

THE IMPACT OF COMPUTERIZED AGRICULTURAL CREDIT
GUARANTEE SCHEME (ACGS) OPERATIONS ON AGRICULTURAL
CREDIT DELIVERY IN NIGERIA FROM 1978 - 1998

By

JOHN AKINTOYE ADEBOYEKU
PGD/MCS/128/96

DEPARTMENT OF MATHEMATICS\COMPUTER SCIENCE
SCHOOL OF SCIENCE AND SCIENCE EDUCATION
FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA
NIGER STATE

DECEMBER, 1999

THE IMPACT OF COMPUTERIZED AGRICULTURAL CREDIT
GUARANTEE SCHEME (ACGS) OPERATIONS ON AGRICULTURAL
CREDIT DELIVERY IN NIGERIA FROM 1978 - 1998

By

JOHN AKINTOYE ADEBOYEKU
PGD/MCS/128/96

A Project Submitted to the department of
Mathematics\Computer Science, Federal University of
Technology, Minna, Niger State in Partial Fulfilment of the
Requirement for the Award of Post-Graduate Diploma in
Computer Science.

DECEMBRE, 1999

CERTIFICATION

Having read through this project carried out by me (Mr. John Akintoye Adeboyeku), we certify that this project work is originally written by above named student and has met the standard OF the requirement for both the department and the institution in general.

Dr. Yomi Aiyesimi
(Project Supervisor)

Date

Dr S.A. Reju
(Head OF Department)

Date

External Examiner

Date

DEDICATION

To the Glory OF God, and the support of my wife (Mrs. Abosede Olufunke Adeboyeku) for giving me all logistics and moral support. May the Lord bless her abundantly, (Amen).

ACKNOWLEDGEMENT

First and foremost, I am particularly grateful to God Almighty for his mercy, love and guidance throughout the duration of this study.

My profound gratitude to my able and kind supervisor Dr. Yomi Aiyesimi whose guidance resulted in the successful articulation and coordination of materials used in this work.

However, my sincere appreciation goes to Professor K. R. Adeboye and to the lecturers of the Mathematics\Computer Science Department for their patience and understanding in the course of imparting knowledge on me. I say millions of thanks to you all and may God the Almighty reward you all abundantly (Amen).

I am grateful to all members of my family, my wife Bose and the children who pleasantly stood by me throughout the duration of the study. I remain ever grateful.

I will also want to acknowledge the effort of Mr. Olaiya Mohammed Mutiu of Ami-Tech Computer Nig. Limited for his untiring effort, patience and support for making sure that this project work becomes a reality. Mutiu, I thank you very much. May the Almighty God in his infinite mercy bestow you with uncountable blessing, (Amen)

Finally, I am also grateful to all my friends and colleagues that are classmates of this study in the Central Bank of Nigeria, Minna and Abuja, for providing with the necessary publication and data used in this study. I pray Almighty God to reward and bless you all, (Amen).

ABSTRACT

One of the major constraints identified as confronting higher agricultural productivity in Nigeria is inadequate capital, modern tools and credit facilities to the farmers in Nigeria.

In order to ameliorate the funding situation of the farmers, the government of Nigeria through the monetary authority directed all commercial Banks to allocate certain percentages of their total lending to agricultural sectors yearly.

Moreover, the Federal Military Government also took a major step by establishing the Agricultural Credit Guarantee Scheme Fund (ACGSF) by decree No. 20 of 1977 to shoulder 75% of the risk of default by farmers.

However, since the establishment of the ACGSF in 1977 to date, the method of storing processing and retrieval of farmers information and records have been a traditional method which therefore causes the following problems :

1. In appropriate method of storing information
2. In adequate methods of processing farmers record on ACGSF
3. Time consumption
4. Duplication of farmers record
5. Mis-handling of farmers file
6. At long last, termites ends up by eating the pile of farmers file at CBN offices.

Because of the importance attached to the farmer's record and also because of the above mentioned problems, there is need to transform the traditional methods of processing (Manual based) to the modern methods that entail the use of "Electronic Data Processing" machine, popularly known as the "COMPUTER". Hence this project work intends to look into the "Impact of Computerised Agricultural Credit Guarantee Scheme (ACGS) on agricultural credit Delivery in Nigeria from 1977

- 1998

TABLE OF CONTENTS

| | |
|-------------------|--------|
| Title Page | i |
| Certification | ii |
| Dedication | iii |
| Acknowledgement | iv |
| Abstract | v |
| Table of Contents | vi-vii |

CHAPTER ONE

1.0 INTRODUCTION

| | |
|------------------------------|---|
| 1.1 Background of the Study | 1 |
| 1.2 Statement of the Problem | 2 |
| 1.3 Objectives of the Study | 3 |
| 1.4 Scope of Study | 4 |
| 1.5 Limitations of the Study | 5 |

CHAPTER TWO

2.0 LITERATURE REVIEW

| | |
|---|----|
| 2.1 Meaning of Agricultural Credit | 6 |
| 2.2 Types of Agricultural Credit | 7 |
| 2.3 Sources of Agricultural Credit | 8 |
| 2.4 Roles of Agricultural Credit | 9 |
| 2.5 Technology, Extension and Agricultural Credit | 11 |
| 2.6 Problems of Agricultural Credit | 13 |

CHAPTER THREE

3.0 THE CONCEPT OF SYSTEM ANALYSIS AND DESIGN

| | |
|-------------------------------------|----|
| 3.1 Process of Program Development | 18 |
| 3.2 Understanding System Analysis | 19 |
| 3.3 System (ACGS) Design Procedures | 21 |

| | | |
|-----|---------------------------|----|
| 3.4 | Input Specification | 31 |
| 3.5 | Output Specification | 31 |
| 3.6 | Structure of the ACGS | 31 |
| 3.7 | Cost and Benefit Analysis | 34 |

CHAPTER FOUR

| | | |
|------|---------------------------------------|----|
| 4.0 | <u>PROGRAMMING AND IMPLEMENTATION</u> | |
| 4.1. | Programming | 36 |
| 4.2. | SystemImplementation | 37 |
| 4.3 | Testing and Debugging | 38 |
| 4.4 | System Conversion | 38 |
| 4.5 | Hardware and Software Requirement | 39 |

CHAPTER FIVE

| | | |
|-----|---------------------------------------|----|
| 5.0 | <u>GENERAL SUMMARY AND CONCLUSION</u> | |
| 5.1 | Summary | 41 |
| 5.2 | Conclusion | 42 |
| 5.3 | Recommendation | 42 |

References

Appendix

CHAPTER ONE

1.0 INTRODUCTION

1.1 BACKGROUND TO THE STUDY

Agriculture used to be the mainstay OF the Nigerian economy well before independence until early 70's when oil become the dominant factor OF the economy. More than 85 percent OF Nigerians were engaged in one form OF farming or the other. However, agriculture suffered severe neglect since the advent OF oil boom in early 70's. As a result, the nation started to import food and other agricultural products that were hitherto, produced in the country. Finally, the value OF food imports rose from N819million in 1970-5 to N7577 million in 1981-85, a rise OF 925.15 percent.

Many factors combined to cause this decline. Important among these factors according to Olaloku are “the low yield variety OF crops and planting materials used by a large proportion OF the farming population, inadequate extension services, the poor condition OF feeder roads and other transport facilities, inadequate supplies OF agricultural inputs, the problem OF diseases and the center OF them all is in adequate credit facilities”.

In order to reverse this undesirable situation successive Government in Nigeria have taken several steps to boost agricultural production in the country. Notable among these efforts were the National Accelerated Food Production Program (1973), Operation feed the Nation (1976), the Green Revolution(1980), Directorate OF food, Road and Rural infrastructure (1986).

Special projects were also embarked upon to enhance agricultural production such as World Bank assisted Agricultural Development projects, River Basin and Rural Development Authority, Mass Enlightenment campaigns, introduction OF technology package and provision OF fertilizers and improved seedlings to farmers were also some OF the attempts made at raising agricultural productions.

In the area OF agricultural financing, policy measures introduced include Rural Banking and Landing programmes, Direct financing OF Agricultural Credit, Concessional interest rate on Agricultural loans, Sectoral Allocation and Agricultural Credit Guarantee Scheme(ACGS) which is our main focus.

1.2 STATEMENT OF THE PROBLEM

One OF the major constraints identified as confronting higher agricultural productivity is inadequate capital and credit facilities to farmers.

In order to ameliorate the finding situation OF farmers, the government through the monetary authority to allocate certain percentages OF their total lending to agricultural sector yearly.

The Federal Military Government also took a major step by establishing the Agricultural Credit Guarantee Scheme Fund(ACGS) by Decree No 20 OF 1977 to shoulder 75percent of the risk of default by farmers.

The decree provides for a fund of N100 million subscribed to by the Federal Military Government (60 per cent) and the central Bank of Nigeria (40 per cent). The purpose of the Fund is to provide guarantee in respect of loans granted by any Bank for agricultural purposes with the aim of increasing the level of Bank credit to the agricultural sector. "Loan" under the Decree includes advances, overdrafts and any Credit facility.

The Fund is under the management of the Agricultural Credit Guarantee Scheme Fund Board and the central bank Of Nigeria is the managing agent for the administration of the scheme. The work relating to the scheme at the Head Office of the Central bank is handled in the Agricultural Finance department of the bank headed by the Director of Agricultural finance. In order to avoid delay, much of the work on the scheme is decentralized to the Central Bank of Nigeria in the states in which transaction under the scheme takes place.

Sectoral Allocations Moratorium on agricultural loans and concessional interest on such loans was also introduced as a policy instrument by the government in order to make loan repayment easy for farmers.

All these policy measures were introduced in order to enhance the financial position of the farmer which hopefully translates into easy access to farm inputs hence higher agricultural productivity.

The problems facing agricultural production either as a foreign exchange earner and more importantly as a source of food for the nation should engage the attention of all and sundry. According to Bola(1985), "A nation that cannot afford to feed itself and has to rely on importation (more so in an era of scarce foreign exchange) is on a suicidal route" ²

Lawal (1993) also notes that "Credit Supply in Agriculture has long been identified as a veritable tool towards agricultural development in developing countries" ³. The case of Asian countries like Indian and Pakistan which share similar Socio-economic stratification of Agricultural Communities with Nigeria readily comes to mind. According to Amin(1976) Bulk of the farmers are in the peasant category.⁴ Thus, their economic and financial status is very weak. This situation is the same with what obtain in Nigeria according to Bola⁵.

The Central bank of Nigeria was mandated to organize, manage and administer the scheme through a Board. However, administrative, technical and inconsistent government policies constitute serious set backs to the success of these programmes resulting in poor performance of the Scheme.

1.3 OBJECTIVE OF THE STUDY

Agricultural Credit to farmers is an important instrument in improving farm productivity and marketing. Therefore, the major theme of this study is the impact of CBN Agricultural Credit Guarantee Scheme(ACGS) on Agricultural Credit Delivery in Nigeria as a whole and Niger State in particular.

Specific objectives of the study include:

1. To show the extent to which the Central Bank of Nigeria has played a significant role towards influencing agricultural credit delivery through the Agricultural Credit Guarantee Scheme (ACGS) over the years.
2. To examine the case study or implementation of Central Bank of Nigeria, Minna, Niger State in particular.
3. To identify the problems hindering the achievement of desired objectives of the Central bank within the stated period.
4. To examine the System development and Implementation
5. To recommend solutions to some of the problems faced by both farmers and institutions providing the credit
6. To design a Software Package(ACGS) that will captured all applicant requesting for loan and
7. Generate all necessary forms of reports for management decision making from the ACGS software.

1.4 **SCOPE OF STUDY**

In this study, emphasis is placed on assessing the impact of Central bank's Agricultural Credit Guarantee Scheme ACGS on Agricultural Credit delivery in Nigeria on one hand and the operations of its activities in Niger State on the other.

The ACGS is going to be examined with the instrument put in place to ensure its success such as moratorium on Agricultural loans, Concessionality interest rate, Sectoral prescription and relations to the Nigeria Agricultural and Cooperative bank.

1.5 **LIMITATIONS OF THE STUDY**

A study of this nature cannot be devoid of problems or limitations. The major constraint was

1. There were few writings on Agricultural financing within reach and the few that are available were far apart.
2. Data collation from annual reports and other publications of CBN proved to be too taxing and time consuming.
3. High cost of traveling from one place to another due to peculiar High cost of petrol in order to gather necessary information.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 MEANING OF AGRICULTURAL CREDIT

Agricultural Credit is money advanced to an individual or an organization for agricultural purposes. The borrower pays the money after a specified period. He pays the principal plus the interest.

According to Baker(1965) "Credit is an asset or a financial reserve, and loan is any portion of this that is used on which the user incurs an interest charged". However, majority of the Nigerian farmers are small scale farm operators and usually the credit approved for them are fully disbursed because they are small and barely enough for their needs. Thus equating credit and loans. The few exceptions are in large scale agricultural projects where approved credits are of large sizes and are therefore drawn-down in trenches of loans.

Abe(1981), also defined agricultural credit as "that which encompasses all loans and advances granted to borrowers whether beneficiaries of agricultural reforms or some other one to finance and service production activities relating to agriculture, fisheries and for processing, marketing, storage and distribution of products resulting from these activities".

In Nigeria, as well as over the world, it is important to note that agriculture is no longer a homogenous industry, as it is made up of several distinct groups with unique characteristics. Whether the farmer is an individual or a family unit or a co-operative group, the types of finance required are conditioned by certain factors such as:

- I. Identification and accessibility to appropriate finance.
- ii. Availability of loanable funds from the identified source.
- iii. Satisfaction by the borrowing farmer of the lender's criteria which are noted in sound

principles of lending and

- iv. Accommodation of such credits within the law of the land and in accordance with economic, political, monetary, fiscal and societal policies of government.

2.2 TYPES OF AGRICULTURAL CREDIT

There seems to be a general consensus in the literature that agricultural credit can be divided into three categories.

1. SHORT TERM FINANCE

This is required to cover annual (seasonal) working capital for growing crops, wages, purchase of feeds and other short term assets that are completely used up during production.

2. MEDIUM TERM FINANCE

This encompasses funds tied up for two to five years in asset that are eventually used in the production process. These assets are partly fixed like vehicles, tractors and livestock.

3. LONG TERM FINANCE

This is usually needed to finance fixed assets such as land and land rights, construction of farm buildings, purchase of machinery, plants and equipment, etc. for development purposes.

The period normally ranges from ten years upward.

The above Classification is based on the duration and usage of the credit. Makings(1987)³ while accepting the above categorization, based his own on the type of farmer using the loan. According to him there are distinct categories of need in agriculture for which credit is required and these are:

1. Emergent Farmer Credit
2. Commercial Farmer Credit
3. Group Development Credit

Emergent farmers are those land users who are a little more than subsistence producers seeking to

establish a foothold in the market Economy. They constitute a majority of the farming population. Commercial farmers on the other hand are the important but smaller class. Group development Credit is an essential category of need for use at the group or Community level.

2.3 SOURCES OF AGRICULTURAL CREDIT

There are different sources of funds for agricultural purposes, which analysts have classified in a number of ways. Formal and Informal, Non institutional and Institutional, Private and Public, Internal and External, etc. These classifications are based on certain features.

When we talk of institutional and non institutional, we are trying to draw a line between funds that can be obtained from banks and non-banking government established agencies and those obtainable from unorganized private money lenders/ unrecognized groups on the other hand.

Also classification of funds sources into internal and external implies distinguishing between funds provided by sponsors of a project and unallocated funds (re-invested funds) generated by the project from previous operations on one hand and funds source from outside the project on the other hand.

In Nigeria, one can identify four broad categories of banks that are involved directly in providing credit to farmers. They are:

1. Central Bank Of Nigeria which by its peculiarities is unique being also the apex bank.
2. The Commercial Bank which supply a substantial portion of total agricultural credit in the economy; especially the short and medium term loans.
3. Merchant Bank are required to give medium to long term credits to Nigerian farmers.
4. The development banks which are specialized financial institutions expected to play significant roles by their long term financing of agro allied industries. The Nigerian Agricultural and Co-Operative Bank(NACB) is entirely required to finance farmers directly or through their bodies all over the Country.

It is important to note the role played by foreign institutional sources notably International Bank for Reconstruction and Development (IBRD) otherwise known as World Bank. Its mode of operation is to extend loans to national governments for the purpose of financing development projects thus, sparing itself the task of dealing directly with numerous investors. In this way, the bank had advanced substantial loans to Federal and state governments for Agricultural Development projects in which a large number of Nigerian rural dwellers are actively engaged.

2.4 **ROLE OF AGRICULTURAL CREDIT**

There are divergent views as to exact causes of the poor state of the agricultural sector. Various scholars seem to have arrived at a consensus that the most important factor that has tended to inhibit any meaningful improvement is the inadequacy of credit facilities, because modern agriculture apart from being capital intensive and mechanized is carried out on a large scale.

Given the predominantly subsistence nature of the peasant farmer, he relies mainly on his own capital or/and borrowing from friends, local money lenders and traders which apart from their inadequacy carry with them exorbitant interest charges.

Miller (1954) observed that "lack of credit is generally recognized as an important constraint in expanding food production and modernizing agriculture. Adopting new technology he believes ordinarily requires the use of some inputs which are not available on the farms and hence must be purchased from others"⁴.

Lack of credit or finance may not necessarily prevent the adoption of new technology by some financially able individuals and organization. Experience in Nigeria has shown that unless agricultural credit is made available on suitable terms, majority of small farmers will be seriously handicapped. In Nigeria, these small scale farmers are the most important group, hence it is then not large producers that Nigeria must look for in most of its expanded food production.

The United Nations through its Food And Agriculture Organization (1965) reported that

“agricultural credit makes a substantial improvement in agricultural production, ultimately, raising the standard of living of the rural population, in as much as it provides the farmer with a means of improving his land, of using modern tools and machinery and of introducing better methods of cultivation”⁵.

From the foregoing assertions and conclusions, one can sum up the assumptions behind conventional credit programmes as being:

- I. The farmers need credit to adopt new technology.
- ii. That they cannot get credit at reasonable cost from unrecognized private sources outside the programme.
- iii. That the provisions of credit breaks the most important constraint of small farmer’s progress.

Many studies have shown the potential effect of increased level of capital in agricultural production.

Ogunfowora(1972) concluded that “when increased capital base is combined with improved technology, greater opportunity exists for larger volume of production and higher level of farm income”⁶. Agricultural Credit is meant to enable the farmer purchase productive equipment and other farm inputs. Credit is therefore, less important to agricultural development of these equipment and supplies are not readily available at convenient markets. Where they are readily available and the farmers are capable of learning how to use them, agricultural credit can accelerate the adoption of improved technique.

It is important to de-emphasize over rating of agricultural credit as a panacea for low productivity. FAO(1965) observed that “Agricultural credit is only one of the many factors playing a part in the complicated process of stepping up agricultural production. It is stressed that, far from being a panacea, credit is not even the harmless patent medicine which it is often thought to be”⁷.

For agricultural Credit to perform its functional role of increasing farm productivity, it has to be properly utilized and this is dependent on the provision of certain preconditions among which are extension service, technology, education and training. FAO(1965)⁸ summarized these pre-

requisites necessary for successful implementation of agricultural credit policies as follows:

- i. The existence of adequate and efficient socio-economic planning and the desire of all parties and groups concerned to implement it. Effective planning presupposes the existence of adequate statistical and an effective administration at all levels.
- ii. An adequate rural infrastructure(roads, rail, storage,etc.)
- iii. An efficient system of stabilizing fluctuation in prices for agricultural produce.
- iv. A proper system of land tenure.
- v. Adequate and effective arrangement for marketing and supply.
- vi. A well organized and satisfactory operated agricultural extension service and
- vii. Continuity in government/policies.

Agricultural Credit Schemes are like any other economic venture and from this perspective, it is important that it has to balance its total cost against total returns. This means that there is the need for adequate feasibility studies and investment appraisal is imperative to the success of the credit programme which depends on the success of the individual farms is envisaged.

2.5 TECHNOLOGY, EXTENSION AND AGRICULTURAL CREDIT

• TECHNOLOGY

Agricultural credit will be ineffective without a technology that is profitable or viable. Technology in farming implies the way things are done i.e. a given set of inputs or factors of production. Thus, traditional technology means the particular way the inputs of land, labour, seeds, organic fertilizer, land implements, oxen, etc. are combined and used. New technology refers to a new set of inputs or factors of production which are difficult and often better than the traditional set.

Two studies(Miller,1966, Meyer 1972) have concluded that “it is not profitable to further expand the use of traditional technology through credit. This is primarily due to the very low production functions that exist for traditional practices”. It is thus, argued that further expansion of

the traditional technology will not increase the small scale farmer's net income commensurate with the adopted costs and risk involved in borrowing capital to finance such expansion. Thus, emphasis is supposed to be on relevant and viable new technology.

It is however presumptions to say that new technology is available or can be readily available when desired. Even if it is available, it may not be profitable having not been tested and adopted under the conditions similar to those in Nigeria.

In conclusion, it can be said that when a new technology is available and has been understood by farmers, perceived appropriately and proved profitable then credit requirement assumes importance. In this circumstance, the lack of capital and credit would pose a serious limitation to the adoption of new technology.

- **EXTENSION SERVICES**

Agricultural extension could be explained as the mechanism by which the new technology is developed, modified, carried and translated to the farmer in coordination with the provision of credit when needed. "Agricultural extension has the long term objective of securing improvements in the efficiency of production and distribution of all food and agricultural products, bettering the condition of the rural population and thus playing the central role of raising the standard of living and the level of national economy"⁹ .

Extension service, credit and other relevant services all aim at a particular purpose, to help farmers increase their production and income hence improve their living standards. Therefore, a good approach towards a successful credit programme should be one integrated with provision of extension services to farmers. FAO(1965) believes that to have an adequate impact on agricultural production, the provision of credit should be accompanied by and coordinated with sufficient amount of technical advice. Particularly at early stage of agricultural development, supervised credit should compensate for greater part of extension education.

Belahaw(1959) says that supervised credit is a technical term which describes a particular system of credit provision aimed at improving both production and family living standards; and also coordinates credit provision with extension services and aims at effective supervision, but usually goes further, for example, in making improved provision for marketing and other auxiliary services”¹⁰

Emphasizing the importance of complementary role of extension services to agricultural credit, Johnson (1982) states that” no agricultural credit programme can be effective unless it is combined with satisfactory extension services and other forms of support..... Production credit given together with technical advice and help is called supervised credit”¹¹.

Despite the complementary nature of credit and extension service, FAO(1965) notes that little initiative has been shown from the side of the agricultural credit institution or from that of extension services to bring about the essential coordination between credit and extension education. It warns that expansion of agricultural credit without the provision of an adequate amount of extension education is a medicine which effect may be worse than the disease. It concludes that agricultural credit and agricultural extension, exercise great influence in the rural areas of developed countries where high level of education and technical know-how is needed. In developing countries, they should be properly co-ordinated so as to render credit as an effective instrument for promoting economic and social progress.

2.6 PROBLEMS IN AGRICULTURAL CREDIT

The Agricultural Credit system generally faces some organizational and operational problems which include the following:

a. **STRUCTURAL**

This has two aspects namely: (i) Bureaucracy and (ii) Top-down Implementation

i. **BUREAUCRACY**

There is excess paper work on one hand and multiple level of decision making on the other hand before credit is granted. Any un-necessary diffusion of decision making process definitely leads to inefficiency, creates room for abuses and political meddling in purposes and procedure of the agricultural credit system.

ii. **TOP-DOWN IMPLEMENTATION PROBLEM**

Usually, Board and Committees of the Institution set the policies. Hardly, then, are clients represented in the committee. One adverse effect of this is that institutions employees look up to the top for support and reward rather than on the clientele for services.

b. **LOAN SUPPLY SHORTAGES**

One major defect in the credit programme of most developing countries and Nigeria in particular is the inability of organizations to meet or a significant proportion of the credit needs of farmers. This even calls to question the arbitrary manner in which amounts applied for are cut and approved by lending institutions without thorough assessment and appraisal of the implication of such actions on the cash position and subsequent viability of the project.

c. **PERSONNEL**

In Nigeria, little time and money is spent to train specialists to run an efficient credit programme. In most cases, top officials are appointed on ethnic, religion and other parochial considerations. Ugoh(1977) in his criticism of the approach where people without the requisite expertise are posted to manage these institutions, stated that "a successful civil Servant is one who can master the good reason why no action should be taken on ^{amateur} .

d. **POOR SUPERVISION OF LOANS**

There are instances where the credit institutions do not provide complementary input and/or supervision (in the form of farm inputs, technical assistance and supervision itself). Extension workers are supposed to supervise the introduction of innovation to farmers as well as monitor and evaluate credit programme on a continuous basis. There has been lack of interest, poor supervision and unsatisfactory record keeping of detailed information on loan beneficiaries. These approaches negatively impact on the efficient and effective operation of Agricultural credit system. The extension workers are not entirely to be blamed. Lack of fund and inadequate facilities which have become prevalent in recent times have tended to constraint the performance of these tasks as expected.

e. **DELAY IN TIME OF LOAN DELIVERY**

Timeliness of loan disbursement is very vital farming. Since most farm operations are seasonal and time specific (such as planting, weeding, harvesting), labour inputs have to be paid for promptly, but what is obtainable in practice is that it takes a long time for funds to be released to users even where they have been approved accordingly. This suggests that timeliness of loan delivery may facilitate proper utilization and thus enhances their effective repayment of such loans.

f. **COST OF CREDIT**

The cost of providing credit is measured by the interest rate charged. Adegaye et al (1982), contended that, "that the cost of providing credit, which is measured by the interest rate, is therefore not only the basic cost of capital (i.e. the next best use of money) but the cost of supervising the loan (administrative charges), and the risk premium for the possibility of default in repayment as well as a charge for possible inflation"¹³. All these contribute to make the cost of institutional lending quite high as charging interest rates below that needed

to cover the cost of lending means subsidization which few less developed countries in reality can afford.

g. DEFAULT IN LOAN REPAYMENT

Loan default can be defined as the “inability of a borrower to fulfil his or her own loan obligation as and when due. An overdue loan obligation usually includes that part of the principal (unpaid) and accrued interest payments stipulated in the loan agreement schedule which were entered into during loan negotiations”¹⁴.

Various factors have been identified as major determinants of loan delinquency. Empirical evidence from Okorie (1986) showed that “the nature, time of disbursement, supervision and profitability of enterprises which benefited from such loans... contributed to the repayment ability and consequent default rates”¹⁵.

Secacy (1985) identified some mutually related factors which influence loan delinquency in farmers credit programme to include “low income, due to poor production and poor prices, which may in turn be related to misuse of loan procedure, lack of technical supervision, inefficient government policies and small size farms”¹⁶. In particular, secacy identified sheer poverty due to poor farm income as being the primary root of delinquency and that the level of indebtedness, mis-allocation of loans, low sales proceeds, low educational attainment and tenurial status might be associated with the low level incomes. Secacy’s study also showed that “about 30 percent of farmers surveyed, for example, admitted using the loans for family expenses”¹⁷.

In all high default rates in credit programmes should be the major concern of policy makers in developing countries, and Nigeria in particular. This is due to its unintended negative impact on agricultural financing and output. According to Von Piselike (1980), some default include “inability to recycle funds to their borrowers, detriment of other

financial intermediaries from serving the needs of farmers, and the creation of distrust”¹⁸.

CHAPTER THREE

3.0 THE CONCEPT OF SYSTEM ANALYSIS AND DESIGN

3.1 PROCESS OF PROGRAM DEVELOPMENT

The software or systems development process is a cooperative effort of the users and computer professionals. While, computer professionals are those familiar with the technology and it can be applied to meet a business information processing needs, users on their side have an in-depth familiarity with their respective functional areas. The skills and knowledge of these two groups complement each other and can be combined to create any type of information system during the system's development process.

However, because systems development is a team effort, most organizations have adopted a standardized Systems development methodology that provides a framework for cooperation. This step-by-step system development approach is illustrated below in tabular form.

| <u>STEPS</u> | <u>ACTIVITIES</u> | <u>MEASURABLE OUTPUT</u> |
|---------------|---|--------------------------------------|
| Analysis & | Appraisal of existing situation. | Assessed need report |
| Specification | Identification of user requirements | Implementation of feasibility report |
| Design | Design the overall program structure | Process logic specifications, using |
| | Design the detailed processing logic. | Pseudocode or flowchart technique |
| Programming | Writing of code in the appropriate processor language. | Hand written program code |
| | Entry of written code into the computer | Computer printed list-out of code |
| Testing | Removal of syntax and logic errors | |
| & | | |
| Debugging | Final testing of program | Error-free program execution |

Installation Error-free program installation

&

Program development and upgrade

Maintenance User training

procedures.

System development process is essentially the same, be it for an airline reservation system, Micro-Credit Scheme System or an Inventory Management System. As member of a *Project Team* progress through the procedures outlined in a systems development methodology, the result of one step provides the input for the next step and/or subsequent steps. The project team typically is made up of both users and computer professionals. The methodological approach to systems development is a tool that information services and user-managers employ to coordinate the efforts of a variety of people engaged in a complex process. For a successful program development of the ACGS software, it is assumed that the writer of the project is part of the project team (programmer) and the following five (5) important phases of system development were fully examined by the author.

3.2 **UNDERSTANDING SYSTEM ANALYSIS/ANALYSIS OF ACGS**

System analysis according to Fapohunda (1995) is defined as the process of examining an existing system. The existing system could be manual-based operation or computerized system. The essence of the analysis is to find the deficiency in an existing system for proper and adequate new system to be put in place of the old system.

However, in order to understand the analysis of ACGS, the existing operation (manual method) at the Central Bank of Nigeria, CBN, is divided into four component parts which are discussed below.

■ **THE EXISTING ACGS (MANUAL) SYSTEM**

Before designing a new or enhanced Information System, sufficient and proper grasp of the existing work and information flow was made. This was done by conducting interview with the staff of

ACGS department of CBN, Minna. The interview and observation gathered from the case study were documented and the information flow of the proposed ACGS System was reduced to the basic component (input, Processing, and Output) for easy design technique of the ACGS software.

■ **SYSTEM'S (ACGS) OBJECTIVES**

Since the existing system is documented, analysis was carried out on the documentation in order to identify the obvious, and the not so obvious problems of the manual existing operation, including inefficiencies in information flow and storage, duplication of effort, deficiencies in information dissemination, staff discontent, problems with customers interaction, inaccuracy of operational data, and so on.

After the identification of all relevant and non-relevant problems, possible solutions were used in order to commence the system design for the attainment of the system objectives.

■ **DESIGN CONSTRAINTS**

Here, the intended ACGS system will be developed subject to specific constraints encountered during the time of system analysis of the existing operation. The purpose of this activity is to detail, at the onset of the systems development process, any costs, hardware schedule, procedural, software, database and operating constraints that may limit the definition and designation of the target system (ACGS Software).

■ **REQUIREMENT DEFINITION**

Having completed the analysis that resulted in the definition of some certain variables for proper data processing and information requirement. The interview and observation made became the functional specifications for the system input, processing and the output requirements.

However, the functional specifications show the logic of the system (Input/Output, Work and Information flow) from the perspective of the users.

3.3 SYSTEMS (ACGS) DESIGN PROCEDURES

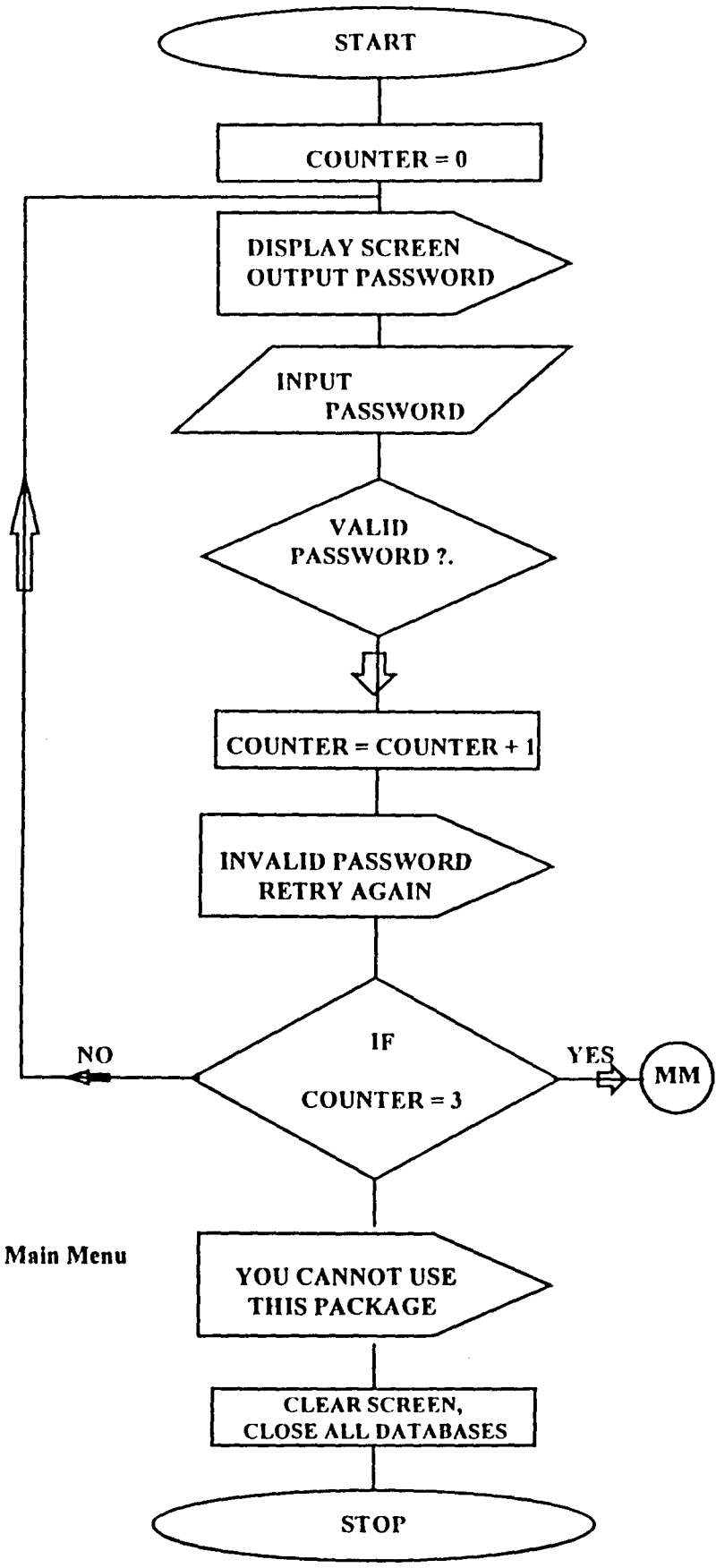
The system design stage of the ACGS Design process produces three (3) output as follows:

1. **GENERAL SYSTEM DESIGN**

The general system design includes the flowchart (procedures), a written explanation (Pseudocode) of the above graphical representation and the general descriptions of the outputs to be produced by the system (ACGS Software). This includes the screen display and hard copy (paper report).

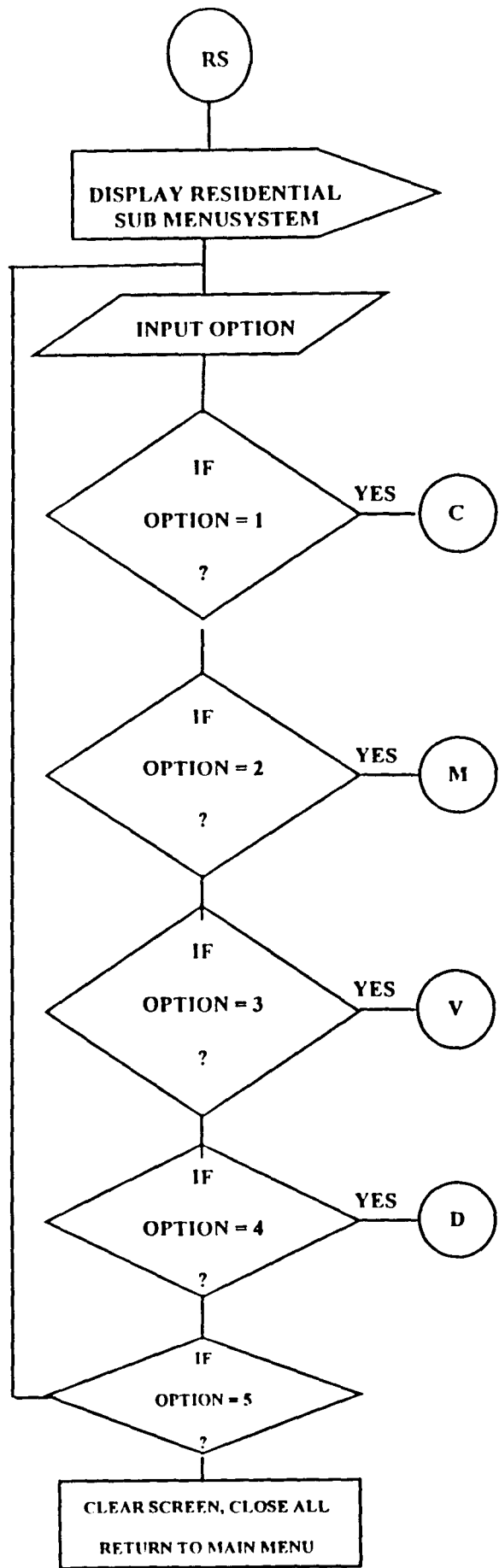
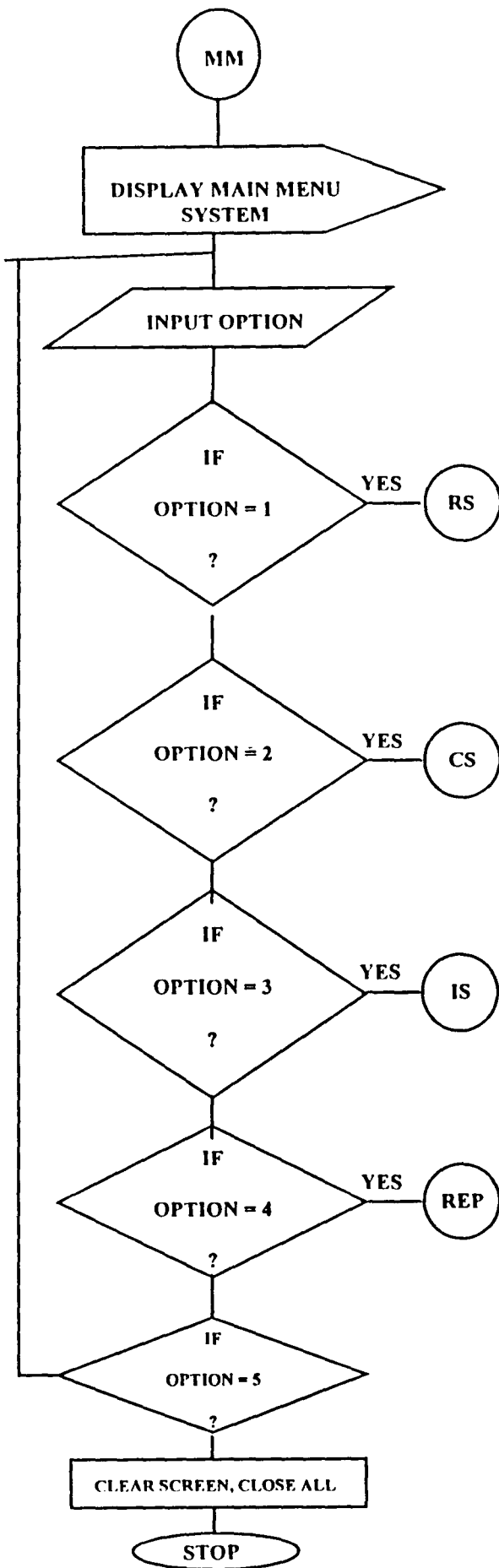
FLOWCHART

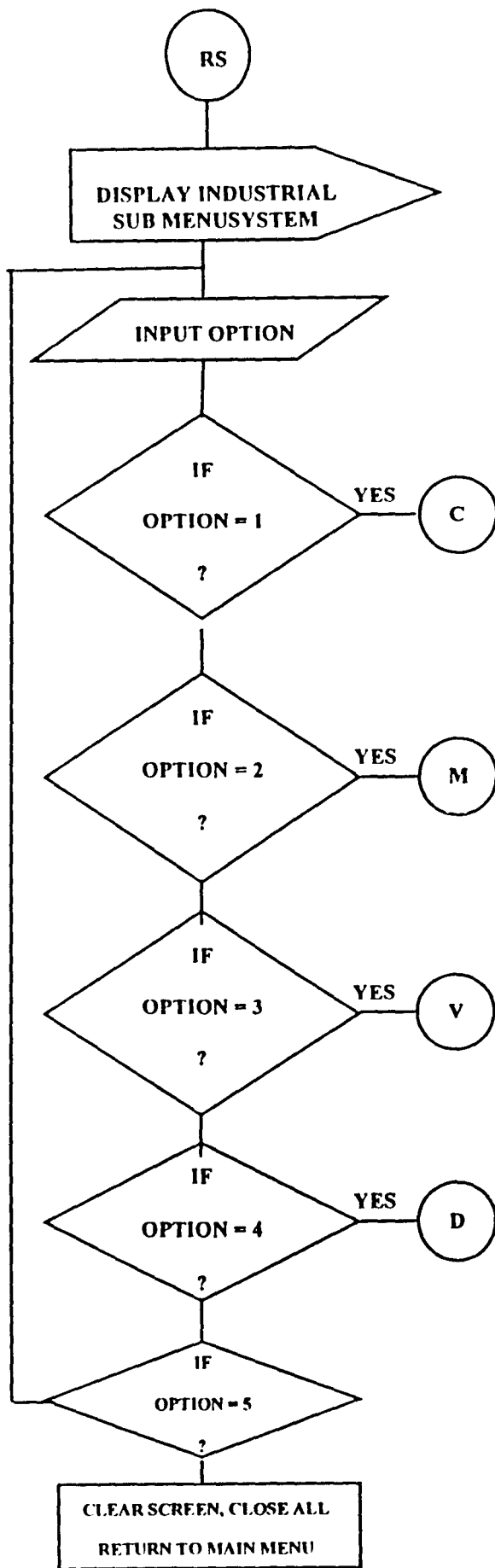
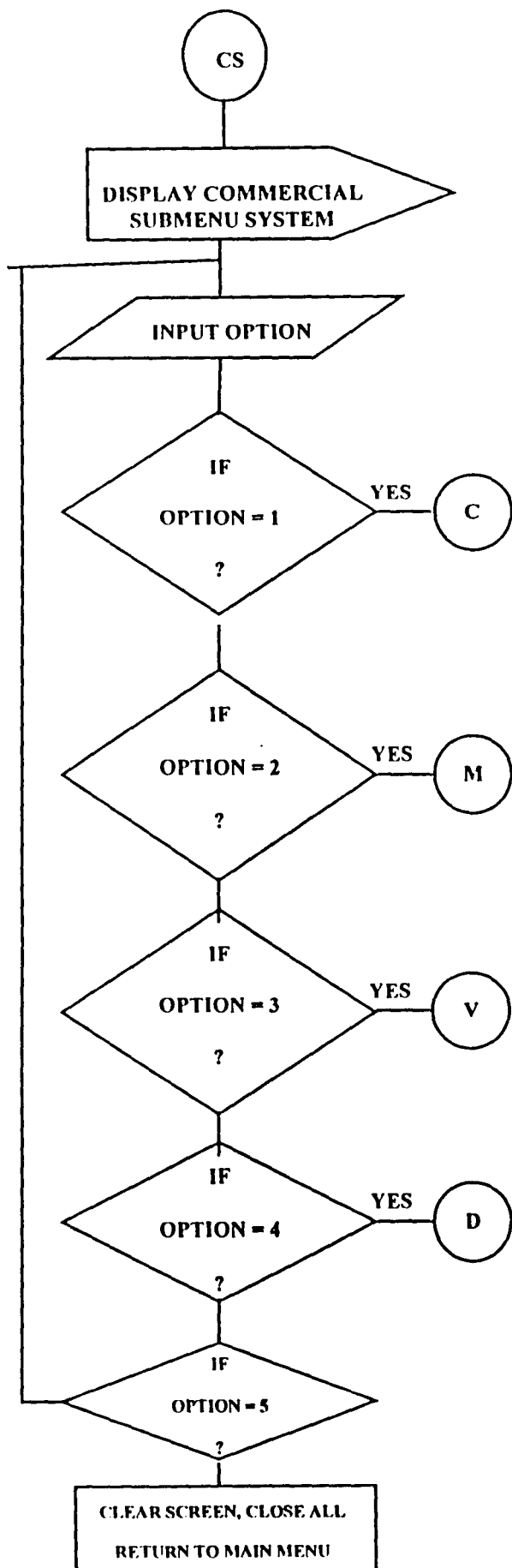
Flowchart is the graphical representation of data, information and work flow by the inter-connection of specialized symbols and flow lines. The combination of symbols and flow-lines portray the logic of the program or system. Below is the ACGS System design flowchart.



KEY

MM - Main Menu





2. DATABASE DESIGN

The database is the common denominator of any system. It contains the raw materials (data) necessary to produce the output, hence all ACGS files (records) are stored in their respective database file as explained below.

BANK.DBF

The Bank database file contains all the record of information that is used for applicant cash transfer and repayment. The physical structure of the database is as shown below: -

| <u>S/No</u> | <u>FIELDNAME</u> | <u>WIDTH</u> | <u>TYPE</u> |
|-------------|------------------|--------------|-------------|
| 1. | BANKCODE | 02 | CHR |
| 2. | BANKNAME | 30 | CHR |
| 3. | SHORTNAME | 03 | CHIR |

TYPE.DBF

The TYPE.DBF is the file that contains all types of available loan information (i.e. Individual loan, Company loan, Informal loan and Co-operative loan). The physical structure of the database is as shown below: -

| <u>S/No</u> | <u>FIELDNAME</u> | <u>WIDTH</u> | <u>TYPE</u> |
|-------------|------------------|--------------|-------------|
| 1. | TYPECODE | 02 | CHR |
| 2. | NAME | 15 | CHR |

CATEGORY.DBF

This database file is used to store all types of project categories' information (i.e. Food Crops, Cash Crops, Mixed Farming, etc.). The physical structure is as shown below: -

| <u>S/No</u> | <u>FIELDNAME</u> | <u>WIDTH</u> | <u>TYPE</u> |
|-------------|------------------|--------------|-------------|
| 1. | CAT_CODE | 02 | CHR |
| 2. | CAT_NAME | 15 | CHR |
| 3. | CATEGORY | 15 | CHR |

REQUEST.DBF

The request database file is the master database file of the ACGS System. A master file is a file that contain all the necessary information needed for the proper and adequate updating/running of the ACGS System. Below is the physical structure of the database file used.

| <u>S/No</u> | <u>FIELDNAME</u> | <u>WIDTH</u> | <u>TYPE</u> |
|-------------|------------------|--------------|-------------|
| 1. | REGDATE | 8 | DATE |
| 2. | REGTIME | 8 | CHR |
| 3. | REQUESTNO | 4 | NUM |
| 4. | LOANCODE | 2 | CHR |
| 5. | LOANDESC | 15 | CHR |
| 6. | NAME | 30 | CHR |
| 7. | ADDRESS | 20 | CHR |
| 8. | LOCATION | 20 | CHR |
| 9. | CAT_CODE | 2 | CHR |
| 10. | CAT_DESC | 15 | CHR |
| 11. | BANKCODE | 2 | CHR |
| 12. | BANKNAME | 30 | CHR |
| 13. | SHORTNAME | 3 | CHR |
| 14. | ACCTNUMBER | 19 | CHR |
| 15. | AMT_WANTED | 14,2 | NUM |
| 16. | AMT_GIVEN | 14,2 | NUM |
| 17. | PAY_TODATE | 8 | DATE |
| 18. | BALANCE | 14,2 | NUM |
| 19. | PAY_DATE | 8 | DATE |
| 20. | DURATION | 3 | NUM |

| | | | |
|-----|-----------|------|------|
| 21. | LOANDATE | 8 | DATE |
| 22. | REMARK | 8 | CHR |
| 23. | CODE1 | 1 | NUM |
| 24. | NAME1 | 20 | CHR |
| 25. | AMT_WANT1 | 12,2 | NUM |
| 26. | AMT_PAID1 | 12,2 | NUM |
| 27. | PAY1 | 12,2 | NUM |
| 28. | BAL1 | 12,2 | NUM |
| 29. | PAYDATE1 | 8 | DATE |
| 30. | CAT_CODE1 | 2 | CHR |
| 31. | CAT_DESC1 | 15 | CHR |
| 33. | CODE2 | 2 | CHR |
| 33. | NAME2 | 20 | CHR |
| 34. | AMT_WANT2 | 12,2 | NUM |
| 35. | AMT_PAID2 | 12,2 | NUM |
| 36. | PAY2 | 12,2 | NUM |
| 37. | BAL2 | 12,2 | NUM |
| 38. | PAYDATE2 | 8 | DATE |
| 39. | CAT_CODE2 | 2 | CHR |
| 40. | CAT_DESC2 | 15 | CHR |
| 41. | CODE3 | 2 | CHR |
| 42. | NAME3 | 20 | CHR |
| 43. | AMT_WANT3 | 12,2 | NUM |
| 44. | AMT_PAID3 | 12,2 | NUM |
| 45. | PAY3 | 12,2 | NUM |

| | | | |
|-----|-----------|------|------|
| 46. | BAL3 | 12,2 | NUM |
| 47. | PAYDATE3 | 8 | DATE |
| 48. | CAT_CODE3 | 2 | CHR |
| 49. | CAT_DESC3 | 15 | CHR |
| 50. | CODE4 | 2 | CHR |
| 51. | NAME4 | 20 | CHR |
| 52. | AMT_WANT4 | 12,2 | NUM |
| 53. | AMT_PAID4 | 12,2 | NUM |
| 54. | PAY4 | 12,2 | NUM |
| 55. | BAL4 | 12,2 | NUM |
| 56. | PAYDATE4 | 8 | DATE |
| 57. | CAT_CODE4 | 2 | CHR |
| 58. | CAT_DESC4 | 15 | CHR |
| 59. | CODE5 | 2 | CHR |
| 60. | NAME5 | 20 | CHR |
| 61. | AMT_WANT5 | 12,2 | NUM |
| 62. | AMT_PAID5 | 12,2 | NUM |
| 63. | PAY5 | 12,2 | NUM |
| 64. | BAL5 | 12,2 | NUM |
| 65. | PAYDATE5 | 8 | DATE |
| 66. | CAT_CODE5 | 2 | CHR |
| 67. | CAT_DESC5 | 15 | CHR |

PAYMENT.DBF

This is the database file that is used to store all loan repayment made by the applicants that requested for ACGS loan scheme. The physical structure of the database is as shown below

| <u>S/No</u> | <u>FIELDNAME</u> | <u>WIDTH</u> | <u>TYPE</u> |
|-------------|------------------|--------------|-------------|
| 1. | REGDATE | 8 | DATE |
| 2. | REGTIME | 8 | CHR |
| 3. | REQUESTNO | 4 | NUM |
| 4. | ACCTNUMBER | 19 | CHR |
| 5. | LOANCODE | 2 | CHR |
| 6. | LOANDESC | 15 | CHR |
| 7. | NAME | 30 | CHR |
| 8. | AMT_WANTED | 14,2 | NUM |
| 9. | AMT_GIVEN | 14,2 | NUM |
| 10. | PAY_TODATE | 8 | DATE |
| 11. | BALANCE | 14,2 | NUM |
| 12. | PAY_DATE | 8 | DATE |
| 13. | CODE1 | 1 | NUM |
| 14. | NAME1 | 20 | CHR |
| 15. | AMT_WANT1 | 12,2 | NUM |
| 16. | AMT_PAID1 | 12,2 | NUM |
| 17. | PAY1 | 12,2 | NUM |
| 18. | BAL1 | 12,2 | NUM |
| 19. | PAYDATE1 | 8 | DATE |
| 20. | CODE2 | 2 | CHR |
| 21. | NAME2 | 20 | CHR |
| 22. | AMT_WANT2 | 12,2 | NUM |
| 23. | AMT_PAID2 | 12,2 | NUM |
| 24. | PAY2 | 12,2 | NUM |

| | | | |
|-----|-----------|------|------|
| 25. | BAL2 | 12,2 | NUM |
| 26. | PAYDATE2 | 8 | DATE |
| 27. | CODE3 | 2 | CHR |
| 28. | NAME3 | 20 | CHR |
| 29. | AMT_WANT3 | 12,2 | NUM |
| 30. | AMT_PAID3 | 12,2 | NUM |
| 31. | PAY3 | 12,2 | NUM |
| 32. | BAL3 | 12,2 | NUM |
| 33. | PAYDATE3 | 8 | DATE |
| 34. | CODE4 | 2 | CHR |
| 35. | NAME4 | 20 | CHR |
| 36. | AMT_WANT4 | 12,2 | NUM |
| 37. | AMT_PAID4 | 12,2 | NUM |
| 38. | PAY4 | 12,2 | NUM |
| 39. | BAL4 | 12,2 | NUM |
| 40. | PAYDATE4 | 8 | DATE |
| 41. | CODE5 | 2 | CHR |
| 42. | NAME5 | 20 | CHR |
| 43. | AMT_WANT5 | 12,2 | NUM |
| 44. | AMT_PAID5 | 12,2 | NUM |
| 45. | PAY5 | 12,2 | NUM |
| 46. | BAL5 | 12,2 | NUM |
| 47. | PAYDATE5 | 8 | DATE |

3. **DETAILED SYSTEM DESIGN**

The detailed *Input/output*, Processing and Control requirements, which this stage entails, is the result of the analysis of users' feedback on the general system design. The system design of the ACGS System depicts the relationship between major processing activities, and it is detailed enough for users to determine whether or not that is what they want. This detail design includes all processing activities and the input/output associated with them.

3.4 **INPUT SPECIFICATION**

The Input Specification is an interactive avenue between the computer and the ACGS System. Users communicate by the process of data entries to the system.

3.5 **OUTPUT SPECIFICATION**

The Output Specification gives a format of the expected result of the data entries previously being made. Here, all necessary reports generated are based on the desired output specification from the ACGS system design. The reports generated are mainly for Management decision-making and for future use analysis.

3.6 **THE STRUCTURE OF THE ACGS**

The proposed ACGS menu is designed and structured to be menu-driven, simple, flexible and easy to use and understand. The structure of the main menu is as shown below in figure 1.0

| AGRICULTURAL CREDIT GUARANTEE SCHEME (A Case Study of Central Bank of Nigeria - CBN Minna Branch) | | | |
|--|-----------------|--------|------|
| CREDIT MANAGER | CODE MANAGEMENT | REPORT | QUIT |
| | | | |
| USE UP & DOWN ARROW KEY TO HIGHLIGHT OPTION AND PRESS ENTER KEY TO EXECUTE | | | |

Fig. 1.0 Main Menu System

■ **CREDIT MANAGER**

This menu option is where data entries are going to be taking place for the following options:

- 1 LOAN REQUEST
- 2 LOAN REPAYMENT
- 3 EXIT TO MENU

The Credit Manager is activated by placing the pointer on the option, followed by a press on the ENTER key to execute and display the sub-menu options mentioned above. The general format of the Credit Manager option is as shown below in fig. 2.0

| AGRICULTURAL CREDIT GUARANTEE SCHEME (A Case Study of Central Bank of Nigeria - CBN Minna Branch) | | | | | | | | | | | |
|--|-----------------|----------------|--------------|--|-------------------|---------------|-------------|---------------|--------------|--|--|
| CREDIT MANAGER | CODE MANAGEMENT | REPORT | QUIT | | | | | | | | |
| <table border="1"> <tr><td>LOAN REQUEST</td></tr> <tr><td>LOAN REPAYMENT</td></tr> <tr><td>EXIT TO MENU</td></tr> </table> | LOAN REQUEST | LOAN REPAYMENT | EXIT TO MENU | <table border="1"> <tr><td>CREATE NEW RECORD</td></tr> <tr><td>MODIFY RECORD</td></tr> <tr><td>VIEW RECORD</td></tr> <tr><td>DELETE RECORD</td></tr> <tr><td>EXIT TO MENU</td></tr> </table> | CREATE NEW RECORD | MODIFY RECORD | VIEW RECORD | DELETE RECORD | EXIT TO MENU | | |
| LOAN REQUEST | | | | | | | | | | | |
| LOAN REPAYMENT | | | | | | | | | | | |
| EXIT TO MENU | | | | | | | | | | | |
| CREATE NEW RECORD | | | | | | | | | | | |
| MODIFY RECORD | | | | | | | | | | | |
| VIEW RECORD | | | | | | | | | | | |
| DELETE RECORD | | | | | | | | | | | |
| EXIT TO MENU | | | | | | | | | | | |
| USE UP & DOWN ARROW KEY TO HIGHLIGHT OPTION AND PRESS ENTER KEY TO EXECUTE | | | | | | | | | | | |

Fig. 2.0 Credit Manager Sub-Menu System

CODE MANAGEMENT

This menu option is used to predefine certain variables into the database file. This variable or information, are those data that are not likely to change for a long period of time. On pressing the ENTER key on this option (Code Management), it displays a submenu that contains the following option.

1. REGISTRATION OF LOAN TYPE
2. REGISTRATION OF BANK
3. PROJECT CATEGORY
4. EXIT TO MENU

Below is the general format of the Code Management option in fig.3.0

| AGRICULTURAL CREDIT GUARANTEE SCHEME (A Case Study of Central Bank of Nigeria - CBN Minna Branch) | | | |
|--|------------------|-------------------|------|
| CREDIT MANAGER | CODE MANAGEMENT | REPORT | QUIT |
| | REG.OF LOAN TYPE | CREATE NEW RECORD | |
| | REG. OF BANK | MODIFY RECORD | |
| | PROJECT CATEGORY | VIEW RECORD | |
| | EXIT TO MENU | DELETE RECORD | |
| | | EXIT TO MENU | |

USE UP & DOWN ARROW KEY TO HIGHLIGHT OPTION AND PRESS ENTER KEY TO EXECUTE

Fig. 3.0 Code Management Sub-Menu System

■ REPORT

This menu option is used to generate all the necessary reports as desired with respect to the output specification. This option is an interactive environment where the end user can decide to choose within two options (i.e. whether the report should be sent to SCREEN or PRINTER). Below are all the report expected to be generated.

- 1 REPORT ON TYPE OF LOAN
- 2 REPORT ON REGISTERED BANK
- 3 LIST OF PROJECT CATEGORY
- 4 LIST OF APPLICANTS REQUESTING FOR LOAN
- 5 LOAN REQUEST BY LOAN TYPE
- 6 LOAN REQUEST BY PROJECT CATEGORY
- 7 LOAN REPAYMENT & BALANCE STATUS

The general format of the Report menu option is as shown below in fig. 4.0

| AGRICULTURAL CREDIT GUARANTEE SCHEME (A Case Study of Central Bank of Nigeria - CBN Minna Branch) | | | |
|--|--|--------|------|
| CREDIT MANAGER | CODE MANAGEMENT | REPORT | QUIT |
| | <ol style="list-style-type: none"> 1. REPORT ON TYPE OF LOAN 2. REPORT ON REGISTERED BANK 3. LIST OF PROJECT CATEGORY 4. LIST OF APPLICATION REQUESTING FOR LOAN 5. REPORT ON LOAN REQUEST BY LOAN TYPE 6. REPORT ON LOAN REQUEST BY PROJECT CATEGORY 7. LOAN REPAYMENT AND BALANCE STATUS 8. EXIT TO MENU | | |
| USE UP & DOWN ARROW KEY TO HIGHLIGHT OPTION AND PRESS ENTER KEY TO EXECUTE | | | |

Fig. 4.0 Report Generation Sub-Menu System

3.7 COST BENEFIT ANALYSIS

The Cost Benefit Analysis, is the analysis of the total cost (Expenses) that is needed in order to put the ACGS into real implementation irrespective of people, and environmental constraints.

However, it is quite important to note that, cost benefit analysis at times is difficult to quantify at best, but must be done in order to estimate the financial and operational impact with the organisation (Case Study).

Below is the cost benefit analysis of implementing the ACGS system in a tabular form.

| S/NO | DESCRIPTION OF ITEM | UNIT | RATE | AMOUNT |
|-------------------------------|---|---------|---------|-------------------|
| (A) OPERATIONAL COST | | | | |
| 1. | Complete Computer System | 1 | 250,000 | 250,000.00 |
| 2. | HP Laser Jet 6L Printer | 1 | 65,000 | 65,000.00 |
| 3. | Epson LQ 2170 Printer | 1 | 95,000 | 95,000.00 |
| 4. | Stabiliser (Automatic Power Regulator) | 1 | 55,000 | 55,000.00 |
| 5. | UPS (Un-Interrupted Power Supply) | 1 | 65,000 | 65,000.00 |
| 6. | Equipment Maintenance | 1 | 70,000 | 70,000.00 |
| 7. | Miscellaneous Expenses | 1 | 20,000 | 20,000.00 |
| TOTAL ON ITEM (A) | | | | 620,000.00 |
| (B) DEVELOPMENTAL COST | | | | |
| 1. | System Analysis, Design and Program Development | Lot | 250,000 | 250,000.00 |
| 2. | ACGS Implementation | Lot | 50,000 | 50,000.00 |
| 3. | Personnel Training for 4 Weeks | 5 Staff | 10,000 | 10,000.00 |
| 4. | ACGS Installation | Lot | 25,000 | 25,000.00 |
| TOTAL ON ITEM (B) | | | | 375,000.00 |
| GRAND TOTAL (A) + (B) | | | | 995,000.00 |

BENEFITS TO BE DERIVED FROM THE PROPOSED

These includes all the benefit to be derived from the use of the new system. They are :

- ◆ **COMPREHENSIVENESS** This refers to the ease with which relevant users can understand what the system is doing and how it functions.
- ◆ **ADAPTABILITY** This is the extend, efficiency and ease with which the system can be adjusted to meet the changing needs of particular users or various users.
- ◆ **TOLERANCE** This refers to the degree to which the system is tolerant of users making slight variations and error in interaction.
- ◆ **PRECISE** This refers to the avoidance of ambiguity in information content and necessary user action.
- ◆ **HELPFULNESS** This is the amount of support given to assist users in performing required tasks.
- ◆ **FRIENDLINESS** This refers to the politeness of messages and dialogues.

CHAPTER FOUR

4.0 PROGRAMMING & IMPLEMENTATION REVIEW

4.1.0 PROGRAMMING

The next stage or challenge is to translate the system design and specification of the ACGS into an instruction that can be interpreted and executed by the computer. This, of course is the programming phase. The ideas formulated in previous chapter are translated into reality. The programming phase of the ACGS program developmental process can be divided into three (3) phases: -

4.1.1 PROGRAM IDENTIFICATION & DESCRIPTION

All the module programs needed for the achievement of the ACGS software are fully identified and described here. This includes the total number of program files, database files needed, report generated and so on.

The essence of this is to enable the writer/programmer determine the application package that will be suitable for the coding of the program. All the program files are available in the project appendix.

4.1.2 CHOICE OF PROGRAMMING LANGUAGE

The choice of programming language of the ACGS System is an important factor in any program development. From our previous analysis, it is suffice to say that the ACGS system is going to be used to store large volume of data/information, and time to time retrieval of records. Because of this fact, the choice of programming language chosen for the development is Database Management System (DBMS) Software with special preference to DBASE IV and CLIPPER 2.0.

Specifically, the choice of DBASE IV and CLIPPER 2.0 is because of the following features possessed by the packages.

1. It is user friendly.
2. It is easy to write an active user interface program and also simple to understand.
3. Individual database files can be designed to meet specific requirements of a particular functional unit of an organisation.
4. It minimizes data redundancy
5. Provide easy and greater access to information
6. Data integrity can be maintained
7. Clipper 2.0 allow the program file to compile to executable object file , thereby allowing the ACGS to be run independent of the Dbase IV and Clipper 2.0 that was used in developing the ACGS software.

4.1.3 **PROGRAM CODING**

Since we are using Dbase IV and Clipper 2.0 for developing the ACGS software, the program can either be Coded Or Written. This choice is a matter of personal preference since the two terms are the same. Here, the graphics and narative design of ACGS developmental stages previously mentioned are translated into machine-readable instruction, or programs.

However, since the analysis was based on top-down modular programming (i.e. fro complex to simple), the coding then becomes very easy, as they are already in a procedural form (modules) for easy and better understanding of the programmer.

4.2 **SYSTEM IMPLEMENTATION**

The essence of program testing is to make sure that the program is error-free and that all the logic involved are well defined and straight forward. On the other hand, system implementation is a broad term that encompasses testing and conversion. The system implementation involved testing

the ACGS with live data by several of the people who will eventually use the system. Testing with live data provides an extra level of assurance that the system will work properly when implemented.

However, once the testing has been completed, the ACGS enter the production stage (Usage) of the system life cycle, hence the ACGS is said to have become operational. Therefore, system implementation is a collective phases of program testing, troubleshooting, and lastly converting from the old system to the new system.

4.3 TESTING AND DEBUGGING OF ACGS

After the coding of the program into the system, it is likely to encounter at least one unexpected bug. A bug is either a syntax error (violation of one of the rules for writing instructions) or a logical error. Ridding a program of bugs is the process of debugging. In view of the above, the ACGS has been fully tested with real life data under different environment and people, and have been found to be efficient in terms of flow of data, speed and accuracy.

4.4 SYSTEM CONVERSION

Once the ACGS is implemented and goes on-line, the emphasis switches from development to operations. This then becomes a cooperative effort between the users and the case study personnel.

The achievement of the system conversion is based on the previous analysis which is a cooperative effort of both the users and the programmer. Here, the method to be used in the conversion process must have been clearly stated in order not to cause set back during program coding process. Thus, system conversion is the process of changing over from an existing old system (computerised or manual) to a newly developed system. Conversion can be implemented by using any of the four method mention below:-

1. Parallel changeover
2. Direct changeover
3. Staged changeover
4. Pilot changeover

Staged changeover was implemented for the ACGS conversion. This is because the program coding was based on the top-down modular programming thereby allowing the program to be written and run in a procedural form. This therefore means that one procedural program must run accurately and well tested for a particular stage before moving to another procedural stage.

Hence, staged changeover is the process of changing from one old stage to a new stage, this is actually implemented based on procedure (from one level to the next level). The final stage will never be reached until the previous stages have proved efficient and accurate.

4.5 **HARDWARE AND SOFTWARE REQUIREMENT**

In order to make maximum utilization of the ACGS system, certain hardware and software need to be acquired. This will include all the hardware components and the software components.

■ **HARDWARE REQUIREMENT**

The hardware facilities to be acquired must be those components that will stand the test of time, especially as we are about going into another millennium when virtually all hardware systems using Microprocessors may not function properly because of the date problems.

Hence, the system to be acquired should be in compliance with the year 2000. A system with the following minimum features is recommended:-

- ◆ PENTIUM 333MHZ CELERON INTEL
- ◆ 32MB RAM
- ◆ 43GB HDD

- ◆ 36X CD-ROM DRIVE +SOUND CARD+ 2 Nos of SPEAKERS
- ◆ SVGA 14"MONITOR
- ◆ ENHANCED KEYBOARD
- ◆ MICROSOFT MOUSE

The following peripherals/Accessories are needed along with the computer hardware. They are:-

- ◆ STABILIZER (Automatic Voltage Regulator)
- ◆ UPS (Un-Interrupted power supply)
- ◆ A PACKET OF 3.5" DISKETTES
- ◆ COMPUTER SYSTEM COVER

■ SOFTWARE REQUIREMENT

Software, which is the non-physical part differs significantly from hardware. Software is thus another term for the program that tell the hardware what to do. Without software, hardware is useless to anyone. Software can be compared to music residing on a cassette tapes or records, without the tapes, the stereo becomes useless to itself.

Therefore, in order to run the ACGS software and to be able to use the computer system for other official purposes, the following software need to be installed on the acquired hardware.

- ◆ Operating System Software (Windors "95 or 98 and MS-DOS 6.22)
- ◆ Application Software Packages
- ◆ WordPerfect for Windows
- ◆ Complete Office 97 Software
- ◆ Dbase IV & Clipper 2.0
- ◆ ACGS Software

However, other application software packages that can be installed on the system later depend on the desired choice of interest.

CHAPTER 5

5.0 GENERAL SUMMARY AND CONCLUSION

5.1 SUMMARY

This project work-The Impact of Computerized Agricultural Credit Guarantee Scheme (ACGS) on Agricultural Credit Delivery in Nigeria from (1978-1998) is written with a particular interest on Central Bank of Nigeria, Minna Branch as a case study. The project is summarised into five chapters for proper reference as explained below:-

Chapter one of this project work emphasised on introduction, Background of the study, statement of the problems, objectives of the study and limitation of the study.

Chapter two on the other hand, did not fail to discuss the literature Review, Meaning of Agricultural Credit, Types of Agricultural Credit, Source of Agricultural Credit and the Role of Agricultural Credit.

Also, chapter three which is believed to be the heart of the ACGS software development was fully discussed under the following heading: Concept of System Analysis and Design, process of program Development, understanding system Analysis, Design of ACGS and procedures (flowchart), input specification, output specification and structure of the ACGS software.

However, chapter four explained, programming and implementation, program Identification and Description, Choice of programming Language, System Implementation, Testing and Debugging of ACGS, System Conversion, Hardware and Software Requirement.

Lastly, chapter five which is the Summary and Conclusion, gives an overall summary of the whole project work of the five chapters.

5.2 CONCLUSION

The use of computer system in our daily life and various organisation have received increased attention in the recent years. One good reason for this keen interest is the general desire to improve and changeover from the old system (Manual method) to a new designed system (ACGS Software) of the CBN, Minna Branch in order to deal with the Management decision making and also improve the improper documentation of loan by storing them into the ACGS data base files.

Therefore, the ACGS manual method was fully analysed without leaving an aspect untouched all in the name to achieve the designation of a new system called the ACGS software. Hence, the ACGS have been tested, validated and have equally proved to be efficient in all ramifications.

5.3 RECOMMENDATION

Since the ACGS software have proved excellent. I therefore recommend the followings :

- (1) The Software (ACGS SOFTWARE) should be implemented in other branches of Central Bank of Nigeria in the country at large.
- (2) I also recommend that it should be networked as the Federal Government of Nigeria waited to increase the percentage of Agricultural Credit Scheme in Nigeria. The Network will indeed help in area of division of labour among the staff during the data entry process.

REFERENCES

1. BADAMASI R.O. System Analysis and Design Lectures Notes (Unpublished) FUT Minna.
2. CAROL B. & JILLI D (1996) Software Systems Deveopment 2nd Edition, Published by McGraw Hill International.
3. D.E. AVISION & G.FITZGERALD (1988) Information System Development 1st Edition, Published by Blackwell Scientific Publications, Oxford London, Edinburgh Boston Melbourne.
4. JOHNSON, D. J (1982) The Business of farming. A guide to farm Business Management in Tropics Macmillan Pub. Ltd London P. 69.
5. LEBCOW G. T. Dbase IV, the Complete Reference, Osborne, McGrae Hill, USA..
6. OKORIE A. (1986) "Major Determinants of Agricultural loan repayment". AERLS Journal on Savings and Development Vol. X No. 1 Page 80 - 99.
7. OGWUMA A.. P. A The role of Central Bank of Nigerian Economy CBN Economic and Financial Review Vol. 32 No. 4 (1994) Pae 359.
8. VON PISCHK J. D (1983) The Pitfall of specialised farm credit in Rural financial markets in Developed countries, Edited by Von Pischeka J. D Et al (c) John Hopkins P. 175 - 182 1983.

APPENDIX

*

*AUTHOR : ADEBOYEKU J. A.
*LICENSEE : CENTRAL BANK OF NIGERIA, MINNA BRANCH
*PROJECT TITLE : THE IMPACT OF AGRICULTURAL CREDIT
GUARANTEE SCHEME (ACGS) ON AGRICULTURAL
CREDIT DELIEVRY IN NIGERIA FRO M (1978-199~~8~~)

*

Set Talk off
Set Echo off
Set Bell off
Set Stat off
Set Scor off
Set Safe off
Set Conf off
Set Esca off
set proc to clap
Set date to brit
private sc_hpop // Saves horizontal popup
clear
*do floor
*do warn
x = inkey (2)
Set color to w+/br, ,
*do lfcarpet
*do lc1
do lc2
@ 00,00 clear to 05,79
@ 00,00 to 05,79 color "w+/r" ^{color}
cbn1="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME
(ACGS)"
cbn2="(A Case Study of Central Bank of Nigeria - CBN, Minna)"
@ 01,(80-len(cbn1))/2 say cbn1
@ 02,(80-len(cbn2))/2 say cbn2

Do while .t.

@21,08 clea to 23,71
@21,08 to 23,71
@22,10 say 'Use '+chr(25)+' or '+chr(24)+' key to highlight option & press
'+chr(17)+chr(196)+chr(217) +' key to select'
@ 04,01 prompt "CREDIT SCHEME"
@ 04,20 prompt "CODE MANAGEMENT"
@ 04,40 prompt "REPORT"
@ 04,55 prompt "UTILITY"

```
@ 04,70 prompt "Exit"  
menu to chl
```

```
Do case  
  case chl = 1  
    do CS  
  case chl = 2  
    do CM  
  case chl = 3  
    do REPorter  
  case chl = 4  
    do UTY  
  the  
  exit  
  case  
  addo
```

```
clear  
@23,01 say "CREDIT SCHEME, Undergoes Normal Shut Down...!!!"  
Return  
,
```

Procedure CS

```
Do while .t.  
set color to w+/b  
@ 06,00 clear to 12,24  
@ 06,00 to 12,24  
  
@21,08 clear to 23,71  
@21,08 to 23,71  
@22,10 say 'Use '+chr(25)+' or '+chr(24)+' key to highlight option & press  
+chr(17)+chr(196)+chr(217) +' key to select'  
@ 07,1 prompt " Loan Request.....»"  
@ 09,1 prompt " Loan Repayment.....»"  
@ 11,1 prompt " Exit Main Menu....."
```

```
menu to fchoice  
save screen to sc_vpop  
Do case  
  CASE fchoice = 0  
  loop  
  case fchoice = 1  
  do LReq  
  case fchoice = 2
```

```
do LRep
othe
  exit
Endcase
clea
Restore Screen from sc_vpop
Enddo
Return
```

*

Procedure LReq

Do while .t.

Set color to w+/b

@ 08,25 clear to 18,44

@ 08,25 to 18,44

@21,08 clear to 23,71

@21,08 to 23,71

@22,10 say 'Use '+chr(25)+' or '+chr(24)+' key to highlight option & press
'+chr(17)+chr(196)+chr(217) +' key to select'

@ 09,26 prompt " Creat New File.."

@ 11,26 prompt " Modify File....."

@ 13,26 prompt " View File....."

@ 15,26 prompt " Delete File....."

@ 17,26 prompt " Exit Menu....."

menu to ch2

Save screen

Do case

case ch2 = 0

loop

case ch2 = 1

do clreq

case ch2 = 2

do mlreq

case ch2 = 3

do vlreq

case ch2 = 4

do dlreq

other

exit

Endcase

Rest Screen
Enddo
Return
*

Procedure LRep

Do while .t.

Set color to w+/b

@ 08,25 clear to 18,44

@ 08,25 to 18,44

@21,08 clear to 23,71

@21,08 to 23,71

@22,10 say 'Use '+chr(25)+' or '+chr(24)+' key to highlight option & press
'+chr(17)+chr(196)+chr(217) +' key to select'

@ 09,26 prompt " Creat New File.."

@ 11,26 prompt " Modify File....."

@ 13,26 prompt " View File....."

@ 15,26 prompt " Delete File....."

@ 17,26 prompt " Exit Menu....."

menu to bchoice

Save screen

Do case

case bchoice = 1

do clrep

case bchoice = 2

do mlrep

case bchoice = 3

do vlrep

case bchoice = 4

do dlrep

case bchoice = 0

loop

other

exit

Endcase

clea

Restore Screen

Enddo

Return

*

Procedure CM


```
private sc_vpop
Do while .t.
Set color to w+/b
@ 06,20 clear to 12,44
@ 06,20 to 12,44

@21,08 clear to 23,71
@21,08 to 23,71
@22,10 say 'Use '+chr(25)+' or '+chr(24)+' key to highlight option & press
'+chr(17)+chr(196)+chr(217) +' key to select'
@ 07,21 prompt " Regist.of Loan Type.»"
@ 09,21 prompt " Registration of Bank.»"
@ 11,21 prompt " Exit Main Menu....."
```

```
menu to fchoice
Save screen to sc_vpop
Do case
  CASE fchoice = 0
    loop
  case fchoice = 1
    do RLT
  case fchoice = 2
    do RB
  othe
  exit
Endcase
clea
Restore Screen from sc_vpop
Enddo
Return
*
```

```
Procedure RLT
Do while .t.
Set color to w+/b
```

```
@ 08,45 clear to 18,63
@ 08,45 to 18,63
```

```
@21,08 clear to 23,71
@21,08 to 23,71
@22,10 say 'Use '+chr(25)+' or '+chr(24)+' key to highlight option & press
'+chr(17)+chr(196)+chr(217) +' key to select'
```

@ 09,46 prompt " Creat New File.."
@ 11,46 prompt " Modify File....."
@ 13,46 prompt " View File....."
@ 15,46 prompt " Delete File....."
@ 17,46 prompt " Exit Menu....."

menu to tchoice

Save screen

Do case

case tchoice = 1

do crlt

case tchoice = 2

do mrlt

case tchoice = 3

do vrlt

case tchoice = 4

do drlt

case tchoice = 0

loop

other

exit

Endcase

Restore Screen

Enddo

Return

*

Procedure RB

Do while .t.

Set color to w+/b

@ 08,45 clear to 18,63

@ 08,45 to 18,63

@21,08 clear to 23,71

@21,08 to 23,71

@22,10 say 'Use '+chr(25)+' or '+chr(24)+' key to highlight option & press
'+chr(17)+chr(196)+chr(217)+' key to select'

@ 09,46 prompt " Creat New File.."

@ 11,46 prompt " Modify File....."

@ 13,46 prompt " View File....."

@ 15,46 prompt " Delete File....."

@ 17,46 prompt " Exit Menu....."

```
menu to tchoice
Save screen
Do case
  case tchoice = 1
    do crb
  case tchoice = 2
    do mrb
  case tchoice = 3
    do vrb
  case tchoice = 4
    do drb
  case tchoice = 0
    loop
  other
  exit
Endcase
Restore Screen
Enddo
Return
*
```

Procedure reporter

```
private sc_vpop
```

```
Do while .t.
```

```
Set color to w+/b
```

```
@ 06,29 clear to 21,78
```

```
@ 06,29 to 21,78
```

```
@22,08 clear to 24,68
```

```
@22,08 to 24,68
```

```
@23,10 say 'Use '+chr(25)+' or '+chr(24)+' key to highlight option & press  
'+chr(17)+chr(196)+chr(217) +' key to select'
```

```
@ 07,30 prompt " 1. Report on Applicants Requesting for Loan"
```

```
@ 09,30 prompt " 2. Report on Approved Loan to Applicants..."
```

```
@ 11,30 prompt " 3. Report on Individual Loan Issued Out..."
```

```
@ 13,30 prompt " 4. Report on Informal Loan Issued Out....."
```

```
@ 15,30 prompt " 5. Report on Co-operative Loan Issued Out.."
```

```
@ 17,30 prompt " 6. Report on Company Loan Out....."
```

```
@ 19,30 prompt " 7. List of Applicants/Certificate Number..."
```

```
@ 21,30 prompt " 8. Exit Main Menu....."
```

```
menu to fchoice
```

```
Save screen to sc_vpop
```

```
Do case
  CASE fchoice = 0
    loop
  case fchoice = 1
    do rep1
  case fchoice = 2
    do rep2
  case fchoice = 3
    do rep3
  case fchoice = 4
    do rep4
  case fchoice = 5
    do rep5
  case fchoice = 6
    do rep6
  case fchoice = 7
    do rep7
  othe
  exit
Endcase
clea
Restore Screen from sc_vpop
Enddo
Return
*
```

Procedure UTY

```
private sc_vpop
Do while .t.
Set color to w+/b
@ 06,55 clear to 14,77
@ 06,55 to 14,77

@21,08 clear to 23,71
@21,08 to 23,71
@22,10 say 'Use '+chr(25)+' or '+chr(24)+' key to highlight option & press
'+chr(17)+chr(196)+chr(217) +' key to select'
@ 07,56 prompt " Introduction.....»"
@ 09,56 prompt " Backup Files.....»"
@ 11,56 prompt " Restore Files.....»"
@ 13,56 prompt " Exit Main Menu....."

menu to fchoice
```

```
Save screen to sc_vpop
Do case
  CASE fchoice = 0
    loop
  case fchoice = 1
    !attrib +r intro.txt
    !edit intro.txt
  case fchoice = 2
    do bf
  case fchoice = 3
    do RFile
  othe
  exit
Endcase
clea
Restore Screen from sc_vpop
Enddo
Return
```

----- PROCEDURE CLREQ -----

```
proc clreq
clear
set color to w+/b
set proc to header
do head
do xpublic
do initial
subh="Registration of Customer's Requesting for Loan"
@05,(80-len(subh))/2 say subh
Do while .t.
  do while .t. &&----- Testing and Identifying Loan TYpe
    xloancode=space(02)
    fb="ENTER LOAN CODE or Press Enter Key on Loan Code to Exit"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    @09,05 say "ENTER LOAN CODE FOR THE LOAN TYPE : " get xloancode pict "99"
    read
    if xloancode=space(02)
      clear
      close all
      return
    endif
    use
    use type
    index on loancode to loancode
    go top
    seek xloancode
    if .not. found()
      fb="LOAN CODE DOES NOT EXIST, Press Any Key To Retry"
      @23,02 clea to 23,77
      @23,(80-len(fb))/2 say fb
      set cons off
      wait " "
      @23,02 clea to 23,77
      xloancode=space(02)
      loop
    endif
    xloandesc=loandesc
    @11,05 say "LOAN DESCRIPTION          : " get xloandesc pict "@!"
    clea gets
    exit
  enddo
  xregdate=dtoc(date())
  xregtime=time()
  set cons off
  @09,05 clear to 11,77
  set cons on
  do getol
do while .t.
```

```

use coder
index on requestno to coder
go bott
xrequestno=requestno
if xrequestno=0
  xrequestno=1
else
  xrequestno=xrequestno+1
endif
exit
enddo

```

----- Decision Making on Type of Loan -----

```

if xloancode="01"
  do indivi
endif
if xloancode="02"
  do company
endif
if xloancode="03"
  do informal
endif
if xloancode="04"
  do corp
endif

```

----- Decision Making on Saving -----

```

@22,02 clea to 23,77
set proc to saver
do boye
  x=0
  do while x=0
    x=inkey()
    if upper (chr(x)) $ "SRA"
      exit
    endif
    x=0
  enddo
  if upper (chr(x)) $ "R"
    @07,03 clea to 19,77
    @22,02 clea to 23,77
    fb="ENTER LOAN CODE or Press Enter Key on Code to Exit"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    xrequestno=xrequestno-1
  loop
endif
if upper (chr(x)) $ "A"
  clear
  close all
  return
endif

```

```

use
use coder
index on requestno to coder
go bott
appe blan
repl regdate with date ()
repl regtime with xregtime
repl requestno with xrequestno
repl name with xname
repl address with xaddress

if xshortname="FBN"
use FBN
index on fbn_no to fbn_no
appe blan
repl regdate with date ()
repl regtime with xregtime
repl requestno with xrequestno
repl name with xname
repl fbn_no with xbank_no
endif
if xshortname="UBN"
use UBN
index on ubn_no to ubn_no
appe blan
repl regdate with date ()
repl regtime with xregtime
repl requestno with xrequestno
repl name with xname
repl ubn_no with xbank_no
endif
if xshortname="UBA"
use UBA
index on uba_no to uba_no
appe blan
repl regdate with date ()
repl regtime with xregtime
repl requestno with xrequestno
repl name with xname
repl uba_no with xbank_no
endif
if xshortname="ABN"
use ABN
index on abn_no to abn_no
appe blan
repl regdate with date ()
repl regtime with xregtime
repl requestno with xrequestno
repl name with xname
repl abn_no with xbank_no

```



```

endif
if xshortname="GTB"
  use GTB
  index on gtb_no to gtb_no
  appe blan
  repl regdate with date ()
  repl regtime with xregtime
  repl requestno with xrequestno
  repl name with xname
  repl gtb_no with xbank_no
endif

```

```

use
use request
appen blan
do replreq
use

```

```
@07,03 clea to 19,77
```

```
@22,02 clea to 23,77
```

```
Enddo
```

```
retu
```

```
*----- PROCEDURE INDIVI -----*
```

```
proc indivi
```

```
* do getol
```

```
@07,33 get xrequestno pict "9999"
```

```
@07,40 get xloancode pict "99"
```

```
@07,45 get xloandesc pict "@"
```

```
clea gets
```

```
do while .t.
```

```
fb="ENTER INDIVIDUAL NAME"
```

```
@23,02 clea to 23,77
```

```
@23,(80-len(fb))/2 say fb
```

```
@09,33 get xname pict "@"
```

```
read
```

```
if xname=space(30)
```

```
fb="INDIVIDUAL NAME CANNOT BE EMPTY, Press Any Key To Retry"
```

```
@23,02 clea to 23,77
```

```
@23,(80-len(fb))/2 say fb
```

```
set cons off
```

```
wait " "
```

```
@23,02 clea to 23,77
```

```
loop
```

```
endif
```

```
exit
```

```
enddo
```

```
do while .t.
```

```
fb="ENTER INDIVIDUAL ADDRESS"
```

```
@23,02 clea to 23,77
```

```
@23,(80-len(fb))/2 say fb
```

```

@11,33 get xaddress pict "@!"
read
if xaddress=space(20)
  fb="INDIVIDUAL ADDRESS CANNOT BE EMPTY, Press Any Key To Retry"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  set cons off
  wait " "
  @23,02 clea to 23,77
  loop
endif
exit
enddo
do while .t.
  fb="ENTER PROJECT LOCATION"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @11,55 get xlocation pict "@!"
  read
  if xlocation=space(20)
    fb="PROJECT LOCATION CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
  endif
  exit
enddo
do while .t.
  fb="ENTER BANK CODE"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @13,33 get xbankcode pict "@!"
  read
  if xbankcode=space(02)
    fb="BANK CODE CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
  endif
  use
  use bank
  index on bankcode to bankcode
  seek xbankcode
  if .not. found()

```

```

fb="INVALID BANK CODE, Press Any Key To Retry"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
set cons off
wait " "
@23,02 clea to 23,77
loop
endif
xbankname=bankname
xshortname=shortname
@13,37 get xbankname pict "@"!
@13,70 get xshortname pict "@"!
clea gets
exit
enddo

```

----- Generating Applicant Account/Certificate Number-----

```

if xshortname="FBN"
  use FBN
  index on fbn_no to fbn_no
  go bott
  xfbn_no=fbn_no
  if xfbn_no=0
    xfbn_no=1
  else
    xfbn_no=xfbn_no+1
  endif
  xbank_no=xfbn_no
endif
if xshortname="UBN"
  use ubn
  index on ubn_no to ubn_no
  go bott
  xubn_no=ubn_no
  if xubn_no=0
    xubn_no=1
  else
    xubn_no=xubn_no+1
  endif
  xbank_no=xubn_no
endif
if xshortname="UBA"
  use uba
  index on uba_no to uba_no
  go bott
  xuba_no=uba_no
  if xuba_no=0
    xuba_no=1
  else
    xuba_no=xuba_no+1
  endif
endif

```

```

    xbank_no=xuba_no
endif
if xshortname="^BN"
    use abn
    index on abn_no to abn_no
    go bott
    xabn_no=abn_no
    if xabn_no=0
        xabn_no=1
    else
        xabn_no=xabn_no+1
    endif
    xbank_no=xabn_no
endif
if xshortname="GTB"
    use gtb
    index on gtb_no to gtb_no
    go bott
    xgtb_no=gtb_no
    if xgtb_no=0
        xgtb_no=1
    else
        xgtb_no=xgtb_no+1
    endif
    xbank_no=xgtb_no
endif

xacctnumber="NG"+"/"+xshortname+"/"+"99"+"/"+ltrim(str(xbank_no))+"/"+ltrim(str(xrequestno))
@15,33 get xacctnumber pict "@"!
clea gets
do while .t.
    fb="ENTER LOAN DURATION IN MONTHS"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    @15,55 get xduration pict "99"
    read
    if xduration=0
        fb="LOAN DURATION CODE CANNOT BE EMPTY, Press Any Key To Retry"
        @23,02 clea to 23,77
        @23,(80-len(fb))/2 say fb
        set cons off
        wait " "
        @23,02 clea to 23,77
        loop
    endif
    exit
enddo
do while .t.
    fb="ENTER PROJECT CATEGORY CODE"
    @23,02 clea to 23,77

```

```

@23,(80-len(fb))/2 say fb
@17,33 get xcat_code pict "@"!
read
if xcat_code=space(02)
  fb="PROJECT CATEGORY CODE CANNOT BE EMPTY, Press Any Key To Retry"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  set cons off
  wait " "
  @23,02 clea to 23,77
  loop
endif
use
use category
index on cat_code to cat_code
seek xcat_code
if .not. found()
  fb="INVALID PROJECT CATEGORY CODE, Press Any Key To Retry"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  set cons off
  wait " "
  @23,02 clea to 23,77
  loop
endif
xcat_desc=cat_desc
@17,37 get xcat_desc pict "@"!
clea gets
exit
enddo
do while .t.
  fb="ENTER AMOUNT REQUESTED"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @19,33 get xamt_wanted pict "999,999,999.99"
  read
  if xamt_wanted=0.00
    fb="AMOUNT REQUESTED CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
  endif
  exit
enddo
do while .t.
  fb="ENTER AMOUNT APPROVED"
  @23,02 clea to 23,77

```

```

@23,(80-len(fb))/2 say fb
@19,50 get xamt_given pict "999,999,999.99"
read
if xamt_given=0.00
  fb="AMOUNT APPROVED CANNOT BE EMPTY, Press Any Key To Retry"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  set cons off
  wait " "
  @23,02 clea to 23,77
  loop
endif
exit
enddo
do while .t.
  fb="REMARKS : [A] => Approved or [N] => Not Approved"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @19,69 get xremark pict "!"
  read
  if xremark="A" .or. xremark="N"
    exit
  endif
enddo
return

```

----- PROCEDURE INFORMAL -----

```

proc informal
  @07,33 get xrequestno pict "9999"
  @07,40 get xloancode pict "99"
  @07,45 get xloandesc pict "@!"
  clea gets
do while .t.
  fb="ENTER INFORMAL NAME"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @09,33 get xname pict "@!"
  read
  if xname=space(30)
    fb="INFORMAL NAME CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
  endif
  exit
enddo
do while .t.
  fb="ENTER INFORMAL ADDRESS"

```

```

@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
@11,33 get xaddress pict "@"!
read
if xaddress=space(20)
fb="INFORMAL ADDRESS CANNOT BE EMPTY, Press Any Key To Retry"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
set cons off
wait " "
@23,02 clea to 23,77
loop
endif
exit
enddo
do while .t.
fb="ENTER PROJECT LOCATION"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
@11,55 get xlocation pict "@"!
read
if xlocation=space(20)
fb="PROJECT LOCATION CANNOT BE EMPTY, Press Any Key To Retry"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
set cons off
wait " "
@23,02 clea to 23,77
loop
endif
exit
enddo
do while .t.
fb="ENTER BANK CODE"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
@13,33 get xbankcode pict "@"!
read
if xbankcode=space(02)
fb="BANK CODE CANNOT BE EMPTY, Press Any Key To Retry"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
set cons off
wait " "
@23,02 clea to 23,77
loop
endif
use
use bank
index on bankcode to bankcode

```

```

seek xbankcode
if .not. found()
  fb="INVALID BANK CODE, Press Any Key To Retry"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  set cons off
  wait " "
  @23,02 clea to 23,77
  loop
endif
xbankname=bankname
xshortname=shortname
@13,37 get xbankname pict "@!"
@13,70 get xshortname pict "@!"
clea gets
exit
enddo

```

----- Generating Applicant Account/Certificate Number-----

```

if xshortname="FBN"
  use FBN
  index on fbn_no to fbn_no
  go bott
  xfbn_no=fbn_no
  if xfbn_no=0
    xfbn_no=1
  else
    xfbn_no=xfbn_no+1
  endif
  xbank_no=xfbn_no
endif
if xshortname="UBN"
  use ubn
  index on ubn_no to ubn_no
  go bott
  xubn_no=ubn_no
  if xubn_no=0
    xubn_no=1
  else
    xubn_no=xubn_no+1
  endif
  xbank_no=xubn_no
endif
if xshortname="UBA"
  use uba
  index on uba_no to uba_no
  go bott
  xuba_no=uba_no
  if xuba_no=0
    xuba_no=1
  else

```



```

    xuba_no=xuba_no+1
endif
    xbank_no=xuba_no
endif
if xshortname="ABN"
    use abn
    index on abn_no to abn_no
    go bott
    xabn_no=abn_no
    if xabn_no=0
        xabn_no=1
    else
        xabn_no=xabn_no+1
    endif
    xbank_no=xabn_no
endif
if xshortname="GTB"
    use gtb
    index on gtb_no to gtb_no
    go bott
    xgtb_no=gtb_no
    if xgtb_no=0
        xgtb_no=1
    else
        xgtb_no=xgtb_no+1
    endif
    xbank_no=xgtb_no
endif

```

```

xacctnumber="NG"+"/"+xshortname+"/"+"99"+"/"+ltrim(str(xbank_no))+"/"+ltrim(str(xrequestno))
@15,33 get xacctnumber pict "@"
clea gets
do while .t.
    fb="ENTER LOAN DURATION IN MONTHS"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    @15,55 get xduration pict "99"
    read
    if xduration=0
        fb="LOAN DURATION CODE CANNOT BE EMPTY, Press Any Key To Retry"
        @23,02 clea to 23,77
        @23,(80-len(fb))/2 say fb
        set cons off
        wait " "
        @23,02 clea to 23,77
        loop
    endif
    exit
enddo
do while .t.

```

```

fb="ENTER PROJECT CATEGORY CODE"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
@17,33 get xcat_code pict "@"!
read
if xcat_code=space(02)
  fb="PROJECT CATEGORY CODE CANNOT BE EMPTY, Press Any Key To Retry"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  set cons off
  wait " "
  @23,02 clea to 23,77
  loop
endif
use
use category
index on cat_code to cat_code
seek xcat_code
if .not. found()
  fb="INVALID PROJECT CATEGORY CODE, Press Any Key To Retry"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  set cons off
  wait " "
  @23,02 clea to 23,77
  loop
endif
xcat_desc=cat_desc
@17,37 get xcat_desc pict "@"!
clea gets
exit
enddo
do while .t.
  fb="ENTER AMOUNT REQUESTED"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @19,33 get xamt_wanted pict "999,999,999.99"
  read
  if xamt_wanted=0.00
    fb="AMOUNT REQUESTED CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
  endif
  exit
enddo
do while .t.

```

```

fb="ENTER AMOUNT APPROVED"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
@19,50 get xamt_given pict "999,999,999.99"
read
if xamt_given=0.00
  fb="AMOUNT APPROVED CANNOT BE EMPTY, Press Any Key To Retry"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  set cons off
  wait " "
  @23,02 clea to 23,77
  loop
endif
exit
enddo
do while .t.
  fb="REMARKS : [A] => Approved or [N] => Not Approved"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @19,69 get xremark pict "!"
  read
  if xremark="A" .or. xremark="N"
    exit
  endif
enddo
do while .t.
  fb="ENTER AMOUNT REQUESTED BY 1st MEMBER"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @09,48 get xamt_want1 pict "9,999,999.99"
  read
  if xamt_want1=0.00
    fb="AMOUNT REQUESTED CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
  endif
  exit
enddo
do while .t.
  fb="ENTER AMOUNT APPROVED TO 1st MEMBER"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @09,65 get xamt_paid1 pict "9,999,999.99"
  read
  if xamt_paid1=0.00

```

```

fb="AMOUNT APPROVED CANNOT BE EMPTY, Press Any Key To Retry"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
set cons off
wait " "
@23,02 clea to 23,77
loop
endif
exit
enddo
*----- 2nd Member -----
xcode2=xcode1+1
@11,07 get xcode2 pict "9"
clea gets
do while .t.
fb="ENTER 2nd MEMBER NAME"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
@11,17 get xname2 pict "@!"
read
if xname2=space(20)
fb="2nd MEMBER NAME CANNOT BE EMPTY, Press Any Key To Retry"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
set cons off
wait " "
@23,02 clea to 23,77
loop
endif
exit
enddo
do while .t.
fb="ENTER AMOUNT REQUESTED BY 2nd MEMBER"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
@11,48 get xamt_want2 pict "9,999,999.99"
read
if xamt_want2=0.00
fb="AMOUNT REQUESTED CANNOT BE EMPTY, Press Any Key To Retry"
@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
set cons off
wait " "
@23,02 clea to 23,77
loop
endif
exit
enddo
do while .t.
fb="ENTER AMOUNT APPROVED TO 2nd MEMBER"

```

```

@23,02 clea to 23,77
@23,(80-len(fb))/2 say fb
@11,65 get xamt_paid2 pict "9,999,999.99"
read
if xamt_paid2=0.00
  fb="AMOUNT APPROVED CANNOT BE EMPTY, Press Any Key To Retry"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  set cons off
  wait " "
  @23,02 clea to 23,77
  loop
endif
exit
enddo
*----- 3rd Member -----
xcode3=xcode2+1
@13,07 get xcode3 pict "9"
clea gets
do while .t.
  fb="ENTER 3rd MEMBER NAME"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @13,17 get xname3 pict "@!"
  read
  if xname3=space(20)
    fb="3rd MEMBER NAME CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
  endif
  exit
enddo
do while .t.
  fb="ENTER AMOUNT REQUESTED BY 3rd MEMBER"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @13,48 get xamt_want3 pict "9,999,999.99"
  read
  if xamt_want3=0.00
    fb="AMOUNT REQUESTED CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
  endif
  exit
enddo

```

```

endif
exit
enddo
do while .t.
  fb="ENTER AMOUNT APPROVED TO 3rd MEMBER"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @13,65 get xamt_paid3 pict "9,999,999.99"
  read
  if xamt_paid3=0.00
    fb="AMOUNT APPROVED CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
  endif
endif
exit
enddo
*----- 4th Member -----
xcode4=xcode3+1
@15,07 get xcode4 pict "9"
clea gets
do while .t.
  fb="ENTER 4th MEMBER NAME"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @15,17 get xname4 pict "@!"
  read
  if xname4=space(20)
    fb="4th MEMBER NAME CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
  endif
endif
exit
enddo
do while .t.
  fb="ENTER AMOUNT REQUESTED BY 4th MEMBER"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @15,48 get xamt_want4 pict "9,999,999.99"
  read
  if xamt_want4=0.00
    fb="AMOUNT REQUESTED CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77

```

```

    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
endif
exit
enddo
do while .t.
    fb="ENTER AMOUNT APPROVED TO 4th MEMBER"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    @15,65 get xamt_paid4 pict "9,999,999.99"
    read
    if xamt_paid4=0.00
        fb="AMOUNT APPROVED CANNOT BE EMPTY, Press Any Key To Retry"
        @23,02 clea to 23,77
        @23,(80-len(fb))/2 say fb
        set cons off
        wait " "
        @23,02 clea to 23,77
        loop
    endif
    exit
enddo
*----- 5nd Member -----
xcode5=xcode4+1
@17,07 get xcode5 pict "9"
clea gets
do while .t.
    fb="ENTER 5th MEMBER NAME"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    @17,17 get xname5 pict "@!"
    read
    if xname5=space(20)
        fb="5th MEMBER NAME CANNOT BE EMPTY, Press Any Key To Retry"
        @23,02 clea to 23,77
        @23,(80-len(fb))/2 say fb
        set cons off
        wait " "
        @23,02 clea to 23,77
        loop
    endif
    exit
enddo
do while .t.
    fb="ENTER AMOUNT REQUESTED BY 5th MEMBER"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb

```

```

@17,48 get xamt_want5 pict "9,999,999.99"
read
if xamt_want5=0.00
  fb="AMOUNT REQUESTED CANNOT BE EMPTY, Press Any Key To Retry"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  set cons off
  wait " "
  @23,02 clea to 23,77
  loop
endif
exit
enddo
do while .t.
  fb="ENTER AMOUNT APPROVED TO 5th MEMBER"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  @17,65 get xamt_paid5 pict "9,999,999.99"
  read
  if xamt_paid5=0.00
    fb="AMOUNT APPROVED CANNOT BE EMPTY, Press Any Key To Retry"
    @23,02 clea to 23,77
    @23,(80-len(fb))/2 say fb
    set cons off
    wait " "
    @23,02 clea to 23,77
    loop
  endif
  exit
enddo
totalpaid=(xamt_paid1+xamt_paid2+xamt_paid3+xamt_paid4+xamt_paid5)
@19,57 say "TOTAL : "get totalpaid pict "9,999,999.99" colo "+w/r*"
clea gets
if totalpaid > xamt_given
  fb="AMOUNT PAID TO MEMBERS CANNOT EXCEED TOTAL AMOUNT APPROVI
Press Any Key To Retry"
  @23,02 clea to 23,77
  @23,(80-len(fb))/2 say fb
  set cons off
  wait " "
  @23,02 clea to 23,77
  loop
endif
exit
enddo
return

```

*-----PROC LOANER-----

proc loaner &&-----REP1 is Converted to Loaner.prg


```

@21,08 clea to 23,71
@21,08 to 23,71
outer="[S]creen          [P]rinter"
@22,(80-len(outer))/2 say outer
x=0
do while x=0
  x=inkey()
  if upper (chr(x)) $ "SP"
    exit
  endif
  x=0
enddo
if upper (chr(x)) $ "S" && ----- Generating Screen Printng-----

  use type
  index on loancode to loancode
  go top

@21,08 clea to 23,71
@21,08 to 23,71
outer="[P]rinting        [C]ancel"
@22,(80-len(outer))/2 say outer
x=0
do while x=0
  x=inkey()
  if upper (chr(x)) $ "PC"
    exit
  endif
  x=0
enddo
if upper (chr(x)) $ "C"
  clear
  close all
  return
endif
clea

  set device to screen
  fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)"
  souter="LIST OF ALL AVAILABLE LOAN TYPE"
  @01,(80-LEN(fouter))/2 say fouter
  @02,(80-LEN(souter))/2 say souter
  @04,01 say "S/NO"
  @04,07 say "REG. DATE"
  @04,20 say "REG. TIME"
  @04,33 say "LOAN CODE"
  @04,45 say "LOAN DESCRIPTION"
  @05,01 say repl ("=",77)
  xpage=1
  row=06
  mrow=62

```

```

xsno=1
go top

xregdate=ctod(" / / ")
xregtime=space(08)
xloancode=space(02)
xloandesc=space(30)
do while .not. eof()
  if row>mrow
    eject
    clear
    xpage=xpage+1
    row=08
    set device to SCREEN
    fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)"
    souter="LIST OF ALL AVAILABLE LOAN TYPE"
    @01,(80-LEN(fouter))/2 say fouter
    @02,(80-LEN(souter))/2 say souter
    @04,01 say "S/NO"
    @04,07 say "REG. DATE"
    @04,20 say "REG. TIME"
    @04,33 say "LOAN CODE"
    @04,45 say "LOAN DESCRIPTION"
    @05,01 say repl ("=",77)
  endif
  xregdate=regdate
  xregtime=regtime
  xloancode=loancode
  xloandesc=loandesc
  @row,01 say xsno      pict "999"
  @row,07 say xregdate  pict "99/99/99"
  @row,20 say xregtime  pict "99:99:99"
  @row,33 say xloancode pict "@!"
  @row,45 say xloandesc pict "@!"
  row=row+2
  xsno=xsno+1

skip
enddo
use
jobend="ACGS HAS COMPLETED PRINTING JOB, Press Any Key To Continue..."
@22,02 clea to 23,77
@22,05 say "MESSAGE" color "w+*/r"
@23,(80-len(jobend))/2 say jobend
set cons off
wait " "
set cons on
clea
return
endif
if upper (chr(x)) $ "P" && ----- Generating Screen Printng-----

```

```
use type
index on loancode to loancode
go top
```

```
@21,08 clea to 23,71
@21,08 to 23,71
outer="[P]rinting      [C]ancel"
@22,(80-len(outer))/2 say outer
x=0
do while x=0
  x=inkey()
  if upper (chr(x)) $ "PC"
    exit
  endif
  x=0
enddo
if upper (chr(x)) $ "C"
  clear
  close all
  return
endif
clea
  set device to print
  fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)"
  souter="LIST OF ALL AVAILABLE LOAN TYPE"
  @01,(80-LEN(fouter))/2 say fouter
  @02,(80-LEN(souter))/2 say souter
  @04,01 say "S/NO"
  @04,07 say "REG. DATE"
  @04,20 say "REG. TIME"
  @04,33 say "LOAN CODE"
  @04,45 say "LOAN DESCRIPTION"
  @05,01 say repl ("=",77)
  xpage=1
  row=06
  mrow=62
  xsno=1
  go top

  xregdate=ctod(" / / ")
  xregtime=space(08)
  xloancode=space(02)
  xloandesc=space(30)
do while .not. eof()
  if row>mrow
    eject
    clear
    xpage=xpage+1
    row=08
    set device to print
```

```

fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS
souter="LIST OF ALL AVAILABLE LOAN TYPE"
@01,(80-LEN(fouter))/2 say fouter
@02,(80-LEN(souter))/2 say souter
@04,01 say "S/NO"
@04,07 say "REG. DATE"
@04,20 say "REG. TIME"
@04,33 say "LOAN CODE"
@04,45 say "LOAN DESCRIPTION"
@05,01 say repl ("=",77)
endif
  xregdate=regdate
  xregtime=regtime
  xloancode=loancode
  xloandesc=loandesc
  @row,01 say xsno      pict "999"
  @row,07 say xregdate  pict "99/99/99"
  @row,20 say xregtime  pict "99:99:99"
  @row,33 say xloancode pict "@!"
  @row,45 say xloandesc pict "@!"
  row=row+2
  xsno=xsno+1
skip
enddo
use
eject
jobend="ACGS HAS COMPLETED PRINTING JOB, Press Any Key To Continue..."
@22,02 clea to 23,77
@22,05 say "MESSAGE" color "w+*/r"
@23,(80-len(jobend))/2 say jobend
set cons off
wait " "
set cons on
clea
return
endif

```

```

*-----PROC Banker-----
proc banker
@21,08 clea to 23,71
@21,08 to 23,71
outer="[S]creen      [P]rinter"
@22,(80-len(outer))/2 say outer
x=0
do while x=0
  x=inkey()
  if upper (chr(x)) $ "SP"
    exit
  endif
enddo
x=0

```

```

        enddo
if upper (chr(x)) $ "S" && ----- Generating Screen Printng-----

        use bank
        index on bankcode to bankcode
        go top

        @21,08 clea to 23,71
        @21,08 to 23,71
        outer="[P]rinting          [C]ancel"
        @22,(80-len(outer))/2 say outer
        x=0
        do while x=0
            x=inkey()
            if upper (chr(x)) $ "PC"
                exit
            endif
            x=0
        enddo
if upper (chr(x)) $ "C"
        clear
        close all
        return
endif
clea
        set device to screen
        fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)"
        souter="LIST OF REGISTERED BANK FOR CASH TRANSFER"
        @01,(80-LEN(fouter))/2 say fouter
        @02,(80-LEN(souter))/2 say souter
        @04,01 say "S/NO"
        @04,07 say "REG. DATE"
        @04,20 say "REG. TIME"
        @04,33 say "BANK CODE"
        @04,50 say "BANK NAME"
        @04,70 say "SHORTNAME"
        @05,01 say repl ("=",77)
        xpage=1
        row=06
        mrow=62
        xsno=1
        go top

        xregdate=ctod(" / / ")
        xregtime=space(08)
        xbankcode=space(02)
        xbankname=space(30)
        xshortname=space(03)
do while .not. eof()
        if row>mrow

```

```

eject
clear
xpage=xpage+1
row=08
set device to SCREEN
  fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)"
  souter="LIST OF REGISTERED BANK FOR CASH TRANSFER"
  @01,(80-LEN(fouter))/2 say fouter
  @02,(80-LEN(souter))/2 say souter
  @04,01 say "S/NO"
  @04,07 say "REG. DATE"
  @04,20 say "REG. TIME"
  @04,33 say "BANK CODE"
  @04,50 say "BANK NAME"
  @04,70 say "SHORTNAME"
  @05,01 say repl ("=",77)
endif
  xregdate=regdate
  xregtime=regtime
  xbankcode=bankcode
  xbankname=bankname
  xshortname=shortname
  @row,01 say xsno      pict "999"
  @row,07 say xregdate  pict "99/99/99"
  @row,20 say xregtime  pict "99:99:99"
  @row,35 say xbankcode pict "@!"
  @row,42 say xbankname pict "@!"
  @row,72 say xshortname pict "@!"
  row=row+2
  xsno=xsno+1
skip
enddo
use
jobend="ACGS HAS COMPLETED PRINTING JOB, Press Any Key To Continue..."
@22,02 clea to 23,77
@22,05 say "MESSAGE" color "w+*/r"
@23,(80-len(jobend))/2 say jobend
set cons off
wait " "
set cons on
clea
return
endif
if upper (chr(x)) $ "P" && ----- Generating Screen Printng-----

  use bank
  index on bankcode to bankcode
  go top

  @21,08 clea to 23,71

```

```

@21,08 to 23,71
outer="[P]rinting      [C]ancel"
@22,(80-len(outer))/2 say outer
x=0
do while x=0
  x=inkey()
  if upper (chr(x)) $ "PC"
    exit
  endif
  x=0
enddo
if upper (chr(x)) $ "C"
  clear
  close all
  return
endif
clea
  set device to print
  fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)"
  souter="LIST OF REGISTERED BANK FOR CASH TRANSFER"
  @01,(80-LEN(fouter))/2 say fouter
  @02,(80-LEN(souter))/2 say souter
  @04,01 say "S/NO"
  @04,07 say "REG. DATE"
  @04,20 say "REG. TIME"
  @04,33 say "BANK CODE"
  @04,50 say "BANK NAME"
  @04,70 say "SHORTNAME"
  @05,01 say repl ("=",77)
  xpage=1
  row=06
  mrow=62
  xsno=1
  go top

  xregdate=ctod(" / / ")
  xregtime=space(08)
  xbankcode=space(02)
  xbankname=space(30)
  xshortname=space(03)
do while .not. eof()
  if row>mrow
    eject
    clear
    xpage=xpage+1
    row=08
    set device to print
    fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)"
    souter="LIST OF REGISTERED BANK FOR CASH TRANSFER"
    @01,(80-LEN(fouter))/2 say fouter

```

```

    @02,(80-LEN(souter))/2 say souter
    @04,01 say "S/NO"
    @04,07 say "REG. DATE"
    @04,20 say "REG. TIME"
    @04,33 say "BANK CODE"
    @04,50 say "BANK NAME"
    @04,70 say "SHORTNAME"
    @05,01 say repl ("=",77)
endif
    xregdate=regdate
    xregtime=regtime
    xbankcode=bankcode
    xbankname=bankname
    xshortname=shortname
    @row,01 say xsno      pict "999"
    @row,07 say xregdate  pict "99/99/99"
    @row,20 say xregtime  pict "99:99:99"
    @row,35 say xbankcode pict "@!"
    @row,42 say xbankname pict "@!"
    @row,72 say xshortname pict "@!"
    row=row+2
    xsno=xsno+1
skip
enddo
use
eject
jobend="ACGS HAS COMPLETED PRINTING JOB, Press Any Key To Continue..."
@22,02 clea to 23,77
@22,05 say "MESSAGE" color "w+*/r"
@23,(80-len(jobend))/2 say jobend
set cons off
wait " "
set cons on
clea
return
endif
*-----PROC REP2-----
proc REP2
    @21,08 clea to 23,71
    @21,08 to 23,71
    outer="[S]creen      [P]rinter"
    @22,(80-len(outer))/2 say outer
    x=0
    do while x=0
        x=inkey()
        if upper (chr(x)) $ "SP"
            exit
        endif
        x=0
    enddo
enddo

```


if upper (chr(x)) \$ "S" && ----- Generating Screen Printng-----

use request
index on requestno to requestno
go top

@21,08 clea to 23,71
@21,08 to 23,71
outer="[P]rinting [C]ancel"
@22,(80-len(outer))/2 say outer
x=0

do while x=0
x=inkey()
if upper (chr(x)) \$ "PC"
exit
endif
x=0
enddo

if upper (chr(x)) \$ "C"
clear
close all
return
endif
clea

set device to screen
fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)"
souter="REPORT ON LOAN REPAYMENT AND THEIR STATUS"

@01,(80-LEN(fouter))/2 say fouter
@02,(80-LEN(souter))/2 say souter
@04,01 say "S/NO"
@04,07 say "REG. DATE"
@04,20 say "REG. TIME"
@04,33 say "R/No."
@04,40 say "TOTAL APPROVED"
@04,57 say "TOTAL PAID"
@04,69 say "BALANCE"
@05,01 say repl ("=",77)

xpage=1
row=06
mrow=62
xsno=1
go top

do while .not. eof()
if row>mrow
*eject
clear
xpage=xpage+1
row=06
set device to SCREEN

fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)

```

souter="LIST OF ALL AVAILABLE LOAN TYPE"
@01,(80-LEN(fouter))/2 say fouter
@02,(80-LEN(souter))/2 say souter
@04,01 say "S/NO"
@04,07 say "REG. DATE"
@04,20 say "REG. TIME"
@04,33 say "R/No."
@04,40 say "TOTAL APPROVED"
@04,57 say "TOTAL PAID"
@04,69 say "TOTAL BALANCE"
@05,01 say repl ("=",77)
endif
xregdate=regdate
xregtime=regtime
xrequestno=requestno
xamt_given=amt_given
xpay_todate=pay_todate
xbalance=balance
@row,01 say xsno      pict "999"
@row,07 say xregdate  pict "99/99/99"
@row,20 say xregtime  pict "99:99:99"
@row,31 say xrequestno pict "9999"
@row,35 say xamt_given pict "99,999,999.99"
@row,55 say xpay_todate pict "9,999,999.99"
@row,67 say xbalance  pict "9,999,999.99"
row=row+2
xsno=xsno+1
skip
enddo
use
*eject
jobend="ACGS HAS COMPLETED PRINTING JOB, Press Any Key To Continue..."
@22,02 clea to 23,77
@22,05 say "MESSAGE" color "w+*/r"
@23,(80-len(jobend))/2 say jobend
set cons off
wait " "
set cons on
clea
return
endif

```

-----PROC REP3-----

```

proc REP3
@21,08 clea to 23,71
@21,08 to 23,71
outer="[S]creen      [P]rinter"
@22,(80-len(outer))/2 say outer
x=0
do while x=0

```

```

x=inkey()
if upper (chr(x)) $ "SPC"
  exit
endif
x=0
enddo
if upper (chr(x)) $ "S" && ----- Generating Screen Printng-----

use category
index on cat_code to cat_code
go top

@21,08 clea to 23,71
@21,08 to 23,71
outer="[P]rinting      [C]ancel"
@22,(80-len(outer))/2 say outer
x=0
do while x=0
  x=inkey()
  if upper (chr(x)) $ "PC"
    exit
  endif
  x=0
enddo
if upper (chr(x)) $ "C"
  clear
  close all
  return
endif
clea
  set device to screen
  fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)"
  souter="LIST OF ALL AVAILABLE PROJECT CATEGORIES"
  lives="10 =>Live Stock, 20 =>Fisheries, 30 =>Mixed Farming, 40 =>Food Crops,"
  live=" 50 =>Cash Crops, 60 => Others"
  @01,(80-LEN(fouter))/2 say fouter
  @02,(80-LEN(souter))/2 say souter
  @21,(80-LEN(lives))/2 say lives
  @22,(80-LEN(live))/2 say live
  @04,01 say "S/NO"
  @04,07 say "REG. DATE"
  @04,20 say "REG. TIME"
  @04,33 say "CATEGORY CODE"
  @04,50 say "PROJECT CATEGORY"
  @05,01 say repl ("=",77)
  xpage=1
  row=06
  mrow=20
  xsno=1
  go top

```

```

xregdate=ctod(" / / ")
xregtime=space(08)
xcat_code=space(02)
xcat_desc=space(15)
do while .not. eof()
  if row>mrow
    jobend="Press Any Key To Continue.....!!!"
    @22,02 clea to 23,77
    @22,05 say "MESSAGE" color "w+*/r"
    live=" 50 =>Cash Crops, 60 => Others"
    @22,(80-LEN(live))/2 say live
    @23,(80-len(jobend))/2 say jobend
    set cons off
    wait " "
    set cons on
    clea
    xpage=xpage+1
    row=06
    set device to SCREEN
    fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACG
    souter="LIST OF ALL AVAILABLE PROJECT CATEGORIES"
    lives="10=>Live Stock, 20=>Fisheries, 30=>Mixed Farming, 40=>Food Crops, 50=> Cas
Crops, (60) => Others"
    live=" 50 =>Cash Crops, 60 => Others"
    @01,(80-LEN(fouter))/2 say fouter
    @02,(80-LEN(souter))/2 say souter
    @21,(80-len(lives))/2 say lives
    @22,(80-LEN(live))/2 say live
    @04,01 say "S/NO"
    @04,07 say "REG. DATE"
    @04,20 say "REG. TIME"
    @04,33 say "CATEGORY CODE"
    @04,50 say "PROJECT CATEGORY"
    @05,01 say repl ("=",77)
  endif
  xregdate=regdate
  xregtime=regtime
  xcat_code=cat_code
  xcat_desc=cat_desc
  @row,01 say xsno pict "999"
  @row,07 say xregdate pict "99/99/99"
  @row,20 say xregtime pict "99:99:99"
  @row,38 say xcat_code pict "@!"
  @row,52 say xcat_desc pict "@!"
  row=row+2
  xsno=xsno+1
skip
enddo
use
jobend="ACGS IIAS COMPLETED PRINTING JOB, Press Any Key To Continue..."

```

```

@22,02 clea to 23,77
@22,05 say "MESSAGE" color "w+*/r"
@23,(80-len(jobend))/2 say jobend
set cons off
wait " "
set cons on
clea
return
endif
if upper (chr(x)) $ "P" && ----- Generating Screen Printng-----

use category
index on cat_code to cat_code
go top

@21,08 clea to 23,71
@21,08 to 23,71
outer="[P]rinting      [C]ancel"
@22,(80-len(outer))/2 say outer
x=0
do while x=0
  x=inkey()
  if upper (chr(x)) $ "PC"
    exit
  endif
  x=0
enddo
if upper (chr(x)) $ "C"
  clear
  close all
  return
endif
clea
  set device to print
  fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)"
  souter="LIST OF ALL AVAILABLE PROJECT CATEGORIES"
  lives="10=>Live Stock, 20=>Fisheries, 30=>Mixed Farming, 40=>Food Crops, 50=> Cash
Crops, (60) => Others"
  live=" 50 =>Cash Crops, 60 => Others"
  @01,(80-LEN(fouter))/2 say fouter
  @02,(80-LEN(souter))/2 say souter
  @21,(80-len(lives))/2 say lives
  @22,(80-LEN(live))/2 say live
  @04,01 say "S/NO"
  @04,07 say "REG. DATE"
  @04,20 say "REG. TIME"
  @04,33 say "CATEGORY CODE"
  @04,50 say "PROJECT CATEGORY"
  @05,01 say repl ("=",77)
  xpage=1

```

```
row=06
mrow=62
xsno=1
go top
```

```
xregdate=ctod(" / / ")
xregtime=space(08)
xcat_code=space(02)
xcat_desc=space(15)
do while .not. eof()
  if row>mrow
    eject
    clear
    xpage=xpage+1
    row=08
    set device to print
    fouter="THE IMPACT OF AGRICULTURAL CREDIT GUARANTEE SCHEME (ACGS)
    souter="LIST OF ALL AVAILABLE PROJECT CATEGORIES"
    lives="10=>Live Stock, 20=>Fisheries, 30=>Mixed Farming, 40=>Food Crops, 50=> Cash
Crops, (60) => Others"
    live=" 50 =>Cash Crops, 60 => Others"
    @01,(80-LEN(fouter))/2 say fouter
    @02,(80-LEN(souter))/2 say souter
    @21,(80-len(lives))/2 say lives
    @22,(80-LEN(live))/2 say live
    @04,01 say "S/NO"
    @04,07 say "REG. DATE"
    @04,20 say "REG. TIME"
    @04,33 say "CATEGORY CODE"
    @04,50 say "PROJECT CATEGORY"
    @05,01 say repl ("=",77)
  endif
  xregdate=regdate
  xregtime=regtime
  xcat_code=cat_code
  xcat_desc=cat_desc
  @row,01 say xsno      pict "999"
  @row,07 say xregdate  pict "99/99/99"
  @row,20 say xregtime  pict "99:99:99"
  @row,38 say xcat_code pict "@!"
  @row,52 say xcat_desc pict "@!"
  row=row+2
  xsno=xsno+1
skip
enddo
use
eject
jobend="ACGS HAS COMPLETED PRINTING JOB, Press Any Key To Continue..."
@22,02 clea to 23,77
@22,05 say "MESSAGE" color "w+*/r"
```

```
@23,(80-len(jobend))/2 say jobend
set cons off
wait " "
set cons on
clea
return
```

```
endif
```

```
*----- End Of Program-----
```

```
-
```