeping for future references.

(vii) Stationeries are wasted in producing the interbranch vouchers once it is so desired to recalculate whenever errors are made.

(viii) Many reports that are needed on the interbranch system, necessary for department/management decision-making cannot readily made available with the present system.

(ix) Long outstanding items

(x) Delay in communication

(xi) Suppression of debit items

(xii) Fraudulent transfers

(xiii) Discrepancies in the use of accounting codes.

3.3 INPUT SPECIFICATION

Interbranch Voucher contains Voucher Number, Date Sent, Originating

Branch, Responding Branch, Date Received, Transaction amount and Description. The input specification is as contained in the table below:

FIELD NAME	TYPE	LENGTH	DESCRIPTION
Voucher Number	Numeric	5	Voucher Number Code
Type	Character	1	Type of Voucher Code
Date	Character	8	Date of receipt of Voucher
Originating Branch	Character	20	Originating Branch Code
Date (1)	Date	8	Date of originating Code
Indicator	Character	1	Transaction indicator code
Responding Branch	Character	20	Responding Branch Code
Date (2)	Date	8	Responding Date Code
Transaction	Numeric	10, 2	Transaction amount Code

OUTPUT SPECIFICATION

The proposed program of seven (7) major parts is:

Add Record: it handles the date record of accounts

Update Record: this procedure allows data stored in database to be processed, organised etc., in the format required.

View Record: this gives a preview of the record (s) contained in the date base.

Report: the reports generated from all interbranch transaction are printed out in a summarised form.

Quit: it terminates the execution of the application system to return to DOS prompt or Data Base Management system.

3.5 SYSTEM DESIGN

The system is presented in modular form with the main menu consisting of options for programs on credit or debit interbranch creation in programs such as Add, Delete, print, view etc are well structured to enhance timelines and efficiency.

3.6 SOURCE PROGRAM DOCUMENTATION

The modules are referenced by the Main System Menu. The program is designed using dbase III+ software. To run this program, run dbase III+ and to go dot prompt, by pressing ESC key on loading the program. From the dot prompt, type DO MENU and press Enter. This takes you to the Main System Menu. The Main Menu enables you to enter your choice code. The modules and their codes are:

MODULES CODE

Add Record ..... (1)

Update Record..... (2)

View Record..... (3)

List Record..... (4)

Quit..... (5)

Each of the routine is described below:

Data Entry Routine

This routine creates;

Voucher number

Type of Voucher

Date Received

Originating Branch

Originating Date

Transaction Indicator

Responding Branch

Responding Date

Transaction Amount

This routine enable you enter the required data into the program.

Update Routine

This routine implements transaction all forms of modifications into existing record of transactions.

View Routine

Enable you to view the format in which the records are arrange, this routine does not allow for modifications.

Delete Routine

Once a record has become obsolete, that is, is no longer required, this routine enables you to delete from the program.

List Routine

This routine lists out various records contained in the program. In this program, it is captioned "WEMA BANK PLC, INTERBRANCH RECONCILIATION DISPLAY LIST". IT HAS UNIQUE FORMAT.

PRINT Report

This routine is responsible for producing reports of records either in hand or soft copies as required.

Ouit Routine

When all data been keyed into the system and all processes completed, next is to quit the record, this is done by choosing quit (7) which takes you back to dot prompt. To quit dbase III+ environment, at dot prompt type QUIT.

Limitations and capabilities of the program

The program works efficiently if correct data are keyed in. wrong input will generate wrong message, just as the saying goes; "Gabbage in Gabbage out." when in trouble go back to Main system menu and begin the process again.

The program has the capability of storing very large record, this is made possible by the use of DBMS, which helps in managing the database files.

CHAPTER FOUR

4.0 SYSTEMS IMPLEMENTATION

Successful systems after relative period of development are implemented and subsequently operated by users who obtain support and guidance from the systems Department as required. The task of management is then to see to a smooth transition of the systems into operational state and thereafter maximise the benefits from the implemented system by working with the users to monitor and adjust its performance. Implementation refers to the transition of a systems from construction stage to live usage.

4.1 WORKSTATION REQUIREMENT

This refers to computer hardware and software required for the smooth take of the computerisation exercise, which is as follows:

4.1.1 HARDWARE REQUIREMENT

To meet the desired objectives, that is, the computerisation of interbranch reconciliation system, it is proposed that it be run no:

(i) Pentium 150 systems processor with 24 MB RAM on Board with 2.36B HDD, the systems should also have CD-ROM drive and 3.5 drive.

(ii) Novelle 3.11 platform

4.1.2 SOFTWARE REQUIREMENTS: -

The system should be run on MS-DOS version 5.0 or higher version as clipper 5.0. clipper 5.0 is required to handle all difficult complication associate reconciliation thereby making interbranch reconciliation a make while experience.

4.2 SYSTEM TESTING

Before implementation and on acquired Hardware and other peripherals, the program will be loaded and test run for 4 weeks, to ascertain its efficacy and where necessary make modification.

4.3 SYSTEM CHANGEOVER

File conversion is a vital activity, which is sometimes underestimated. File conversion simply put is the process of converting the old file date into the form required by the new system. it is often referred to as part of changeover, infact, file conversion is often a complete and separate system task in itself, involving fact-finding, analysis, the design of clerical methods and computer processes among others.

The changeover from the old to the new systems may take place when:

- The system has been proven to the satisfaction of the systems analyst and the other implementation activities have been completed
- User managers are satisfied with the results of the system tests, staff training and reference manuals.
- The target date for changeover is due.

The changeover may be achieved in a number of ways. The most common methods are: Direct, parallel running, and staged changeover.

(A) DIRECT CHANGEOVER

This method is the bold move, which should be undertaken only when everyone concerned has confidence in the new system. When a direct changeover is planned, system tests and training should be comprehensive, and the changeover itself planned, in detail. This method is potentially the least expensive but the most risky.

(B) PARALLEL CHANGEOVER

This means processing current data by the old and systems to cross-check the result. It main attraction is that old systems is kept alive and operational until the new systems has been proved beyond all reasonable doubt. This systems allows the results of the new system to be compared with

CHAPTER FIVE

5.1 CONCLUSION

Computerization can be defined as planned and articulated change form a manual system to automation using computers. Computers were introduced to various organization for some reasons which include but not limited to:

(1) To enhance efficiency of service by cutting administrative cost, avoiding data duplication and offering greater management control and accountability.

(2) To improve the speed and efficiency in collection, manipulation, storage, reporting and dissemination of data. Computerization of interbranch reconciliation in Wema Bank is designed to eliminate boredom and stress associated with manual reconciliation of interbranch. The new system will also help to maintain all transaction all records on a continuous basis due to its updating nature. All transaction records are updated automatically and kept into a master file so to have a quick reach of any record needed at any at time on request.

The new system enables management have access to information on interbranch when required for decision making processes.

The world is dynamic so also is technology, therefore it is expected that this work can be improved upon in the nearest future to make it more versatile and meet the ever changing needs in Wema Bank plc.

5.2 RECOMMENDATIONS

The following recommendation are made to ensure that program achieve the set objectives:

(a) Duality of reporting system should give separate reports to Head office operations to ensure that debit items are not suppressed.

(b) A security coding system should be put in place to forestall fraudulent transfer associated with interbranch transactions.

(c) Follow-up prompt adjustment of items in transit should be ensured after reconciliation.

(d) Reconciliation exercise should be carried out promptly.

(e) To ensure that optimal results are reaped from this initiative current hardware should be procured as recommended elsewhere.

REFERENCE

(1) AKIN. F. (1995) Understanding and Using Microcomputers. At Lon books, Abuja- Nigeria

(2) Aiktas, A. Z. (1987) Structured Analysis and Design of Information System. Englewood Cliff N. J. Prentice, Toronto Canada.

(3) Alan S. (1989) Guide to Programming dbase 3+ and clipper 5.0 (2nd Edition)

(4) Badmus R. (1997) Lecture Note on Systems Analysis & Design. F. U. T. Minna (Unpublished).

(5) Cross. E.M (1983) How to Buy a Business Computer, Restin Va, Combers, USA.

(6) Dapo I. (1991) Accountancy for Banking Finance Tutors Ibadan.

(7) Ofodi. J. A. (1995) General Overview of Inter branch Accounting (Unpublished paper).

```
SET TALK OFF
SET ESCAPE OFF
SET BELL OFF
SET SAFETY OFF
SET STATUS OFF
SET SCOREBOARD OFF
USE BANK
STORE 0 TO R
DO WHILE .T.
  R = 0
  @1,1 TO 20,70 DOUB
  @2,15 SAY "**** MAIN SYSTEM MENU ****"
  @4,10 SAY "ADD RECORD                [1]"
  @6,10 SAY "UPDATE RECORD             [2]"
  @8,10 SAY "VIEW RECORD               [3]"
  @10,10 SAY "DELETE RECORD             [4]"
  @12,10 SAY "LIST RECORD               [5]"
  @14,10 SAY "PRINT REPORTS             [6]"
  @16,10 SAY "QUIT                      [7]"
  @17,8 TO 19,55
  @18,17 SAY "ENTER YOUR CHOICE     [ ]"
  @18,45 GET R PICT "9"
  READ
    DO CASE
       CASE R = 1
            DO ADD
       CASE R = 2
            DO EDIT
       CASE R = 3
            DO VIEW
       CASE R = 4
            DO DELETE
       CASE R = 5
            DO DISPLAY
       CASE R = 6
            DO PRINTER
       CASE R = 7
            EXIT
    ENDCASE
ENDDO

PROCEDURE ADD
SET TALK OFF
SET ESCAPE OFF
SET BELL OFF
SET SAFETY OFF
SET STATUS OFF
SET SCOREBOARD OFF
USE BANK
CLEAR
DO WHILE .T.
STORE 0 TO MVOUCH_NO, MTRANS
STORE SPACE(1) TO MINDIC, MTYPE
STORE SPACE(20) TO MORIGIN, MRESP
STORE CTOD("  /  /  ") TO MDATE, MDATE1, MDATE2
@1,23 SAY "DATA ENTRY ROUTINE"
@2,23 SAY "=================="
@4,4  SAY "VOUCHER NUMBER : "
@4,45 GET MVOUCH_NO PICT "9999"
@6,4  SAY "TYPE OF VOUCHER : "
```

```
@6,45 GET MTYPE
@8,4  SAY "DATE RECEIVED : "
@8,45 GET MDATE
@10,4  SAY "ORIGINATING BRANCH : "
@10,45 GET MORIGIN
@12,4  SAY "ORIGINATING DATE : "
@12,45 GET MDATE1
@14,4  SAY "TRANSACTION INDICATOR : "
@14,45 GET MINDIC
@16,4  SAY "RESPONDING DATE : "
@16,45 GET MDATE2
@18,4  SAY "TRANSACTION AMOUNT : "
@18,45 GET MTRANS
READ
S = .T.
@22,15 SAY "SAVE DATA (Y/N) ? " GET S PICT "Y"
READ
IF S
APPEND BLANK
REPL VOUCH_NO WITH MVOUCH_NO
REPL TYPE WITH MTYPE
REPL DATE WITH MDATE
REPL ORIGIN WITH MORIGIN
REPL DATE1 WITH MDATE1
REPL INDIC WITH MINDIC
REPL RESP WITH MRESP
REPL DATE2 WITH MDATE2
REPL TRANS WITH MTRANS
ELSE
@22,0
WAIT "   DATA NOT SAVED, PRESS ANY KEY ....."
CLEAR
ENDIF
MORE = .T.
@22,0
@22,15 SAY "ADD MORE DATA (Y/N) ? " GET MORE PICT "Y"
READ
IF MORE
   CLEAR
   LOOP
ELSE
   CLEAR
   EXIT
ENDIF
ENDDO
RETURN

PROCEDURE EDIT
SET TALK OFF
SET ESCAPE OFF
SET BELL OFF
SET SAFETY OFF
SET STATUS OFF
SET SCOREBOARD OFF
USE BANK
CLEAR
DO WHILE .T.
STORE 0 TO MVOUCH_NO, MTRANS
STORE SPACE(1) TO MINDIC, MTYPE
STORE SPACE(20) TO MORIGIN, MRESP
```

```
STORE CTOD("  /  /  ") TO MDATE, MDATE1, MDATE2
@1,10 SAY "VOUCH NUMBER TO EDIT " GET MVOUCH_NO PICT "9999"
READ
IF MVOUCH_NO = 0
   CLEAR
   RETURN
ENDIF
LOCATE FOR VOUCH_NO = MVOUCH_NO
IF .NOT. FOUND()
   WAIT "******  RECORD NOT FOUND, PRESS ANY KEY ******"
   CLEAR
   LOOP
ENDIF
MVOUCH_NO = VOUCH_NO
MTYPE = TYPE
MDATE = DATE
MORIGIN = ORIGIN
MDATE1 = DATE1
MINDIC = INDIC
MRESP = RESP
MDATE2 = DATE2
MTRANS = TRANS
@1,0
@1,23 SAY "DATA EDIT ROUTINE"
@2,23 SAY "=================="
@4,4  SAY "VOUCHER NUMBER : "
@4,45 GET MVOUCH_NO PICT "9999"
@6,4  SAY "TYPE OF VOUCHER : "
@6,45 GET MTYPE
@8,4  SAY "DATE RECEIVED : "
@8,45 GET MDATE
@10,4  SAY "ORIGINATING BRANCH : "
@10,45 GET MORIGIN
@12,4  SAY "ORIGINATING DATE : "
@12,45 GET MDATE1
@14,4  SAY "TRANSACTION INDICATOR : "
@14,45 GET MINDIC
@16,4  SAY "RESPONDING DATE : "
@16,45 GET MDATE2
@18,4  SAY "TRANSACTION AMOUNT : "
@18,45 GET MTRANS
READ
S = .T.
@22,4 SAY "SAVE EDITED DATA (Y/N) ? " GET S PICT "Y"
READ
IF S
REPL VOUCH_NO WITH MVOUCH_NO
REPL TYPE WITH MTYPE
REPL DATE WITH MDATE
REPL ORIGIN WITH MORIGIN
REPL DATE1 WITH MDATE1
REPL INDIC WITH MINDIC
REPL RESP WITH MRESP
REPL DATE2 WITH MDATE2
REPL TRANS WITH MTRANS
ELSE
@22,0
WAIT "   DATA NOT SAVED, PRESS ANY KEY ........."
CLEAR
ENDIF
```

```
MORE = .T.
@22,0
@22,10 SAY "EDIT MORE DATA (Y/N) ? " GET MORE PICT "Y"
READ
IF MORE
   CLEAR
   LOOP
ELSE
   CLEAR
   EXIT
ENDIF
ENDDO
RETURN


PROCEDURE DELETE
SET TALK OFF
SET ESCAPE OFF
SET BELL OFF
SET SAFETY OFF
SET STATUS OFF
SET SCOREBOARD OFF
USE BANK
CLEAR
DO WHILE .T.
STORE 0 TO MVOUCH_NO, MTRANS
STORE SPACE(1) TO MINDIC, MTYPE
STORE SPACE(20) TO MORIGIN, MRESP
STORE CTOD("  /  /  ") TO MDATE, MDATE1, MDATE2
@1,10 SAY "VOUCH NUMBER TO DELETE " GET MVOUCH_NO PICT "9999"
READ
LOCATE FOR VOUCH_NO = MVOUCH_NO
IF .NOT. FOUND()
   WAIT "******  RECORD NOT FOUND, PRESS ANY KEY ******"
   CLEAR
   LOOP
ENDIF
MVOUCH_NO = VOUCH_NO
MTYPE = TYPE
MDATE = DATE
MORIGIN = ORIGIN
MDATE1 = DATE1
MINDIC = INDIC
MRESP = RESP
MDATE2 = DATE2
MTRANS = TRANS
@1,0
@1,23 SAY "DATA DELETE ROUTINE"
@2,23 SAY "==================="
@4,4  SAY "VOUCHER NUMBER : "
@4,45 GET MVOUCH_NO PICT "9999"
@6,4  SAY "TYPE OF VOUCHER : "
@6,45 GET MTYPE
@8,4  SAY "DATE RECEIVED : "
@8,45 GET MDATE
@10,4  SAY "ORIGINATING BRANCH : "
@10,45 GET MORIGIN
@12,4  SAY "ORIGINATING DATE : "
@12,45 GET MDATE1
@14,4  SAY "TRANSACTION INDICATOR : "
```

```
@14,45 GET MINDIC
@16,4  SAY "RESPONDING DATE : "
@16,45 GET MDATE2
@18,4  SAY "TRANSACTION AMOUNT : "
@18,45 GET MTRANS
S = .T.
@22,4 SAY "DELETE THIS DATA (Y/N) ? " GET S PICT "Y"
READ
IF S
 DELETE
 PACK
ELSE
@22,0
WAIT "   DATA NOT DELETED, PRESS ANY KEY ....."
CLEAR
ENDIF
MORE = .T.
@22,0
@22,10 SAY "DELETE MORE DATA (Y/N) ? " GET MORE PICT "Y"
READ
IF MORE
   CLEAR
   LOOP
ELSE
   CLEAR
   EXIT
ENDIF
ENDDO
RETURN

PROCEDURE DISPLAY
SET TALK OFF
SET ESCAPE OFF
SET BELL OFF
SET SAFETY OFF
SET STATUS OFF
SET SCOREBOARD OFF
USE BANK
CLEAR
STORE 1 TO S,L
DO WHILE .NOT. EOF()
 IF L = 1
 @1,30 SAY "WEMA BANK PLC, AGBARA - LAGOS"
 @2,25 SAY "INTERBRANCH RECONCILIATION DISPLAY LIST"
 @3,1  SAY REPL("=",75)
 @4,1 SAY "S/N"
 @4,5 SAY "VOUCHER NO"
 @4,15 SAY "ORIGINATING BRANCH"
 @4,36 SAY "ORIGINATING DATE"
 @4,53 SAY "RESPONDING"
  4,64 SAY "TRANSACTION AMT"
  5,1 SAY REPL("=",75)
   =6
  NDIF
   ,1 SAY STR(S,3)
   ,5 SAY VOUCH_NO PICT "999999"
   ,16 SAY ORIGIN
    ,37 SAY DATE1
    ,54 SAY RESP
    ,64 SAY TRANS
```

```
L = L+1
@L,1  SAY REPL("-",75)
SKIP
L = L+1
S = S+1
IF L =20
   @L+1,0
   WAIT "  PRESS ANY KEY TO CONTINUE ........"
   @6,1 CLEAR TO 19,79
   L = 6
ENDIF
IF EOF()
   EXIT
ENDIF
ENDDO

@23,0
WAIT "***** END OF FILE REACHED, PRESS ANY KEY ******"
CLEAR
RETURN

PROCEDURE PRINTER
SET TALK OFF
SET ESCAPE OFF
SET BELL OFF
SET SAFETY OFF
SET STATUS OFF
SET SCOREBOARD OFF
SET DEVICE TO FILE "BANK.OUT"
USE BANK
CLEAR
STORE 1 TO S,L
DO WHILE .NOT. EOF()
IF L = 1
@1,30 SAY "WEMA BANK PLC, AGBARA - LAGOS"
@2,28 SAY "INTERBRANCH RECONCILIATION REPORT"
@3,1  SAY REPL("=",75)
@4,1 SAY "S/N"
@4,5 SAY "VOUCHER NO"
@4,15 SAY "ORIGINATING BRANCH"
@4,36 SAY "ORIGINATING DATE"
@4,53 SAY "RESPONDING"
@4,64 SAY "TRANSACTION AMT"
@5,1 SAY REPL("=",75)
L =6
ENDIF
@L,1 SAY STR(S,3)
@L,5 SAY VOUCH_NO PICT "999999"
@L,16 SAY ORIGIN
@L,37 SAY DATE1
@L,54 SAY RESP
@L,64 SAY TRANS
L = L+1
@L,1  SAY REPL("-",75)
SKIP
L = L+1
S = S+1
IF L =50
   EJECT
   L = 6
```

```
@14,45 GET MINDIC
@16,4  SAY "RESPONDING DATE : "
@16,45 GET MDATE2
@18,4  SAY "TRANSACTION AMOUNT : "
@18,45 GET MTRANS
S = .T.
@22,4 SAY "DELETE THIS DATA (Y/N) ? " GET S PICT "Y"
READ
IF S
 DELETE
 PACK
ELSE
@22,0
WAIT "   DATA NOT DELETED, PRESS ANY KEY ....."
CLEAR
ENDIF
MORE = .T.
@22,0
@22,10 SAY "DELETE MORE DATA (Y/N) ? " GET MORE PICT "Y"
READ
IF MORE
   CLEAR
   LOOP
ELSE
   CLEAR
   EXIT
ENDIF
ENDDO
RETURN

PROCEDURE DISPLAY
SET TALK OFF
SET ESCAPE OFF
SET BELL OFF
SET SAFETY OFF
SET STATUS OFF
SET SCOREBOARD OFF
USE BANK
CLEAR
STORE 1 TO S,L
DO WHILE .NOT. EOF()
IF L = 1
@1,30 SAY "WEMA BANK PLC, AGBARA - LAGOS"
@2,25 SAY "INTERBRANCH RECONCILIATION DISPLAY LIST"
@3,1  SAY REPL("=",75)
@4,1 SAY "S/N"
@4,5 SAY "VOUCHER NO"
@4,15 SAY "ORIGINATING BRANCH"
@4,36 SAY "ORIGINATING DATE"
@4,53 SAY "RESPONDING"
@4,64 SAY "TRANSACTION AMT"
@5,1 SAY REPL("=",75)
L =6
ENDIF
@L,1 SAY STR(S,3)
@L,5 SAY VOUCH_NO PICT "999999"
@L,16 SAY ORIGIN
@L,37 SAY DATE1
@L,54 SAY RESP
@L,64 SAY TRANS
```

```
ENDIF
IF EOF()
   EXIT
ENDIF
ENDDO
WAIT ""
SET DEVICE TO SCREEN
CLEAR
RETURN
```

WEMA BANK PLC, AGBARA - LAGOS
INTERBRANCH RECONCILIATION REPORT

N	VOUCHER NO	ORIGINATING BRANCH	ORIGINATING DATE	RESPONDING	TRANSACTI(
	3555	ORILE	08/10/99	AGBARA	543535.0(
	7488	ABULE EGBA	12/08/99	SABO	33544.0(
	5454	DOPEMU	04/23/99	DOPEMU	7867775.0(
	6546	OJUELEGBA	10/03/99	AGBARA	233133.0(
	9786	ISOLO	09/09/99	IGANMU	563435.0(
	6543	IGANMU	12/10/99	ORILE	562344.0(
7	6545	SABO	08/14/99	AGBARA	652345.0(

```
* Program Name.....: Slip.prg
* Purpose.....: To registered a NEW project.
* Note.....: Develop for

* Set working environment

SET TALK OFF
SET ESCAPE ON
SET DATE BRIT
SET BELL OFF
SET CONFIRM ON
SET SCORE OFF
SET STATUS OFF

* Set up database file that we will use
* to store registered project

* Declaring memory variable

DO WHILE .T.

CLEAR
Store space(40) to pro
Store space(25) to con

store ctod ("  /  /  ") to mdate1,mdate2,mdate4,mdate3
store 0 to c_sum,p_con,v_no,m
CLEAR
@ 1,25 to 3,60
@ 4,2 to 23,78 double
@ 2,28 say "V A L U A T I O N   S L I P:"
@ 5,20 say "Enter Valuation Number[0 to Exit]:-" get v_no pict "999"
read
Use SLIP.DBF
locate for vv_no = v_no
if found()
clear
@ 12,7 say "The Valuation number already exist"
wait space(7) + "Press any key to continue....."
loop
endif
if v_no = 0
clear
return
endif
@ 7,20 say "Enter Month of Valuation:-" get m pict "999"
@ 9,5 SAY "Project            :" get pro
@ 11,5 say "Main Contractor    :" get con
@ 13,5 say "Commencement Date  :" get mdate1 pict "99/99/99"
@ 13,44 say "Date on site      :" get mdate3 pict "99/99/99"
@ 15,5 say "Completation Date  :" get mdate2 pict "99/99/99"
@ 17,5 say "Contract Sum       :" get c_sum pict ;
  "999,999,999,999.99"
@ 15,44 say "Date of Issue     :" get mdate4 pict "99/99/99"
READ

* Verify if data entered is correct

store space(1) to repl
@ 22,22 say "Is the above data entry correct [Y/N]"  get repl ;
                pict "!"
READ

IF repl = "N" .or. repl = "n"
   clear
   loop
ENDIF
```

```
APPEND BLANK
REPLACE CONTRACT WITH CON,PROJECT WITH PRO
REPLACE DATE1 WITH MDATE1,DATE2 WITH MDATE2
REPLACE DATE3 WITH MDATE3,DATE4 WITH MDATE4
REPLACE CC_SUM WITH C_SUM,PP_SUM WITH P_CON
REPLACE VV_NO WITH V_NO
  close all
  clear
@ 9,5 SAY "If you are ready to Enter Other valuation data,"
@ 10,5 SAY "select appropriate option in the MAIN MENU    "
wait space(5) + "Press any key to Continue..... "
return
Enddo
```

```
* Program Name.....: back_R.PRG
* Purpose..........: To Generate backup file for
*                    databse records
* Note.............: Called from v_main

* Memo 1
* Setting working environment.
CLOSE ALL
CLEAR
SET TALK OFF
SET ESCAPE ON
SET DATE BRIT
SET BELL OFF
 mval = " "
 @ 10,10 TO 12,50 DOUBLE
 @ 11,13 SAY "Backup all DBF to A:[Y/N] " get mval pict "!"
 read
 if upper(mval) = "N"
   clear
   return
 endif

@14,10 say "Insert Diskette in drive A:, Press Enter when ready"
wait " "
run copy slip.dbf a:
run copy pre.dbf a:
run copy sub.dbf a:
run copy con1.dbf a:
run copy block.dbf a:
run copy roof.dbf a:
run copy cap.dbf a:
run copy metal.dbf a:
run copy elect.dbf a:
run copy mech.dbf a:
run copy fur.dbf a:
run copy glaz.dbf a:
run copy plate.dbf a:
run copy ext.dbf a:
run copy msite.dbf a:
run copy var.dbf a:
run copy nomi1.dbf a:
run copy nomi2.dbf a:
run copy flc.dbf a:
@ 19,2 say " "
@ 20,5 say "Backup sucessful, press ny key to continue..."
wait " "
return
```

```
* Program Name:... cap.prg
* Purpose:..... Allow the user's to enter data
*              relating to capentary work on site.
* Note:.....Called from main_c.prg

* Set up the working environment.
SET SCOREBOARD OFF
SET STATUS OFF
SET TALK OFF
SET BELL OFF
SET DATE BRITISH
SET ESCAPE ON

* Initialize memory variable

  store 0 to sn
  store 7 to row
  store space(15) to name
  store 0 to amm,rate,valu
  store 0 to tot
  store 0 to msub,mmonth
do while .t.
 clear
 set color to gr+/b+
 @ 12,10 TO 14,50 DOUBLE
 @ 13,11 SAY "Enter Valuation Number:-" get msub pict "999"
 read
 USE SLIP.DBF
 locate for vv_no = msub
 if .not. found()
  @ 17,10 say "Match record not found"
  @ 18,10 say "Press any key to continue...."
  wait space(10) + " "
  close databases
  return
 endif
* Test If data already exist
  use cap
  locate for m_cap = msub
  if found()
   clear
   @ 12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue..."
   clear
   loop
  endif
* IF RECORD HAS BEEN ENTERED
 clear
 @ 2,5 to 24,70
 @ 1,20 SAY "C a p e n t a r y  W o r k"
 @5,7 say "No"
 @ 5,14 say "[Enter DONE to exit]"
 @ 6,14 say "Description"
 @ 6,33 say "Amount"
 @ 6,49 say "% of work done"
 @ 6, 65 say "Result"
 sn = sn + 1
 row = row + 1
Do while .t.
 @ row,7 say ltrim(str(sn))
 @ row,14 get name
 read
 If upper(name) = "DONE"
  @row+6,10 say "Total:"
  @row+6,58 say ltrim(str(tot,14,2))
  wait space(10) + "Press any key to continue...."
  clear
```

```
 return
 Endif
 @ row,33 get amm pict "99,999,999.99"
 @ row,49 get rate pict "999"
 read
 ans = " "
 @ row+1,33 say "Correct [Y/N]" get ans
 read
 if upper(ans) = "N"
  loop
 endif
 @ row+1,33 say space(20)
valu = ( (rate/100) * amm )
tot = ( tot + valu )
@ row,65 say ltrim(str(valu,8,2))
sn = sn + 1
row = row + 2
if row > 15
 @ 7,7 CLEAR TO 20,78
 row = 7
 loop
endif
append blank
replace m_cap with msub
replace month with mmonth
replace t_cap with tot
store 0 to rate,amm
name = space(15)
loop
enddo
```

```
* Program Name:... elect.prg
* Purpose:..... Allow the user's to enter data
*              relating to electrical work on site.
* Note:..... Called from main_c.prg

* Set up the working environment.
SET SCOREBOARD OFF
SET STATUS OFF
SET TALK OFF
SET BELL OFF
SET DATE BRITISH
SET ESCAPE ON

* Initialize memory variable

  store 0 to sn
  store 7 to row
  store space(15) to name
  store 0 to amm,rate,valu
  store 0 to tot
  store 0 to msub,mmonth
do while .t.
 clear
 set color to gr+/b+
 @ 12,10 TO 14,50 DOUBLE
 @ 13,11 SAY "Enter Valuation Number:-" get msub pict "999"
 read
 USE SLIP.DBF
 locate for vv_no = msub
 if .not. found()
  @ 17,10 say "Match record not found"
  @ 18,10 say "Press any key to continue...."
  wait space(10) + " "
  close databases
  return
 endif
* Test If data already exist
  use elect
  locate for elect= msub
  if found()
   clear
   @ 12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue..."
   clear
   loop
  endif
* IF RECORD HAS BEEN ENTERED
 clear
 @ 2,5 to 24,70
 @ 1,20 SAY "E l e c t r i c a l  W o r k"
 @5,7 say "No"
 @ 5,14 say "[Enter DONE to exit]"
 @ 6,14 say "Description"
 @ 6,33 say "Amount"
 @ 6,49 say "% of work done"
 @ 6, 65 say "Result"
 sn = sn + 1
 row = row + 1
Do while .t.
 @ row,7 say ltrim(str(sn))
 @ row,14 get name
 read
 If upper(name) = "DONE"
  @row+6,10 say "Total:"
  @row+6,69 say ltrim(str(tot,14,2))
  wait space(10) + "Press any key to continue...."
  clear
```

```
 return
Endif
@ row,33 get amm pict "99,999,999.99"
@ row,49 get rate pict "999"
read
ans = " "
@ row+1,33 say "Correct [Y/N]" get ans
read
if upper(ans) = "N"
 loop
endif
@ row+1,33 say space(20)
valu = ( (rate/100) * amm )
tot = ( tot + valu )
@ row,65 say ltrim(str(valu,14,2))
sn = sn + 1
row = row + 2
if row > 15
 @ 7,7 CLEAR TO 20,75
 row = 7
 loop
endif
append blank
replace elect with msub
replace month with mmonth
replace t_elect with tot
store 0 to rate,amm
name = space(15)
loop
enddo
```

```
* Program Name.....: FLC.prg
* Purpose.....: Allow the users to enter data
*                relating to Project Fluctation

* Memo 1
* Setting working environment.

SET TALK OFF
SET ESCAPE ON
SET DATE BRIT
SET BELL OFF
SET SCORE OFF
SET STATU OFF
store 0 to mvalnum,mmonth

* Check if record has beeb entered into valuation slip

do while .t.
 clear
 @ 12,10 TO 14,50 DOUBLE
 @ 13,11 SAY "Enter Valuation Number:-" get mvalnum pict "999"
 read
 USE SLIP.DBF
 locate for vv_no = mvalnum
 if .not. found()
  @ 16,10 say "Match record not found in valuation slip"
  wait space(10) + " "
  close databases
  return
 endif
* test if valuation number alreay exist
  use flc.dbf
  locate for flc = mvalnum
  if found()
   clear
   @12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue...."
   clear
   return
  endif
* If record has been entered

 clear
 @ 2,5 to 24,65
 @ 1,28 SAY "F l u c t u a t i o n  D a t a"
 @ 3,10 SAY "Month          :-" get mmonth pict "99"
 @ 5,35 say "[Enter 0 to Exit]"
* Initialize memory variable

   store 0 to sn
   store 6 to row
   store 0 to valu1,tot

   sn = sn + 1

* display heading 2

   @ 5,7 say "No"
   @ 5,14 say "Fluctuation"
   @ 5,25 say "Value"
   row = row + 1
 Do while .t.
  @ row,7 say ltrim(str(sn))
  @ row,10 say "Fluctation No  "
  @ row,23 say ltrim(str(sn))
  @ row,25 get valu1 pict "999,999,999,999.99"
  read
```

```
* Test exit condition
  If valu1 = 0
  @ row+2,10 say "Total: "
  @ row+2,25 say ltrim(str(tot,14,2))
  @ row+3,25 say "Press any key to continue...."
  wait ""
  return
  endif
  ans = " "
  @ row,40 say "Is this correct [Y/N]" get ans pict "!"
  read
   If UPPER(ans) = "N"
     store 0 to valu1
     loop
    endif
    tot = (tot + valu1)
    APPEND BLANK
    replace flc with MVALNUM
    replace t_flc with valu1
    replace MONTH with MMONTH
    valu1 = 0
    sn = sn + 1
    row = row + 2
    if row > 17
     @ 7,7 CLEAR TO 23,62
     row = 7
    endif
    loop
    line = 9
 enddo
return
enddo
```

```
* Program Name:... fur.prg
* Purpose:..... Allow the user's to enter data
*              relating to furnishing work on site.
* Note:.....Called from main_c.prg

* Set up the working environment.
SET SCOREBOARD OFF
SET STATUS OFF
SET TALK OFF
SET BELL OFF
SET DATE BRITISH
SET ESCAPE ON

* Initialize memory variable

  store 0 to sn
  store 7 to row
  store space(15) to name
  store 0 to amm,rate,valu
  store 0 to tot
  store 0 to msub,mmonth
do while .t.
 clear
 set color to gr+/b+
 @ 12,10 TO 14,50 DOUBLE
 @ 13,11 SAY "Enter Valuation Number:-" get msub pict "999"
 read
 USE SLIP.DBF
 locate for vv_no = msub
 if .not. found()
  @ 17,10 say "Match record not found"
  @ 18,10 say "Press any key to continue...."
  wait space(10) + " "
  close databases
  return
 endif
* Test If data already exist
  use fur
  locate for fur = msub
  if found()
   clear
   @ 12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue..."
   clear
   loop
  endif
* IF RECORD HAS BEEN ENTERED
 clear
 @ 2,5 to 24,70
 @ 1,20 SAY "F u r n i s h i n g  W o r k"
 @5,7 say "No"
 @ 5,14 say "[Enter DONE to exit]"
 @ 6,14 say "Description"
 @ 6,33 say "Amount"
 @ 6,45 say "% of work done"
 @ 6, 62 say "Result"
 sn = sn + 1
 row = row + 1
Do while .t.
 @ row,7 say ltrim(str(sn))
 @ row,14 get name
 read
 If upper(name) = "DONE"
  @row+6,10 say "Total:"
  @row+6,62 say ltrim(str(tot,14,2))
  wait space(10) + "Press any key to continue...."
  clear
```

```
 return
Endif
@ row,33 get amm pict "99,999,999.99"
@ row,45 get rate pict "999"
read
ans = " "
@ row+1,33 say "Correct [Y/N]" get ans
read
if upper(ans) = "N"
 loop
endif
@ row+1,33 say space(20)
valu = ( (rate/100) * amm )
tot = ( tot + valu )
@ row,58 say ltrim(str(valu,14,2))
sn = sn + 1
row = row + 2
if row > 15
 @ 7,7 CLEAR TO 20,75
 row = 7
 loop
endif
append blank
replace fur with msub
replace month with mmonth
replace t_fur with tot
store 0 to rate,amm
name = space(15)
loop
enddo
```

```
* Program Name:... glaz.prg
* Purpose:..... Allow the user's to enter data
*              relating to glazing work on site.
* Note:.....Called from main_c.prg

* Set up the working environment.
SET SCOREBOARD OFF
SET STATUS OFF
SET TALK OFF
SET BELL OFF
SET DATE BRITISH
SET ESCAPE ON

* Initialize memory variable

  store 0 to sn
  store 7 to row
  store space(15) to name
  store 0 to amm,rate,valu
  store 0 to tot
  store 0 to msub,mmonth
do while .t.
 clear
 set color to gr+/b+
 @ 12,10 TO 14,50 DOUBLE
 @ 13,11 SAY "Enter Valuation Number:-" get msub pict "999"
 read
 USE SLIP.DBF
 locate for vv_no = msub
 if .not. found()
  @ 17,10 say "Match record not found"
  @ 18,10 say "Press any key to continue...."
  wait space(10) + " "
  close databases
  return
 endif
* Test If data already exist
  use glaz
  locate for glaz = msub
  if found()
   clear
   @ 12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue..."
   clear
   loop
  endif
* IF RECORD HAS BEEN ENTERED
 clear
 @ 2,5 to 24,70
 @ 1,20 SAY "G l a z i n g W o r k"
 @5,7 say "No"
 @ 5,14 say "[Enter DONE to exit]"
 @ 6,14 say "Description"
 @ 6,33 say "Amount"
 @ 6,45 say "% of work done"
 @ 6, 62 say "Result"
 sn = sn + 1
 row = row + 1
Do while .t.
 @ row,7 say ltrim(str(sn))
 @ row,14 get name
 read
 If upper(name) = "DONE"
  @row+6,10 say "Total:"
  @row+6,62 say ltrim(str(tot,14,2))
  wait space(10) + "Press any key to continue...."
  clear
```

```
 return
 Endif
 @ row,33 get amm pict "99,999,999.99"
 @ row,45 get rate pict "999"
 read
 ans = " "
 @ row+1,33 say "Correct [Y/N]" get ans
 read
 if upper(ans) = "N"
  loop
 endif
 @ row+1,33 say space(20)
valu = ( (rate/100) * amm )
tot = ( tot + valu )
@ row,58 say ltrim(str(valu,14,2))
sn = sn + 1
row = row + 2
if row > 15
 @ 7,7 CLEAR TO 20,75
 row = 7
 loop
endif
append blank
replace glaz with msub
replace month with mmonth
replace t_glaz with tot
store 0 to rate,amm
name = space(15)
loop
enddo
```

```
* Program Name:.... Helpo.prg
* Purpose:..... Teach the user's the BASIC on
* ..... how to USE the package
* Note:..... Developed by.


* Set working environment

  SET TALK OFF
  SET STATUS OFF
  SET SCORE OFF
  SET ESCAPE ON


@ 1,19 say "H E L P  I N F O R M A T I O N  M E N U"

* Type in the help information
opt = " "
TEXT

    This PACKAGE is titled PROJECT VALUATION SYSTEM. It is develop
    with simple logic and provide efficient and easy way of valueing
    architectural project.

    The whole package is divided into Four(4) menu viz:-

      1.  The input Menu
      2.  The Maintenance Menu
      3.  The Report Menu and
      4.  The information Menu

    Each of the menu above will be briefly analyse to assist you in
    using the PACKAGE
    To get HELP on any of the menu select option(1-4),or X to exit"
endtext
do while .t.
@ 20,5 say "Enter Option: :"
@ 20,19 get opt
read
do case opt
   case opt = "1"
      clear
      do inpo
   case opt = "2"
       clear
      do may
   case opt = "3"
      do ree
   case opt = "4"
      do info
   case UPPER(opt) = "X"
      return
otherwise
@19,5 say "Invalid option, Select from 1-4,or X"
wait space(5) + "Press any key to continue"
@19,5 say "                                    "
loop
endcase
enddo



clear
text

    1. The INPUT MENU
       ~~~~~~~~~~~~~~~
```

```
    The objectives of a data processing system is to derived
    an output that will be useful for decision making.
    But OUTPUT in itself can not be generated until some data
    has been entered into the system.
    The INPUT MENU of this program therefore provide you a way
    of getting your data into the data processing system.
    Various data to be imputed are display in the first menu
    of the control program. To imput any of the data set you
    will have to choose appropriate option represented by the
    alpabetical character attached to it. For example to input
    your data on PRELIMINARIES, you will select option B from
    the control menu and the preliminary bill will be displayed
    for your data entry. Note that selecting option Q will
    terminate the program.
endtext
@23,5 say ""
wait
@2,0 clear to 24,79
@1,0 say ""
text
    1.1 The input design.
      ----------------
    This program apply data validation check to the procedure
    of imputing data. For example you are asked to confirm any
    time you enter data relating to any of the input data set
    if the data is correct or not. If I enter variation data
    now for example. It will display a message:-
    "Is this correct [Y/N" waiting for me to enter appropriate
    option. If Y is selected, the data is considered valid and
    saved into the appropriate Database, but if otherwise, the
    user will be prompted to re-entered data.  But before you
    can even be allowed to entered data, you will be required
    to enter the project valuation number of the record you
    wish to enter. This is use as a record key to search the
    database file in which records of project at hand is stored
    to confirm if project slip has been entered or not.
    If you enter a valuation number that can not be found
    that will an error, a message will be displayed to tell
    you that valuation number does not exist
endtext
@22,5 say ""
wait space(5) + "Press any key to return to help menu"
clear
do helpo
```

```
* Program Name:... M_site.prg
* Purpose:..... Allow the user's to enter data
*              relating to Material on site.
* Note:.....Called fron v_main

* Set up the working environment.
SET SCOREBOARD OFF
SET STATUS OFF
SET TALK OFF
SET BELL OFF
SET DATE BRITISH
SET ESCAPE ON

* Initialize memory variable
  CLOSE ALL
  store 0 to sn
  store 7 to row
  store space(15) to name
  store space(10) to unit
  store 0 to qty,rate,valu
  store 0 to tot
  store 0 to msite,mmonth

* Check if project has been registered

do while .t.
 store 0 to mvalnum
 clear
 set color to gr+/b+
 @ 12,10 TO 14,50 DOUBLE
 @ 13,11 SAY "Enter Valuation Number:-" get msite pict "999"
 read
 SELECT 1
 USE SLIP.DBF
 locate for vv_no = msite
 if .not. found()
  @ 17,10 say "Match record not found"
  @ 18,10 say "Press any key to continue...."
  wait space(10) + " "
  close databases
  return
 endif
* Test If data already exist
  SELECT 2
  use msite
  locate for m_site = msite
  if found()
   clear
   @ 12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue..."
   clear
   loop
  endif

* IF RECORD HAS BEEN ENTERED

 clear
 @ 2,1 to 24,79
 @ 1,20 SAY "MATERIAL ON SITE DATA"
 @ 5,2 say "No"
 @ 5,6 say "[Enter DONE to exit]"
 @ 6,6 say "Materials"
 @ 6,23 say "Quantity"
 @ 6,40 say "Unit"
 @ 6,52 say "Rate"
 @ 6,67 say "Total"
 sn = sn + 1
```

```
 row = row + 1
Do while .t.
  @ row,2 say ltrim(str(sn))
  @ row,6 get name
  read
* Test exit condition
  If upper(name) = "DONE"
   @row+6,10 say "Total:"
   @row+6,67 say ltrim(str(tot,16,2))
   wait space(10) + "Press any key to continue...."
   clear
   return
  Endif
  @ row,23 get qty pict "99,999,999.99"
  @ row,40 get unit
  @ row,52 get rate pict "99,999,999.99"
  read
  ans = " "
  @ row+1,33 say "Correct [Y/N]" get ans
  read
  if upper(ans) = "N"
    loop
  endif
  @ row+1,33 say space(30)
  valu = ( qty * rate )
  tot =  (tot + valu)
  @ row,67 say ltrim(str(valu,16,2))
  sn = sn + 1
  row = row + 2
  if row > 17
   @ 7,2 CLEAR TO 23,79
   row = 7
  endif
  APPEND BLANK
  replace m_site with msite
  replace T_SITE WITH valu
  replace Nsite with NAME
  replace qt with qty
  replace rool with rate
  replace un with unit
  name = space(15)
  unit = space(10)
  store 0 to valu,qty,rate
  loop
enddo
```

```
* Program Name.....: MAIN_R.PRG
* Purpose.....: Generate report on main contract's work
* Note.....: Called from v_main

* Memo 1
* Setting working environment.
CLOSE ALL
CLEAR
SET TALK OFF
SET ESCAPE ON
SET DATE BRIT
SET BELL OFF
SET SCORE OFF
SET STATU OFF
store 0 to tot1
do while .t.
 set color to gr+/b+
 clear
 mval = 0
 @ 10,10 TO 12,50 DOUBLE
 @ 11,13 SAY "Enter Valuation Number  :-" get mval pict "999"
 read
 USE slip
 locate for vv_no = mval
 if .not. found()
  @ 17,10 say "Valuation number does not exist"
  @ 18,10 say "Press any key to continue....."
  wait space(10) + " "
  close databases
  return
 endif
 clear
 @ 1,20 SAY "Main Contractor's Work"
 @ 2,20 say "This report is produce on "+dtoc(date())
 @ 3,5 say "Valution no :"+ltrim(str(mval))
 @ 4,50 say "Value"
 @ 5,5 say repl("-",55)
 @ 6,5 say "Substructure"
 @ 7,5 say "Concrete Work"
 @ 8,5 say "Block work"
 @ 9,5 say "Roofing"
 @ 10,5 say "Carpentary & Joinery"
 @ 11,5 say "Metalwork"
 @ 12,5 say "Electrical Installation"
 @ 13,5 say "Mechanical Installation"
 @ 14,5 say "Furnishing"
 @ 15,5 say "Glazings"
 @ 16,5 say "Plating & Decoration"
 @ 17,5 say "External Work"
 store 0 to int6,int7,int8,int9,int10,int11,int12,int13
 store 0 to int14,int15,int16,int17
 l = 6
use sub
do while .not. eof()
  if m_sub<> mval
    skip
    loop
  endif
  int6 = int6 + t_sub
  skip
  loop
enddo
@ l,50 say int6
close databases
use con1
do while .not. eof()
 if m_con <> mval
```

```
    skip
    loop
  endif
  int7 = int7 + t_con
  skip
  loop
enddo
@ l+1,50 say int7
close databases
use block
do while .not. eof()
  if m_block <> mval
    skip
    loop
  endif
  int8 = int8 + t_block
  skip
  loop
enddo
@ l+2,50 say int8
close database
use roof
do while .not. eof()
  if roof <> mval
    skip
    loop
  endif
  int9 = int9 + t_roof
  skip
  loop
enddo
@ l+3,50 say int9
close databases
use cap
do while .not. eof()
   if m_cap <> mval
     skip
     loop
   endif
   int10 = int10 + t_cap
   skip
   loop
enddo
@ l+4,50 say int10
close databases
use metal
do while .not. eof()
   if metal <> mval
      skip
      loop
   endif
   int11 = int11 + t_metal
   skip
   loop
enddo
@ l+5,50 say int11
close databases
use elect
do while .not. eof()
   if elect <> mval
      skip
      loop
   endif
   int12 = int12 + t_elect
   skip
   loop
enddo
@ l+6,50 say int12
```

```
close databases
use mech
do while .not. eof()
  if mech <> mval
    skip
    loop
  endif
  int13 = int13 + t_mech
  skip
  loop
enddo
@ l+7,50 say int13
close databases
use fur
do while .not. eof()
  if fur <> mval
    skip
    loop
  endif
  int14 = int14 + t_fur
  skip
  loop
enddo
@ l+8,50 say int14
close databases
use glaz
do while .not. eof()
  if glaz <> mval
    skip
    loop
  endif
  int15 = int15 + t_glaz
  skip
  loop
enddo
@ l+9,50 say int15
close databases
use plate
do while .not. eof()
  if plate <> mval
    skip
    loop
  endif
  int16 = int16 + t_plate
  skip
  loop
enddo
@ l+10,50 say int16
close databases
use ext
do while .not. eof()
  if ext <> mval
    skip
    loop
  endif
  int17 = int17 + t_ext
  skip
  loop
enddo
@ l+11,50 say int17
close databases
int19 = int6+int7+int8+int9+int10+int11+int12+int13+int14+;
int15+int16+int17
@ l+12,5 say repl("-",60)
@ l+13,5 say "Total:"
@ l+14,5 say repl("-",60)
@ l+13,50 say int19
@ l+15,5 say "Press any key to Return to Main Menu"
```

```
wait space(5)
close all
return


set talk off
clear
text
   2. MAINTENANCE
      ************
      Maintenance Menu provides you with three option,

        * How to Void record
        * About Help
        * How to Make Backup
     2.1 To void record.
        **************
        It is good to void all the record in the databases
        file before the program will be used for project.
        To do this you will select void record option
        from the control program. Then the system authomatically
        void all the record in every database file, and
        the system is ready for another project
        if this is not done there may be error in the process
        of using unvoid database file for a new project.
     2.2 Copying databases.
        *****************
        There will be a warning to you that before you void your
        databases, you need to make a BACKUP of the
        file using the option the maintenance menu.
endtext
@22,5 say " "
wait space(5) + "Press any key to continue....."
clear
@ 2,0 say " "
text

     2.2 Help
        ****
        This provides you help on how to use the application.
        Most especially, the BASIC about the VALUATION SYSTEM
        Press J from main menu to get help.
     2.3 Backup
        ******
        The system also provides you with the facility
        to make backup of you record in database file. Infact
        you need to make backup of your database before you
        void them so that you can have their record for future
        use. To make backup of your record, you will select
        option Z from the control program, make sure that the
        message the floppy disk(3.5) is inseted to drive A:
        and press enter key. You can label the disk to be able
        to know the project records it contains.


endtext
@22,5 SAY " "
wait space(5) + "Press any key to continue...."
clear
@2,0 say " "
text

     Know this
     =========
     You can make use of your record stored in the database file
     any other time you want to use them in the VALUATION SYSTEM
     to do this you need to copy them back into the directory in
     which your programs are install (for both the (.PRG) and
```

```
    (.DBF) must be in the same directory before the program
    can work).
endtext
@22,5 say " "
wait space(5) + "Press any key to return to help menu"
do helpo
```

```
* Program Name:... metal.prg
* Purpose:..... Allow the user's to enter data
*              relating to metal work on site.
* Note:..... Called from main_c.prg

* Set up the working environment.
SET SCOREBOARD OFF
SET STATUS OFF
SET TALK OFF
SET BELL OFF
SET DATE BRITISH
SET ESCAPE ON

* Initialize memory variable

  store 0 to sn
  store 7 to row
  store space(15) to name
  store 0 to amm,rate,valu
  store 0 to tot
  store 0 to msub,mmonth
do while .t.
 clear
 set color to gr+/b+
 @ 12,10 TO 14,50 DOUBLE
 @ 13,11 SAY "Enter Valuation Number:-" get msub pict "999"
 read
 USE SLIP.DBF
 locate for vv_no = msub
 if .not. found()
  @ 17,10 say "Match record not found"
  @ 18,10 say "Press any key to continue...."
  wait space(10) + " "
  close databases
  return
 endif
* Test If data already exist
  use metal
  locate for metal = msub
  if found()
   clear
   @ 12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue..."
   clear
   loop
  endif
* IF RECORD HAS BEEN ENTERED
 clear
 @ 2,5 to 24,70
 @ 1,20 SAY "M e t a l  W o r k"
 @5,7 say "No"
 @ 5,14 say "[Enter DONE to exit]"
 @ 6,14 say "Description"
 @ 6,33 say "Amount"
 @ 6,49 say "% of work done"
 @ 6, 65 say "Result"
 sn = sn + 1
 row = row + 1
Do while .t.
 @ row,7 say ltrim(str(sn))
 @ row,14 get name
 read
 If upper(name) = "DONE"
  @row+6,10 say "Total:"
  @row+6,62 say ltrim(str(tot,14,2))
  wait space(10) + "Press any key to continue...."
  clear
  return
```

```
 Endif
 @ row,33 get amm pict "99,999,999.99"
 @ row,49 get rate pict "999"
 read
 ans = " "
 @ row+1,33 say "Correct [Y/N]" get ans
 read
 if upper(ans) = "N"
  loop
 endif
 @ row+1,33 say space(20)
valu = ( (rate/100) * amm )
tot = ( tot + valu )
@ row,65 say ltrim(str(valu,14,2))
sn = sn + 1
row = row + 2
if row > 15
 @ 7,7 CLEAR TO 20,79
 row = 7
 loop
endif
append blank
replace metal with msub
replace month with mmonth
replace t_metal with tot
store 0 to rate,amm
name = space(15)
loop
enddo
```

```
* Program Name:... metal.prg
* Purpose:..... Allow the user's to enter data
*              relating to metal work on site.
* Note:..... Called from main_c.prg

* Set up the working environment.
SET SCOREBOARD OFF
SET STATUS OFF
SET TALK OFF
SET BELL OFF
SET DATE BRITISH
SET ESCAPE ON

* Initialize memory variable

  store 0 to sn
  store 7 to row
  store space(15) to name
  store 0 to amm,rate,valu
  store 0 to tot
  store 0 to msub,mmonth
do while .t.
 clear
 set color to gr+/b+
 @ 12,10 TO 14,50 DOUBLE
 @ 13,11 SAY "Enter Valuation Number:-" get msub pict "999"
 read
 USE SLIP.DBF
 locate for vv_no = msub
 if .not. found()
  @ 17,10 say "Match record not found"
  @ 18,10 say "Press any key to continue...."
  wait space(10) + " "
  close databases
  return
 endif
* Test If data already exist
  use metal
  locate for metal = msub
  if found()
   clear
   @ 12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue..."
   clear
   loop
  endif
* IF RECORD HAS BEEN ENTERED
 clear
 @ 2,5 to 24,70
 @ 1,20 SAY "M e t a l  W o r k"
 @5,7 say "No"
 @ 5,14 say "[Enter DONE to exit]"
 @ 6,14 say "Description"
 @ 6,33 say "Amount"
 @ 6,49 say "% of work done"
 @ 6, 65 say "Result"
 sn = sn + 1
 row = row + 1
Do while .t.
 @ row,7 say ltrim(str(sn))
 @ row,14 get name
 read
 If upper(name) = "DONE"
  @row+6,10 say "Total:"
  @row+6,62 say ltrim(str(tot,14,2))
  wait space(10) + "Press any key to continue...."
  clear
```

```
 return
 Endif
 @ row,33 get amm pict "99,999,999.99"
 @ row,49 get rate pict "999"
 read
 ans = " "
 @ row+1,33 say "Correct [Y/N]" get ans
 read
 if upper(ans) = "N"
  loop
 endif
 @ row+1,33 say space(20)
valu = ( (rate/100) * amm )
tot = ( tot + valu )
@ row,65 say ltrim(str(valu,14,2))
sn = sn + 1
row = row + 2
if row > 15
 @ 7,7 CLEAR TO 20,79
 row = 7
 loop
endif
append blank
replace metal with msub
replace month with mmonth
replace t_metal with tot
store 0 to rate,amm
name = space(15)
loop
enddo
```

```
* Program Name.....: Nomi2.prg
* Purpose.....: Allow for Nominated sub contract
*                data entery
* Note.....: called form v_main

* Memo 1

* Setting working environment.
CLEAR
SET TALK OFF
SET ESCAPE ON
SET DATE BRIT
SET BELL OFF
SET SCORE OFF
SET STATU OFF

* Initialize memory variable
store 0 to mnomi,mmonth

* Check if valuation slip has been entered
do while .t.
 clear
 @ 10,10 TO 15,50 DOUBLE
 @ 12,12 SAY "Enter Valuation Number:-" get mnomi pict "999"
 @ 14,12 say "Month of Valuation    :-" get mmonth pict "99"
 read
 USE SLIP.DBF
 locate for vv_no = mnomi
 if .not. found()
  @ 16,10 say "This is not right, you have not entered data into"
  @ 17,10 say "Valuation slip for this Valuations"
  @ 18,10 say "Please do that now "
  wait space(10) + " "
  close databases
  return
 endif

* If data already exist
  use nomi2.dbf
  locate for nomi_num = mnomi
  if found()
   clear
   @ 12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue..."
   clear
   loop
  endif
clear
 @ 2,5 to 24,79
 @ 1,20 SAY "Nominated Sub Contract Data"
* Initialize memory variable
   name=space(15)
   store 0 to sn
   store 6 to row
   store 0 to valu1,tot

   sn = sn + 1

* display heading 2

   @ 5,7 say "No"
   @ 5,14 say "Name[Enter DONE to exit]"
   @ 5,41 say "Value"

   row = row + 1

* Declare a logical variable to test loop condition
```

```
Do while .t.

    @ row,7 say ltrim(str(sn))
    @ row,14 get NAME pict "@!"
    read
    If name = "DONE" .OR. name = "done"
      @ row+2,10 say "Total: "
      @ row+2,41 say ltrim(str(tot,14,2))
      wait space(10)+ "Press any key to continue...."
      clear
      return
    Endif
    @ row,41 get valu1 pict "999,999,999,999.99"
    read
    ans = " "
    @ row,62 say "Correct [Y/N]" get ans
    read
    if upper(ans) = "N"
      loop
    endif
    tot = ( tot + valu1 )
    sn = sn + 1
    row = row + 2
    if row > 17
      @ 7,7 CLEAR TO 23,60
      row = 7
    endif
    APPEND BLANK
    replace NOMI_NUM with MNOMI
    replace T_NOMI with valu1
    replace MONTH with MMONTH
    replace NOMIN with NAME
    replace e_nomi with tot
    name = space(15)
    store 0 to valu1
    loop
    endif
CLOSE ALL
CLEAR
RETURN
enddo


* Program Name.....: PRE_R.prg
* Purpose..........: Generate report on prelimnaries
* Note.............: Developed for

* Memo 1
* Setting working environment.
CLOSE ALL
CLEAR
SET TALK OFF
SET ESCAPE ON
SET DATE BRIT
SET BELL OFF
SET SCORE OFF
SET STATU OFF
store 0 to tot1,tot2,tot3,tot4
do while .t.
 set color to gr+/b
 clear
 mpre = 0
 @ 10,10 TO 12,50 DOUBLE
 @ 11,13 SAY "Enter Valuation Number  :-" get mpre pict "999"
 read
 USE slip
 locate for vv_no = mpre
 if .not. found()
```

```
 @ 17,10 say "Match Record not found"
 @ 18,10 say "Press any key to continue....."
 wait space(10) + " "
 close databases
 return
endif
clear
@ 1,20 SAY "Preliminary  Bill  Report"
@ 2,20 say "This report is produce on "+dtoc(date())
@ 2,0 say "Valution no :"+ltrim(str(mpre))
@ 3,35 say "Initial"
@ 3,47 say "During"
@ 3,59 say "Completion"
@ 4,2 say " A. Foreman In Charge        "
@ 5,2 say " B. Insurance                "
@ 6,2 say " C. Drawing                  "
@ 7,2 say " D. Levelling Instrument     "
@ 8,2 say " E. Site Accomodation        "
@ 9,2 say " F. Site Office              "
@ 10,2 say " G. Water                   "
@ 11,2 say " H. Electricity             "
@ 12,2 say " I. Watching & Lighting     "
@ 13,2 say " J. Welfare and Safety      "
@ 14,2 say " K. First Aid Box           "
@ 15,2 say " L. Site Meeting            "
@ 16,2 say " M. Progress Photograph     "
@ 17,2 say " N. Scaffolding & Plant     "
@ 18,2 say " O. Protection of Works     "
@ 19,2 say " P. Text and Sample         "
@ 20,2 say " Q. Hoarding & Fencing      "
@ 21,2 say " R. Site Board              "
@ 22,2 say " S. Clearing Rubbing        "
*@ 23,2 say " Total    "
use pre
l = 4
go top
do while .not. eof()
  if pre <> mpre
    skip
    loop
  endif
  @ l,28 say mint
  @ l,42 say mdur
  @ l,58 say mcomp
  tot1 = (tot1 + mint)
 tot2 = (tot2 + mdur)
  tot3 = (tot3 + mcomp)
  l = l + 1
  skip
  loop
enddo
*tot4 = tot1 + tot2 + tot3
*@ 23,58 say tot4
wait space(5) + "Press any key to Continue..."
return
enddo



clear
text

   1. The REPORT MENU
      ~~~~~~~~~~~~~~~
      This helps you produce both SOFT and HARD copy of info-
      rmation. Various data supplied into the system
      are processed, and the result of the processed that are
      either displayed on the screen or printer.  To generate
```

```
    any report of your choice, select appropriate option
    from the control menu program. The system prompt
    you to entered the project valuation number, and the out
    put of the processesd data display on screen or printer.
    The NET VALUATION REPORT is the sub program instruction
    that produce detailed result of the whole project work.

  2. Printing Information on Screen.
  ----------------------------------
    If you are not prompted "Sent Output to Printer[Y/N]"
    then you can print the information on screen by
    using the print screen SysRq key on the keyboard.
endtext
@22,5 say ""
wait space(5) + "Press any key to return to help menu"
clear
do helpo
```

```
* Program Name:... roof.prg
* Purpose:..... Allow the user's to enter data
*              relating to roofing work on site.
* Note:..... Called from main_c.prg

* Set up the working environment.
SET SCOREBOARD OFF
SET STATUS OFF
SET TALK OFF
SET BELL OFF
SET DATE BRITISH
SET ESCAPE ON

* Initialize memory variable

  store 0 to sn
  store 7 to row
  store space(15) to name
  store 0 to amm,rate,valu
  store 0 to tot
  store 0 to mcon,mmonth
do while .t.
 clear
 set color to gr+/b+
 @ 12,10 TO 14,50 DOUBLE
 @ 13,11 SAY "Enter Valuation Number:-" get mcon pict "999"
 read
 USE SLIP.DBF
 locate for vv_no = mcon
 if .not. found()
  @ 17,10 say "Match record not found"
  @ 18,10 say "Press any key to continue...."
  wait space(10) + " "
  close databases
  return
 endif
* Test If data already exist
  use roof
  locate for roof = mcon
  if found()
   clear
   @ 12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue..."
   clear
   loop
  endif
* IF RECORD HAS BEEN ENTERED
 clear
 @ 2,5 to 24,70
 @ 1,20 SAY "R o o f i n g  W o r k"
 @5,7 say "No"
 @ 5,14 say "[Enter DONE to exit]"
 @ 6,14 say "Description"
 @ 6,33 say "Amount"
 @ 6,49 say "% of work done"
 @ 6, 65 say "Result"
 sn = sn + 1
 row = row + 1
Do while .t.
 @ row,7 say ltrim(str(sn))
 @ row,14 get name
 read
 If upper(name) = "DONE"
  @row+6,10 say "Total:"
  @row+6,58 say ltrim(str(tot,14,2))
  wait space(10) + "Press any key to continue...."
  clear
```

```
 return
Endif
@ row,33 get amm pict "99,999,999.99"
@ row,49 get rate pict "999"
read
ans = " "
@ row+1,33 say "Correct [Y/N]" get ans
read
if upper(ans) = "N"
 loop
endif
@ row+1,33 say space(20)
valu = ( (rate/100) * amm )
tot = ( tot + valu )
@ row,65 say ltrim(str(valu,14,2))
sn = sn + 1
row = row + 2
if row > 15
 @ 7,7 CLEAR TO 20,78
 row = 7
 loop
endif
append blank
replace roof with mcon
replace month with mmonth
replace t_roof with tot
store 0 to rate,amm
name = space(15)
loop
enddo
```

```
* Program Name:... block.prg
* Purpose:..... Allow the user's to enter data
*              relating to block work on site.
* Note:..... Called from main_c.prg

* Set up the working environment.
SET SCOREBOARD OFF
SET STATUS OFF
SET TALK OFF
SET BELL OFF
SET DATE BRITISH
SET ESCAPE ON

* Initialize memory variable

  store 0 to sn
  store 7 to row
  store space(15) to name
  store 0 to amm,rate,valu
  store 0 to tot
  store 0 to msub,mmonth
do while .t.
 clear
 set color to gr+/b+
 @ 12,10 TO 14,50 DOUBLE
 @ 13,11 SAY "Enter Valuation Number:-" get msub pict "999"
 read
 USE SLIP.DBF
 locate for vv_no = msub
 if .not. found()
  @ 17,10 say "Match record not found"
  @ 18,10 say "Press any key to continue...."
  wait space(10) + " "
  close databases
  return
 endif
* Test If data already exist
  use block
  locate for m_block = msub
  if found()
   clear
   @ 12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue..."
   clear
   loop
  endif
* IF RECORD HAS BEEN ENTERED
 clear
* @ 2,5 to 24,79
 @ 1,20 SAY "B l o c k  W o r k"
 @5,7 say "No"
 @ 5,14 say "[Enter DONE to exit]"
 @ 6,14 say "Description"
 @ 6,33 say "Amount"
 @ 6,49 say "% of work done"
 @ 6, 65 say "Result"
 sn = sn + 1
 row = row + 1
Do while .t.
 @ row,7 say ltrim(str(sn))
 @ row,14 get name
 read
 If upper(name) = "DONE"
  @row+6,10 say "Total:=>"
  @row+6,58 say ltrim(str(tot,14,2))
  wait space(10) + "Press any key to continue...."
  clear
```

```
 return
 Endif
 @ row,33 get amm pict "99,999,999.99"
 @ row,49 get rate pict "999"
 read
 ans = " "
 @ row+1,33 say "Correct [Y/N]" get ans
 read
 if upper(ans) = "N"
  loop
 endif
 @ row+1,33 say space(20)
valu = ( (rate/100) * amm )
tot = ( tot + valu )
@ row,65 say ltrim(str(valu,14,2))
sn = sn + 1
row = row + 2
if row > 15
 @ 7,7 CLEAR TO 20,75
 row = 7
 loop
endif
append blank
replace m_block with msub
replace month with mmonth
replace t_block with tot
store 0 to rate,amm
name = space(15)
loop
enddo
```

```
wait space(5) + "This screen is full, press any key to continue"
clear
@5,5 SAY "Enter Repayment"
store 0 to ad2
@5,60 get ad2 pict "999,999,999,999.99"
read
store 0 to ada
ada = int22 - ad2
use flc
do while .not. eof()
  if flc <> mval
    skip
    loop
  endif
  int23 = int23 + t_flc
  skip
  loop
enddo
close databases
@ 10,5 say "Fluctation"
@ 10,60 say ltrim(str(int23,14,2))
@ 11,5 say "Net Valuations"
int24 = int23 + ada
@ 11,60 say ltrim(str(int24,14,2))
@ 12,5 say "Less amount previously certify"
@ 12,60 get int25 PICT "999,999,999,999.99"
read
int26 = int24 - int25
@ 13,5 say "Total Value:"
@ 13,60 say ltrim(str(int26,14,2))
wait space(5) + "End of Valuation"
do v_main
```

```
* Program Name:... con1.prg
* Purpose:..... Allow the user's to enter data
*              relating to concreate work on site.
* Note:..... Called from main_c.prg

* Set up the working environment.
SET SCOREBOARD OFF
SET STATUS OFF
SET TALK OFF
SET BELL OFF
SET DATE BRITISH
SET ESCAPE ON

* Initialize memory variable

  store 0 to sn
  store 7 to row
  store space(15) to name
  store 0 to amm,rate,valu
  store 0 to tot
  store 0 to mcon,mmonth
do while .t.
 clear
 set color to gr+/b+
 @ 12,10 TO 14,50 DOUBLE
 @ 13,11 SAY "Enter Valuation Number:-" get mcon pict "999"
 read
 USE SLIP.DBF
 locate for vv_no = mcon
 if .not. found()
  @ 17,10 say "Match record not found"
  @ 18,10 say "Press any key to continue...."
  wait space(10) + " "
  close databases
  return
 endif
* Test If data already exist
  use con1
  locate for m_con = mcon
  if found()
   clear
   @ 12,10 say "Valuation number already exist"
   wait space(10) + "Press any key to continue..."
   clear
   loop
  endif
* IF RECORD HAS BEEN ENTERED
 clear
 @ 2,5 to 24,79
 @ 1,20 SAY "C o n c r e a t e  W o r k"
 @5,7 say "No"
 @ 5,14 say "[Enter DONE to exit]"
 @ 6,14 say "Description"
 @ 6,33 say "Amount"
 @ 6,49 say "% of work done"
 @ 6, 69 say "Result"
 sn = sn + 1
 row = row + 1
Do while .t.
 @ row,7 say ltrim(str(sn))
 @ row,14 get name
 read
 If upper(name) = "DONE"
  @row+6,10 say "Total:=>"
  @row+6,58 say ltrim(str(tot,14,2))
  wait space(10) + "Press any key to continue...."
  clear
```

```
 return
 Endif
 @ row,33 get amm pict "99,999,999.99"
 @ row,49 get rate pict "999"
 read
 ans = " "
 @ row+1,33 say "Correct [Y/N]" get ans
 read
 if upper(ans) = "N"
  loop
 endif
 @ row+1,33 say space(20)
valu = ( (rate/100) * amm )
tot = ( tot + valu )
@ row,69 say ltrim(str(valu,14,2))
sn = sn + 1
row = row + 2
if row > 15
 @ 7,7 CLEAR TO 20,75
 row = 7
 loop
endif
append blank
replace m_con with mcon
replace month with mmonth
replace t_con with tot
store 0 to rate,amm
name = space(15)
loop
enddo
**********************************************
* Cut.prg
**********************************************

@ 12,60 say ltrim(str(int19,14,2))
int20 = (int18 + int19)
@ 17,57 say repl ("-",22)
@ 18,5 say "Total value of works & Materials on site"
@ 18,60 say ltrim(str(int20,18,2))
@ 19,5 SAY "Less Retention"
 store 0 to A,A2,INT21,INT22,INT23,INT24,INT25,yoyo,yoyo2
 A = ((10/100) * INT20)
 USE SLIP
 LOCATE FOR VV_NO = MVAL
 A2 = ((10/100) * CC_SUM)
 IF A > A2
    A = ((5/100) * CC_SUM)
    CLOSE DATABASES
 ENDIF
INT21 = (INT20 - A)
@ 19,60 say ltrim(str(A,14,2))
@ 20,60 SAY LTRIM(STR(INT21,14,2))
@ 21,5 SAY "Enter Mobilisation Advance"
STORE 0 TO B
@ 21,60 get b pict "999,999,999,999.99"
read
store space(1) to addo
do while .not. addo$"AaLl"
addo = " "
@ 22,5 say "[A]dd or [L]ess mobilisation," get addo pict "!"
  read
enddo
if UPPER(ADDO) = "A"
  int22 = (int21 + b)
else
  int22 = (int21 - b)
endif
@ 22,60 SAY ltrim(str(int22,14,2))
```