

**COMPUTERISATION OF BANK RECONCILIATION
STATEMENT (A CASE STUDY OF MINISTRY
OF FINANCE - NIGER STATE)**

BY

HABIBA MAHMOUD

PGD/MCS/063/96

**A PROJECT SUBMITTED TO THE DEPARTMENT OF
MATHS/COMPUTER SCIENCE, FEDERAL UNIVERSITY
OF TECHNOLOGY, MINNA IN PARTIAL FULFILMENT
OF THE REQUIREMENT FOR THE AWARD OF THE
POST-GRADUATE DIPLOMA IN
COMPUTER SCIENCE**

MARCH, 1998

CERTIFICATION

Having read through this project carried out b Mrs Habiba Mahmoud,
it is out opinion that it is up to the standard for Postgraduate
Diploma in Computer Science.

PRINCE R BADAMOSI
SUPERVISOR

DATE

DR K. R. ADEBOYE
HEAD OF DEPARTMENT

DATE

EXTERNAL EXAMINER

DATE

DEDICATION

This Project Work is dedicated to my beloved husband, Mahmoud K Bello and generally to the course of Allah.

ACKNOWLEDGEMENT

Praise be to Allah to whom all service is due and may his blessings be on Prophet Mohammed (SAW).

I must also express my deep appreciation to my Supervisor, Prince R Badamosi for the guidance and encouragement I received from him, he has been so helpful throughout the duration of this Project Work. Thank you sir.

My unbounded gratitude goes to my husband - Mahmoud Bello for his untiring supports both morally and financially in making this course a success.

I also wish to acknowledge with pleasure the able advice and concern shown by my Head of Department in person of Dr. K. R. Adeboye throughout the duration of the course.

My special gratitude to all my lecturers in the Department for their contributions to my achievement during my period of study in the University.

Equally worth mentioning are my children - Usman, Hamza and Hauwa for their patience, support and understanding. May God bless them all.

Finally, thanks to all who have contributed to the success of this work. May Allah reward and bless them all.

ABSTRACT

Generally, organisations be private or public needs a more efficient, reliable and quicker way in preparing of their bank reconciliation statement of accounts.

With the modern technology, it has become imperative that bank reconciliation statement be computerised in order to provide adequate information on the financial status of the organisation at anytime when required for a quick decision-making on financial matters. It is based on this assumption that the project focuses more on the need for computerization of bank reconciliation statement.

TABLE OF CONTENTS

	Page
Certification	i
Dedication	ii
Acknowledgement	iii
Abstract	iv
Table of Content	v-vi
 <u>CHAPTER ONE</u>	
Introduction	1
Objective of the Study	3
Definition of Terms	4
Bank Reconciliation In Perspective.....	6
 <u>CHAPTER TWO</u>	
Origin of Banking	7
Functions of Banks	9
Bank Reconciliation (Formal Approach)	10
Need For Reconciliation In Non-Banking Sectors .	11
Why Computer In Reconciling Bank Statement	12
 <u>CHAPTER THREE</u>	
Systems Analysis And Design	14
Feasibility Study	14
Testing Project Feasibility	14
The Current System	15
System Deisgn	15
Cost And Benefit Analysis	16
Input Specification	17

Output Specification	17
<u>CHAPTER FOUR</u>	
SOFTWARE/PROGRAM DEVELOPMENT	
Introduction	18
Choice of Language	18
Features of Language Chosen	18
Hardware Requirement	18
Staff Training	19
Change Over Procedure	19
Starting The System	20
Main Menu	21
<u>CHAPTER FIVE</u>	
Summary, Conclusion and Recommendation	22
References	23

CHAPTER ONE

INTRODUCTION

It is common feature in any organisation that cash must be received inwards and disbursed outwards, documented in an account book and the accuracy of recording the cash inflows and cash outflows is determined by the periodical preparation of statement recording cash receipts and cash payments as they arise from time to time in the cash book.

Cash receipts include the various items of incomes accruing to an organisation while cash payments include the various items of expenses incurred by the organisation.

Bank reconciliation is the process of agreeing the balance of cash on one account with the balance on another account.

In the Ministry, this involves reconciling the Ministry's Cashbooks balance with the Ministry's bank balance at the end of every month. However, this procedure might be a bit cumbersome in a real banking sector.

Indeed, bank reconciliation is vital so as to ascertain the current and true picture of the customer's accounts with the bank. The reconciliation in the banks are done on manual basis. That is opening a card for each customer by crediting all the deposits made into that account and debiting all the withdrawals from that accounts and obtain a new balance at the end of each transaction. Reconciliation of banks statement manually is not without its shortcomings. This shortcomings include:

- a) General repetitive nature of tasks of book-keeping.

- b) Copying and rearranging of information already produced.
- c) Computations of balance anytime transaction takes place.
- d) Involvement of much paper work in the operation.

However, with the introduction of computer, a lot of work done manually can now be effectively done using computer. Indeed, using computer in preparing reconciliation by banks do have the following advantages:

- a) Eliminating of many repetition works of book-keeping.
- b) Automatic mathematics computation
- c) Minimal reduction of use of stationery
- d) Automatic updating of records and maintenance continuously.

OBJECTIVES OF THE STUDY

The objectives of this project is as follows:

- 1) To ensure prompt postings of financial transactions of the Ministry into the cashbook.
- 2) To examine in details the present system of reconciliation in the ministry.
- 3) Analysing the short comings of the present system.
- 4) To achieve accuracy of posting of all financial transactions through bank reconciliation statement preparation.
- 5) Assessing the performance and evolution of the new system and offer recommendations for better performance.

DEFINITION OF TERMS

1. Collating - This is the gathering of all the bank statements and cash vouchers to facilitate the preparation of bank reconciliation statement at the end of every month.
2. Bank Statement - This is a summary statement of a customer's account in the bank's ledger. This statement is usually sent to the customers from time to time or at the request of the customer.
3. Cashbook - This is a book of original entry where cash receipts and cash payment are recorded during any financial transactions.
4. Ledger - It is a summary statement of a transaction in both words and figures.
5. Computerization - This is the use of programmed computers to process cash receipts and cash payments as they arise in order to generate the desired results that shall be used for management decision making.
6. Unpresented Cheques - These are cheques drawn by the organisation on itself and given to an individual or corporate body who fails to present those cheques before the bank statement is prepared or sent.
7. Reconciliation of Account - This is the process of agreeing the balance of one account with the balance of another account.

8. Cash Receipts - These are the various items of cash receipts received by bank as recorded on the debit side of the main cashbook.
9. Uncleared Effect/Uncredited Lodgement - These are cheques paid into a business or individual account of which the bank has not given a direct credit in the bank statement as a result of the cheque undergoing clearance.
10. Posting - This is the recording of financial transactions on the proper side of a cashbook, e.g, the receipts are always recorded on the debit side.
11. Dishonoured Cheques - This is a situation whereby a cheque initially credited in a bank statement to a customer is later debited back in the bank statement as a result of some irregularity regarding the cheque.

BANK RECONCILIATION IN PERSPECTIVE

For the fact that banks do prepare reconciliation for deposits and withdrawals of individuals and organisations accounts with them, such individual and organisations do also maintain cash book records of all the deposits and withdrawals they have in the banks. The method of approach used by the individuals and organisations is somehow similar to that of the banks. The main difference is that money own by individual or organisation is shown on the debit side and liabilities on the credit side. In all the cases, record keeping process is carried out normally.

However, with the introduction of the computer, the reconciliation can be done quickly and faster. The importance of timely and accurate bank reconciliation statement cannot be over-emphasised in an organisation like the Ministry of Finance. They needed the information for the following reasons in addition to the benefits to be derive using computers:

- a) To know its financial position at any particular period of time.
- b) To avoid over issuance of cheques so as not to lead to been bounced.
- c) To avoid the embarrassment of dishonouring cheques due to lack of sufficient fund in the account.
- d) To avoid the overdrawing of their bank accounts and hence incur overdraft charges by the banks.

CHAPTER TWO

ORIGIN OF BANKING

Banking as we know it today is generally known to have started by the Halian Gold Smiths who settled down into business in London around the 17th Century. They began by accepting deposits of gold, coins and other valuables from their customers for safe keeping. As the volume of this business grew, they had to build large strongrooms where these customers valuable items were kept until demands were at any particular time.

And so, they began to give out part of the money deposited to interested borrowers by law of loans. They charged some amount of interest. The acceptance of deposit and granting of loans are still some of the basic banking functions all over the world today. It must be borne in mind that the forerunner of the modern banking started and performed virtually all the present functions of modern banking. The acceptance of their customer's letter of instruction to transfer funds from his/her holding to another represent the present day cheque system.

After all, the cheque is merely an instruction on legalised paper from one customer to the banker requesting him (Bank) to pay money written on cheque to a named beneficiary.

The goldsmiths receipts to their clients become the first known issue to notes, though they were not legal tender. These receipts later became transferable instruments.

As the individual goldsmith's business expanded, it became necessary for them to organise themselves into groups to form

merchants and private banks. As a result of the fast expanding activities of these gold-smiths and huge financial involvement to protect both the depositors and gold-smiths.

In consequences, therefore the British Government in 1694 established the Bank of England to regulate and control these Merchants and Private banks amongst other functions.

In Nigeria, banking came with the advent of colonial masters. The introduction of the first modern banking dated to 1892 when the African Banking Corporation was established in Lagos at the invitation of Elder Dempster and Company. African Banking Corporation was based in South Africa but merely open a branch office in Lagos to finance shipping business of Elder Dempster Company who was operating steamship service between Liverpool and the West Coast of Africa. Problem as a result of the good performance of the African Banking Corporation another bank opened its branch office in Lagos in 1894. The bank was the Bank of British West Africa which was registered in London in 1894 with an authorized capital of N120,000 and this bank enjoyed the monopoly over banking business in Nigerian until 1916.

Until this date however, the bank of British West Africa was the sales agent for Custody and Distribution of British Silver Currency in West Africa as issued by the West African Currency Board which was established in 1912.

The Bank of British West African remained dominated in the field until 1914 when the colonial bank which was more in the business of

banking was established. As a result of its dynamism, the bank opened 15 branches within the year it was established in West Africa.

In 1915 the asset and liabilities of these banks were then taken over by a Consortium of banks comprising of Barclays Bank, Anglo Egyptian Bank and National Bank of South Africa to form a new bank called Barclays Bank D.C.O. This new bank had to change its name from its earlier colonial name as a result of new banks that sprang up. Such banks include United Bank for Africa (UBA), Nigerian-Arab Bank, Savannah Bank, etc.

FUNCTIONS OF BANKS

Commercial banks can be defined as business enterprises set up to do banking business. They are the most common type of banks in Nigeria and they started much earlier than all other types of banks. The commercial bank accept deposits from the public and make profit by lending money to the individual or corporate body. The functions of the bank include thus:

1. Accepting and keeping of deposits on behalf of their various customers.
2. Provision of credit facilities to their respective customers.
3. Provision of banking facilities to their customers both within the country and outside the country.
4. Banks also provide facilities for the safe keeping of valuables like jewelleries and documents for their customers.
5. Banks also give financial advice to their customers on the use

and management of funds and on how to manage their business.

6. To safe guard the deposits of customers and to prevent bank failures.

BANK RECONCILIATION (FORMAL APPROACH)

Bank reconciliation can be defined as a way of keeping both bank record and general ledger cash account record in balance. It is a good control procedure for identifying inaccuracies introduced by the banks accounting of cheques and deposits, disbursements, possibly unauthorized that have not been accounted for through cash disbursement records, old outstanding cheques that may never be cashed and adjustments generated by the banks.

Indeed, the relationship between a bank and its customers is one of debtor and creditor. For example, when company ZAB pays in N20,000.00 cash into the bank. The bank will debit cash account and credit the company account ZAB, meaning that it owes the company N20,000.00 which the company has paid in. If the company ZAB gives out a cheque for N12,000.00 to another dealer, the company's record in the bank account will read thus:-

BANK ACCOUNT OF ZAB COMPANY

Capital - N20,000.00	Cheque drawn N12,000.00
	<u>Balance c/d 8,000.00</u>
N20,000.00	N20,000.00

Balance b/d - N8,000.00

From the above record, it shows that the company's balance now is N8,000.00, but until the cheque issued out to their dealer is

presented for payment at the bank, the balance on the company's account will still stand as N20,000.00.

It must at this point be shown that differences in a bank statement and a customer's own cash book records can be as a result of the following:

- a) Unpresented cheques
- b) Uncredited cheques
- c) Bank charges
- d) Bank commission on Turn-over (COT)
- e) Interest Paid
- f) Dishonoured cheques
- g) Direct payments into or from the bank account.

NEED FOR RECONCILIATION IN NON-BANKING SECTORS

The importance of preparing bank reconciliation in a non-banking sectors can never be overemphasised. For the fact that individuals, organisations, government and institutions have an account in the bank(s), they have to, from time to time prepare a reconciliation statement to know the balance of what they have at anytime in the bank.

The purpose of a bank reconciliation statement in a non-banking sectors is to:

- a) Ensure that management is kept abreast of the actual cash backing for meeting organizational needs.
- b) Reflects bank charges or other deduction from the balance at the bank that may have escaped management attention or was

never initiated by management, so that the institutions book of accounts reflect such discrepancies.

- c) Shows all payments to the institution and other credits made directly to the bank by the individual paying it so that full account can be taken of all such receipts.
- d) Ensure that all cheques drawn on the account or other payments or transfer instructions made or given by the institution are fully debited to the account.
- e) Ensures that any discrepancies between the bank entries and entries into the institutions cash books are fully looked into and corrected in the books of account.
- f) Ensures that all bank charges or bank initiated debits are in line with prevailing regulations or agreement with the bank.

WHY COMPUTER IN RECONCILING BANK STATEMENT?

In business, banks and some organisations, computers are been used for many purposes, such include:- preparing payroll, book-keeping, inventory and stock control, billing, etc.

With the invention of computers, many softwares were also introduced. Most especially the accounting packages available include ACCPAC PLUS, REALWORD Payroll and DacEasy Accounting. Indeed, DacEasy accounting is a low-cost package with a large number of accounting functions. It is an integrated package that provides several functions, such as General Ledger, Account Recovable, Account Payable, Purchase Orders, Billing, etc.

One of the recent software that deals more on the Bank

reconciliation is the Data Perfect. This software is rather very wide and comprehensive. Using this software, the reconciliation report will help in balancing cheque books so that books will be the same as that of the bank.

From the sample software enumerated, it could be seen that application of computer in preparing bank reconciliation is a new development. Even in some organisations like banks, companies that use the computers in their operations, the reconciliation aspect is still done manually.

The main advantages of using computers in reconciling bank statements include thus:

- a) It helps to eliminate many of the repetitive task of book keeping.
- b) It eliminates copying and re-arranging of information which has already been entered into the system.
- c) It performs all the mechanical computations faster and accurately.
- d) It eliminates most of the paperwork.
- e) It helps in updating of records automatically
- f) It produces reports that are timely and accurate.

CHAPTER THREE

SYSTEMS ANALYSIS AND DESIGN

Systems analysis and design is a process similar to problem solving. The process of system analysis involves a number of steps that can be applied to any study.

FEASIBILITY STUDY

Possible feasibility study was carried out on the existing system and a look at all the possible alternative solutions were done.

In preparing bank reconciliation of the Ministry manually, the following problems were encountered:

- a) Repetitive task of book-keeping
- b) Computations of balance anytime transaction takes place.
- c) Delay in generating of reports when required.
- d) Frequent mistakes made while preparing a reconciliation.
- e) Time consuming and wasting
- f) Once reconciliation is prepared manually, it becomes very difficult to insert or delete any information, as alterations could render it useless.

TESTING THE PROJECT FEASIBILITY

For project feasibility, the following have to be undertaken:

- 1) Operational Feasibility - this relates or is concerned with the workability of the proposed information system when developed and installed.
- 2) Technical Feasibility - This test seeks to clarify if the proposed project can be done with current equipment, existing software and available personnel.

- 3) Economical Feasibility - This relates to the financial feasibility. This has to be undertaken to assess cost of implementing the proposed project.

All in all, the proposed system can be carried out based on the main testing project feasibility.

THE CURRENT SYSTEM

Bank reconciliation in the Ministry under review is prepared on monthly basis. To prepare a bank reconciliation, one needs to get the record of all the payments and receipts by the Ministry and a bank statement for a particular month to be reconciled.

This is followed by producing the cash book balance for the period under review. This will show the previous month's closing balance, which will serve as the new opening balance for the month to be reconcile. This is then added to the total receipts and deduct the total payments in order to get a new cash book balance.

After obtaining the cash book balance, it is then transfer to the appropriate bank for reconciliation with the bank statement.

SYSTEM DESIGN

Design is the process whereby the systems analyst applies his judgement, skills and knowledge to interpret the requirements specification that provides detailed documentation of the new system.

For effective design to be accomplished, some basic factors must be considered:

- 1) Production of desired information at the right time, and

amount with an acceptable level of accuracy.

- 2) The need to minimise cost and time spent on data preparation.
- 3) Effective safeguards for prevention of frauds.
- 4) Effective security measures to avoid loss of data stored in files.
- 5) Efficient design of documents and reports.

COST AND BENEFIT ANALYSIS OF THE NEW SYSTEM

OPERATING COST

	N
Supplies (Diskettes, Stationery)	- 50,000.00
Equipment Maintenance	- 15,000.00
Program Maintenance	- 20,000.00
Labour Cost (5 Operators)	- 10,000.00
Utilities	- 80,000.00
Miscellaneous Expenses	- <u>20,000.00</u>
	<u>195,000.00</u>
	=====

DEVELOPMENT COST

Systems Analysis and Design for Three (3) weeks	- 25,000.00
Software Development	- 15,000.00
Five (5) Personal Computers	- 250,000.00
Installations	- 10,000.00
Miscellaneous Expenses	- <u>15,000.00</u>
	<u>90,000.00</u>
	=====

Grand Total = N195,000.00 + N90,000.00 = N285,000.00

BENEFITS OF THE PROPOSED SYSTEM

- a) Reduction in the use of stationery (paper work)
- b) Sorting and arranging of information in various ways can be done easily and quickly.
- c) Automatic updating of records and maintenance.
- d) Elimination of many repetitive work of book keeping

INPUT SPECIFICATION

This describes the manner in which data enters the system for processing. Indeed, the input specification is influenced by the need of the output. The main input specification for this work involves about six database files been used. This includes:

- a) Receipt . Dbf
- b) Payment . dbf
- c) Unpresented . dbf
- d) Uncredit . dbf
- e) Dishonour . dbf
- f) Reconcile . dbf

OUTPUT SPECIFICATION

One of the most relevant features of an information system for users is the output it produces. The programm is design based on the requirement of the system. This include producing monthly reports that show -

- 1. Cash book balance
- 2) Bank reconciliation statement
- 3) List of unpresented, uncredited and dishonoured cheques.

CHAPTER FOUR

SOFTWARE/PROGRAM DEVELOPMENT

Introduction

Programming is the preparation of a detailed sequence of operating instruction for particular problem to be run on a computer. It involves identification of the problem into a program flow chart, testing and running the program. In programming, input and output must be specified because the output can always be determined by the inputs.

CHOICE OF LANGUAGE

In developing this system dBASE IV programming language is used. This is due to numerous facilities available.

Features of Language Chosen

1. It provides a full relational database environment to users.
2. Using dbase IV, one can design databases, manipulate and edit records and files, generate reports, perform database query, design labels, and database without the use of command language.
3. Data can be verified automatically as they are entered into fields. Up to 255 fields can be specified per record.
4. Pop-up menus and windows can be designed.
5. It has a larger number of memory variables, user define functions up to 99 files can be opened at a time.

Hardware Requirement

The proposed system requires the following:

- Personal Computer 836 main processor

- RAM - 8MB
- Floppy Disk Drive - 3.5/5.25
- Colour Monitor
- Laser Jet Printer (6L)
- Stabilizer 250VA
- UPS 1000VA

Software Requirement

- MS-DOS 6.0 Version
- DBASE IV

Staff Training

The amount and period of training for this system will depend upon its complexity and the available skill on the ground presently.

The proposed system will be users-friendly. However, it is necessary to have an in-house training for the various personnel in the Finance Section.

The training will cover areas like Basic Computing and Operations guidelines for the account staff in general. This training should not exceed three weeks of rigorous practicals in the usage of the packages designed.

Change-over Procedures/System Conversion

The following approaches/Procedures could be used during conversion:

1. The Parallel Approach - This is a method whereby the old method is operated simultaneously for some time with the new system to make sure that the new system meets the requirements

that the old system has been meeting all along and to determine whether the new system will be able to stand the test of time.

2. Direct Method - This is a method where the old system is discontinued and the new system becomes operational immediately.
3. Piecemeal Method - This is a method whereby changing to a new system is done gradually until the desired result is installed in other parts of the Ministry gradually.

In the case of the Ministry of Finance, the new system will be implemented using the parallel method because of the advantage it has over other methods, since the new system will be operated simultaneously with the old system.

Starting The System

The system has been designed in such a way that it can be used by everyone. To operate the system after successful installation on the hard disk, the software can be invoked by the following procedures:

1. At the c prompts, type CD DBASE(4) and press Enter = c:\>CD DBASE(4)
2. When the directory is changed to DBASE(4) you then type Dbase (4) and press enter = C:\DBASE(4)>DBASE (4).
3. After pressing enter, the control center of the drive C will appear. You then press "ESC" (ESCAPE) to quit the control panel to Dot Prompt.

4. At the Dot Prompt, you type "Set Default to A:", that is, to diskette.
5. You then type "set directory to A:\ RECONCILE and press enter.
6. Lastly, you then type "Do Reconcile" and press enter. After pressing enter, the main menu screen will be display.

Main Menu

The main menu can be classified into three main sections namely:

- a) the record/data input section
- b) The report section
- c) Exit section

Under the record section, we have payments, receipts, unrepresented, uncredited, dishonoured cheques.

In any of this one can either append, modify or delete records.

For the report section, we have payments and receipts, bank reconciliation, unrepresented, uncredited and dishonoured cheques.

The Exit menu is basically for leaving the system and return to C prompts by pressing Y/N.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

Computerization in any organisation is carried out with the hope of eliminating or reducing to certain minimum level, the use of manual method in carrying out its activities. It is also done with the sole aim of improving the speed, accuracy and efficiency in collection, manipulation, storage, reporting and dissemination of data.

Computerization of Bank Reconciliation Program is aimed at eliminating most of the manual work involved in book-keeping of the Ministry. Indeed with automation, reports can be generated in good time, thus enabling the Ministry officials to take quick decision over its financial obligations.

RECOMMENDATIONS

Systems are bound to face changes as technology, economy and society change, therefore, the new system should be modified from time to time to meet the challenges given brain tasking bank reconciliation.

Also a well planned approach to system maintenance and follow-up is essential to the continued effectiveness of an information system.

REFERENCES

1. Anderson R.G. (1990) - Principles and Practice of Data Processing (Vol. I)
2. Badmus R. (1998) - Lecture Notes On Systems Analysis & Design (Unpublished)
3. Elliason A.L. (1979) - Business Computer and Application. Science Research Associations, Chicago
4. John R. et al (1979) - Accounting and Information Systems. Reston Publishing Company, Reston V.A.
5. Potter D.A (1991) - Automation Accounting System and Procedures Handbook. John Wiley & Sons Inc. New York.

MENU.PRG

```
set echo off
set scoreboard off
set escape on
set status off
set talk off
set bell off
SET COLOR TO B+/BG
clear
@ 1,1 to 23,78 double
@ 2,20 say "COMPUTERISATION OF BANK RECONCILIATION STATEMENT"
@ 3,2 TO 20,77
  define popup main_mnu from 4,17 to 15,58;
message "Highlight menu choice or press enter"
  define bar 1 of main_mnu prompt "***** RECONCILIATION MENU
*****" skip
  define bar 2 of main_mnu prompt ""skip
  define bar 3 of main_mnu prompt "RECIEPT UPDATE"
  define bar 4 of main_mnu prompt "PAYMENT UPDATE"
  define bar 5 of main_mnu prompt "UNPRESENTED CHEQUE"
  define bar 6 of main_mnu prompt "UNCREDITED CHEQUE"
  define bar 7 of main_mnu prompt "CASH BALANCE REPORT"
  define bar 8 of main_mnu prompt "CASH BALANCE STATEMENT"
  define bar 9 of main_mnu prompt "EXIT"
  on selection popup main_mnu do barpop
  ACTIVATE POPUP MAIN_MNU
return

procedure barpop
  DO CASE
CASE bar()=3
  DO RECPT
CASE bar()=4
  DO PAY
CASE bar()=5
  DO UNPCH
CASE bar()=6
  DO UNCCH
CASE bar()=7
  DO RPT
CASE bar()=8
  DO CASHBAL
CASE bar()=9
  clear
  DEACTIVATE POPUP
  RETURN
  ENDCASE
SET BELL ON
SET ESCAPE ON
SET TALK ON
set scoreboard on
set echo on
*set status on
Return
```

CASHBAL.PRG

```
SET ESCAPE On
SET EXACT ON
close all
*do while .t.
store 0 to trec
store repl('-',15) to line
CLEAR
USE bank_stm
do while .not. eof()
  @ 1,15 to 7,65 doub
  @ 2,18 SAY "MINISTRY OF FINANCE, MINNA"
  store 0 to bank_cod
  @ 3,18 say "BANK CODE: " get bank_cod
  read
  locate for ban_code=bank_cod
  if found()
    set filter to ban_code=bank_cod
    store space(30) to acc_typ
    STORE 0 TO ACCT
    @ 4,18 get ACC_TYP
    @ 4,49 say ":" GET ACCT
    READ
    locate for acc_num=acct
    if found()
      set filter to acc_num=acct
      store ctod(' / / ') to ddate
      @ 5,18 say "CASH BALANCE AS AT " get ddate
      read
      locate for dates=ddate
      if found()
        set filter to dates=ddate
        @ 6,18 say upper(bank)
        @ 8,4 SAY "Balance brought forward as at "
        @ 8,35 say ddate
        @ 8,60 say balance pict '999,999,999.99'
        @10,4 say "RECEIPTS "
        bank_bal=bank_st_ba
        lincnt=11
        use
        use reciepts
        locate for ban_code=bank_cod .and. acc_num=acct .and.
        dates=ddate
        if found()
          set filter to ban_code=bank_cod .and. acc_num=acct .and.
          dates=ddate
          do while .not. eof()
            go top
            @lincnt,4 say detail
            @lincnt,45 say amount
            skip
            if lincnt>22
              @23,10 say "Press a key to go on.."
```

```

wait+ ' '
clear
lincnt=3
endif
loop
enddo
    if eof()
    @lincnt+1,45 say line
    sum amount to trec
    lincnt = lincnt +2
    @lincnt,60 say trec
    endif
    endif
    use
    tsum1 = trec + bank_bal
    @lincnt+2,4 SAY "PAYMENTS "
    use payment
    store 0 to tpay
    locate for ban_code=bank_cod .and. acc_num=acct .and.
dates=ddate
    if found()
    set filter to ban_code=bank_cod .and. acc_num=acct .and.
dates=ddate
    do while .not. eof()
    go top
    @lincnt+3,4 say detail
    @lincnt+3,45 say amount
    skip
    if lincnt>22
    @23,10 say "Press a key to go on.."
    wait+ ' '
    clear
    lincnt=3
    endif
    loop
enddo
    if eof()
    @lincnt+4,45 say line
    sum amount to tpay
    @lincnt+5,60 say tpay
    endif
    endif
    use
    @lincnt+6,60 say line
    tsum2=tsum1 - tpay
    @lincnt+8,4 say "Total cash bal. as per above "
    @lincnt+8,60 say tsum2
    @lincnt+9,4 say "Add unrepresented cheques "
use unprech
    store 0 to tunprec
    store 0 to bal
    locate for ban_code=bank_cod .and. acc_num=acct .and.
dates=ddate
    if found()

```

```

        set filter to ban_code=bank_cod .and. acc_num=acct .and.
dates=ddate
        sum amount to tunprec
        @lincnt+9,60 say tunprec
    endif
    bal=tsum2+tunprec
    if lincnt>22
    @23,10 say "Press a key to go on.."
    wait+' '
    clear
    lincnt=3
endif
    use
    tsum3=tsum2 + tunprec
*@lincnt+10,60 say line
@lincnt+10,4 say "Less uncredited cheques "
use uncrech
    store 0 to tuncrec
    store 0 to fbal
    locate for ban_code=bank_cod .and. acc_num=acct .and.
dates=ddate
    if found()
        set filter to ban_code=bank_cod .and. acc_num=acct .and.
dates=ddate
        sum amount to tuncrec
        @lincnt+9,60 say tuncrec
    endif
    fbal=bal-tuncrec
    if lincnt>22
    @23,10 say "Press a key to go on.."
    wait+' '
    clear
    lincnt=3
endif
    use
    tsum4=tsum3 - tuncrec
    @lincnt+11,60 say line
    @lincnt+12,4 say "Balance as per Bank Statement"
    @lincnt+12,60 say fbal
use bank_stm
replace bank_st_ba with tsum2
if lincnt>22
    @23,10 say "Press a key to go on.."
    wait+' '
    clear
    lincnt=3
endif
    @lincnt+13,60 say line
    wait+' '
    return
endif
    endif
    endif
enddo

```

PAY.PRG

```
set talk off
set console off
clear
@ 1,1 to 23,79 double
@ 2,34 SAY "PAYMENT UPDATE MENU"
@ 3,2 to 5,78
define menu pay_mnu
define pad add of pay_mnu prompt "ADD" at 4,5
define pad edt of pay_mnu prompt "EDIT" at 4,18
define pad del of pay_mnu prompt "DELETE" at 4,31
define pad vew of pay_mnu prompt "VIEW" at 4,44
define pad prt of pay_mnu prompt "PRINT" at 4,57
define pad ext of pay_mnu prompt "EXIT" at 4,70
on selection pad add of pay_mnu do payadd
on selection pad edt of pay_mnu do payedit
on selection pad del of pay_mnu do paydel
on selection pad vew of pay_mnu do payvew
on selection pad prt of pay_mnu do payprt
on selection pad ext of pay_mnu do payexit

activate menu pay_mnu pad add
return

procedure payexit
clear
@ 1,1 to 23,78 double
@ 2,22 say "COMPUTERISATION OF BANK RECONCILIATION STATEMENT"
@ 3,2 TO 20,77
deactivate menu
return

PROCEDURE PAYADD
do while .t.
CLEAR
@ 3,2 to 3,78
@ 1,1 to 23,79 double
@ 2,28 say "PAYMENT RECORD ADDITION"
STORE CTOD(" / / ") TO DDATE
STORE 0 TO DCHEQ
STORE 0 TO AMNT
store 0 to bank_cod
STORE 0 TO ACCT
store space(30) to bbank
STORE SPACE(100) TO DETAILS
@ 6,10 say "BANK CODE: "
@ 6,25 get bank_cod
@ 8,10 say "BANK: "
@ 8,25 get bbank pict "@!"
@10,10 SAY "ACCOUNT NO.:"
@10,25 GET ACCT
@12,10 SAY "DATE: "
@12,25 GET DDATE PICT '99/99/99'
@ 7,8 TO 7,72
```

```

@14,10 SAY "CHEQUE NO.: "
@14,25 GET DCHEQ PICT '9999999999'
@16,10 SAY "AMOUNT: "
@16,25 GET AMNT PICT '99,999,999.99'
@18,10 SAY "DETAILS: "
@18,25 GET DETAILS PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
READ
USE PAYMENT
APPEND BLANK
replace ban_code with bank_cod
replace bank with bbank
REPLACE DATES WITH DDATE
REPLACE CHEQ_NUM WITH DCHEQ
REPLACE AMOUNT WITH AMNT
REPLACE ACC_NUM WITH ACCT
REPLACE DETAIL WITH DETAILS
USE
USE BANK_STM
replace ban_code with bank_cod
REPLACE ACC_NUM WITH ACCT
REPLACE DATES WITH DDATE
USE
@19,8 TO 19,72 DOUBLE
STORE " " TO QUE
@20,10 SAY "ADD MORE PAYMENT RECORDS (Y/N)?" GET QUE PICT 'A'
READ
DO CASE
  CASE QUE $ 'yY'
  LOOP
    CASE QUE $ 'nN'
    use
    clear
    @ 1,1 to 23,79 double
    @ 2,30 SAY "PAYMENT UPDATE MENU"
    @ 3,2 to 5,78
  RETURN
ENDCASE
ENDDO WHILE .T.

```

PROCEDURE PAYEDIT

```

do while .t.
  CLEAR
  @ 3,2 to 3,78
  @ 1,1 to 23,79 double
  @ 2,25 say "PAYMENT EDIT MODULE"
  STORE 0 TO BANK_COD
  STORE 0 TO ACCT
  STORE 0 TO DCHEQ
  @ 5,2 say "BANK CODE: " GET BANK_COD pict '9999999'
  @ 5,23 SAY "ACCT. NUMBER: " GET ACCT &&pict 'xxxxxxxxxx'
  @ 5,53 SAY "CHEQ. NUMBER: " GET DCHEQ
  READ
  USE PAYMENT

```



```

DO WHILE .NOT. EOF()
  locate for ban_code=bank_cod
  set filter to BAN_CODE=BANK_COD .AND. acc_num=acct .and.
cheq_num=dcheq
  IF FOUND()
    bank_cod=ban_code
    bbank=bank
    ddate=dates
    dcheq=cheq_num
    amnt=amount
    acct=acc_num
    details= detail
    @ 4,2 clear to 5,77
    @ 6,10 say "BANK CODE: "
    @ 6,25 get bank_cod
    @ 8,10 say "BANK: "
    @ 8,25 get bbank pict "@"!
    @10,10 say "DATE: "
    @10,25 get ddate
    @12,10 say "CHEQUE NUMBER: "
    @12,25 get dcheq &&pict 'xxxxxxxxxxx'
    @14,10 say "AMOUNT: "
    @14,25 get amnt pict '99,999,999.99'
    @16,10 say "ACCOUNT NUMBER: "
    @16,25 get acct
    @18,10 SAY "DETAILS: "
    @18,25 GET DETAILS PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
    READ
    replace bank with bbank
    replace ban_code with bank_cod
    REPLACE DATES WITH DDATE
    REPLACE CHEQ_NUM WITH DCHEQ
    REPLACE AMOUNT WITH AMNT
    REPLACE ACC_NUM WITH ACCT
    REPLACE DETAIL WITH DETAILS
    USE
    set filter to BAN_CODE=BANK_COD .AND. acc_num=acct
    USE BANK_STM
    replace ban_code with bank_cod
    REPLACE ACC_NUM WITH ACCT
    REPLACE DATES WITH DDATE
    USE
    @19,8 TO 19,72 DOUBLE
    STORE " " TO QUE
    @20,10 SAY "EDIT MORE PAYMENT RECORDS (Y/N)?" GET QUE PICT
'A'
    READ
    DO CASE
    CASE QUE $'YY'
      LOOP
    CASE QUE $'nN'
      CLOSE ALL
      clear
      @ 1,1 to 23,79 double

```

```

        @ 2,30 SAY "PAYMENT UPDATE MENU"
        @ 3,2 to 5,78
    RETURN
ENDCASE
ENDIF
ENDDO WHILE .NOT. EOF()
ENDDO WHILE .T.

```

PROCEDURE PAYDEL

```

do while .t.
    CLEAR
    @ 3,2 to 3,78
    @ 1,1 to 23,79 double
    @ 2,25 say "PAYMENT DELETE MODULE"
    STORE 0 TO BANK_COD
    STORE 0 TO ACCT
    STORE 0 TO DCHEQ
    @ 5,2 say "BANK CODE: " GET BANK_COD pict '99999999'
    @ 5,23 SAY "ACCT. NUMBER: " GET ACCT &&pict 'xxxxxxxxxx'
    @ 5,53 SAY "CHEQ. NUMBER: " GET DCHEQ
    READ
    USE PAYMENT
    DO WHILE .NOT. EOF()
        locate for ban_code=bank_cod
        set filter to BAN_CODE=BANK_COD .AND. acc_num=acct .and.
cheq_num=dcheq
        IF FOUND()
            bank_cod=ban_code
            bbank=bank
            ddate=dates
            dcheq=cheq_num
            amnt=amount
            acct=acc_num
            details= detail
            @ 4,2 clear to 5,77
            @ 6,10 say "BANK CODE: "
            @ 6,25 say bank_cod
            @ 8,10 say "BANK: "
            @ 8,25 say bbank pict "@!"
            @10,10 say "DATE: "
            @10,25 say ddate
            @12,10 say "CHEQUE NUMBER: "
            @12,25 say dcheq &&pict 'xxxxxxxxxx'
            @14,10 say "AMOUNT: "
            @14,25 say amnt pict '99,999,999.99'
            @16,10 say "ACCOUNT NUMBER: "
            @16,25 say acct
            @18,10 say "DETAILS: "
            @18,25 say DETAILS PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
            READ
            STORE " " TO resp
            @20,10 SAY "ARE YOU SURE YOU WANT TO DELETE THIS
RECORD(Y/N)?" GET RESP PICT 'A'

```

```

READ
DO CASE
CASE RESP $'yY'
  DELETE
  PACK
ENDC
USE
@19,8 TO 19,72 DOUBLE
@20,10 CLEA TO 20,76
STORE " " TO QUE
@20,10 SAY "EDIT MORE PAYMENT RECORDS (Y/N)?" GET QUE PICT
'A'
READ
DO CASE
CASE QUE $'yY'
  LOOP
CASE QUE $'nN'
  CLOSE ALL
  clear
  @ 1,1 to 23,79 double
  @ 2,30 SAY "PAYMENT UPDATE MENU"
  @ 3,2 to 5,78
  RETURN
ENDCASE
ENDIF
ENDDO WHILE .NOT. EOF()
ENDDO WHILE .T.

```

```

PROCEDURE PAYVIEW
SET ESCAPE On
SET EXACT ON
CLEAR
USE PAYMENT
do while .not. eof()
  @ 1,15 to 7,65 doub
  @ 2,18 SAY "MINISTRY OF FINANCE, MINNA"
  store 0 to bank_cod
  @ 3,18 say "BANK CODE: " get bank_cod
  read
  locate for ban_code=bank_cod
  set filter to ban_code=bank_cod
  if found()
    store space(30) to acc_type
    STORE 0 TO ACCT
    @ 4,18 get acc_type
    @ 4,49 say ":" GET ACCT
    READ
    set filter to acc_num=acct
    if found()
      store ctod(' / / ') to ddate
      @ 5,18 say "PAYMENTS VIEW AS AT " get ddate
      read
      set filter to dates=ddate
      if found()

```

```

@ 6,18 SAY BANK
@ 9,4 SAY "DATE"
@ 9,16 SAY "DETAILS"
@ 9,55 SAY "CHEQUE NO."
@ 9,66 SAY "AMOUNT"
@ 10,2 TO 10,77
LINCNT=11
@LINCNT,3 SAY DATES PICT '99/99/99'
@LINCNT,15 SAY DETAIL
@LINCNT,55 SAY CHEQ_NUM
@LINCNT,66 SAY AMOUNT PICT '99,999,999.99'
LINCNT=LINCNT + 1
if lincnt=20
    store " " to res
    @21,6 say "press enter to see the rest..."get res
    read
    wait + ""
    clear
    @1,4 say "DATE"
    @1,16 say "DETAILS"
    @1,55 say "CHEQUE NO."
    @1,66 say "AMOUNT"
    @2,2 to 2,77
    lincnt=3
endif
SKIP
LOOP
endif
endif
endif
enddo
if eof()
lincnt=lincnt + 3
@lincnt, 30 say " *** END OF FILE ***"
store " " to que
@21,15 say "press a key to return..." get que
read
if que = ""
*wait + ""
clear
@ 1,1 to 23,79 double
@ 2,30 SAY "PAYMENT UPDATE MENU"
@ 3,2 to 5,78
return
endif
ENDIF
SET EXACT OFF

PROCEDURE PAYPRT
SET TALK OFF
SET ESCAPE On
SET CONSOLE OFF
CLOSE DATABASES
CLEAR

```

```

USE PAYMENT
do while .not. eof()
  @ 1,15 to 7,65 doub
  @ 2,18 SAY "MINISTRY OF FINANCE, MINNA"
  store 0 to bank_cod
  @ 3,18 say "BANK CODE: " get bank_cod
  read
  locate for ban_code=bank_cod
  if found()
    set filter to ban_code=bank_cod
    STORE 0 TO ACCT
    @ 4,18 SAY "ACCOUNT NUMBER: " GET ACCT
    READ
    if found()
      set filter to acc_num=acct
      store ctod(' / / ') to ddate
      @ 5,23 say "PAYMENTS VIEW AS AT " get ddate
      read
      SET DEVICE TO PRINT
      @PROW(),1 SAY REPL("-",132)
      @PROW()+1,18 SAY "MINISTRY OF FINANCE, MINNA"
      @PROW()+1,18 say "BANK CODE: "+ bank_cod
      @PROW()+1,18 SAY "MAIN ACCOUNT NUMBER: "+ ACCT
      @PROW()+1,18 say "PAYMENTS VIEW AS AT "+ ddate
      @PROW()+1,1 SAY REPL("-",132)
      @PROW()+2,6 SAY "DATE"
      @PROW(),29 SAY "DETAILS"
      @PROW(),70 SAY "CHEQUE NO."
      @PROW(),83 SAY "AMOUNT"
      @PROW()+1,1 SAY REPL("-",132)
      @PROW()+2,6 SAY DATES PICT '99/99/99'
      @PROW(),29 SAY DETAIL
      @PROW(),70 SAY CHEQ_NUM
      @PROW(),83 SAY AMOUNT
      if PROW()>58
        SET DEVICE TO SCREEN
        store " " to res
        @21,6 say "press enter to print the rest..."get res
        read
        wait + ""
        SET DEVICE TO PRINT
      endif
      SKIP
      loop
    else
      SET DEVICE TO SCREEN
      @21,15 say "Account number not found!"
      endif
    else
      @21,15 say "bank code not found!"
    endif
  endif
enddo
if eof()
  @PROW()+3, 30 say " *** END OF FILE ***"

```

```
SET DEVICE TO SCREEN
@21,15 say "press a key to return..."
WAIT+ " "
endif
SET EXACT OFF
clear
@ 3,2 to 5,78
@ 1,1 to 23,79 double
return
```

RPT.PRG

```
SET ESCAPE On
SET EXACT ON
close all
*do while .t.
store 0 to trec
store repl('-',15) to line
store repl('-',50) to lines
CLEAR
USE bank_stm
do while .not. eof()
  @ 1,15 to 7,65 doub
  @ 2,18 SAY "MINISTRY OF FINANCE, MINNA"
  store 0 to bank_cod
  @ 3,18 say "BANK CODE: " get bank_cod
  read
  locate for ban_code=bank_cod
  if found()
    set filter to ban_code=bank_cod
    STORE 0 TO ACCT
    @ 4,18 SAY "ACCOUNT NUMBER: " GET ACCT
    READ
    locate for acc_num=acct
    if found()
      set filter to acc_num=acct
      store ctod(' / / ') to ddate
      @ 5,18 say "CASH BALANCE AS AT " get ddate
      read
      locate for dates=ddate
      if found()
        set filter to dates=ddate
        set device to print
        @ prow(),15 say lines
        @ prow()+1,18 SAY "MINISTRY OF FINANCE, MINNA"
        @ prow()+1,18 say "BANK CODE:"+bank_cod
        @ prow()+1,18 SAY "ACCOUNT NUMBER:"+ACCT
        @ prow()+1,18 say "CASH BALANCE AS AT "+ddate
        @ prow()+1,18 say upper(bank)
        @ prow()+1,15 say lines
        @ prow()+2,4 SAY "Balance brought forward as at "
        @ prow(),35 say ddate
        @ prow(),60 say balance pict '999,999,999.99'
        @ prow()+2,4 say "RECEIPTS "
        bank_bal=bank_st_ba
        use
        use reciepts
        locate for ban_code=bank_cod .and. acc_num=acct .and.
        dates=ddate
        if found()
          set filter to ban_code=bank_cod .and. acc_num=acct .and.
          dates=ddate
          do while .not. eof()
            go top
            @prow()+1,4 say detail
```

```

    @prow(),45 say amount
    skip
    if prow()>55
        set device to screen
        @22,10 say "Feed in another paper."
        @23,10 say "Press a key to go on.."
        wait+' '
        eject
        set device to print
    endif
    loop
enddo
if eof()
@prow()+1,45 say line
    sum amount to trec
    @prow()+2,60 say trec
endif
endif
use
tsum1 = trec + bank_bal
@prow()+2,4 SAY "PAYMENTS "
use payment
store 0 to tpay
locate for ban_code=bank_cod .and. acc_num=acct .and.
dates=ddate
if found()
    set filter to ban_code=bank_cod .and. acc_num=acct .and.
dates=ddate
do while .not. eof()
    go top
    @prow()+1,4 say detail
    @prow(),45 say amount
    skip
    if prow()>55
set device to screen
        @22,10 say "Feed in another paper."
@23,10 say "Press a key to go on.."
        wait+' '
        eject
        set device to print
    endif
    loop
enddo
if eof()
    @prow()+1,45 say line
    sum amount to tpay
    @prow()+1,60 say tpay
endif
endif
use
@prow()+1,60 say line
tsum2=tsum1 - tpay
@prow()+2,4 say "Total cash bal. as per above "
@prow(),60 say tsum2

```



```
@prow()+1,4 say "Add unpresented cheques "  
use unprech  
store 0 to tunprec  
store 0 to bal  
locate for ban_code=bank_cod .and. acc_num=acct .and.  
dates=ddate  
if found()  
set filter to ban_code=bank_cod .and. acc_num=acct .and.  
dates=ddate  
sum amount to tunprec  
@prow(),60 say tunprec  
endif  
bal=tsum2+tunprec  
if prow()>55  
set device to screen  
@22,10 say "Feed in a paper."  
@23,10 say "press a key to go on.."  
wait+ ' '  
eject  
set device to print  
endif  
use  
tsum3=tsum2 + tunprec  
@prow()+1,60 say line  
@prow()+1,4 say "Less uncredited cheques "  
use unprech  
store 0 to tuncrec  
store 0 to fbale  
locate for ban_code=bank_cod .and. acc_num=acct .and.  
dates=ddate  
if found()  
set filter to ban_code=bank_cod .and. acc_num=acct .and.  
dates=ddate  
sum amount to tuncrec  
@prow(),60 say tuncrec  
endif  
fbale=bal-tuncrec  
if prow()>55  
set device to screen  
@22,10 say "Feed in a paper."  
@23,10 say "press a key to go on.."  
wait+ ' '  
eject  
set device to print  
endif  
use  
tsum4=tsum3 + tuncrec  
@prow()+1,60 say line  
@prow()+1,4 say "Balance as per Bank St  
@prow(),60 say fbale  
use bank_stm  
replace bank_st_ba with bank_bal  
if prow()>55  
set device to screen
```

```
@22,10 say "Feed in another paper."  
@23,10 say "Press a key to go on."  
    wait+ ' '  
    eject  
    set device to print  
    endif  
    @prow()+1,60 say line  
    wait+ ' '  
    set device to screen  
    return  
    endif  
    endif  
    endif  
enddo
```

UNPCH.PRG

```
set talk off
set console off
clear
@ 1,1 to 23,79 double
@ 2,34 SAY "UNPRESENTED CHEQUES MENU"
@ 3,2 to 5,78
define menu unp_mnu
define pad add of unp_mnu prompt "ADD" at 4,4
define pad edt of unp_mnu prompt "EDIT" at 4,18
define pad del of unp_mnu prompt "DELETE" at 4,31
define pad vew of unp_mnu prompt "VIEW" at 4,44
define pad prt of unp_mnu prompt "PRINT" at 4,57
define pad ext of unp_mnu prompt "EXIT" at 4,70
on selection pad add of unp_mnu do unpadd
on selection pad edt of unp_mnu do unpedit
on selection pad del of unp_mnu do unpdel
on selection pad vew of unp_mnu do unpview
on selection pad prt of unp_mnu do unpprt
on selection pad ext of unp_mnu do unpexit

activate menu unp_mnu pad add
return

procedure unpexit
clear
@ 1,1 to 23,78 double
@ 2,22 say "COMPUTERISATION OF BANK RECONCILIATION STATEMENT"
@ 3,2 TO 20,77
deactivate menu
return

PROCEDURE UNPADD
do while .t.
CLEAR
@ 3,2 to 3,78
@ 1,1 to 23,79 double
@ 2,23 say "UNPRESENTED CHEQUE UPDATE MODULE"
STORE CTOD(" / / ") TO DDATE
STORE 0 TO DCHEQ
STORE 0 TO AMNT
store 0 to bank_cod
STORE 0 TO ACCT
store space(30) to bbank
STORE SPACE(100) TO DETAILS
@ 6,10 say "BANK CODE: "
@ 6,25 get bank_cod
@ 8,10 say "BANK: "
@ 8,25 get bbank
@10,10 SAY "ACCOUNT NO.:"
@10,25 GET ACCT
@12,10 SAY "DATE: "
@12,25 GET DDATE PICT '99/99/99'
@ 7,8 TO 7,72
```

```

@14,10 SAY "CHEQUE NO. : "
@14,25 GET DCHEQ PICT '9999999999'
@16,10 SAY "AMOUNT: "
@16,25 GET AMNT PICT '99,999,999.99'
@18,10 SAY "DETAILS: "
@18,25 GET DETAILS PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
READ
USE UNPRECH
APPEND BLANK
replace ban_code with bank_cod
replace bank with bbank
REPLACE DATES WITH DDATE
REPLACE CHEQ_NUM WITH DCHEQ
REPLACE AMOUNT WITH AMNT
REPLACE ACC_NUM WITH ACCT
REPLACE DETAIL WITH DETAILS
USE
@19,8 TO 19,72 DOUBLE
STORE " " TO QUE
@20,10 SAY "ADD MORE UNPRESENTED CHEQUE RECORDS (Y/N)?" GET QUE
PICT 'A'
READ
DO CASE
  CASE QUE $ 'yY'
  LOOP
  CASE QUE $ 'nN'
  use
  clear
  @ 1,1 to 23,79 double
  @ 2,30 SAY "UNPRESENTED CHEQUES MENU"
  @ 3,2 to 5,78
  RETURN
ENDCASE
ENDDO WHILE .T.

```

```

PROCEDURE UNPEDIT
do while .t.
  CLEAR
  @ 3,2 to 3,78
  @ 1,1 to 23,79 double
  @ 2,22 say "UNPRESENTED CHEQUES EDIT MODULE"
  STORE 0 TO BANK_COD
  STORE 0 TO ACCT
  STORE 0 TO DCHEQ
  @ 5,2 say "BANK CODE: " GET BANK_COD pict '9999999'
  @ 5,23 SAY "ACCT. NUMBER: " GET ACCT &&pict 'xxxxxxxxxx'
  @ 5,53 SAY "CHEQ. NUMBER: " GET DCHEQ
  READ
  USE UNPRECH
  DO WHILE .NOT. EOF()
    locate for ban_code=bank_cod
    set filter to BAN_CODE=BANK_COD .AND. acc_num=acct .and.
    cheq_num=dcheq

```

```

IF FOUND( )
  bank_cod=ban_code
  bbank=bank
  ddate=dates
  dcheq=cheq_num
  amnt=amount
  acct=acc_num
  details= detail
  @ 4,2 clear to 5,77
  @ 6,10 say "BANK CODE: "
  @ 6,25 get bank_cod
  @ 8,10 say "BANK: "
  @ 8,25 get bbank
  @10,10 say "DATE: "
  @10,25 get ddate
  @12,10 say "CHEQUE NUMBER: "
  @12,25 get dcheq && pict 'xxxxxxxxxx'
  @14,10 say "AMOUNT: "
  @14,25 get amnt PICT '99,999,999.99'
  @16,10 say "ACCOUNT NUMBER: "
  @16,25 get acct
  @18,10 SAY "DETAILS: "
  @18,25 GET DETAILS PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  READ
  replace bank with bbank
  replace ban_code with bank_cod
  REPLACE DATES WITH DDATE
  REPLACE CHEQ_NUM WITH DCHEQ
  REPLACE AMOUNT WITH AMNT
  REPLACE ACC_NUM WITH ACCT
  REPLACE DETAIL WITH DETAILS
  USE
  @19,8 TO 19,72 DOUBLE
  STORE " " TO QUE
  @20,10 SAY "EDIT MORE UNPRESENTED CHEQUE RECORDS (Y/N)?" GET
  QUE PICT 'A'
  READ
  DO CASE
  CASE QUE '$'yY'
    LOOP
  CASE QUE '$'nN'
    CLOSE ALL
    clear
    @ 1,1 to 23,79 double
    @ 2,27 SAY "UNPRESENTED CHEQUES MENU"
    @ 3,2 to 5,78
  RETURN
  ENDCASE
ENDIF
ENDDO WHILE .NOT. EOF( )
ENDDO WHILE .T.

PROCEDURE UNPDEL

```

```

do while .t.
  CLEAR
  @ 3,2 to 3,78
  @ 1,1 to 23,79 double
  @ 2,22 say "UNPRESENTED CHEQUES DELETE MODULE"
  STORE 0 TO BANK_COD
  STORE 0 TO ACCT
  STORE 0 TO DCHEQ
  @ 5,2 say "BANK CODE: " GET BANK_COD pict '9999999'
  @ 5,23 SAY "ACCT. NUMBER: " GET ACCT &&pict 'xxxxxxxxxxx'
  @ 5,53 SAY "CHEQ. NUMBER: " GET DCHEQ
  READ
  USE UNPRECH
  DO WHILE .NOT. EOF()
    locate for ban_code=bank_cod
    set filter to BAN_CODE=BANK_COD .AND. acc_num=acct .and.
cheq_num=dcheq
    IF FOUND()
      bank_cod=ban_code
      bbank=bank
      ddate=dates
      dcheq=cheq_num
      amnt=amount
      acct=acc_num
      details= detail
      @ 4,2 clear to 5,77
      @ 6,10 say "BANK CODE: "
      @ 6,25 get bank_cod
      @ 8,10 say "BANK: "
      @ 8,25 get bbank
      @10,10 say "DATE: "
      @10,25 get ddate
      @12,10 say "CHEQUE NUMBER: "
      @12,25 get dcheq &&pict 'xxxxxxxxxxx'
      @14,10 say "AMOUNT: "
      @14,25 get amnt PICT '99,999,999.99'
      @16,10 say "ACCOUNT NUMBER: "
      @16,25 get acct
      @18,10 SAY "DETAILS: "
      @18,25 GET DETAILS PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
      READ
      STORE " " TO RESP
      @20,10 SAY "ARE YOU SURE YOU WANT TO DELETE THIS RECORD
(Y/N)?" GET RESP PICT 'A'
      READ
      DO CASE
      CASE RESP $'yY'
        DELETE
        PACK
      ENDC
      USE
      @19,8 TO 19,72 DOUBLE
      @20,10 CLEA TO 20,76
      STORE " " TO QUE

```

```

        @20,10 SAY "DELETE MORE UNPRESENTED CHEQUE RECORDS (Y/N)?"
GET QUE PICT 'A'
  READ
  DO CASE
  CASE QUE '$'yY'
    LOOP
  CASE QUE '$'nN'
    CLOSE ALL
    clear
    @ 1,1 to 23,79 double
    @ 2,27 SAY "UNPRESENTED CHEQUES MENU"
    @ 3,2 to 5,78
  RETURN
  ENDCASE
  ENDIF
  ENDDO WHILE .NOT. EOF()
ENDDO WHILE .T.

```

PROCEDURE UNPVIEW

```

do while .t.
  SET ESCAPE On
  SET EXACT ON
  CLEAR
  USE UNPRECH
  do while .not. eof()
    @ 1,15 to 7,65
    @ 2,18 SAY "MINISTRY OF FINANCE, MINNA"
    store 0 to bank_cod
    @ 3,18 say "BANK CODE: " get bank_cod
    read
    locate for ban_code=bank_cod
    set filter to ban_code=bank_cod
    if found()
      store space(30) to acc_typ
      STORE 0 TO ACCT
      @ 4,18 get ACC_TYP
      @ 4,49 say ":" GET ACCT
      READ
      set filter to acc_num=acct
      if found()
        store ctod(' / / ') to ddate
        @ 5,18 say "UNPRESENTED CHEQUES VIEW AS AT " get ddate
        read
        set filter to dates=ddate
        if found()
          @ 6,18 SAY BANK
          @ 9,4 SAY "DATE"
          @ 9,15 SAY "DETAILS"
          @ 9,55 SAY "CHEQUE NO."
          @ 9,66 SAY "AMOUNT" PICT '99,999,999.99'
          @ 10,2 TO 10,77
          LINCNT=11
          @LINCNT,3 SAY DATES PICT '99/99/99'
          @LINCNT,15 SAY DETAIL
        endif
      endif
    endif
  endwhile
endwhile

```

```

@LINCNT,55 SAY CHEQ_NUM
@LINCNT,66 SAY AMOUNT
LINCNT=LINCNT + 1
if lincnt=20
    store " " to res
    @ 21,6 say "press enter to see the rest..."get res
    read
    wait + ""
    clear
    @1,4 say "DATE"
    @1,15 say "DETAILS"
    @1,55 say "CHEQUE NO."
    @1,66 say "AMOUNT"
    @2,2 to 2,77
    lincnt=3
endif
SKIP
endif
endif
endif
enddo
if eof()
lincnt=lincnt + 3
@lincnt, 30 say " *** END OF FILE ***"
store " " to que
@21,15 say "press a key to return..." get que
read
if que = ""
*wait + ""
clear
return
endif
endif
ENDDO WHILE .T.
SET EXACT OFF

```

```

PROCEDURE UNPPRT
SET ESCAPE On
SET CONSOLE OFF
SET TALK OFF
CLEAR
USE UNPRECH
do while .not. eof()
    @ 1,15 to 7,65 doub
    @ 2,18 SAY "MINISTRY OF FINANCE, MINNA"
    store 0 to bank_cod
    @ 3,18 say "BANK CODE: " get bank_cod
    read
    locate for ban_code=bank_cod
    if found()
        set filter to ban_code=bank_cod
        STORE 0 TO ACCT
        @ 4,18 SAY "ACCOUNT NUMBER: " GET ACCT
        READ
    endif
endif

```



```

if found()
set filter to acc_num=acct
store ctod(' / / ') to ddate
@ 5,20 say "UNPRESENTED CHEQUES VIEW AS AT " get ddate
read
*set filter to dates=ddate
*if found()
SET DEVICE TO PRINT
@PROW(),1 SAY REPL("-",132)
@PROW()+1,18 SAY "MINISTRY OF FINANCE"
@PROW()+1,18 say "BANK CODE: "+ bank_cod
@PROW()+1,18 SAY "ACCOUNT NUMBER: "+ ACCT
@PROW()+1,18 say "UNPRESENTED CHEQUES VIEW AS AT "+ ddate
@PROW()+1,1 SAY REPL("-",132)
@PROW()+2,6 SAY "DATE"
@PROW(),29 SAY "DETAILS"
@PROW(),70 SAY "CHEQUE NO."
@PROW(),83 SAY "AMOUNT"
@PROW()+1,1 SAY REPL("-",132)
@PROW()+2,6 SAY DATES PICT '99/99/99'
@PROW(),29 SAY DETAIL
@PROW(),70 SAY CHEQ_NUM
@PROW(),83 SAY AMOUNT
if PROW()>58
    SET DEVICE TO SCREEN
    store " " to res
    @21,6 say "press enter to see the rest..."get res
    read
    wait + ""
    SET DEVICE TO PRINT
ENDIF
SKIP
loop
SET DEVICE TO SCREEN
else
@21,15 say "Account number not found!"
endif
else
    @21,15 say "bank code not found!"
endif
enddo
if eof()
@PROW()+3, 30 say " *** END OF FILE ***"
SET DEVICE TO SCREEN
@21,15 say "press a key to return..."
WAIT+" "
endif
SET EXACT OFF
clear
@ 3,2 to 5,78
@ 1,1 to 23,79 double
return

```

UNCCH.PRG

```
set talk off
set console off
clear
@ 1,1 to 23,79 double
@ 2,34 SAY "UNCREDITED CHEQUES MENU"
@ 3,2 to 5,78
define menu unc_mnu
define pad add of unc_mnu prompt "ADD" at 4,5
define pad edt of unc_mnu prompt "EDIT" at 4,18
define pad del of unc_mnu prompt "DELETE" at 4,31
define pad vew of unc_mnu prompt "VIEW" at 4,44
define pad prt of unc_mnu prompt "PRINT" at 4,57
define pad ext of unc_mnu prompt "EXIT" at 4,70
on selection pad add of unc_mnu do uncadd
on selection pad edt of unc_mnu do uncedit
on selection pad del of unc_mnu do uncdel
on selection pad vew of unc_mnu do uncvew
on selection pad prt of unc_mnu do uncppt
on selection pad ext of unc_mnu do unccxit

activate menu unc_mnu pad add
return

procedure unccxit
clear
@ 1,1 to 23,78 double
@ 2,22 say "COMPUTERISATION OF BANK RECONCILIATION STATEMENT"
@ 3,2 TO 20,77
deactivate menu
return

PROCEDURE UNCADD
do while .t.
CLEAR
@ 3,2 to 3,78
@ 1,1 to 23,79 double
@ 2,24 say "UNCREDITED CHEQUES ADD MODULE"
STORE CTOD(" / / ") TO DDATE
STORE 0 TO DCHEQ
STORE 0 TO AMNT
store 0 to bank_cod
STORE 0 TO ACCT
store space(30) to bbank
STORE SPACE(100) TO DETAILS
@ 6,10 say "BANK CODE: "
@ 6,25 get bank_cod
@ 8,10 say "BANK: "
@ 8,25 get bbank
@10,10 SAY "ACCOUNT NO.:"
@10,25 GET ACCT
@12,10 SAY "DATE: "
@12,25 GET DDATE PICT '99/99/99'
@ 7,8 TO 7,72
```

```

@14,10 SAY "CHEQUE NO.: "
@14,25 GET DCHEQ PICT '9999999999'
@16,10 SAY "AMOUNT: "
@16,25 GET AMNT PICT '99,999,999.99'
@18,10 SAY "DETAILS: "
      @ 1 8 , 2 5      G E T      D E T A I L S      P I C T
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  READ
  USE UNCRECH
  APPEND BLANK
  replace ban_code with bank_cod
  replace bank with bbank
  REPLACE DATES WITH DDATE
  REPLACE CHEQ_NUM WITH DCHEQ
  REPLACE AMOUNT WITH AMNT
  REPLACE ACC_NUM WITH ACCT
  REPLACE DETAIL WITH DETAILS
  USE
  @19,8 TO 19,72 DOUBLE
  STORE " " TO QUE
  @20,10 SAY "ADD MORE UNCREDITED CHEQUE RECORDS (Y/N)?" GET QUE
PICT 'A'
  READ
  DO CASE
    CASE QUE $ 'yY'
    LOOP
    CASE QUE $ 'nN'
    use
    clear
    @ 1,1 to 23,79 double
    @ 2,27 SAY "UNCREDITED CHEQUES MENU"
    @ 3,2 to 5,78
  RETURN
  ENDCASE
ENDDO WHILE .T.

PROCEDURE UNCREDIT
do while .t.
  CLEAR
    @ 3,2 to 3,78
    @ 1,1 to 23,79 double
  @ 2,23 say "UNCREDITED CHEQUES EDIT MODULE"
  STORE 0 TO BANK_COD
  STORE 0 TO ACCT
  STORE 0 TO DCHEQ
  @ 5,2 say "BANK CODE: " GET BANK_COD pict '9999999'
  @ 5,23 SAY "ACCT. NUMBER: " GET ACCT &&pict 'xxxxxxxxxx'
  @ 5,53 SAY "CHEQ. NUMBER: " GET DCHEQ
  READ
  USE UNCRECH
  DO WHILE .NOT. EOF()
    locate for ban_code=bank_cod
    set filter to BAN_CODE=BANK_COD .AND. acc_num=acct .and.
    cheq_num=dcheq

```

```

IF FOUND( )
  bank_cod=ban_code
  bbank=bank
  ddate=dates
  dcheq=cheq_num
  amnt=amount
  acct=acc_num
  details= detail
  @ 4,2 clear to 5,77
  @ 6,10 say "BANK CODE: "
  @ 6,25 get bank_cod
  @ 8,10 say "BANK: "
  @ 8,25 get bbank
  @10,10 say "DATE: "
  @10,25 get ddate
  @12,10 say "CHEQUE NUMBER: "
  @12,25 get dcheq &&pict 'xxxxxxxxxxx'
  @14,10 say "AMOUNT: "
  @14,25 get amnt pict '99,999,999.99'
  @16,10 say "ACCOUNT NUMBER: "
  @16,25 get acct
  @18,10 SAY "DETAILS: "
  @18,25 GET DETAILS PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  READ
  replace bank with bbank
  replace ban_code with bank_cod
  REPLACE DATES WITH DDATE
  REPLACE CHEQ_NUM WITH DCHEQ
  REPLACE AMOUNT WITH AMNT
  REPLACE ACC_NUM WITH ACCT
  REPLACE DETAIL WITH DETAILS
  USE
  @19,8 TO 19,72 DOUBLE
  STORE " " TO QUE
  @20,10 SAY "EDIT MORE UNCREDITED CHEQUE RECORDS (Y/N)?" GET
  QUE PICT 'A'
  READ
  DO CASE
  CASE QUE $'yY'
    LOOP
  CASE QUE $'nN'
    CLOSE ALL
    clear
    @ 1,1 to 23,79 double
    @ 2,27 SAY "UNCREDITED CHEQUES MENU"
    @ 3,2 to 5,78
  RETURN
  ENDCASE
ENDIF
ENDDO WHILE .NOT. EOF( )
ENDDO WHILE .T.

```

```

PROCEDURE UNCDEL

```

```

do while .t.
  CLEAR
    @ 3,2 to 3,78
    @ 1,1 to 23,79 double
  @ 2,23 say "UNCREDITED CHEQUES DELETE MODULE"
  STORE 0 TO BANK_COD
  STORE 0 TO ACCT
  STORE 0 TO DCHEQ
  @ 5,2 say "BANK CODE: " GET BANK_COD pict '9999999'
  @ 5,23 SAY "ACCT. NUMBER: " GET ACCT &&pict 'xxxxxxxxxx'
  @ 5,53 SAY "CHEQ. NUMBER: " GET DCHEQ
  READ
  USE UNCRECH
  DO WHILE .NOT. EOF()
    locate for ban_code=bank_cod
    set filter to BAN_CODE=BANK_COD .AND. acc_num=acct .and.
cheq_num=dcheq
    IF FOUND()
      bank_cod=ban_code
      bbank=bank
      ddate=dates
      dcheq=cheq_num
      amnt=amount
      acct=acc_num
      details= detail
      @ 4,2 clear to 5,77
      @ 6,10 say "BANK CODE: "
      @ 6,25 say bank_cod
      @ 8,10 say "BANK: "
      @ 8,25 say bbank
      @10,10 say "DATE: "
      @10,25 say ddate
      @12,10 say "CHEQUE NUMBER: "
      @12,25 say dcheq &&pict 'xxxxxxxxxx'
      @14,10 say "AMOUNT: "
      @14,25 say amnt pict '99,999,999.99'
      @16,10 say "ACCOUNT NUMBER: "
      @16,25 say acct
      @18,10 say "DETAILS: "
      @18,25 say DETAILS PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
      READ
      USE
      STORE " " TO resp
      @20,10 SAY "ARE YOU SURE YOU WANT TO DELETE THIS
RECORD(Y/N)?" GET RESP PICT 'A'
      READ
      DO CASE
      CASE RESP $'YY'
        DELETE
        PACK
      ENDC
      @19,8 TO 19,72 DOUBLE
      STORE " " TO QUE
      @20,10 SAY "DELETE MORE UNCREDITED CHEQUE RECORDS (Y/N)?"

```

```

GET QUE PICT 'A'
  READ
  DO CASE
  CASE QUE '$'yY'
    LOOP
  CASE QUE '$'nN'
    CLOSE ALL
    clear
    @ 1,1 to 23,79 double
    @ 2,27 SAY "UNCREDITED CHEQUES MENU"
    @ 3,2 to 5,78
  RETURN
  ENDCASE
ENDIF
ENDDO WHILE .NOT. EOF()
ENDDO WHILE .T.

```

```

PROCEDURE UNCVIEW
do while .t.
  SET ESCAPE On
  SET EXACT ON
  CLEAR
  USE UNCRECH
  do while .not. eof()
    @ 3,2 to 3,78
    @ 1,1 to 23,79 double
    @ 2,18 SAY "MINISTRY OF FINANCE, MINNA"
    store 0 to bank_cod
    @ 3,18 say "BANK CODE: " get bank_cod
    read
    locate for ban_code=bank_cod
    set filter to ban_code=bank_cod
    if found()
      STORE 0 TO ACCT
      @ 4,18 SAY "ACCOUNT NUMBER: " GET ACCT
      READ
      set filter to acc_num=acct
      if found()
        store ctod(' / / ') to ddate
        @ 5,18 say "UNCREDITED CHEQUES VIEW AS AT " get ddate
        read
        set filter to dates=ddate
        if found()
          @ 6,18 SAY BANK
          @ 9,4 SAY "DATE"
          @ 9,15 SAY "DETAILS"
          @ 9,55 SAY "CHEQUE NO."
          @ 9,66 SAY "AMOUNT"
          @ 10,2 TO 10,77
          LINCNT=11
          @LINCNT,3 SAY DATES PICT '99/99/99'
          @LINCNT,15 SAY DETAIL
          @LINCNT,55 SAY CHEQ_NUM
        endif
      endif
    endif
  endwhile
endwhile

```

```

    @LINCNT,66 SAY AMOUNT PICT '99,999,999.99'
    LINCNT=LINCNT + 1
    if lincnt=20
        store " " to res
        @21,6 say "press enter to see the rest..."get res
        read
        wait + ""
        clear
        @1,4 say "DATE"
        @1,15 say "DETAILS"
        @1,55 say "CHEQUE NO."
        @1,66 say "AMOUNT"
        @2,2 to 2,77
        lincnt=3
    endif
    SKIP
endif
endif
endif
enddo
if eof()
lincnt=lincnt + 3
@lincnt, 30 say " *** END OF FILE ***"
store " " to que
@21,15 say "press a key to return..." get que
read
if que = ""
*wait + ""
clear
@ 1,1 to 23,79 double
@ 2,27 SAY "UNCREDITED CHEQUES MENU"
@ 3,2 to 5,78
return
endif
endif
ENDDO WHILE .T.
SET EXACT OFF

```

PROCEDURE UNCPRT

```

SET ESCAPE On
SET TALK OFF
SET CONSOLE OFF
CLEAR
USE UNCRECH
do while .not. eof()
@ 1,15 to 7,65 doub
@ 2,18 SAY "MINISTRY OF FINANCE, MINNA"
store 0 to bank_cod
@ 3,18 say "BANK CODE: " get bank_cod
read
locate for ban_code=bank_cod
if found()
set filter to ban_code=bank_cod
STORE 0 TO ACCT

```

```

@ 4,18 SAY "ACCOUNT NUMBER: " GET ACCT
READ
if found()
set filter to acc_num=acct
store ctod(' / / ') to ddate
@ 5,20 say "UNCREDITED CHEQUES VIEW AS AT " get ddate
read
SET DEVICE TO PRINT
@PROW(),1 SAY REPL("-",132)
@PROW()+1,18 SAY "MINISTRY OF FINANCE, MINNA"
@PROW()+1,18 say "BANK CODE: "+ bank_cod
@PROW()+1,18 SAY "ACCOUNT NUMBER: "+ ACCT
@PROW()+1,18 say "UNCREDITED CHEQUES VIEW AS AT "+ ddate
@PROW()+1,1 SAY REPL("-",132)
@PROW()+3,6 SAY "DATE"
@PROW(),29 SAY "DETAILS"
@PROW(),70 SAY "CHEQUE NO."
@PROW(),83 SAY "AMOUNT"
@PROW()+1,1 SAY REPL("-",132)
@PROW()+2,6 SAY DATES PICT '99/99/99'
@PROW(),29 SAY DETAIL
@PROW(),70 SAY CHEQ_NUM
@PROW(),83 SAY AMOUNT
if PROW()>58
    SET DEVICE TO SCREEN
    store " " to res
    @21,6 say "press enter to see the rest..."get res
    read
    wait + ""
    SET DEVICE TO PRINT
ENDIF
SKIP
loop
SET DEVICE TO SCREEN
else
@21,15 say "Account number not found!"
endif
else
    @21,15 say "bank code not found!"
endif
enddo
if eof()
    @PROW()+3, 30 say " *** END OF FILE ***"
    SET DEVICE TO SCREEN
    @21,15 say "press a key to return..."
    WAIT+" "
endif
SET EXACT OFF
clear
@ 3,2 to 5,78
@ 1,1 to 23,79 double
return

```


RECPT.PRG

```
set talk off
set console off
clear
@ 1,1 to 23,79 double
@ 2,34 SAY "RECEIPT UPDATE MENU"
@ 3,2 to 5,78
define menu rcpt_mnu
define pad add of rcpt_mnu prompt "ADD" at 4,5
define pad edt of rcpt_mnu prompt "EDIT" at 4,18
define pad del of rcpt_mnu prompt "DELETE" at 4,31
define pad vew of rcpt_mnu prompt "VIEW" at 4,44
define pad prt of rcpt_mnu prompt "PRINT" at 4,57
define pad ext of rcpt_mnu prompt "EXIT" at 4,70
on selection pad add of rcpt_mnu do recadd
on selection pad edt of rcpt_mnu do recedit
on selection pad del of rcpt_mnu do recdel
on selection pad vew of rcpt_mnu do recview
on selection pad prt of rcpt_mnu do recprt
on selection pad ext of rcpt_mnu do recexit

activate menu rcpt_mnu pad add
return

procedure recexit
clear
@ 1,1 to 23,78 double
@ 2,22 say "COMPUTERISATION OF BANK RECONCILIATION STATEMENT"
@ 3,2 TO 20,77
deactivate menu
return

PROCEDURE RECADD
do while .t.
CLEAR
@ 3,2 to 3,78
@ 1,1 to 23,79 double
@ 2,30 say "RECIPT UPDATE MODULE"
STORE CTOD(" / / ") TO DDATE
STORE 0 TO DCHEQ
STORE 0 TO AMNT
store 0 to bank_cod
STORE 0 TO ACCT
store space(30) to bbank
STORE SPACE(100) TO DETAILS
@ 6,10 say "BANK CODE: "
@ 6,25 get bank_cod
@ 8,10 say "BANK: "
@ 8,25 get bbank pict '@!'
@10,10 SAY "ACCOUNT NO.:"
@10,25 GET ACCT
@12,10 SAY "DATE: "
@12,25 GET DDATE PICT '99/99/99'
@ 7,2 TO 7,78
```

```

@14,10 SAY "CHEQUE NO.: "
@14,25 GET DCHEQ PICT '9999999999'
@16,10 SAY "AMOUNT: "
@16,25 GET AMNT PICT '99,999,999.99'
@18,10 SAY "DETAILS: "
@18,25 GET DETAILS PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
READ
USE reciepts
APPEND BLANK
replace ban_code with bank_cod
replace bank with bbank
REPLACE DATES WITH DDATE
REPLACE CHEQ_NUM WITH DCHEQ
REPLACE AMOUNT WITH AMNT
REPLACE ACC_NUM WITH ACCT
REPLACE DETAIL WITH DETAILS
USE
USE BANK_STM
replace ban_code with bank_cod
REPLACE ACC_NUM WITH ACCT
REPLACE DATES WITH DDATE
USE
@19,8 TO 19,72 DOUBLE
STORE " " TO QUE
@20,10 SAY "ADD MORE RECIEPT RECORDS (Y/N)?" GET QUE PICT 'A'
READ
DO CASE
    CASE QUE $ 'yY'
    LOOP
        CASE QUE $ 'nN'
        use
        clear
            @ 1,1 to 23,79 double
            @ 2,27 SAY "RECIEPT UPDATE MENU"
            @ 3,2 to 5,78
        RETURN
    ENDCASE
ENDDO WHILE .T.

PROCEDURE RECEDIT
do while .t.
CLEAR
@ 3,2 to 3,78
@ 1,1 to 23,79 double
@ 2,30 SAY "RECIEPT MODIFICATION MODULE"
STORE 0 TO BANK_COD
STORE 0 TO ACCT
STORE 0 TO DCHEQ
@ 5,2 say "BANK CODE: " GET BANK_COD pict '99999999'
@ 5,23 SAY "ACCT. NUMBER: " GET ACCT &&pict 'xxxxxxxxxx'
@ 5,53 SAY "CHEQ. NUMBER: " GET DCHEQ
READ
USE RECIEPTs
DO WHILE .NOT. EOF()

```

```
locate for ban_code=bank_cod
set filter to BAN_CODE=BANK_COD .AND. acc_num=acct &&.and.
cheq_num=dcheq
IF FOUND()
```

```
bank_cod=ban_code
bbank=bank
ddate=dates
dcheq=cheq_num
amnt=amount
acct=acc_num
details= detail
@ 4,2 clear to 5,77
@ 6,10 say "BANK CODE: "
@ 6,25 get bank_cod
@ 8,10 say "BANK: "
@ 8,25 get bbank pict '@!'
@10,10 say "DATE: "
@10,25 get ddate
@12,10 say "CHEQUE NUMBER: "
@12,25 get dcheq &&pict 'xxxxxxxxxxx'
@14,10 say "AMOUNT: "
@14,25 get acct PICT '9,999,999,999.99'
@16,10 say "ACCOUNT NUMBER: "
@16,25 get amnt pict '99,999,999.99'
@18,10 SAY "DETAILS: "
@18,25 GET DETAILS PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
READ
```

```
USE RECIEPTs
set filter to BAN_CODE=BANK_COD .AND. acc_num=acct
replace bank with bbank
replace ban_code with bank_cod
REPLACE DATES WITH DDATE
REPLACE CHEQ_NUM WITH DCHEQ
REPLACE AMOUNT WITH AMNT
REPLACE ACC_NUM WITH ACCT
REPLACE DETAIL WITH DETAILS
USE
```

```
set filter to BAN_CODE=BANK_COD .AND. acc_num=acct
```

```
USE BANK_STM
replace ban_code with bank_cod
REPLACE ACC_NUM WITH ACCT
REPLACE DATES WITH DDATE
USE
```

```
@19,8 TO 19,72 DOUBLE
```

```
STORE " " TO QUE
```

```
@20,10 SAY "EDIT MORE RECIEPT RECORDS (Y/N)?" GET QUE PICT
```

'A'

```
READ
DO CASE
CASE QUE $'yY'
LOOP
CASE QUE $'nN'
clear
@ 1,1 to 23,79 double
```

```

        @ 2,27 SAY "RECIEPT UPDATE MENU"
        @ 3,2 to 5,78
        RETURN
    ENDCASE
ENDIF
ENDDO WHILE .NOT. EOF()
ENDDO WHILE .T.

```

```

PROCEDURE RECDEL

```

```

do while .t.
    CLEAR
    @ 3,2 to 3,78
    @ 1,1 to 23,79 double
    @ 2,30 SAY "RECIEPT DELETE MODULE"
    STORE 0 TO BANK_COD
    STORE 0 TO ACCT
    STORE 0 TO DCHEQ
    @ 5,2 say "BANK CODE: " GET BANK_COD pict '9999999'
    @ 5,23 SAY "ACCT. NUMBER: " GET ACCT &&pict 'xxxxxxxxxx'
    @ 5,53 SAY "CHEQ. NUMBER: " GET DCHEQ
    READ
    USE RECIEPTs
    DO WHILE .NOT. EOF()
        locate for ban_code=bank_cod
        set filter to BAN_CODE=BANK_COD .AND. acc_num=acct &&.and.
cheq_num=dcheq
        IF FOUND()
            bank_cod=ban_code
            bbank=bank
            ddate=dates
            dcheq=cheq_num
            amnt=amount
            acct=acc_num
            details= detail
            @ 4,2 clear to 5,77
            @ 6,10 say "BANK CODE: "
            @ 6,25 say bank_cod
            @ 8,10 say "BANK: "
            @ 8,25 say bbank pict '@!'
            @10,10 say "DATE: "
            @10,25 say ddate
            @12,10 say "CHEQUE NUMBER: "
            @12,25 say dcheq &&pict 'xxxxxxxxxx'
            @14,10 say "AMOUNT: "
            @14,25 say acct PICT '9,999,999,999.99'
            @16,10 say "ACCOUNT NUMBER: "
            @16,25 say amnt pict '99,999,999.99'
            @18,10 say "DETAILS: "
            @18,25 say DETAILS PICT 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
            READ
            store ' ' to resp
            @20,10 SAY "ARE YOU SURE YOU WANT TO DELETE THIS
RECORD(Y/N)?" GET RESP

```

```

READ
DO CASE
CASE RESP '$'yY'
  DELETE
  PACK
ENDC
USE
@19,8 TO 19,72 DOUBLE
@20,10 CLEA TO 20,75
STORE " " TO QUE
@20,10 SAY "DELETE MORE RECIEPT RECORDS (Y/N)?" GET QUE PICT
'A'
READ
DO CASE
CASE QUE '$'yY'
  LOOP
CASE QUE '$'nN'
  clear
  @ 1,1 to 23,79 double
  @ 2,27 SAY "RECIEPT UPDATE MENU"
  @ 3,2 to 5,78
  RETURN
ENDCASE
ENDIF
ENDDO WHILE .NOT. EOF()
ENDDO WHILE .T.

```

PROCEDURE RECVIEW

```

SET ESCAPE On
SET EXACT ON
CLEAR
USE RECIEPTs
do while .not. eof()
  @ 1,15 to 7,65 doub
  @ 2,25 SAY "MINISTRY OF FINANCE, MINNA"
  store 0 to bank_cod
  @ 3,18 say "BANK CODE: " get bank_cod
  read
  locate for ban_code=bank_cod
  if found()
    set filter to ban_code=bank_cod
    store space(30) to acc_type
    STORE 0 TO ACCT
    @ 4,18 get ACC_TYP
    @ 4,49 say ":" GET ACCT
    READ
    if found()
      set filter to acc_num=acct
      store ctod(' / / ') to ddate
      @ 5,23 say "RECIEPTS VIEW AS AT " get ddate
      read
      @ 9,6 SAY "DATE"
      @ 9,19 SAY "DETAILS"
    endif
  endif
endif

```

```

@ 9,50 SAY "CHEQUE NO."
@ 9,63 SAY "AMOUNT"
@ 10,5 TO 10,75
LINCNT=11
@LINCNT,6 SAY DATES PICT '99/99/99'
@LINCNT,19 SAY DETAIL
@LINCNT,50 SAY CHEQ_NUM
@LINCNT,63 SAY AMOUNT
LINCNT=LINCNT + 1
if lincnt=20
  store " " to res
  @21,6 say "press enter to see the rest..."get res
  read
  wait + ""
  clear
  @1,6 say "DATE"
  @1,19 say "DETAILS"
  @1,50 say "CHEQUE NO."
  @1,63 say "AMOUNT"
  @2,5 to 2,75
  lincnt=3
endif
SKIP
loop
*endif
else
@21,15 say "Account number not found!"
endif
else
@21,15 say "bank code not found!"
endif
enddo
if eof()
lincnt=lincnt + 3
@lincnt, 30 say " *** END OF FILE ***"
store " " to que
@21,15 say "press a key to return..." get que
read
if que = ""
*wait + ""
endif
endif
SET EXACT OFF
clear
@ 3,2 to 5,78
@ 1,1 to 23,79 double
return

PROCEDURE RECPRT
SET ESCAPE On
SET EXACT ON
CLEAR
USE RECIEPTs
do while .not. eof()

```

```

@ 1,15 to 7,65 doub
@ 2,25 SAY "MINISTRY OF FINANCE, MINNA"
store 0 to bank_cod
@ 3,18 say "BANK CODE: " get bank_cod
read
locate for ban_code=bank_cod
if found()
  set filter to ban_code=bank_cod
  STORE 0 TO ACCT
  @ 4,18 SAY "ACCOUNT NUMBER: " GET ACCT
  READ
  if found()
    set filter to acc_num=acct
    store ctod(' / / ') to ddate
    @ 5,23 say "RECIEPTS VIEW AS AT " get ddate
    read
    *set filter to dates=ddate
    *if found()
    SET DEVICE TO PRINT
    @PROW(),1 SAY REPL("-",132)
    @PROW()+1,18 SAY "MINISTRY OF FINANCE, MINNA"
    @PROW()+1,18 say "BANK CODE: "+ ban_code
    @PROW()+1,18 SAY "ACCOUNT NUMBER: "+ ACC_NUM
    @PROW()+1,18 say "RECIEPTS VIEW AS AT "+ date()
    @PROW()+1,1 SAY REPL("-",132)
    @PROW()+2,6 SAY "DATE"
    @PROW(),29 SAY "DETAILS"
    @PROW(),70 SAY "CHEQUE NO."
    @PROW(),83 SAY "AMOUNT"
    @PROW()+4,1 SAY REPL("-",132)
    @PROW()+2,6 SAY DATES PICT '99/99/99'
    @PROW(),29 SAY DETAIL
    @PROW(),70 SAY CHEQ_NUM
    @PROW(),83 SAY AMOUNT
    if PROW()>58
      SET DEVICE TO SCREEN
      store " " to res
      @21,6 say "press enter to see the rest..."get res
      read
      wait + ""
      SET DEVICE TO PRINT
    ENDIF
    SKIP
    loop
    else
    SET DEVICE TO SCREEN
    @21,15 say "Account number not found!"
    endif
    else
    @21,15 say "bank code not found!"
  endif
endif
enddo
if eof()
  @PROW()+3, 30 say " *** END OF FILE ***"

```

```
SET DEVICE TO SCREEN
@21,15 say "press a key to return..."
WAIT+ " "
endif
SET EXACT OFF
clear
@ 3,2 to 5,78
@ 1,1 to 23,79 double
return
```