COMPUTERISED PAYROLL SYSTEM

A CASE STUDY OF

NIGER STATE AGRICULTURAL DEVELOPMENT PROJECT, MINNA

BY

ABDULKADIR HAUWA YAHAYA
PGD/MSC/114/96

A PROJECT SUBMITTED TO THE DEPARTMENT OF MATHEMATICS/COMPUTER SCIENCE OF FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA IN PARTIAL FULFILLMENT FOR THE AWARD OF POST-GRADUATE DIPLOMA IN COMPUTER SCIENCE.

MARCH 1998

APPROVAL PAGE

This is to certify that this project is an original work undertaken by Abdulkadir Hauwa Yahaya and has been prepared in accordance with the regulations governing the preparation and presentation of project in the University of Technology.

DR. AIYESIMI, Y. M. PROJECT SUPERVISOR	DATE
DR. K. R. ADEBOYE HEAD OF DEPARTMENT	DATE
EXTERNAL EXAMINER	DATE

DECLARATION

I hereby declare that this research project has been conducted solely by me under the guidance of Dr. Aiyesimi Y. M. Mathematic lecturer of the Deptartment of Mathematics & Computer Science, Federal University of Technology, Minna and has neither copied someone's work nor has someone done it for me.

DEDICATION

This research work is dedicated to my beloved parents Late Alhaji Yahaya Sayuti and Late Hajiya Hassana.

Dedication also goes to the following:

1. Hajiya Mama Yahaya
2. Mallama Maryam Yahaya
3. Master Isah Abdulkadir
4. Master Idris "
5. Miss Fatima "
6. Miss Hassana "
7. Miss Aishat "
8. Miss Hauwa "

ACKNOWLEDGEMENT

The task of undertaking and compiling a project work is by no maens an easy job. In view of this I wish to express my unflinching gratitude to those who contributed in one way or the other to the success of this work.

My sincere appreciation and thanks goes to Prof. K.A. Adeboye, H.O.D, my project supervisor, Dr. Aiyesimi Y.M., Prince R.A. Badmus, Mr. L.N. Ezeako, Mr. I.K. Adewale, Mr. Kola Raheem, Mr. Micah Dogara, Mall. Audu and Mr. Echiogar who despite their daily committments took pains to ensure the success of this work by giving me professional advices and necessary corrections.

I must also express my sincere appreciation to the Managing Director, Niger State Agricultural Development Project, Alhaji Abdullahi Danyaya for his co-operation towards the completion of my course.

I wish to express my sincere appreciation to the Director of Finance, Niger State Agricultural Development Project, Mallam Suleiman Abubakar for his co-operation towards the completion of this course.

I also wish to express my profound gratitude to my Wives, Children, Brothers, Sisters & Friends especially Alh. Moh'd B. Mustafa, Director Planning, Niger State Agricultural Development Project, Minna for their moral and materials supports and to the Computer Staff particularly Mr. Babatunde Omojola and Mrs. Moji Ibrahim for typing the manuscripts.

ABSTRACT

This project work establishes the need to develop new computerised procedures for the payroll activities in Niger State Agricultural Development Project (NSADP) Minna. It emphasizes the need for a new system to replace the existing one based on the current and the expected problems that are inherent in the current procedures used in the project.

To this end, this project first discusses the Agric. Dev. Project and the existing payroll system. However, after the consideration of the problems, it was recommended that a new system needs to be designed. Based on this objective and the system study, the logical and the physical specification of the proposed system was designed. The logical design computerises of the output specification, input speficication, file design and the systems procedures of the proposed system while the physical design contains the physical coustruction in terms of program developed to achieve the objective of the system.

Finally, the mode of operation of the proposed system is analysed and its stages of implementation. The implimentation stage is done in a way to ensure reliability and continuity of payroll activities in Niger State Agric. Dev. Project, Minna.

TABLE OF CONTENTS

CHAPTER FOUR

SYSTEM IMPLEMENTATION

CHAPTER FIVE

CHAPTER ONE

AN OVERVIEW OF PAYROLL SYSTEM IN NSADP MINNA

In a task of this nature, there is always the need to develop a proper understanding of the important concepts that will be involved in the study. This will ensure having a good background of the task as well as establishing the relationship between the concepts. It is as a result of this that this chapter, first deals with the necessary preambles to the subject matter. It then goes further to study the existing payroll system in Niger State A.D.P Minna in order to identify its problem areas.

1.1 INTRODUCTION

The link between computer technology and our present days is evident by the increased activities world wide. This has gone to such an extent that it is obvious that human efforts alone can no longer cope with the increased pace of these activities. Therefore, with the industrial revolution which brought about the introduction and application of computers in many diverse fields the expected problems in one way under control. This is because a computer has the capability of processing a large data within a very short period of time and with the most possible accuracy.

It is therefore, important to note that one of the areas that has benefitted mostly from the expansion of computer technology is the area of business. Business of course, uses computers for a variety of tasks which include payroll processing, Inventory Control, Account Receivable and account-payable, information

management personnel information management etc.

Howevers of all the various uses of computers in business environment, the one that readily comes to mind most frequently is in the area of payroll processing. This is probably due to the fact that major savings can be made in the area of clerical labour since many hands are always required in manual operation. Furthermore, there is need for accurate and urgent payroll processing and necessary reports generated without wasting much time. This is essential so that employee's morale is not adversaly affected through delayed payment of salaries and wages. Also as a side benefit a payroll system provide, management with accounting information such as data for general ladger entries job costing and labour distribution and federal yearly, quarterly or monthly tax reports.

In recognition of the above facts the project gives a vivid analysis of the step by step requirements on the computarisation of a payroll system in a public organisation, a case study of Niger State Agricultural Development Project (NSADP) Minna. The State ADP is however chosen as a case study mainly because of its large number of employees and all other relevant reasons which makes it a public organisation.

BRIEF HISTORY OF NIGER STATE ADP

Niger State Agricultural Development Project Minna, is one of the five Agric. Project, covered by second Multi-State Agricultural Development Project (MSADP II) whose loan of $85.2 million was

negotiated and approved in 1988 with the World Bank by Federal Government of Nigeria. The amount of this loan to Niger State was US $ 18.2 million while the total project cost was estimated at US $ 28.5 million. The balance of the cost was to be provided by Federal and Niger State Governments. The loan became effective in November, 1989. The expected loan lifespan was four years but had to be extended two times of one year each thereby moking June 30th, 1995 the closing date.

The Niger State Agricultural Development Project, Minna's main objective is to increase small holder productivity and rural incomes and welfare through:

(a) The strengthening of Agricultural Services
(b) Improving the coverage and maintenance of supporting rural infrastructure
(c) Strengthening of supporting commercial services.
(d) Strengthening of the management and technical staff capabilities; and
(e) Strengthening of the capacity for policy formulation, planning and expenditure programming in the state Ministry of Agric. and Natural Resoures.

Generally, the main objective is to prouide management and administrative support to project for the efficient and effective day to day running of the activities of other subprogrammes.

A proper adminstrative machinary has been put in place. The Administrative Department has handled recruitment of staff the project and production of series of administrative guidelines for compliance. The Niger State Government (NGSG) edict establishing the project was enacted in 1988 and the Niger State Agricultural Development Project Executive Committee (NADPEC) was constituted in earnest. The project is hence managed by a Project Management Unit (PMU) headed by a Managing Director under the policy guidelines and directions formulated by the NADPEC chaired by the state chief Executive.

The operational areas of the project are the three Zones Viz: Zone I whose Headquarters is at Bida, Zone II whose Headquarters is at Kuta and Zone III whose Headquarters is at Kontagora. The project has three (3) operational (Core) subprogrammes and four (4) supportives. The former are Agricultural and Technical Services, Commercial Services and Engineering (Rural Infrastructural Services) while the later are management and Administration, Accounts and Finance, Human Resources Development and Planning Monitoring and Evaluation. For effective implementation of project activities both the NAPEC and PMU have got sub committes important of which are Establishment sub-committee, Finance and General Purpose Sub-committee, Agricultural Development Services Sub-committee for the NAPEC and Establishment sub-committee for staff on GL 01-10 and Disciplinary committee for the PMU. Meanwhile, meetings of the ADPEC committees and its sub-committees have not been regular while the main PMU committee meetings are held monthly

(the last Wednesday of every month) and those of its sub-committees are held as needs arise.

Two other main components viz INTERNAL AUDIT and procurement and stores belong to this sub-programme. Each of these components has performed its assigned roles. Infact the activities of the former are better appreciated considering its support to the management while the later resulted into the 3 International Competitive biddings (ICBs) floated by the funding project.

CHAPTER TWO

2.1 LITERATURE REVIEW

A payroll system is a system which involves all activities that has to do with the preparation of salaries and wages to employees. It handles all informations necessary for processing employees take-home pay and the generation of necessary reports. The system takes cognisance of the basic salary or wage rate and number of hours worked, tax charges, and allowances and deductions as may be agreed upon in order to calculate employees net pay at regular intervals. It goes further to produce various reports such as payslip, bank schedule, tax report, deduction list etc.

Therefore the overall objectives of payroll system are as follows:-

1. To allow accruals of salaries and wages only for services actually rendered to the entity.
2. To comply with government regulations like on tax charges and to keep the payroll rocords in sufficient details so as to provide information for use in compliance with the regulations.
3. To process data quickly enough to permit prompt payment of salaries and wages that are due.

However, a payroll system takes two forms viz: Salaries and Wages. Salaries refer to the compensation of employees based upon an agreed stable monthly or annual rate pay wages, on the other hand refer to earnings of employees who work for an hourly or

weekly rate of pay based on the number of hours worked or quantity of output produced or any other yardstick that can be applied.

The Niger State ADP Minna, pays salaries to its employees and this is done on monthly basis by consideraing the basic salary, the allowances, and the deduction applited to each of the employees.

As stated above, the Niger State ADP has total number of 1380 employees to pay salary at the end of each month. In order to cater for the walfare of these staff especially, in the area of prompt payment of salary without any delay and to ensures arithmetic accuracy, the Agric. Project introduced an automated procedures via the use of computer in processing the payroll monthly.

A computer is an electronic device that solves problems by applying prescribed operations on data entered into it. It has the capacity to input, process, store and output data and information. Computer can perform data processing operations accurately at high speeds without much human intervention. Therefore, the basic functions of every computer system are input, storage, control, process and output which are determined by the variety of programs designed largely to assist users to run jobs

2.6 FEASIBILITY STUDY

In system development, feasibility study is an important stage since it involves the process of gathering and interpreting facts in order to evolve a proper understanding of a system so as to diagoinsed the problems associated to it. The outcome of this

analysis is used to determine what must be done to solve the problem that could emsnate from the system.

However, in an attempt to analyse the existing payroll system in NSADP Minna, an investigation was carried out in order to determine how the efficiency of the payroll activities can be enhanced. Given this objective, the investigation was carried out in collaboration with the users of the existing system. The users also made suggestions which can be considered in case of the need to design a new system.

2.3 METHODS OF INVESTIGATION

Specifically, the methods adopted in gattering informations on the existing system and other considerations are stated below:-

i. **OBSERVATION:-** This method is used to directly study the operations of the existing system.

ii. **RECORD REVIEW:-** Written information such as forms and reports used in the operation at the system were reviewed and analysed.

iii. **INTERVIEWING:-** This is used mainly to confirm some informations gathered using the above two methods. It was also used to obtain informations or suggestions that can be

considered relevant to the proposed system.

2.4 THE EXISTING SYSTEM

In Niger State ADP, Minna the existing computerised procedures is ran on monthly baris to process payroll and generate various reports that would be required. All that the users perform is to input the necessary data which may be in the form of including or modifying existing staff records in the payroll or perform the available variation entries after which the payroll is processed. The payroll processing and printing of reports are done mainly by selecting from the menu available in the system.

There are other tasks the users of the existing system performs and they include updating some referance files like Allowances and Deduction file, Bank file, Department file etc. These are also done through menu selection.

2.5 OBSERVATIONS ON THE EXISTING SYSTEM

In the course of analysing the existing system the following imminent problem were identified:-

i. The existing system is very slow to the extent that it takes one week or more to complete necessary payroll activities for each month.

ii. It is highly vulnerable to different problems especially in the last six months. This has however required various manipulations to be performed before a payroll processing is completed.

iii. Some modules (menu) of the system are no longer working which always require adhoc solutions

iv. The complete source code of the system is not available which implies that no major changes can be effected on the system as a whole.

v. The system is not fully automated as it requires some computations to be performed before inputting into the computer.

vi. Finally, the system is not documented and as such, it is not easy to properly understand how it works.

In recognition of the above problems it was earlier advised that the payroll processing needs to be commences much more earlier than expected so as not to delay payment of salaries. Based on this advice, it is still observed that the problem get worse every month. This, however, implies that a big problem is immeient within a very short time. It is in response of this that it is concluded that a new computerised payroll system needs to be designed and implemented to take care of the above problems and other suggestion made.

CHAPTER THREE

THE DESIGN OF THE PROPOSED SYSTEM

After the collection and analysis of informations on the existing system, the next phase is the transformation of these informations into logical and physical designs of the new system which is the aim of this chapter. Given this facts, the design of the proposed computerised payroll system in Niger State Agricultural Development Project, Minna is based on the informations gathered on the existing system and suggestions made for improvement. It is designed in a way that basic payroll activities and generation of reports are performed as effective as possible and in accordance with need of the users. In addition, provision is also made for a full automation of the tasks involved in the proposed system.

The chapter however, considers the logical design of the proposed system which contains the design specification of the system. It describes the features of the system in terms of output, input, files and precedures. The later part of the chapter states the physical construction of the system. It contains the program software that would be used to achieve the physical design of the specification.

3.1 OUTPUT SPECIFICATION

Output refers to the results and information that are generated by a system. The output from a computer system are

required primairily to communicate the results of processing to users or other system or more importantly, to provide a permanent copy of these results for consultation. The design process of the output begins by the identification of the output the system must produce. It is as a result of this that in designing output for the purposed system, the needs of the users were fully considered.

Specifically the output of the proposed system is designed to generate five different reports viz: Payslip, Bank list, Deductions list, Staff list, and Departmental Summary. Each of this report is described as follows:-

PAYSLIP:-

This report contains individual details of payroll information for a pay period. It gives information such as the name and bank of the employee, the basic salary, tax, list of allowances and amounts, list of deductions and amounts, the payment history and the net pay of the employee.

BANK LIST

This is an external report and it is used to convey the net pay of the employees to their respective banks. It contains information that would enable the bankers trace and process the account of the employees who keep account with them.

STAFF LIST

This report contains the list of employees in the Niger State Agricultural Development Project whose names are in the payroll at

any particular time. It consists of information on annual salary, salary grade level and step, staff number and so on.

DEPARTMENTAL SUMMARY

This report is made up of summation of each allowances and deductions, total number of staff, total gross pay and netpay of a department.

DEDUCTION LIST

This report contains all the deductions available and the list of employees that each of the deductions affect. It also shows the amount deducted on each of the employees.

3.2 INPUT SPECIFICATION

Having considered the output that need to be gnerated with the proposed system, there is the need to design the input which will bring about the output. Input refers to the mode of entering data into a system. The consideration for input design is very important because it serves as the point of most contact for the users with the system and it is prove to errors. Based on this, the input design should be made to attain the following objectives:-

i. To produce a cost effective method of input.

ii. To achieve the highest level of accuracy.

iii. To ensure that the input is acceptable to and understand by the users.

Basically the above objectives considered in designing the input for the proposed system which is mainly in an interactive mode. This is done through dialoguing with an online system which the computer system prompts for entry. In data entry, conding method, in which conditions, words, ideas, or relationships are expressed by a code, are developed to reduce input task, control errors, and speed the entire process. Therefore, with code, fewer details are necessary in input without loss of information.

Further more the input is designed to reject non-existing codes and in-appropriate data entered. This is again accompanied by a message which gives instructions to the entire users. However, the input data into the system is the payroll data of the employees which contained in a source document. For example, if a new staff is to be entered into the payroll system, the details of that employee are entered into the source document and based on the entries of this document, the user will now key in the data into the system.

The design of the source document is as follows:-

PAYROLL SYSTEM - STARTER'S FORM

DESCRIPTION	DATA
Staff Employee Number	
Surname	
First Name/Others Names	
Marital Status Code	
Date of Appointment	
Incremental Flag	
Department Code	
Salary Level/Step	
Bank Code	
Bank Account Number	
Annual Basic Salary	
Annual Free-pay.	

3.3 FILE DESIGN

Fles design gives a complete decriptions of all the files that are used in a system. This includes the description of the contents and their structures.

The proposed computerised payroll system in Niger State ADP, Minna consists of master files and a transaction file. The Master files are staff dbf, staff 2 dbf and staff 3 dbf. There are as well others which serve as refrence files. These contain static records. They are Dept.dbf, Rank.dbf, Bank.dbf, Appt.dbf, Marital.dbf, Allce.dbf and Deduction.dbf.

However, the description of contents and structure of some of the above database files are as follows:-

STAFF1.DBF:-

This file contains the personal details of all the employees in the Niger State Agricultural Development Project. It consist of 24 fields which are described below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	S_numb	Staff number	Character	8
2.	Surname	Surename	Character	18
3.	F_name	First name	Character	18
4.	O_names	Other names	Character	18
5.	Initial	Initial	Character	6
6.	Ms_code	Marital Status Code	Character	1
7.	App_code	Appointment Code	Character	1
8.	T_appt	Appointment Type	Character	1
9.	DD_appt	Date of First Appt	Data	8
10.	Per_sal	Percentage Salary	Numeric	3/0
11.	Inc_Flag	Incremental Flag (Y or N)	Character	1
12.	Dept_code	Department Code	Character	3
13.	Cen_code	Centre Code	Character	1
14.	Rank_code	Rank Code	Character	2
15.	Sal_L	Salary Grade Level	Character	2
16.	Sal_s	Salary Step	Character	2
17.	Bank_code	Bank Code	Character	2
18.	Bank_no	Bank Number	Character	12
19.	A_Sal	Annual Salary	Numeric	6/2
20.	Acc_F pay	Accumulated Free Pay	Numeric	6/2
21.	A_f pay	Annual Free Pay	Numeric	6/2
22.	Acc_tax	Accumulated Tax	Numeric	5/2
23.	Tax_gr	Taxable Gross	Numeric	7/2
24.	Tax_pay	Taxable Pay	Numeric	5/2

STAFF2.DBF:-

It is a file which keeps informations about the details of the allowances of all the NSADP employees. Its structure is stated below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	S_numb	Staff number	Character	8
2.	A_code 1	Code for allw. 1	Numeric	3/0
3.	A_sign 1	Sign for allw. 1	Character	1
4.	A_amt 1	Amount for allw.1	Numeric	6/2
5.	A_cnt 1	Counter for allw.1	Numeric	3/0
6.	A_dur 1	Duration for allw.1	Numeric	3/0
7.	A_code 2	Code for allw. 2	Numeric	3/0
8.	A_sign 2	Sign for allw. 2	Character	1
9.	A_amt 2	Amount for allw.2	Numeric	6/2
10.	A_cnt 2	Counter for allw.2	Numeric	3/0
11.	A_dur 2	Duration for allw.2	Numeric	3/0
12.	A_code 3	Code for allw. 3	Numeric	3/0
13.	A_sign 3	Sign for allw. 3	Character	1
14.	A_amt 3	Amount for allw.3	Numeric	6/2
15.	A_cnt 3	Counter for allw.3	Numeric	3/0
16.	A_dur 3	Duration for allw.3	Numeric	3/0
17.	A_code 4	Code for allw. 4	Numeric	3/0
18.	A_sign 4	Sign for allw. 4	Character	1
19.	A_amt 4	Amount for allw.4	Numeric	6/2
20.	A_cnt 4	Counter for allw.4	Numeric	3/0
21.	A_dur 4	Duration for allw.4	Numeric	3/0
22.	A_code 5	Code for allw. 5	Numeric	3/0
23.	A_sign 5	Sign for allw. 5	Character	1
24.	A_amt 5	Amount for allw.5	Numeric	6/2
25.	A_cnt 5	Counter for allw.5	Numeric	3/0
26.	A_dur 5	Duration for allw.5	Numeric	3/0
27.	A_code 6	Code for allw. 6	Numeric	3/0
28.	A_sign 6	Sign for allw. 6	Character	1
29.	A_amt 6	Amount for allw.6	Numeric	6/2
30.	A_cnt 6	Counter for allw.6	Numeric	3/0
31.	A_dur 6	Duration for allw.6	Numeric	3/0

STAFF3.DBF

This file keeps information about the details of deductions of all the NSADP's employees. Its structure is stated below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	S_numb	Staff number	Character	8
2.	A_code 1	Code for allw. 1	Numeric	3/0
3.	A_sign 1	Sign for allw. 1	Character	1
4.	A_amt 1	Amount for allw.1	Numeric	6/2
5.	A_cnt 1	Counter for allw.1	Numeric	3/0
6.	A_dur 1	Duration for allw.1	Numeric	3/0
7.	A_code 2	Code for allw. 2	Numeric	3/0
8.	A_sign 2	Sign for allw. 2	Character	1
9.	A_amt 2	Amount for allw.2	Numeric	6/2
10.	A_cnt 2	Counter for allw.2	Numeric	3/0
11.	A_dur 2	Duration for allw.2	Numeric	3/0
12.	A_code 3	Code for allw. 3	Numeric	3/0
13.	A_sign 3	Sign for allw. 3	Character	1
14.	A_amt 3	Amount for allw.3	Numeric	6/2
15.	A_cnt 3	Counter for allw.3	Numeric	3/0
16.	A_dur 3	Duration for allw.3	Numeric	3/0
17.	A_code 4	Code for allw. 4	Numeric	3/0
18.	A_sign 4	Sign for allw. 4	Character	1
19.	A_amt 4	Amount for allw.4	Numeric	6/2
20.	A_cnt 4	Counter for allw.4	Numeric	3/0
21.	A_dur 4	Duration for allw.4	Numeric	3/0
22.	A_code 5	Code for allw. 5	Numeric	3/0
23.	A_sign 5	Sign for allw. 5	Character	1
24.	A_amt 5	Amount for allw.5	Numeric	6/2
25.	A_cnt 5	Counter for allw.5	Numeric	3/0
26.	A_dur 5	Duration for allw.5	Numeric	3/0
27.	A_code 6	Code for allw. 6	Numeric	3/0
28.	A_sign 6	Sign for allw. 6	Character	1
29.	A_amt 6	Amount for allw.6	Numeric	6/2
30.	A_cnt 6	Counter for allw.6	Numeric	3/0
31.	A_dur 6	Duration for allw.6	Numeric	3/0

MONTHLY.DBF:-

The monthly. dbf is an output file which keeps the results of a payroll process. It contains informations about all the NSADP's employees' one pay period. The structure of this file is as shown below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	S_numb	Staff number	Character	8
2.	Surname	Surename	Character	18
3.	F_name	First name	Character	18
4.	O_names	Other names	Character	18
5.	Initial	Initial	Character	6
6.	Per_sal	Percentage Salary	Numeric	3/0
7.	Dept_code	Department Code	Character	2
8.	Cen_code	Code Centre	Character	1
9.	Rank_code	Rank Code	Character	2
10.	Sal_l	Salary Grade Level	Character	2
11.	Sal_s	Salary Step	Character	2
12.	Bank_code	Bank Code	Character	2
13.	Bank_no	Bank Number	Character	12
14.	A_Sal	Annual Salary	Numeric	6/2
15.	Basic_sal	Monthly Basic Salary	Numeric	6/2
16.	A_f pay	Annual free pay	Numeric	6/2
17.	Acc_f pay	Accumulated Freepay	Numeric	7/2
18.	Tax	Monthly Tax	Numeric	6/2
19.	Acc_tax	Accumulated Tax	Numeric	7/2
20.	Tax_gr	Taxable Gross	Numeric	7/2
21.	Tax_pay	Taxable Pay	Numeric	7/2
22.	Tax_ref	Tax Refund	Numeric	5/2

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
23.	Tax_Jan	Tax_paid since Jan.	Numeric	7/2
24.	Gr_pay	Gross pay	Numeric	7/2
25.	Tot_ded	Total Deduction	Numeric	7/2
26.	Net_pay	Net pay	Numeric	7/2
27.	A_code 1	Code for Allw. 1	Numeric	3/0
28.	A_sign 1	Sign for Allw. 1	Character	1
29.	A_amt 1	Amount for Allw. 1	Numeric	6/2
30.	A_code 2	Code for Allw. 2	Numeric	3/0
31.	A-sign 2	Sign for Allw. 2	Character	1
32.	A_amt 2	Amount for Allw.2	Numeric	6/2
33.	A_code 3	Code for Allw. 3	Numeric	3/0
34.	A-sign 3	Sign for Allw. 3	Character	1
35.	A_amt 3	Amount for Allw.3	Numeric	6/2
36.	A_code 4	Code for Allw. 4	Numeric	3/0
37.	A-sign 4	Sign for Allw. 4	Character	1
38.	A_amt 4	Amount for Allw.4	Numeirc	6/2
39.	A_code 5	Code for Allw. 5	Numeric	3/0
40.	A-sign 5	Sign for Allw. 5	Character	1
41.	A_amt 5	Amount for Allw.5	Numeric	6/2
42.	A_code 6	Code for Allw. 6	Numeric	3/0
43.	A-sign 6	Sign for Allw. 6	Character	1
44.	A_amt 6	Amount for Allw.6	Numeric	6/2
45.	A_code 7	Code for Allw. 7	Numeric	3/0
46.	A-sign 7	Sign for Allw. 7	Character	1
47.	A_amt 7	Amount for Allw.7	Numeric	6/2
48.	A_code 8	Code for Allw. 8	Numeric	3/0
49.	A-sign 8	Sign for Allw. 8	Character	1
50.	A_amt 8	Amount for Allw.8	Numeric	6/2
51.	A_code 9	Code for Allw. 9	Numeric	3/0
52.	A-sign 9	Sign for Allw. 9	Character	1
53.	A_amt 9	Amount for Allw.9	Numeric	6/2
54.	A_code 10	Code for Allw. 10	Numeric	3/0
55.	A-sign 10	Sign for Allw. 10	Character	1
56.	A_amt 10	Amount for Allw.10	Numeric	6/2

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
57.	D_code 1	Code for Deduc. 1	Numeric	3/0
58.	D_sign 1	Sing for Deduc. 1	Character	1
59.	D_amt 1	Amount for Deduc. 1	Numeric	6/2
60.	D_code 2	Code for Deduc. 2	Numeric	3/0
61.	D_sign 2	Sing for Deduc. 2	Character	1
62.	D_amt 2	Amount for Deduc. 2	Numeric	6/2
63.	D_code 3	Code for Deduc. 3	Numeric	3/0
64.	D_sign 3	Sing for Deduc. 3	Character	1
65.	D_amt 3	Amount for Deduc. 3	Numeric	6/2
66.	D_code 4	Code for Deduc. 4	Numeric	3/0
67.	D_sign 4	Sing for Deduc. 4	Character	1
68.	D_amt 4	Amount for Deduc. 4	Numeric	6/2
69.	D_code 5	Code for Deduc. 5	Numeric	3/0
70.	D_sign 5	Sing for Deduc. 5	Character	1
71.	D_amt 5	Amount for Deduc. 5	Numeric	6/2
72.	D_code 6	Code for Deduc. 6	Numeric	3/0
73.	D_sign 6	Sing for Deduc. 6	Character	1
74.	D_amt 6	Amount for Deduc. 6	Numeric	6/2
75.	D_code 7	Code for Deduc. 7	Numeric	3/0
76.	D_sign 7	Sing for Deduc. 7	Character	1
77.	D_amt 7	Amount for Deduc. 7	Numeric	6/2
78.	D_code 8	Code for Deduc. 8	Numeric	3/0
79.	D_sign 8	Sing for Deduc. 8	Character	1
80.	D_amt 8	Amount for Deduc. 8	Numeric	6/2
81.	D_code 9	Code for Deduc. 9	Numeric	3/0
82.	D_sign 9	Sing for Deduc. 9	Character	1
83.	D_amt 9	Amount for Deduc. 9	Numeric	6/2
84.	D_code 10	Code for Deduc. 10	Numeric	3/0
85.	D_sign 10	Sing for Deduc. 10	Character	1
86.	D_amt 10	Amount for Deduc. 10	Numeric	6/2

DEPT. DBF

This database file contains the list of departments available in the Niger State A.D.P and their respective codes. Its structure is given below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	Dept_code	Department Code	Character	3
2.	Dept_name	Department Name	Character	34

RANK DBF:-

It is a file that contains different ranks available in the Niger State A. D. P and their respecting codes. Its structure is shown below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	Rank_code	Rank Code	Character	2
2.	Rank_Desc	Rand Description	Character	17

ALLCE. DBF:-

The allwce. dbf is a file that contains all the allowances that the N S A D P employees are entitle to and their respective codes. The format is as shown below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	A_code	Allowance Code	Numeric	3/0
2.	A_Desc	Allw. Description	Character	17

DEDUCT. DBF

It is a file that contains the list of deduction charge on the N S A D P employees and their respective codes. Its structure is as shown below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	D_code	Deduction Code	Numeric	3/0
2.	D_Desc	Deduc. Description	Character	17

APPT. DBF

This is a file that contains the list of the type of appointments being offer to any employee by the N S A D P and their respective codes. The format of this file is given below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	Appt_code	Appointment Code	Character	1
2.	Appt_Desc	Appt. Description	Character	17

MARITAL. DBF

It is a file which contains the available types of marital status and their respective code. Its structure is as shown below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	Ms_code	Marital Status Code	Character	1
2.	Ms_Desc	Marital Status Description	Character	10

BANK DBF

This file contains the list of the Banks which N S A D P employees use for their salary. The structure is as shown below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	Bank_code	Bank Code	Character	2
2.	Bank_Desc	Bank Description	Character	35

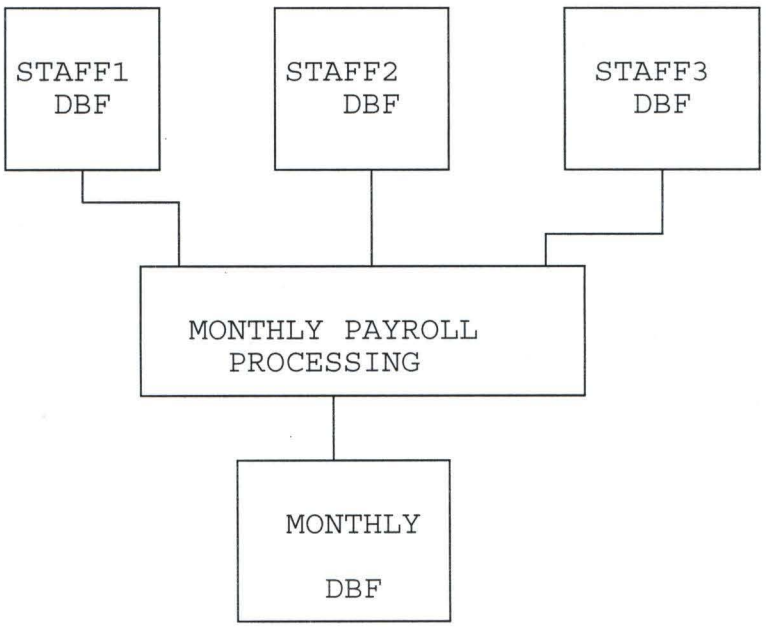
CURRENT DBF

This file is used to control some operations within the system as well as keeping its state in any point in time. It is made up of two files which are described below:-

S/NO	FIELD NAME	FIELD DESCRIPTION	FIELD TYPE	FIELD WIDTH
1.	C_date	Current Date	Date	8
2.	C_process	Current Process	Character	1

3.4 DESCRIPTION OF SYSTEM PROCEDURE

As stated above the input data is the payroll records of all the N S A D P employees which are distributed into the three master files available in this system. This file serves as input for the monthly payroll processing performed by the system while the output is the transaction file (Monthly.DBF). The system procedure for this processing is describe below:-



The monthly payroll processing is the actual computation of the employees' netpay and the update of necessary information that would be required. Each of the record is taken from the master file for processing after which it is transfered to monthly. dbf. Since the monthly. dbf suppose to contain the details of all the employees for one pay period, the file is always made empty at the begining of each subsequent processing.

However, the procedures of the proposed system will be completely menu-based where an interactive user could accomplish a task by selecting menu.

3.5 THE PHYSICAL DESIGN OF THE SYSTEM

This section has to do with program specification for files, input, output and processing into computer software. It deals with the physical construction of the logical design described above. The designing of the software is important to ensure that the actual progress produced perform all task as intended and to allow for future modifications to be performed in an efficient manner and with a minimum destruction to the design of the system.

CHAPTER FOUR

SYSTEM IMPLEMENTATION

After the physical system has been desigued as contained in the last chapter, the next stage is to turn the design into a working system and then to monitor the operation of the system to ensure that it is working efficiently and effectively. Therefore system implementation is the stage of system development when the conceptional requirement of the new system and the overall objectives are to be transformed into physical reality. This stage is very important because it is the most crucial stage in achieving a successful new system and in giving the users confidence that the new system will work and be effective.

However, for proper analysis of the tasks of implemntation, this chapter begins with the description of the proposed system and into hardware requirements. This is followed by system testing before it goes further to dicuss the mode of system conversion. Finally, this chapter discusses the procedures that will be required in carrying out amendment on the system.

4.1 THE PROPOSED SYSTEM

The new system is made up of a main menu which consist of seven options viz:- Staff File Maintenance, Variation Entry Monthly Run, Report Generation, Files Update, Utilities and Quit. At this main Menu the system will prompt the user to enter the first letter of any of the available options to pick choice. the screen format of this menu is shown in figure 1 overleaf.

However, each of the seven menu are discussed as follows:-

1. **STAFF FILE MAINTENANCE:-**

This option allows the users to update the master files containing the payroll records of all the employees. At this menu another level of options appears on the screen as shown in figure 2 in which the User is expected to pick a choice. Each of the submenu is discussed below.

ADDING STAFF RECORD:-

This submenu will afford the user an opportunity to add new payroll data into the system. It is made up of five different screens which comes one after the other. The five screens are being represented by figure 3 through 5 below.

VIEWING STAFF RECORD

This allows the user to see an existing record and it is done by first entering the staff number of the employee. The system then goes further to show the detail at the employee which are displayed in five different screens similar to figure 3 through 5 below.

MODIFYING STAFF RECORD:-

This procedure allows for change to be performed on the personal details of an employee. In order to locate the record to modify the staff number of the employee is entered and the personal details of the employee are displayed on one screen for changes. The screen format for this is also similar to figure 3 below.

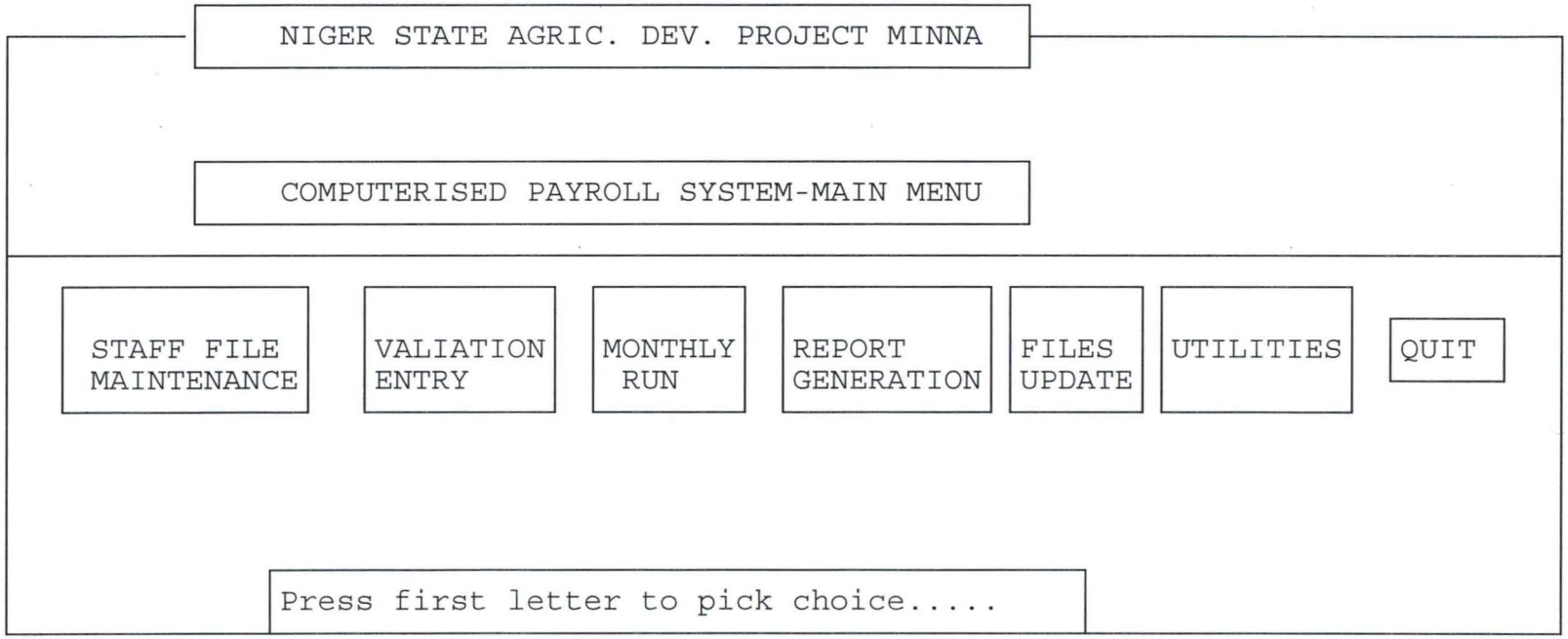


FIGURE 1

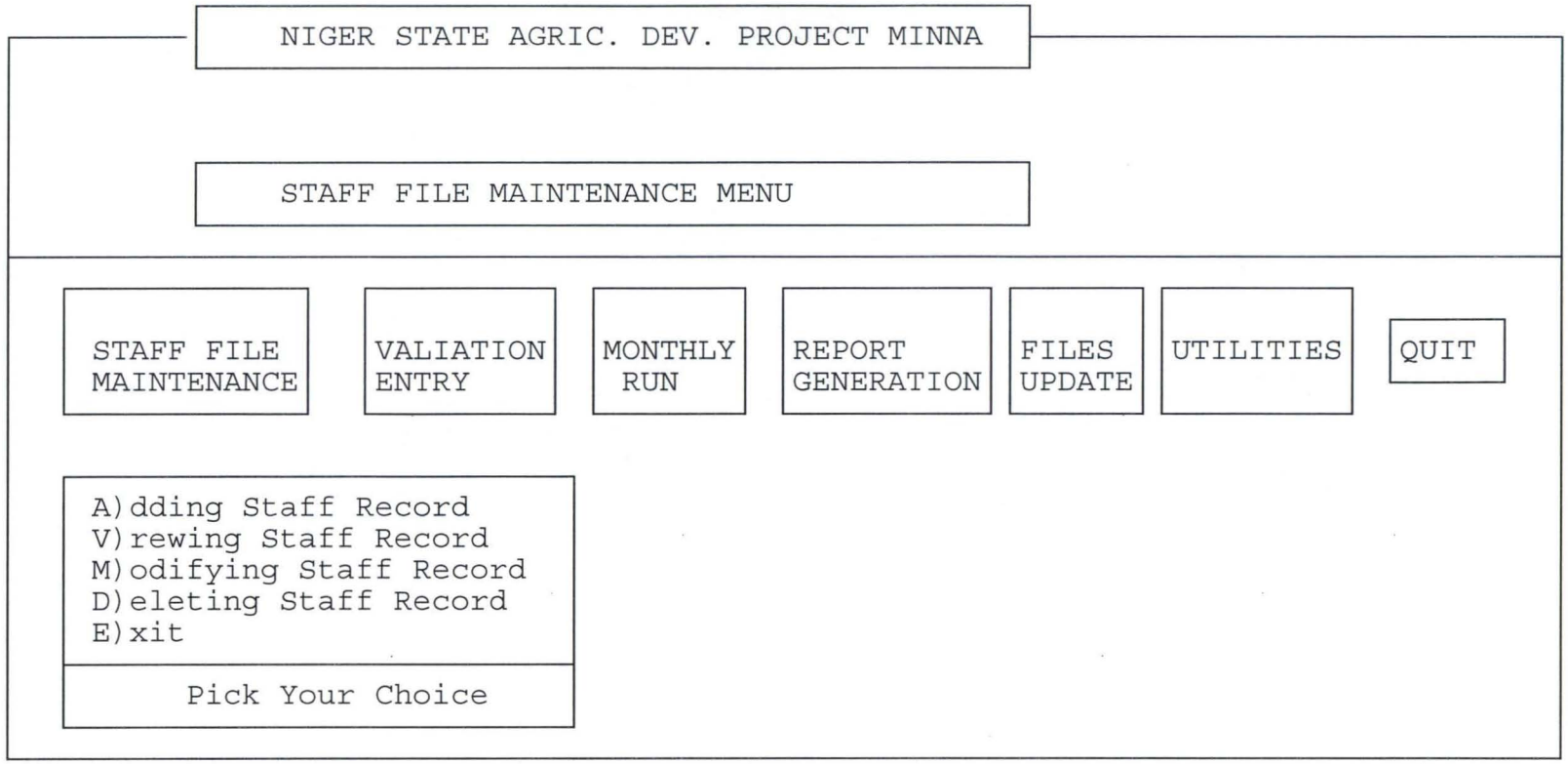

NIGER STATE AGRIC. DEV. PROJECT MINNA
STAFF FILE MAINTENANCE MENU
STAFF FILE MAINTENANCE
VALIATION ENTRY
MONTHLY RUN
REPORT GENERATION
FILES UPDATE
UTILITIES
QUIT
A)dding Staff Record
V)rewing Staff Record
M)odifying Staff Record
D)eleting Staff Record
E)xit
Pick Your Choice

FIGURE 2

NIGER STATE AGRIC. DEV. PROJECT MINNA

COMPUTERISED PAYROLL SYSTEM

DATA ENTRY FORM

STAFF NUMBER: 203Y0026 DATE 01/08/97

SURNAME	FIRST NAME	OTHER NAME	INITIAL
YAHAYA	ABDULKADIR	HAUWA	A H.

MARITAL ST. CODE:3 DEPARTMENT CODE:06 RANK CODE: 10

DATE OF FIRST APPOINTMENT	APPOINTMENT CODE	TYPE OF APPOINTMENT
01/03/83	7	7

GRADE LEVEL	SALARY STEP	ANNUAL SALARY
12	09	₦30,000.00

BANK CODE: 10 BANK NUMBER: 05123461 ANNUAL FREE PAY 20,000.00

Press any key to go to next screen.....

FIGURE 3

NIGER STATE AGRIC. DEV. PROJECT MINNA

COMPUTERISED PAYROLL SYSTEM

DATA ENTRY FORM

DETAILS OF ALLOWANCES STAFF NUMBER: 203Y0026

ALLOWANCE	CODE	DESCRIPTION	SIGN	AMOUNT	DURATION
ALLOWANCE	31	RENT SUBSIDY	+	545.00	0
ALLOWANCE	12	TRANS. ALLW	+	406.00	0
ALLOWANCE	30	UTILITY	+	60.00	0
ALLOWANCE	25	ENTER. ALLW	+	150.00	1
	20	RESPONS. ALLW	+		
	28	ACTING ALLW	+		
	30	GUARD/STEWARD ALLW	+		

PRESS ANY KEY TO GO TO NEXT SCREEN

FIGURE 4

NIGER STATE AGRIC. DEV. PROJECT MINNA

COMPUTERISED PAYROLL SYSTEM

DATA ENTRY FORM

DETAILS OF DEDUCTION STAFF NUMBER

DEDUCTION	CODE	DESCRIPTION	SIGN	AMOUNT	DURATION
DEDUCTION 1	60	NAT HOUS SCHEME	-	40.00	0
DEDUCTION 2	62	UNION DEV	-	23.00	0
DEDUCTION 3	65	MOTOR CYCLE LOAN	-	250.00	30
DEDUCTION 4	70	BICYCLE LOAN	-		
DEDUCTION 5	75	REFURE. LOAN	-	100.00	6
DEDUCTION 6	69	FURN. LOAN	-		
DEDUCTION 7	70	SALARY ADVANCE	-	50.00	5

PRESS ANY KEY TO GO TO NEXT SCREEN

FIGURE 5

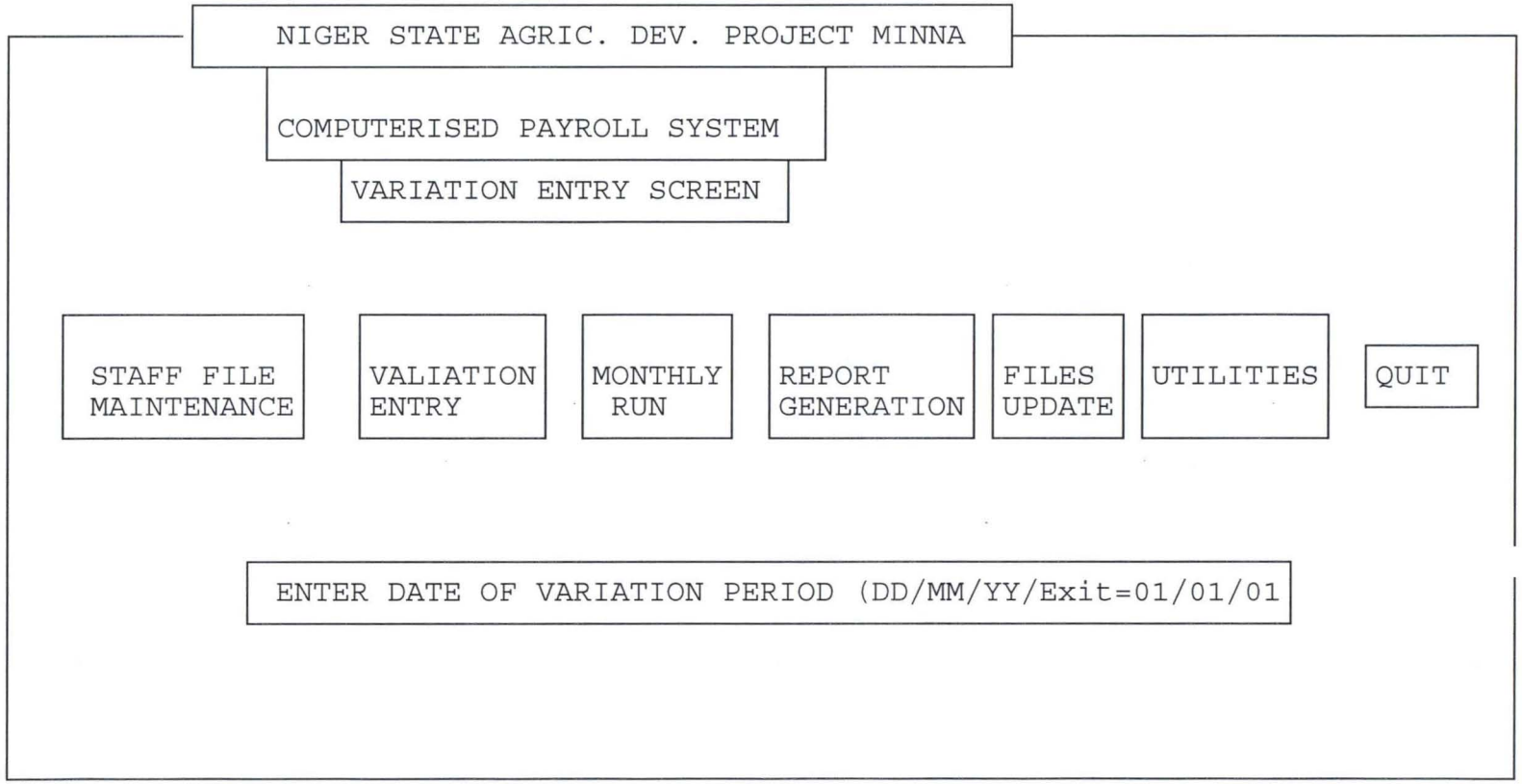

NIGER STATE AGRIC. DEV. PROJECT MINNA
COMPUTERISED PAYROLL SYSTEM
VARIATION ENTRY SCREEN
STAFF FILE MAINTENANCE
VALIATION ENTRY
MONTHLY RUN
REPORT GENERATION
FILES UPDATE
UTILITIES
QUIT
ENTER DATE OF VARIATION PERIOD (DD/MM/YY/Exit=01/01/01

FIGURE 6

DELETING STAFF RECORD:-

This is similar to modifying staff record abOVe except that it is used to get rid of a particular record.

EXIT:-

This takes the user back to the main menu

2. **VARIATION ENTRY:-**

This option is used to carry out changes in the details of allowances and deduction of the employees. Once this option is selected the system will prompt the user to enter the date of variation period as shown in figure 6 above. The user will be expected to enter the last date of the month for which variation entry is to be carried out. However the system expected the staff number of each of the employee's records to perform vatriation upon. Then the system will display four different screens which are similar to figures 4 and 5 above.

3. **MONTHLY RUN:-**

This procedure carries out the actual computations and the necessary update involve in payroll activities. On the selection of this menu, the user will be expected to enter the last day of the month for which payroll is to be processed as represented by figure 7 below. The date entered are then validated before the processing begins.

4. **REPORT GENERATION**

This option offers an opportunity to the users to print out any of the reports that is to be generated by this system. If this option is selected, the next level of options appears on the screen as shown in figure 8 below. The user is then expected to pick a choice which represents the type of report to be generated at any particular time.

5. **FILES UPDATE:-**

The files update menu is used to update all the reference files used in the system. Once this menu is selected, another set of options are displayed on the screen as represented by figure 9 below from which the usar selects the files to be updated. As the selection is made, another choice appears on the screen as shown in figure 10 below from which the user picks the type of update to be made.

6. **UTILITIES:-**

The utilities menu comprises some additional capabilities of the system which are not shown in the main menu. These other capabilities are Backup and Form printing as indicated in figure 11 below.

BACKUP:-

This submenu enable the user to copy the necessary database files into floppy diskettes. It is done mainly to aid the users in case there is a problem with the data on the hard disk.

FORM PRINTING:-

In the input specification discussed above it is desighed that his system should use a source document in which area the information on the new employees detail are expected to be filled before eventually entered into the payroll system.

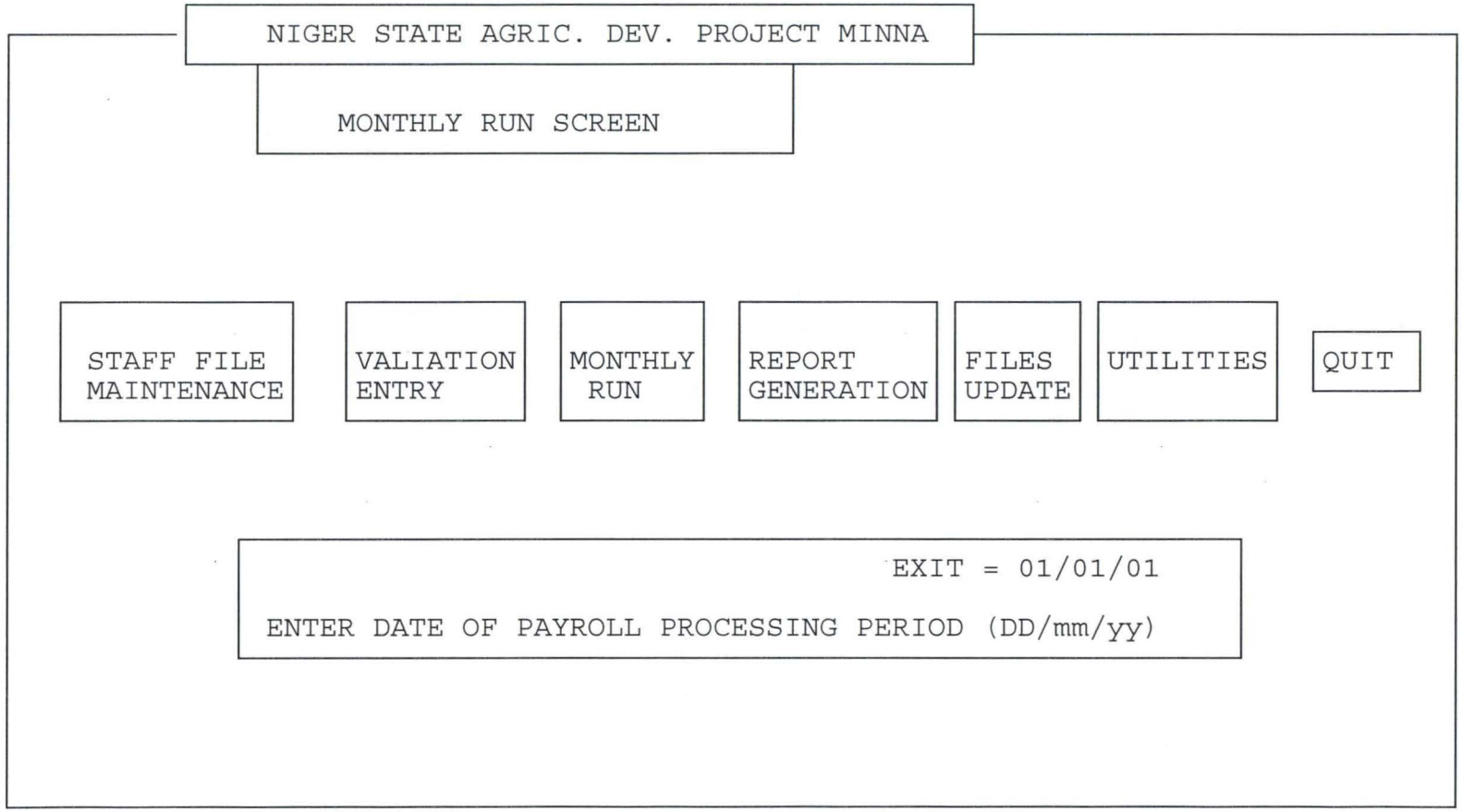

NIGER STATE AGRIC. DEV. PROJECT MINNA
MONTHLY RUN SCREEN
STAFF FILE MAINTENANCE
VALIATION ENTRY
MONTHLY RUN
REPORT GENERATION
FILES UPDATE
UTILITIES
QUIT
EXIT = 01/01/01
ENTER DATE OF PAYROLL PROCESSING PERIOD (DD/mm/yy)

FIGURE 7

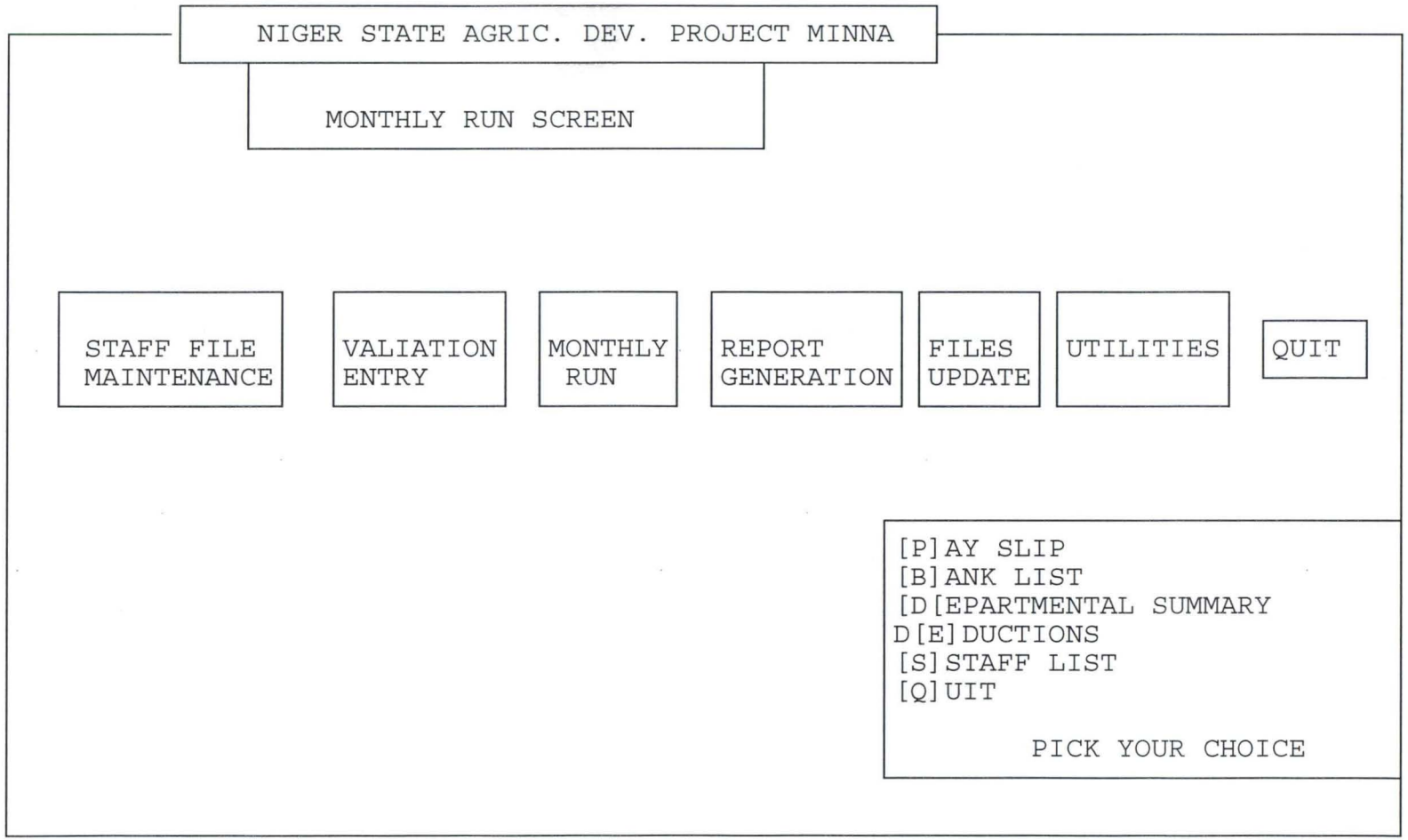

NIGER STATE AGRIC. DEV. PROJECT MINNA
MONTHLY RUN SCREEN
STAFF FILE MAINTENANCE
VALIATION ENTRY
MONTHLY RUN
REPORT GENERATION
FILES UPDATE
UTILITIES
QUIT
[P]AY SLIP
[B]ANK LIST
[D[EPARTMENTAL SUMMARY
D[E]DUCTIONS
[S]STAFF LIST
[Q]UIT
PICK YOUR CHOICE

FIGURE 8

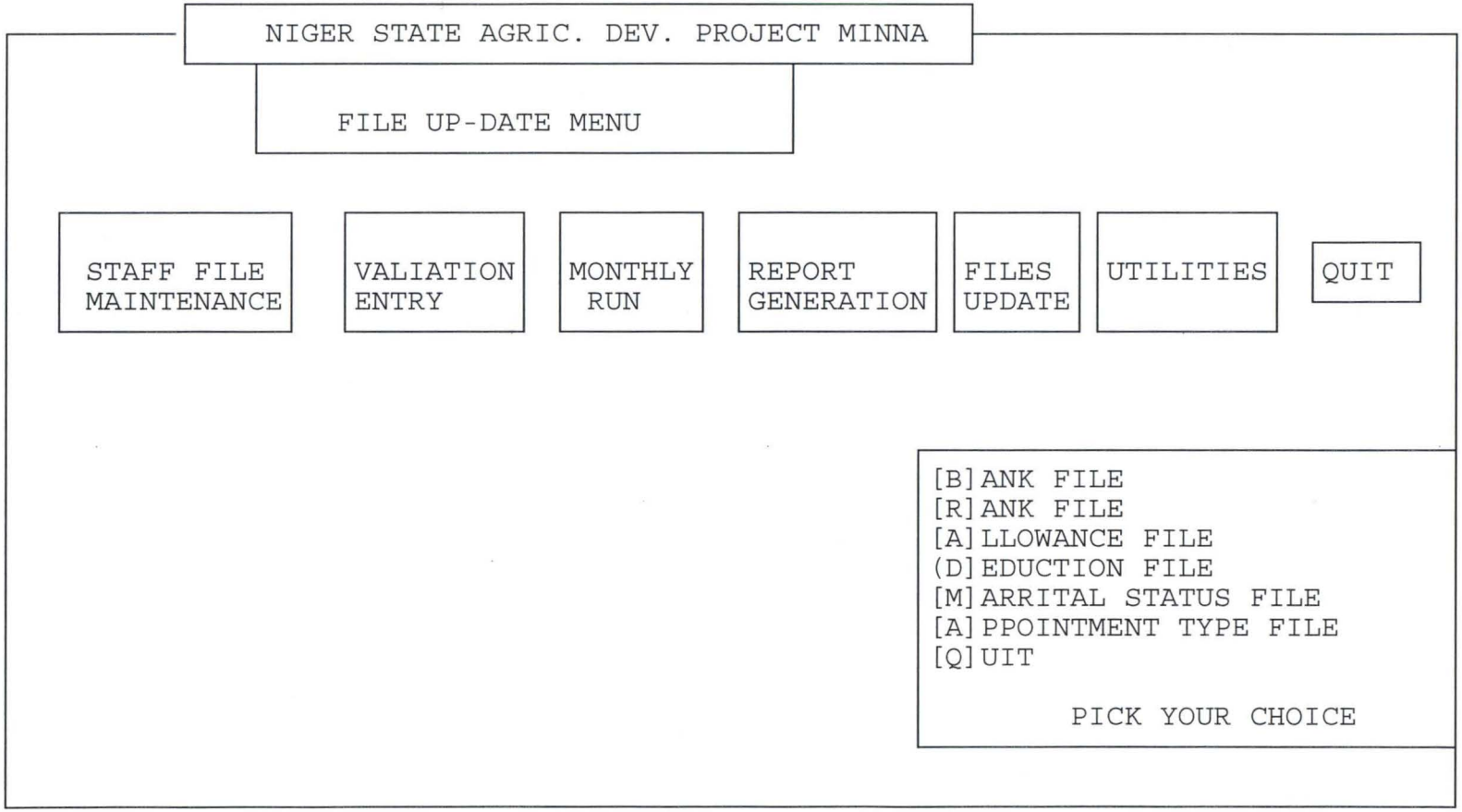

NIGER STATE AGRIC. DEV. PROJECT MINNA
FILE UP-DATE MENU
STAFF FILE MAINTENANCE
VALIATION ENTRY
MONTHLY RUN
REPORT GENERATION
FILES UPDATE
UTILITIES
QUIT
[B]ANK FILE
[R]ANK FILE
[A]LLOWANCE FILE
(D]EDUCTION FILE
[M]ARRITAL STATUS FILE
[A]PPOINTMENT TYPE FILE
[Q]UIT
PICK YOUR CHOICE

FIGURE 9

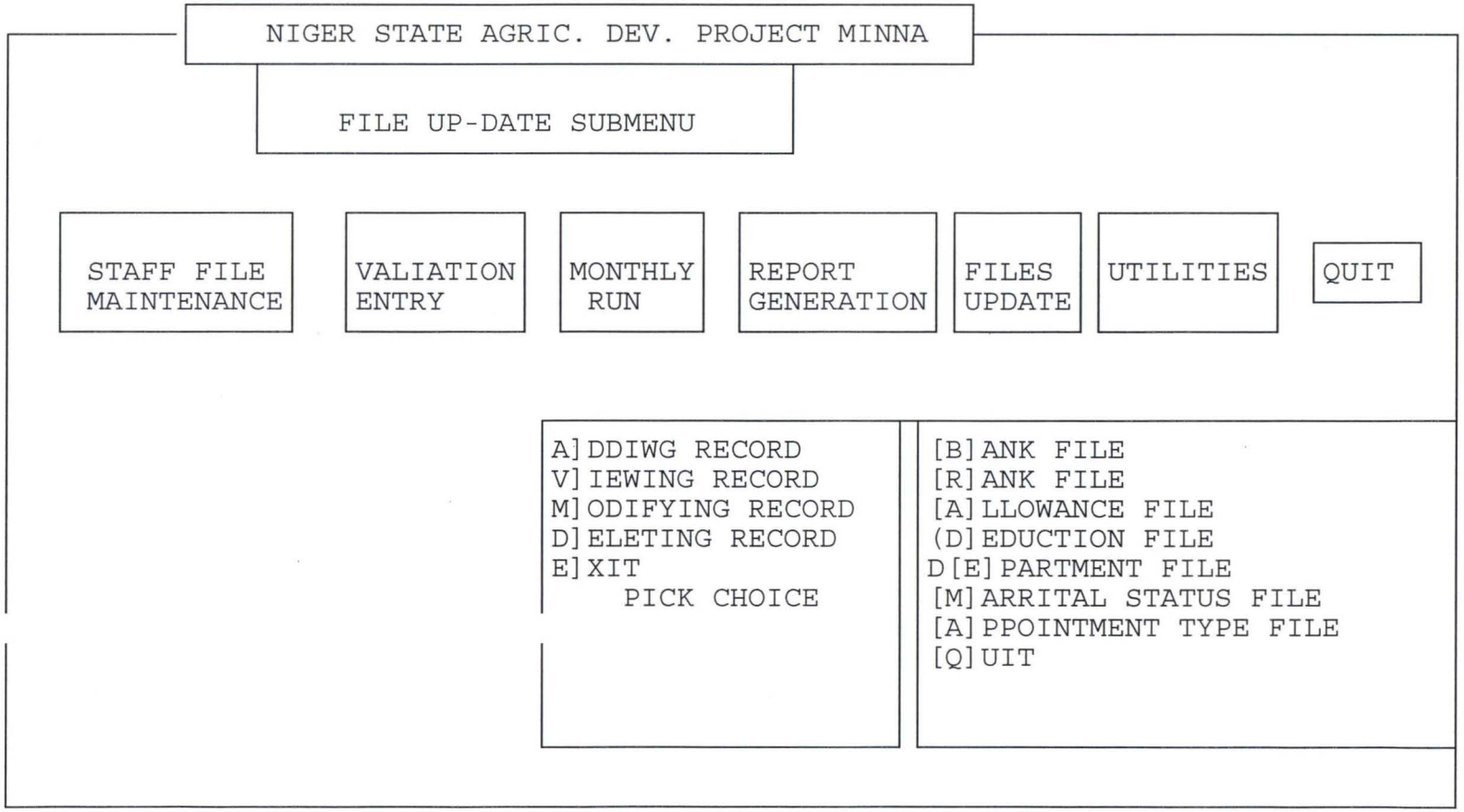

NIGER STATE AGRIC. DEV. PROJECT MINNA
FILE UP-DATE SUBMENU
STAFF FILE MAINTENANCE
VALIATION ENTRY
MONTHLY RUN
REPORT GENERATION
FILES UPDATE
UTILITIES
QUIT
A]DDIWG RECORD
V]IEWING RECORD
M]ODIFYING RECORD
D]ELETING RECORD
E]XIT
PICK CHOICE
[B]ANK FILE
[R]ANK FILE
[A]LLOWANCE FILE
(D]EDUCTION FILE
D[E]PARTMENT FILE
[M]ARRITAL STATUS FILE
[A]PPOINTMENT TYPE FILE
[Q]UIT

FIGURE 10

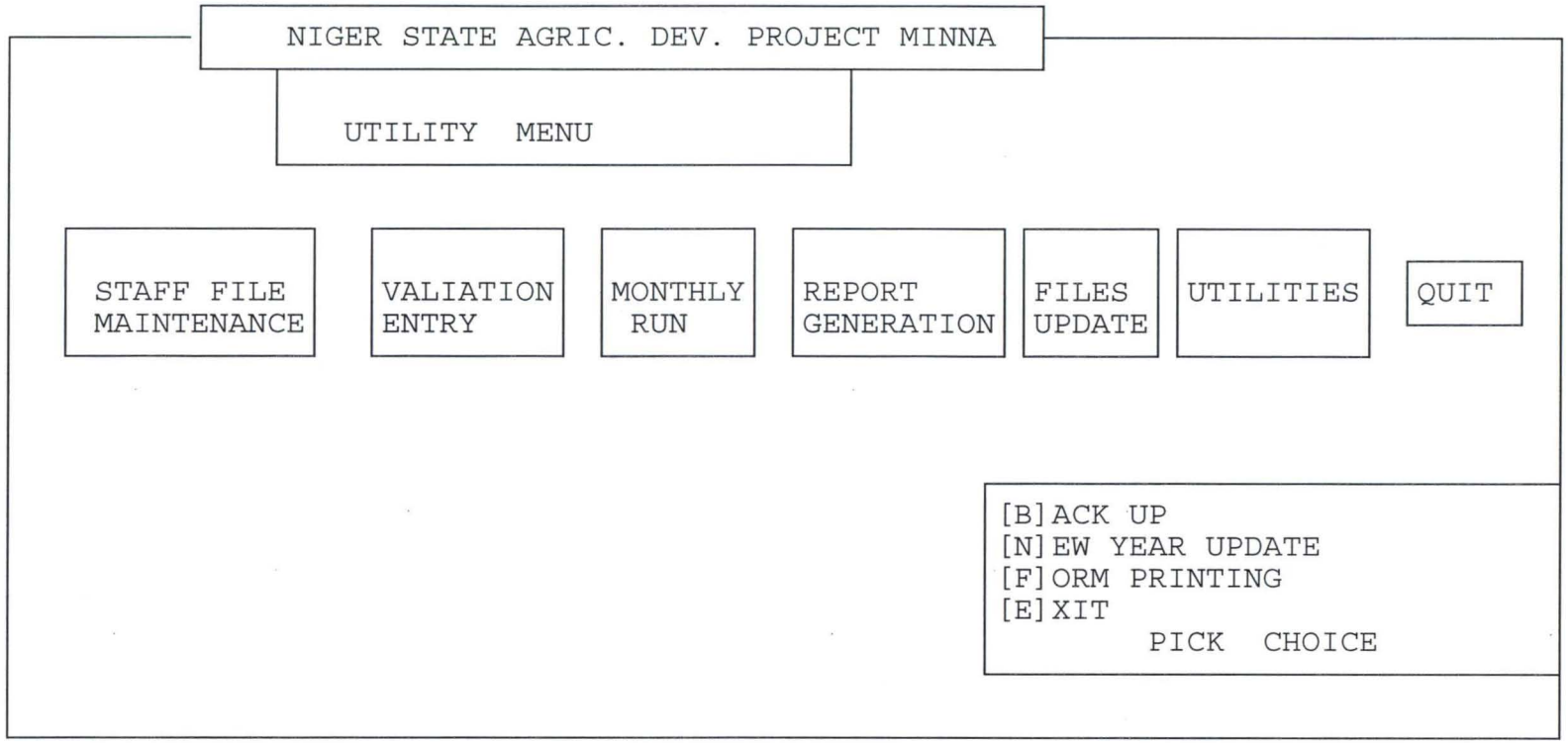


FIGURE 11

7. QUIT

This menu serves as the only way of exiting from the system. Once it is selected, it takes the user back to the operating system prompt.

4.2 SYSTEM REQUIREMENT

The system requirement has to do with the computer configuration needed for the new system. A computer configuration is a collection of hardware which forms a complete computer system. The selection of the computer configuration is done to suit both the current and the foreseeable future needs of the organisation with respect to the volume and types of data to be processed.

However, with this newly developed system, a computer with higher speed and lorger storage space is required. This is expected to take care of the future need of the organisation. It is also necessary for this organisation to procure an uninterrupted power supply (UPS), a facility to ensure constant power supply to the computer and its environment. This is needed mainly to avoid interruption especially when the computer is performing the monthly processing.

In summary, a computer with a hard disk of a minimum of 100mB and having two floppy drive units is recommended. The two floppy units should be the "male-female twin" in which one should be for 3.5 inches and the other providing for 5.25 inches floppys. These two floppy units will provide for the transfer of the software from diskettes into the hard disk as well as making back up on floppy diskettes. The computer should have a speed of about 40MHZ to aid

fast processing of records and UPS which can store power for about 45 minutes in case of power failure would also be required.

4.3 SYSTEM TESTING

System testing is a key stage in system implementation. It invoves the use of test data on the new system in order to ensure that system works accurately and efficiently before live operation commences. At this stage, the logical design and physical design are thoroughly examined to ensure its workability. Therefore, the system test in implementation seves as a confirmation that all is correct and an opportunity to show the users that the system works as required.

However, the new system has been tested using some test data on all the modules of the system. At the end of the test, it was confirmed that it worked efficiently. Infact, result of this testing is showed in the varous reports displayed in the last chapter.

4.4 SYSTEM CONVERSION

Having confirmed above that the new system is working efficiently, there is need to carry out filed set up, file conversion and change over. These are done to aid in the transformation of the existing system to the newly developed one.

In this payroll system, the records of some database files like dept.dbf and others that were not used before entered newly. other database files were converted to the format required by the new system. All these were done using dbase III plus appending mode.

However, system conversion is not completed until the actual change over from the existing system to the new system takes place. Changeover is the stage of moving over from the old system to the newly developed one. The changeover may be achieved in a number of ways viz: direct changeover, parallel running pilot running and stage changeover.

Given the above four methods of changeover parallel running is chosen for this system. This implies processing the current data by both the old and new systems. Its main attraction is that the old system is kept alive and operational until the new system has been proved for at least one system cycle using file data in the real operational environment of place, people, equipment and time. In addition it gives an opportunity of comparing the results of the new system with the existing one before acceptance by the users thereby promoting user's confidence.

4.5 POST IMPLEMENTATION REVIEW

After the system is implemented and conversion is completed, provision needs to be made for a review of the system. This has to do with the maintenance of the system against environmental changes which may affect either the computer or other parts of computer-based system. This may lead to the improvement of ystem functions and the correction of faults which arise during the operation of a system.

Specifically, the objectives of the post implementation review is to:-

1. Determine whether the system goals and objectives have been achieved.
2. Determine whether known or unexpected limitations of the system need attention.
3. Determine whether personnel procedures operating activities and order control have been improved.
4. Determing whether use service requirements have been met, while simultausously reducing errors and costs.

However the amendment procedure agreed upon with the use of this system is directly through the users. The users are expected to identify any problem areas or external requirement of the system. Based on this, the system will further be designed to meet the requirement.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

The contiuned substitution of computer based systems for manual procedures has, in modern days, become a wolrd-wide affairs. This is due to its relevance in virtually all aspects of human endeavour. This interest is intensified by the capability of computers in performing a given set of procedures with all the necessary accuracy. It is not subjected to commiting error and its ability to accomplish any task with high speed and within a reasonable time makes it applicable in the present time.

However, it would be accepted that a computer-base procedures needs to be designed in a way to achieve the benefit of computer usage in terms of speed, full automation of procedures, aviod constant problems, ensure data security and so on. It is in recongnition of this fact that a newly designed computerised payroll system is recommended for this organisition.

Specifically, Niger State Agricultural Development project, MInna derive the following benefits from this newly designed system.

(i) Creation of speedy payroll processing and generation of necessary reports

(ii) Enhance the efficient operation of the payroll section of this organisation.

(iii) Avoidance of constant problems as being experienced with the existing system.

(iv) Maintenance of data security

(v) Allow for carrying out major changes since the new system is fully decummented.

(vi) Provision of automated procedures especially in terms of computations that is required during data entry.

(vii) Finally, some procedures were introduced within the new system which reduces the task of the users as well as making provision for the facility required by the system.

Given the above benefits of the newly designed system, it is highly recommended that the hardware requirement for this new system as stated in the last chapter should be provided immediately. This will allow for immediate commencement of system conversion as from month of August, 1997. In addition, the intending users of this new system needs to be trained for about three weeks on the usage of the system.

Conclusively, the pursance of the installation of this newly designed system needs to be absolute as all the procedures have been tested and confirmed efficient.

Therefore, its appliocation in the payroll section will meet both the present and future needs of the payroll activities in NSADP, Minna.

REFERENCE

(1) A. Fapohunda — Understanding and Using Micro Computer 1995 - AFLON Ltd.

(2) C.J. Date — An introduction to System Dbase IV Programming 1990 - Addison-Wesley Publishing Company.

(2) C. K. AYO — Computer Literacy Operation and Appreciation 1994 - Alanukitan Commercial Press Ltd.

(4) H. Spice — Database Design - Tricon Technology Corporation 1994

(5) I.W.J. Owler & J.L. Brown — Wheldon's Cost Accounting and Costing Methods Pulication, Macdonald & Evans Limited 1975. London.

(6) M. James — Information Engineering Introducton Book 1 - Prentice Hall 1989.

(7) S. Greeme - Van — Data Modelling Essentials Analysis - Nostrand Reinhold 1994

(8) T. Luccy — Cost and Management Accounting Published 1981.

(9) T. Q. Do — Fundamental of Information System - LT Associates Washington DC Nov., 1994.

APPENDIX

PAYROLL.PRG

```
set colo to
set scor off
set dele off
set stat off
set echo off
set safe off
set talk off
*set inte off
set bell off
set date brit
clea
do whil .t.
  @ 1,0 to 24,79 doub
  @ 0,18 to 2,61 doub
  @ 4,19 to 6,60 doub
  @ 7,1 to 7,78
  @ 8,2 to 11,14
  @ 8,16 to 11,26
  @ 8,28 to 11,36
  @ 8,38 to 11,49
  @ 8,51 to 11,58
  @ 8,60 to 10,70
  @ 8,72 to 10,77
  @ 21,21 to 23,58
  @ 1,19 say " AGRICULTURAL DEVELOPMENT PROJECT - MINNA "
  @ 5,20 say "COMPUTERISED PAYROLL SYSTEM -- MAIN MENU"
  @ 9,3 say "STAFF FILE "
  @ 9,17 say "VARIATION"
  @ 9,29 say "MONTHLY"
  @ 9,39 say "  REPORT  "
  @ 9,52 say "FILES "
  @ 9,61 say "UTILITIES"
  @ 9,73 say "QUIT"
  @ 10,3 say "MAINTENANCE"
  @ 10,17 say "  ENTRY  "
  @ 10,29 say "  RUN  "
  @ 10,39 say "GENERATION"
  @ 10,52 say "UPDATE"
  @ 22,22 say "Press 1st letter to pick choice ..."
```

```
  p=0
  do whil p=0
    p=inkey()
    if upper(chr(p)) $ "SVMRFUQ"
      exit
    endi
    p=0
  endd
  @ 21,21 clea to 23,58
  do case
    case upper(chr(p)) $ "S"
      do maint
    case upper(chr(p)) $ "V"
      do vary
    case upper(chr(p)) $ "M"
      do run
    case upper(chr(p)) $ "R"
      do report
    case upper(chr(p)) $ "F"
      do f_update
    case upper(chr(p)) $ "U"
      do utility
    othe
      exit
  endc
endd
clea all
clea
retu


MAINT.PRG
@ 4,19 clea to 6,60
@ 4,25 to 6,53 doub
@ 5,26 say "STAFF FILE MAINTENANCE MENU"
do whil .t.
  set colo to n/w*
  @ 9,3 say 'STAFF FILE '
  @ 10,3 say 'MAINTENANCE'
  set colo to
  @ 12,2 to 21,27
  @ 18,3 to 18,26
  @ 13,3 say "[A]dding Staff Record"
  @ 14,3 say "[V]iewing Staff Record"
  @ 15,3 say "[M]odifying Staff Record"
  @ 16,3 say "[D]eleting Staff Record"
```

```
  @ 17,3 say "[E] X I T"
  @ 20,5 say "Pick your choice ..."
  ma=0
  do whil ma=0
    ma=inkey()
    if upper(chr(ma)) $ "AVMDE"
      exit
    endi
    ma=0
  endd
  save scre
  do case
    case upper(chr(ma)) $ "A"
      clear
      do add
    case upper(chr(ma)) $ "V"
      clea
      do view
    case upper(chr(ma)) $ "M"
      clea
      do edit
    case upper(chr(ma)) $ "D"
      clea
      do delete
    othe
      @ 12,2 clea to 21,27
      exit
  endc
  rest scre
endd
@ 12,2 clea to 21,27
retu


ADD.PRG

mdate=date()
use current
cur_pro=c_process
cur_date=c_date
use staff1
mt_appt=t_appt
use
if mt_appt='1'
  copy file asuu_sal.dbf to sal.dbf
else
```

```
  copy file nasu_sal.dbf to sal.dbf
endi
cur_day=day(cur_date)
cur_mth=month(cur_date)
cur_year=year(cur_date)
today=date()
do while .t.
  clea
  use staff1 index staff1
  RESTORE FROM mem_add addi
* control='N'
  @ 2,14 SAY "C O M P U T E R I S E D  P A Y R O L L  S Y S T E
M"
  @ 4,26 say "D A T A  E N T R Y  F O R M"
  @ 3,25 TO 5,53
  @ 1,13 to 3,65 doub
  @ 0,0 to 24,79 doub
  @ 0,4 say "A G R I C U L T U R A L  D E V E L O P M E N T  P R
O J E C T- M I N N A"
  @ 6,4 SAY 'STAFF NUMBER (Enter "*****" to Exit):'
  do whil .t.
    @ 6,42 GET MS_NUMB
    read
    if ms_numb="*****"
      exit
    endi
    ms_numb=rtrim(ms_numb)
    n=len(ms_numb)
    if n<5
      n1=5-n
      ms_numb=repl("0",n1)+ms_numb
    endi
    if .not. eof()
      seek ms_numb
      if found()
        ? chr(7)
        @ 23,15 say "STAFF NUMBER already exist !!! - Press any
key ..."
        set cons off
        wait
        set cons on
        @ 23,15 say spac(55)
        ms_numb=spac(5)
        loop
      endi
    endi
```

```
    exit
  endd
  if ms_numb="*****"
    exit
  endi
  @ 6,4 say space(45)
  @ 6,4 say "STAFF NUMBER:" get ms_numb
  @ 6,53 say "DATE:"
  set colo to n/w
  @ 6,58 say dtoc(today)
  set colo to
  clea gets
  @ 20,1 to 20,78
  do while .t.
    do add_scr1
    do add_scr2
    do add_scr3
    @ 23,21 say "(S)ave      (R)epeat      (A)bandon ..."
    a=0
    do while a=0
      a=inkey()
      if upper(chr(a)) $ "SRA"
        exit
      endif
      a=0
    enddo
    if upper(chr(a)) $ "R"
      @ 8,1 clea to 23,78
      mcontrol='Y'
      loop
    endif
    exit
  enddo
  if upper(chr(a)) $ "S"
    use staff1 index staff1
    appe blan
    do repl1a
    use staff2 index staff2
    appe blan
    do repl2a
    use staff3 index staff3
    appe blan
    do repl3a
  endi
  rele all like m*
endd
```

```
rele all
clos all
eras sal.dbf
retu


EDIT.PRG
use staff1
mt_appt=t_appt
use
if mt_appt='1'
  copy file asuu_sal.dbf to sal.dbf
else
  copy file nasu_sal.dbf to sal.dbf
endi
rest from mem_edit
sele a
  use staff1 index staff1
sele b
  use marital
sele c
  use dept index dept
sele d
  use rank index rank
sele e
  use appt
sele f
  use sal
sele g
  use bank index bank
@ 1,0 to 24,79 doub
@ 0,19 to 2,60 doub
@ 1,20 say "AGRICULTURAL DEVELOPMENT PROJECT - MINNA"
@ 4,16 say "COMPUTERISED PAYROLL SYSTEM - DATA EDITING FORM"
@ 5,16 to 5,62
@ 22,1 to 22,78
sele a
do whil .t.
  ms_numb=spac(5)
  @ 6,4 say 'STAFF NUMBER (Enter "*****" to Exit):' get ms_numb
  read
  if ms_numb="*****"
    exit
  endi
  ms_numb=rtrim(ms_numb)
  n=len(ms_numb)
```

```
  if n<5
    n1=5-n
    ms_numb=repl("0",n1)+ms_numb
  endi
  seek ms_numb
  if .not. found()
    ? chr(7)
    @ 23,16 say "Invalid STAFF NUMBER entered !!! - Press any key
..."
    set cons off
    wait
    set cons on
    @ 23,16 say spac(55)
    loop
  endi
  do transf1a
  @ 6,4 say spac(45)
  @ 6,4 say "STAFF NUMBER:" get ms_numb
  @ 6,53 say "DATE:"
  set colo to n/w
  @ 6,58 say date()
  set colo to
  do edit1
  do whil .t.
    do edit2
    @ 23,21 say "(S)ave      (R)epeat      (A)bandon ..."
    e=0
    do whil e=0
      e=inkey()
      if upper(chr(e)) $ "SRA"
        exit
      endi
      e=0
    endd
    @ 23,21 say spac(50)
    if upper(chr(e)) $ "R"
      loop
    endi
    exit
  endd
  if upper(chr(e)) $ "S"
    do repl_edi
  endi
  @ 6,1 clea to 21,78
enddo
clos data
```

```
eras sal.dbf
retu


VIEW.PRG
rest from mem_add
sele a
  use allce index allce
sele b
  use deduct index deduct
sele c
  use staff1 index staff1
sele d
  use staff2 index staff2
sele e
  use staff3 index staff3
@ 1,0 to 24,79 doub
@ 0,19 to 2,60 doub
@ 1,20 say "AGRICULTURAL DEVELOPMENT PROJECT - MINNA"
@ 4,16 say "COMPUTERISED PAYROLL SYSTEM - DATA VIEWING FORM"
@ 5,16 to 5,62
@ 21,1 to 21,78
sele c
do whil .t.
  ms_numb=spac(5)
  @ 6,4 say 'STAFF NUMBER (Enter "*****" to Exit):' get ms_numb
  read
  if ms_numb="*****"
    exit
  endi
  ms_numb=rtrim(ms_numb)
  n=len(ms_numb)
  if n<5
    n1=5-n
    ms_numb=repl("0",n1)+ms_numb
  endi
  seek ms_numb
  if .not. found()
    ? chr(7)
    @ 23,16 say "Invalid STAFF NUMBER entered !!! - Press any key
..."
    set cons off
    wait
    set cons on
    @ 23,16 say spac(55)
    loop
```

```
endi
do transf1a
@ 6,4 say spac(45)
@ 6,4 say "STAFF NUMBER:" get ms_numb
@ 6,53 say "DATE:"
set colo to n/w
@ 6,58 say date()
set colo to
do vie_edit
@ 23,19 say "Press any key to go to the next screen ..."
set cons off
wait
set cons on
@ 6,1 clea to 20,78
@ 23,19 say spac(45)
sele d
seek ms_numb
do tr2_vary
sele e
seek ms_numb
do tr3_vary
do vary2
@ 8,1 clea to 20,78
@ 23,19 say spac(45)
@ 7,26 say "L I S T  O F  A L L O W A N C E S"
do add_2a
do loop2a
do add_2b
do loop2b
@ 23,19 say "Press any key to go to the next screen ..."
set cons off
wait
set cons on
@ 7,40 say "D E D U C T I O N S"
@ 8,2 say "DEDUCTIONS"
do add_3a
do loop3a
do add_3b
do loop3b
@ 23,26 say "Press any key to continue ..."
set cons off
wait
set cons on
ms_numb=spac(5)
@ 6,1 clea to 20,78
@ 23,26 say spac(30)
```

```
  sele c
endd
clos all
retu


DELETE.PRG
stor spac(11) to mbank_no
stor spac(5) to ms_numb,minitial
stor spac(15) to msurname,mf_name,mo_names
stor spac(1) to mms_code,mappt_code,mt_appt,minc_flag,mlg_code
stor 0 to
mdept_code,msch_code,msal_l,msal_s,mbank_code,mrank_code
stor ctod("  /  /  ") to mdf_appt
stor 0 to mper_sal,ma_sal,ma_fpay
sele a
  use staff1 index staff1
sele b
  use staff2 index staff2
sele c
  use staff3 index staff3
sele a
@ 1,0 to 24,79 doub
@ 0,19 to 2,60 doub
@ 1,20 say "AGRICULTURAL DEVELOPMENT PROJECT - MINNA"
@ 4,16 say "COMPUTERISED PAYROLL SYSTEM - DATA DELETING FORM"
@ 5,16 to 5,63
@ 21,1 to 21,78
do whil .t.
  ms_numb=spac(5)
  @ 6,4 say 'STAFF NUMBER (Enter "*****" to Exit):' get ms_numb
  read
  if ms_numb="*****"
    exit
  endi
  ms_numb=rtrim(ms_numb)
  n=len(ms_numb)
  if n<5
    n1=5-n
    ms_numb=repl("0",n1)+ms_numb
  endi
  seek ms_numb
  if .not. found()
    ? chr(7)
    @ 23,16 say "Invalid STAFF NUMBER entered !!! - Press any key
..."
```

```
    set cons off
    wait
    set cons on
    @ 23,16 say spac(55)
    loop
  endi
  do transf1a
  @ 6,4 say spac(45)
  @ 6,4 say "STAFF NUMBER:" get ms_numb
  @ 6,53 say "DATE:"
  set colo to n/w
  @ 6,58 say date()
  set colo to
  do vie_edit
  @ 23,18 say "Is this the record to be DELETED (Y/N)? ..."
  d=0
  do whil d=0
    d=inkey()
    if upper(chr(d)) $ "YN"
      exit
    endi
    d=0
  endd
  @ 23,18 say spac(45)
  if upper(chr(d)) $ "Y"
    dele
    pack
    sele b
    seek ms_numb
    dele
    pack
    sele c
    seek ms_numb
    dele
    pack
    sele a
    @ 23,16 say "Record has been deleted !!! - Press any key ..."
  else
    @ 23,14 say "Record has not been deleted !!! - Press any key
..."
  endi
  set cons off
  wait
  set cons on
  @ 6,1 clea to 20,78
  @ 23,14 say spac(60)
```

```
enddo
clear
clos all
retu


VARY.PRG
@ 4,19 clea to 6,60
@ 4,28 to 6,51 doub
@ 5,29 say "VARIATION ENTRY SCREEN"
set colo to n/w*
@ 9,17 say "VARIATION"
@ 10,17 say "  ENTRY  "
set colo to
use current
mc_date=c_date
mc_process=c_process
use
mc_year=year(mc_date)
mc_month=month(mc_date)
namemth=cmonth(mc_date)
lastmth=substr(dtoc(mc_date),4,2)
lastmth=val(lastmth)
thismth=lastmth+1
nextmth=thismth+1
nextmth=str(nextmth)
no=len(ltrim(nextmth))
if no=1
  nextmth="0"+ltrim(nextmth)
endif
mdd="01"
myy=ltrim(str(mc_year))
myy=substr(myy,3,2)
maindate=space(8)
maindate=mdd+"/"+nextmth+"/"+myy
maindate=ctod(maindate)
maindate=maindate-1
*save screen
do while .t.
  @ 16,13 to 19,65
  @ 17,14 say spac(36)+"EXIT = 01/01/01"
  @ 18,14 say "Enter Date of Variation Period [DD/MM/YY]"
  date=ctod("  /  /  ")
  @ 18,57 get date
  read
  if dtoc(date) = "01/01/01"
```

```
    @ 16,13 clea to 19,65
    retu
  endi
  cur_year=year(date)
  cur_mth=cmonth(date)
  if mc_date=date
    if mc_process='Y'
      @ 16,13 clea to 19,65
      @ 15,16 say "SORRY - You can't carry out any variation
entry"
      @ 16,24 say "for the month of "+upper(cur_mth)+",
"+ltrim(str(cur_year))+"."
      @ 17,24 say "Payroll has been processed for the month."
      @ 20,25 say "P R E S S  A N Y  K E Y  !!!"
      set cons off
      wait
      set cons on
      @ 15,15 clea to 20,65
      retu
    endi
    exit
  endi
  if mc_date<>date
    year=year(date)
    month=month(date)
    if month=1
      month=13
    endif
    if year<mc_year .and. month<mc_month
      @ 23,25 say "Invalid date entered - Press any key ..."
      set cons off
      wait
      set cons on
      @ 23,25 say spac(42)
      loop
    endif
    if year>mc_year .and. month>mc_month+1
      @ 23,25 say "Invalid date entered - Press any key ..."
      set cons off
      wait
      set cons on
      @ 23,25 say spac(42)
      loop
    endif
    if year<mc_year .or. year>mc_year+1
      @ 23,25 say "Invalid year entered - Press any key ..."
```

```
      set cons off
      wait
      set cons on
      @ 23,25 say spac(42)
      loop
    endif
    if month<mc_month .or. month>mc_month+1
      @ 23,25 say "Invalid month entered - Press any key ..."
      set cons off
      wait
      set cons on
      @ 23,25 say spac(42)
      loop
    endif
    if month=1 .and. year<>mc_year+1
      @ 23,25 say "Invalid year entered - Press any key ..."
      set cons off
      wait
      set cons on
      @ 23,25 say spac(42)
      loop
    endif
    if month=12
      mc_year=mc_year+1
      myy=ltrim(str(mc_year))
      myy=substr(myy,3,2)
      nextmth="01"
      maindate=space(8)
      maindate=mdd+"/"+nextmth+"/"+myy
      maindate=ctod(maindate)
      maindate=maindate-1
    endif
    if month=13
      myy=ltrim(str(mc_year+1))
      myy=substr(myy,3,2)
      nextmth="02"
      maindate=space(8)
      maindate=mdd+"/"+nextmth+"/"+myy
      maindate=ctod(maindate)
      maindate=maindate-1
    endif
    if date<>maindate
      @ 23,5 say "Invalid day entered, enter last date of the
month - Press any key ..."
      set cons off
      wait
```

```
      set cons on
      @ 23,5 say spac(72)
      loop
    endi
    @ 16,13 clea to 19,65
    @ 16,19 say "You are about to commence Variation Entry"
    @ 17,19 say "for the Month of "+upper(cur_mth)+",
"+ltrim(str(cur_year))+"."
    @ 19,22 say "Do you wish to continue (Y/N)? ..."
    v=0
    do while v=0
      v=inkey()
      if upper(chr(v)) $ "YN"
        exit
      endif
      v=0
    enddo
    @ 16,18 clear to 19,65
    if upper(chr(v)) $ "N"
    *  rest scre
      loop
    endi
    use current
    repl c_date with date,c_process with 'N'
    use
  endi
  exit
enddo
rest from mem_vary addi
clea
@ 1,0 to 24,79 doub
@ 0,4 to 2,74 doub
@ 1,5 say "AGRICULTURAL DEVELOPMENT PROJECT, MINNA - COMPUTERISED
PAYROLL SYSTEM"
@ 3,11 say "OPERATION:"
@ 3,22 say "VARIATION ENTRY FOR THE MONTH OF "+upper(cur_mth)+",
"+ltrim(str(cur_year))
do whil .t.
  @ 6,5 say 'STAFF NUMBER (Enter "*****" to Exit):' get ms_numb
  read
  if ms_numb='*****'
    exit
  endi
  ms_numb=rtrim(ms_numb)
  n=len(ms_numb)
  if n<5
```

```
    n1=5-n
    ms_numb=repl("0",n1)+ms_numb
  endi
  use staff1 index staff1
  seek ms_numb
  if .not. found()
    ? chr(7)
    @ 23,14 say "Invalid STAFF NUMBER Entered !!! - Press any key
..."
    set cons off
    wait
    set cons on
    @ 23,14 say spac(55)
    ms_numb=spac(5)
    loop
  endif
  v_name=rtrim(surname)+', '+rtrim(initial)
  v_dept=dept_code
  v_rank=rank_code
  leave_code=lg_code
  use dept index dept
  seek v_dept
  dept=rtrim(dept_name)
  use rank index rank
  seek v_rank
  rank=rtrim(rank_desc)
  @ 4,1 say "STAFF NUMBER:" get ms_numb
  @ 4,50 say "NAME:" get v_name
  @ 5,1 say "DEPARTMENT:" get dept
  @ 5,50 say "RANK:" get rank
  clea gets
  @ 6,1 to 6,78
  @ 7,20 say "D E T A I L S  O F  A L L O W A N C E S"
  use staff2 index staff2
  seek ms_numb
  do tr2_vary
  use staff3 index staff3
  seek ms_numb
  do tr3_vary
  use
  sele a
    use allce index allce
  sele b
    use deduct index deduct
  do vary2
  do whil .t.
```

```
    do add_2a
    do loop2a
    sele a
    do va_seek
    @ 7,40 say "D E D U C T I O N S"
    @ 8,2 say "DEDUCTIONS"
    do add_3a
    do loop3a
    sele b
    do vd_seek
    @ 23,21 say "(S)ave      (R)epeat      (A)bandon ..."
    a=0
    do while a=0
      a=inkey()
      if upper(chr(a)) $ "SRA"
        exit
      endif
      a=0
    enddo
*   close data
    if upper(chr(a)) $ "R"
      @ 23,21 say spac(40)
      @ 10,1 clea to 20,78
      @ 7,40 say "A L L O W A N C E S"
*     @ 10,1 clea to 23,78
      sele a
      loop
    endif
    exit
  enddo
  clos data
  if upper(chr(a)) $ "S"
    use staff2 index staff2
    seek ms_numb
    do repl2v
    use staff3 index staff3
    seek ms_numb
    do repl3v
  endif
  ms_numb=spac(5)
  @ 23,21 say spac(40)
  @ 4,1 clea to 20,78
*  sele a
enddo
rele all
clos all
```

fire, water etc by storage in metal cabinets, the space taken up by these cabinets must also be rented.

(ii) How to find these documents when needed. It takes time before such documents are seen.
(Davies D. Information Technology at Work).

In view of these statements. It is seen that where information processing is done manually, frauds are committed and the organisation can be set ablaze and the whole documents are burnt and destroyed and nobody is able to detect exactly who committed such fraud. Like what happened in **NITEL HOUSE** that was engulfed by fire and also the **DEFENCE HOUSE** at Lagos where so many important documents were lost as a result of the fire incident.

However, in our ministries or government parastatals, some people use the manual system to make money. When your file is being processed, some clerks retrieve these files and hide them causing delayment in processing the files. The clerks have to be bribed before such files are released. In fact so much have been said concerning the problems associated with the manual system of information processing. The record keeping system of the maintenance workshop of the organisation cannot be free from such problems if the recording system is not computerized.

```
clea
retu


VARY2.PRG
  sele a
  if ma_code1<>0
    seek ma_code1
    ma_desc1=a_desc
  else
    ma_desc1=spac(15)
  endif
  if ma_code2<>0
    seek ma_code2
    ma_desc2=a_desc
  else
    ma_desc2=spac(15)
  endif
  if ma_code3<>0
    seek ma_code3
    ma_desc3=a_desc
  else
    ma_desc3=spac(15)
  endif
  if ma_code4<>0
    seek ma_code4
    ma_desc4=a_desc
  else
    ma_desc4=spac(15)
  endif
  if ma_code5<>0
    seek ma_code5
    ma_desc5=a_desc
  else
    ma_desc5=spac(15)
  endif
  if ma_code6<>0
    seek ma_code6
    ma_desc6=a_desc
  else
    ma_desc6=spac(15)
  endif
  if ma_code7<>0
    seek ma_code7
    ma_desc7=a_desc
  else
```

```
  ma_desc7=spac(15)
endif
if ma_code8<>0
  seek ma_code8
  ma_desc8=a_desc
else
  ma_desc8=spac(15)
endif
if ma_code9<>0
  seek ma_code9
  ma_desc9=a_desc
else
  ma_desc9=spac(15)
endif
if ma_code10<>0
  seek ma_code10
  ma_desc10=a_desc
else
  ma_desc10=spac(15)
endif
if ma_code11<>0
  seek ma_code11
  ma_desc11=a_desc
else
  ma_desc11=spac(15)
endif
if ma_code12<>0
  seek ma_code12
  ma_desc12=a_desc
else
  ma_desc12=spac(15)
endif
sele b
if md_code1<>0
  seek md_code1
  md_desc1=d_desc
else
  md_desc1=spac(15)
endif
if md_code2<>0
  seek md_code2
  md_desc2=d_desc
else
  md_desc2=spac(15)
endif
if md_code3<>0
```

```
  seek md_code3
  md_desc3=d_desc
else
  md_desc3=spac(15)
endif
if md_code4<>0
  seek md_code4
  md_desc4=d_desc
else
  md_desc4=spac(15)
endif
if md_code5<>0
  seek md_code5
  md_desc5=d_desc
else
  md_desc5=spac(15)
endif
if md_code6<>0
  seek md_code6
  md_desc6=d_desc
else
  md_desc6=spac(15)
endif
if md_code7<>0
  seek md_code7
  md_desc7=d_desc
else
  md_desc7=spac(15)
endif
if md_code8<>0
  seek md_code8
  md_desc8=d_desc
else
  md_desc8=spac(15)
endif
if md_code9<>0
  seek md_code9
  md_desc9=d_desc
else
  md_desc9=spac(15)
endif
if md_code10<>0
  seek md_code10
  md_desc10=d_desc
else
  md_desc10=spac(15)
```

```
  endif
  if md_code11<>0
    seek md_code11
    md_desc11=d_desc
  else
    md_desc11=spac(15)
  endif
  if md_code12<>0
    seek md_code12
    md_desc12=d_desc
  else
    md_desc12=spac(15)
  endif
  retu


RUN.PRG
@ 4,19 clea to 6,60
@ 4,30 to 6,49 doub
@ 5,31 say "MONTHLY RUN SCREEN"
set colo to n/w*
@ 9,29 say "MONTHLY"
@ 10,29 say "  RUN  "
set colo to
use current
mc_date=c_date
mc_process=c_process
use
mc_year=year(mc_date)
mc_month=month(mc_date)
namemth=cmonth(mc_date)
lastmth=substr(dtoc(mc_date),4,2)
lastmth=val(lastmth)
thismth=lastmth+1
nextmth=thismth+1
nextmth=str(nextmth)
no=len(ltrim(nextmth))
if no=1
  nextmth="0"+ltrim(nextmth)
endif
mdd="01"
myy=ltrim(str(mc_year))
myy=substr(myy,3,2)
maindate=space(8)
maindate=mdd+"/"+nextmth+"/"+myy
```

```
maindate=ctod(maindate)
maindate=maindate-1
*save screen
do while .t.
  @ 16,9 to 19,70
  date=ctod("  /  /  ")
  @ 17,10 say spac(44)+"Exit =  01/01/01"
  @ 18,10 say "Enter Date of Payroll Processing Period
[DD/MM/YY]:" get date
  read
  if dtoc(date) = "01/01/01"
    @ 16,9 clea to 19,70
    retu
  endi
  cur_year=year(date)
  cur_mth=cmonth(date)
  if mc_date=date .and. mc_process='Y'
    @ 16,9 clea to 19,70
    @ 16,23 say "Payroll has been processed for the"
    @ 17,23 say "Month of "+cur_mth+", "+ltrim(str(cur_year))+"."
    @ 19,25 say "Press any key to continue ..."
    set cons off
    wait
    set cons on
    @ 16,23 clea to 19,58
    retu
  endi
  if mc_date<>date
    year=year(date)
    month=month(date)
    if month=1
      month=13
    endif
    if year<mc_year .and. month<mc_month
      @ 23,25 say "Invalid date entered - Press any key ..."
      set cons off
      wait
      set cons on
      @ 23,25 say spac(42)
      loop
    endif
    if year>mc_year .and. month>mc_month+1
      @ 23,25 say "Invalid date entered - Press any key ..."
      set cons off
      wait
      set cons on
```

```
  @ 23,25 say spac(42)
  loop
endif
if year<mc_year .or. year>mc_year+1
  @ 23,25 say "Invalid year entered - Press any key ..."
  set cons off
  wait
  set cons on
  @ 23,25 say spac(42)
  loop
endif
if month<mc_month .or. month>mc_month+1
  @ 23,25 say "Invalid month entered - Press any key ..."
  set cons off
  wait
  set cons on
  @ 23,25 say spac(42)
  loop
endif
if month=1 .and. year<>mc_year+1
  @ 23,25 say "Invalid year entered - Press any key ..."
  set cons off
  wait
  set cons on
  @ 23,25 say spac(42)
  loop
endif
if month=12
  mc_year=mc_year+1
  myy=ltrim(str(mc_year))
  myy=substr(myy,3,2)
  nextmth="01"
  maindate=space(8)
  maindate=mdd+"/"+nextmth+"/"+myy
  maindate=ctod(maindate)
  maindate=maindate-1
endif
if month=13
  myy=ltrim(str(mc_year+1))
  myy=substr(myy,3,2)
  nextmth="02"
  maindate=space(8)
  maindate=mdd+"/"+nextmth+"/"+myy
  maindate=ctod(maindate)
  maindate=maindate-1
endif
```

```
    if date<>maindate
      @ 23,5 say "Invalid day entered, enter last date of the
month - Press any key ..."
      set cons off
      wait
      set cons on
      @ 23,5 say spac(72)
      loop
    endi
    if mc_process='Y'
      @ 16,9 clea to 19,70
      @ 17,19 say "You have not enter variation for
"+upper(cur_mth)+', '+ltrim(str(cur_year))+'.'
      @ 19,26 say "To continue [Y/N]? ..."
      r=0
      do while r=0
        r=inkey()
        if upper(chr(r)) $ "YN"
          exit
        endif
        r=0
      enddo
      @ 17,19 clear to 19,75
      if upper(chr(r)) $ "N"
        *restore screen
        loop
      endi
    endi
  endi
  @ 16,9 clea to 19,70
  @ 16,20 say "You are about to process payroll for the"
  @ 17,20 say "Month of "+upper(cur_mth)+",
"+ltrim(str(cur_year))+"."
  @ 19,23 say "Do you want to continue (Y/N)? ..."
  r=0
  do while r=0
    r=inkey()
    if upper(chr(r)) $ "YN"
      exit
    endif
    r=0
  enddo
  @ 16,20 clear to 19,65
  if upper(chr(r)) $ "N"
    retu
  endi
```

```
  exit
enddo
@ 18,28 say "WAIT - Preparing necessary files !!!"
use amonthly
appe from monthly
use monthly
zap
use monthly2
zap
rest from mem_run addi
use staff1
tot_rec=reccount()
cur_rec=0
use
sele a
  use staff1 index staff1
sele b
  use staff2 index staff2
sele c
  use staff3 index staff3
sele d
  use monthly
sele e
  use monthly2
*@ 12,22 say spac(40)
@ 18,28 say spac(40)
clea
@ 3,12 to 22,67 doub
*@ 4,25 to 6,53
@ 2,19 to 4,60 doub
@ 5,25 to 7,53
@ 3,20 say "AGRICULTURAL DEVELOPMENT PROJECT - MINNA"
*@ 4,26 say spac(27)
@ 6,26 say "COMPUTERISED PAYROLL SYSTEM"
@ 10,14 say "Payroll Processing for the Month of
"+upper(cur_mth)+", "+ltrim(str(cur_year))
@ 11,32 say "IN PROGRESS !!!"
@ 13,29 to 17,51
@ 14,45 to 16,45
@ 15,30 to 15,49
@ 14,31 say "TOTAL RECORD"
@ 14,47 get tot_rec pict '999'
@ 16,31 say "CURRENT RECORD"
@ 20,34 say "W A I T  !!!"
sele a
do whil .not. eof()
```

```
mtax_ref=0
cur_rec=cur_rec+1
@ 16,47 get cur_rec pict '999'
stor 0 to mth_all,mth_ded
do tr1_run
sele b
seek ms_numb
do tr2_run
do all_run
skip
sele c
seek ms_numb
do tr3_run
do ded_run
skip
mth_all=round(mth_all,2)
mth_ded=round(mth_ded,2)
if msal_l=0 .or. msal_l>5
  if ma_fpay1>ma_fpay2
    if mno_mth>0
      do refund
    endi
  endi
endi
taxgr=mper_sal/100*ma_sal
if ma_code1>=51 .and. ma_code1<=70 .and. ma_sign1='+'
  taxgr=taxgr+ma_amt1*12
endi
if ma_code2>=51 .and. ma_code2<=70 .and. ma_sign2='+'
  taxgr=taxgr+ma_amt2*12
endi
if ma_code3>=51 .and. ma_code3<=70 .and. ma_sign3='+'
  taxgr=taxgr+ma_amt3*12
endi
if ma_code4>=51 .and. ma_code4<=70 .and. ma_sign4='+'
  taxgr=taxgr+ma_amt4*12
endi
if ma_code5>=51 .and. ma_code5<=70 .and. ma_sign5='+'
  taxgr=taxgr+ma_amt5*12
endi
if ma_code6>=51 .and. ma_code6<=70 .and. ma_sign6='+'
  taxgr=taxgr+ma_amt6*12
endi
if ma_code7>=51 .and. ma_code7<=70 .and. ma_sign7='+'
  taxgr=taxgr+ma_amt7*12
endi
```

```
if ma_code8>=51 .and. ma_code8<=70 .and. ma_sign8='+'
  taxgr=taxgr+ma_amt8*12
endi
if ma_code9>=51 .and. ma_code9<=70 .and. ma_sign9='+'
  taxgr=taxgr+ma_amt9*12
endi
if ma_code10>=51 .and. ma_code10<=70 .and. ma_sign10='+'
  taxgr=taxgr+ma_amt10*12
endi
if ma_code11>=51 .and. ma_code11<=70 .and. ma_sign11='+'
  taxgr=taxgr+ma_amt11*12
endi
if ma_code12>=51 .and. ma_code12<=70 .and. ma_sign12='+'
  taxgr=taxgr+ma_amt12*12
endi
mtax_gr=mtax_gr+taxgr/12
if msal_l>0 .and. msal_l<6
  mtax=(0.5/100*taxgr)/12
else
  do tax
endi
mtax=round(mtax,2)
mbasic_sal=ma_sal/12
mbasic_sal=mper_sal/100*mbasic_sal
mbasic_sal=round(mbasic_sal,2)
mfpay=ma_fpay1/12
mfpay=round(mfpay,2)
macc_fpay=macc_fpay+mfpay
mtax_jan=macc_tax
macc_tax=macc_tax+mtax
mno_mth=mno_mth+1
mgr_pay=mbasic_sal+mth_all+mtax_ref
mtot_ded=mth_ded+mtax
mnet_pay=mgr_pay-mtot_ded
sele d
appe blan
do repl_run
sele e
appe blan
do repl_r2
sele a
repl acc_tax with macc_tax,tax_gr with mtax_gr
repl acc_fpay with macc_fpay,tax_pay with mtax_pay
repl a_fpay2 with ma_fpay1,no_mth with mno_mth
mtax_ref=0
skip
```

```
endd
clos data
use current
repl c_date with date,c_process with "Y"
*@ 16,34 say spac(15)
clear
@ 20,16 say "PAYROLL PROCESSING COMPLETED - Press any key ..."
set cons off
wait
set cons on
@ 20,16 say spac(60)
clos data
retu


REPORT.PRG

@ 4,19 clea to 6,60
@ 4,28 to 6,51 doub
@ 5,29 say "REPORT GENERATION MENU"
do whil .t.
  set colo to n/w*
  @ 9,39 say "  REPORT  "
  @ 10,39 say "GENERATION"
  set colo to
  @ 12,38 to 23,61
  @ 13,39 say "[P]ayslip"
  @ 14,39 say "[B]ank Schedule"
  @ 15,39 say "[T]ax Report"
  @ 16,39 say "[D]epartmental Summary"
  @ 17,39 say "D[E]duction List"
  @ 18,39 say "[S]taff List"
  @ 19,39 say "T[R]ansfer Voucher"
  @ 20,39 say "[Q] U I T"
  @ 22,40 say "Pick your choice ..."
  r=0
  do whil r=0
    r=inkey()
    if upper(chr(r)) $ "PBTDESRQ"
      exit
    endi
    r=0
  endd
*  @ 22,40 say spac(22)
  do case
    case upper(chr(r)) $ "P"
```

```
      do payslip
    case upper(chr(r)) $ "B"
      do schedule
    case upper(chr(r)) $ "T"
      do tax_rep
    case upper(chr(r)) $ "D"
      do dsummary
    case upper(chr(r)) $ "E"
      do deduct
    case upper(chr(r)) $ "S"
      do staff
    case upper(chr(r)) $ "R"
      do tv
    othe
      exit
  endc
endd
clos data
@ 12,38 clea to 23,61
retu


PAYSLIP.PRG
set colo to n/w*
@ 13,39 say "[P]ayslip"
set colo to
@ 13,7 to 20,31 doub
@ 14,9 say "Switch Printer On"
@ 15,9 say "Be sure it is ready."
@ 18,8 say 'PRESS: "C" to CONTINUE'
@ 19,8 say '        "S" to STOP ...'
p=0
do whil p=0
  p=inkey()
  if upper(chr(p)) $ "CS"
    exit
  endi
  p=0
endd
@ 13,7 clea to 20,31
if upper(chr(p)) $ "S"
  retu
endi
do whil .t.
  exitting='N'
  @ 13,10 to 19,27
```

```
@ 14,12 say "To Print  ll"
@ 15,12 say "To Print  art"
@ 16,12 say "To  xit"
set inte on
set colo to r
@ 14,21 say "A"
@ 15,21 say "P"
@ 16,15 say "E"
set colo to
set inte off
@ 18,11 say "Pick Choice ..."
p=0
do whil p=0
  p=inkey()
  if upper(chr(p)) $ "APE"
    exit
  endi
  p=0
endd
if upper(chr(p)) $ "E"
  @ 13,10 clea to 19,27
  clea all
  retu
endi
rec1=0
if upper(chr(p)) $ "P"
  @ 14,21 say "A"
  @ 15,21 say "P"
  @ 16,15 say "E"
  set colo to n/w
  @ 15,12 say "To Print Part"
  set colo to
  use monthly
  @ 21,3 say "Enter Staff No to start from:"
  @ 23,24 say 'EXIT = "*****"'
  @ 20,2 to 22,37
  num='     '
  do whil .t.
    @ 21,32 get num
    read
    @ 23,23 clea to 23,37
    if num="*****"
      use
      exitting='Y'
      @ 13,2 clea to 22,37
      exit
```

```
      endi
      num=rtrim(num)
      n=len(num)
      if n<5
        n1=5-n
        num=repl("0",n1)+num
      endi
      loca for s_numb=num
      if .not. found()
        @ 23,5 say "Invalid STAFF NUMBER entered - Press any key
..."
        set cons off
        wait
        set cons on
        num='     '
        @ 23,5 say spac(50)
        @ 23,24 say 'EXIT = "*****"'
        loop
      endi
      use
      @ 20,2 clea to 22,37
      exit
    endd
  endi
  if exitting='Y'
    loop
  else
    exit
  endi
endd
@ 13,10 clea to 19,27
@ 16,6 say "WAIT - Preparing files ..."
use current
cur_mth=cmonth(c_date)
cur_year=year(c_date)
use monthly
sort on dept_code,s_numb to smonthly
use monthly2
index on s_numb to monthly2
use
rest from mem_pay addi
sele a
  use smonthly
sele b
  use dept index dept
sele c
```

```
  use bank index bank
sele d
  use allce index allce
sele e
  use deduct index deduct
sele f
  use monthly2 index monthly2
r=1
part=0
sele a
if upper(chr(p)) $ "P"
  loca for s_numb=num
endi
@ 16,6 say spac(28)
@ 14,5 to 18,32 doub
@ 15,10 say "K E E P  O F F  !!!"
@ 17,7 say "Printing in progress ..."
set devi to prin
do whil .not. eof()
  stor 'Y' to line,sub_t
  stor 0 to car,house,p_bank,furniture
  stor 'N' to c_choice,h_choice,p_choice,f_choice
  do tr_pay
  sele b
  seek mdept_code
  dept=dept_name
  sele c
  seek mbank_code
  bank=bank_name
  do seek_pay
  sele f
  do loan_rec
  peg=5
  if c_choice='Y'
    peg=peg+1
  endi
  if h_choice='Y'
    peg=peg+1
  endi
  if p_choice='Y'
    peg=peg+1
  endi
  if f_choice='Y'
    peg=peg+1
  endi
  if peg>5
```

```
    line='N'
    sub_t='N'
  endi
  @ r,1 say "AGRICULTURAL DEVELOPMENT PROJECT - MINNA"
  @ r,53 say "PAYSLIP FOR:"
  @ r,66 say upper(rtrim(cur_mth))+', '+ltrim(str(cur_year))
  r=r+1
  @ r,1 say repl("=",80)
  r=r+1
  @ r,1 say "DEPARTMENT: "+dept
  r=r+1
  @ r,1 say "BANK      : "+bank
  r=r+1
  @ r,1 say "STAFF NO  : "+ms_numb
  @ r,40 say "STAFF NAME: "+rtrim(msurname)+', '+rtrim(mf_name)+'
'+minitial
  r=r+1
  @ r,1 say repl("=",80)
  r=r+1
  @ r,4 say "PAYMENTS"
  @ r,17 say "AMOUNT"
  @ r,27 say "DEDUCTIONS"
  @ r,41 say "AMOUNT"
  @ r,52 say "PAYMENT HISTORY"
  @ r,74 say "AMOUNT"
  r=r+1
  @ r,1 say repl("-",80)
  r=r+1
  @ r,1 say "BASIC SALARY"
  @ r,16 say mbasic_sal pict '9999.99'
  @ r,25 say "TAX"
  @ r,40 say mtax pict '9999.99'
  @ r,49 say "ANNUAL FREEPAY"
  @ r,73 say ma_fpay pict '99999.99'
  stor ' ' to tag1,tag2
  stor 0 to cnt
  do whil .t.
    r=r+1
    cnt=cnt+1
    if cnt=1
      @ r,49 say "TAXABLE GROSS TO DATE"
      @ r,73 say mtax_gr pict '99999.99'
      loop
    endi
    if cnt=2
      @ r,49 say "FREEPAY TO DATE"
```

```
      @ r,73 say macc_fpay pict '99999.99'
      loop
    endi
    if cnt=3
      @ r,49 say "TAXABLE PAY TO DATE"
      @ r,73 say mtax_pay pict '99999.99'
      loop
    endi
    if cnt=4
      @ r,49 say "TAX DUE TO DATE"
      @ r,73 say macc_tax pict '99999.99'
      loop
    endi
    if cnt=5
      @ r,49 say "TAX PAID SINCE JANUARY"
      @ r,73 say mtax_jan pict '99999.99'
      loop
    endi
    if line='N'
      line='Y'
      loop
    endi
    if sub_t='N'
      @ r,49 say 'LOAN DETAILS:'
      sub_t='Y'
      loop
    endi
    if c_choice='Y'
      @ r,49 say "MONTHS REMAIN(C/REF. LOAN) - "+ltrim(str(car))
      c_choice='N'
      loop
    endi
    if h_choice='Y'
      @ r,49 say "MONTHS REMAIN(HOUS. LOAN) -
"+ltrim(str(house))
      h_choice='N'
      loop
    endi
    if p_choice='Y'
      @ r,49 say "MONTHS REMAIN(PL's B/LOAN) -
"+ltrim(str(p_bank))
      p_choice='N'
      loop
    endi
    if f_choice='Y'
      @ r,49 say "MONTHS REMAIN(FURNI. LOAN) -
```

```
"+ltrim(str(furniture))
      f_choice='N'
      loop
    endi
    if tag1='Y' .and. tag2='Y' .and. cnt>=peg
      exit
    endi
  endd
  part=part+1
  if part=1
    r=23
  else
    r=56
  endi
  @ r,1 say "T O T A L :-"
  @ r,14 say mgr_pay pict '99,999.99'
  @ r,38 say mtot_ded pict '99,999.99'
  r=r+1
  @ r,1 say repl("=",80)
  r=r+1
  @ r,48 say "N E T - P A Y ="
  @ r,69 say mnet_pay pict '99,999.99'
  r=r+1
  @ r,1 say repl("=",80)
  if part=1
    r=r+4
    @ r,1 say repl("=",80)
    r=r+4
  else
    r=1
    part=0
  endif
  sele a
  skip
endd
eject
set devi to scre
@ 14,5 clea to 18,32
@ 15,7 say "PRINTING IS COMPLETED !!!"
@ 17,11 say "Press any key ..."
set cons off
wait
set cons on
@ 15,7 clea to 17,32
clos data
eras smonthly.dbf
```

```
retu


SCHEDULE.PRG
set colo to n/w*
@ 14,39 say "[B]ank Schedule"
set colo to
@ 13,7 to 20,31 doub
@ 14,9 say "Switch Printer On"
@ 15,9 say "Be sure it is ready."
@ 18,8 say 'PRESS: "C" to CONTINUE'
@ 19,8 say '        "S" to STOP ...'
p=0
do whil p=0
  p=inkey()
  if upper(chr(p)) $ "CS"
    exit
  endi
  p=0
endd
@ 13,7 clea to 20,31
if upper(chr(p)) $ "S"
  retu
endi
do whil .t.
  exitting='N'
  @ 13,10 to 19,27
  @ 14,12 say "To Print  ll"
  @ 15,12 say "To Print  art"
  @ 16,12 say "To  xit"
  set inte on
  set colo to r
  @ 14,21 say "A"
  @ 15,21 say "P"
  @ 16,15 say "E"
  set colo to
  set inte off
  @ 18,11 say "Pick Choice ..."
  p=0
  do whil p=0
    p=inkey()
    if upper(chr(p)) $ "APE"
      exit
    endi
    p=0
  endd
```

```
if upper(chr(p)) $ "E"
  @ 13,10 clea to 19,27
  clea all
  retu
endi
if upper(chr(p)) $ "P"
  @ 14,21 say "A"
  @ 15,21 say "P"
  @ 16,15 say "E"
  set colo to n/w
  @ 15,12 say "To Print Part"
  set colo to
  use monthly
  @ 21,2 say "Enter BANK CODE to start from:"
  @ 23,24 say 'EXIT = "99"'
  @ 20,1 to 22,37
  num=0
  do whil .t.
    @ 21,32 get num pict '99'
    read
    @ 23,23 clea to 23,37
    if num=99
      use
      exitting='Y'
      @ 13,1 clea to 22,37
      exit
    endi
    loca for bank_code=num
    if .not. found()
      @ 23,7 say "BANK CODE not available - Press any key ..."
      set cons off
      wait
      set cons on
      num=0
      @ 23,5 say spac(50)
      @ 23,24 say 'EXIT = "99"'
      loop
    endi
    use
    @ 20,1 clea to 22,37
    exit
  endd
endi
if exitting='Y'
  loop
else
```

```
    exit
  endi
endd
@ 13,10 clea to 19,27
@ 16,6 say "WAIT - Preparing files ..."
use current
cur_mth=cmonth(c_date)
cur_year=year(c_date)
title="***** BANK SCHEDULE FOR THE MONTH "+upper(cur_mth)+",
"+ltrim(str(cur_year))+" *****"
n1=len(title)
n2=int((132-n1)/2)
use monthly
sort on bank_code,dept_code to bmonth
use
sele a
  use bmonth
sele b
  use bank index bank
sele c
  use dept index dept
sele a
if upper(chr(p)) $ "P"
  loca for bank_code=num
endi
mbank_code=bank_code
mdept_code=dept_code
sele b
seek mbank_code
bank=rtrim(bank_name)
sele c
seek mdept_code
dept=dept_name
@ 16,6 say spac(28)
@ 14,5 to 18,32 doub
@ 15,10 say "K E E P  O F F  !!!"
@ 17,7 say "Printing in progress ..."
set devi to prin
pg=1
DO TITLE_B
r=11
stor 0 to sno,stot,gtot
sele a
do whil .not. eof()
  name=rtrim(surname)+' '+rtrim(f_name)
  if o_names<>spac(15)
```

```
  mname=rtrim(o_names)
  name=name+' '+substr(mname,1)+'.'
endi
name=name+' '+rtrim(initial)
mbank_no=bank_no
mnet_pay=net_pay
stot=stot+mnet_pay
gtot=gtot+mnet_pay
sno=sno+1
r=r+1
@ r,17 say sno pict '9999'
@ r,24 say name
@ r,60 say dept
@ r,93 say mbank_no
@ r,107 say mnet_pay pict '99,999.99'
sele a
skip
IF EOF()
  EXIT
ENDI
if mdept_code<>dept_code
  mdept_code=dept_code
  sele c
  seek mdept_code
  dept=dept_name
  sele a
endi
if mbank_code<>bank_code
  mbank_code=bank_code
  r=r+1
  @ R,17 SAY REPL("-",99)
  R=R+1
  @ R,30 SAY "G R A N D - T O T A L : -"
  @ r,104 say gtot pict '9,999,999.99'
  R=R+1
  @ R,17 SAY REPL("=",99)
  stor 0 to sno,stot,gtot
  sele b
  seek mbank_code
  bank=rtrim(bank_name)
  pg=1
  eject
  DO TITLE_b
  R=11
  sele a
endi
```

```
  if r>54
    r=r+1
    @ R,17 SAY REPL("-",99)
    R=R+1
    @ R,30 SAY "S U B - T O T A L : -"
    @ r,104 say stot pict '9,999,999.99'
    R=R+1
    @ R,17 SAY REPL("=",99)
    stor 0 to stot
    bank=bank+' '+'Continued'
    pg=pg+1
    eject
    DO TITLE_b
    r=11
  endi
endd
IF EOF()
  R=R+1
  @ R,17 SAY REPL("-",99)
  R=R+1
  @ R,30 SAY "G R A N D - T O T A L : -"
  @ R,104 SAY gtot pict '9,999,999.99'
  R=R+1
  @ R,17 SAY REPL("=",99)
ENDI
eject
set devi to scre
@ 13,5 clea to 20,32
@ 15,7 say "PRINTING IS COMPLETED !!!"
@ 17,11 say "Press any key ..."
set cons off
wait
set cons on
@ 15,7 clea to 17,32
clos all
clea all
erase smonthly.dbf
retu


TAX_REP.PRG
use monthly
sort on dept_code,s_numb to tax.dbf
use current
cur_mth=cmonth(c_date)
cur_year=year(c_date)
```

```
use
stor 0 to sno,stot,gtot
sele 1
  use tax
sele 2
  use dept index dept
title2="***** TAX REPORT FOR THE MONTH OF
"+upper(rtrim(cur_mth))+", "+ltrim(str(cur_year))+" *****"
n1=len(title2)
n2=int((80-n1)/2)
set devi to prin
DO TITLE_t
r=6
sele 1
mdept_code=dept_code
sele 2
seek mdept_code
mdept_name=rtrim(dept_name)
sele 1
do whil .not. eof()
  name=rtrim(surname)+' '+initial
  ms_numb=s_numb
  mtax=tax
  if mtax=0.00
    skip
    loop
  endi
  stot=stot+mtax
  gtot=gtot+mtax
  sno=sno+1
  r=r+1
  @ r,2 say sno pict '9999'
  @ r,10 say ms_numb pict '99999'
  @ r,20 say name
  @ r,43 say mdept_name
  @ r,73 say mtax pict '999.99'
  skip
  if mdept_code<>dept_code
    mdept_code=dept_code
    sele 2
    seek mdept_code
    mdept_name=rtrim(dept_name)
    sele 1
  endi
  if r>49
    r=r+1
```

```
    @ R,2 SAY REPL("-",77)
    R=R+1
    @ R,9 SAY "S U B - T O T A L : -"
    @ r,70 say stot pict '99,999.99'
    R=R+1
    @ R,2 SAY REPL("=",77)
    eject
    DO TITLE_t
    R=6
  endi
enddo
IF r<50
  R=R+3
else
  eject
  r=1
endif
@ R,9 SAY "G R A N D - T O T A L : -"
@ R,67 SAY gtot PICT '9,999,999.99'
R=R+1
@ R,2 SAY REPL("=",77)
eject
set devi to scre
clos all
RETU


DSUMMARY.PRG
set colo to n/w*
@ 16,39 say "[D]epartmental Summary"
set colo to
@ 13,7 to 20,31 doub
@ 14,9 say "Switch Printer On"
@ 15,9 say "Be sure it is ready."
@ 18,8 say 'PRESS: "C" to CONTINUE'
@ 19,8 say '        "S" to STOP ...'
p=0
do whil p=0
  p=inkey()
  if upper(chr(p)) $ "CS"
    exit
  endi
  p=0
endd
@ 13,7 clea to 20,31
if upper(chr(p)) $ "S"
```

```
  retu
endi
do whil .t.
  exitting='N'
  @ 13,10 to 19,27
  @ 14,12 say "To Print  ll"
  @ 15,12 say "To Print  art"
  @ 16,12 say "To  xit"
*  set inte on
  set colo to r
  @ 14,21 say "A"
  @ 15,21 say "P"
  @ 16,15 say "E"
  set colo to
*  set inte off
  @ 18,11 say "Pick Choice ..."
  p=0
  do whil p=0
    p=inkey()
    if upper(chr(p)) $ "APE"
      exit
    endi
    p=0
  endd
  if upper(chr(p)) $ "E"
    @ 13,10 clea to 19,27
    clea all
    retu
  endi
  if upper(chr(p)) $ "P"
    @ 14,21 say "A"
    @ 15,21 say "P"
    @ 16,15 say "E"
    set colo to n/w
    @ 15,12 say "To Print Part"
    set colo to
    use dept index dept
    @ 21,2 say "Enter DEPT. CODE to start from:"
    @ 23,25 say 'EXIT = "99"'
    @ 20,1 to 22,37
    do whil .t.
      num=0
      @ 21,34 get num pict '99'
      read
      @ 23,23 clea to 23,37
      if num=99
```

```
        use
        exitting='Y'
        @ 13,1 clea to 22,37
        exit
      endi
      seek num
      if .not. found()
        @ 23,6 say "Invalid DEPT. CODE entered - Press any key
..."
        set cons off
        wait
        set cons on
        num=0
        @ 23,5 say spac(50)
        @ 23,25 say 'EXIT = "99"'
        loop
      endi
      use
      @ 20,1 clea to 22,37
      exit
    endd
  endi
  if exitting='Y'
    loop
  else
    exit
  endi
endd
@ 13,10 clea to 19,27
@ 16,6 say "WAIT - Preparing files ..."
use current
cur_mth=cmonth(c_date)
cur_year=year(c_date)
title="***** DEPARTMENTAL SUMMARY FOR THE MONTH OF
"+upper(cur_mth)+", "+ltrim(str(cur_year))+" *****"
n1=len(title)
n2=int((80-n1)/2)
use empty1
if .not. eof()
  zap
endi
use empty2
if .not. eof()
  zap
endi
use monthly
```

```
index on dept_code to dsummary
use
sele a
  use monthly index dsummary
sele b
  use dept index dept
sele c
  use allce index allce
sele d
  use deduct index deduct
sele e
  use empty1
sele f
  use empty2
p1='N'
if upper(chr(p)) $ 'P'
  p1='Y'
  @ 16,3 say 'Making necessary computations ...'
else
  @ 16,6 say spac(28)
  @ 14,5 to 18,32 doub
  @ 15,10 say "K E E P  O F F  !!!"
  @ 17,7 say "Printing in progress ..."
  set devi to prin
endi
desc=spac(15)
stor 0 to ttal,tded,tcnt,code
sele a
mdept_code=dept_code
sele b
seek mdept_code
do whil .not. eof()
*  rest from all_mem addi
  mdept_code=dept_code
  dept=dept_name
  stor 0 to tal,ded,cnt,a_amt,d_amt,m_sal,m_tax,mtax_ref
  sele a
  go top
  set filter to dept_code=mdept_code
  do whil .not. eof()
    m_sal=m_sal+basic_sal
    m_tax=m_tax+tax
    mtax_ref=mtax_ref+tax_ref
    cnt=cnt+1
    do calc_ds1
    do calc_ds2
```

```
  sele a
  skip
endd
if cnt=0
  sele b
  skip
  loop
endi
tcnt=tcnt+cnt
tal=tal+m_sal+mtax_ref
ttal=ttal+tal
ded=ded+m_tax
tded=tded+ded
netpay=tal-ded
if p1='Y'
  if mdept_code>=num
    p1='N'
    @ 16,3 say spac(34)
    @ 14,5 to 18,32 doub
    @ 15,10 say "K E E P  O F F  !!!"
    @ 17,7 say "Printing in progress ..."
    set devi to prin
  else
    sele e
    zap
    sele f
    zap
    sele b
    skip
    loop
  endi
endi
sele e
go top
sele f
go top
if r>30
  do title_d2
  r=8
else
  r=r+5
  @ r,1 say repl("=",80)
  r=r+5
  @ r,12 say "DEPARTMENT: "+dept
  r=r+2
  @ r,11 say "ALLOWANCES"
```

```
  @ r,30 say "AMOUNT"
  @ r,43 say "DEDUCTIONS"
  @ r,62 say "AMOUNT"
  r=r+1
  @ r,11 say repl("-",58)
  r=r+1
endi
@ r,11 say "BASIC SALARY"
@ r,27 say m_sal pict '999,999.99'
@ r,43 say "TAX"
@ r,60 say m_tax pict '99,999.99'
stor spac(1) to e1,e2
do whil .t.
  r=r+1
  if mtax_ref<>0
    @ r,11 say "TAX REFUND"
    @ r,28 say mtax_ref pict '99,999.99'
    mtax_ref=0
  else
    sele e
    if .not. eof()
      @ r,11 say a_desc
      @ r,28 say amt pict '99,999.99'
      SKIP
    else
      e1='Y'
    endi
  endi
  sele f
  if .not. eof()
    @ r,43 say d_desc
    @ r,60 say amt pict '99,999.99'
    SKIP
  else
    e2='Y'
  endi
  if e1='Y' .and. e2='Y'
    sele e
    zap
    sele f
    zap
    exit
  endi
endd
r=r+2
@ r,11 say "TOTAL STAFF FOR UNIT/DEPARTMENT     :
```

```
"+ltrim(str(cnt))
  r=r+1
  @ r,11 say "TOTAL GROSS PAY FOR UNIT/DEPARTMENT : #"
  @ r,50 say tal pict '@b 9,999,999.99'
  r=r+1
  @ r,11 say "TOTAL DEDUCTIONS FOR UNIT/DEPARTMENT: #"
  @ r,50 say ded pict '@b 999,999.99'
  r=r+1
  @ r,11 say "TOTAL NET PAY FOR UNIT/DEPARTMENT   : #"
  @ r,50 say netpay pict '@b 9,999,999.99'
  sele b
  skip
endd
tnetpay=ttal-tded
eject
r=5
@ r,14 say "***** G R A N D  T O T A L S *****"
r=r+1
@ r,19 say repl("*",24)
r=r+2
@ r,11 say "T O T A L  S T A F F  P A I D   : "+ltrim(str(tcnt))
r=r+3
@ r,11 say "T O T A L  G R O S S  P A Y     : #"
@ r,46 say ttal pict '@b 999,999,999.99'
r=r+3
@ r,11 say "T O T A L  D E D U C T I O N S  : #"
@ r,46 say tded pict '@b 99,999,999.99'
r=r+3
@ r,11 say "T O T A L  N E T  P A Y         : #"
@ r,46 say tnetpay pict '@b 999,999,999.99'
eject
set devi to scre
@ 14,5 clea to 18,32
@ 15,7 say "PRINTING IS COMPLETED !!!"
@ 17,11 say "Press any key ..."
set cons off
wait
set cons on
@ 15,7 clea to 17,32
clos data
erase smonthly.dbf
retu


DEDUCT.PRG
set colo to n/w*
```

```
@ 17,39 say "D[E]duction List"
set colo to
@ 13,7 to 20,31 doub
@ 14,9 say "Switch Printer On"
@ 15,9 say "Be sure it is ready."
@ 18,8 say 'PRESS: "C" to CONTINUE'
@ 19,8 say '        "S" to STOP ...'
p=0
do whil p=0
  p=inkey()
  if upper(chr(p)) $ "CS"
    exit
  endi
  p=0
endd
@ 13,7 clea to 20,31
if upper(chr(p)) $ "S"
  retu
endi
do whil .t.
  exitting='N'
  @ 13,10 to 19,27
  @ 14,12 say "To Print  ll"
  @ 15,12 say "To Print  art"
  @ 16,12 say "To  xit"
  set inte on
  set colo to r
  @ 14,21 say "A"
  @ 15,21 say "P"
  @ 16,15 say "E"
  set colo to
  set inte off
  @ 18,11 say "Pick Choice ..."
  p=0
  do whil p=0
    p=inkey()
    if upper(chr(p)) $ "APE"
      exit
    endi
    p=0
  endd
  if upper(chr(p)) $ "E"
    @ 13,10 clea to 19,27
    clea all
    retu
  endi
```

```
  if upper(chr(p)) $ "P"
    @ 14,21 say "A"
    @ 15,21 say "P"
    @ 16,15 say "E"
    set colo to n/w
    @ 15,12 say "To Print Part"
    set colo to
    use deduct index deduct
    @ 21,2 say "Enter DEDUCT. CODE to start from:"
    @ 23,25 say 'EXIT = "99"'
    @ 20,1 to 22,37
    num=0
    do whil .t.
      @ 21,35 get num pict '99'
      read
      @ 23,23 clea to 23,37
      if num=99
        use
        exitting='Y'
        @ 13,1 clea to 22,37
        exit
      endi
      seek num
      if .not. found()
        @ 23,6 say "DEDUCT. CODE not available - Press any key
..."
        set cons off
        wait
        set cons on
        num=0
        @ 23,5 say spac(50)
        @ 23,25 say 'EXIT = "99"'
        loop
      endi
      use
      @ 20,1 clea to 22,37
      exit
    endd
  endi
  if exitting='Y'
    loop
  else
    exit
  endi
endd
@ 13,10 clea to 19,27
```

```
@ 16,6 say "WAIT - Preparing files ..."
use current
cur_mth=cmonth(c_date)
cur_year=year(c_date)
use monthly
sort on dept_code,s_numb to staff
*index on dept_code+s_numb to staff
use monthly2
index on s_numb to monthly2
use current
cur_mth=cmonth(c_date)
cur_year=year(c_date)
use
sele 1
  use staff
*  use monthly index staff
sele 2
  use monthly2 index monthly2
sele 3
  use deduct index deduct
sele 4
  use dept index dept
r=0
title='N'
sele 3
if upper(chr(p)) $ "P"
  seek num
endi
@ 16,6 say spac(28)
@ 14,5 to 18,32 doub
@ 15,10 say "K E E P  O F F  !!!"
@ 17,7 say "Printing in progress ..."
set devi to prin
do whil .not. eof()
  stor 0 to tot,sno
  md_code=d_code
  m=d_code
  md_desc=d_desc
*  md_desc=rtrim(d_desc)
  if md_desc=space(15)
    skip
    loop
  endi
  md_desc=rtrim(md_desc)
  md_desc="***** "+md_desc+" DEDUCTION FOR THE MONTH OF
"+upper(cur_mth)+", "+ltrim(str(cur_year))+" *****"
```

```
  n1=len(md_desc)
  n2=int((80-n1)/2)
*  do title_d
*  r=7
  sele 1
  go top
  set filt to d_code1=m .or. d_code2=m .or. d_code3=m .or.
d_code4=m;
  .or. d_code5=m .or. d_code6=m .or. d_code7=m .or. d_code8=m;
  .or. d_code9=m .or. d_code10=m .or. d_code11=m .or. d_code12=m
  do whil .not. eof()
    ms_numb=s_numb
    name=rtrim(surname)+', '+rtrim(f_name)+' '+initial
    mdept_code=dept_code
    md_code1=d_code1
    md_sign1=d_sign1
    md_amt1=d_amt1
    md_code2=d_code2
    md_sign2=d_sign2
    md_amt2=d_amt2
    md_code3=d_code3
    md_sign3=d_sign3
    md_amt3=d_amt3
    md_code4=d_code4
    md_sign4=d_sign4
    md_amt4=d_amt4
    md_code5=d_code5
    md_sign5=d_sign5
    md_amt5=d_amt5
    md_code6=d_code6
    md_sign6=d_sign6
    md_amt6=d_amt6
    md_code7=d_code7
    md_sign7=d_sign7
    md_amt7=d_amt7
    md_code8=d_code8
    md_sign8=d_sign8
    md_amt8=d_amt8
    md_code9=d_code9
    md_sign9=d_sign9
    md_amt9=d_amt9
    md_code10=d_code10
    md_sign10=d_sign10
    md_amt10=d_amt10
    md_code11=d_code11
    md_sign11=d_sign11
```

```
md_amt11=d_amt11
md_code12=d_code12
md_sign12=d_sign12
md_amt12=d_amt12
sele 4
seek mdept_code
mdept_name=dept_name
sele 2
seek ms_numb
if md_code1=md_code .and. md_sign1="-"
  sno=sno+1
  amount=md_amt1
  cnt=d_cnt1
  dur=d_dur1
  if sno=1 .or. title='Y'
    do title_d
    r=7
    title='N'
  endi
  do ded_prn1
endi
if md_code2=md_code .and. md_sign2="-"
  sno=sno+1
  amount=md_amt2
  cnt=d_cnt2
  dur=d_dur2
  if sno=1 .or. title='Y'
    do title_d
    r=7
    title='N'
  endi
  do ded_prn1
endi
if md_code3=md_code .and. md_sign3="-"
  sno=sno+1
  amount=md_amt3
  cnt=d_cnt3
  dur=d_dur3
  if sno=1 .or. title='Y'
    do title_d
    r=7
    title='N'
  endi
  do ded_prn1
endi
if md_code4=md_code .and. md_sign4="-"
```

```
      sno=sno+1
      amount=md_amt4
*      cnt=d_cnt4
      dur=d_dur4
      cnt=d_cnt4
      if sno=1 .or. title='Y'
        do title_d
        r=7
        title='N'
      endi
      do ded_prn1
    endi
    if md_code5=md_code .and. md_sign5="-"
      sno=sno+1
      amount=md_amt5
      cnt=d_cnt5
      dur=d_dur5
      if sno=1 .or. title='Y'
        do title_d
        r=7
        title='N'
      endi
      do ded_prn1
    endi
    if md_code6=md_code .and. md_sign6="-"
      sno=sno+1
      amount=md_amt6
      cnt=d_cnt6
      dur=d_dur6
      if sno=1 .or. title='Y'
        do title_d
        r=7
        title='N'
      endi
      do ded_prn1
    endi
    if md_code7=md_code .and. md_sign7="-"
      sno=sno+1
      amount=md_amt7
      cnt=d_cnt7
      dur=d_dur7
      if sno=1 .or. title='Y'
        do title_d
        r=7
        title='N'
      endi
```

```
  do ded_prn1
endi
if md_code8=md_code .and. md_sign8="-"
  sno=sno+1
  amount=md_amt8
  cnt=d_cnt8
  dur=d_dur8
  if sno=1 .or. title='Y'
    do title_d
    r=7
    title='N'
  endi
  do ded_prn1
endi
if md_code9=md_code .and. md_sign9="-"
  sno=sno+1
  amount=md_amt9
  cnt=d_cnt9
  dur=d_dur9
  if sno=1 .or. title='Y'
    do title_d
    r=7
    title='N'
  endi
  do ded_prn1
endi
if md_code10=md_code .and. md_sign10="-"
  sno=sno+1
  amount=md_amt10
  cnt=d_cnt10
  dur=d_dur10
  if sno=1 .or. title='Y'
    do title_d
    r=7
    title='N'
  endi
  do ded_prn1
endi
if md_code11=md_code .and. md_sign11="-"
  sno=sno+1
  amount=md_amt11
  cnt=d_cnt11
  dur=d_dur11
  if sno=1 .or. title='Y'
    do title_d
    r=7
```

```
        title='N'
      endi
      do ded_prn1
    endi
    if md_code12=md_code .and. md_sign12="-"
      sno=sno+1
      amount=md_amt12
      cnt=d_cnt12
      dur=d_dur12
      if sno=1 .or. title='Y'
        do title_d
        r=7
        title='N'
      endi
      do ded_prn1
    endi
    sele 1
    skip
    if r>50
      r=r+1
      @ r,1 say repl ("-",80)
      r=r+1
      @ r,20 say "S U B - T O T A L : -"
      @ r,54 say tot pict '9,999,999.99'
      r=r+1
      @ r,1 say repl('=',80)
      eject
      title='Y'
    endi
  endd
  if tot<>0
    r=r+1
    @ r,1 say repl ("-",80)
    r=r+1
    @ r,20 say "G R A N D - T O T A L : -"
    @ r,53 say tot pict '99,999,999.99'
    r=r+1
    @ r,1 say repl('=',80)
    tot=0
    eject
  endi
  sele 3
  skip
endd
set devi to scre
@ 14,5 clea to 18,32
```

```
@ 15,7 say "PRINTING IS COMPLETED !!!"
@ 17,11 say "Press any key ..."
set cons off
wait
set cons on
@ 15,7 clea to 17,32
clos data
erase staff.dbf
retu


STAFF.PRG
set colo to n/w*
@ 18,39 say "[S]taff List"
set colo to
@ 13,7 to 20,31 doub
@ 14,9 say "Switch Printer On"
@ 15,9 say "Be sure it is ready."
@ 18,8 say 'PRESS: "C" to CONTINUE'
@ 19,8 say '        "S" to STOP ...'
p=0
do whil p=0
  p=inkey()
  if upper(chr(p)) $ "CS"
    exit
  endi
  p=0
endd
@ 13,7 clea to 20,31
if upper(chr(p)) $ "S"
  retu
endi
do whil .t.
  exitting='N'
  @ 13,10 to 19,27
  @ 14,12 say "To Print  ll"
  @ 15,12 say "To Print  art"
  @ 16,12 say "To  xit"
  set inte on
  set colo to r
  @ 14,21 say "A"
  @ 15,21 say "P"
  @ 16,15 say "E"
  set colo to
  set inte off
  @ 18,11 say "Pick Choice ..."
```

```
p=0
do whil p=0
  p=inkey()
  if upper(chr(p)) $ "APE"
    exit
  endi
  p=0
endd
if upper(chr(p)) $ "E"
  @ 13,10 clea to 19,27
  clea all
  retu
endi
if upper(chr(p)) $ "P"
  @ 14,21 say "A"
  @ 15,21 say "P"
  @ 16,15 say "E"
  set colo to n/w
  @ 15,12 say "To Print Part"
  set colo to
  use dept index dept
  @ 21,2 say "Enter DEPT. CODE to start from:"
  @ 23,25 say 'EXIT = "99"'
  @ 20,1 to 22,37
  num=0
  do whil .t.
    num=0
    @ 21,34 get num pict '99'
    read
    @ 23,23 clea to 23,37
    if num=99
      use
      exitting='Y'
      @ 13,1 clea to 22,37
      exit
    endi
    seek num
    if .not. found()
      @ 23,6 say "Invalid DEPT. CODE entered - Press any key
..."
      set cons off
      wait
      set cons on
      num=0
      @ 23,5 say spac(50)
      @ 23,25 say 'EXIT = "99"'
```

```
        loop
      endi
      use
      @ 20,1 clea to 22,37
      exit
    endd
  endi
  if exitting='Y'
    loop
  else
    exit
  endi
endd
@ 13,10 clea to 19,27
@ 16,6 say "WAIT - Preparing files ..."
use current
cur_mth=cmonth(c_date)
cur_year=year(c_date)
title="***** PAYROLL STAFF LIST FOR THE MONTH OF
"+upper(cur_mth)+", "+ltrim(str(cur_year))+" *****"
n1=len(title)
n2=int((80-n1)/2)
use staff1
sort on dept_code,s_numb to staff
*index on dept_code+val(s_numb) to staff
use
sele 1
  use staff
*  use monthly index staff
sele 2
  use rank index rank
sele 3
  use dept index dept
stor 0 to sno
p1='N'
sele 1
if upper(chr(p)) $ 'P'
  p1='Y'
  @ 16,6 say spac(26)
  @ 16,8 say 'Making computation ...'
else
  mrank_code=rank_code
  mdept_code=dept_code
  sele 2
  seek mrank_code
  mrank_desc=rank_desc
```

```
  sele 3
  seek mdept_code
  mdept_name=dept_name
  @ 16,6 say spac(28)
  @ 14,5 to 18,32 doub
  @ 15,10 say "K E E P  O F F  !!!"
  @ 17,7 say "Printing in progress ..."
  set devi to prin
  do title_s
  r=6
  sele 1
endi
tsno=0
do whil .not. eof()
  tsno=tsno+1
  if p1='Y'
    mdept_code=dept_code
    if mdept_code>=num
      p1='N'
      mrank_code=rank_code
      sele 2
      seek mrank_code
      mrank_desc=rank_desc
      sele 3
      seek mdept_code
      mdept_name=dept_name
      @ 16,6 say spac(28)
      @ 14,5 to 18,32 doub
      @ 15,10 say "K E E P  O F F  !!!"
      @ 17,7 say "Printing in progress ..."
      set devi to prin
      do title_s
      r=6
      sele 1
    else
      skip
      loop
    endi
  endi
  ms_numb=s_numb
  name=rtrim(surname)+' '+rtrim(f_name)+' '+rtrim(initial)
  msal_l=ltrim(str(sal_l))
  if len(msal_l)<2
    msal_l='0'+msal_l
  endi
  grade=msal_l+"/"+ltrim(str(sal_s))
```

```
   ma_sal=a_sal
   r=r+2
   sno=sno+1
   @ r,1 say sno pict '9999'
   @ r,8 say ms_numb pict '99999'
   @ r,17 say name
   @ r,42 say grade
   @ r,51 say mrank_desc
   @ r,70 say ma_sal pict '99,999.99'
   skip
   if eof()
     r=r+1
     @ r,1 say repl('-',80)
     r=r+2
     @ r,1 say "TOTAL NO OF STAFF IN THE DEPT = "+ltrim(str(sno))
*     eject
     exit
   endi
   if mrank_code<>rank_code
     mrank_code=rank_code
     sele 2
     seek mrank_code
     mrank_desc=rank_desc
     sele 1
   endi
   if mdept_code <> dept_code
     r=r+1
     @ r,1 say repl('-',80)
     r=r+2
     @ r,1 say "TOTAL NO OF STAFF IN THE DEPT = "+ltrim(str(sno))
     mdept_code=dept_code
     sele 3
     seek mdept_code
     mdept_name = dept_name
     eject
     do title_s
     r=6
     sele 1
     sno=0
   endi
   if r>52
     mdept_name=mdept_name+" continued"
     eject
     do title_s
     r=6
   endi
```

```
endd
r=r+3
ejec
r=5
@ r,1 say repl("=",80)
r=r+1
@ r,1 say "TOTAL NO OF STAFF LISTED = "+ltrim(str(tsno))
r=r+1
@ r,1 say repl("=",80)
ejec
set devi to scre
@ 13,5 clea to 20,32
@ 15,7 say "PRINTING IS COMPLETED !!!"
@ 17,11 say "Press any key ..."
set cons off
wait
set cons on
@ 15,7 clea to 17,32
clos data
retu


F_UPDATE.PRG
do whil .t.
  @ 4,19 clea to 6,60
  @ 4,30 to 6,48 doub
  @ 5,31 say "FILES UPDATE MENU"
  set colo to n/w*
  @ 9,52 say "FILES "
  @ 10,52 say "UPDATE"
  set colo to
  @ 12,51 to 23,75
  @ 13,52 say "[B]ank File"
  @ 14,52 say "[R]ank File"
  @ 15,52 say "[A]llowance File"
  @ 16,52 say "[D]eduction File"
  @ 17,52 say "D[E]partment File"
  @ 18,52 say "[M]arital Status File"
  @ 19,52 say "A[P]pointment Type File"
  @ 20,52 say "[Q] U I T"
  @ 22,53 say "Pick your choice ..."
  f=0
  do whil f=0
    f=inkey()
    if upper(chr(f)) $ "BRADEMPQ"
      exit
```

```
  endif
  f=0
endd
@ 22,53 say spac(20)
@ 4,29 to 6,50 doub
@ 5,30 say "FILES UPDATE SUBMENU"
do case
  case upper(chr(f)) $ "B"
    set colo to n/w*
    @ 13,52 say "[B]ank File"
    set colo to
    @ 13,8 say "You are about to update BANK FILE."
    @ 15,8 say "Do you want to continue (Y/N) ..."
    fa=0
    do whil fa=0
      fa=inkey()
      if upper(chr(fa)) $ "YN"
        exit
      endi
      fa=0
    enddo
    @ 13,7 clea to 15,48
    if upper(chr(fa)) $ "N"
      loop
    endi
    do bank
  case upper(chr(f)) $ "R"
    set colo to n/w*
    @ 14,52 say "[R]ank File"
    set colo to
    @ 14,8 say "You are about to update RANK FILE."
    @ 16,8 say "Do you want to continue (Y/N) ..."
    fb=0
    do whil fb=0
      fb=inkey()
      if upper(chr(fb)) $ "YN"
        exit
      endi
      fb=0
    enddo
    @ 14,7 clea to 16,48
    if upper(chr(fb)) $ "N"
      loop
    endi
    do rank
  case upper(chr(f)) $ "A"
```

```
    set colo to n/w*
    @ 15,52 say "[A]llowance File"
    set colo to
    @ 15,6 say "You are about to update ALLOWANCE FILE."
    @ 17,8 say "Do you want to continue (Y/N) ..."
    fc=0
    do whil fc=0
      fc=inkey()
      if upper(chr(fc)) $ "YN"
        exit
      endi
      fc=0
    endd
    @ 15,6 clea to 17,48
    if upper(chr(fc)) $ "N"
      loop
    endi
    do all
  case upper(chr(f)) $ "D"
    set colo to n/w*
    @ 16,52 say "Deduction File"
    set colo to
    @ 16,6 say "You are about to update DEDUCTION FILE."
    @ 18,8 say "Do you want to continue (Y/N) ..."
    fd=0
    do whil fd=0
      fd=inkey()
      if upper(chr(fd)) $ "YN"
        exit
      endi
      fd=0
    endd
    @ 16,6 clea to 18,48
    if upper(chr(fd)) $ "N"
      loop
    endi
    do ded
  case upper(chr(f)) $ "E"
    set colo to n/w*
    @ 17,52 say "Department File"
    set colo to
    @ 17,5 say "You are about to update DEPARTMENT FILE."
    @ 19,8 say "Do you want to continue (Y/N) ..."
    fe=0
    do whil fe=0
      fe=inkey()
```

```
    if upper(chr(fe)) $ "YN"
      exit
    endi
    fe=0
  enddo
  @ 17,5 clea to 19,50
  if upper(chr(fe)) $ "N"
    loop
  endi
  do dept
case upper(chr(f)) $ "M"
  set colo to n/w*
  @ 18,52 say "Marital Status File"
  set colo to
  @ 18,3 say "You are about to update MARITAL STATUS FILE."
  @ 20,8 say "Do you want to continue (Y/N) ..."
  ff=0
  do whil ff=0
    ff=inkey()
    if upper(chr(ff)) $ "YN"
      exit
    endi
    ff=0
  endd
  @ 18,3 clea to 20,50
  if upper(chr(ff)) $ "N"
    loop
  endi
  do marital
case upper(chr(f)) $ "P"
  set colo to n/w*
  @ 19,52 say "Appointment Type File"
  set colo to
  @ 19,2 say "You are about to update APPOINTMENT TYPE FILE."
  @ 21,8 say "Do you want to continue (Y/N) ..."
  fg=0
  do whil fg=0
    fg=inkey()
    if upper(chr(fg)) $ "YN"
      exit
    endi
    fg=0
  endd
  @ 12,51 clea to 23,75
  if upper(chr(fg)) $ "N"
    loop
```

```
      endi
      do appt
    othe
      exit
  endc
endd
clos all
@ 12,51 clea to 23,75
retu
```