COMPUTERIZATION OF WAREHOUSE STOCK TAKING AND ANALYSIS PROCEDURE
CASE STUDY OF TONY STORE ENTERPRISES

BY

AZUBUIKE, JOHN MARY MBAAMALU

PGD/MCS/048/96

A PROJECT SUBMITTED TO THE DEPARTMENT OF MATHS/STATISTICS/COMPUTER SCIENCE, FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGER STATE, AS A PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF POSTGRADUATE DIPLOMA IN COMPUTER SCIENCE.

1
pces

MARCH, 1998

CERTIFICATION

We testify that this work - The Computerization of Warehouse stocktaking and analysis procedure - was done by John M. Azubuike a student of Department of Maths/Statistics/Computer Science, Federal University of Technology, Minna, Niger State.

DR. Y. M. AIYESIMI
PROJECT SUPERVISOR

DATE

PROF. K. R. ADEBOYE
HEAD OF DEPARTMENT

DATE

EXTERNAL EXAMINER

DATE

DEDICATION

I dedicate this work to the followings:

GOD, ALMIGHTY through the BLESSED VIRGIN MARY

Anthony Oguejiofor

Peter Onyibor

Rev. Fr. Kizito Raphael

Dr. & Mrs. Peter Ozemoya

Luke Duru

ACKNOWLEDGEMENT

No one will achieve any meaningful, scientific and research work without meeting people on the way going and coming down. Hence, the writer is therefore, grateful to the all those who in one way or other assisted him to making this work a success. First and foremost to God, who owns life and knowledge.

I do not know how to express my gratitude for the wonderful contribution and personal life experience which I acquired from my supervisor Dr. Y. M. Aiyesimi, who took his time to go through this write-up. I also remain grateful to my HOD, Prof. K.R.Adeboye and all my lecturers who in one way or the other tolerated my short comings. Another person is Engr. Yakubu Ibrahim who is like father and Senior brother. He is my own stock control.

I am equally grateful to the following for their supports and encouragements. Mr. Lawrence Anekwe for his kindness, Prince Badmos who really co-ordinated me during the one year course. All the lecturers who taught me what computer is all about.

To my coursemates, I must sincerely say that I am very grateful and happy to have associated with them, and from them I got a very high inspiration to write. Naming then will make me fall into the error of omission. Bravo to your kind gestures.

My heart is full of thanks to all my friends who have been assisting me both financially and otherwise, people like Mr. Paulinus Nwaora, Emeka Olinya, Mohammed Danlami Bosso and others whose names may not contain here.

May we remain blessed in the lord. Amen.

ABSTRACT

Most companies, manufacturers, wholesalers, or retailers, keep some sort of warehouse stock inventory and transactional analysis which may be carried out at every six(6) months, or annually as the case may be, on goods bought for resale.

Computerization of warehouse stocktaking and analysis procedure is a well-defined, organized, systematic way of taking stock of goods in the warehouse, but also finding the correct means of analyzing past transactions in order to determine maximum profit by minimizing loss.

The project is aimed at assisting petty traders to re-structure and re-organize their businesses so as to gain strategic advantage over their competitors.

TABLE OF CONTENTS

CHAPTER ONE

CHAPTER TWO

CHAPTER THREE

CHAPTER FOUR

CHAPTER FIVE

CHAPTER ONE

1.0 THE CONCEPT OF COMPUTERIZING WAREHOUSE STOCK INVENTORY AND ANALYSIS PROCEDURES

1.1 GENERAL INTRODUCTION

No company whatsoever, or manufacturing houses or wholesalers, or retailers wants to record at the end of every month or annually any deficit (loss) after each of their stocktaking and analysis events. The main purpose of either establishments is not loss, but onward movement in progress. Hence, the contents of their dictionaries are progress profit and prosperity. So, proper warehouse stock inventory and analysis, and coupled with managerial account-ability assures one of the above P's.

Companies, manufacturing houses, different kinds businesses; wholesalers and retailers take stock and regularly analyze their daily transactions, some at every six(6) months and others annually. The feedback from the above operation (stocktaking and analysis) are forms of business control in the sense that they help the purchasing houses to assess their current performances and to make continuous adjustments in the activities in the activities of the business to ensure that the establishment achieves its objectives.

Warehouse stock inventory and analysis procedure simply means well-defined, organized, systematic way of not only taking stock of goods in the warehouse, but also the correct means of analysing past transactions. The above have proved, in no small measure, to be powerful instruments for decision making both in a wide or narrow variety of managerial context for management information system. There is no business that does not make use of this management system: It appears on the balance sheets, visible on the,manufacturing statements, and behold on the profit and loss statement.

Generally, warehouse stock inventory and analysis have a two-way effect on the financial message and movement of the business, a firm, an organization, etc. Goods to be counted are assets and therefore represents stored values that when sold will generate revenues that when sold will generate revenues and hopefully profits. On the other hand, inventories are usually major investments and are financed byu equity or debt. Therefore inventory level (ROL-Re-order level) directly affects the return on investment.

An inventory and analysis records are set-up the first time an item is ordered or purchased either within or outside the business. These records are used once or twice a year to physically count and analyze all items stocked in the warehouse within a specific period. Most often this clerical inventory - analysis procedures do not match the on hand balance on the records in use since too many may have been purchased the course of transaction, receipts may have been counted in correctly, and pilferage may have occurred along the line.

Furthermore, these records help to provides good customer services (customer service here refer to the percentage of customer requests or orders which are fulfilled without delay).. Since these require having goods enough in the warehouse in order to satisfy the customer, the inventory and analysis can be determined by the followings:

- Minimum balance
- Re-order level and re-order quantity.

These two conditions are concerned with finding the equilibrium between the maximum quantity and the quality which cannot go beyond the re-order level. In a way, it assures us of a correct balance between the cost of stockholding and the benefits from stockholding.

It is no uncommon that computer integrated information processing is of vital importance to this kind of business environment. In most establishment like the Tony Store Enterprises, the ultimate aim in integrating these technologies is to improve effectiveness and efficiency in performing repetitive tasks. If these technologies, especially the services of Arago is applied to stocktaking analysis it will make this old cleric procedure less tedious, accurate of accounting and recording, and fast in execution.

The in-built facilities in Arago is what is recommended to the developing of the new software:

Those facilities include:

- It's relational approach to data.
- Its password facilities.
- The saving of its object codes for faster execution.
- Its pop-up menus and windows.
- Its large number of memory variables.
- Large range of command, etc.

1.1.1 THE SCOPE AND TARGET OF THE PROJECT

Very many people have done so well in writing their project on big business and governmental establishments, this include: Supermarkets, chemist stores, institutions, Banks, various ministries. This project concentrates more on the peasant traders with low income and with little or no assets.

The intention of undertaking this task, is to assist businessmen and women to find some motivating principles that will enable them invest on a solid background towards proper effectiveness, efficiency and enhancement of their business operations. It will also enable them to keep to the ethics of business world and grow steadily to the interpretational standards of commerce and economics.

1.1.2 THE PROJECT ORGANIZATION

The project is organized into five chapters, and each of these chapters is further subdivided as the case may be (where necessary). Every other chapter except chapter one has an introduction or a brief explanation of what the entire chapter is all about. This gives us a birds-eye-view of the chapter in question.

Chapter one simply introduces and explains the topic and demonstrates the target. Chapter two gives us a initiative analysis of the current system by bringing to light its strengths and weaknesses, and suggests an alternative to problematic areas. Chapter three explicitly demonstrates the actual stocktaking techniques. Chapter four explicitly shows the new system specification: Its input, output, processing, file structure and file organization. Within this chapter, we also delve into the different ways of system code with an aim to choosing a code suitable for the new program. Finally, chapter five pictures the system conversion, implementation and program listing for the benefits of the users and with recommendation and conclusion, the project comes to a close.

1.2 DEFINITION OF THE TOPIC

Computer is that small but mighty 'monster' ever ready to change course, ever ready to abandon projects, ever ready to betray the users. It does not process any instruction without the assistance of reason, and without the aid of logic. It is man's long anticipated reality (dreams) designed to assist him in accomplishing his multifaceted missions on earth.

Computer is a device capable of manipulating data by accepting data as inputs, performing prescribed operations on the data, and supplying the results of these operations via an output (in a soft or hard copy). Computerization, therefore, is the act of computing simply put, it is the act of using computer to solve varied problems.

Warehousing is the dynamic aspect of commercial storage. It is the storage place of commodities (goods). It includes also material handling systems, and control and information systems whose function is to facilitate the efficient placing of stored materials for convenient or scheduled retrieval. Hence, large stock must be stored in order that essential commodities may be available at all times to avoid stock out.

Tony store enterprises can be described as a business established by Anthony Oguejiofor in 1991. He deals on motorcycle spare parts on wholesale basis, which means that he purchases these parts on economic order quantity (EOQ) and sells them in small quantities to retailers or consumers,and at times to manufacturers. Hence, he needs warehouse.

Stock inventory is also stocktaking. Stock is goods available for sales, distribution or use. Goods kept in the warehouse by a trader or so. In this case one can say in/out of stock. While inventory is the detailed listing of goods in the warehouse. Hence, stocktaking of stock inventory is the detailed analysis or examination of the business transactions within certain specified period.

Procedure means the scientific order of carrying out inventory. Thus, "Computerization of Warehouse stocktaking and analysis vis-avis Tony store Enterprises" can simply be defined as the scientific method of warehouse inventory analysis directed on Tony's store through the application of computer technologies in order to enhance the profit and minimize loss.

Actually, we have said a little about the sole proprietor, but it is worthwhile that we should know how the director began the business. Thus a brief history of Tony's store Enterprises will be a welcomed idea.

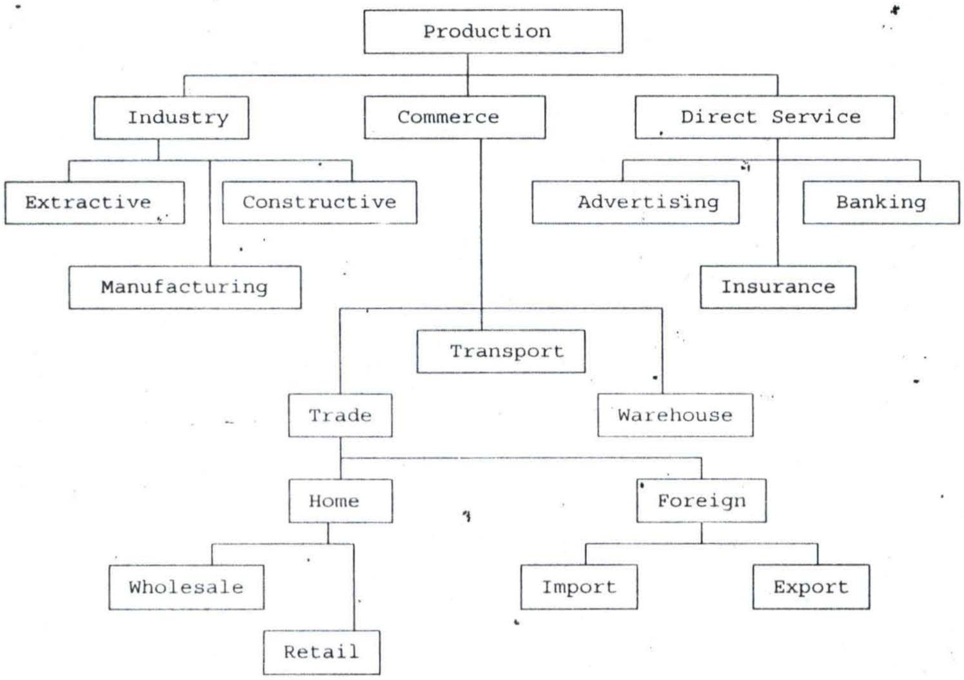


Fig. 1.1 The position of wholesale and warehouse in production chart

1.3 A BRIEF HISTORY OF TONY STORE ENTERPRISES

Tony store Enterprises is relatively a small sole proprietorship that retails motorcycle spare parts. It is managed and directed by Anthony S. Oguejiofor who started the business in 1991 and still operates it in its original location Fogun/Lalemi Street, Bida.

He runs the business of his own account, and therefore, is under no obligation to share his profits with any body, as is the case in the large-scale organizations. He is constantly connectes with the retail traders. This kind of trade has been identified to the earliest and the most simplest type of business organization. The set-up capital is small in amount and is obtained from his years of services under his master's guard. It was given to him by his master as a compensation for his loyalty and apprenticeship. Under such limitations, the scope of his activities was drastically reduced. Despite the fact that the initial capital was small, Tony managed it and tried to meet up with the fast growing competitors.

This kind of sole trade has great opportunities, if it can continue to transact those goods for which there is a regular demand. Hence, his decision to invest his limited resources on motorcycle spare parts. His stronghold is in the suburbs of the town and outlying districts, where his shop is conveniently situated to supply the daily needs of his customers. He has working personnel, especially apprentices who assist in the daily transactions of the business for the years as agreed with the manager and the person's parents, before they are freed to start off their own.

1.4 PREVIEW TO THE CURRENT SYSTEM

The daily operations in the Tony's business is manually done. The business uses clerical methods in sorting distributing goods in processing, inventing, analysing and recording information, and in preparing and distributing documents with exceptions where sales are done on cash and carry basis.

In this condition, therefore, the business need no equipment to process input data since most of the customer's buy in small quantity. It's processing capacity consists of adding up sales (which at times are done by the apprentices) and of determining which items to re-order or purchase (normally carried out in an initiative manner). The output preparation of the business consists of writing a few cheque (by hand) and ordering merchandise by phone or handwritten letter.

warehouse.

- Date.
- Item Name.
- Quantity stocked in.
- Quantity stocked out.
- Item code.

c. **EXPENSES BOOK**

Expenses book keeps information on money spent daily by the management and the expenses done during the course of ordering and carrying of goods.

- Date
- Description.
- Destination..
- Amount.

d. **INCOME BOOK**

Income book stores data on revenues of the business, that is, the realisations of the business. The income is achieved from sales.

- Date.
- Description of income.
- Bank.
- Amount.

e. **BANK BOOK**

This retains the information about money deposited or withdrawn from bank and which bank this transaction was carried out.

- Date.
- Bank Name.

It is now a principle to the business that once or twice a year, all items stocked in the warehouse are physically counted. Most often, this counting does not match the on hand balance on the inventory and the analysis records. Then, it becomes a problem and almost impossible to decipher the problem areas.

Despite the fact of the existing obstacles the management still adopts manual method in book-keeping without the intention of finding a better means of getting accurate result due to fear of misuses of fund. That notwithstanding, the need of using computer to generate an accurate report need not be overemphasised knowing fully well that record keeping involves how much cash is available for use, the number of debtors and creditors, and the amount spent previously. So, clerical methods make these (inventory and analysis procedure) very cumbersome and inefficient.

The management uses physical method to record information and update these information in files. The files used and kind of information kept in them are as follows:

a. **SALES BOOK**

Sales book stores information on daily sales indicating what item is sold and its unit price.

- Date.
- Number of item sold.
- Unit price.
- Total price.

b. **STOCK CARED (PURCHASE) BOOKS**

This is a miniature stock inventory analysis done as soon as there is addition of goods or deletion of goods from the

- Amount deposited/withdrawn.
- Name of the payee.
- Bank address.

f. **CREDIT BOOK**

The credit book keeps information on the business credits and the possible date to pay.

- Date.
- Suppliers name.
- Suppliers address.
- Item purchased.
- Unit price.
- Total amount.
- Amount paid.
- Balance.
- Date to pay.

g. **DEBIT BOOK**

This book keeps the data of some debtors and the debts to the management.

- Date
- Customer's name.
- Item sold/Name.
- Amount.
- Deposit.
- Balance.
- Date of payment.

When inventory and analysis are due, the only equipment used for the summaries is pen, sheets of paper, and a calculator. Due to the weakness of batteries and tiredness borne out of overwork, the manual method does not give a satisfactory result. This unsatisfactory answer may also be a mere mistake of wanting to press 9 one presses 8 unknowingly.

Again, retrieval and storage of information manually done causes a lot of untold misunderstanding between the management and the customers. In the case of storage, the office file is used and is prone to damage either by flood, ants, rats, fire, etc. Moreover, constant opening and using of files causes the file to wear out easily, and the contents are bound to loss. It is all these and many others that gave birth to this project.

1.5 THE ECONOMICS OF WAREHOUSING AND INVENTORY ANALYSIS

Commerce, of almost any kind, creates the need and ways to accumulate goods for subsequent distribution to the market place and inventory analysis to the management on the basis of fluctuating demand. Initially, the emphasis for these was much on security from the weather, pests, and theft, but now its importance is glaring.

The function of warehousing and inventory procedures can be applied to:

1. Products such as crops which are produced during a limited period but sold all year round.
2. Products such as seasonal or sports equipments produced year round, but consumed only during a limited period.
3. Products for which demand is unpredictable, such as pharmaceutical, and
4. Products that are manufactured in large quantity (production) runs for economy's sake and dispersed over a

long period.

If all products were manufactured as they were demanded (ordered), rather than held in large quantities for possible future sales, there would be little or no need for warehousing and stock analysis. Hence, once there is need for warehousing and stock analysis, there is unlimited economy for them.

Like commerce, transportation is intermittent. Once a car, lorry, ship or aeroplane is loaded, it takes hours, days, weeks, months to get at its destination, thereby causing delay in services. Again, if there is no stock analysis the management will not know the units to order for resale. Hence, there is need for accumulating the product and stock analysis in a systematic way. Warehousing, therefore, accommodates transportation surges, similarly, inventory (stocktaking) accommodates surges of stockout.

Warehousing and stock analysis, also permits economies through flexibility of transportation. A manufacturer with sufficient volume can use warehousing and inventory (stock analysis) to achieve the most economical combination, while at the same time satisfying other requirements.

Warehouse and inventory provide place-time value on production process. Raw materials, components, and subassemblies must be fed to production lines in a steady flow. To shut down production because of the unavailability of materials is extremely costly, and the warehouse and inventory gives insurance against this contigency.

Warehouse cannot be described as a distribution centre without inventory playing a vital role. Any manufacturing houses that takes stock and seldomly manufactures its complete production line, e.g multiplant, cars, can sell to its customers a selection that includes products form all its plants. It is the warehouse and inventory analysis that makes this feasible.

Another important of warehouse and stock analysis is accuracy in forecasting of market requirements. This is necessary so as to ensure that the right quantities are applied, assuring the distribution centre of adequate stocks on hand to fill customers orders.

It is always important to take stock and analyze it. At all times, as in any warehouse, the risks and costs of not having all products in stock, at all times, must be weighed against the risks and costs of having an over supply and paying the penalties of obsolescence and deterioration, as well as, the cost of the products themselves and the warehousing expenses of holding them.

1.5 OBJECTIVES AND PURPOSE OF THE PROJECT

On of the objectives of the project is to design a computer interactive and user friendly data base management system accessible to peasant (petty) traders and most especially for the management of Tony store Enterprises. Also, it is geared towards reducing the problems relating to the storage, processing and retrieval of information manually stored in office paper files and invoice books.

In the course of exposing the weakness of existing system, designing a proposed system and implementation, an application software which takes care of all the problems of warehouse stocktaking and analysis will be developed.

No doubt, there are various and vast reasons that will lure one into making an expose of warehouse stock inventory and analysis procedures. Specifically, what championed the course of this write-up is that many projects on stock inventory focused their aim not only governmental agencies, but also on large establishments.

Coming from an area that is invested with commerce and economics, where everybody's language is business, therefore, the project intends to educate traders that computer can be applied to any level of transactions for greater proficiency.

Few of our traders, especially the so-called "petty traders" keep proper records of what they buy and sell, or what they pay and receives, that is, their business transactions, consequently, they are unable to tell at any given time the value of goods and other properties in their possession, what they owe to others and what others owe them, and the amount of profit and loss they have made.

That recording the movement of goods and services in a manner that the financial position of a business can be ascertained readily at any time is complicated and cumbersome is not far fetch because the recordings are manually done. Therefore, the most important motivator to undertake this task, apart from the above point is

1. to show the result of business transactions, that is, whether the business has made any profit or loss within the period in question, and

2. to show also the financial position of a business at a certain date.

Apart from the above reasons, there are other reasons that are being experimented for the benefits of introducing computer to such business.. They are as follows:

a. To survey the means necessary to restructure the entire operation used in the business in order to gain strategic advantage over completion.

b. To facilitate some expansion of the business.

c. To improve on the system. of rendering services to customers.

d. To improve effectiveness of group decision-making.

e. To make appropriate information more readable, reliable, ready and available to decision-makers.

f. To find an alternative way of accomplishing a new tasks that are not feasible using only manual methods.

g. To improve efficiency and effectiveness in performing routine (repetitive) tasks.

h. To create changes in organization and management tasks.

i. To facilitate implementation of new business strategies and taking up new competition chances.

j. To create the relationship to producers and markets.

These objectives can only be achieved if proper records of business transactions are kept and accurate financial statements prepared. The manner(and files to be used) in which these records are kept, and financial statements prepared is fully discussed in this job (project).

CHAPTER TWO

2.0 ANALYSIS OF THE CURRENT SYSTEM

This chapter analytically brings out more clearly the operations of the current system as seen in Tony'S undertakings. With this analysis we will be able to propose an alternative system (which taken collectively can be called OFFICE AUTOMATION bringing change in an office by means of introducing computer technologies).

The current system analysis is a kind of detailed study of the existing system especially its procedures, information flows, and methods of operation and control. Next is the analysis costs and benefits of the propose system. This analaysis helps us to know whether the changover can be controlled: single or mixed. It further demonstrates the advantages of the new installations over the old one and reviews some books (if any) on the above project topic. Here we begin with the literature review.

2.1 LITERATURE REVIEW

The computer industry can be traced as fare back as 450 B.C with ABACUS as a stepping stone. The actual computer industry can be said to have received more attention in the late forties with a very small initial investment, and has been increasing both in strength and importance. When one looks back with analytical mind, he or she can conclude that computer technology keeps on advancing with remarkable increase in speed, accuracy and reliability.

Computing in whatever field: Science business, and industry is reaching directly or indirectly into various aspects of our society thereby, without over exaggeration has shrunk the world into such a compactness that no part can afford to lag behind or live in isolation.

The advent of electronic computer was hailed by the world as the pict of computer development, and also seen as a great revolution of 20th century. Like any industrial revolution, it promised to free man from simple routine jobs of repetitive nature by providing computing power. The first to have exploited this facility was the search for scientific enquiries after science was business.

It would be tedious and exhaustible to list all the possible applications of computer. There is no limit to the uses and applications of computer. Common examples of data processing are Payroll, Accounting, Inventory management, Banking, Air-lines seat reservation, etc.

A quick look at the complete history of computer science reveals that authors have not approached it from the perspective of warehouse stock inventory and analysis procedure. One can rightly say that there is no existing literature and if there is any it has not come to light.

So, it is therefore, pertinent that the inclusion of computer into warehouse stock inventory and analysis procedures would facilitate the spirit of business in the world. It is necessary that this should move in pariparsu with the advent and onward movement of computer technologies warns that the human activities in the near future are tied up with the question of computerization.

2.2 FEASIBILITY STUDY OF THE CURRENT SYSTEM

Feasibility study can be described as an investigation to determine whether a particular system is practicable and controllable. It is simply a miniature system analysis and design. The importance of feasibility study is so vital in the sense that it enables us to have a good grasp of the information flow whether the system uses clerical or computer method, and possible way of improvement.

To this effect, therefore, the business in discussion processes the operational records by clerical method. This physical method of operation has never stopped experiencing problems of various kind. In recording the money paid by the customers, there is a kind of fool-play by the apprentices. This kind of fraud not only makes the records not tally with the services rendered, but also faces the problem of improper accountability.

The manual method of filing information goes a long way to give the needed satisfaction. When one wants to retrieve the file, the person has to encounter a lot of problems and even more hardship when trying to update or edit inventory and analyze transactions. In cases where there is mistake or certain records are to be deleted not only that mistakes occur all the more, but most often rats eat up the manually stored information.

The file which is made up of paper is easily damaged by termites. Even trying to get the documents that were used to commence the business originally becomes a fruitless venture. Hence; history is completely eroded.

Furthermore, there is the problem of storage system. As long as the business grows, and as long as the year goes by information yet to-be-stored becomes larger and larger. Hence, retrieval of the previous documents becomes impossible and results to poor services to the customers.

The above are the major problems facing the business. Apparently, rattled by fraud practised by the apprentices, the director general decided without delay the application of computer into his business, despite the fact of its low income finance. So, in order to enhance and ensure accountability and responsibility, he believes that computer can

- Keep more accurate records of transactions,
- Make the daily operations less tedious,
- Allow more frequent updating of inventory and transactional analysis,
- Ensure accurate account,
- Make retrieval easy and simple,
- Allow better storage facilities.

The application of computer and the use of the developed software, most of the enumerated problems above cease to exist. All information stored in a file would be stored in a floppy disk either in 3.5" or 5.25". Hence, the computerization of Tony Store Enterprises will then increase the profit and services of the business. Here, computer technologies are called office automation.

2.3 **WHAT IS OFFICE AUTOMATION**

Office Automation (OA) is evolving change in an office system centered on employing computer technologies to process information used in the office by improving the content, format, and quality of work performed. These are the targets of the director of Tony Store Enterprises. Office automation refers, collectively to those computer-based applications associated with general office work.

It (office automation) is an ongoing process of deploying tools that will aid office workers at all levels of an organisation to work more productively, efficiently and currently. Thus, it encompasses all the personal productivity software tools (word processing, Desktop Publishing, Electronic spreadsheet, Database and Graphics), Fax systems, Electronic mail, Image processing, Voice processing and Office information systems.

In most organizations, office automation is a gradual, evolutionary process of change implementation in response to the growing needs of the business. In this respect, OA can be viewed as a "process", rather than a one-time-implementation. The evolution of increasingly sophisticated technology and society hasten this process of change.

Finally, OA creates positive changes in the design and description of jobs, and in the organizational chart of the business. When properly applied and implemented, it can provide tremendous benefits that can be of more meritorious than clerical method. Cost and benefits analysis briefly demonstrates the above advantages more clearly, before then let us see some shortcomings of the current system.

2.4 THE SHORTCOMINGS OF THE CURRENT OPERATIONAL SYSTEM

The current system, as already indicated, operates basically on clerical method, that is, physical manipulation of data and information. In Tony Store Enterprises, these problems are visible under the above method.

a. **CAPABILITY**

The clerical method lacks the ability to handle transactions, and process these transactions towards a comfortable requirements, and also meet up with the desired performance.

b. **INFORMATION ACCESSIBILITY**

During data collection, it is observable that the information needed to accomplish a job in a useful form is not available and when available it is not accessible.

c. **CONTROL**

There is no mechanism to detect and report instances when activities are not carried out. The existence of control helps in the handling unanticipated events. Information that is non-existent is detected and if possible updated.

e. **ECONOMY**

The present condition of things, which, as a result of clerical method, seem to be economically, but it is not. Hence, economical method can be adopted.

f. **FLEXIBILITY**

The current system does not meet up with the increase in the volume of work to be done. It is also expected to be processed quickly (fast).

g. **EXISTING OF SYSTEM**

With the computerization of the business it is expected that some facilities and members of staff of the present system should be incorporated in the running of the new procedures.

2.5 **METHODS OF DATA COLLECTION**

Amidst all means of gathering information, the ones used in this project are OBSERVATION and INTERVIEW.

a. **OBSERVATION**

It simply means being involved in an operation for some times to see personally what is exactly happening within the system that calls for change. That is to say "seeing-eye-to-the-job".

b. **INTERVIEW**

This is the most effective and efficient way of data collection such as objectives, constraints, allocation of duties, problems and failures in the existing system. Also, it seems to be the most economical and face-to-face interaction with the users and operators.

CHAPTER THREE

3.0 THE WAREHOUSE STOCKTAKING TECHNIQUES

Generally, it is costly not only to hold stock, but also costly in terms of storage space, equipment, personnel, insurance (which many businessmen and women do not appreciate), deterioration and obsolescence. On the other hand, it costs darely to run out of stock. That is to say that lost of production, lost of profits, placing of order late, lost of appropriate services are not within the mainframe of any business, and such disorganizes the business.

Thus, this section is concerned with finding the equilibrium between the above costs. In any way, it assures us of a correct balance between the cost of stockholding and the benefits accrued from stockholding.

Inventory (stocktaking) can only be achieved if we arrived at the following principal functions of stock control. These functions are that:

a. Stock control enables the product to be produced in an economic order quantity (EOQ), and that

b. Stock control acts as a buffer against an unpredictably high rate of use.

The principal objective of stocktaking is to reveal through a sequence of mathematical models, the significance of certainty and uncertainty of inventory in decision-making. Thus, our study will be made in two phases. First and foremost, we shall examine the situation where demand and lead times are certain, and secondly, where demand and lead times are uncertain. Hence, the second probability to be considered introduces the uncertainty aspect of inventory, and also serves as transition to the more important and complicated models.

Let us begin with a principal method of analysing stock and how re-order level can be determined.

3.1 THE PRINCIPAL METHODS OF WAREHOUSE STOCK ANALYSIS AND RE-ORDER LEVEL TECHNIQUES.

The stock record provides an inferred stock balance (book stock): The quantities that should be available if all receipts and issues have been accurately entered and all calculations correctly made. To complete the cycle of control, it is necessary to count the physical stock and compare it with the book stock. If there is a difference between the two, it should be reported immediately to the store manager and to the office so that the records in cash terms my be adjusted.

There are two principal methods of warehouse stock analysis which can be easily adopted by any peasant trader.

- Periodic Review method.
- Continuous Review method.

3.1.1 THE PERIODIC REVIEW METHOD

Under this system the balance of stock on hand is checked periodically, and if the stock is found to be below the re-order level an order is placed. In order words, under this system it would be necessary to ensure that the stock could not fall between reviews to a level less that the demand during the lead time.

. This kind stocktaking depends upon the size of the store which may take a few hours or several days.

If, for example, the lead time is 15 days, the daily demand 200 units. If we assume the stock is reviewed every 20 days, this means that it must be above.

15 x 200 = 3,000 + 20 x 200 = 7,000 units.

So, adding to our buffer stock of 500 we have a 7,500 re-order level (ROL).

3.1.2 THE CONTINUOUS REVIEW METHODS

The continuous review model, otherwise known as the perpetual inventory, is the inventory system whereby the stock is reviewed continuously. Also, it means that the stock on hand is checked after every issue, and order of size, Y, is placed every time the stock level reaches at certain order point, R.

The purpose of this method is to avoid the overtime and disorganization of work associated with periodic stocktaking. After the inventory, any differences with the book stocks are reported and the records are adjusted to agree with the physical stock found.

If, for example, the lead time is 15 days, the daily demand 200 units and a 500 units buffer stock is held, then the ROL is

15 x 200 + 500 = 3,500 units

The objective of the above system is to determine the optimum value of Y and R which will minimize the costs of holding and the ordering cost. Hence, let us see how the complex system of warehouse stock control tends to make the long road shorter.

3.2 THE COMPLEX METHOD OF WAREHOUSE STOCK CONTROL

The complex method of warehouse stock control deals with those mathematical models that cannot be handled by the above methods. One major advantage is that the complex methods enable us handle the more complicated techniques of stock control. Here, we will begin with the stock analysis when demand is certain.

3.2.1 STOCK CONTROL ANALYSIS UNDER CERTAINTY OF DEMAND

Inventory is popular in management science. It is because almost all types of business organization have inventory. One can take stock not only on goods on the shelf, it can be done on a variety of forms, such as finished products at different stages of manufacturing process, raw materials, resources, labour, or cash.

In addition to the main purpose of inventory, companies can take stock on large inventories of raw materials as a hedge against strikes. Whatever form inventory takes or whatever its purpose may be, it often represents a significant cost of a business or a firm.

In the stock control under certainty the purpose is to develop a model that can help the manager make the decision of minimizing the holding and the ordering costs by means of EOQ model. It will be developed with the assumption that demand is known with certainty, that demand is constant over time, and that orders are made and received instantaneously with no shortages developing.

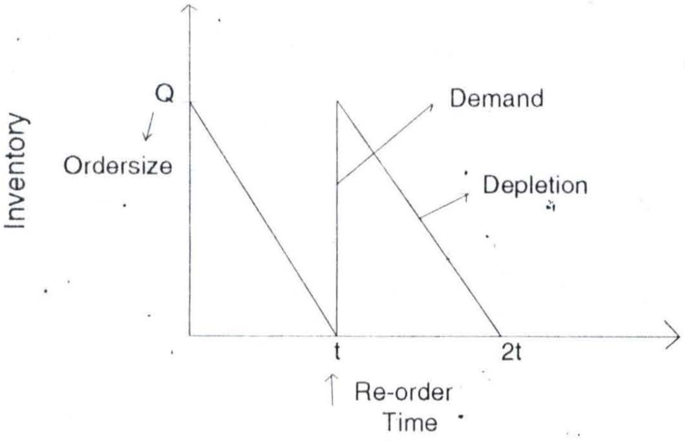
. This will be possible only if we can analyze the costs involved in it. They are two costs:

- Carrying cost (CC), and
- Ordering cost (OC).

Carrying costs are the costs incurred by the store for carrying the stock inventory. The total carrying costs (TCC) includes:

- Direct storage costs (rent, heat, light, maintenance,
- Security, record-keeping, labour, etc).
- Deferred profit on investment.
- Interest on the investment in inventory.
- Obsolescence.
- Depreciation, Taxes, Insurance.

CC is usually expressed on per unit basis for the same period of time, but mostly referred to as annual costs (peryear). For better understanding, let us illustrate this graphically.



- Fig. 3.1: Re-order level

In the above figure, Q represents the size of the order needed to replenish inventory. The line connecting Q to time, t, in our graph represents the rate at which inventory is depleted, demand, during the time period (lead time). Here demand is assumed to be known with certainty, and is this constant, which explains why the line representing demand is straight.

Also, inventory never goes below zero, a condition which the manager of Tony Store Enterprises originally specified. In addition, when the inventory level does reach zero, it is assumed that an order arrives immediately after an infinitely small passage of time, referred to as instantaneous receipt.

The amount of inventory is totally zero for an infinitely small period of time, since there is no inventory at the specific time, t. As such, the amount of inventory available is somewhere between these two extremes. Hence, a logical deduction is that the inventory available is the average inventory level, defined as:

Average Inventory = Q/2

In order to verify this relationship, specific any number of points - values of Q - over the entire time, t, will be divided by the number of points. For example.

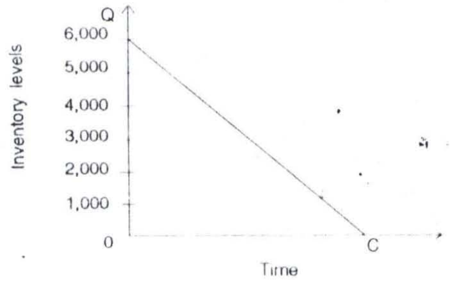


Fig. 3.2 : Average Inventory level

$$\text{Average Inventory} = \frac{5000+4000+3000+2000+1000}{6}$$

$$= 2,500$$

Now that we have the inventory available on an annual basis, is the average inventory, Q/2, we can determine the total annual carrying cost by multiplying the average number of item in the inventory by the CC per item per year, C.

The total Annual carrying cost = CQ/2.

The second inventory cost is the Ordering Cost (OC). This is the cost of placing an order incurred by the item. The total ordering cost generally includes:

- The cost of processing an order.
- Transportation costs to get the order from the supplier.
- The cost of unloading the order and placing it in inventory.
- All supplies used in ordering: forms, postages, telephone, computer time, etc.

The ordering cost is expressed per order basis, which is represented symbolically with 0. This represents the cost per order and not total ordering cost. Similarly, to total carrying cost, we shall also determine the annual ordering cost. This is derived from the number of orders that will be made during the year. Since the manage will not order more item than the amount demanded, and we know that demand is certain, then the number of orders per year is defined as:

Orders per year = D/Q

Where,

D = Demand per year.

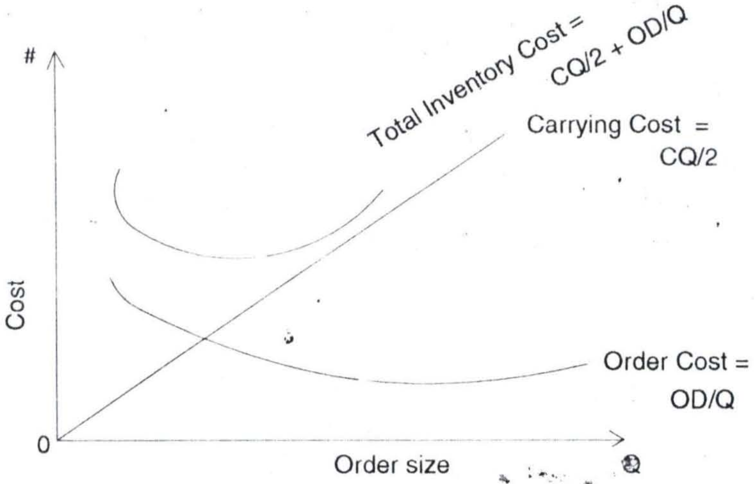
Thus, the total annual ordering cost (TAOC) can now be computed as:

TAOC = OD/Q

By this therefore, the total inventory cost (TIC) is computed by summing the total annual carrying cost (TACC) and the TAOC).

TIC = $CQ/2 + OD/Q$

This is explained graphically,



.Fig. 3.3: The Total Inventory Cost

Let us carefully analyze the three curves firstly, we sill observe the general upward trend of TACC curve. As the order size, Q increases, the TACC increase. This is logical since larger orders will result in more units carried in inventory.

Q	Total Annual Carrying Cost = CQ/2
1,000	375
2,000	750
3,000	1,125
4,000	1,500
5,000	1,875

Table 3.1 TACCC for selected order size.

Again, one could noticed from the same graph, the ordering cost curve. As the ordering size, Q, increases the ordering cost decreases. It is because an increase in the size of the order will result in fewer orders being placed each year.

Q	Total Annual Ordering Cost = OD/Q
1,000	1500
2,000	750
3,000	500
4,000	375
5,000	300

Table 3.2 TAOC for selected order size.

Finally, combining the two-costs will give us the curve, which is the total inventory cost.

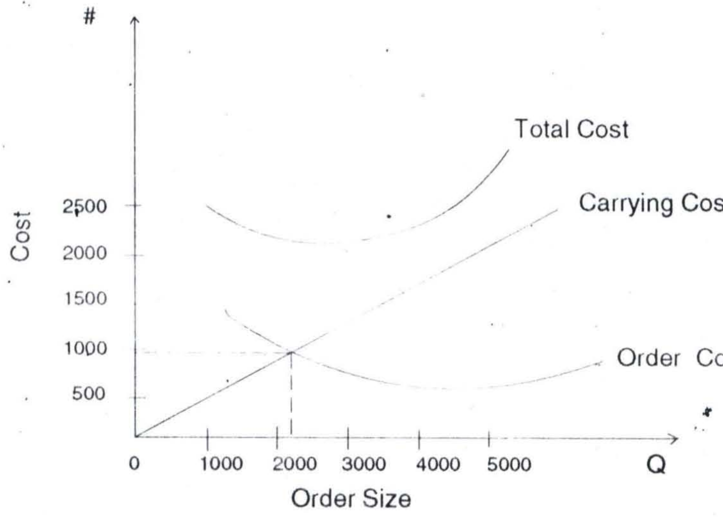


Fig. 3.4: The Total Annual Inventory Cost Curve

The main aim of this combination is to find the best or optimal value of Q, that is, the minimal total annual inventory cost. This could be directly read from the value of Q from the inventory graph by locating the lowest point on the total cost curve. But it is time consuming and cumbersome method, and it depends on completely accurate graph. Hence, the relationship is mathematically calculated thus:

$$TIC = CQ/2 + OD/Q$$

$$= CQ2 = OD/Q$$

Multiply both sides by Q

$$= CQ^2/2 = OD$$

Multiply both sides of the equation by 2

$$= CQ^2 = 20D$$

Divide both side by C

$$= Q^2 = \frac{20D}{C}$$

$$= Q^* = \frac{20D}{C}$$

Q^* signifies that this value Q is referred to as the EOQ.

Hence,

$$Q^* = \sqrt{\frac{2OD}{C}}$$

For instance, if a good or service is demanded 1000 units for a year, it costs N100.00 per unit per annum to store. Also, it costs N150.00 to make an order and a buffer stock of 20 is carried. What is the EOQ?

O = Cost for a single order = N150.00

D = Annual demand for product = 1000 units

C = Cost of holding 1 unit in stock per annum

= N100.00

$$EOQ = \frac{2x150x1000}{100}$$

$$= 3000 \text{ units}$$

3.2.1.1 THE ECONOMICS ORDER QUANTITY WITH A RE-ORDER LEVEL (POINTS)

In handling this topic, we will assume that an order is received an infinitely short time after it is placed. Better put, an order is placed and then, after a reasonable period of time it is received. The time required between the placement of an order and its receipt is referred to as the re-order lead time. The concept of lead-time is shown below graphically.

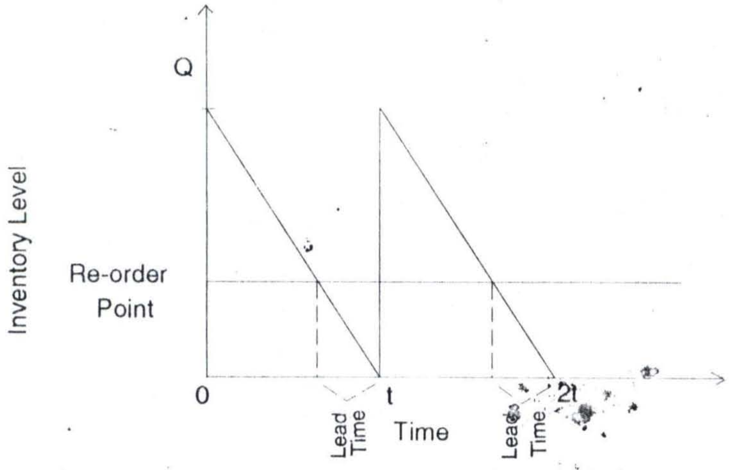


Fig. 3. 5: The EOQ with a Re-order point

The graph shows that the re-order must now be made prior to the level of inventory falling to zero since the demand for the item is consuming the inventory, while the order is being shipped, the order must be made while there is enough inventory in stock to meet demand during the lead-time period. This level of inventory is referred to as the re-order point (level).

The re-order level is computed mathematically, the lead time, L, by the demand per day. If we assume tat a year consists of.365 days, the demand per day id D/365. Thus, the formula for the re-order level, R, is:

$$R = L \frac{D}{365}$$

3.2.1.2 NONINSTANTANEOUS RECEIPT MODEL

Normally, an order is received all at once. In this situation, an order is received gradually over a period of time. In the above case, the situation is mostly applied to a firm that produces and sells as well, which means after the production the whole products go into inventory.

In fact this form of inventory analysis is called production lot size model because inventory is replenished directly from production. However, this model is not only applied to a production situation, but to any situation where the delivery of an order cannot be made at one point in time.

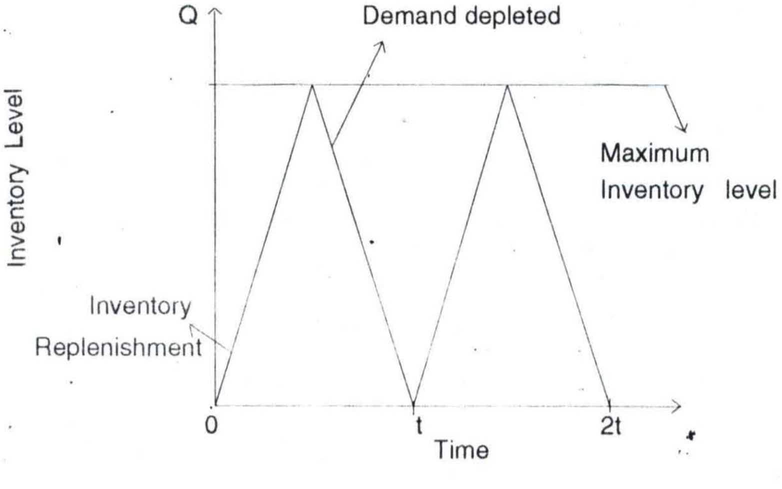


Fig. 3. 6: The Noninstantaneous Replenishment Model

In the graph above, the inventory is not replenished instantaneously, but instead rise gradually to a point (maximum point) where the entire order, Q, has been received. As inventory is being replenished from production, the demand for the product is depleting the inventory stock at the same time. That is why the replenishment is gradual.

However, replenishment could not occur at all if items were going out at a faster rate than they were coming in. It must be assumed that production exceeds demand in this model.

To develop a model to accommodate the gradual replenishment of inventory by production, a new model will consist of carrying cost (CC) and ordering cost. The OC is not affected by the gradual replenishment, since it is not depending on the average inventory. Let us have daily production rate for replenishment of inventory as, r.

We must determine the maximum inventory level in order to compute average inventory. First, we will compute the number of days required to receive the order.

Number of days to receive an order = Q/r.

Hence if Q = 10000 units of 250 x 17 tube. and 150 is the daily production rate (per day). Compute the number of days to receive an order which we express as, N

$$N = Q/R$$

$$= \frac{10000}{150}$$

= 66.7 days to receive an order

For the demand during this period, we must use the daily rate of demand, Do, which is found by dividing annual demand by 365 days.

$$Do = \frac{10000}{150}$$

$= 27.4$ 250 x 17 tubes demand per day.

While the number of tubes demanded during the order receipt period expressed as Do, is:

$$Do = Q/r \times Do$$

$$= 66.7 \times 27.4$$

$= 1827.6$ tubes demanded.

The maximum amount of inventory that could be on hand if nothing was demanded during the order receipt period is Q. However, Q is depleted by the amount of the demand. Thus, maximum inventory expressed as, M, is:

$$M = Q - Q/r = D.$$

While the average inventory level is one-half of this maximum inventory level expressed as Ma.

$$Ma = 1/2 \ (Q - QDo/r)$$

$$= Q/2 \ (1 - Do/r)$$

Hence, the total carrying cost is computed by multiplying the per unit carrying cost, C, by the average inventory.

$$TCC = CQ/2 \ (1-Do/r)$$

We equally use the annual CC even though our equation uses daily demand and daily replenishment, because by dividing the two they form a ratio that is independent of time. So, for us to do this, the total inventory cost model requires the addition of TCC and TOC.

TAIC (Total Annual Inventory Cost)

$$= OD/Q + CQ/2 \ (1-Do/r)$$

$$= CQ/2(1-Do/r) = OD/Q$$

Multiplying both sides of equations by Q

$= CQ^2/2\ (1\text{-}Do/r) = OD$

Multiplying both sides by 2

$= CQ^2/1\text{-}Do/r) = 20D$

Divide both sides of the equation by (1-Do/r)

$$= \quad CQ = \frac{20D}{(1\text{-}Do/r)}$$

$$= \quad Q2 = \frac{20D}{C(1\text{-}Do/r)}$$

$$Q^* = \sqrt{\frac{20D}{C}(1\text{-}D_O/R)}$$

3.2.1.3 ECONOMIC ORDER QUANTITY WITH SHORTAGES AND BACK ORDERING

It is a good managerial decision that an item that is more economical must be available at all times. Also, it is a good management and economical as well to allow shortages and back order demand and incur the cost associated with not being able to meet demand than keeping an excessive amount of inventory on hand to avoid shortages. This is graphically demonstrated thus.

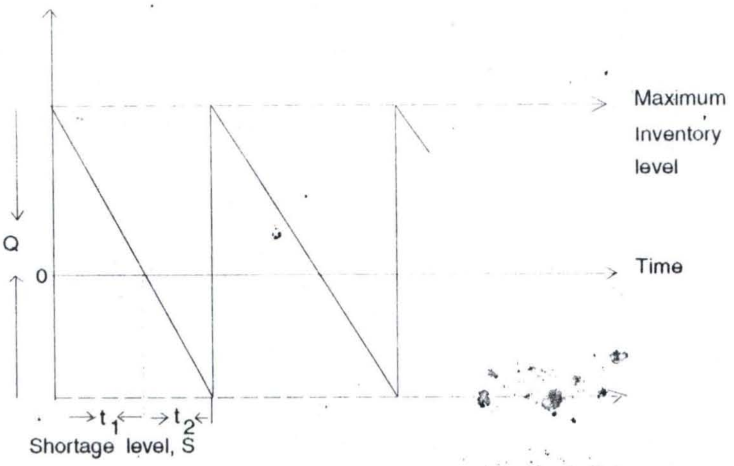


Fig. 3.7: The EOQ Model with Shortages and Backordering.

The graph is telling us that the time between order receipts has been divided into two other times; t1 and t_2. t_1 is the time which inventory is available and the time during which there is a shortage, t_2, at one order cycle.

During the shortages, the item will be unable to meet demand and will back order. Typically, a cost is assigned to shortages. Shortage costs are primarily related to lost present and future sales due to customers dissatisfaction, which we often referred to as a customer's goodwill cost. The longer a customer must wait, the more goodwill that will be lost. In addition, shortage costs can include late delivery costs for back orders and special labour cost for handling back orders.

In order to develop a model for this a shortage cost, the shortage cost must be added to our total cost equation. The development of both CC and shortage cost for this model is achieved using plane geometry. We shall not go into full detail of its derivation. Hence, let us define the shortage cost per unit as, Cs, and the maximum shortage level as, S. The total shortage cost (TSC) is

$$TSC = CsS^2/2Q$$

$$TCC = \frac{C(Q - S)^2}{2Q}$$

$$TOC = OD/Q$$

Total Annual inventory cost $= CsS^2/2Q + \frac{(CQ-S)^2}{2Q}+OD/Q$

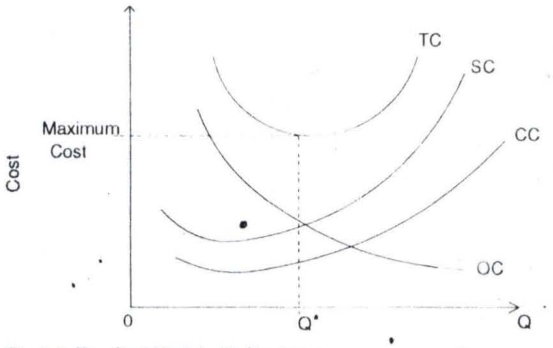


Fig. 3.8: The Cost Model with Shortages

The cost model with shortage graphs show that the lowest point on the total cost curve no longer occurs where all the component cost curves intersect.. Hence, we cannot just equate all the individual cost equations in order to determine the EOQ. The only alternative method is to use calculus.

$$Q^* = \frac{2OD}{C} \quad \frac{(Cs + C)}{Cs}$$

The minimum shortage level can also be computed as:

$$S = Q\left(\frac{C}{C+Cs}\right)$$

The time during which inventory exists, t_1, and the time during which a shortage exists, t_2, during which each order cycle can be computed using the formula:

$$t_1 = \frac{Q - S}{D}$$

and

$$t_2 = S/D$$

There is a situation where a larger demand can result in a discount from the supplier. It is called Quantity Discount model. Hence, the analysis requires that the CC be defined as a percentage of the price of the item in inventory paid to the supplier.

Let price be, P, then our total annual inventory cost model becomes

$$TAIC = OD/Q + CPQ/2 + PD$$

$$*PD = \text{Price x Demand}$$

$$Q^* = \sqrt{\frac{2OD}{CP}}$$

In order to receive the discount, the condition is that the order must be at least 4000, the customer gets 10% discount. If others are also given we can compute the total cost.

3.2.2 STOCK CONTROL ANALYSIS UNDER UNCERTAINTY OF DEMAND

This inventory analysis handles the situation where demand is not certain. It does not mean that the assumption that demand was always constant and known with certainty has been dropped. The difference is the certain future demand can rarely be predicted with certainty.

One of the major problems in performing this analysis is that it is difficult to determine a re-order level(ROL). For instance, a ROL is needed to stop a stockout when a period of time elapses between the placement of an order and its receipt.

The ROL is a level of inventory remaining in stock that is equal to the demand occurring during the time required to receive and order. But if demand is certain this inventory will be depleted at a known rate so that the order arrives at the sametime that the inventory level reaches zero e.g

R = LD/365 (This is still used)

However, although lead time may be constant, if demand is uncertain, then, it is not possible to exactly predict the demand that will occur during the lead time (say 10 days). Therefore, it is not possible that even though we have a ROL, shortages might occur anyway. As a solution to shortages when demand is uncertain, a buffer of extra inventory called a safety stock is often used.

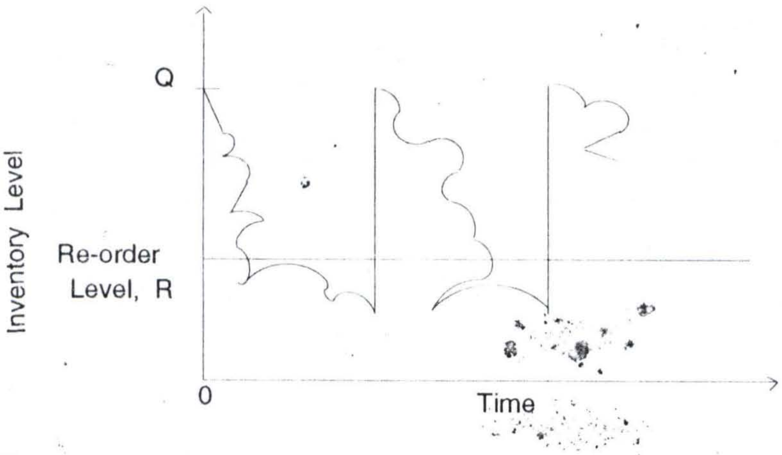


Fig. 3.9 : The ROL under Uncertainty

. Here a stockout occurs because demand exceeds the expected ROL during lead time. To solve this we add a safety stock to our graph.

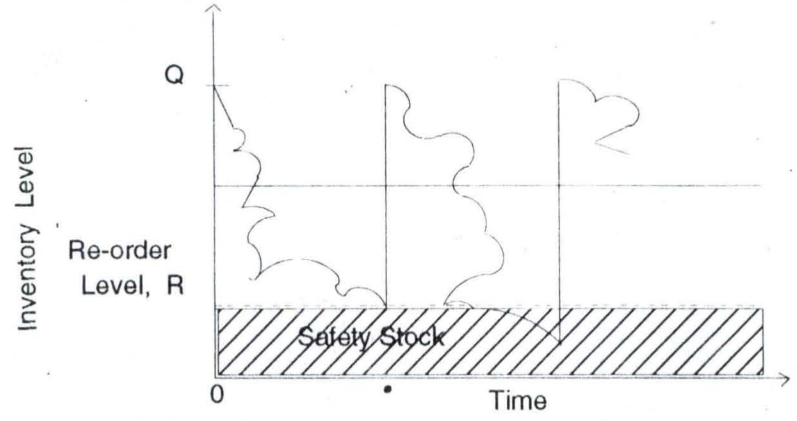


Fig. 3.1.0 : The ROL with Safety Stock

Maintaining a safety stock is not cost free. A CC is charged on it as it is on regular inventory. It is assumed that the frequency and amounts that actual demand is above the SS (safety stock). Hence to determine the annual cost of SS is by multiplying the annual CC per unit by the SS level, Ss

TACC = CSs

Though demand is not known with certainty we assume that it can be described by a probability distribution. Based on the Tony's Store, the manager has determined the following probability distribution for demand of 250 x 17 tube during the lead time period.

Demand during lead time	Probability
500	.10
600	.20
700	.40
800	.20
900	.10

Table 3.3: Determining the ROL for SS

Using this distribution, we will compute the expected demand during lead time, as follows:

$$\begin{aligned} E(\text{demand}) &= .10(5.00)+.20(600)+.40(700)+.20(800)+.10(900) \\ &= 700 \text{ tubes.} \end{aligned}$$

We will call the 700 tubes our initial ROL. The amount by which our finally determined ROL exceeds 700 tubes will be the safety stock. Hence, by using a ROL of 700 and no safety stock, it is possible for the actual demand during lead time to exceed this amount by 100 tubes 20% of the time, and by 200 tube 10% of the time.

If ROL of 800 including a safety stock of 100 is used, we could expect a stockout of 100 tubes only 10% of the time. If the ROL is 900 including a safety stock of 200, there would never be a stockout. Thus, the expected shortages for each of these possible ROL with safety stock are as follows:

ROL	SS	Actual Demand during lead time	Resulting shortages	Probability of shortages	Expected shortages Per D	Total Expected Per ROL
700	0	700	0	.40	0	
		800	100	.20	20	40
		900	200	.10	20	
800	100	800	0	.20	0	
		900	100	.10	10	10
900	200	900	0	.10	0	0

Table 3:4 ROL with safety stock.

Since the ROL tested are for a single order, the expected shortages are per order. Therefore, we need to determine annual shortages. This is accomplished by multiplying the average shortages per order by the member of orders per year. Recalling the number of orders per year to be = D/Q.

Since demand is uncertain, we will use an average demand of any size to compute both the order size, Q, and the number of orders per year. And from the EOQ equation we can compute the average annual shortages and cost of these shortages for each of the ROL's

So,

$$TASC = CsD/Q$$

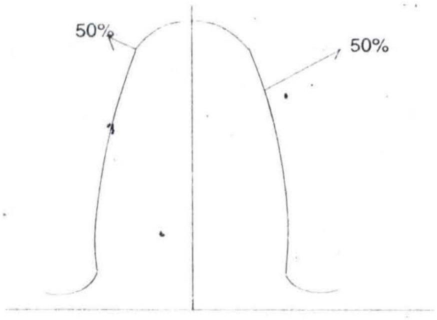
While,

TACC of safety stock = CSc

3.2.2.1 DETERMINING SAFETY STOCK USING SERVICE LEVELS

This method is mainly geared towards satisfying the demand of a specified percentage of total customers during the lead time period. The percentage of customer the store is able to service during the lead time is referred to as service level. This is a useful method for proper determination of safety stock when a shortage cost cannot be determined.

Remember that it is by adding a safety stock to the ROL; the store would be able to meet the demand of a larger percentage of the customers. Hence, the store is able to increase the service level by increasing the safety stock. Now, let us see the normal distribution of demand during the lead time in a graph for better and more elaboration.



μ = 300 tubes
α = 50 tubes

Fig. 3.1.1 : Normal Distribution of Demand During Lead Time

In so far as the average during lead time is 300, this becomes the ROP (ROL). Thus, the actual demand during lead time period will be more than 300, 50% of the time and less that 300, the service level of 50% of the time. Therefore, with ROP of 300, the service level of 50% during the ROL. To increase the service level, we must add a safety stock. That is to say, we establish a service level of 95%. This can be seen graphically.

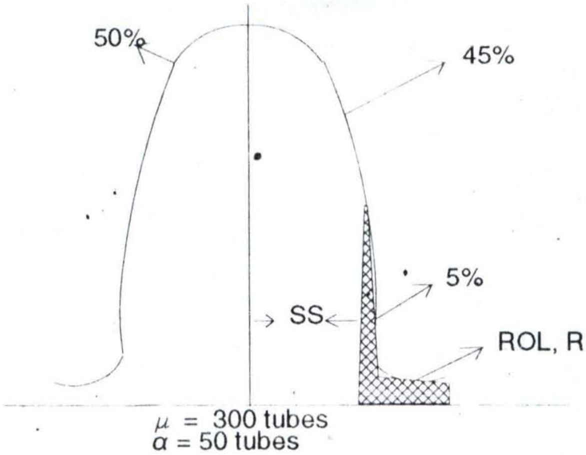


Fig. 3.1.2 : Increase in Service Level by Increasing the Safety Stock, Ss

To calculate the safety stock we have

Safety stock = $Z\alpha$

Where,

z = Number of standard deviations the ROL is from the mean

α = The standard deviation

For Re-order level, R

R = Ss x μ

3.2.3 SIMULATION OF INVENTORY

We have adopted this as a final analysis of inventory in a uncertain situation of demand. Simulation is a technique which involves the construction of mathematical models, that is, sets of equations, so as to represent in symbolic form the working of real-life situations thus, a retailing organization may wish to test the probable effects of an increase in rates of V.A.T on its profits. A computer would be used to process data based on an increased rate of say, 1,2,3, or 4 percent, elasticity of demand for products, etc. The computer's output, in the form of a statement as to probable effects, could be used by the organisation in determining pricing policies.

To demonstrate or relate•how simulation is used or applied to inventory we have to note that the selling costs, the inventory costs and the cost of the item remain the same. Also, the probability distribution for demand, except for this random numbers ranges

Demand(Dozen/ week	Probability	Cumulative Probability	Random Numbers
9	.15	.15	1-55
10	.40	..55	16-55
11	.25	.80	56-80
12	.20	1.00	81-99,00

Table 3:4 Random Numbers Ranges for probability Distribution .Demand.

Let us simulate demand for a single order size Q = 10 by selecting random numbers from the above table.

Random Demand	Demand	Sales DP/QP	Order Cost (QC)	C Cost (Q/2) C	SC Cs (D-Q)	Profit
39	10	N30	N15	N2.50	N0	N12.50
73	11	30	N15	2.50	1	11.50
72	11	30	N15	2.50	1	11.50
75	11	30	15	2.50	1	11.50
37	10	30	15	2.50	0	12.50
2	9	27	15	2.50	0	9.50
87	12	30	15	2.50	2	10.50
98	12	30	15	2.50	2	10.50
10	9	27	15	2.50	0	9.50
47	10	30	15	2.50	0	12.50
						N112.00

Table 3:6 Simulating a single order size Q = 10

The profit for each demand value was computed by sing the same profit formulas developed already. For Q = 10 simulation, the total profit is N112.00. Thus, the average profit is, AP.

$$AP = \frac{112}{10}$$

$$= N11.20$$

Another important aspect of simulation that made us to use it in inventory is that it is very useful in the sense that the problems of inventory increase in complexity, hence computerized simulation becomes a necessity, especially if more than one of the inventory model components is subject to uncertainty.

That is to say that there is situation very difficult to solve mathematically, especially if the lead time and demand can equal a large range of values. In order to simulate this type of situation, one has to simulate the inventory cost for a specific order size similar to the simulation of Q = 10 first, except if the lead time and demand vary.

3.3 OTHER TECHNIQUES OF STOCK CONTROL

In any business, it is necessary to price and value goods for the purposes of stock controlling stock is according the requirements from different companies, manufacturing houses, warehouses, stores, etc.

The need for doing on any of these is that sometimes the nature and high value of goods demands, that items be individually identified and priced. Than, applying different methods becomes necessary especially where goods are handled in quantity. The followings are the various techniques that can be used.

a. LOT METHOD

In this method each consignment of goods received is kept separately and given a lot number: An average price per unit is calculated for each lot and is used there after for stock control, valuation and cost accounting purposes.

This method demands that each issue from stock must be identified by the lot number and priced accordingly. It also infers that a separate stock record or sub-record is kept for each lot.

b . **FIRST IN(COME) FIRST OUT (SERVE) (FIFO)**

This method is not similar to the above method. The difference is that when each consignment is received a price is calculated and issues charged at this price until a quantity equal to that of the consignment has been issued. Subsequent issues are charged at the price of the next consignment and so on. It is assumed that stocks are used in strict rotation.

c. **LAST IN FIRST OUT (LIFO)**

Issues from stock are priced at the price paid for the last consignment received. When prices are use, it is to ensure that costs are not underestimated, though it will obviously give use to discrepancies in stock reconciliation. Lifo is not acceptable as the basis of stock valuation for tax-computation purposes.

d. **CONTINUING'AVERAGE METHOD (CAM)**

In this method, a new average price is calculated each time a consignment is received. This is done by taking the value of the stock in hand at the old price, adding to it the value of the new consignment at the new price, and calculating a new average price to be effective from that date.

e **STANDARD PRICE METHOD (SPM)**

Here, the estimated standard prices are used over a period for charging out issues, regardless of the actual cost of the various consignments. The standard price is, therefore, calculated on the expected cost of each item of goods. When a new consignment is received, the actual value is compared with the standard value.

If there is difference after the comparism, this is transferred to a price-variance account, so that the charge to the relative control account can be made at the standard price. If significant difference occur and are maintained, it may be necessary to alter the standard price from time to time.

CHAPTER FOUR

4.0 THE SYSTEM SPECIFICATION AND DESIGN

After the detailed analysis of the current system which in way spelt out the strengths of the existing system, this is used at the beginning (onset) of the system development to design the objectives for the proposed alternative. The analysis might lead to a number of possible alternative designs, or instance, different combinations of manual and computerized elements ,may be considered.

Each of these alternatives should have its own required specification worked out. The system specification will be detailed set of input documents, forms, report layouts which provide details of all features of the system from there we go into the design stage. A system specification provides analysis documentation of the entire system. It serves as

- Communication medium to management information system, programmers, operating staff and users.
- Provides complete record of the system used for evaluation modification and training purposes.

So, the system should be designed in such a way that analyst by applying his judgment, skill and knowledge can interpret the requirement specification into so many ways. This will then lead him to documenting the system so that charge of employment will not cause great harm to the business or management. We begin this chapter by viewing various system codes.

4.1· THE CODING SYSTEM SPECIFICATION

Before we can make a useful proposal for Tony Store Enterprises, for Warehouse stock inventory and analysis procedures, it is good for us to establish and briefly expose few of the existing system codes and their functions, especially the ones that gears us towards our achievement. From these system codes we pick the coding method for the new program, reasons for that are necessary.

4.1.1 VARIOUS KINDS OF SYSTEM CODE

a. **SEQUENCE CODE (SC)**

It is the assignment of consecutive numbers, beginning with one, to a list of items as they occur, just like assigning numbers to employees as they are hired.

Code No	Employee Name
1	John
2	Anthony
3	Ibrahim
4	Akin

b. **BLOCK CODE (BC)**

This a minor refinement of the simple sequence code., A series of consecutive code numbers is divided into blocks, each block of numbers being reserved for the identification of groups of items having some common characteristics.

Block code provides a data classification system that uses a few code digits to identify each item. Expansion of the code to include additional items within each block is confined to specific item is therefore determined by the physical make up of the item itself.

Codes	Descriptions
TT 250 18 B1	Tube and Tyre,Size 250x10, blackwall, 1st line
TT 300 17 W1	Tube and Tyre,Size 300x17, whitewall,1st line
TT 350 8 B1	Tube and Tyre,Size 350x8, blackwall, 1st line

e. **DECIMAL CODES**

The decimal codes method are used primarily for indexing libraries or classifying written correspondence. They are used for subject classification.

Decimal Code	Subjects
520.	Astronomy
530.	Physics
532.	Mechanics
531.1	Machines
531.11.	Liver and Balance
531.12	Wheel and Axle

f. **MNEMONIC CODE (MC)**

This is the combination of letters and numbers to describe the items coded, the combination having been derived from the descriptions of the items themselves. The combinations are designed to be an aid to memorize the codes by associating them with the items they represent.

Code No

Item	Size	Colour and style	Mfr	Item Description
By	010	RB	01	Bicycle,10",red boy's comet.
By	010	RB	02	Bicycle,10", red boy's, Redstar.
By	020	RB	01	Bicycle, 20",red boy's, comet.

g. <u>CONSONANT CODE (cc)</u>

This coding method is made up of abbreviations of the alphabetic data items themselves. The initial letter is always retained, but all subsequent vowels are dropped. Codes derived in these fashion from the differbnt item characteristics will be unique to the items and may be used for sorting procedures.

Consonant codes	Item Name
PMPHNDLS	Pumphandles
JNS	Jones
SMTH	Smith
SNDL	Snodel
SNWLY	Snowley
TRCK MTR	Track Material

4.1.2 **<u>THE CODING SYSTEM SUITABLE FOR THE NEW PROGRAM</u>**

The coding method we shall adopt in order to achieve our aim is the combination of codes: Bock, significant - digit and mnemonic codes: It becomes necessary and essential for us to describe each items sufficiently well. Also, the combination is important and necessary because of the following reasons.

a. **EXPANDABLE**

The code provides space for additional entries within each classification for new terms. There must be capacity to expand existing classification and add new ones to take care of future changes.

b. **PRECISE**

The code structure is such that only one code may be correctly applied to a given item.

c. **CONCISE**

The code required the least possible number of digits that would adequately describe each item.

d. **CONVENIENT**

The code is easily understood by each user and simple to apply, whether encoding or decoding.

e. **MEANINGFUL**

The code is meaningful because it indicates some of the characteristics of the items

f. **OPERATABLE**

The code is adequate for present and future processing machine methods, as well as for manual operation.

4.2 THE SYSTEM SPECIFICATION

The system specification is a full description, an analysis and documentation of the input, output, processing and file interaction layouts created in the study of the preliminary information, and this in turn will help us make a concrete system design.

4.2.1 THE TRANSPUT AND PROCESSING SPECIFICATION

The transput specification includes the input, processing and output layouts.

4.2.1 THE INPUT SPECIFICATION

According to the name, the input is the medium by which data, instructions are entered into the computer system. That is to say that it serves as the point of entrance of any job to be processed by the computer and is later passed onto the central processing Unit (CPU).

In the input specification, eight (9) digits two have been adjusted to make up the item code number and 9 digits 2 letters for inventory code (stock item code). The letters and the first three digits represents the inventory code and specific spare part respectively. The next two digits show the importer, or the state of origin, or the manufacturer, and the last number indicates the kind of machine to use the parts. Thus, anyone working with inventory item in Tony store Enterprises will know the following information:

Spare part - machine - Importer/Manufacturer. For input data for this program go to the Appendix.

4.2.1.2 THE PROCESSING SPECIFICATION

Computer can be instructed to perform operations like addition, subtract, multiplication and division. It has logic capability to make comparisons between numbers and between words. As the case may be a computer properly instructed can do not only random and sequential processing, but also do analysis and report on the result.

In random processing, the desired information are accessed directly from the storage medium. While in sequential processing, the same computer system searches the storage medium from the beginning to the end medium to find the desired data.

Every system requires some operations which instructs the computer system to carry out certain specific task of database management operations in order to upset the problems with Tony Store Enterprises.

<u>Addition</u>
Purchase
Sales
Debit/Credit]
Expenses
Income
Stock
Item
Customer
Bank

<u>Modification</u>
Purchase
Sales
Debit/Credit
Expenses
Income
Stock
Item
Customer
Bank

<u>Deletion</u>
Purchase
Sales
Debit/Credit
Stock
Item
Customer
Bank

Inventory
Purchase
Stock
Item
Bank

Analysis
Purchase
Sales
Debit/Credit
Expenses
Income
Bank

4.2.1.3 THE OUTPUT FILE SPECIFICATION

The output is the result of computer execution. While the output devices are used to communicate the result of the operation performed onto the operator. The output may be on monitor or in a hard copy (printer form).

The output file contains the processed information. The output devices such as magnetic disks and magnetic tapes are used to store data, results and programs.

4.2.2 THE FILE STRUCTURE AND ORGANIZATION

In the design stage, the analysis is concerned with file structure and organization because of the main purpose of file creation. It is obvious that files are set-up in order to centralise information. They (files) are also organized in such a way that a person can have easy access to the stored information without hunting for it in a hit and Miss fashion. Hence, it becomes necessary and essentially important that we shall indicate the operational file structure and organization suitable for the problem at hand.

4.2.2.1 THE OPERATIONAL FILE STRUCTURE

The operation to be performed by a file depends on the role of its records and also on its possible interrelationship with other existing files within database, especially Arago. These are the files that will play a prominent role in Tony Store Enterprises.

a. **MASTER FILE**

This will contain all information that are permanent and principal to the business. The up-to-date and accuracy of this file can be achieved by continuous maintenance by the management.

b. **TRANSACTION FILE**

The transaction file will contain the details of transactions, such as item and prices customers name and address. It (transaction file) will be used to update the master file. At certain determined period this file will be emptied for fresh records to enter.

c. **BACKUP FILE**

Backup file, though not essential to the main stream of the business, is still necessary because it is used as a security file. It is normally used to backup master file and/, or transaction file against file corruption.

d. **TABLE FILE**

This is a special kind of master file that will be used by the user for data that is to be referenced repeatedly during processing. Also, it will contain reference data that can be used in processing transaction, updating master file and producing output.

4..2.2.2 THE FILE ORGANIZATION

The way a file is organized depends on the storage medium used and the way the file is to be processed. The stocktaking of Tony Store Enterprises needs a special kind of file organization that will allow access to a large records in a random fashion. This is because the physical sequence of storage imposed on the storage medium will be incompatible with the above logical requirement of the organization. Also, the data processing requirement has to be customized to the need of the management.

Thus, the suitable file organization that satisfy the needs and conditions is the

INDEXED SEQUENTIAL FILE ORGANIZATION

Which is the combination of the services of the sequential organization (which provides efficient processing of file sequentially) and the services of the random file organization (which provides very fast direct access). Hence, this will help the director to process information directly without going through all the records in the file. Again, it will enable him to make for insertion and deletion when and where necessary.

4.3 THE SYSTEM DESIGN

What we need to do in the design phase is to specify the relative position of each of the items in the master file and the transaction file. Each of the characters in the system specification is important in making these decisions. Thus, the results of our design will be demonstrated in the flowchart diagram.

The system design is the designing of the overall program structure provided already by the logics of system specification. This processing logic specification is possibly illustrated by a number of techniques available to help programmers analyse a problem and design the program. The technique we shall use here is already stated above which shall be eventually converted to database management system, arago to be specific.

4.3.1 FLOWCHART DIAGRAM

It is the documentation of the design of the new system software. Flowchart diagram is most often used in data flow analysis which studies the use of data in each activity. Hence, it is concerned with the flow of data through organization processes, especially in handling transactions and competing tasks.

Flowchart diagram documents the findings in data flow diagram which physically shows the relation between processes and data. Then, it graphically displays the fundamental operations of the intended system. Thus, it serves not only as a general descriptions of the output reports to be produced by the system, but also serves as a data dictionary which formerly describes the system data and where they are used.

Furthermore,, it is used to graphically illustrate data, information and work flow by the interconnection of specialized symbols with flow lines. The combination of symbols and flow lines portrays the logic of the program. It is logic if and only if the transformation of input to output via processes can be described logically and independent of the physical component, e.g. files, operator.

Each symbol indicates the type of operation to be performed. The flowchart diagram graphically demonstrates the sequence in which the operations are to be performed, while the flow lines indicates the direction of the program logic. In the flowchart diagram, we can use Yordon notations.

4.3.2 THE PROPOSE SYSTEM SOFTWARE

The software to be used in designing the program is called Arago. It is in the family of Database Management system. It seems better that Database because of the following reasons:

a. It is faster than Database.

b. It can work on Local Area Network.

c. It can run under any operating system.

d. It can be compiled like DOS and also it can generate exe. file.

e. It is object oriented.

f. It has highly advanced features.

4.4 COSTS AND BENEFITS ANALYSIS OF THE NEW SYSTEM

The costs and benefits analysis clearly manifest to the management of Tony Enterprises in quantitative rather than qualitative terms the costs of introducing those computer technologies and the benefits his business will accrue from these innovations. This analysis will also help us to select from several alternatives the one that is more realistic. Hence, we discuss these under three headings in two stages (present and future).

a. Operating Costs.

b. Development costs.

c. Installation costs.

d. The benefits of the new system.

4.4.1 STAGE ONE: PRESENT

-	**OPERATING COSTS**	**N**	**K**
-	Software package	5,000 =	00
-	Hiring operator/computer twice a year	1,000 =	00
-	**DEVELOPMENT COSTS**		
-	Staff training (for 9 months at N2,500)	7,500 =	00
-	**INSTALLATION COSTS**		
-	1 Packet of Diskette	1;500 =	00
-	Miscellaneous	500 =	00
	Total	6,000 =	00
		7,500 =	00
		2,000 =	00
		15,000 =	00

4.4.2 STAGE TWO: FUTURE

-	**OPERATING COSTS**	**N**	**K**
-	Computer	60,000	=
-	Printer	40,000	=
-	Mouse	2,000	=
		102,000.	00
-	**DEVELOPMENT COSTS**		
-	System Analyst (2months at N10,000.00 each)		
		20,000 =	
-	Programmer (1 month at N10,000.00)		

		10,000	=
-	Operator (3 months for N2,500)		
		2,500	=
		32,500	00
-	**INSTALLATION COSTS**		
-	Stationeries (bulk purchase)	20,000	=
-	Miscellaneous	5,000	=
		25,000	00
	Totals	N102,000.00	
		+ 32,500.00	
		+ 25,500. 00	
		N159,500. 00	

4.4.3 **BENEFITS GAINED FOR INSTALLING THE NEW SYSTEM**

The benefits analysis covers the present and the future growth of the business. The first to be mentioned and discussed in

1. **RAPID AND ACCURATE CALCULATION**

Computer speed up complex calculation. It is even faster and more efficient at doing complex analysis the warehouse stock inventory in a very shorttime. Once the procedures are specifically defined, and appropriate data is fed, rapid processing and accurate output is guarantee.

2. **RELIABILITY**

Computer systems are perfect in doing repetitive tasks. The do not go on break nor complain of being "overworked" expect at, "sickness".

3. **MEMORY CAPABILITY**

Computers have assumed unlimited capacity of memory storage. This memory capacity in computer systems has the total and instant recall of data - no forgetfulness in computer once the appropriate command for retrieval is given.

4. **STORAGE OF DATA**

Through the means of disks (diskettes), to be specific 3.5" or 5.25", and tapes, all computer systems can store large amount of information, provided the spaces are well managed.

5. **RETRIEVAL OF INFORMATION**

Due to the complexity in Tony's Enterprise which needs the indexing system for cataloguing information based on a single key field, manual storage and retrieval cannot give satisfied accessibility to information on individual markets store. With a computer, the necessary and essential data will be accessed in all these ways and more and they may also be accessed by more than one person at the same time (Network).

6. **MODIFICATION OF STORED DATA**

It is possible to change and modify data stored on a computer.

7. **DATA ANALYSIS**

Once computer has been applied, the information stored in it can be summarized and be broken down. Within a marketing information system, the same data can be analysed by individual market, by price bracket, or request a list of all good sold by a particular seller or a given day and date.

8.· **TRANSFER OF DATA**

A computer can provide data or information for another computer, or two or more files sharing the information.

9. **CONTINUOUS OPERATION**

With precise and correct command, a well-designed computer system will allow untrained, but educated users, even very busy directors, to use an information system or access a database with no more still than is needed to operate a video recorder. It will be available all day and night; and will not get toked with repeating the same task over and over again.

CHAPTER FIVE

5.0 THE SYSTEM CONVERSION, IMPLEMENTATION AND PROGRAM LISTING

Following the steps already developed, next is to write a source programm code which will be coded into the computer system for a required output. When the program has been finally coded,then,testing and debugging will be done to ensure error-free program. After that then we go into recommendation and conclusion.

Testing of the program will be in two phases:

a. With artificial data by testing and compiling each subsystem.

b. With real data, in which the system is tested with the actual data to provide an extra level of assurance that the system will work properly when implemented.

An information system to be what it should be must be however dynamic, totally objective, and must be responsive to the changing needs of the business that uses it. There must be room for the system to be modified to meet the needs of the changing environment. This process of modification is called the system maintenance. At each step the file is protected against operations that call for file destruction.

5.1 WRITING THE PROGRAM

This is the programming phase of the system development process, through which ideas formulated in the above stages are transacted into reality.

Writing the program is another way of saying coding the program. It is coding the program with the appropriate computer language. Also, it is the translation and transformation of the system specification and design into instructions that can be interpreted and executed by computer. This program development is actually a project within a project in which case if the logic is sound and the design documentation is thorough, the coding process is relatively straightforward and simple before testing and debugging.

5.2 TESTING AND DEBUGGING THE PROGRAMM

Testing involves the development of quality assurance procedures, including data security, backup and recovery, and system controls. It also involves testing programs with both artificial and actual data. In-short it covers everything that makes up the information system:- the hardware, the software, the end users, the procedure and the data. Testing and modification are so important and necessary in order to ensure that the components of the system work as they should and all transput devices are validated.

Once we have succeeded in coding the program there may bug or bugs as the case may be. As error or a bug exists either as a syntax error (violation of one of the grammar) rules of writing instructions in a program language), or a.logic error (wrong logical steps). Once this is noticed within the program debugging becomes essentially important. Hence ridding off a program bug is the process of debugging. The program errors are dictated as a result of testing the system with data. So, errors in this program can be seen by testing the software with data.

Thus, testing and debugging of a program is the testing and removing of syntax and logic errors for the program to be error-free before execution by computer. What of the system conversion.

5.3 THE SYSTEM CONVERSION (CHANGEOVER)

In Tony Store Enterprises, the old system of operation has remained manual, now, as we are changing over to a new system, we need to keep both the old and the new systems a life, where and when necessary. For this reason, therefore, the system conversion suitable for this business is PARALLEL RUNNING.

This (parallel Running) involves processing current data by both the old and new systems to cross-check the results. In this case therefore the old system is kept working until the new system is kept working until the new system is proved beyond reasonable doubt to be working.

5.4 FILE SECURITY

Magnetic disk or tape is the mechanisms that when properly arranged stores data according to files for computer usage. In other to allow or let the stored information last long there is need for the mechanisms to be secured against viruses and unauthorized persons getting access into the stored information. Hence, file security is the steps taken to protect a computer system and the stored information from certain operation that will result to loss of data.

Among all the file control available, the use of password and backup are recommended to the management of Tony Store

Enterprises because they will give the management the necessary satisfaction. By the use of password, the stored data on the new system will be accessible only to those authorized. But it requires one to remember the password.

Backup is mainly used for the system being infected with viruses. The general system of backup to be used is known as "grandfather-father-son". The grandfather should be kept away from the father and the sone, while the father will be housed away from the son. The grandfather should be recalled at the extreme case of eventualities.

5.5 INSTALLATION AND MAINTENANCE

Installation is the process of preparing a program for use on a computer system. It is possible if and only if the following conditions hare been satisfied.

- The system must undergo tests.
- The training of operators must have been completed.
- Date and time for the installation are due.
- The type of system changeover must be made clearer to the management.
- The system implementation and maintenance must be simple and clear.

Maintenance includes whatever changes and enhancements needed to be made after the system is up and operating. Program maintenance can be difficult if the program documentation is not complete and up-to-date because documentation is part of the programming process.

5.6 **PROGRAM LISTING:** Program listing serves as a guide to the user. It helps the person to know which program is for which operation. They are as follows:

INTRODUCTION PROG: This program introduces the entire program. It is as kind of general introduction to the project.

PASS-D-PGR: The password program is the gateway to the whole information. Not knowing the pass-d means no access to the information.

MAINMENU-PRG: The mainmenu program allows us to select from the multi options what to do.

RECADD.PRG: The mainmenu program allows us to select from the multi options what to do.

RECMODI.PROG: This is program modification which gives room for one to select the file concern.

RECDEL.PRG: Recdel.Prg is the record deletion program which also allows one to select the file to delete record from.

A-WARES.PRG: A-Wares is a subprogram that concatenates the spare parts with the importers, or town of make, of manufacturers and the kind of machine.

A-CASH.PRG: It is a subprogram that only calculates the amount with regards to purchases and sales.

DEBCRED.PRG: This is a subprogram and also a main program that keeps the records of creditors and debtors.

J-PURCHASE.PRG: It is a subprogram that works with the modification main program for necessary modification.

INVENTORY.PRG: Inventory.Prg is a program used to make for inventory and mini analysis of stock in the warehouse.

ANAL-REP.PRG: This is analysis and report program which helps us to make either quarterly or annual analysis of business transactions, and prints the reports either in a soft or hard copy form.

5.7 RECOMMENDATION

For proper managerial information system, this computer system and peripherals are found appropriately suitable and has been recommended for use in Tony Store Enterprises. They are as follows:

Computer:	IBM PC/AT.
Size:	486 DX2.
Memory:	Over 32 MB.
Hard Disk:	Over 120 MB.
Disk drive:	Hard Disk Drive Floppy Disk Drive.
Cache Type:	Write-Through.
Accessories:	MS DOS 5.0 Dr. Solomon (Anti-Virus). Windows.
Software:	Wordperfect.

DBase IV

Arago.

Fox Pro

Clipper

Power Supply: UPS (Uninterupted Power Supply).

Printer: HP Laser jet Mac version (MO).

There is need for the end user to know why the above computer systrem with its accessories are recommended. The reasons for such recommendations are as follows:

- 486 Dx2 has an ibuilt over drive which can be upgraded to pentuim.
- High speed and reliability of processing is assured.
- The ROM and RAM have high storage capacity.
- Cost factors are considrable cheaper.
- Connectively is possible: .Can be used as central server in a network environment.
- Efficiently accommodates parallel processing to a fairly good degree.
- The printer can be used in IBM computers.

Insofar as most of our national projects, like the ongoing National Elections, Football leagues series of competitons could be conducted and accomplish with optimum speed by the use of computer. Also due to the fact that computer provides longer life indirectly, reduces working hours and assures us of greater remuneration in return. It is important that Federal Government of Nigeria should inshrine Maths/Statistic/Computer Science into shcools curricula and make it a compulsory subject idea, they should draw up shemes that should cover certain strategic areas of computer science from Nursery/Primary up to University level because mathematics is a key to every technology and computer is the cement that corlds the skills.

5.8 CONCLUSION

Computer literacy is already a prerequisite to employment in some professionss and businesses. Thus, persons seeking employment these days are often embarrased with the question: "Are you computer literate?

In most advertisments, computer education is now an added advantage so that a secretary without computer literacy will soon be out of job. Bank clerks, Bank supervisors, accountants, telphone operators, etc. Without computer literacy will soon discover that the world is not bed of roses.

Let us make haste while sum shines. A good philosopher findss gold where all others are able to grab bronze. It is because a stitch in time saves nine.

REFERENCES

1. AKIN, F., Understanding And Using Microcomputers, Abuja-Nigeria: AFLON LED, 1995.

2. AYO, C.K. Computer Literacy, Kogi-Nigeria: Alahukitan Commercial Press, 1994.

3. GREENE, J. J.(ed.), Production And Inventory Control Handbook, New York: Mchraw-Hill BK. Comp., 1987.

4. HARMAN, K.' W. AND BORDEN, J. P., Lotus 1-2-3 and Dbase software Applied to Ais Cases. New York: John Wiley and Sons, Inc; 1992.

5. HARPER, W.M., Operation Research, 2nd ed., Britain: Longman Pub. Comp., 1998.

6. JONES, H.L. AND JONES R.P., Groundwork of Commerce London: African University Press, 1973.

7. KAZMIER, L. J. AND POGL, N. F., Statistics for Business and Economics, London: McGraw-Hill BK. Comp., 1987.

8. MASON, R. D. AND LIND, D. A., Statistical Techniques in Business And Economics, Boson: Homeword trwin Pubs., 1976.

9. MILLS, G. AND STANDINGFORD, O., Office organization And Method: A Manual of Administrative Management, London: Pitman Bks, Ltd, 1983.

10. PAYNE, B. P., Management systems and Computers, New York: Alexander Hamilton Institute, 1986.

11. SALVENDRY, G., Handbook of Industrial Engineering, New York: John Wiley and Sons, Inc, 1992.

12. TAYLOR, B. W., Introduction to Management Science, 2nd ed., IWA: W.C. Brown Publishers, 1990.

13. THOMAS, E. S., Commerce: Its Theory and Practice, London: Macmillan Pub. Inc, 1981.

14. CHAMBER 20TH CENTURY DICTIONARY.

15. ENCYCLOPEADIA BRITANNICA, VOL. 17.

APPENDIX A

FIGURE 1.0: INTRODUCTORY FLOWCHART

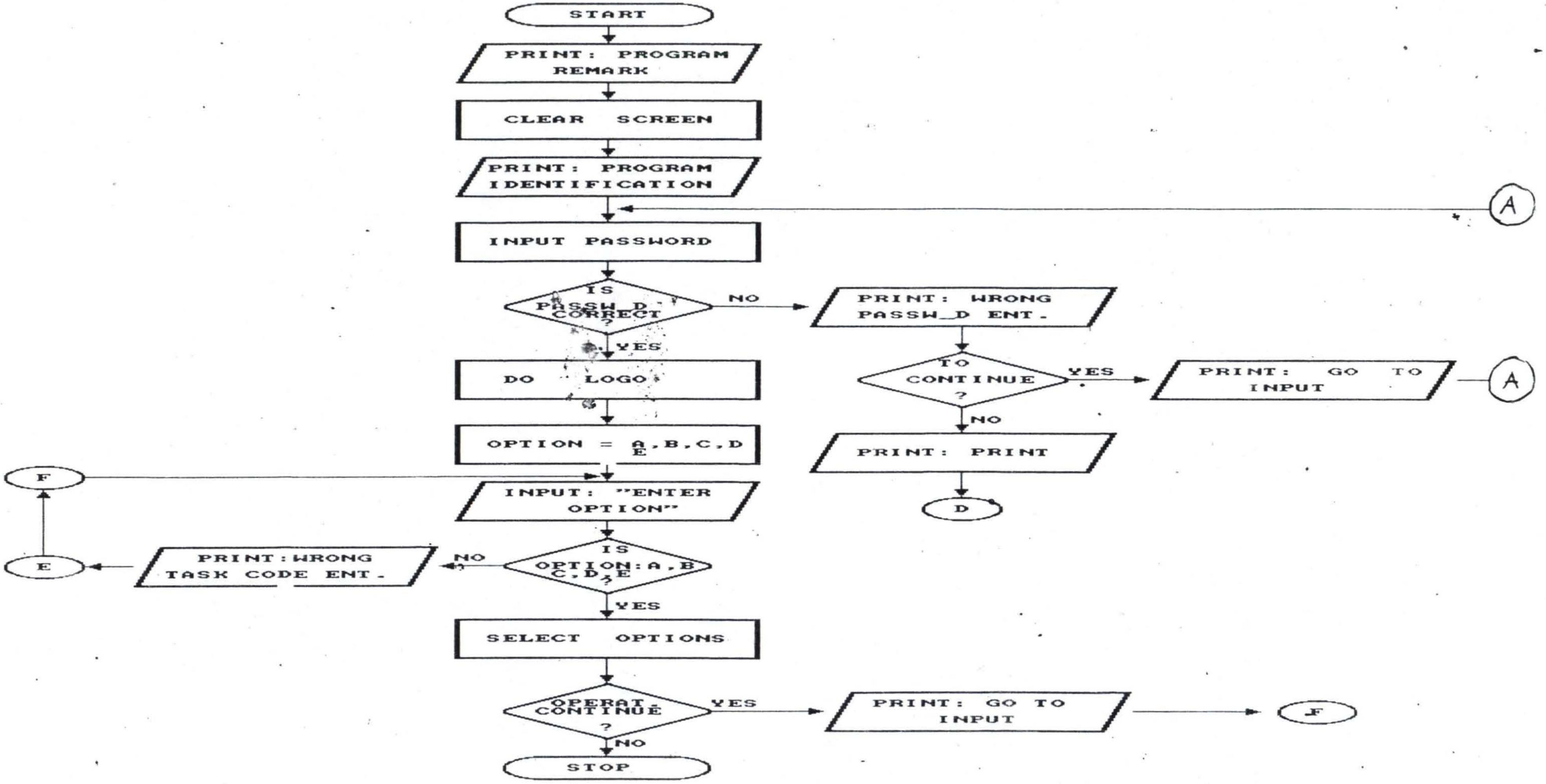


FIGURE 1.1: FLOWCHART FOR ADDITION (MAIN PROGRAM)

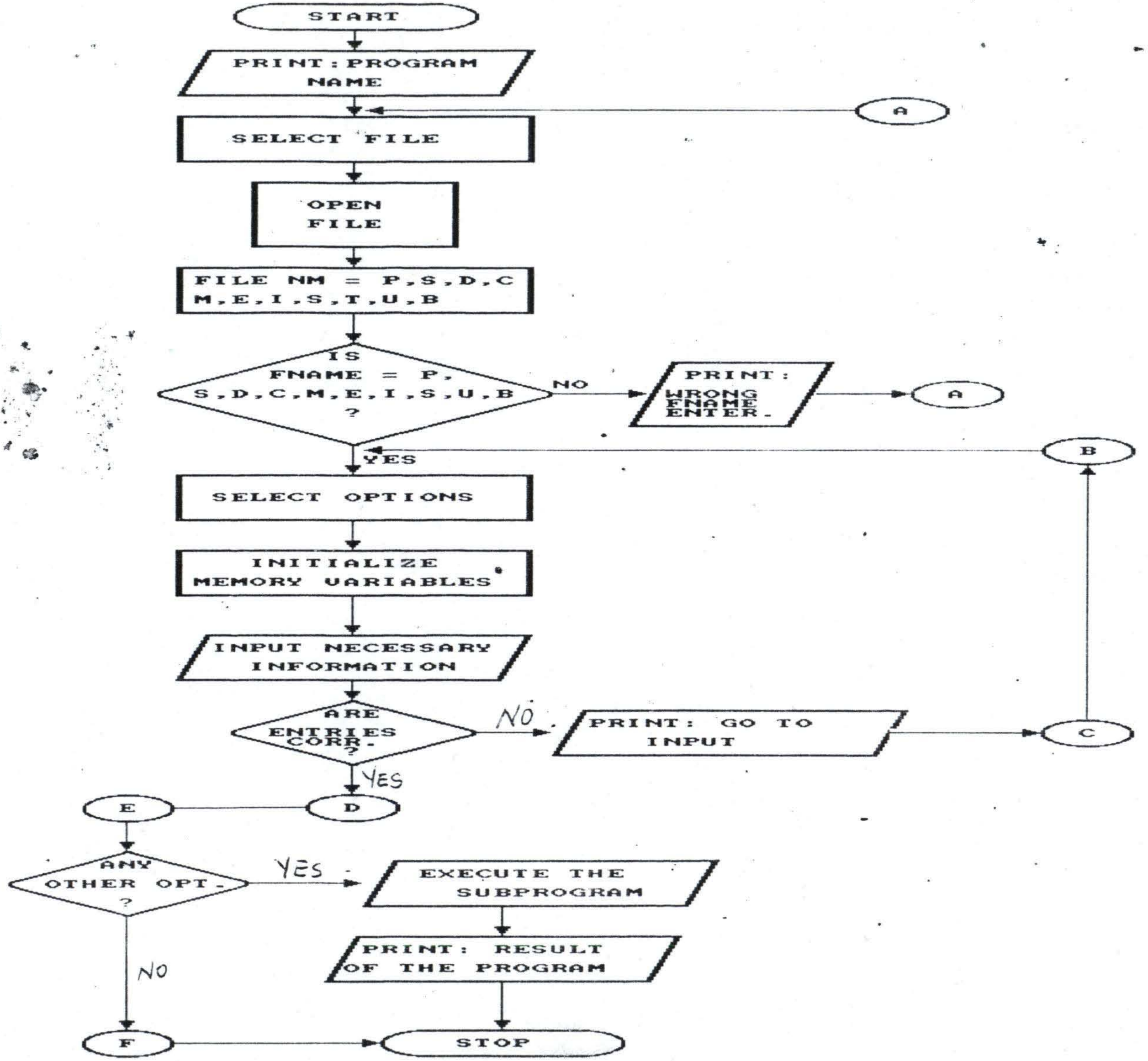

START
PRINT: PROGRAM NAME
A
SELECT FILE
OPEN FILE
FILE NM = P,S,D,C M,E,I,S,T,U,B
IS FNAME = P, S,D,C,M,E,I,S,U,B ?
NO
PRINT: WRONG FNAME ENTER.
A
B
YES
SELECT OPTIONS
INITIALIZE MEMORY VARIABLES
INPUT NECESSARY INFORMATION
ARE ENTRIES CORR. ?
NO
PRINT: GO TO INPUT
C
YES
E
D
ANY OTHER OPT. ?
YES
EXECUTE THE SUBPROGRAM
PRINT: RESULT OF THE PROGRAM
NO
F
STOP

FIGURE 1.2: FLOWCHART FOR PROGRAM MODIFICATION

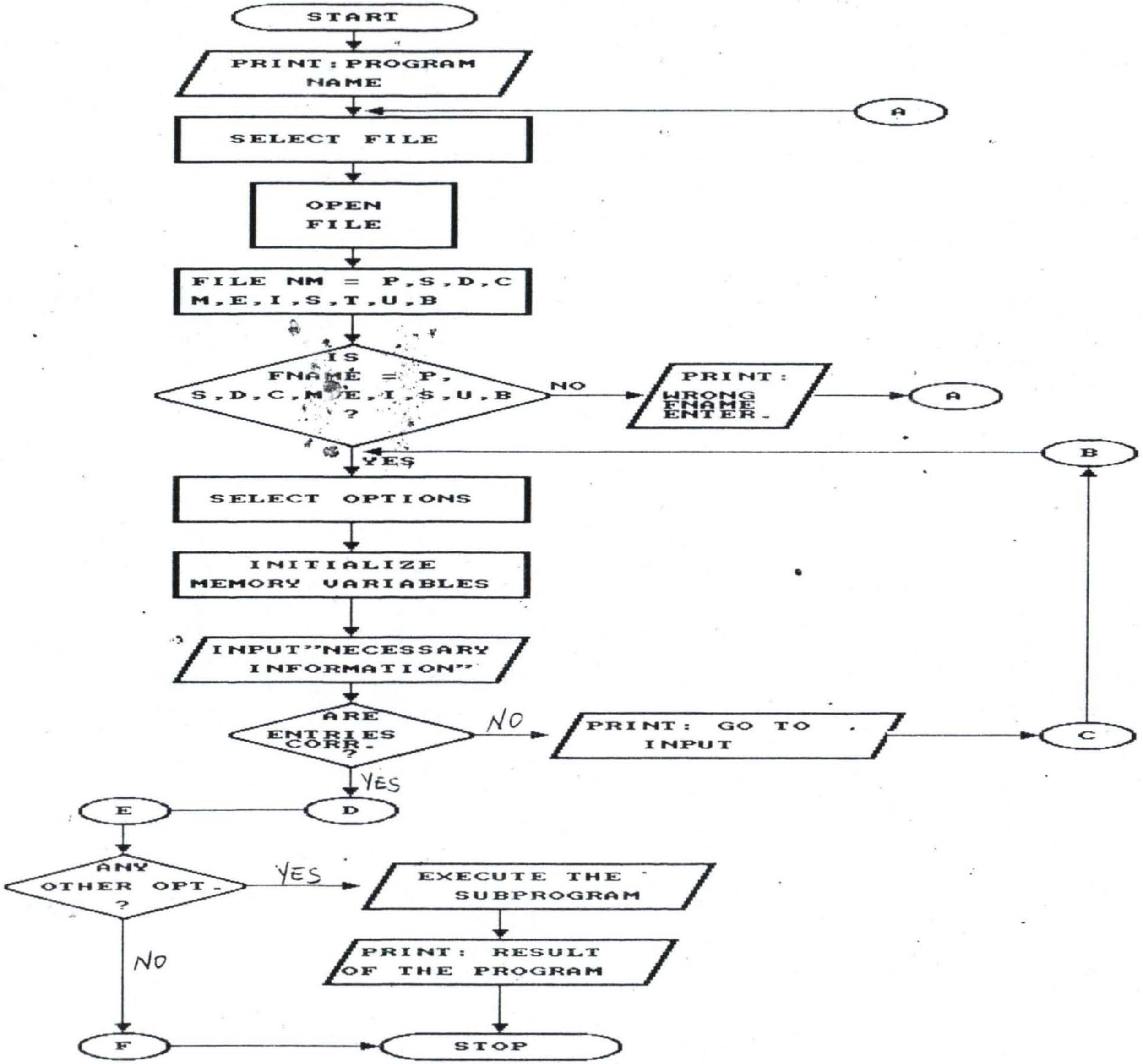
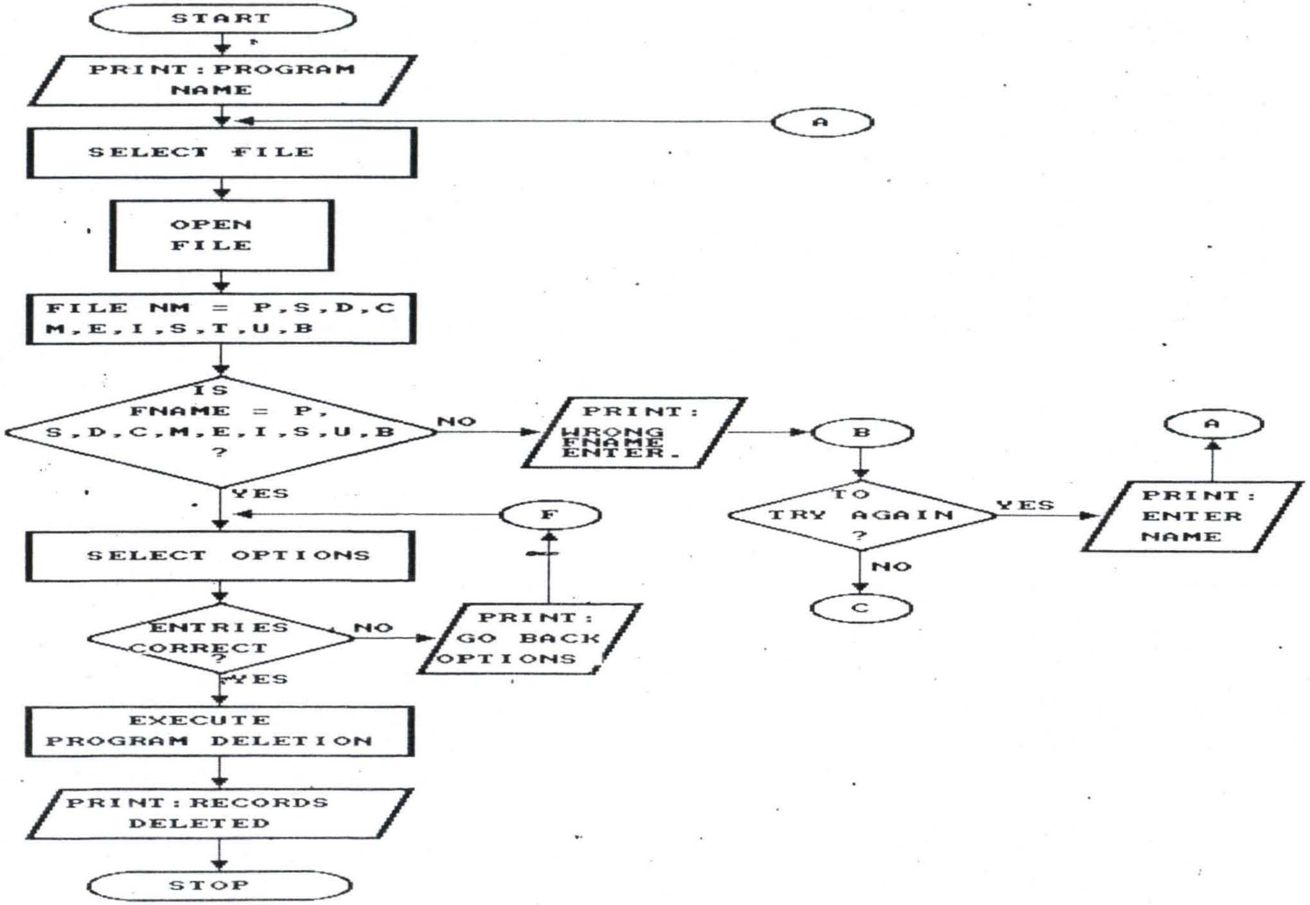

START
PRINT: PROGRAM NAME
A
SELECT FILE
OPEN FILE
FILE NM = P,S,D,C M,E,I,S,T,U,B
IS FNAME = P, S,D,C,M,E,I,S,U,B ?
NO
PRINT: WRONG FNAME ENTER.
A
B
YES
SELECT OPTIONS
INITIALIZE MEMORY VARIABLES
INPUT"NECESSARY INFORMATION"
ARE ENTRIES CORR. ?
NO
PRINT: GO TO INPUT
C
YES
E
D
ANY OTHER OPT. ?
YES
EXECUTE THE SUBPROGRAM
PRINT: RESULT OF THE PROGRAM
NO
F
STOP

FIGURE 1.3: FLOWCHART FOR PROGRAM DELETION



APPENDIX B

```
                    ****PROGRAM INTRODUCTION*******
 SET PROCEDU TO
 SET ECHO OFF
 SET COLOR TO B/W
 SET TALK OFF
 SET CLOCK ON
 CLEA
 @1,1 TO 24,79 DOUBL
 @4,2 CLEA TO 22,78
 SET COLOR TO B/W
 @4,2 TO 22,78 PANEL COLOR G
 SET COLOR TO R+/W
 @13,30 SAY "VIS - A - VIS"
 @14,25 SAY "TONY STORE ENTERPRISES"
 M = 1
 SET COLOR TO G+/W
 DO WHILE .T.
 @11,05 SAY " "
 @11,15 SAY "COMPUTERIZATION OF WAREHOUSE STOCK INVENTORY"
 @12,25 SAY "AND ANALYSIS PROCEDURE"
 M = M + 1
 SEC = 0
 SET COLOR TO B+
 AN = SPACE(1)
 IF SEC = INKEY(0.2)
    CLEA
    @2,3 TO 22,78 DOUBL
    @3,10 SAY "COMPUTERIZATION OF WAREHOUSE STOCK INVENTORY"
          @3,54 SAY "AND ANALYSIS PROCEDURES"
    @4,14 SAY SPACE(15) + "VIS - A - VIS"
    @5,14 SAY SPACE(10) + "TONY STORE ENTERPRISES"
    LT = 0
    J_L = 0
    @10,37 SAY SPACE(15) + "MINNA - NIGER STATE"
    @8,10 SAY "A SOFTWARE PREPARED BY:"
    @9,5 SAY "----------------------"
    @9,30 SAY "AZUBUIKE JOHN MARY M."
    @12,5 SAY "AS A PARTIAL FULFILMENT FOR THE AWARD OF PGD IN"
    @13,5 SAY "COMPUTER SCIENCE  FOR MATHS/COMPUTER SCIENCE"
    @14,5 SAY "DEPARTMENT OF FEDERAL UNIVERSITY OF TECHNOLOGY,"
    @15,10 SAY "FOR THE SCHOOL OF SCIENCE AND SCIENCE EDUCATION"
    SET COLOR TO B/W
    SEC = INKEY(0.1)
    CL = 1
    ENDIF
    SET ECHO OFF
    @20,15 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET AN PICT "Y"
    READ
    IF AN = 'Y'
         WAIT
         CLEA
         @14,5 SAY " "
         @17,5 SAY "SUPERVISED BY:"
         @18,5 SAY "--------------"
         @17,20 SAY "DR. YOMI M. AIYESIMI"
         @19,40 SAY "MARCH, 1998"
         CL = CL + 1
         SET COLOR TO B
         SEC = INKEY(0.1)
         WAIT
   ENDIF
IF AN = 'N'
```

```
        CLEA
        EXIT
        ENDIF
        ENDDO
        DO PASS_D
        SET TALK ON
        RETURN


*PROCEDURE MOD_ITEM
*THIS SUBPROGRAM  IS FOR DATA ADDITION INTO ITEM DBASE FILE.
CLEA
SET TALK OFF
SET ECHO OFF
@2,5 TO 15,76 DOUBL
SET COLOR TO GR+/W+
M_STOCK_CD = SPACE(3)
RS = SPACE(1)
M_ICODE = SPACE(9)
M_INAME = SPACE(20)
M_ITEM1 = SPACE(15)
STORE SPACE(10) TO M_ITEM2,M_ITEM3,M_ITEM4
STORE 0 TO M_SPARTC,M_ITMAC,M_KMACHC,M_PRATE
CLEA
DO WHILE .NOT. EOF()
GO TOP
@2,5 TO  22,70 DOUBL
YN = SPACE(1)
SET ECHO OFF
SET COLOR TO GR+
USE ITEM.DBF
@15,20 SAY "ENTER SPARE PART CODE:" GET YN PICT "Y"
READ
*LOCATE COMMAND WILL BE USED TO SEARCH FOR THE RECORD
LOCATE FOR SPART_CODE = M_SPARTC
IF .NOT. FOUND()
CLEA
@11,10 TO 17,70 PANEL COLOR B
@13,25 SAY "WRONG SPARE PART CODE ENTERED ---------!!!"
@15,25 SAY "PLEASE ENTER CURRENT CODE -------------!!!"
@20,28
WAIT "PRESS A KEY TO CONTINUE WORK ----------------!!!"
CLEA
RP = " "
@23,20 SAY "TO CONTINUE (Y/N)?:" GET RP PICT "Y"
READ
ENDIF
IF FOUND()
CLEA
@12,5 TO 16,74 DOUBL
SET COLOR TO W/G+
SET ECHO OFF
@14,25 SAY "TO VIEW RECORD (Y/N)?:" GET RS  PICT "Y"
READ
IF RS = "Y"
@4,6 CLEA TO 17,77
@5,7
LIST ALL
CLEA
SET FORMAT TO ITEM
READ
ENDIF
IF RS = "N"
```

```
SET FORMAT TO ITEM
READ
ENDIF
ENDIF
AN = SPACE(1)
CLEA
SET ECHO OFF
SET COLOR TO G+
@12,10 TO 16,70 DOUBL
@14,15 SAY "TO MODIFY MORE RECORDS (Y/N)?:" GET AN PICT "Y"
READ
IF AN = "Y"
CLEA
RETURN
ENDIF
IF AN = "N"
CLEA
*TRANSFER OF FIELD CONTENTS INTO MEMORY VARIABLES
REPLA STOCK_CD WITH M_STOCK_CD, ITEM_CODE WITH M_ICODE
REPLA ITEM_NAME WITH M_INAME, SPARE_PT WITH M_ITEM1
REPLA I_TMAN WITH M_ITEM2, K_MACH WITH  M_ITEM3
REPLA STOCK_INM WITH M_ITEM4, SPART_CODE WITH M_SPARTC
REPLA I_TMAC WITH M_ITMAC, K_MACHC WITH M_KMACHC
REPLA P_RATE WITH M_PRATE
CLEA
CLOSE ALL
EXIT
ENDIF
ENDDO
SET TALK ON
SET CLOCK OFF
RETURN


PROCEDURE CAL_CASH
*THIS SUBPROGRAM IS USED FOR CALCULATION  OF MONIES LODGED INTO  BANK
CLEA
SET TALK OFF
@13,10 CLEA TO 20,70
SET ECHO OFF
M_LODGE = M_CHEQUE
DO WHILE .T.
AMT1 = M_CD50  * 50
AMT2 = M_CD20 * 20
AMT3 = M_CD10 * 10
AMT4 = M_CD5 *  5
AMT5 = M_CDC1 * 1
AMT6 = M_CDC_50 * 0.50
AMT7 = M_CDC_10 * 0.10
AMT8 = M_CDC_5 * 0.05
AMT9 = M_CDC_1 * 0.01
MGT_AMT =  AMT1 + AMT2 + AMT3 + AMT4 + AMT5 + AMT6 + AMT7 + AMT8 + AMT9
MC_BANK = 0
USE BANK.DBF
@13,5 TO 20,70 DOUBL
@14,10 GET CASHBANK
MGTT_AMT =  0
USE INCOME.DBF
@16,10 GET G_TOTAMT
MGTT_AMT  =  G_TOTAMT
M_CHHAND  = MGTT_AMT - MC_BANK
@12,5 CLEA TO 20,70
```

```
@12,5  TO  20,70 DOUBL
@14,10 SAY MGT_AMT(MGT_AMT,8,2)
@16,10 SAY M_CHHAND(M_CHHAND,10,2)
Q = INKEY(0.3)
WAIT
CLEA
CLOSE  ALL
EXIT
ENDDO
SET TALK ON
RETURN


*PROCEDURE A_EXPENSES
CLEA
SET TALK OFF
@10,1 CLEA TO  20,70
SET DATE BRITISH
SET CLOCK ON
SET COLOR TO GR/RB+
RP  = SPACE(1)
M_DT_EXPEN  = CTOD("TODAY")
STORE 0 TO  MLCSTOCK,M_ETBILL,M_WTBILL,M_RRENT,M_PREMIS,M_LANDB
STORE 0 TO M_MISCELL,M_PPAY,M_LDINWD,MCOUTWD,MLDRAWS,M_BDEBT,M_SNO
DO WHILE .NOT. EOF ( )
GO TOP
USE EXPENSES INDEX EXP_DATE
*USE APPEND BLANK TO CREATE SPACE FOR  MORE IN THE EXPENSES FILE
APPEND BLANK
SET FORMAT TO EXPENSES
READ
*CALCULATE BY ADDING THE EXISTING MONEY WITH THE INCOMING ONE
M_GTAMT_E = M_GTAMT_E + M_EXPAMT
*TRANSFER OF FIELD CONTENT INTO MEMORY VARIABLES
M_DT_EXPEN = DT_EXPENS
MLCSTOCK = LC_STOCK
M_GTAMT_E = GTOT_EXPEN
M_ETBILL = ELECBILL
M_WTBILL = WATERBILL
M_RRENT = RATERENT
M_PREMIS = PREMISES
M_LANDB = LANDBD
M_MISCELL = MISCELL
M_PPAY = PPAYMENT
MCOUTWD = C_OUTWD
M_LDINWD = LD_INWD
M_BDEBT  = BADDEBT
M_LDRAWS  = L_DRAWS
M_SNO = S_NO
*REPLACE COMMAND WILL BE USE TO REPLACE CONTENTS INTO DATABASE FILE
REPLA DT_EXPENS WITH M_DT_EXPEN, GTOT_EXPEN WITH M_GTAMT_E
REPLA LC_STOCK WITH MLCSTOCK, ELECBILL  WITH  M_ETBILL
REPLA WATERBILL WITH M_WTBILL, RATERENT WITH M_RRENT
REPLA PREMISES WITH M_PREMIS, LANDBD WITH M_LANDB
REPLA MISCELL WITH M_MISCELL, PPAYMENT WITH M_PPAY
REPLA C_OUTWD WITH MCOUTWD, LD_INWD  WITH M_LDINWD
REPLA BADDEBT WITH M_BDEBT, L_DRAWS WITH M_LDRAWS
REPLA S_NO WITH M_SNO
@22,25
WAIT "ADD MORE DATA (Y/N)?:" TO RP
IF UPPER(RP) = "Y"
CLEA
```

```
LOOP
ELSE
CLEA
CLOSE ALL
EXIT
ENDIF
ENDD
SET TALK  ON
SET CLOCK OFF
RETURN


*PROCEDURE ADD_INCOME
*THIS SUBPROGRAM DEMONSTRATE INCOME DATA FILE.
CLEA
SET CLOCK ON
SET TALK OFF
SET COLOR TO N/W
@3,5 CLEA TO 20,70
SET DATE BRITISH
STORE " " TO ANS
@2,10 CLEA TO 21,78
DO WHILE .NOT. EOF()
@2,5 CLEA TO 21,78
M_I_DATE = CTOD(TODAY)
M_DESCPT = SPACE(25)
M_FILE_NO =  SPACE(3)
STORE 0 TO M_AMOUNT,M_GD_TAMT
USE INCOME INDEX INCOM_DATE
*WE USE  APPEND BLANK  TO MAKE ROOM FOR THE ADDITION  OF MORE DATA.
APPEND BLANK
SET FORMAT TO INCOME
READ
*TRANSFER OF FIELD CONTENTS INTO MEMORY VARIABLES.
M_I_DATE = INCOM_DATE
M_DESCPT =  INCOM_DES
M_FILE_NO  = FILE_NUMB
M_AMOUNT =  TOT_AMT
M_GD_TAMT = G_TOTAMT
*WE USE REPLACE COMMAND TO REPLACE  THE RECORDED INFORMATION.
REPLA INCOM_DES WITH M_DESCPT, FILE_NUMB  WITH M_FILE_NO
REPLA TOT_AMT WITH M_AMOUNT, G_TOTAMT WITH M_GD_TAMT
@22,15 SAY "ADD MORE RECORDS (Y/N)?:" GET ANS PICT "Y"
READ
IF UPPER(ANS) = "Y"
CLEA
LOOP
ELSE
CLEA
CLOSE DATABASE
EXIT
ENDIF
ENDDO
SET TALK ON
SET CLOCK OFF
RETURN

*PROCEDURE A_CUSTOMER
*THIS SUBPROGRAM ADD THE CUSTOMERS PARTICULARS.
CLEA
SET TALK OFF
@3,10 CLEA  TO 18,70
SET COLOR TO RG+
SET CLOCK ON
YN  = " "
M_CUST_CD = SPACE(4)
M_CUST_ADD = SPACE(25)
STORE SPACE(10) TO M_LASTNM,M_FIRSTNM,M_OTHERNM
@2,5 CLEA TO 20,72
DO WHILE .NOT. EOF()
GO TOP
@3,5 CLEA TO 20,70
USE CUSTOMER INDEX CUST_CODE
*APPPEND BLANK COMMAND IS USED TO CREATE ROOM FOR DATA ENTRY.
APPEND BLANK
SET FORMAT TO CUSTOMER
READ
M_CUST_CD = CUST_CODE
M_CUST_ADD = CUST_ADD
M_LASTNM = LAST_NM
M_FIRSTNM = FIRST_NM
M_OTHERNM = OTHER_NM
REPLA CUST_CODE WITH M_CUST_CD, CUST_ADD WITH M_CUST_ADD
REPLA LAST_NM WITH M_LASTNM, FIRST_NM WITH  M_FIRSTNM
REPLA OTHER_NM WITH M_OTHERNM
@20,10 SAY "TO ADD MORE RECORDS (Y/N)?:" GET YN PICT "Y"
READ
IF UPPER(YN) = "Y"
CLEA
RETURN
ENDIF
IF UPPER(YN)  =  "N"
CLEA
CLOSE  ALL
EXIT
ENDIF
ENDD
SET TALK ON
SET CLOCK  OFF
RETURN

* PROCEDURE A_WARES
*THIS SUBPROGRAM IS USED FOR DATA CONCARTINATION
CLEA
SET DATE BRITISH
SET TALK OFF
@2,5 CLEA TO 20,78
@3,10 CLEA TO 22,70
SET ECHO OFF
SET COLOR TO RB
YN = " "
DO WHILE .NOT. EOF()
GO TOP
@2,2 CLEA TO 22,78
USE ITEM INDEX SPART_CODESTORE SPACE(25) TO ITME1, ITEM2, ITEM3, ITEM4,
ITEM5, ITEM6, ITEM7, ITEM8, ITEM9
STORE SPACE(25) TO ITEM10
APPEND BLANK
SET FORMAT TO ITEM
READ
DO CASE
CASE M_STOCK_CD = 'C10'
M_ITEM4 = 'CHAINS ----- A'
CASE M_STOCK_CD = 'C20'
M_ITEM4 = 'CHAINS ----- B'
CASE M_STOCK_CD = 'C30'
```

```
M_ITEM4 = 'CHAINS ----- C'
CASE M_STOCK_CD = 'K40'
M_ITEM4 = 'KEYS'
CASE M_STOCK_CD = 'L45'
M_ITEM4 = 'LIVERS'
CASE M_STOCK_CD = 'B50'
M_ITEM4 = 'BOLT/NUTS'
CASE M_STOCK_CD  = 'G60'
M_ITEM4 = 'GUMS'
CASE M_STOCK_CD = 'L65'
M_ITEM4 = 'LIGHTS'
CASE M_STOCK_CD = 'F70'
M_ITEM4 = 'FOOTRESTS'
CASE M_STOCK_CD = 'F80'
M_ITEM4 = 'TANKS'
ENDCASE
DO  CASE
CASE M_SPARTC = '001'
M_ITEM1 = '428 CHAINS'
CASE M_SPARTC = '002'
M_ITEM1 = '420 CHAINS'
CASE M_SPARTC = '003'
M_ITEM1 = '520 CHAINS'
CASE M_SPARTC = '027'
M_ITEM1 = 'IGNITION KEYS'
CASE M_SPARTC = '033'
M_ITEM1 =  'CLUTCH LIVERS'
CASE M_SPARTC = '117'
M_ITEM1 = 'M8/60'
CASE M_SPARTC = '089'
M_ITEM1 = 'SUPERGLUE GUMS'
CASE M_SPARTC = '111'
M_ITEM1 = 'HEAD LIGHTS'
CASE M_SPARTC = '212'
M_ITEM1 = 'FRONT FOOTREST'
CASE M_SPARTC = '215'
M_ITEM1 = 'FUEL TANKS'
ENDCASE
DO CASE
CASE M_ITMAC = '01'
M_ITEM2 = 'GOOSE'
CASE M_ITMAC = '02'
M_ITEM2 = 'DL'
CASE M_ITMAC = '03'
M_ITEM2 = 'JAPANAT'
CASE M_ITMAC = '04'
M_ITEM2 = 'TKR'
CASE M_ITMAC = '05'
M_ITEM2 = 'JAPAN'
CASE M_ITMAC = '06'
M_ITEM2 = 'UNION'
CASE M_ITMAC ='07'
M_ITEM2 = 'CTI'
CASE M_ITMAC = '08'
M_ITEM2 = 'HWA KONG'
CASE M_ITMA = '09'
M_ITEM2 = 'INNOSON'
CASE M_ITMAC = '10'
M_ITEM2 = 'HAFUN'
CASE M_ITMAC = '11'
M_ITEM2 = 'SUPER 10'
CASE M_ITMAC = '12'
M_ITEM2 = 'TKC'
```

```
ENDCASE
DO CASE
CASE M_KMACHC = '1'
M_ITEM3 = 'SUZUKI 100'
CASE M_KMACHC = '2'
M_ITEM3 = 'SUZUKI 120'
CASE M_KMACHC = '3'
M_ITEM3 = 'YAMAHA 100'
CASE M_KMACHC = '4'
M_ITEM3 = 'YAMAHA 125'
CASE M_KMACHC = '5'
M_ITEM3 = 'SUZUKI 125'
CASE M_KMACHC = '6'
M_ITEM3 = 'VESPA'
CASE M_KMACHC = '7'
M_ITEM3 = 'TORIS'
CASE M_KMACHC = '8'
M_ITEM3 = 'KAWASACKY 100'
CASE M_KMACHC = '9'
M_ITEM3 = 'SUPERZACKY'
ENDCASE
RP = " "
STORE 0 TO MTOT_AMT1,MTOT_AMT2,MTOT_AMT3,MTOT_AMT4,MTOT_AMT5,MTOT_AMT6
STORE 0 TO MTOT_AMT7,MTOT_AMT8,MTOT_AMT9,MTOT_AMT10,MTOT_AMTA,MTOT_AMTB
STORE 0 TO MTOT_AMTC,M_SAMT1,M_SAMT2,M_SAMT3,M_SAMT4,M_SAMT5,M_SAMT6
STORE 0 TO M_SAMT7,M_SAMT8,M_SAMT9,M_SAMT10,M_SAMTA,M_SAMTB,M_SAMTC
STORE 0 TO M_GD_TOTAL,M_GTAMT
@18,10
WAIT "IS ITEM CODE = ITEM1+ITEM2+ITEM3 (Y/N/W):" TO  RP
IF UPPER(RP) = 'Y'
DO CASE
CASE M_STOCK_CD = 'C10'
IF M_SPARTC = .001 .AND. M_ITMAC = 01 .AND. M_KMACHC = 1
ITEM1 = '428 CHAIN, GOOSE, SUZUKI 100'
@18,10 SAY "IS THE FNAME PURCHASE  (Y/N)?:" GET YN PICT "Y"
READ
IF UPPER(RP) = 'Y'
USE PURCHASE.DBF
MTOT_AMT1 = M_IQTY * M_IRATE
ELSE
USE SALES.DBF
M_SAMT1 = M_IQTYS *  M_SRATE
ENDIF
ENDIF
CASE M_STOCK_CD = 'C20'
IF M_SPARTC = 002 .AND. M_ITMAC = 02 .AND. M_KMACHC = 2
ITEM2 = '420 CHAIN, DL, SUZUKI 120'
@18,10 SAY "IS THE FNAME PURCHASE (Y/N)?:" GET YN PICT "Y"
READ
IF UPPER(YN) ='Y'
USE PURCHASE.DBF
MTOT_AMT2 = M_IQTY * M_IRATE
ELSE
USE SALES.DBF
M_SAMT2 = M_IQTYS * M_SRATE
ENDIF
ENDIF
CASE M_STOCK_CD = 'C30'
IF M_SPARTC = 003 .AND. M_ITMAC = 03 .AND. M_KMACHC = 3
ITEM3 = '520 CHAIN, JAPANAT, YAMAHA 100'
@18,10 SAY "IS THE FNAME  PURCHASE (Y/N)?:" GET YN PICT "Y"
READ
IF UPPER(YN) = 'Y'
```

```
USE PURCHASE.DBF
MTOT_AMT3 = M_IQTY * M_IRATE
ELSE
USE SALES.DBF
M_SAMT3 = M_IQTYS * M_SRATE
ENDIF
ENDIF
CASE M_STOCK_CD = 'K40'
IF M_SPARTC = 027  .AND.  M_ITMAC = 04  .AND. M_KAMCHC = 4
ITEM4 = 'IGNITION KEY, TKR, YAMAHA 125'
@18,10 SAY "IS THE FNAME PURCHASE (Y/N)?:" GET YN
READ
IF UPPER(YN) = 'Y'
USE PURCHASE.DBF
MTOT_AMT4 = M_IQTY * M_IRATE
ELSE
USE SALES.DBF
M_SAMT4 = M_IQTYS * M_SRATE
ENDIF
ENDIF
CASE M_STOCK_CD = 'L45'
IF M_SPARTC =  033 .AND. M_ITMAC = 05  .AND. M_KMACHC = 5
ITEM5 = 'CLUTCH LIVER, JAPAN, SUZUKI 125'
@18,10 SAY "IS THE FNAME PURCHASE (Y/N)?:" GET YN PICT "Y"
READ
IF UPPER(YN) = 'Y'
USE PURCHASE.DBF
MTOT_AMT5 = M_IQTY * M_IRATE
ELSE
USE SALES.DBF
M_SAMT5 = M_IQTYS * M_SRATE
ENDIF
ENDIF
@18,10
WAIT "IS THE FNAME PURCHASE (Y/N)?:" TO YN
IF UPPER(YN) = 'Y'
MTOT_AMTA = MTOT_AMT1 + MTOT_AMT2 +  MTOT_AMT3 + MTOT_AMT4 + MTOT_AMT5
ELSE
M_SAMTA = M_SAMT1 + M_SAMT2 + M_SAMT3 + M_SAMT4 + M_SAMT5
ENDIF
OTHERWISE
EXIT
ENDCASE
ENDIF
IF UPPER(RP) = 'N'
DO CASE
CASE M_STOCK_CD = 'B50'
IF M_SPARTC = 117 .AND. M_ITMAC = " " .AND. M_KMACHC = " "
ITEM6 = 'M8/60'
@18,10 SAY "IS THE FNAME PURCHASE(Y/N)?:" GET YN PICT "Y"
READ
IF UPPER(YN) =  'Y'
USE PURCHASE.DBF
MTOT_AMT6 = M_IQTY  * M_IRATE
ELSE
USE SALES.DBF
M_SAMT6  = M_IQTYS * M_SRATE
ENDIF
ENDIF
CASE  M_STOCK_CD  = 'G60'
IF M_SPARTC = 089 .AND. M_ITMAC = " " .AND. M_KMACHC =  " "
ITEM7 =  'SUPERGLUE GUM'
@18,10 SAY "IS THE FNAME PURCHASE(Y/N)?:" GET YN PICT "Y"
```

```
READ
IF UPPER(YN) = 'Y'
USE PURCHASE.DBF
MTOT_AMT7  =  M_IQTY * M_IRATE
ELSE
USE SALES.DBF
M_SAMT7  = M_IQTYS * M_SRATE
ENDIF
ENDIF
@18,10 SAY "IS THE FNAME PURCHASE(Y/N)?:" GET YN PICT "Y"
READ
IF UPPER(YN) =  'Y'
MTOT_AMTB = MTOT_AMT6 +  MTOT_AMT7
ELSE
M_SAMTB  =  M_SAMT6  + M_SAMT7
ENDIF
ENDCASE
ENDIF
IF  UPPER(RP) = 'W'
DO  CASE
CASE M_STOCK_CD = 'L65'
IF M_SPARTC = 111 .AND. M_ITMAC = " " .AND. M_KMACHC = 6
ITEM8 = 'HEAD LIGHT, VESPA'
@18,10 SAY "IS THE FNAME  PURCHASE(Y/N)?:" GET YN PICT "Y"
READ
IF UPPER(YN) = 'Y'
USE PURCHASE.DBF
MTOT_AMT8  =  M_IQTY * M_IRATE
ELSE
M_SAMT8 = M_IQTYS * M_SRATE
ENDIF
ENDIF
CASE M_STOCK_CD = 'F70'
IF M_SPARTC = 212 .AND. M_ITMAC = " " .AND. M_KMACHC = 7
ITEM9 = 'FRONT FOOTREST, TORIS'
@18,10 SAY "IS THE FNAME PURCHASE(Y/N)?:" GET YN
READ
IF UPPER(YN)  = 'Y'
USE PURCHASE.DBF
MTOT_AMT9 =  M_IQTY * M_IRATE
ELSE
USE SALES.DBF
M_SAMT9  = M_IQTYS * M_SRATE
ENDIF
ENDIF
CASE M_STOCK_CD = 'F80'
IF M_SPARTC = 215 .AND. M_ITMAC =  " " .AND. M_KMACHC = 8
ITEM10 = 'FUEL TANK, KAWASACKY 100'
@18,10 SAY "IS THE FNAME PURCHASE(Y/N)?:" GET YN
READ
IF UPPER(YN) =  'Y'
USE PURCHASE.DBF
MTOT_AMT10 =  M_IQTY * M_IRATE
ELSE
USE  SALES.DBF
M_SAMT10  = M_IQTYS * M_SRATE
ENDIF
ENDIF
@18,10
WAIT "IS THE FILE PURCHASE(Y/N)?:"  TO YN
IF UPPER(YN)  = 'Y'
MTOT_AMTC = MTOT_AMT8 +  MTOT_AMT9 + MTOT_AMT10
ELSE
```

```
M_SAMTC = M_SAMT8 + M_SAMT9 + M_SAMT10
ENDIF
ENDCASE
ENDIF
AS =  SPACE(1)
@17,5 CLEA TO 20,70
@20,10 SAY "IS THE FNAME PURCHASE(Y/N)?:" GET AS PICT "Y"
READ
IF AS = 'Y'
M_GD_TOTAL =  MTOT_AMTA + MTOT_AMTB  + MTOT_AMTC
@11,5 CLEA TO 22,70
@12,15 SAY M_INAME PICT "@!"
@13,15 SAY M_IQTY PICT "*****"
@14,15 SAY M_IRATE(M_IRATE,6,2)
@15,15 SAY MTOT_AMTA + SPACE(2) + MTOT_AMTB + SPACE(2) + MTOT_AMTC
ELSE
M_GTAMT = M_SAMTA + M_SAMTB + M_SAMTC
@13,5 CLEA TO 20,72
@14,10 SAY M_INAME
@15,10 SAY M_IQTYS PICT "*****"
@16,10 SAY M_SRATE(M_SRATE,6,2)
@17,10 SAY M_SAMTA + SPACE(2) + M_SAMTB + SPACE(2) + M_SAMTC
ENDIF
ENDDO
SET TALK ON
RETURN


* PROCEDURE  M_PURCHASE
*SUBPROGRAM FOR PURCHASE RECORD MODIFICATION
CLEA
SET DATE BRITISH
SET TALK OFF
@4,5 CLEA TO 18,78
@4,5 TO 18,78
SET ECHO OFF
SET COLOR TO R/B+,G/B+
STORE SPACE(3) TO M_FILE_NO,M_STOCK_CD,M_SCODE
STORE SPACE(10) TO M_ITEM2,M_ITEM3,M_ITEM4
STORE SPACE(15) TO M_ITEM1
STORE SPACE(20) TO M_SNAME,M_INAME
M_SADD = SPACE(25)
M_ICODE = SPACE(9)
M_PDATE = CTOD(TODAY)
RP = SPACE(1)
STORE 0 TO M_INVNO,M_SPARTC,M_ITMAC,M_IQTY,M_KMACHC,M_IRATE,MTOT_AMT
STORE 0 TO M_GD_TOTAL
@2,5 CLEA TO 20,70
DO WHILE .NOT. EOF()
@9,10 TO 13,74
USE PURCHASE INDEX ITEM_CODE
@11,20 SAY "ENTER ITEM CODE:" GET M_ICODE PICT "@!"
READ
SET ECHO OFF
SET COLOR TO  G/B+
LOCATE FOR ITEM_CODE = M_ICODE
IF .NOT. FOUND()
CLEA
AN = SPACE(1)
@5,10 SAY  "RECORD NOT FOUND -----------------!!!"
@11,10 SAY "WRONG ITEM CODE ENTERED ----------!!!"
@17,10
WAIT "PRESS ANY KEY TO CONTINUE ------------!!!"
```

```
@20,25 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET AN PICT "Y"
READ
IF AN = "Y"
CLEA
RETURN
ELSE
CLEA
CLOSE ALL
EXIT
ENDIF
ENDIF
IF FOUND()
CLEA
SET COLOR TO RB
SET ECHO  OFF
@13,5 TO 15,72
@14,25 SAY "TO VIEW THESE RECORDS (Y/N)?:"  GET RP PICT "Y"
READ
IF  UPPER(RP) =  "Y"
CLEA
LIST ALL
Q = INKEY(0.1)
CLEA
SET FORMAT TO PURCHASE
READ
ENDIF
IF UPPER(RP) =  "N"
CLEA
SET FORMAT TO PURCHASE
READ
ENDIF
ENDIF
@14,20 SAY "INPUT CURRENT  RECORDS  ---- PRESS ENTER KEY!"
SET ECHO OFF
STORE "  " TO YN
CLEA
@21,20
WAIT "TO MODIFY MORE RECORDS (Y/N)?:" TO YN
IF UPPER(YN) $ "Y"
CLEA
LOOP
ELSE
CLEA
REPLA  PUR_DATE WITH M_PDATE, ITEM_QTY WITH M_IQTY
REPLA ITEM_NAME WITH M_INAME, ITEM_RATE  WITH M_IRATE
REPLA TOTAL_AMT WITH MTOT_AMT, GTOT_AMT WITH M_GD_TOTAL
REPLA FILE_NUMB WITH M_FILE_NO, S_NAME WITH M_SNAME
REPLA S_ADD WITH M_SADD, INV_NO WITH M_INVNO
REPLA STOCK_CD WITH M_STOCK_CD, CUST_CODE WITH M_CUS_CODE
REPLA STOCK_INM WITH M_ITEM4
CLOSE ALL
EXIT
ENDIF
ENDDO
SET TALK ON
SET CLOCK OFF
RETURN

**** RECDEL.PRG ****
*THIS PROGRAM IS A SUBPROGRAM TO RECORDS FROM THE MAIN PROGRAM.
CLEA
SET COLOR TO W+/B+
SET CLOCK ON
```

```
CLEA
SET TALK OFF
SET ECHO OFF
@9,2 CLEA TO 23,70
*@9,2 TO 20,70 DOUBL COLOR N+
SET COLOR TO W+/RB
FNAME = SPACE(6)
SET DATE BRITISH
@10,20 SAY "DATA DELETE ROUTINE"
*@11,20 TO 11,38 DOUBL
Q = INKEY(0.4)
IF  Q = INKEY(0.4)
@21,27 SAY "PRESS A KEY TO CONTINUE -----------!!!"
CLEA
SKIP
ENDIF
DO WHILE .T..
@8,5 CLEA TO 19,70
*SET THE FILES IN A LOOP FOR NECESSARY OPERATION.
SET ECHO OFF
@8,10 TO 19,70 DOUBL
SET COLOR TO GR+/B
@8,15 SAY "1" + SPACE(5) + "PURCHASE"
@9,15 SAY "2" + SPACE(5) + "SALES"
@10,15 SAY "3" + SPACE(5) + "DEBTS"
@11,15 SAY "4" + SPACE(5) + "CREDIT"
@12,15 SAY "5" + SPACE(5) + "ITEM"
@13,15 SAY "6" + SPACE(5) + "EXPENSES"
@14,15 SAY "7" + SPACE(5) + "INCOME"
@15,15 SAY "8" + SPACE(5) + "INVENTORY"
@16,15 SAY "9" + SPACE(5) + "CUSTOMER"
@17,15 SAY "0" + SPACE(5) + "BANK"
@18,5 CLEA TO 20,70
@20,5 TO 22,70 DOUBL
@20,25 SAY "ENTER FILENAME:" GET FNAME PICT "9"
READ
*TO ANALYSE INPUT FOR THE FILENAME
DO CASE
CASE FNAME = "1"
USE PURCHASE.DBF
LOCATE FOR S_NO = M_SNO
CASE FNAME = "2"
USE SALES.DBF
LOCATE FOR S_NO = M_SNO
CASE FNAME = "3"
USE DEBTS.DBF
LOCATE FOR S_CODE = M_SCODE
CASE FNAME = "4"
USE CREDIT.DBF
LOCATE FOR CUST_CODE = M_CCODE
CASE FNAME = "5"
USE ITEM.DBF
LOCATE  FOR S_NO = M_SNO
CASE FNAME = "6"
USE EXPENSES.DBF
LOCATE FOR S_NO = M_SNO
CASE FNAME = "7"
USE INCOME.DBF
LOCATE FOR S_NO = M_SNO
CASE FNAME = "8"
USE STOCK.DBF
LOCATE FOR STOCK_CD =  M_STCODE
CASE FNAME = "9"
```

```
USE CUSTOMER.DBF
LOCATE FOR CUST_CODE = M_CCODE
CASE FNAME = "0"
USE BANK.DBF
LOCATE FOR S_NO = M_SNO
OTHERWISE
CLEA
@13,10 CLEA TO 15,78
SET COLOR TO RB
@13,10 TO 15,75
CH = SPACE(1)
@14,15 SAY "WRONG FILENAME ---------------- TRY AGAIN !!!"
@17,20
WAIT "PRESS ANY KEY TO CONTINUE -----------------------!!!"
CLEA
EXIT
ENDCASE
IF FOUND()
CLEA
@15,15 SAY "TO VIEW RECORD BEFORE DELETING(Y/N)?:" GET CH PICT "!"
READ
IF UPPER(CH)  = "Y"
SET ECHO OFF
SET COLOR TO R
AN = SPACE(1)
CLEA
@2,3
LIST ALL
Q = INKEY(0.2)
@20,15
WAIT "ARE THESE THE RECORDS YOU WANT TO DELETE(Y/N)?:" TO  AN
IF UPPER(AN) = 'Y'
DELETE
PACK
ENDIF
IF UPPER(AN) = 'N'
CLEA
LOOP
ENDIF
ELSE
DELETE
ZAP
ENDIF
RP = "!"
@10,2 CLEA TO 20,75
@12,5 CLEA TO 16,72
@10,2 TO 20,75 DOUBL
@12,5 TO 16,72 DOUBL
SET COLOR TO  R+/G*
SET ECHO OFF
@14,10 SAY "TO DELETE MORE RECORDS (Y/N)?:"
SET COLOR TO W+/B+,R+
@13,45 GET RP PICT "Y"
READ
*USER INPUT ANALYSIS
IF UPPER(RP) = "Y"
CLEA
LOOP
ENDI
IF UPPER(RP) = "N"
CLEA
EXIT
ENDIF
```

```
ENDIF
IF .NOT. FOUND()
CLEA
DC = SPACE(1)
@11,5 TO 16,70 DOUBL
@12,15 SAY "WRONG FILE NAME ENTERED -----------------!!!"
@13,15 SAY "RECORD NOT FOUND ----------------------!!!"
@14,15 SAY "PLEASE ENTER CORRECT FILE NAME -------------!!!"
@15,25 SAY "DO YOU WISH TO CONTINUE (Y/N)?:" GET DC
READ
IF DC = "Y"
WAIT "PRESS A KEY TO CONTINUE  WORK ----------------------------------!!!"
LOOP
ENDIF
IF DC = "N"
CLOSE DATABASE
EXIT
ENDIF
ENDIF
ENDDO
ENDDO
SET CLOCK OFF
SET TALK ON
RETURN

*PROCEDURE M_EXPENSES
*MODIFICATION OF RECORED INTO EXPENSES FILE VIA EXPENSES SUBPROGRAM.
CLEA
SET TALK OFF
SET ECHO OFF
@10,1 CLEA TO  20,70
SET DATE BRITISH
SET CLOCK ON
SET COLOR TO   GR/RB+
RP  =    SPACE(1)
M_DT_EXPEN  = CTOD(TODAY)
STORE 0 TO  MLCSTOCK,M_ETBILL,M_WTBILL,M_RRENT,M_PREMIS,M_LANDB
STORE 0 TO M_MISCELL,M_PPAY,M_LDINWD,MCOUTWD,MLDRAWS,M_BDEBT
DO WHILE .T.
GO TOP
USE EXPENSES.DBF
@4,6 CLEA TO 18,77
@14,15 SAY "ENTER DATE OF EXPENSES:" GET M_DT_EXPEN PICT "(DD/MM/YY)"
READ
LOCATE FOR EXP_DATE =  M_DT_EXPEN
IF.NOT. FOUND()
CLEA
RS = SPACE(1)
@5,20 SAY "WRONG DATE OF EXPENSES ENTERED -------------------------!!!"
@11,20 SAY "RECORD NOT IN THE DBASE FILE ----------------------------!!!"
@17,10
WAIT "OK! ------ PRESS ANY KEY TO CONTINUE!!!"
@20,25 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET RS PICT "Y"
READ
IF RS = "Y"
CLEA
LOOP
ELSE
CLEA
CLOSE ALL
EXIT
ENDIF
ENDIF
```

```
IF  FOUND()
@14,15 SAY "TO VIEW RECORD(Y/N)?:" GET RP PICT "Y"
READ
IF UPPER(RP) ='Y'
@4,6 CLEA TO  18,77
LIST ALL
CLEA
SET FORMAT TO EXPENSES
READ
ENDIF
IF UPPER(RP) = 'N'
CLEA
SET FORMAT TO EXPENSES
READ
ENDIF
SET ECHO OFF
@20,25
WAIT "TO MODIFY MORE RECORD(Y/N)?:" TO RS
IF UPPER(RS) = 'Y'
CLEA
LOOP
ELSE
CLEA
*REPLACE COMMAND WILL BE USE TO REPLACE CONTENTS INTO DATABASE FILE
REPLA DT_EXPENS WITH M_DT_EXPEN, GTOT_EXPEN WITH M_GTAMT_E
REPLA LC_STOCK WITH MLCSTOCK, ELECBILL  WITH  M_ETBILL
REPLA WATERBILL WITH M_WTBILL, RATERENT WITH M_RRENT
REPLA PREMISES WITH M_PREMIS, LANDBD WITH M_LANDB
REPLA  MISCELL WITH M_MISCELL, PPAYMENT WITH M_PPAY
REPLA C_OUTWD WITH MCOUTWD, LD_INWD  WITH M_LDINWD
REPLA BADDEBT WITH M_BDEBT, L_DRAWS WITH M_LDRAWS
REPLA S_NO WITH M_SNO
CLEA
CLOSE ALL
EXIT
ENDIF
ENDIF
ENDDO
SET TALK ON
SET CLOCK OFF
RETURN

*PROCEDURE M_INVENTORY
*TO MODIFY THE ITEM IN THE STOCK FILE
CLEA
SET ECHO OFF
SET CLOCK ON
SET TALK OFF
@2,5 CLEA TO 20,70
SET COLOR TO G
SET DATE BRITISH
@3,5 TO 20,70 DOUBL
RP = SPACE(1)
*INITIALIZATION OF MEMORY VARIABLES
M_ICODE = SPACE(9)
M_INAME = SPACE(25)
M_STOCK_CD = SPACE(3)
DS = SPACE(3)
STORE 0 TO M_IQTYS,MQISTOCK,M_SRATE,M_TOTAL1,M_TOTAL2,M_SAMT
MLUPDATE = CTOD("TODAY")
@2,5 CLEA TO 15,78
DO WHILE .T.
GO TOP
```

```
CLEA
@4,10 TO 20,70 DOUBL
USE STOCK.DBF
@15,20 SAY "ENTER ITEM STOCK CODE:" GET M_STOCK_CD PICT "@!"
READ
*LOCATE THIS ITEM IN THE STOCK FILE
LOCATE FOR STOCK_CD = M_STOCK_CD
IF .NOT. FOUND()
CLEA
RA = " "
@11,2 TO 16,70
SET COLOR TO B
@12,15 SAY "WRONG ITEM STOCK CODE "+OPT+" ENTERED ------------------!!!"
@14,15 SAY "RECORD NOT FOUND --- ENTER CORRECT STOCK CODE ----------!!!"
@15,10
WAIT "PLEASE! PRESS ANY KEY TO CONTINUE ----------------------------!!!"
@20,20 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET RA PICT "Y"
READ
IF RA = "Y"
CLEA
RETURN
ENDIF
IF RA = "N"
CLEA
CLOSE ALL
EXIT
ENDIF
ENDIF
IF FOUND()
CLEA
@88,5 TO 18,78  DOUBL
SET COLOR TO B+/W
A = RECNO()
A = A + 7
@8,5 TO 18,78 DOUBL
@10,7 SAY "STOCK_CD"+S + "ITEM_NM" + S + "QTY_ISTOCK" + S + "ITEM_RATE"
@10,50 SAY "AMT_ISTOCK"
@11,7 SAY  STOCK_CD + S + ITEM_NM + S + QTY_ISTOCK + S + ITEM_RATE
@11,50 SAY AMT_ISTOCK
READ
@16,15  CLEA
@16,10 TO 17,75 PANEL
ENDIF
USE SALES.DBF
CLEA
@10,10 TO 14,70 DOUBL
@12,25 SAY "ENTER ITEM STOCK CODE:" GET M_STOCK_CD PICT "@!"
READ
LOCATE FOR STOCK_CD = M_STOCK_CD
IF FOUND()
CLEA
SET COLOR TO G+
@14,5 TO 18,78  DOUBL
@16,7 SAY "ITEM_QTY"+S+"ITEM_NAME"+S +"SELLING_RATE" + S + "TOTAL_AMT"
@17,7 SAY ITEM_QTY + S + ITEM_NAME + S + SELL_RATE + S + TOTAL_AMT
Q = 0
IF Q = INKEY(0.1)
CLEA
SKIP
ENDIF
ENDIF
*UPDATING THE STOCK FILE BY  ADDITION.
MQOHAND = QTY_ISTOCK - I_QTY_SD
```

```
M_TOTAL2 = AMT_ISTOCK - TOTAL_AMT
MQISTOCK = MQOHAND
@21,5 CLEA TO 24,78
@21,10 SAY MQISTOCK PICT "9999999999.99"
@22,10 SAY M_TOTAL2 PICT "9999999999.99"
@23,10 SAY M_STCODE
@20,5 TO 24,78 PANEL
@13,5 TO 17,70 DOUBL
@15,15
WAIT "TO CONTINUE STOCK UPDATE(Y/N)?:" TO RP
IF UPPER(RP) =  "Y"
CLEA
LOOP
ELSE
USE STOCK.DBF
SET FORMAT TO STOCK
READ
REPLA AMT_ISTOCK WITH M_TOTAL2, STOCK_CD WITH M_STCODE
REPLA ITEM_NAME WITH M_INAME, Q1_STOCK WITH MQISTOCK
CLOSE DATABASE
EXIT
ENDIF
ENDDO
SET TALK ON
SET CLOCK  OFF
RETURN

* PROCEDURE MOD_BANK
SET CLOCK ON
SET ECHO OFF
SET DATE BRITISH
CLEA
@2,15 CLEA TO 15,70
SET COLOR TO B+
YN = SPACE(1)
STORE "CTOD(TODAY)" TO M_DATE_DP,M_DATE_CQ
STORE SPACE(20) TO M_BK_NM,M_PAYER,M_BKN_OCQ
STORE SPACE(25) TO M_BK_ADD,M_N_ISSUE
STORE 0 TO M_BK_CODE,M_BK_ACNM,M_AMT_IBK,M_LODGE,M_CHEQUE,MGT_AMT
STORE 0 TO M_CHHAND,M_CHBANK,M_CD50,M_CD20,M_CD10,M_CD5,M_CDC1
STORE 0 TO M_CDC_50,M_CDC_10,M_CDC_5,M_CDC_1,AMT1,AMT2,AMT3,AMT4
STORE 0 TO AMT5,AMT6,AMT7,AMT8,AMT9,M_SNO
@3,5 CLEA TO 22,75
SET ECHO OFF
DO WHILE .T.
GO TOP
CLEA
@2,5 TO 22,75 DOUBL
USE BANK.DBF
@14,20 SAY "ENTER BANK CODE:" GET M_SNO PICT "999"
READ
SET ECHO OFF
LOCATE FOR BANK_CODE = M_BK_CODE
IF .NOT. FOUND()
CLEA
RS = SPACE(1)
@11,3 TO 16,75
@12,10 SAY " WRONG BANK CODE ENTERED -------------!!!"
@13,10 SAY "RECORD NOT FOUND ----------------------!!!"
@14,10 SAY "PLEASE ENTER CORRECT BANK CODE -------!!!"
@15,20
WAIT "PRESS ANY KEY TO RESUME WORK ---------------!!!"
@20,20 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET RS PICT "Y"
```

```
READ
IF RS = "Y"
CLEA
LOOP
ENDIF
IF RS = "N"
CLEA
CLOSE DATA
EXIT
ENDIF
ENDIF
IF FOUND()
SET ECHO OFF
SET COLOR TO W+/G
@14,15 SAY "TO  VIEW RECORDS(Y/N)?:" GET YN PICT "Y"
READ
IF UPPER(YN) = "Y"
@4,6 CLEA TO 17,77
@4,6 TO 17,76 DOUBL
LIST ALL
CLEA
SET FORMAT TO BANK
READ
ENDIF
IF UPPER(YN) = "N"
SET FORMAT TO BANK
READ
ENDIF
STORE " " TO RP
SET ECHO  OFF
CLEA
@12,10 TO 16,75 DOUBL
@14,15 SAY "TO MODIFY MORE RECORDS(Y/N)?:" GET RP
READ
IF UPPER(RP) = "Y"
CLEA
LOOP
ENDIF
IF UPPER(RP) = "N"
CLEA
REPLA DT_DEPOSIT WITH M_DATE_DP, DT_OCHEQUE  WITH M_DATE_OCQ
REPLA BANK_NM WITH M_BK_NM, PAYER_NM WITH M_PAYERNM
REPLA BANKN_OCQ WITH M_BKN_OCQ, BANK_ADD  WITH M_BK_ADD
REPLA CUST_NAME WITH M_CUST_NM, BANK_CODE WITH M_BK_CODE
REPLA BANK_ACNO WITH M_BK_ACNO, CASHBANK WITH M_CHBANK
REPLA LODGEMENT WITH M_LODGE, CHEQUE_AMT WITH M_CQ_AMT
REPLA CQ_NUMBER WITH M_CQ_NUMB, GTOT_AMT  WITH MGT_AMT
REPLA CASHHAND WITH M_CHHAND, CASHN_50 WITH M_CD50
REPLA CASHN_20 WITH M_CD10, CASHN_5 WITH M_CD5
REPLA COINS1 WITH M_CDC1, COINS_50 WITH M_CDC50
REPLA COINS_10 WITH M_CDC10, COINS_5 WITH M_CDC5
REPLA COINS_1 WITH M_CDC_1, S_NO WITH M_SNO
CLOSE ALL
EXIT
ENDIF
ENDIF
SET TALK ON
SET CLOCK OFF
ENDDO
RETURN
```

```
****PASS_D.PRG ****
* THIS PROGRAM IS CALLED PASS_D PROGRAM INFORMATION ACCESSIBLE
        CLEA
@2,5 CLEA TO 15,78
SET CLOCK ON
SET ECHO OFF
@2,3 CLEA TO 15,70
SET COLOR TO G+
MPASS_D =  "CWSIAAP"
MPASS_D = SPACE(7)
ROW = 0
DO WHILE .NOT. EOF()
        CLEA
        @2,5 CLEA TO 17,78
        SET COLOR TO W/G+
        SET ECHO OFF
        @2,5 TO 17,78 DOUBL COLOR B
        @14,20 SAY "MAKE SURE YOU ENTER CORRECT NAME ---!!!"
        @16,20 SAY "LET YOUR PASWD BE THE CORRECT ONE --!!!"
        @10,5 CLEA TO 16,78
        ROW = ROW + 1
        @20,25
        WAIT "PLEASE! PRESS ANY KEY TO CONTINUE -----------!!!"
        @10,5 CLEA TO 16,78
        @10,6 FILL TO 16,78 COLOR B
        @13,20 SAY "ENTER PASSW_D OR 'XX' TO QUIT:" GET MPASS_D
        READ
        SET COLOR TO N/N
       * TO ANALYSE THE INPUTS ---------------------------!!!"
                 IF UPPER(MPASS_D) = "CWSIAAP"
                         SET COLOR TO BG/RB+
                         DO LOGO
                 ENDIF
                IF UPPER(MPASS_D) = "XX"
                         CLEA
                         EXIT
              ENDIF
        SET COLOR TO W/B
               IF UPPER (MPASS_D) <> "CWSIAAP"
               @12,5 CLEA TO 16,74
               @12,5 TO 16,74 DOUBL COLOR W
               SET COLOR TO GR/B+
               @13,15 SAY "WRONG PASWD ENTERED ---!!!"
               @14,15 SAY "PUT CORRECT PASWD & NAME --!!!"
               @15,15 SAY "NO INFORMATION ACCESS -----!!!"
               WAIT "PRESS ANY KEY TO CONTINUE ----------!!!"
                           CLEA
                         J_CHIDI = ".F."
                         SET COLOR TO N
                         EXIT
                 ENDIF
            ENDDO
SET STAT ON
SET SCORE ON
SET CLOCK OFF
RETURN
```

```
**** ANAL_REP ****
* BUSINESS. TRANSACTIONAL ASSESSEMENTAL ANALYSIS
SET TALK OFF
SET COLOR TO RG+/B
CLEA
SET ECHO OFF
SET DATE BRITISH
@2,2 CLEA TO 23,78
@4,5 CLEA TO 20,70
@2,5 TO 23,78 DOUBL
MG = "ANALYSIS_REPORT ON BUSINESS TRANSACTIONS --------"
@4,20 TO 4,68
K = 1
DC = SPACE(1)
*INITIALIZATION OF MEMORY  VARIABLES.
STORE 0 TO M_OSTOCK,MCKNWD,MCG_AVAL,MLCSTOCK,MCG_SOLD,M_SALES
STORE 0 TO M_LDINWD,M_TOTAL1,MG_PROF,M_RRENT,M_WTBILL,M_ETBILL
STORE 0 TO MCOUTWD,M_BDEBT,M_TOTAL2,M_NTPROF,MLDRRAWS,M_TOTAL3
STORE 0 TO M_TOTAL4,M_ACCRU,M_TOTAL5,M_PREMIS,M_LANDB,M_MISCELL
STORE 0 TO M_TOTAL6,M_AMTST,M_DEBT,M_CHBANK,M_PPAY,M_CHHAND
STORE 0 TO M_TOTAL7,M_TOTAL8,M_CAPACC,M_BALCD,M_TOTAL9,M_CDAMT
CLEA
DO WHILE .T.
@2,5 TO 6,70 DOUBL
@4,10 SAY MG
Q = 0
IF Q = INKEY(0.6)
CLEA
SKIP
ENDIF
*POSITIONING OF DIFFERENT FILES IN THRIR RESPECTIVE PLACES
M_ARDATE = CTOD(TODAY)
DO WHILE .T.
GO TOP
CLEA
@1,5 CLEA TO 24,78
@1,5 TO 24,78  PANEL COLOR W
@3,25 SAY "TRADE ACCOUNT AS AT --------"
@3,60 GET M_ARDATE PICT "(DD/MM/YY)"
READ
@4,25 TO 4,53
@5,12 SAY "PURCHASE ACCOUNT"
@6,12 SAY "----------------"
@5,50 SAY "SALES ACCOUNT"
@6,50 SAY "-------------"
@7,2 SAY "OPENING STOCK:="
@8,2 SAY "PURCHASES:="
@9,2 SAY "CARRIAGE INWARDS:="
@10,2 TO 10,38
@11,2 SAY "COST OF GOODS AVAIL.:="
@12,2 SAY "LESS CLOSING STOCK:="
@13,2 TO 13,38
@14,2 SAY "COST OF GOODS SOLD:="
@7,42 SAY "AMOUNT OF SALES:="
@8,42 SAY "LESS DELIVERY INWARDS:="
@15,2 TO 15,38
@16,2 SAY "GROSS PROFIT:="
@17,42 SAY "SALES TOTAL:="
@17,2 SAY "BALANCE  C/D:="
DO  WHILE .T.
SELECT 1
USE STOCK.DBF
SELECT 2
```

```
USE PURCHASE.DBF
SELECT 3
USE EXPENSES.DBF
SELECT 4
USE SALES.DBF
SELECT 5
USE CREDIT.DBF
SELECT 6
USE DEBTS.DBF
SELECT 7
USE INCOME.DBF
SELECT 8
USE BANK.DBF
M_REC = RECNO()
B = M_REC
B = B + 6
*SELECT FILE AND RETRIEVE INFORMATION.
CLEA
@4,5 TO 20,78  DOUBL
SELECT 1
@5,6 SAY AMT_ISTOCK(AMT_ISTOCK,12,2)
@5,25 SAY OPEN_STOCK(OPEN_STOCK,7,2)
READ
SELECT 2
@6,6 SAY GTOT_AMT(GTOT_AMT,10,2)
@6,25 SAY C_INWARD(C_INWARD,6,2)
READ
SELECT 3
@7,6 SAY LC_STOCK(LC_STOCK,8,2)
@7,20 SAY LD_INWDS(LD_INWDS,6,20)
@7,35 SAY RATERENT(RATERENT,10,2)
@7,50 SAY WATERBILL(WATERBILL,6,2)
@7,60 SAY ELECBILL(ELECBILL,6,2)
@8,6 SAY C_OUTWD(C_OUTWD,6,2)
@8,20 SAY BADDEBT(BADDEBT,6,2)
@8,35 SAY L_DRAWS(L_DRAWS,6,2)
@8,50 SAY PREMIS(PREMIS,6,2)
@8,60 SAY LANDBD(LANDBD,8,2)
@9,6 SAY MISCELL(MISCELL,6,2)
@9,20 SAY PPAYMENT(PPAYMENT,8,2)
READ
SELECT 4
@10,6 SAY GT_AMT(GT_AMT,12,2)
READ
SELECT 5
@11,6 SAY ACCRUALS(ACCRUALS,6,2)
@11,25 SAY C_BAL(C_BAL,8,2)
READ
SELECT 6
@12,6 SAY D_BAL(D_BAL,8,2)
READ
SELECT 7
@13,6 SAY CASHAND(CASHHAND,12,2)
READ
SELECT 8
@14,6 SAY CASHBANK(CASHBANK,12,2)
READ
B = B + 1
*TRANSFER OF FIELD CONTENTS INTO MEMORY VARIABLES.
M_OSTOCK = OPEN_STOCK
M_AMTST = AMT_ISTOCK
M_CINWD = C_INWARD
MTOT_AMT = GTOT_AMT
```

```
MCG_AVAL = CG_AVAIL
MLCSTOCK = LC_STOCK
MCG_SOLD = CG_SOLD
M_SALES = GT_AMT
M_LDINWD = LD_INWDS
M_TOTAL1 = S_TOTAL
M_GPROF = G_PROFIT
M_RRENT = RATERENT
M_WTBILL = WATERBILL
M_ETBILL = ET_BILL
M_COUTWD = C_OUTWD
M_BDEBT =  BADDEBT
M_TOTAL2 = E_TOTAL
M_BALCD  = B_CDOWN
M_NTPROF = NT_PROFIT
MCDRAWS = L_DRAWS
M_TOTAL3 = FD_TOTAL
M_TOTAL4 = CP_TOTAL
M_ACCRU = ACCRUALS
M_CDAMT = C_BAL
M_TOTAL5 = CL_TOTAL
M_PREMIS =  PREMISES
M_LANDB = LANDBD
M_MISCELL = MISCELL
M_TOTAL6 = CU_TOTAL
M_DEBTS - DEB_BAL
M_CHBANK = CASHBANK
M_PPAY = PPAYMENT
M_CHHAND =  CASHHAND
M_TOTAL7 = BAL_CACC
M_CAPACC =  CAP_ACCT
M_TOTAL8 = B_LSIDE
M_TOTAL9 = B_RSIDE
*CALCULATION OF ANNUAL TRADE TRASACTIONS.
MCG_AVAL = M_OSTOCK + MTOT_AMT + M_CINWD
MCG_SOLD = MCG_AVAL - MLCSTOCK
M_TOTAL1 = M_SALES - M_LDINWD
M_GPROF = M_TOTAL1 - MCG_SOLD
M_TOTAL2 = M_RRENT + M_WTBILL + M_ETBILL + M_COUTWD + M_BDEBT
M_NTPROF = M_GPROF - M_TOTAL2
M_OSTOCK = M_CAPACC
M_TOTAL4 = M_CAPACC + M_NTPROF
M_TOTAL7 = M_TOTAL4 - M_LDRAWS
M_TOTAL3 = M_PREMIS + M_LANDB + M_MISCELL
M_TOTAL5 = M_CDAMT + M_ACCRU
M_A_STOCK = M_AMTST
M_TOTAL8 = M_TOTAL4 - M_TOTAL5
M_AMTST = MLCSTOCK
M_TOTAL6 = M_A_STOCK + M_DEBTS + M_CHBANK + M_PPAY + M_CHHAND
M_TOTAL9 = M_TOTAL3 + M_TOTAL6
M_BALCD = MCG_SOLD + M_GPROF
*DISPLAY THE REPORT OF THE CALCULATIONS.
A = RECNO()
A = A + 7
@A,19 SAY M_OSTOCK PICT "99999999.99"
@A,16 SAY MTOT_AMT PICT "99999999.99"
@A,23 SAY M_CKNWD PICT "9999.99"
@A,26 SAY MCG_AVAL PICT "999999.99"
@A,25 SAY MLCSTOCK PICT "99999999.99"
@A,62 SAY M_SALES PICT "9999999999.99"
@A,67 SAY M_LDINWD PICT "9999.99"
@A,18 SAY M_BALCD PICT "99999999.99"
@A,58 SAY M_TOTAL1 PICT "9999999.99"
```

```
@A,18 SAY M_GPROF PICT "99999999.99"
A = A + 1
R = R + 1
IF R = 20 .OR. EOR()
@23,20
WAIT "PRESS A KEY TO CONTINUE --------------------------------!!!"
CLEA
R = 7
ELSE
AN = SPACE(1)
@24,20 SAY "TO DISPLAY ANOTHER TRADE ANALYSIS (Y/N)?:"  GET AN

READ
IF AN = "N"
CLEA
SKIP
ELSE
C = RECNO()
C = C + 6
@1,2 CLEA TO 23,78
@1,10 TO 3,70 PANEL
SET COLOR TO B/G+
@3,20 SAY "PROFIT AND LOSS ACCOUNT AS AT --------"
@3,50 GET M_ARDATE PICT "(DD/MM/YY)"
READ
@4,2 TO 4,78 DOUBL
@5,12 SAY "EXPENSES"
@5,48 SAY "GROSS PROFIT B/D"
@6,12 SAY "--------"
@6,48 SAY "----------------"
@7,2 SAY "RATE AND RENT:"
@7,17 SAY M_RRENT(M_RRENT,7,2)
@8,2 SAY "WATER BILL:"
@8,14 SAY M_WTBILL(M_WTBILL,6,2)
@9,2 SAY "ELECTRIC BILL:"
@9,17 SAY M_ETBILL(M_ETBILL,6,2)
@10,2 SAY "CARRIAGE OUTWARD:"
@10,20 SAY M_COUTWD(M_COUTWD,6,2)
@11,2 SAY "BAD DEBTS:"
@11,13 SAY M_BDEBT(M_BDEBT,6,2)
@13,2 SAY "TOTAL AMT OF EXP.:"
@13,21 SAY M_TOTAL2(M_TOTAL2,8,2)
@12,2 TO 12,38
@14,2 SAY "NET PROFIT:"
@14,14 SAY M_NTPROF(M_NTPROF,10,2)
@7,42 SAY "GROSS PROFIT B/F:"
@7,60 SAY M_GPROF(M_GPROF,10,2)
C = C + 1
R = R + 1
SKIP
ENDIF
ENDIF
ENDDO
SA = " "
@24,25 SAY "TO DISPLAY FINAL TRADE ACCOUNT (Y/N)?:" GET SA
READ
IF R = 20 .OR. EOR()
@23,20
WAIT "PRESS ENTER KEY TO CONTINUE  --------------------------!!!"
CLEA
R = 7
DO WHILE .T.
CLEA
```

```
IF UPPER(SA) = "N"
CLEA
SKIP
ENDIF
IF UPPER(SA) =  "Y"
D = RECNO()
D = D +  6
R = 7
M_CAPACC = M_OSTOCK
@3,2 CLEA TO 24,78
@3,20 SAY "BALANCE SHEET ACCOUNT AS AT --------"
@3,48 GET M_ARDATE PICT "(DD/MM/YY)"
READ
@4,20 TO 4,55
@5,12 SAY "SET-UP INCOME ACCOUNT"
@6,12 SAY "----------------------"
@5,48 SAY "FIXED ASSETS"
@6,48 SAY "------------"
@7,2 SAY "CAPITAL A/C:"
@7,15 SAY M_CAPACC(M_CAPACC,8,2)
@8,2 SAY "ADD NET PROFIT:"
@8,18 SAY M_NTPROF(M_NTPROF,10,2)
@7,42 SAY "PREMISES:"
@7,52 SAY M_PREMIS(M_PREMIS,8,2)
@8,42 SAY "LAND & BUILDING:"
@8,59 SAY M_LANDB(M_LANDB,10,2)
@9,42 SAY "MISCELLANEOUS:"
@9,57 SAY M_MISCELL(M_MISCELL,6,2)
@10,2 TO 10,38
@10,40 TO 10,78
@11,2 SAY "CAPITAL A/C TOTAL:"
@11,21 SAY M_TOTAL4(M_TOTAL4,10,2)
@12,2 SAY "LESS DRAWINGS:"
@12,17 SAY M_LDRAWS(M_LDRAWS,8,2)
@13,2 SAY "BALANCE OF CAPITAL:"
@13,22 SAY M_TOTAL7(M_TOTAL7,9,2)
@14,12 SAY "CURRENT LIABILITIES"
@14,50 SAY "CURRENT ASSETS"
@14,12 SAY "-------------------"
@14,50 SAY "--------------"
@15,2 SAY "ADD CREDITS:"
@15,15 SAY M_CDAMT(M_CDAMT,8,2)
@16,2 SAY "ACCRUALS:"
@16,12 SAY M_ACCRU(M_ACCRU,6,2)
@16,42 SAY "CASH AT BANK:"
@16,54 SAY M_CHBANK(M_CHBANK,12,2)
@17,42 SAY "PREPAYMENT:"
@17,54 SAY M_PPAY(M_PPAY,6,2)
@18,42 SAY "CASH AT HAND:"
@18,56 SAY M_CHHAND(M_CHHAND,12,2)
@19,2 SAY "CURRENT LIAB. TOT.:"
@19,22 SAY M_TOTAL5(M_TOTAL5,9,2)
@19,42 SAY "CURRENT ASSETS TOT.:"
@19,63 SAY M_TOTAL6(M_TOTAL6,12,2)
@20,2 SAY "BALANCE LH SIDE:"
@20,20 SAY M_TOTAL8(M_TOTAL8,10,2)
@20,42 SAY "BALANCE RH SIDE:"
@20,59 SAY M_TOTAL9(M_TOTAL9,10,2)
D = D + 1
R = R + 1
ENDIF
RS = " "
IF R = EOR()
```

```
@23,20
WAIT "PRESS A KEY TO CONTINUE -------------------------------------------!!!"
CLEA
R = 7
SKIP
ELSE
@23,20
WAIT "PRESS ANY TO CONTINUE ---------------------------------------------!!!"
CLEA ALL
@13,10 TO 19,70 DOUBL
@16,20 SAY "ANY OTHER INFORMATION TO DISPLAY(D/N)?:" GET RS
READ
IF UPPER(RS) = 'D'
CLEA
LOOP
ELSE
CLEA
CLOSE ALL
EXIT
ENDIF
ENDIF
ENDDO
ENDIF
ENDDO
ENDDO
SET TALK ON
SET CLOCK OFF
RETURN

**** VIEW.PRG ****
*THIS PROGRAM IS DESIGNED SPECIFICALLY FOR VIEWING RECORDS
*IN THE DBASE FILES.
CLEA
SET TALK OFF
SET COLOR TO B+
@2,3 CLEA TO 24,79
@12,5 CLEA TO 16,70
SET ECHO OFF
FN = SPACE(6)
R = 2
C = 2
DO WHILE .T.
@2,1 TO, 24,79 DOUBL
@11,4 TO 16,78
@13,20 SAY "PUR,SAL,DEB,CRED,ITEM,EXP,INCO,INVENT,CUST,BANK,QUIT:"
@14,30 SAY "ENTER CURRENT FNAME:" GET FN PICT "@!"
READ
* TO DISPLAY THE INFORMATION IN THE FILE SELECTED
SET ECHO OFF
SELECT 1
USE PURCHASE INDEX S_NO
SELECT 2
USE SALES INDEX S_NO
SELECT 3
USE DEBTS INDEX S_CODE
SELECT 4
USE CREDIT INDEX CUST_CODE
SELECT 5
USE ITEM  INDEX S_NO
SELECT 6
USE EXPENSES INDEX S_NO
SELECT 7
USE INCOME INDEX S_NO
```

```
SELECT 8
USE STOCK INDEX STOCK_CD
SELECT 9
USE CUSTOMER INDEX CUST_CODE
SELECT 0
USE BANK INDEX S_NO
DO CASE
CASE FN = "PUR"
USE PURCHASE.DBF
SET COLOR TO G
CLEA
@2,2
DISPLAY  ALL
CLEA
R = 2
CASE FN = "SAL"
SET COLOR TO RB/W+
USE SALES.DBF
CLEA
@2,2
DISPLAY ALL
CLEA
R = 2
CASE FN  = "DEB"
SET COLOR TO R+/W
USE DEBTS.DBF
CLEA
@2,2
DISPLAY ALL
CLEA
R = 2
CASE FN = "CRED"
SET COLOR TO G/N+
USE CREDIT.DBF
CLEA
@2,2
DISPLAY ALL
CLEA
R = 2
CASE FN = "ITEM"
SET COLOR TO RB
USE ITEM.DBF
CLEA
@2,2
DISPLAY ALL
CLEA
R = 2
CASE FN = "EXP"
SET COLOR TO R
USE EXPENSES.DBF
CLEA
@2,2
DISPLAY ALL
CLEA
R = 2
CASE FN = "INCO"
SET COLOR TO G+/W,R+
USE INCOME.DBF
CLEA
@2,2
DISPLAY ALL
CLEA
R = 2
```

```
CASE FN = "INVENT"
SET COLOR TO GB/W
USE STOCK.DBF
CLEA
@2,2
DISPLAY ALL
CLEA
R = 2
CASE FN = "CUST"
SET COLOR TO B/W
USE CUSTOMER.DBF
CLEA
@2,2
DISPLAY ALL
CLEA
R = 2
CASE FN = "BANK"
SET COLOR TO W/B
USE BANK.DBF
CLEA
@2,2
DISPLAY ALL
CLEA
R = 2
CASE FN  = "QUIT"
CLEA
CLOSE ALL
QUIT
OTHERWISE
CLEA
EXIT
R = R + 1
ENDCASE
ENDD
SET TALK ON
RETURN

***** RECADD.PRG *********
* THIS IS RECORD ADDITION FOR DATA ENTRY ROUTINE.
     CLEA
     SET CLOCK ON
     SET TALK OFF
     CLEA
     SET ECHO OFF
     SET DATE BRITISH
     @2,3 CLEA TO 23,78
     @2,5 CLEA TO 22,74
     SET COLOR TO B/W
     @2,10 TO 23,70 DOUBL
     @3,12 FILL TO 22,76
     @4,14 TO 21,74 DOUBL
     @5,16 TO 20,72
     @6,18 TO 19,70 DOUBL
     * TELLS US WHAT OPERATION TO PERFORM
     DO WHILE .T.
         go top
         @7,20 CLEA TO 19,67
         @7,20 SAY "DATA ENTRY ROUTINE" PICT "@B"
         @8,20 SAY "------------------"
         @14,20 SAY "YOU ARE ABOUT TO ADD RECORDS -----------!!!"
         @18,20 SAY "PRESS ANY KEY TO CONTINUE --------------!!!"
         SEC = INKEY(0.6)
         CH = " "
```

```
@6,5 CLEA TO 19,78
@6,5 TO 19,78 DOUBL
SET COLOR TO G
@8,15 SAY "1" + SPACE(14) + "PURCHASES"
@9,15 SAY "2" + SPACE(14) + "SALES"
@10,15 SAY "3" + SPACE(14) + "DEBTS"
@11,15 SAY "4" + SPACE(14) + "CREDIT"
@12,15 SAY "5" + SPACE(14) + "ITEM"
@13,15 SAY "6" + SPACE(14) + "EXPENSES"
@15,15 SAY "7" + SPACE(14) + "INCOME"
@16,15 SAY "8" + SPACE(14) + "INVENTORY"
@17,15 SAY "9" + SPACE(14) + "CUSTOMER"
@18,15 SAY "0" + SPACE(14) + "BANK"
@14,54 SAY "ENTER CHOICE CODE:" GET CH PICT "9"
READ
* SET UP DBASE FILES IN A LOOP PLEASE STAY TUNED

* WE USE DO CASE AND ENDCASE to test which file is selected
DO CASE
   SET ECHO OFF
       CASE UPPER(CH) = "1"
            CLEA
            SET COLOR TO RB+
            DO AZZ_6
       CASE UPPER(CH) = "2"
            CLEA
            SET COLOR TO BG/G+
            DO AZZ_7
       CASE UPPER(CH) = "3"
            CLEA
            SET COLOR TO W/R+
            DO AZZ_8
       CASE UPPER(CH) = "4"
            CLEA
            SET COLOR TO B+
            DO AZZ_9
       CASE UPPER(CH) = "5"
            CLEA
            SET COLOR TO RG
            DO AZZ_17
       CASE UPPER(CH) = "6"
            CLEA
            SET COLOR TO R+
            DO AZZ_10
       CASE UPPER(CH) = "7"
            CLEA
            SET COLOR TO G
            DO AZZ_12
       CASE UPPER(CH) = "8"
            CLEA
            SET COLOR TO G+
            DO AZZ_13
       CASE UPPER(CH) = "9"
            CLEA
            SET COLOR TO RG
            DO AZZ_14
       CASE UPPER(CH) = "0"
            CLEA
            SET COLOR TO  RB
            DO AZZ_15
       OTHERWISE
            @10,5 CLEA TO 24,78
            @10,5 TO 18,70 PANEL COLOR G
```

```
@14,25 SAY "WRONG DECISION MADE ---------TRY AGAIN!"
RS = " ".
@21,20 SAY "ANY OTHER OPERATION (Y/N)?:" GET RS PICT "Y"
       READ
       CLEA
       AS = SPACE(1)
       SET COLOR TO B/W
       @20,5 CLEA TO 24,78
       IF RS = "Y"
           CLEA
           SET COLOR TO GB+/B+
           DO WHILE .T.
                  GO TOP
                  CLEA
                  AS = " "
                  @10,1 TO 19,79 DOUBL
                  @10,2 TO 19,78 PANEL COLOR R
                  @10,2 FILL TO 19,78 COLOR RB
                  @12,10 SAY "R" + SPACE(5) + "RECADD"
                  @13,10 SAY "M" + SPACE(5) + "RECMODI"
                  @14,10 SAY "D" + SPACE(5) + "RECDEL"
                  @15,10 SAY "A" + SPACE(5) + "ANAL_REP"
                  @16,10 SAY "V" + SPACE(5) + "VIEW"
                  @17,10 SAY "Q" + SPACE(5) + "QUIT"
                  @15,31 SAY "ENTER PROGRAM LETTER:"
                  @15,67 GET AS PICT "A"
                  READ
          IF .NOT.UPPER(AS) $"R,M,D,A,V,Q"
          @13,5 CLEA TO 17,76
          @13,5 TO 17,76 DOUBL
          SET COLOR TO GB+
          @14,10 SAY "IMPROPER KEY SELECTION !!!"
          @16,10 SAY "PLEASE TRY AGAIN -!!!"
          WAIT SPACE(2) + "PRESS ANY KEY TO CONTINUE !!!"
                      CLEA
                      SKIP
                  ENDIF
    * TO EVALUATE THE INPUT WITH DO CASE/ENDCASE COMMAND
                  SET ECHO OFF
                  DO CASE
                        CASE UPPER(AS) = "R"
                             CLEA
                             SET COLOR TO B/W
                             DO RECADD
                        CASE UPPER(AS) = "M"
                             CLEA
                             SET COLOR TO BG/R,W/R
                             DO RECMODI
                        CASE UPPER(AS) = "D"
                             CLEA
                             SET COLOR TO R
                             DO RECDEL
                        CASE UPPER(AS) = "A"
                             CLEA
                             SET COLOR TO G+
                             DO ANAL_REP
                        CASE UPPER(AS) = "V"
                             CLEA
                             SET COLOR TO W/B
                             DO VIEW
                        CASE UPPER(AS) = "Q"
                             CLEA
                             CLOSE DATA
```

```
                                           EXIT
                                      OTHERWISE
                                           CLEA
                                           SET COLOR TO B
                                           SKIP
                               ENDCASE
        ENDDO
     ELSE
        CLEA
        EXIT
     ENDIF
     ENDCASE
     ENDDO
     SET CLOCK OFF
     SET TALK ON
     RETURN


* PROCEDURE A_SALES
* THIS SUBPROGRAM IS MAINLY USED FOR DATA ADDITION INTO SALES FILE.
CLEA
SET TALK OFF
@2,2 CLEA TO 20,70
@3,5 CLEA TO 22,78
R = 4
SET DATE BRITISH
SET COLOR TO BG/W
AN = " "
STORE 0 TO M_IQTYS,M_SRATE,M_SAMT,M_GTAMT,M_S_INVNO,M_SPARTC,M_ITMAC
STORE 0 TO M_KMACHC
STORE SPACE(20) TO M_INAME,M_CNAME
*STORE SPACE(10) TO M_ITEM2,M_ITEM3,M_ITEM4
*STORE SPACE(15) TO M_ITEM1
STORE SPACE(3) TO M_FILE_NO,M_STOCK_CD,M_SCODE,M_SNO
M_ICODE = SPACE(9)
M_CADD = SPACE(25)
M_SDATE = CTOD(TODAY)
@2,2 CLEA TO 20,70
DO WHILE .NOT. EOF()
@3,5 CLEA TO 22,78
USE SALES INDEX S_NO

*SUBPROGRAM FOR ITEM_CODE CONCARTINATION
DO AZZ_16
@3,5 TO 22,70 DOUBL
SP = SPACE(4)
SET ECHO OFF
@4,10 SAY M_ICODE + LTRIM(STR(M_ICODE)) + SP + M_INAME PICT "@!"
IF .NOT. EOF()
@23,20
WAIT "PRESS ANY KEY TO CONTINUE ---------------------!!!"
@3,6 CLEA TO 22,69
R = 4
@4,10 SAY  M_SRATE +  SP + M_IQTYS + SP + M_SAMT
ENDIF
IF.NOT. EOF()
@23,20
WAIT "PRESS ANY KEY TO CONTINUE -------------------------------!!!"
@3,6 CLEA TO 22,69
R = 4
@4,10 SAY M_GTAMT(M_GTAMT,10,2)
ENDIF
SET ECHO OFF
```

```
*APPEND BLANK COMMAND IS USED TO MAKE SPACE FOR THE INCOMING DATA
APPEND BLANK
SET FORMAT TO SALES
READ
M_CCODE = CUST_CODE
M_SRATE = SELL_RATE
M_SAMT = TOTAL_AMT
M_FILE_NO = FILE_NUMB
M_STOCK_CD = STOCK_CD
M_SDATE  = SELL_DATE
M_CNAME = CUST_NAME
M_CADD  =  CUST_ADD
M_INAME = ITEM_NAME
M_S_INVNO = S_INVNO
M_GTAMT = GT_AMT
M_SNO = S_NO
*REPLACE COMMAND IS USED TO REPLACE ALL MEMORY VARIABLES
*WITH THEIR RESPECTIVE FIELD NAMES.
REPLA SELL_RATE WITH M_SRATE, TOTAL_AMT WITH M_SAMT
REPLA  FILE_NUMB WITH M_FILE_NO,  STOCK_CD WITH M_STOCK_CD
REPLA CUST_ADD WITH M_CADD, CUST_NAME WITH M_CNAME
REPLA SELL_DATE WITH M_SDATE, ITEM_NAME WITH M_INAME
REPLA  S_INVNO WITH M_S_INVNO,  ITEM_CODE WITH M_ICODE
REPLA GT_AMT WITH M_GTAMT, CUST_CODE WITH M_CCODE
REPLA S_NO WITH M_SNO
@22,10 CLEA TO 22,70
@22,15 SAY "IS SALES MADE ON CREDIT (Y/N)?:" GET AN PICT "Y"
READ
IF UPPER(AN) = "Y"
DO AZZ_9
USE CREDIT INDEX CUST_CODE
ELSE
CLEA
@23,20 SAY "THE CUSTOMER PAID CASH ---------- THANKS!"
EXIT
ENDIF
DS = SPACE(1)
@23,15 SAY "ADD MORE RECORDS (Y/N)?:" GET DS PICT "!"
READ
*THE USER INPUT ANALYSIS FOR THE ADDITION OF MORE DATA
IF DS = "Y"
CLEA
LOOP
ENDIF
IF DS  = "N"
CLEA
CLOSE ALL
EXIT
ENDIF
ENDDO
SET TALK ON
RETURN


PROCEDURE A_CREDIT
*THIS SUBPROGRAM IS USED TO ADD RECORDS TO CREDIT FILE.
SET TALK OFF
CLEA
SET DATE BRITISH
SET COLOR TO  G+/W+
@2,2 CLEA TO 22,74
@3,5 CLEA TO 20,70
DS = " "
```

```
M_C_DATE = CTOD("TODAY")
M_CNAME = SPACE(20)
M_CADD = SPACE(25)
M_INAME  = SPACE(20)
M_ICODE =  SPACE(9)
STORE O TO  M_G_TAMT,M_IQTYC,M_C_AMT,M_IRATE,M_CCODE,M_AMT_PAID
@3,2 CLEA TO 22,74
DO WHILE .NOT. EOF()
GO TOP
USE CREDIT INDEX CUST_CODE
*APPEND BLANK IS USED TO MAKE ROOM IN THE DBASE  FILE FOR MORE DATA.
APPEND BLANK
SET FORMAT TO CREDIT
READ
*TRANSFER OF FIELD CONTENTS INTO MEMORY VARIABLES.
M_C_DATE = CRED_DATE
M_CNAME =  CUST_NAME
M_CADD = CUST_ADD
M_INAME = ITEM_NAME
M_ICODE  = ITEM_CODE
M_G_TAMT  =  GTOT_AMT
M_IQTYC = IQTY_CRED
M_C_AMT =  C_BAL
M_IRATE = ITEM_RATE
M_CCODE = CUST_CODE
M_AMT_PAID = AMT_DEPO
M_GT_AMD  =    GT_AMT
*CALCULATION FOR THE BALANCE TO  BE PAID
SET COLOR TO G+.
M_G_TAMT = M_IQTYC  *  M_IRATE
M_C_AMT = M_G_TAMT - M_AMT_PAID
STORE 0 TO M_GT_AMT,MGT_AMT
USE CREDIT.DBF
@10,20 SAY C_BAL(C_BAL,10,2)
M_GT_AMT  = C_BAL
MGT_AMT =  M_GT_AMT + M_C_AMT
DISPLAY  M_C_AMT, M_GT_AMT, MGT_AMT
@23,20
WAIT "PRESS ANY KEY TO RESUME OPERATION ---------------------------!!!"
CLEA
*WITH REPLACE COMMAND WE SHALL ALL THE RECORDS IN THEIR RESPECTIVE
*PLACES.
REPLA CRED_DATE WITH M_C_DATE, CUST_NAME WITH M_CNAME
REPLA CUST_ADD WITH M_CADD, ITEM_NAME WITH M_INAME
REPLA ITEM_CODE WITH  M_ICODE, GTOT_AMT  WITH M_G_TAMT
REPLA IQTY_CRED WITH M_IQTYC, C_BAL WITH  M_C_AMT
REPLA ITEM_RATE WITH  M_IRATE,  CUST_CODE WITH M_CCODE
REPLA AMT_DEPO WITH M_AMT_PAID, GT_AMT WITH MGT_AMT
@23,15 SAY "ADD MORE RECORDS (Y/N)?:" GET DS PICT "Y"
READ
IF UPPER(DS) = "Y"
CLEA
LOOP
ELSE
CLEA
CLOSE ALL
EXIT
ENDIF
ENDD
SET TALK ON
RETURN
```

```
                 **** MAINMENU.PRG ****
* THIS PROGRAM IS CALLED MAIN MENU PROGRAM
* IT SETS THE MAIN OPERATIONS IN A DESIGN MANNER
   CLEA
SET TALK OFF
KOUNT = 0
CLEA
SET CLOCK ON
SET COLOR TO GR/B+
@1,5 CLEA TO 5,78
SET COLOR TO R/B
DO WHILE .T.
        GO TOP
        CLEA
        SET COLOR TO N+/RB
        @1,5 TO 6,78 DOUBL COLOR W
        SET COLOR TO R+/G+
       @2,7 SAY "COMPUTERIZATION OF WAREHOUSE STOCK INVENTORY"
        @3,7 SAY SPACE(6) + "AND ANALYSIS PROCEDURES"
        @4,7 SAY SPACE(15) + "VIS - A - VIS"
        @5,7 SAY SPACE(10) + "TONY STORE ENTERPRISES"
        SEC = 0
        IF SEC = INKEY(0.3)
                WAIT
                @7,6 CLEA TO 12,70
                SET COLOR TO BR
                @8,20 SAY "M A I N   M E N U"
                @9,20 TO 9,36 DOUBL
                @10,15 SAY "TASK CODE"
                @10,40 SAY "TASK OPERATIONS"
                @11,15 TO 11,59
                SET COLOR TO B+
                @13,6 CLEA TO 23,77
                @13,6 TO 23,77 DOUBL
                @13,6 FILL TO 23,77 COLOR G+/W
                STORE "X" TO CHOICE
                SET COLOR TO W/G*,R+/B+
             @14,14 SAY "A" + SPACE(22) + "ADDITION"
             @15,14 SAY "B" + SPACE(22) + "MODIFICATION"
             @16,14 SAY "C" + SPACE(22) + "DELETION"
             @17,14 SAY "D" + SPACE(22) + "INVENTORY"
             @18,14 SAY "E" + SPACE(22) + "ANAL_REP"
             @19,14 SAY "F" + SPACE(22) + "QUIT"
             @20,10 CLEA TO 22,75
             @21,10 TO 22,75 PANEL COLOR G+
             @20,15 SAY "ENTER CODE NUMBER :" GET CHOIC PICT"A"
             READ
        ENDIF
SET COLOR TO GR
DO CASE
        CASE UPPER (CHOIC) = "A"
                CLEA
                SET COLOR TO B/W
                DO REC_ADD
        CASE UPPER (CHOIC) = "B"
                CLEA
                SET COLOR TO BG/R,W/R
                DO REC_MODI
        CASE UPPER (CHOIC) = "C"
                CLEA
                SET COLOR TO R
                DO REC_DEL
        CASE UPPER (CHOIC) = "D"
```

```
                        CLEA
                        SET COLOR TO GB/RG
                        DO INVENTORY
                CASE UPPER (CHOIC) = "E"
                        CLEA
                        SET COLOR TO G+/R,B+
                        DO ANAL_REPT
                OTHERWISE .
                        STORE " " TO ANS
                        @18,5 CLEA TO 20,70
                        @7,10 TO 16,70 DOUBL
        ENDCASE
        @18,5 CLEA TO 20,76
        SET COLOR TO N
        @18,10 TO 20,70 PANEL COLOR R
        @19,15 SAY "DO YOU WANT TO QUIT DBASE (Y/N)?:"
        @19,50 GET ANS PICT"Y"
        READ
        * TO ANALYSE THE USER'S RESPONSE
        KOUNT = KOUNT + 1
        IF UPPER (ANS) = "N"
                CLEA
                SET COLOR TO GR
                LOOP
        ENDIF
        IF UPPER (ANS) = "Y"
                CLEA
                SET COLOR TO N
                SKIP
                CLOSE ALL
                SET TALK ON
                SET SCORE ON
                EXIT
        ENDIF
        ENDDO
        RETURN

**** RECMODI.PRG ****
* THIS PROGRAM IS CALLED RECORD MODIFICATION TO CHANGE DATE IN DBF.
CLEA
SET CLOCK ON
SET BELL OFF
SET COLOR TO BG/R,W/R+
CLEA
@2,1 CLEA TO 20,78
@2,2 TO 20,78 DOUBL COLOR W
SET COLOR TO BG/R+
@4,4 TO 18,76 DOUBL COLOR N
@6,6 TO 16,74
DO WHILE .T.
    GO TOP
    CLEA
    FNAME = SPACE(1)
    @11,25 TO 11,49
    @13,15 SAY "YOU ARE ABOUT TO MODIFY RECORD ------------!!!"
    @12,13 TO 14,66 PANEL
    Q = 0
    IF Q = INKEY(0.1)
         @16,25
         WAIT "PRESS ANY KEY TO CONTINUE -------------------------!!!"
         CLEA
    ENDIF
    SET COLOR TO GB
```

```
CLEA
@6,5 CLEA TO 19,78
@6,5 TO 19,78 DOUBL COLOR B+
@6,6 FILL TO 19,78 COLOR RB+
@8,20 SAY "0" + SPACE(5) + "PURCHASE"
@9,20 SAY "1" + SPACE(5) + "SALES"
@10,20 SAY "2" + SPACE(5) + "DEBTS"
@11,20 SAY "3" + SPACE(5) + "CREDIT"
@12,20 SAY "4" + SPACE(5) + "ITEM"
@13,20 SAY "5" + SPACE(5) + "EXPENSES"
@14,20 SAY "6" + SPACE(5) + "INCOME"
@15,20 SAY "7" + SPACE(5) + "INVENTORY"
@16,20 SAY "8" + SPACE(5) + "CUSTOMER"
@17,20 SAY "9" + SPACE(5) + "BANK"
    @20,25 SAY "PRESS THE UPPER CASE KEY:" GET FNAME PICT "9"
    READ
    * TO SHOW THE USER THAT THE FILE IS NOT EXISTING
          SET ECHO OFF
          SET COLOR TO W+/G+
          DO CASE
                CASE UPPER(FNAME) = "0"
                     DO AZZ_6
                CASE UPPER(FNAME) = "1"
                     DO AZZ_7
                CASE UPPER(FNAME) = "2"
                     DO AZZ_8
                CASE UPPER(FNAME) = "3"
                     DO AZZ_9
                CASE UPPER(FNAME) = "4"
                     DO AZZ_17
                CASE UPPER(FNAME) = "5"
                     DO AZZ_10
                CASE UPPER(FNAME) = "6"
                     DO AZZ_12
                CASE UPPER(FNAME) = "7"
                     DO AZZ_13
                CASE UPPER(FNAME) = "8"
                     DO AZZ_14
                CASE UPPER(FNAME) = "9"
                     DO AZZ_15
                OTHERWISE
                     SET COLOR TO B+/W,B+
                     AS = SPACE(1)
                 @12,5 CLEA TO 16,70
                 @12,5 TO 16,70
                 @13,10 SAY "INCORRECT FNAME SELECTED --------!!!"
                 @15,10 SAY "TO TRY AGAIN (Y/N)?" GET AS PICT "Y"
                     READ
                     IF AS = "Y"
                         CLEA ALL
                         LOOP
                    ENDIF
                    IF AS = "N"
                    CH = " "
                    RS = SPACE(1)
                    SET ECHO OFF
                    @22,1 CLEA TO 24,78
                    @22,10 TO 24,72 PANEL COLOR W
                   @23,15 SAY "ANY OTHER RECORD TO MODIFY (Y/N)?:"
                         @23,40 GET RS PICT "N"
                         READ
                       * USER'S INPUT ANDLYSIS
                         IF UPPER(RS) = "Y"
```

```
                                    CLEA
                                    RETURN
                             ELSE
                                    CLEA
                                    M_ZENGER = SPACE(1)
                                    DO WHILE .T.
                                        @9,5 CLEA TO 20,70
                                        @9,5 TO 20,70 DOUBL
                                        @9,6 FILL TO 20,70 COLOR G
                                        @11,20 SAY "R        RECADD"
                                        @12,20 SAY "M        RECMODI"
                                        @13,20 SAY "D        RECDEL"
                                        @14,20 SAY "A        ANAL_REP"
                                        @15,20 SAY "V        VIEW"
                                        @16,20 SAY "Q        QUIT"
                                          @18,3 TO 20,73 PANEL
                                          @19,25 SAY "ENTER CODE:"
                                          @14,70 GET CH PICT "!"
                                          READ
                                          DO CASE
                                              CASE  UPPER(CH) = "R"
                                                   CLEA
                                                    SET COLOR TO B/W
                                                    DO RECADD
                                               CASE UPPER(CH) = "M"
                                                    CLEA
                                               SET COLOR TO BG/R,W/R
                                                    DO RECMODI
                                               CASE UPPER(CH) = "D"
                                                    CLEA
                                                    SET COLOR TO R
                                                    DO RECDEL
                                               CASE UPPER(CH) = "I"
                                                    CLEA
                                                  SET COLOR TO GB/RG
                                                    DO INVENTORY
                                               CASE UPPER(CH) = "A"
                                                    CLEA
                                                    SET COLOR TO G+
                                                    DO ANAL_REP
                                              CASE UPPER(CH)  =  "V"
                                                    CLEA
                                                    SET COLOR TO W/B
                                                    DO VIEW
                                              CASE UPPER(CH) = "Q"
                                                    CLEA
                                                    CLOSE ALL
                                                    EXIT
                                              OTHERWISE
                                                    CLEA
                                                    SET COLOR TO B
                                                    QUIT
                                         ENDCASE
                     ENDDO
    ENDIF
ENDIF
ENDCASE
ENDDO
CLOSE DATABASE
SET BELL ON
SET CLOCK OFF
RETURN
```

```
*PROCEDURE A_INVENTORY
*ADD ITEM TO THE STOCK FILE
CLEA
SET CLOCK ON
SET TALK OFF
@2,5 CLEA TO 20,70
SET COLOR TO G
SET DATE BRITISH
@2,5 TO 23,78 DOUBL COLOR W
RP = SPACE(1)
*INITIALIZATION OF MEMORY VARIABLES
M_INAME = SPACE(25)
M_STOCK_CD = SPACE(3)
S = SPACE(3)
STORE 0 TO M_IQTY,MQISTOCK,M_IRATE,M_TOTAL1,M_TOTAL2
MLUPDATE = CTOD("TODAY")
@2,5 CLEA TO 15,78
DO WHILE .NOT. EOF()
GO TOP
CLEA
@2,5 TO 20,70 DOUBL
M_STOCK_CD = STOCK_CD
M_INAME = ITEM_NAME
M_STCODE = ST_CODE
MLUPDATE = LT_UPDATE
M_IQTY =  ITEM_QTY
MQISTOCK =  QI_STOCK
M_IRATE = ITEM_RATE
M_TOTAL1 = TOTAL_AMT
M_TOTAL2 = AMT_ISTOCK
USE STOCK.DBF
@15,20 SAY "ENTER ITEM STOCK CODE:" GET M_STOCK_CD
READ
*LOCATE THIS ITEM IN THE STOCK FILE
LOCATE FOR STOCK_CD = M_STOCK_CD
IF .NOT. FOUND()
CLEA
@11,2 TO 16,70
SET COLOR TO B
@12,15 SAY "WRONG ITEM STOCK CODE "+OPT+" ENTERED -----------------!!!"
@14,15 SAY "RECORD NOT FOUND --- ENTER CORRECT STOCK CODE ---------!!!"
@15,10 SAY "PLEASE! PRESS ANY KEY TO CONTINUE --------------------!!!"
CLEA
LOOP
ENDIF
IF FOUND()
CLEA
SET COLOR TO B+/W
@14,5 TO 18,79  DOUBL
A = RECNO()
A = A + 7
@15,10 SAY LT_UPDATE PICT "(DD/MM/YY)"
@16,20 SAY QI_STOCK(QI_STOCK,10,2)
@16,30 SAY ITEM_NAME PICT "@!"
@16,65 SAY AMT_ISTOCK(AMT,10,2)
READ
@18,10  CLEA
@19,25 SAY "OPENING STOCK:=" + "#" + M_OSTOCK(AMT,8,2)
@20,10 TO 21,78 PANEL
ENDIF
USE PURCHASE.DBF
@2,5 CLEA TO 20,79
@14,15 SAY "ENTER ITEM STOCK CODE:" GET M_STOCK_CD PICT "@!"
```

```
READ
LOCATE FOR STOCK_CD  = M_STOCK_CD
IF FOUND()
CLEA
SET COLOR TO G+,B/W
@14,5 TO 18,78  DOUBL
@16,7 SAY "ITEM_QTY" + S + "ITEM_NAME" + S + "ITEM_RATE" + S +
"TOTAL_AMT"
@17,7 SAY ITEM_QTY + S + ITEM_NAME + S + ITEM_RATE + S + TOTAL_AMT
*UPDATING THE STOCK FILE BY  ADDITION.
SP = SPACE(3)
MQISTOCK = QI_STOCK + ITEM_QTY
M_TOTAL1 =  ITEM_QTY * ITEM_RATE
M_TOTAL2 = M_TOTAL2 + AMT_ISTOCK
@20,10 SAY MQISTOCK + SP + M_TOTAL1 + SP + M_TOTAL2
@22,5 TO 23,78 PANEL
ENDIF
CLEA ALL
@2,5 TO 22,70 DOUBL
@15,15
WAIT "----ANY OTHER RECORD TO ADD --------- (Y/N)?:" TO RP
IF UPPER(RP) =  "Y"
CLEA
LOOP
ELSE
USE STOCK.DBF
APPEND BLANK
SET FORMAT TO STOCK
READ
REPLA AMT_ISTOCK WITH M_TOTAL2, QI_STOCK WITH MQISTOCK
REPLA ITEM_NAME WITH M_INAME, OPEN_STOCK WITH M_OSTOCK
CLOSE DATABASE
EXIT
ENDIF
ENDD
SET CONFIRM ON
SET TALK ON
SET CLOCK  OFF
RETURN


PROCEDURE ADD_BANK
*  WE USE THIS SUBPROGRAM TO ADD MONEY IN OUR BANK ACCOUNT
CLEA
SET CLOCK ON
SET ECHO OFF
SET DATE BRITISH
CLEA
@2,15 CLEA TO 15,70
SET COLOR TO B+
RP = SPACE(1)
STORE "CTOD(TODAY)" TO M_DATE_DP,M_DATE_CQ
STORE SPACE(20) TO M_BK_NM,M_PAYER,M_BKN_OCQ
STORE SPACE(25) TO M_BK_ADD,M_N_ISSUE
STORE 0 TO M_BK_CODE,M_BK_ACNM,M_AMT_IBK,M_LODGE,M_CHEQUE,MGT_AMT
STORE 0 TO M_CHHAND,M_CHBANK,M_CD50,M_CD20,M_CD10,M_CD5,M_CDC1
STORE 0 TO M_CDC_50,M_CDC_10,M_CDC_5,M_CDC_1,AMT1,AMT2,AMT3,AMT4
STORE 0 TO AMT5,AMT6,AMT7,AMT8,AMT9
@3,5 CLEA TO 22,75
SET ECHO OFF
DO WHILE .NOT. EOF()
GO TOP
@2,5 CLEA TO 22,75
```

```
APPEND BLANK
SET FORMAT TO BANK
READ
STORE " " TO RS
@14,15 SAY " IS YOUR LODGEMENT CHEQUE (Y/N)?:" GET RS PICT "Y"
READ
IF UPPER(RS) = "N"
CLEA
DO CAL_CASH WITH M_CD50,M_CD20,M_CD10,M_CD5,M_CDC1,M_CDC_50,M_CDC_10
DO CAL_CASH WITH M_CDC_5,M_CDC_1,AMT1,AMT2,AMT3,AMT4,AMT5,AMT6,AMT7
DO CAL_CASH WITH AMT8,AMT9,M_CHEQUE,M_CHHAND,M_CHBANK,M_LODGE,MGT_AMT
@16,15 SAY M_LODGE(M_LODGE,8,2)
@17,15 SAY M_CHBANK(M_CHBANK,10,2)
@18,15 SAY M_CHHAND(M_CHHAND,8,2)
ENDIF
USE INCOME.DBF
@19,20 GET G_TOTAMT PICT "99999999.99"
MGTOT_AMT = G_TOTAMT
IF UPPER(RS) = "Y"
CLEA
M_LODGE = M_CHEQUE
MGT_AMT = M_CHBANK + M_LODGE
M_CHHAND = MGTOT_AMT - MGT_AMT
@16,20 SAY "CHEQUE AMOUNT:=" + M_LODGE(M_LODGE,8,2)
@17,20 SAY "AMOUNT IN BANK:=" +  M_CHBANK(M_CHBANK,10,2)
@18,20 SAY  "AMOUNT AT HAND:=" + M_CHHAND(M_CHHAND,8,2)
ENDIF
M_DATE_DP =  DT_DEPOSIT
M_DATE_CQ = DT_CHEQUE
M_BK_NM  =  BANK_NM
M_PAYERNM = PAYER_NM
M_BKN_OCQ =  BANKN_OCQ
M_BK_ADD = BANK_ADD
M_CUST_NM = CUST_NAME
M_BK_CODE = BANK_CODE
M_BK_ACNO = BANK_ACNO
M_CHBANK = CASHBANK
M_LODGE = LODGEMENT
M_CHEQUE =  CHQ_AMT
ANS = SPACE(1)
M_CQ_NUMB =  CQ_NUMB
MGT_AMT =  GTOT_AMT
M_CHHAND  = CASHHAND
M_CD50 = CASHN50
M_CD20 = CASHN20
M_CD10 = CASHN10
M_CD5 = CASHN5
M_CDC1 = COINS1
M_CDC_50 = COINS_50
M_CDC_10 = COINS_10
M_CDC_5 = COINS_5
M_CDC_1 = COINS_1
@23,10
WAIT "ARE YOUR ENTRIES CORRECT (Y/N)?:" TO RP
@14,20 SAY "ADD MORE RECORDS (Y/N)?:"
SET COLOR TO G
@14,50 GET ANS PICT "Y"
READ
IF UPPER(RP)  = "Y"
IF UPPER(ANS)  = "Y"
CLEA
LOOP
ENDIF
```

```
ENDIF
IF UPPER(RP)   =   "N"
IF UPPER(ANS)  =   "N"
CLEA
CLOSE ALL
EXIT
ENDIF
ENDIF
ENDDO
SET TALK ON
SET CLOCK OFF
RETURN


PROCEDURE ADD_ITEM
*THIS SUBPROGRAM  IS FOR DATA ADDITION INTO ITEM DBASE FILE.
CLEA
SET TALK OFF
SET ECHO OFF
@2,5 TO 15,76 DOUBL
SET COLOR TO GR+/W+
M_STOCK_CD = SPACE(3)
RP = SPACE(1)
M_ICODE = SPACE(9)
M_INAME = SPACE(20)
M_ITEM1 = SPACE(15)
STORE SPACE(10) TO M_ITEM2,M_ITEM3,M_ITEM4
STORE 0 TO M_SPARTC,M_ITMAC,M_KMACHC,M_PRATE
CLEA
DO WHILE .NOT. EOF()
GO TOP
@2,5 TO  22,70 DOUBL
SET ECHO OFF
SET COLOR TO GR+
USE ITEM.DBF
*APPEND BLANK IS USED TO CREATE ROOM IN THE DABSE FILE
APPEND BLANK
SET FORMAT TO ITEM
READ
*TRANSFER OF FIELD CONTENTS INTO MEMORY VARIABLES
M_STOCK_CD = STOCK_CD
M_ICODE = ITEM_CODE
M_INAME = ITEM_NAME
M_ITEM1 = SPARE_PT
M_ITEM2 = I_TMAN
M_ITEM3 = K_MACH
M_ITEM4 = STOCK_INM
M_SPARTC = SPART_CODE
M_ITMAC = I_TMAC
M_KMACHC = K_MACHC
M_PRATE =  P_RATE
@23,10
WAIT "ADD MORE RECORDS(Y/N)?:" TO RP
IF RP $ 'Y'
CLEA
RETURN
ENDIF
IF RP $ 'N'
CLEA
CLOSE ALL
EXIT
ENDIF
ENDDO
```

```
SET TALK ON
SET CLOCK OFF
RETURN


PROCEDURE  MOD_SALES
*SUBPROGRAM FOR SALES RECORD MODIFICATION
CLEA
SET DATE BRITISH
SET TALK OFF
@4,5 CLEA TO 18,78
@4,5 TO 18,78
SET ECHO OFF
SET COLOR TO R/B+,G/B+
STORE SPACE(3) TO M_FILE_NO,M_STOCK_CD,M_CCODE
STORE SPACE(10) TO M_ITEM2,M_ITEM3,M_ITEM4
STORE SPACE(15) TO M_ITEM1
STORE SPACE(20) TO M_CNAME,M_INAME
M_CADD = SPACE(25)
M_ICODE = SPACE(9)
M_SDATE = CTOD(TODAY)
RP = SPACE(1)
STORE 0 TO M_S_INVNO,M_SPARTC,M_ITMAC,M_IQTYS,M_KMACHC,M_SRATE,M_SAMT
STORE 0 TO M_GTAMT
@12,5 CLEA TO 20,70
DO WHILE .T.
GO TOP
@9,10 TO 13,74
USE SALES.DBF
@11,20 SAY "ENTER ITEM CODE:" GET M_ICODE PICT "@!"
READ
SET ECHO OFF
SET COLOR TO  G/B+
LOCATE FOR ITEM_CODE = M_ICODE
IF .NOT. FOUND()
CLEA
RP = " "
@5,10 SAY  "RECORD NOT FOUND ----------------!!!"
@11,10 SAY "WRONG ITEM CODE ENTERED ---------------!!!"
@17,10
WAIT "PRESS ANY KEY TO CONTINUE ----------------!!!"
@20,25 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET RP PICT "Y"
READ
IF RP = "Y"
CLEA
LOOP
ELSE
CLEA
CLOSE DATA
EXIT
ENDIF
ENDIF
IF FOUND()
CLEA
SET COLOR TO RB
SET ECHO OFF
@13,5 TO 15,72
@14,25 SAY "TO VIEW THESE RECORDS (Y/N)?:"  GET RP PICT "Y"
READ
IF  UPPER(RP) =  "Y"
CLEA
LIST ALL
Q = INKEY(0.1)
```

```
CLEA
SET FORMAT TO PURCHASE
READ
ENDIF
IF UPPER(RP) =  "N"
CLEA
SET FORMAT TO PURCHASE
READ
ENDIF
ENDIF
@14,20 SAY "INPUT CURRENT  RECORDS  ---- PRESS ENTER KEY!"
SET ECHO OFF
SET COLOR TO R/B+
STORE "  " TO YN
CLEA
@21,20
WAIT "TO MODIFY MORE RECORDS (Y/N)?:" TO YN
IF UPPER(YN) $ "Y"
CLEA
LOOP
ELSE
CLEA
REPLA  SELL_DATE WITH M_SDATE, ITEM_QTY WITH M_IQTYS
REPLA ITEM_NAME WITH M_INAME, ITEM_RATE  WITH M_SRATE
REPLA TOTAL_AMT WITH M_SAMT, GD_TOT_AMT WITH M_GTAMT
REPLA FILE_NUMB WITH M_FILE_NO, C_NAME WITH M_CNAME
REPLA C_ADD WITH M_CADD, S_INVNO WITH M_S_INVNO
REPLA STOCK_CD WITH M_STOCK_CD, CUST_CODE WITH M_CUS_CODE
REPLA I_ST_NAME WITH M_ITEM4
CLOSE ALL
EXIT
ENDIF
ENDDO
SET TALK ON
SET CLOCK OFF
RETURN


PROCEDURE  MOD_DEBTS
*THIS SUBPROGRAM IS FOR DEBIT RECORD MODIFICATION
CLEA
SET DATE BRITISH
SET TALK OFF
@4,5 CLEA TO 18,78
@4,5 TO 18,78
SET ECHO OFF
SET COLOR TO R/B+
STORE 0 TO M_SCODE,M_IQTY,M_IRATE,M_DBAL,M_AMT_PAID,M_GD_TOTAL
STORE SPACE(20) TO M_SNAME,M_INAME
M_SADD = SPACE(25)
M_ICODE = SPACE(9)
M_D_DATE = CTOD(TODAY)
RP = SPACE(1)
@6,5 CLEA TO 20,70
DO WHILE .T.
@9,10 TO 13,74 DOUBL
USE DEBTS.DBF
@15,20 SAY "ENTER SUPPLIERS CODE:" GET M_SCODE PICT "999"
READ
SET ECHO OFF
SET COLOR TO  G/B+
LOCATE FOR S_CODE = M_SCODE
IF .NOT. FOUND()
```

```
CLEA
DC = SPACE(1)
@5,10 SAY  "RECORD NOT FOUND -----------------!!!"
@11,10 SAY "WRONG ITEM CODE ENTERED -----------!!!"
@17,20
WAIT " PRESS ANY KEY TO CONTINUE ---------------!!!"
@20,25 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET DC PICT "Y"
READ
IF DC = "Y"
CLEA
LOOP
ENDIF
IF DC = "N"
CLEA
CLOSE ALL
EXIT
ENDIF
ENDIF
IF FOUND()
CLEA
SET ECHO OFF
SET COLOR TO RB
12,5 TO 16,72
@14,25 SAY "TO VIEW THESE RECORDS (Y/N)?:"  GET RP PICT "Y"
READ
IF  UPPER(RP) =  "Y"
CLEA
DISPLAY ALL
Q = INKEY(0.1)
CLEA
SET FORMAT TO DEBTS
READ
ENDIF
IF UPPER(RP) =  "N"
CLEA
SET FORMAT TO DEBTS
READ
ENDIF
ENDCASE
ENDIF
@14,20 SAY "INPUT CURRENT  RECORDS ---- PRESS ENTER KEY!"
STORE "  " TO YN
CLEA
@21,20
WAIT "TO MODIFY MORE RECORDS (Y/N)?:" TO YN
IF UPPER(YN) $ "Y"
CLEA
LOOP
ELSE
CLEA
REPLA D_DATE WITH M_D_DATE, ITEM_QTY WITH M_IQTY
REPLA ITEM_NAME WITH M_INAME, ITEM_RATE  WITH M_IRATE
REPLA DEB_BAL WITH M_DBAL, GTOT_AMT WITH M_GTAMT
REPLA FILE_NUMB WITH M_FILE_NO, S_NAME WITH M_SNAME
REPLA S_ADD WITH M_SADD, INV_NO WITH M_INVNO
REPLA ADV_PAYT WITH M_AMT_PAID
CLOSE ALL
EXIT
ENDIF
ENDDO
SET TALK ON
SET CLOCK OFF
RETURN
```

```
PROCEDURE  M_CREDIT
*THIS SUBPROGRAM IS FOR CREDIT RECORD MODIFICATION
CLEA
SET DATE BRITISH
SET TALK OFF
@4,5 CLEA TO 18,78
@4,5 TO 18,78
SET ECHO OFF
SET COLOR TO R+/B+
STORE SPACE(20) TO M_CNAME,M_INAME
M_CADD = SPACE(25)
M_ICODE = SPACE(9)
M_C_DATE = CTOD(TODAY)
RP = SPACE(1)
STORE 0 TO M_G_AMT,M_IQTYS,M_C_AMT,M_SRATE,M_CCODE,M_AMT_PAID
@12,5 CLEA TO 20,70
DO WHILE .T.
GO TOP
@9,10 TO 13,74
USE CREDIT.DBF
@11,20 SAY "ENTER ITEM CODE:" GET M_CCODE PICT "999"
READ
SET ECHO OFF
SET COLOR TO  G/B+
LOCATE FOR CUST_CODE = M_CCODE
IF .NOT. FOUND()
CLEA
AS = " "
@5,10 SAY  "RECORD NOT FOUND ----------------!!!"
@11,10 SAY "WRONG ITEM CODE ENTERED ---------!!!"
@17,10
WAIT "PRESS ANY KEY TO CONTINUE -------------!!!"
@20,25 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET AS PICT "Y"
READ
IF UPPER(AS) = "Y"
CLEA
LOOP
ELSE
CLEA
CLOSE DATA
EXIT
ENDIF
ENDIF
IF FOUND()
CLEA
SET COLOR TO RB+
@13,5 TO 15,70 DOUBL
@14,10 SAY "TO VIEW THE RECORDS (Y/N)?:" GET RP PICT "Y"
READ
IF RP = "Y"
CLEA
SET ECHO OFF
SET COLOR TO RB+/G
@12,10 TO 16,70 DOUBL
LIST ALL
CLEA
SET FORMAT TO CREDIT
READ
ENDIF
IF RP = "N"
@14,15 CLEA TO 24,78
SET FORMAT TO CREDIT
READ
```

```
ENDIF
SET ECHO  OFF
@14,20 SAY "INPUT CURRENT  RECORDS  ---- PRESS ENTER KEY!"
STORE "   " TO YN
CLEA
@21,20
WAIT "TO MODIFY MORE RECORDS (Y/N)?:" TO YN
IF UPPER(YN) $ "Y"
CLEA
LOOP
ELSE
CLEA
REPLA CRED_DATE WITH M_C_DATE, ITEM_QTY WITH M_IQTYS
REPLA ITEM_NAME WITH M_INAME, ITEM_RATE  WITH M_SRATE
REPLA C_BAL WITH M_C_AMT, GTOT_AMT WITH M_G_TAMT
REPLA AMT_DEPO WITH M_AMT_PAID, C_NAME WITH M_CNAME
REPLA C_ADD WITH M_CADD, CUST_CODE WITH M_CCODE
REPLA ITEM_CODE WITH M_ICODE
CLOSE ALL
EXIT
ENDIF
ENDIF
ENDDO
SET TALK ON
SET CLOCK OFF
RETURN


PROCEDURE M_INCOME
*THIS SUBPROGRAM MODIFYS INCOME DATA FILE.
CLEA
SET CLOCK ON
SET COLOR TO B/W
SET TALK OFF
@3,5 CLEA TO 20,70
SET DATE BRITISH
STORE " " TO ANS
@3,10 TO 21,78
DO WHILE .T.
@2,5 CLEA TO 21,78
M_I_DATE = CTOD(TODAY)
M_DESCPT = SPACE(25)
M_FILE_NO =  SPACE(3)
STORE 0 TO M_AMOUNT,M_GD_TAMT
USE INCOME.DBF
@4,6 TO 18,77 DOUBL
@14,10 SAY "ENTER DATE INCOME:" GET INCOM_DATE PICT "(DD/MM/YY)"
READ
SET ECHO OFF
LOCATE  FOR INCOM_DATE = M_I_DATE
IF .NOT.  FOUND()
CLEA
DS = " "
@5,15  SAY "WRONG INCOME DATE ENTERED --------------!!!"
@11,15 SAY "RECORD NOT FOUND ----------------------!!!"
@17,20
WAIT "THANKS! PRESS ANY KEY TO CONTINUE -------------!!!"
@20,20 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET DS PICT "Y"
READ
IF DS = "Y"
CLEA
LOOP
ENDIF
```

```
IF DS = "N"
CLEA
CLOSE DATA
EXIT
ENDIF
ENDIF
IF FOUND()
SET COLOR TO G+/W
@14,15 SAY "TO VIEW RECORD(Y/N)?:"  GET ANS PICT "Y"
READ
IF UPPER(ANS) = 'Y'
@4,6 CLEA TO  18,77
@4,6 TO 20,70 DOUBL
LIST ALL
CLEA
SET FORMAT TO INCOME
READ
ENDIF
IF UPPER(ANS)  = 'N'
CLEA
SET FORMAT TO INCOME
READ
ENDIF
SET ECHO OFF
STORE " " TO DS
CLEA
*WE USE REPLACE COMMAND TO REPLACE  THE RECORDED INFORMATION.
REPLA INCOM_DES WITH M_DESCPT, FILE_NUMB  WITH M_FILE_NO
REPLA TOT_AMT WITH M_AMOUNT, G_TOTAMT WITH M_GD_TAMT
REPLA S_NO WITH M_SNO
@22,15 SAY "ADD MORE RECORDS (Y/N)?:" GET ANS PICT "Y"
READ
IF UPPER(ANS) = "Y"
CLEA
LOOP
ELSE
CLEA
CLOSE DATABASE
EXIT
ENDIF
ENDIF
ENDDO
SET TALK ON
SET CLOCK OFF
RETURN


PROCEDURE M_CUSTOMER
*THIS SUBPROGRAM IS MEANT FOR DATA MODIFICATION.
CLEA
SET TALK OFF
SET COLOR TO RG+/B+
SET CLOCK ON
DS = " "
M_CUST_CD = SPACE(4)
M_CUST_ADD = SPACE(25)
STORE SPACE(10) TO M_LASTNM,M_FIRSTNM,M_OTHERNM
@2,5 CLEA TO 20,72
DO WHILE .T.
GO TOP
@3,5 CLEA TO 20,70
USE CUSTOMER.DBF
@15,20 SAY "ENTER CUSTOMER'S CODE:" GET M_CUST_CODE
```

```
READ
SET ECHO OFF
LOCATE FOR CUST_CODE = M_CUST_CODE
IF FOUND()
SET COLOR TO B+/W+
@14,10 SAY "TO VIEW RECORDS (Y/N)?:" GET DS PICT "Y"
READ
IF UPPER(DS) = 'Y'
@4,6 CLEA TO 17,77
LIST ALL
CLEA
SET FORMAT TO CUSTOMER
READ
ENDIF
IF UPPER(DS) = 'N'
SET FORMAT TO CUSTOMER
READ
ENDIF
SET ECHO OFF
IF .NOT. FOUND()
CLEA
YN  = " "
@5,10 SAY "WRONG CUSTOMERS CODE ENTERED ------------!!!"
@11,10  SAY "RECORDS NOT FOUND ---------------------!!!"
@17,15
WAIT "THANKS! PRESS A KEY TO CONTINUE  --------------!!!"
@20,25 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET YN PICT "Y"
READ
IF UPPER(YN) = "Y"
CLEA
LOOP
ENDIF
IF UPPER(YN) = "N"
CLEA
CLOSE ALL
EXIT
ENDIF
ENDIF
STORE SPACE(1) TO RP
CLEA
SET ECHO OFF
SET COLOR TO B+
@12,5 TO 16,70 DOUBL
@14,15 SAY "TO MODIFY MORE  RECORDS(Y/N)?:" GET RP PICT "Y"
READ
IF RP = 'Y'
CLEA
LOOP
ELSE
CLEA
REPLA CUST_CODE WITH M_CUST_CD, CUST_ADD WITH M_CUST_ADD
REPLA LAST_NM WITH M_LASTNM, FIRST_NM WITH  M_FIRSTNM
REPLA OTHER_NM WITH M_OTHERNM
CLEA
CLOSE ALL
EXIT
ENDIF
ENDIF
ENDD
SET TALK ON
SET CLOCK  OFF
RETURN
```

```
**** RECDEL.PRG ****
*THIS PROGRAM IS A SUBPROGRAM TO RECORDS FROM THE MAIN PROGRAM.
CLEA
SET COLOR TO W+/B+
SET CLOCK ON
CLEA
SET TALK OFF
SET SCORE OFF
SET ECHO OFF
@9,2 CLEA TO 23,70
@9,2 TO 20,70 DOUBL COLOR N+
SET COLOR TO W+/RB
FNAME = " "
SET DATE BRITISH
@11,20 SAY "DATA DELETE ROUTINE"
@12,20 TO 12,38 DOUBL
Q = 0
IF  Q = INKEY(0.6)
CLEA
SKIP
ENDIF
DO WHILE .T.
@7,5 CLEA TO 18,70
* SET THE FILES IN A LOOP FOR NECESSARY OPERATION.
SET ECHO OFF
@7,5 TO 18,70 DOUBL
SET COLOR TO GR+/B
@8,20 SAY "A                  PURCHASE"
@9,20 SAY "B                  SALES"
@10,20 SAY "C                 DEBTS"
@11,20 SAY "D                 CREDIT"
@12,20 SAY "E                 ITEM".
@13,20 SAY "F                 EXPENSES"
@14,20 SAY "G                 INCOME"
@15,20 SAY "H                 INVENTORY"
@16,20 SAY "I                 CUSTOMER"
@17,20 SAY "J                 BANK"
@20,10 CLEA TO 22,65
@20,10 TO 22,65 DOUBL
@21,20 SAY "ENTER FILENAME:" GET FNAME PICT "A"
READ
*TO ANALYSE INPUT FOR THE FILENAME
DO WHILE .T.
SELECT 1
USE PURCHASE.DBF
SELECT 2
USE SALES.DBF
SELECT 3
USE DEBTS.DBF
SELECT 4
USE CREDIT.DBF
SELECT 5
USE EXPENSES.DBF
SELECT 6
USE INCOME.DBF
SELECT 7
USE STOCK.DBF
SELECT 8
USE CUSTOMER.DBF
SELECT 9
USE BANK.DBF
DO CASE
CASE FNAME = "A"
```

```
USE PURCHASE.DBF
LOCATE FOR S_NO = M_SNO
CASE FNAME = "B"
USE SALES.DBF
LOCATE FOR S_NO = M_SNO
CASE FNAME = "C"
USE DEBTS.DBF
LOCATE FOR S_CODE = M_SCODE
CASE FNAME = "D"
USE CREDIT.DBF
LOCATE FOR CUST_CODE = M_CCODE
CASE FNAME = "E"
USE ITEM.DBF
LOCATE FOR S_NO = M_SNO
CASE FNAME = "F"
USE EXPENSES.DBF
LOCATE FOR S_NO = M_SNO
CASE FNAME = "G"
USE INCOME.DBF
LOCATE FOR S_NO = M_SNO
CASE FNAME = "H"
USE STOCK.DBF
LOCATE FOR STOCK_CD =  M_STCODE
CASE FNAME = "I"
USE CUSTOMER.DBF
LOCATE FOR CUST_CODE = M_CCODE
CASE FNAME = "J"
USE BANK.DBF
LOCATE FOR S_NO = M_SNO
OTHERWISE
CLEA
@13,10 CLEA TO 15,78
SET COLOR TO RB
@13,10 TO 15,75
CH = SPACE(1)
@14,15 SAY "WRONG FILENAME ----------------- TRY AGAIN !!!"
@17,20
WAIT "PRESS ANY KEY TO CONTINUE -----------------------!!!"
CLEA
SKIP
ENDCASE
IF FOUND()
CLEA
@15,15 SAY "TO VIEW RECORD BEFORE DELETING(Y/N)?:" GET CH PICT "!"
READ
IF UPPER(CH)  = "Y"
SET ECHO OFF
SET COLOR TO R
AN = SPACE(1)
CLEA
@2,3
LIST ALL
Q = INKEY(0.2)
@20,15
WAIT "ARE THESE THE RECORDS YOU WANT TO DELETE(Y/N)?:" TO  AN
IF UPPER(AN) = 'Y'
DELETE
PACK
ENDIF
IF UPPER(AN) = 'N'
CLEA
LOOP
ENDIF
```

```
ELSE
DELETE
ZAP
ENDIF
RP = "!"
@10,2 CLEA TO 20,75
@12,5 CLEA TO 16,72
@10,2 TO 20,75 DOUBL
@12,5 TO 16,72 DOUBL
SET COLOR TO  R+/G
SET ECHO OFF
@14,10 SAY "TO DELETE MORE RECORDS (Y/N)?:"
SET COLOR TO W+/B+,R+
@13,45 GET RP PICT "Y"
READ
*USER INPUT ANALYSIS
IF UPPER(RP) = "Y"
CLEA
LOOP
ENDI
IF UPPER(RP) = "N"
CLEA
EXIT
ENDIF
ENDIF
IF .NOT. FOUND()
CLEA
YN = SPACE(1)
@11,5 TO 16,70 DOUBL
@12,15 SAY "WRONG FILE NAME ENTERED ---------------------!!!"
@13,15 SAY "RECORD NOT FOUND ----------------------------!!!"
@14,15 SAY "PLEASE ENTER CORRECT FILE NAME --------------!!!"
@15,25
WAIT "PRESS A KEY TO CONTINUE  WORK ----------------------!!!"
@20,25 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET YN PICT "Y"
READ
IF YN = "Y"
CLEA
LOOP
ENDIF
IF YN = "N"
CLEA
CLOSE DATA
QUIT
ENDIF
ENDIF
ENDDO
ENDDO
SET CLOCK OFF
SET TALK ON
RETURN
                      **** LOGO.PRG ****
        * THIS PROGRAM DESIGNS THE LOGO OF THE SOFTWARE
        CLEA
        SET ECHO OFF
        CLEA
        SET CLOCK ON
        SET COLOR TO RB/W+
        @2,1 CLEA TO 24,78
        @2,1 TO 24,78 DOUBL
        @3,1 CLEA TO 23,1
        SET COLOR TO R/G,W/R+
        @3,1 TO 23,1 DOUBL
```

```
    @3,77 TO 23,77 DOUBL
    J_M = 0
    SET COLOR TO BG, R/W
    DO WHILE .T.
          SET ECHO OFF
          @4,28 SAY "WELCOME!  WELCOME!!  WELCOME!!!"
           @5,45 SAY "TO"
           @6,33 SAY "TONY'S STORE ENTERPRISES"
           SEC = 0
           IF SEC = INKEY(0.1)
              WAIT
              @7,5 CLEA TO 11,78
              SET COLOR TO R+/B+
              KT = 0
            * DESIGNING OF THE LOGO --------------!!!
            * PLEASE STAY TUNE ----------------!!!
              @7,05 CLEA TO 11,78
              SET COLOR TO G/W
              SET ECHO OFF
            @15,20 SAY "
            @16,20 SAY "
            @17,20 SAY "
            @18,20 SAY "
            @19,20 SAY "
```

```
            KT = KT + 1
            SEC = INKEY(0.1)
    @21,20
    WAIT "PRESS A KEY TO CONTINUE -------------------------!!!"

    @13,5 CLEA TO 24,70
    SET COLOR TO BG/GR
    SET ECHO OFF
    @13,10 TO 13,39 DOUBL COLOR G
    @13,41 TO 13,68 DOUBL COLOR G
    @14,12 TO 14,39
    @14,41 TO 14,66
    @15,14 TO 15,39 DOUBL COLOR W
    @15,41 TO 15,64 DOUBL COLOR W
    @16,16 TO 16,39
    @16,41 TO 16,62
    @17,18 TO 17,39 DOUBL COLOR G
    @17,41 TO 17,60 DOUBL COLOR G
    @18,20 TO 18,39 DOUBL COLOR G
    @18,41 TO 18,58 DOUBL COLOR G
    @19,22 TO 19,39
    @19,41 TO 19,56
    @20,24 TO 20,39 DOUBL COLOR W
    @20,41 TO 20,54 DOUBL COLOR W
    @21,26 TO 21,39
    @21,41 TO 21,52
    @23,33 SAY "BIDA, NIGER ST."
    SEC = 0
    RS = SPACE(1)
    CLEA
@15,20 SAY "DO YOU WANT TO CONTINUE (Y/N)?:" GET RS PICT "Y"
    READ
    IF RS = 'Y'
    CLEA
    LOOP
    ELSE
    CLEA
    CLOSE ALL
    EXIT
```

```
ENDIF
ENDIF
ENDDO
SET SCORE ON
SET CONSOL ON
SET CLOCK OFF
RETURN

**** MAINMENU.PRG ****
* THIS PROGRAM IS CALLED MAIN MENU PROGRAM
* IT SETS THE MAIN OPERATIONS IN A DESIGN MANNER
SET TALK OFF
   CLEA
SET STAT OFF
KOUNT = 0
CLEA
SET CLOCK ON
SET COLOR TO GR/B+
@1,5 CLEA TO 5,78
SET COLOR TO R/B
DO WHILE .T.
        GO TOP
        CLEA
        SET COLOR TO N+/RB
        @1,5 TO 6,78 DOUBL COLOR W
       SET COLOR TO R+/G+
      @2,22 SAY "COMPUTERIZATION OF WAREHOUSE STOCK INVENTORY"
       @3,28 SAY "AND ANALYSIS PROCEDURES"
       @4,33 SAY "VIS - A - VIS"
       @5,27 SAY "TONY STORE ENTERPRISES"
       SEC = 0
       IF SEC = INKEY(0.1)
                WAIT "PRESS ANY KEY TO CONTINUE"
                CLEA
                SET COLOR TO BR
                @8,30 SAY "M A I N   M E N U"
                @ROW() + 1,10 SAY "================="
                @ROW() + 1,10 SAY "TASK CODE"
                @ROW(),40 SAY "TASK OPERATIONS"
                STORE "X" TO CHOIC
            @14,14 SAY "A" + SPACE(22) + "ADDITION"
            @15,14 SAY "B" + SPACE(22) + "MODIFICATION"
            @16,14 SAY "C" + SPACE(22) + "DELETION"
            @17,14 SAY "D" + SPACE(22) + "INVENTORY"
            @18,14 SAY "E" + SPACE(22) + "ANAL_REP"
            @19,14 SAY "F" + SPACE(22) + "QUIT"
            @20,10 CLEA TO 22,75
            @21,10 TO 22,75 PANEL COLOR G+
            @20,15 SAY "ENTER CODE NUMBER :" GET CHOIC PICT "X"
                READ
        ENDIF
SET COLOR TO GR
DO CASE
        CASE UPPER (CHOIC) = "A"
                CLEA
                SET COLOR TO B/W
                DO REC_ADD
        CASE UPPER (CHOIC) = "B"
                CLEA
                SET COLOR TO BG/R,W/R
                DO REC_MODI
        CASE UPPER (CHOIC) = "C"
                CLEA
```

```
                        SET COLOR TO R
                        DO REC_DEL
                CASE UPPER (CHOIC) = "D"
                        CLEA
                        SET COLOR TO GB/RG
                        DO INVENTORY
                CASE UPPER (CHOIC) = "E"
                        CLEA
                        SET COLOR TO G+/R,B+
                        DO ANAL_REPT
                OTHERWISE
                        STORE " " TO ANS
                        @18,5 CLEA TO 20,70
                        @7,10 TO 16,70 DOUBL
        ENDCASE
        @18,5 CLEA TO 20,76
        SET COLOR TO N
        @18,10 TO 20,70 PANEL COLOR R
        @19,15 SAY "DO YOU WANT TO QUIT DBASE (Y/N)?:"
        @19,50 GET ANS PICT "Y"
        READ
        * TO ANALYSE THE USER'S RESPONSE
        KOUNT = KOUNT + 1
        IF UPPER (ANS) = "N"
                CLEA
                SET COLOR TO GR
                LOOP
        ENDIF
        IF UPPER (ANS) = "Y"
                CLEA
                SET COLOR TO N
                SKIP
                CLOSE ALL
                SET TALK ON
                SET SCORE ON
                SET STAT ON
                EXIT
        ENDIF
        ENDDO
        RETURN


* PROCEDURE A_PURCHASE
*SUBPROGRAM FOR DATA ADDITION IN A_PURCHASE
CLEA
SET DATE BRITISH
@2,5 CLEA TO 20,78
@3,10 CLEA TO 22,70
SET ECHO OFF
SET COLOR TO B+/G
STORE SPACE(3) TO M_FILE_NO,M_STOCK_CD,M_SCODE
STORE SPACE(10) TO M_ITEM2,M_ITEM3,M_ITEM4
STORE SPACE(15) TO M_ITEM1
STORE SPACE(20) TO M_SNAME,M_INAME
M_SADD = SPACE(25)
M_ICODE = SPACE(9)
M_PDATE = CTOD(TODAY)
RP = SPACE(1)
STORE 0 TO M_INVNO,M_SPARTC,M_ITMAC,M_IQTY,M_KMACHC,M_IRATE,MTOT_AMT
STORE 0 TO M_GD_TOTAL,M_SNO
@2,5 CLEA TO 20,70
DO WHILE .T.
@3,10 TO 22,74
```

```
USE PURCHASE INDEX S_NO
USE ITEM INDEX SPART_CD
DO AZZ_16
*RETURN OF THE RESULT FROM THE SUBPROGRAM
@2,5 TO 20,78 DOUBL
@3,10 TO 22,70 DOUBL
SET ECHO OFF
SET COLOR TO W/R
SP = SPACE(2)
R = 4
@4,6 SAY M_ICODE + LTRIM(STR(M_ICODE)) + SP + M_INAME PICT "@!"
IF .NOT. EOF()
@3,10 CLEA TO 22,70
@23,7
WAIT "PRESS A KEY TO CONTINUE -------------------!!!"
CLEA
R = 4
ENDIF
@4,6 SAY M_IRATE(AMT,6,2) + SP + M_IQTY + SP + MTOT_AMT(MTOT_AMT,8,2)
*ENDIF
*ENDDO
*RETURN
IF .NOT. EOF()
@23,20
WAIT "PRESS ANY TO CONTINUE ----!!!"
@3,10 CLEA TO 22,70
R = 4
ENDIF
@4,15 SAY M_GD_TOTAL (AMT,10,2)
*WE USE APPEND BLANK TO CREATE SPACE FOR THE DATA TO GO INTO.
APPEND BLANK
SET FORMAT TO PURCHASE
READ
M_SCODE = S_CODE
M_IRATE = ITEM_RATE
MTOT_AMT = TOTAL_AMT
M_FILE_NO = FILE_NUMB
M_SNAME = S_NAME
M_SADD = S_ADD
M_PDATE = PUR_DATE
M_INVNO = INV_NUMB
M_GD_TOTAL = GTOT_AMT
M_ICODE = ITEM_CODE
M_INAME = ITEM_NAME
M_STOCK_CD = STOCK_CD
M_SNO = S_NO
*TO USE REPLACE COMMAND TO REPLACE ALL THE ITEMS INTO THE DBASE FILE.
REPLA ITEM_RATE WITH M_IRATE, TOTAL_AMT WITH MTOT_AMT
REPLA FILE_NUMB WITH M_FILE_NO, S_NAME WITH M_SNAME
REPLA S_ADD WITH M_SADD, PUR_DATE WITH M_PDATE
REPLA INV_NUMB WITH M_INVNO, GTOT_AMT WITH M_GD_TOTAL
REPLA ITEM_CODE WITH M_ICODE, ITEM_NAME WITH  M_INAME
REPLA STOCK_CD WITH M_STOCK_CD, S_CODE WITH M_SCODE
REPLA ITEM_QTY WITH M_IQTY, S_NO WITH M_SNO
@22,10 CLEA TO 22,78
@22,15 SAY "IS PURCHASE MADE ON CREDIT (Y/N)?:" GET RP PICT "Y"
READ
IF RP = "Y"
DO AZZ_8
USE DEBTS INDEX S_CODE
ELSE
CLEA
@23,20
```

```
WAIT "PURCHASE WAS MADE ON CASH -------------------------------!!!"
CLEA
EXIT
ENDI
AN = ""
@23,15
WAIT "ADD MORE RECORDS (Y/N)?:" TO AN
*ANALYSIS OF THE USER'S INPUT AND END OF THE SUBPROGRAM
IF UPPER(AN) = "Y"
CLEA
LOOP
ENDIF
IF UPPER(AN) = "N"
CLEA
CLOSE DATA
EXIT
ENDIF
ENDD
CLOSE ALL
RETURN


* PROCEDURE ADD_DEBT
*THIS PROGRAM IS USED TO RECORD THE DEBT OF THE BUSINESS OF
*TONY STORE ENTERPRISES.
CLEA
SET TALK OFF
@2,2 CLEA TO 22,70
@3,5 CLEA TO 20,70
SET COLOR TO R
SET DATE BRITISH
CH = " "
M_D_DATE  = CTOD(TODAY)
M_SNAME   = SPACE(20)
M_SADD = SPACE(25)
M_ICODE   = SPACE(9)
M_INAME =  SPACE(20)
STORE 0 TO M_SCODE,M_IQTYS,M_IRATE,M_AMT_PAID,M_GD_TOTAL,M_DBAL
STORE 0 TO M_GDT_AMT,M_GT_AMT
@3,5 CLEA TO 20,78
DO WHILE .NOT. EOF()
GO TOP
CLEA
USE DEBTS INDEX S_CODE
*PUBLI M_SNAME,M_SADD,M_ICODE,M_INAME,M_IRATE,M_GD_TOTAL
APPEND BLANK
SET FORMAT TO DEBTS
READ
*TRANSFER OF FIELD CONTENTS INTO MEMORY  VARIABLES
M_D_DATE =  D_DATE
M_SNAME =  S_NAME
M_SADD  = S_ADD
M_SCODE = S_CODE
M_ICODE = ITEM_CODE
M_INAME =  ITEM_NAME
M_IQTYD = DEB_IQTY
M_IRATE  = ITEM_RATE
M_AMT_PAID  = ADV_PAYT
M_GTAMT  = GTOT_AMT
M_DBAL = D_BAL
*CALCULATION OF THE BALANCE, AMOUNT TO BE PAID.
SET COLOR TO RB/R+
M_GTAMT = M_IQTYD * M_IRATE
```

```
M_DBAL = M_GTAMT - M_AMT_PAID
USE  DEBTS
@10,20 SAY GTOT_AMT(GTOT_AMT,10,2)
M_GDT_AMT = GDT_AMT
M_GT_AMT = M_GDT_AMT + M_DBAL
@13,15 SAY "TOTAL AMT OF PURCHASE:" + M_GTAMT(M_GTAMT,6,2)
@16,15 SAY "TOTAL AMT TO  BALANCE:" + M_GT_AMT(M_GT_AMT,8,2)
@18,10
WAIT "PRESS ANY KEY TO CONTINUE ---------------------------------!!!"
CLEA
*USING REPLACE COMMAND  TO REPLACE THE INPUTTED RECORDS .
REPLA D_DATE WITH  M_D_DATE, S_NAME WITH M_SNAME
REPLA  S_ADD WITH M_SADD, S_CODE WITH  M_SCODE
REPLA ITEM_CODE WITH M_ICODE, ITEM_NAME WITH  M_INAME
REPLA DEB_IQTY  WITH M_IQTYD, ITEM_RATE WITH M_IRATE
REPLA ADV_PAYT WITH M_AMT_PAID,  GTOT_AMT  WITH M_GTAMT
REPLA D_BAL WITH M_DBAL
@23,10
WAIT  "ADD MORE RECORDS (Y/N)?:"  TO CH
IF UPPER(CH) =  "Y"
CLEA
LOOP
ENDIF
IF UPPER(CH)  = "N"
CLEA
SET COLOR TO B+
EXIT
ENDIF
ENDD
CLOSE ALL
SET TALK ON
RETURN


PROCEDURE CAL_CASH
*THIS SUBPROGRAM IS USED FOR CALCULATION  OF MONIES LODGED INTO  BANK
CLEA
SET TALK OFF
@13,10 CLEA TO 20,70
SET ECHO OFF
M_LODGE = M_CHEQUE
DO WHILE .T.
AMT1 = M_CD50  * 50
AMT2 = M_CD20 * 20
AMT3 = M_CD10 * 10
AMT4 = M_CD5 *  5
AMT5 = M_CDC1 * 1
AMT6 = M_CDC_50 * 0.50
AMT7 = M_CDC_10 * 0.10
AMT8 = M_CDC_5 * 0.05
AMT9 = M_CDC_1 * 0.01
MGT_AMT =  AMT1 + AMT2 + AMT3 + AMT4 + AMT5 + AMT6 + AMT7 + AMT8 + AMT9
MC_BANK = 0
USE BANK.DBF
@13,5 TO 20,70 DOUBL
@14,10 GET CASHBANK
MGTT_AMT =  0
USE INCOME.DBF
@16,10 GET G_TOTAMT
MGTT_AMT  =  G_TOTAMT
M_CHHAND  = MGTT_AMT - MC_BANK
@12,5 CLEA TO 20,70
@12,5  TO  20,70 DOUBL
```

```
@14.,10 SAY MGT_AMT(MGT_AMT,8,2)
@16,10 SAY M_CHHAND(M_CHHAND,10,2)
Q = INKEY(0.3)
WAIT
CLEA
CLOSE  ALL
EXIT
ENDDO
SET TALK ON
RETURN
```

APPENDIX C

BUSINESS TRANSACTIONAL ANALYSIS

TRADE ACCOUNT ANALYSIS

SET-UP ACCOUNT	BUSINESS TRANSACTIONS
OPENING STOCK: 100,000.00	SALES: 250,000.00
PURCHASES: 80,000.00	LESS DELIVER INWD: 7,000.00
CARRIAGE INWDS: 5,000:00	
COST OF GDS AVAIL: 185,000.00	
LESS CLOSING STK: 20,000.00	
COST OF GDS SOLD: 165,000.00	
GROSS PROFIT: 78,000.00	
C/DOWN: 243,000.00	243,000.00

PROFIT AND LOSS ACCOUNT

EXPENSES	GROSS PROFIT
RATE RENT: 4,000.00	GROSS PROFIT B/D: 78,000.00
WATER BILL: 1,000.00	
ELECTRIC BILL: 950.00	
CARRIAGE OUTWD: 550.00	
BAD DEBT: 2,500.00	
NET PROFIT: 69,000.00	
C/DOWN: 78,000.00	78,000.00

BALANCE SHEET ACCOUNT OF TRADE

INCOME ACCOUNT	**FIXED ASSETS**
CAPITAL ACC: 150,000.00	PREMISES: 25,000.00
NET PROFIT: 69,000.00	LAND & BUILD.: 50,000.00
	MISCELLANEOUS: 5,000.00
219,000.00	
	80,000.00
CURRENT LIABILITIES	**CURRENT ASSETS**
CREDITS: 10,000.00	DEBTS: 20,000.00
ACCRUALS: 5,000.00	CASH AT BANK: 64,000.00
	PREPAYMENT: 10,000.00
	CASH AT HAND: 10,000.00
TOTAL: 15,000.00	STOCK: 20,000.00
B/DOWN: 204,000.00	204,000.00