RELEVANCE OF COMPUTERS IN THE MILITARY

BY

AHMED TIJJANI IBRAHIM
PGD/MCS/98/99/814

DEPARTMENT OF MATHEMATICS / COMPUTER SCIENCE FEDERAL UNIVERSITY OF TECHNOLOGY MINNA

RELEVANCE OF COMPUTERS IN THE MILITARY

BY

AHMED TIJJANI IBRAHIM
PGD/MCS/98/99/814

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

APPROVAL PAGE

SUPERVISOR…………………….. DATE:…………………………
DR. S.A. REJU

H.O.D………….…………… DATE:………………………
MR. L.N. EZEAKO

EXTERNAL EXAMINER……………….. DATE:………………………

CERTIFICATION

I certify that this project was carried out by Ahmed Tijjani Ibrahim a Post Graduate Diploma Student of the Department of Mathematics /Computer Science.

DEDICATION

I dedicate this project work to my late friend Officer Cadet Umaru Ibrahim Sabo (48 RC) my heart, my love and prayers go out to him.

ACKNOWLEDGEMENT

In writing this project, I wish to first and foremost give thanks, honour and praise to Almighty Allah for my birth and everything that has followed.

My sincere thanks goes to my supervisor, Dr. S.A. Reju and my lecturers, Dr. Y. Aiyesimi, Prince R.O. Badmus, Mallam Audu, Mr. Peter Emmanuel, for their patience and guidance.

I would like to specially thank my mum, Mrs. P.A. Ibrahim for being my "river in the desert", much love to my dad, Mallam S.T. Ibrahim, my uncle, Maj. Y.A. Ibrahim and Lt. Col Apochi O. Suleiman for their wisdom and warmth. I am still working to make those lessons pay off. To my brother Ayuba, thanks for holding the forth while I was away. To my sisters, Rabi, Helen, and Ene (Ramatu), thanks for keeping the boat steady. I love you all.

Special thanks to Mr. A.U. Ejembi, Hajiya Hadiza Yakubu, Hajiya Ladi Lawal Aliyu, Engr. & Mrs. Ishiaku Barau, Nanpon & Linda Bala Gambo, Mr. Alex Kato Ba, Mr. Waziri, Col I.D. Bello, Col. I.N Odagbali, Col B.T Obasa, Col. M. said, Lt col S.f. Obidu, Lt. Col. P.G. Ogah, Maj M.I.U Adeka, Maj. O.O. Adeleke, Maj. A.A. Kurah, Dr. Ameh, Mr. M.J. Oije, Mr. M. Oguche, Mr. A.G. shall Holma, Mr. A.A. Anyiloibi, Mrs. E.A. Ademu, Col. Mrs. Odeh, Hajiya Gambo Sawaba, Martins Ekyo, Flt Lt. Aldris, Capt. Y. Yahaya, Lt. I.M. Lafal, Fgottr F.O. Ankeli, Lt. I.N. Garba, Mallam Abdu Usman, Lt. Buhari Madaki, John Ademu, Yusuf Hassan, Comfort Musa, Christy Ishaya.

I would also like to recognize the efforts of my friends, Moh'd Kazir, Aminu Yakubu, Lt. A. Suleiman, Omopariola (Omo), O.O. Diya, B.M. Sule, Oloyede, Y. Habila, A.M. Auwalu, Z. Ibiyeye, KenZebere, Joshua Gambo, Aisha Aliyu, Hauwa Moh'd, Abiodun Adisa, Anglin, Z. Bala, Uyo, Koni.

The following people made my stay in Minna worthwhile, Aminu Ahmed, Alhaji Yakubu Sallau, Bala Bigss, Amina Ndatsu, Ali J. Chitumu Danjuma, Abubakar, S.A. Kolo, Idris Dasuki, Hamza Nasco, Francis. C. Nnona, Toyin

Apata, Ohi, Alice S. Manvong, Henry Bitachi, Sharafa (Bigg Daddy), Chimex, Rotex, Kayode, Chidi. I say thank you for your creative inputs.

Finally, I want to give shout outs to my crew, Adamu Aliyu, Alex Igoji, Moreno, Kayode Yusuf, Seth, P.J. Gadzama, Condy, Ali Moh'd.

To every one I had the good fortune of working with to every one that I forgot to mention, I say thank you for what you have been to me.

Ahmed
2000

SCOPE AND LIMITATIONS

However, in writing this project I have tried to limit my study to the Nigerian Armed Forces with particular reference to Nigerian Army.

I have also given relevant examples of wars that took place all over the world.

Attempts were made to describe how we have been evaluating our past, examining the present and seeking discern indicators trends that could guide us into future.

ABSTRACT

The purpose of this project is to examine the importance of computers in the military through technology, It is penitent for our armed forces to keep barest [illegible] with the ever changing war scenario, not to promote war, but to have the technological know-how to adequately repel possible aggressors

TABLE OF CONTENTS

CHAPTER ONE

GENERAL INTRODUCTION

Modern computers form a class of electronic machines for the processing of information. They have become such a versatile tool that they are so readily applicable in virtually all areas of human activities. They also accepted as one of man's most significant technological inventions carrying out instructions at the speed of the order of nano second to the dictates and controls of man. It is worth mentioning that in the last fifty years, no other machine in human endeavour has received such a quick response and popular acceptance from the society as electronic computer.

The effort of computers on the conduct of strategic military operations have been so dramatic that no commander today can afford to remain unaware of what the computer can do to improve his command and control capabilities in military operations and general administration of is command.

In the Nigerian Army (NA), organization can be defined as the means whereby the corporate arms and services are designed, developed, deployed and directed so that they can be relied upon for rapid and concerted action when required. Also management, be it operational or administrative can be defined under the essential principles of forecasting, planning and directing. It is very important to understand that organizational attempt one automatically affects the other. In effect they can be termed as mathematical warrants.

In today's global search and control of the Universe, the role of computer has become increasingly significant to our weaponary system in the Nigerian Army, the computer is required as a means of increasing administrative efficiency in payroll. Stock control, planning, surveillance, navigation, surrey, intelligence, gathering, strength documentation, communication, command and control etc.

In the Nigerian Army (NA) computer technology is aimed at gaining tactical, logistical stock control, manpower planning and administrative efficiency. Thus, for the NA to remain efficient as a modern force to be reckoned with in the world, she must harness the computer technology. The NA can evaluate the best possible areas to achieve personal resources in making quick and accurate decisions in the use of weapons and resources available to achieve success in defence of the country. This means that almost everyone in the NA will need to become what may be termed "literate" in the computer context.

1.1 GENERATIONS OF COMPUTER AND TECHNOLOGICAL CHANGES

At present, punching machines, sorters are becoming relics for the museum. The first known machines were built in the U.S.A and Britain during the second World War (WWII). After which computers were developed into the following generations:

1.11 FIRST GENERATION (1951-1958): The war provided needed urge for research into computing machines.

In this generation, the computer uses vacuum tubes, which controls the internal operations of the computer. At the time computers were mostly huge and require cooling system. Magnetic drum was used by computer as a means of storage. Use of punch card was also common at the time. Machine language and symbolic language became the language of the computer.

1.12 SECOND GENERATION COMPUTER (1959-1964): Solid state transistors replaced vacuum tubes the size of computer was greatly reduced. Computers became faster, hard increased storage capacities and do not require cooling system. The magnetic types, disks are supplementary memories invention of the low level languages for computers. (i.e. Assembly language)

1.13 THIRD GENERAFTION (1965-1970): The development of integrated circuit (i.e.) soft ware industry emerged. Manuscripts came in to being. Operating system, greater compatibility of components allowing easier expansion of computers system. Remote terminal (i.e. microcomputers and minicomputers came in to being) development of high level languages i.e. Pascal, Basic and FORTRAN .

1.14 FOURTH GENERATION:

Introduction of large scale integrated circuits technology. User friendly. Development of data recording equipment that capture data an example is optical character recognition. Densely packed chips were developed. Microprocessors which led to the manufacture of home computers usually called microcomputer.

1.15 FIFTH GENERATION: Development of Japanese industrial robots as a distinct generation, this generation is influenced by the advent or artificial intelligence (A I), expert system.

From 1990 the fifth generation computer (Pentiums), were on commercial production in Japan Asian countries and the U.S.A. since then development in the computer industry have favoured the packing of more devices into smaller space. Hence, computer technology is said to be growing in an inverted manner. This trend continued and has already made possible the introduction of smaller brand of computer at affordable prices.

1.2 CLASSIFICATION OF COMPUTERS

There are basically two ways to classify computers, one way in the manner in which data is represented within it. In this way computers can be classified into three types

1.21 TYPES OF COMPUTERS

DIGITAL COMPUTER: Digital due to its function by taking discrete number and performing mathematical calculations on them. They are used in commercial data processing.

ANALOGUE COMPUTER: Measures physical magnitude such as temperature, pressure, speed. Analogue computers are used for scientific and engineering purpose.

1.22 PURPOSE

SPECIAL PURPOSE: Computers designed for a particular job only, to solve problems of a restricted nature. Example in weapon guidance system.

GENARAL PURPOSE: Computers designed to solve a wide varrety of problems are called general purposes machine (GPM). This can adapt to perform particular tasks or solve problems by means of specially written programs.

1.23 CAPACITY

MICRO COMPUETRS: - computers realised on a small number of silicon clips. They are usually the smallest in size by means of memory (data storage facility) capacity.

MINI COMPUETRS: - Middle range computers, between the smallest microcomputer and the biggest main frame computers.

MAINFRAME COMPUTERS: - Large computers having wide range of memory facilities. This computer usually occupies large space than both the min and micro. It is important to note that, nowadays a few microcomputers are out performing minis such as notebook computers.

1.3 ADVANTAGES OF COMPUTERS IN THE MILITARY

In the past, the management techniques employed by the military, has been based on the conventional staff procedure and the operational analysis. However, the awareness of the military today in using automation of data in solving management problem is an innovation of the computer service.

One of the major result of the computer is the ability of processing large volume of data quickly, accurately and economically in the development of data information system. Information systems improve military precision and the ability to make quick and effective decisions. Hence the following will be achieved in the long run.

Better management of personnel records and data both in peacetime and during operations.

More effective communication, command and control between the service and in the units and formations.

Routine jobs information and units will obtain faster result, while accuracy will be of most benefit.

The military have used computer to achieve the following:

1.31 OPERATIONAL

Acquisition of all combat intelligence about the enemy signal processing to enhance rapid, reliable and secure communication.

Acquisition of targets and counter bombardment data solving survey and metrological problems.

Effective command and control at all levels Battle simulation and prosecution.

1.32 SCIENTIFIC

The tactical doctrine of an army is mostly affected with the introduction of modern weapons and equipment. The invention of this new technology is developing rapidly, therefore the need to review the tactical doctrine in the military sphere. Thereby creating opportunities for personnel's to be educated so they can compete favorably with their counterpart overseas.

1.33 ADMINISTRATION

Large amount of paper work is involved in the personnel management and supply system of the military in the post because of the manual system of filling in use then. Much could not be achieved, information contained in records were belated. However, the introduction of computers in the management of the military has facilitated speedy accessibility to statistical information, which are vital for staff action.

CHAPTER TWO

ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS

Today Artificial Intelligence (AI) is increasingly regarded as central in the development of new generation system.

AI is the science of making machines that would do things that will require intelligence if done by men.

By 1960, it was obvious that computers developments would profoundly affect views about human psychology. It was already clear that the new electronic devices could do many of the things that were formerly the sole prerogative of human beings and other intelligent creatures. For instance computers could learn. Store and recall information, solve problems, take decisions and carryout simple reasoning tasks.

Expert systems (ES) are one of the key developments contributing to the fifth generation programme. These are in one clean sense a manifestation of AI. Typically, such systems make inferences following interrogation of a knowledge base. In addition, a degree of probability may be assigned to the conclusion that is dawn. Expert systems, structured around a body of (human) expert knowledge, are intended to assist human deliberations in such fields as medicals diagnosis and geological prospecting. Such systems predate the fifth generation declarations but are clearly relevant to the next generation of computers, it is expected that the fifth generation systems comprising the AI and ES, will among many other tasks be able to provide a conventional advisory facility when working with human beings.

2.1 MILITARY APPLICATIONS

The application of computers in the military can be considered under three principal areas: Operational, Administrative and scientific as it may affect the armed forces.

2.2 OPERATIONAL SYSTEM

Pertaining to information needs, a field commander will require the following:

Means for lively acquisition of a mass of tactical and pertinent enemy information.

Analysis to determine what resources is required to launch and support a given operation for a given period.

Consolidated battle maps consisting of the latest operational events and up to date intelligence report.

Readiness of all data for decision-making.

2.21 INTELLIGENCE REPORTING

Mostly, the systems are used for the swift collection, collation and disemination of the raw materials (battle information) leaving the computers to sort and re-evaluate the analyzed end product, making it instantly available to the commander for decision making. Important data of broader nature are selected by the computers and made available by computers through communication link to the higher commanders at the rear echelons. In addition to these, the computers has a vital role in:

* Interrelation and cover cross checking of intelligence
* Storing data regarding enemy capabilities and limitations
* String intelligence summaries
* Automatic dissemination/information.

2.22 SIGNAL PROCESSING

The simplest system for signal processing in the battle field is the radar system of the surveillance elements of the infantry battalion and reconnssance units. The programme and instruction for this type of target acquisition has an advantage over the aged electromechanical device, it is comparatively uncomplicated and optimized for speed and accuracy, resulting in instantaneous display on the \operation screen. The computer will also play its part in:

Electronic warfare

Crypto analysis

Frequency analysis

Electronic switching in exchange and communication centres.

2.23 WEAPONS SYSTEMS

The combat computers installed to complement the artillery field guns and air defence system, amoured vehicles and other weapon systems are usually for fire control and target acquisition. These computer systems operate with an ultra-light processing speed as opposed to those employed in the signals processing systems. For example, the flight of a shell, its trajectory, impact, effect of explosions, stress on the barrel and other relevant effects can be calculated using the computer systems, during fire support. Computers are use to work out automatically the survey, and metrological data's, locate the target and compute the bearings quadrant elevations to eliminate human errors. Computers therefore solve survey and metrological problems, tracking and guiding of missiles and interceptors on the target. Computers also aid in counter bombardment.

With the introduction of computers in weapon systems, firing table of weapon system are prepared under standard ballistic metrological conditions, because such weapons are expected to be used in different places with different weather conditions in order to suit any given metrological condition. As against those days that artillery guns had problems with achieving first round target. The failure to achieve first round target has some advantages amongst which are loss of suprice in an operation, waste of ammunition, waste of time in pre registration of targets. Before computers were introduced. There had to be a first round target after which adjustment will be made.

2.24 COMMAND AND CONTROL

The expansion of corps and formation in size and scale of resources are becoming very complex. In this area, the command and control system, unlike other computer systems, are not usually taken into battle but designed for a fixed based installation for use at higher command level. It is designed to provide assistance in answering the increasingly complex combat requirement. This system is a combination of sophisticated hardware and software. This computer system will assist the staff officer of all levels in summarizing, comparing, collating, evaluating and recording information.

It is worth mentioning that records of officers and men under command can be kept in files in the computer; and under proper programming instructions. It will be possible to select the best officer or soldier for any assignment, which requires optimum efficiency. While discussing the command and control, it is essential to emphasize the importance and interrelationship of communication in some computer systems. The ability to process vast volume of information without any means of communication link within the chain of processing will render the system impotent. Thus, this computer system requires an effective and reliable communication network for fast transmission of the acquired information between users terminal devices and the central processor. This method is known as the "on line system" on the contrary other data may be processed in batches in a given fixed period, in which event the user does not need to have a fixed communication link with the processor. This is an "off line system".

One of the most modern computer communication net work is the advanced research project agency (ARPA) network set up by the United States, Ministry of Defence. This network links over sixty computers across the entire United States of America and London.

2.25 BATTLE SIMULATION

Simulation is the representation of a local real or hypothetical system by a computer process. Its function is to indicate system performance under various conditions by program performance. Battle condition could be translated into models and the computer would process such models repeatedly in a variety of forms giving different probable states and would display also all possible expectations. The interactions of the variables used to describe the battle situation would also be displayed.

The advantage here is that simple and cheap events could be used to determine hypothetical future interaction, simulation routines using computers have been developed for evaluating damage done to communications in a nuclear war, valuating the effects of a controversial bombing on troops and equipments in non-nuclear attack, accessing the effects of terrain on the coverage of the defensive radars, and in computing the rapidity with which a stack force could be deployed from one country to a foreign theatre.

2.3 SCIENTIFIC SYSTEM

The system embark on the use of computes for research programmes, development of research, and the study of the science of warfare. Standardization of weapons and equipment has become the major problem of the Nigerian Army during the period of re-organization. In most of the developing countries weapons and equipment are being designed after series of research. Some of these countries have even gone further in to production of the weapons and equipment under license. During the research stages computers are gainfully utilized.

2.31 RESEARCH AND DEVELOPMENT

This system requires a more sophisticated base analogue computer for its operation. Its functions cover a wide variety of researches and military developments. Some of these are:

- Weapon developments
- Research and developments of explosives and other functionary elements
- Research into enemy Electronic Counter-Measures (ECM) and the developments of equipments and devices for electronic counter counter measure (ECCM)

- Research into present communication equipments and the subsequent developments of some equipments to be compatible with new equipments during the reorganization and re-equipment excise.
- Research into the possibilities and methods of building some components and parts needed for the maintenance of radios, weapons and other mechanical devices.
- Research into the cause of constant faults in components or part of an equipment, which has been reliably detected as a result of components scaling made by formations to the Army Headquaters.

2.32 WAR GAMING

War gaming refers to methods of analysis, which are available for the study of military problems. These methods range from complete realism of war or battle situations to the pure obstructions of a mathematical nature, which may be represented by equations. They may include among other directs observations of real situations, field maneuvers, map exercises, simulated situations and mathematical representation of military situation. There are some military operations that cannot be carried out without just attempting to represent both our own capabilities and courses open to us and also those to the enemy. This is with a view to seeing how both sides would interact in military confrontation.

War games could be in the form of field exercise which would involve literarily tens of thousands of real troops in a mock combat and the combat operations would be conducted in a symbolic fashion. One important area of application is in the writing of our operational orders. In writing operation orders, various situations are considered and this include; the assessment of the military strength and courses open to the enemy. There is the need also to take into consideration the amount of resources available to own troops, given logistics and be able to decide which techniques should be adopted. The military strength of each side engaged in a battle could be programmed and fed into the computer. The computer would give a print out "win" and "loose" stating the side that would win or loose under the given situation. This would enable the commander who is predicted to loose to re-adjust withdraw tactically; or take another strategy until he is able to have a win. It should be emphasized however, that the validity of war games does not ordinarily lies with the accuracy with which the computation is done, but rather in the extent to which the sides engaged in battle could be faithfully represented and the rules could be so designed to bear some relationships to real operations.

2.4 ADMINISTRATIVE SYSTEM

The configuration of the administrative computer system are of higher sensitivity, complexity and are of massive capacity compared with other systems. The reason is that they offer new opportunities for the optimum use of the total manpower and material resources available to the military. The ability to mange a large organization effectively has increased because of the massive data emanating from the organizational set up. Also, high productivity must always be the first priority of any organization. This concept of high productivity is by no way latest or foreign to the military minds.

Mr. J. C. T. Downey in his book entitled management in the armed forces highlighted that in 1992, at the inauguration ceremony of the Royal Air Force Staff College, trenchard said the one great thing to which you should at all times apply your thoughts and brains is the expansion of the power of materials and personal without increasing either". Infact, these system which are to be discussed in the succeeding paragraphs will refine and provide suggestive information for the Nigeria Army(NA) upon which daily decisions can be based. The tasks most concerned are; supply systems, combat engineering, pay and personal records.

2.41 SUPPLY SYSTEMS

The management of supplies withing a large organization like the military requires an effective management system through a reliable and responsive department where responsibilities are clearly defined and understood. In the Nigerian Army (NA) for instance, the processing of supply systems, stock level determination, and issue of stocks are channeled through the various departments under the senior provincial officer (S.P.O.) in the Nigeria Army Ordinance (NAOC) units.

The approval of formation and units demands are processed in the S.P.O's office by making reference to catalogues, different voluminous maters stock ledgers, units scales and unit equipment table (U.E.F.) This procedure is the very cubersome and will demand a system that will assist the mental alertness and physique of the of the conventional ledger clerks. Often the staff at the Army Headquarters (AHQ) who are responsible for releases, issues and provision of stores and equipments are unable to carryout contingency planning due to want of timely information in thee NAOC units. A school of thought once received these problems and eventually concluded on the application of accounting and punched card machines in alleviating these problems. It was a good idea but failed to take cognizance of the growth in the families of equipments, weapons, vehicles, armaments and the wide range of spares as a result of the re-equipment exercises. Also the punched card machine will not be sensible and sensitive to the present storage systems. For example, stocks are now stored by considering the characteristic of issue and size and not be stored and readily available in the punched card machine system. It is important to mention that the supply computer system will not in any way revolutionize the present manual system employed in the base ordinance depot (BOD) but will speedily assist in stock selection, voucher preparation, stock taking and stock budgeting control. This will provide up-to-date stock position, which is required for timely computation to effect prompt provisioning of stores. More essentially, the processing of store and equipments, located in the forward ordinance unit can be rationalized in order to reduce the administrative tail.

2.42 COMBAT ENGINEERS

The engineering computers system is useful to the military for technical data recording and project management. This system has mores inclination to the research computer system than the other. The technical data is essentially needed in the military to enhance the creation of historical records of equipment, components and spares performances. This should be the frent of events in any developing army because procurement of spares and components must therefore be based on result of sculling exercise. This historical data and fault analysis of the faulty components will be stored in the computer and made readily available to the research centre.

The second role of the system, which includes project management may be undertaken to guide the engineers and workshop units in implementing massive and major tasks assigned to them. In this case a min-computer can be provided at the disposal of the resident engineer and a major servicing unit to help in the allocation of resources and the distribution of workload between the work centres. Often, the microprocessor computers are suitable for this task.

2.43 PERSONAL AND PAY SYSTEM

Te growth in size of the Nigerian Army (NA) during the war made people of different characters to be recruited into in to Army indiscriminately. This was because the proper attestation rules and recruitment policy were relaxed. In addition, the NA was unable to account for the strength of the entire army even after several year of the civil war. The unscrupulous officers and men exploited these flaws on the parts of the NA and fraud became rampart in the army. In 1976 the NA took a bold step by conducting a very well planned and effective census throughout the units and divisions.

Prior to the 1976 census personal documents were never updated. Most of the information contained in them were belated and many years behind the current happenings. For instance, by 1975 the record of a soldier who was a corporal in 1972. Proper career planning for officers and men could not be achieved because the NA did not know which officer or other rank has attended what course or courses.

However, with the introduction of computers in the NA for personnel' management overtook all these problems as records of crimes and other disciplinary cases were kept up to date. Not allowing officers and other ranks who have been convicted to later get promotions and nomination for courses they don't deserve, unlike in the past, when people got what they didn't deserve due to lack of accurate records.

Promotion examination rules are now effective because the NA maintained a proper record. In the past officers who did not pass the prescribed examination got promoted because of lack of record.

Computers have replaced the massive and voluminous manual filing system used formerly in the NA. Thereby saving us time to get things done. Adequate staff coordination between departments has also been achieved.

Before computers were introduced into the NA, a lot of paper work had to be involved in handling officers and troops payment. Considering the case of officers giant ledgers and binders maintained on each officer. Also cumulative figures have to monthly be entered on each officers pay record monthly. In addition, monthly pay slips are manually prepared. The work involved has in the past caused delay between pay preparation and when payment are made to the officers banks. Invariably, entitlements, coming two (2) or three (3) weeks to the pay time will be too late to be effected.

The computer undoubtedly has improved the organization of the NA personnel management.

Data stored in computers can always be updated easily therefore, the computer can be used for the following:

Payroll

Qualitative inventory of officer's personal asset

Day to day work and career planning

Long range planning information and bank ground information of officers

Mustering/discharge

Posting and promotion

Crime information retrieval

2.5 COMPUTERS IN MODERN WAR FARE.

One very important result of the invention of computer is the extraordinary complexity of "modern" warfare as compared to all its predecessors. This complexity simply reflects the growing sophistications of the hardware itself. After all there is no comparing a modern self propelled gun with its tens of thousands of precision made components to the artillery of fifties or even fifteen (15) years ago.

However, I must say it is not surprising that the diaries and memoirs coming out the World war 1 (WW1) are riddled with complaints concerning the way in which modern technology has helped turn war into an excise in management.

Essentially there has been the need of merging both the hardware and personnel's in to integrated teams capable of surviving on the battle field and of fulfilling their mission under Tremendous pressure and to be able to cater for momentous variety of equipments used in modern war, backed up with hundreds of thousands even millions of different spare parts that often requires different kinds of storage facilities and have different expected life spans. To achieve these objectives there has been the need to train personals in specialized fields.

Since computers were originally considered rather esoteric, one would expect the field to owe every thing to civilians and nothing to the military. Never the less, the military, particularly in America, had been involved with computers almost from the beginning in world war 11 (ww11) frequently laid down specifications and provided funds for development.

Undoubtedly, one important reason behind the love story between brass hats and their computers in the sheer size of armed forces as compared to virtually all other social organizations. If the military was to be administered with any thing like efficiency of other organizations, automation presented the only way. Second and perhaps more importantly. Computers with their binary on –off logic seem to appeal the military mind. This is because the military, in order to counter the inherent confusion and danger of war, is forever seeking ways to make communication as terse and unambiguous as humanly possible. Computers their very nature do just that. Had they only been able to stand at attention and salute, in many ways they would have made ideal soldiers.

In summary, the computer technology had evolved over the years through the information revolution. There may still be a lot of people who have not yet nuzzled up to computer there is none unaffected by the explosion of computers technology. Every thing from the media to medicine, supporting international humanitarian and peace keeping operations has been radically transformed by the use of computer technology. The revolution has only just begun, but already it's starting to over whelm the Nigerian army (NA) operational policies.

2.6 TECHNOLOGICAL SOPHISTICATION OF MILITARY EQUIPMENT

A part from general advancement in knowledge as described earlier, tremendous technological stride have been made in all spheres of life. The benefits of animation and the use of computers has tricked down everywhere. Recently the AWF war in the Middle East is a clear example of technological sophistication of military equipments. For example the ise of laser guided missiles is a clear testimony of computer-programmed attack on pinpoint targets, without necessarily bombing civilian targets in operations. In the armed forces of today particularly those of advanced countries the use of sophisticated gadgetry has become more of the rule than the exception.

Field marshal Montgomery described " the least glorious arm of all" the infantry uses sophisticated equipment like the blowpipe and rasit 3190 radar which are in the NA inventory.

Technological sophistication of military equipment will never be complete without highlighting in detail some example where the incorporation of computer gadgets in to missiles brought a major success of the operation.

2.61 ARAB ISRAELI CONFLICTS

Although air power was a major factor that determined the outcome of the Arab- Israeli war in 1967 when the six (6) day way broke out, the Israeli achieved surprise by computer based electronic deception over Egypt to the extent that all Egyptian planes were destroyed on the ground by the Israeli Air Force before they could be launched in the battle. The deception employed was that before H-hour (0745 hrs on 6 June 1967) no electronic warfare (EW) action was taken against Egyptian radar. However, during and after the attack, Israeli radio operations speaking fluent Egyptian Arabic transmitted false orders in to the Egyptian air defence communication, canceling correct orders and generally causing confusion withing the Egyptian high command. In same cases, Russian and American computer operated radars and radio communications were jammed.

2.62 THE SOUTH ATLANTIC (FALKLANDS) CONFLICTS;

The Falkland war, which started on the 2nd of April 1982. This war was fought with the most modern weapons but the British and the Argentines relied extensively on electronic intelligence (ELINT) because of their awareness that the political effect of attacking the enemy with a single missile was far greater than the physical damage such missile could cause .it was important for the British and Argentines to obtain accurate and timely information about, each others capabilities and dispositions. By (ELINT). For instance, the British needed to know the radar frequencies of Argentina's missiles so that the missiles could be confused by electronic counter measure. The argentine, on her part needed to know the position of the British ship so that they could estimate when and how an attack could develop.

2.63 THE GULF WAR

The gulf war demonstrated lessons I all the element of computer in the modern warfare, using electronic warfare as a hypothetical example. the allied FW efforts were refined during series of exercise such as us green flag against postulated formidable soriet threat. No wonder, allied EN disrupted Iraqis command, control communications and intelligence (C_3I) system. It served the command tank from Baghdad from field forces, which led directly to quick collapse of the Iraqi army

CHAPTER THREE

DATABASE MANAGEMENT SYSTEMS

The environment in which a database management system is likely to be used for a microcomputer in different form that of a mainframe. The buyer of a microcomputer is likely to be the proprietor of a small business or a home computer.

The mainframe computer system will probably be supported by a team of specialists, whereas the microcomputer user frequently has little previous experiences of computing and has to rely upon the dealer support and database documentation. Ease of use, therefore is an important consideration for the lonely user of a microcomputer database. Relational databases tends to be more popular with non-professional users because they are easier to understand and use.

The restriction on database systems for microcomputer are primarily the obvious ones processors and storage capacity. The file between a user requesting an item o data and that item being upon the screen in the most important performance characteristics as far as the user is concerned. This access time will depend upon factors such as the speed of the processor, the choice of disk (Hard or floppy) and the accessing algorithm used by the database management system. The storage capacity of the disks attached to a microcomputer is considerable less than the storage capacity of a mainframe, and consequently the DBMS may impose limitation upon the user. The user may find restriction upon the number of fields/bytes per record, the number of record or the number of files that he may define in the database. The use of either an8-bit or a 16-bit processor will impose limitation upon the signs and accuracy of numeric data field, they may also be limitation upon the available data types.

Database Administrator

An organization that implement a centralize policy for controlling its data should appoint one person to be responsible for the contents of the database. The description of the contents of a database is generally called a scheme, and the database administrator uses a data definition language (DDC) to create or utter the scheme.

The responsibility of the databased administrator can be summarised as:

(a) To define the data elements. The database does not define the values of the data but specify the size and type of the data elements.

(b) To define the relationship between data elements.

(c) To define the security requirement of the database.

(d) To ensure that the content of the database are constant and that the database contains as little redundant information as possible.

PROGRAM IMPLEMENTATION

System requirement (minimum)

386 33MHz Mother Board

4MB RAM

10 MB Hard Drive Space

Mouse

Boot and go to DOS

Change directory to stock

from the roof directory

type stock to run the program

password = Alex

processing date

What the program can do

1 Code file maintenance

Suppliers

Location

In the codes file maintenance you can carry out the following

Add

Modify

Delete

Print

2 Stock file maintenance

This is where codes are defined for every new item

The following are defined

Code number

Code description

Quantity

Unit price

Location

3 Stock purchases. Here you can

Add

Modify

or Delete

4 Requisition. Here you can

Add

Modify

Delete

and Print

5 Post Ledger Card - This is where all postings are done list of items ready to be printed out after transaction

6 Reversal - Here you can reverse items you have posted already if you change your mind. It automatically ignores what you posted initially

7 EXIT - This is were you exit the program

UTILITIES

Rebuild index

Back up

Restore

Change screen appearance

Select predefined colours

HELP

About stock

Weight & Measure

F1 - Time

F2 - Calculator

F4 - Calendar

F 10 - Quit

CHAPTER FOUR

RECOMMENDATION AND CONCLUSION

RECOMMENDATIONS

The Military environment is a very sensitive one and its roles are unique. It is reasonable to state that the self-reliance and strength of a National depends to the extent of reliance of its defence. On less of foreign goods and expertise. A nation like Nigeria aiming at self-reliance must be technologically independent and this implies that it should take necessary steps to reduce her reliance on foreign goods and expertise.

The acquisition, management and manufacture of computers especially for military applications should not be left entirely and for much longer in the hands of the manufacturers who are foreigners. There is therefore, the need to embark on an intensive training of qualified military personals in the computer hardware fields, especially in maintenance and design, this could be followed up gradually with training towards the manufacturing stage. Agreements could be reached with willing foreign Computer Companies to set up factories, import parts of their computer and assemble them in Nigeria. The same company could be encouraged to start manufacturing components of this computer system in Nigeria, thus gradually decreasing the import necessary.

CONCLUSION

Computers have come to stay as a total planning, storage and resources management. This will reduce the work load of units/formation, thereby improving their efficiency and work turn around of the entire personnel of the armed forces (AF) if will equally ensure the Nigerian Army (NA) combat readiness there by enhancing personal performances in future peace keeping operations (PKO) application of computer in the NA is a useful means of training to enhance efficiency of personnel. This could reduce lost in men and equipment while in battle.

I must say it is long over due processing due to change from manual technique of doing things to computer processing. Hence the need for information technology to enhance the (AF) personnel proficiency in rendering records, returns and reports. However, with a sound computer education a Nigerian Officer stands a good chance of facing the challenge in modern warfare. Modern command and control procedures will not be new to him especially when deployed in PKO along side officers of other nations.

With adequate awareness and better working conditions using computers, the NA forces will produce faster results, bearing in mind that computer assist to produce accurate results, reports at short time and should be considered a problem solving machine on the battle field.

There are civilians who have been trained as system programmers or system engineers They could be recruited into the military, their number could be increased by sending military personnel to undergo the required training in local universities. They could be sent abroad. There is the need to establish a central software library were local and foreign packages are kept maintained and properly documented for easy access and retrieval.

It is pertinent therefore for every unit/formation in the armed forces to be computerized. Personnels should be given the opportunity to attend computer awareness course.

The headquarters training and doctrine command should embark on a computer training policy to be included in all Nigerian Army training schools syllabus. The headquarters Naval training command and Airforce training command should follow suite.

The ministry of defence (MOD) should embark on a battle digitalization programme and frequent, armed forces joint training exercise.

REFERENCES

1. Vogt M.A. and Koko A.E.A. (1992): Nigeria In International Peace Keeping 1960-1992.

2. Watson B.W. (1993): Military Lessons of the Gulf War. London Green Hill Book.

3. CROWSQUILL (1983): Journal of Royal Signals. Blandford. Vol. Xvi No. 3

4. PAPA NASS (1992): Computer Awareness Course, Hard Book.

5. MARTIN VAN CREVALD (1999): Technology and War. From 2000 B.C. to the Present.

```
PROGRAM : STOCK.PRG
AUTHOR  : AHMED TIJJANI IBRAHIM

#include "\fp\Common.ch"
#include "\fp\include\Set.ch"
#include "\fp\include\Std.ch"
#include "\fp\include\Inkey.ch"
#include "\fp\include\Setcurs.ch"
#include "\fp\include\Box.ch"
#include "\fp\PayDefa.ch"

#translate CODEBLOCK(<b>) => <{b}>

LOCAL lActive1  := .t., lActive2 := .t., lActive3:= .t.
LOCAL bExitOk   := {||MESSYN("  Exit Application now ?")}

LOCAL aMiceHot  := { {24,5,24,14,CODEBLOCK(msg(200,"The time is: ";
                     +time12(time()))) },;
                     {24,58,24,70,CODEBLOCK( bungeequit() )},;
                     {0,1,0,3,CODEBLOCK( bungeequit() )},;
                     {24,20,24,34,CODEBLOCK(GETCALC())},;
                     {24,40,24,53,CODEBLOCK(GETDATE())} }

LOCAL aKeysHot  := { {K_F10,CODEBLOCK(msg(200,"The time is: ";
                     +time12(time()))) },;
                     {K_ALT_X,CODEBLOCK( bungeequit() )},;
                     {K_F2,CODEBLOCK( GETCALC() )},;
                     {K_F4,CODEBLOCK( GETDATE() )} }

LOCAL aMenu

PUBL MPERIOD,CN,OPTION:=0, QueryFields := 12, Sno := 1, Page := 1,;
mPrnDevice, aRounder, aCurrency
public DataEntColor
REQUEST DBFCDX
RDDSETDEFAULT("DBFCDX")
PATHNO := 1
nMaxRow:=MaxRow()
nMaxCol:=MaxCol()
cStartScreen:=SaveScreen(0,0,MaxRow(),MaxCol())
IF !FILE("Password\PaPassWd.dbf"); QUIT; ENDIF
USERS := 1
aDefa := aDefault("Default.prl")
mPrnDevice := AllTrim(aDefa[ PRN_PORT ])
SET(_SET_CURSOR,SC_NONE);SET(_SET_SCOREBOARD,.F.); SET(_SET_WRAP,.T.)
SET(_SET_DATEFORMAT,aDefa[ DATE_FORMAT ])
SET(_SET_DELETED, .t.)
SET(_SET_CANCEL,.t.); SET(_SET_BELL,.F.); CLEAR
SET CENTURY ON
slsf_color("S3COLOR")
initsup()
SETCOLOR(sls_normMENU())
CLEAR
setcolor(Sls_PopCol())
DATAENTCOLOR:=setcolor(SLS_NORMCOL())
MSG(5,[Copyright (C) 1999 - 2000 Ahmed Tijjani Ibrahim],;
        [ ],;
        [               All Rights Reserved       ])

If !PassWord(1,"Papasswd"); Quit; Endif
IF FILE("26061969.TMP")
  BUZZ()
  if nChoice==1
    copy file 26061969.TMP to prn
    delete file 26061969.TMP
  elseif nChoice==2
    delete file 26061969.TMP
  endif
endif
ProcessDate(18,23,"default.prl")
External KBDESC
CN:=[Ahmed Tijjani Ibrahim]
ORGSELECT:={}; ORGOPTION := {}
USE default.prl NEW; MDATAPATH := ALLTRIM(DATAPATH); Use
SET PATH TO &(MDATAPATH)
MAINMENUCOLOR := SETCOLOR()

BUNGSTART()
BUNGOPTION("Files")
   BUNGDROP()
      BUNGOPTION("~Codes File Maintenance         "+chr(16))
      bungdrop(1,28)
         BUNGOPTION("Add ~New Code",{||SSR(.t.),;
         P_code(1),SSR(.f.)})
         BUNGOPTION("~Change Existing Code",{||SSR(.t.),;
         P_code(2),SSR(.f.)})
         BUNGOPTION("Delete ~Existing Code",{||SSR(.t.),;
         P_code(3),SSR(.f.)})
         BUNGOPTION("~Display Codes",{||SSR(.t.),;
         P_code(4),SSR(.f.)})
         BUNGOPTION("Prin~t Codes",{||SSR(.t.),;
         P_code(5),SSR(.f.)})
      bungundrop()
      BUNGOPTION("CROSSBAR")
      BUNGOPTION("Stock Opening Balance          "+chr(16))
      Bungdrop(1,28)
         BUNGOPTION("Add ~New Stock Item",{||SSR(.t.),;
         P_persmenu(1),SSR(.f.)})
         BUNGOPTION("~Change Existing Stock Item",{||SSR(.t.),;
         P_persmenu(2),SSR(.f.)})
         BUNGOPTION("Delete ~Existing Stock Item",{||SSR(.t.),;
         P_persmenu(3),SSR(.f.)})
         BUNGOPTION("~Display Stock Items",{||SSR(.t.),;
         P_persmenu(4),SSR(.f.)})
         BUNGOPTION("Print Stock Items",{||SSR(.t.),BinPrn(),SSR(.f.)})
      Bungundrop()
      BUNGOPTION("Stock ~Purchases                 "+chr(16))
      Bungdrop(1,28)
         BUNGOPTION("Add ~New Stock Item",{||SSR(.t.),;
         P_purchase(1),SSR(.f.)})
         BUNGOPTION("~Change Existing Stock Item",{||SSR(.t.),;
         P_purchase(2),SSR(.f.)})
         BUNGOPTION("Delete ~Existing Stock Item",{||SSR(.t.),;
         P_purchase(3),SSR(.f.)})
         BUNGOPTION("~Display Stock Items",{||SSR(.t.),;
         P_purchase(4),SSR(.f.)})
      Bungundrop()
      BUNGOPTION("Stock ~Movement Data Entry      "+chr(16))
      Bungdrop(1,28)
         BUNGOPTION("Add ~New Stock Item",{||SSR(.t.),;
         P_movement(1),SSR(.f.)})
         BUNGOPTION("~Change Existing Stock Item",{||SSR(.t.),;
         P_movement(2),SSR(.f.)})
         BUNGOPTION("Delete ~Existing Stock Item",{||SSR(.t.),;
         P_movement(3),SSR(.f.)})
         P_movement(4),SSR(.f.)})
      Bungundrop()
      BUNGOPTION("CROSSBAR")
      BUNGOPTION("Post Stock ~Ledger Card",{||ssr(.t.),;
      UpdateStock(),ssr(.f.)})
      BUNGOPTION("Reverse Posted Requisition",{||ssr(.t.),;
      ReversePosting(),ssr(.f.)})
      BUNGOPTION("CROSSBAR")
      BUNGOPTION("~Exit",{||bungeequit()})
    BUNGUNDROP()
BUNGOPTION("Reports")
  BUNGDROP()
     BUNGOPTION("Stock ~Analysis Report",{||ssr(.t.),;
     StockSummary(1),ssr(.f.)})
     BUNGOPTION("Stock Balan~ce Report",{||ssr(.t.),;
     StockSummary(2),ssr(.f.)})
  bungundrop()
BUNGOPTION("Utilities")
   BUNGDROP()
      BUNGOPTION("~File Re-contruction",{||SSR(.t.),reorg(),SSR(.f.)})
      BUNGOPTION("CROSSBAR")
      BUNGOPTION("Data Files ~Backup",{||SSR(.t.),;
      if(Password(2,"papasswd"),backup(),),SSR(.f.)})
      BUNGOPTION("Data Files ~Restore",{||SSR(.t.),;
      if(Password(18,"papasswd"),restore(),),SSR(.f.)})
      BUNGOPTION("CROSSBAR")
      BUNGOPTION("Change Data Files ~Directory",{||SSR(.t.),;
      mpath("Default.Prl"),SSR(.f.)})
      BUNGOPTION("CROSSBAR")
      BUNGOPTION("Files ~Cleaning Routine",{||SSR(.t.),;
      P_DUPDELE(),SSR(.f.)})
      BUNGOPTION("Delete ~Unwanted Files",{||SSR(.t.),;
      CleanSystem(),SSR(.f.)})
      BUNGOPTION("CROSSBAR")
      BUNGOPTION("~Measures & Weights Conversion",{||SSR(.t.),;
      Wgt_Meas(),SSR(.f.)})
      BUNGOPTION("CROSSBAR")
      BUNGOPTION("Change ~Screen Appearances",{||SSR(.t.),;
      setcolors(),SSR(.f.)})
      BUNGOPTION("Select ~Predefined Colors",{||SSR(.t.),;
      ColPik(),SSR(.f.)})
   BUNGUNDROP()
BUNGOPTION("Help")
   BUNGDROP()
      BUNGOPTION("~About Stock",{||assist()})
   BUNGUNDROP()
AMENU := BUNGEND()
Setcolor(SLS_popcol())
DISPBOX(0,0,2,79,sls_frame())
@ 24,1 say padr("      "+chr(16)+[F10=Time]+chr(17)+"      "+;
chr(16)+[F2=Calculator]+chr(17)+"      "+chr(16)+[F4=Calendar]+;
chr(17)+"      "+chr(16)+[ALT-X=exit]+chr(17),78)
selection:=bungee(1,2,77,aMenu,nil,nil,aMiceHot,;
aKeysHot,bExitOk,nil,nil,1)
if selection=nil
  ss_rise(0,0,MaxRow(),MaxCol(),cStartScreen,3000)
  @ nMaxRow,nMaxCol say ""
  return
EndIf
Return

function assist()
```

```
Function Dotmatrx()
UseFile("DOTMATRX",nil,"NoIndex")
EditDb(.t.,nil,.t.,.t.)
CloseArea(Alias())
Return("")
*
FUNCTION vRange(a,b,c)
return (if(a >= b .and. a <= c,.t.,.f.))

Function DiseEdit()
OpenFiles(16,16)
EditDb(.t.,nil,.t.,.t.)
CloseArea(Alias())
Return("")

func Closearea(cFile)
use (cFile)
use
return ""

Func KeyInput(getfield,a,b)
Local MPASS := [], mPassa :="",mRowLine,mColLine
mROWLINE := a
mCOLLINE := b
MROW := MROWLINE; MCOLUMN := MCOLLINE
FOR I = 1 TO getfield
   mpassa:=[ ]
   @ mROW,mCOLUMN GET MPASSA PICT [x]
   READ
   IF LASTKEY()=13
      EXIT
   ELSE
      @ mROW,mCOLUMN++ SAY [.]
      MPASS = MPASS + MPASSA
   ENDIF
NEXT
return (Mpass)

function ss_fade(nTop,nLeft,nBottom,nRight,cInScreen)
local cCurrent := savescreen(nTop,nLeft,nBottom,nRight)
local nIter
for nIter = 10 to 1 step -1
  for a=1 to 12000;next
  restscreen(nTop,nLeft,nBottom,nRight,ssprinkle(cInScreen,;
  savescreen(nTop,nLeft,nBottom,nRight),nIter))
next
return nil

function Buzz()
tone(523,2)
tone(698,2)
tone(880,2)
tone(1046,4)
tone(880,2)
tone(1046,8)
return .t.
*
function xNannyBoo()
tone(196,4)
tone(196,4)
tone(164,4)
tone(164,8)
return .t.
*
Function xTheFifth()
tone(392,2)
tone(392,2)
tone(392,2)
tone(311,10)
tone(15,12)
tone(349,2)
tone(349,2)
tone(349,2)
tone(293,10)
return .t.

#include "prl.prg"
#include "\newp\speb\prlhead.prg"
#include "\FP\SOURCE\S_BXX.prg"
#include "\FP\SOURCE\x\S_filer.prg"
```

```
PROGRAM : BINCARDS.PRG
AUTHOR  : AHMED TIJJANI IBRAHIM

#include "all.ch"
PROC P_PERSMENU(Action)
PUBLIC cInScreen := savescreen(0,0,MaxRow(),MaxCol())
SupplierCode:={};ItemCode:={};LocationCode:={};MoveCode:={};DeptCode:={}
MLIST:={}
PUBL XX:={},OPTION,ACODETYPE:={},nCategory:=" ",C
mItemNo:=space(9)
LOADCODES()
DbCloseAll()
PLSWAIT(.T.,"Opening Bincards Data Files")
OPENFILES(1,2)
Plswait(.f.)
SELECTFILE([BINCARDS],[ITEMNO])
SetCursor(3)
DO WHILE .T.
   setcolor(dataentcolor)
   DO CASE
   CASE Action = 1; MACTION := [RECORD WILL BE ADDED]
     CODEID:=ACODE(.T.)
     FOR I = 1 TO 4
       AADD(MLIST, CHR(I+64)+[ - ]+CODEid[I,3])
     NEXT
     SETCURSOR(3)
     Option:=0
     aa:=SaveScreen(0,0,maxRow(),MaxCol())
     DO WHILE .T.
       OPTION:=MCHOICE(mList,4,30,nil,nil,"[ SELECT: ]")
       IF OPTION > 0;EXIT;ELSE;LOOP;ENDIF
     ENDDO
     SS_SLIDELEFT(0,0,MaxRow(),MaxCol(),aa)
     C:=CODEID[OPTION,2]
     cTmpCode:=c
     CodeItem:=ItemCode
     PersAmd(1)
   CASE Action = 2; MACTION := [RECORD WILL BE CHANGED]
     persamd(2)
   CASE Action = 3; MACTION := [RECORD WILL BE DELETED]
     persamd(3)
   CASE Action = 4; MACTION := [RECORD VISUALIZATION]
     persamd(4)
   ENDCASE
   EXIT
enddo
*
PROC PERSAMD(Action)
MNEXTSENT := .F.
DO WHILE .T.
  Setcolor(DataEntColor)
  PERSFORM()
  @ 24,0
  PersCls()
  IF !MNEXTSENT
    IF !PERSHEAD(@mItemNo,Action)
      ss_IMPLODE(0,0,MaxRow(),MaxCol(),cInscreen,2500)
      RETURN
    ENDIF
  ENDIF
   If(ACTION=1,Memorize(.T.),Memorize())
    DO CASE
```

```
      CASE ACTION = 2
        PERSHOWREC(); PERSEDIT(Action)
      CASE ACTION = 3
        PERSHOWREC()
        Buzz()
        If Password(1,"PAPASSWD")
          DelRecord("BINCARDS",.t.)
        EndIf
        Select("BINCARDS")
        Loop
      CASE ACTION = 4
        PERSHOWREC()
    ENDCASE
  IF ACTION != 1
    MM = NEXTPREV(24)
    IF(MM != 3,MNEXTSENT := .T.,MNEXTSENT := .F.)
      IF MM = 6
        RETURN
      ENDIF
  ENDIF
ENDDO
*
PROC PERSFORM
setcolor(dataentcolor)
CLEAR
SET DATE FORMAT TO "DD-MM-YYYY"
CENTMSG(0,CN+[ - ]+[STOCK CONTROL SYSTEM])
@ 01,0 TO 1,79
@ 02,0 SAY [DATE: ]+DTOC(DATE())+SPACE(3)+;
[Stock Opening Balance]+SPACE(32)
@ 03,0 TO 3,79
@ 04,0 SAY PADl([Item Code],30)
@ 05,0 TO 5,79
@ 06,0 SAY padl([Item Description],30)
@ 08,0 SAY padl([Quantity Received],30)
@ 10,0 SAY padl([Unit Cost],30)
@ 12,0 SAY padl([Location Code],30)
@ 14,0 SAY padl([Sub-Location],30)
@ 23,0 TO 23,79 DOUBLE
RETURN
*
FUNC PERSHEAD(mItemNo,Action)
WHILE .T.
   selectFILE([bincards],[ITEMNO])
   PersCls()
   CENTMSG(24,[Enter Item Number or Press ESC to Exit])
   GetSay(4,35,@mItemNo,[@!]); @ 24,0
   If LASTKEY() = 27; RETURN(.F.); ENDIF
   IF SEEKEY(mItemNo) .AND. ACTION = 1
     BUZZ()
     MSG("Item Number Already Exist")
     LOOP
   ENDIF
   IF !SEEKEY(mItemNo) .AND. ACTION != 1
     buzz()
     MSG("Item Number does not exist.")
     LOOP
   ENDIF
   RETURN .T.
ENDDO
*
PROC SAVEPERS(Action)
If Hold(1)
```

```
   if action=1
     repl Balance with mQuantity
   endif
   repl ItemNo with mItemNo,Quantity with mQuantity
   repl UnitCost with mUnitCost,Location with mLocation
   REPL Flag with c, Shell with mShell
   Release()
   AutoSave(1)
EndIf
RETURN
*
PROC PERSEDIT(Action)
DO WHILE .T.
   mPersRow:=6
   GETSAY(mPersRow,35,@mDesc,[@!K])
   mPersRow+=2
   GETSAY(mPersRow,35,@mQuantity,[999,999])
   mPersRow+=2
   GETSAY(mPersRow,35,@mUnitCost,[999,999.99])
   mPersRow+=2
   bGet := {|| GetOrPick(-1,24,0,"[F2]",mPersRow,35,@mLocation,[@!],;
   LocationCode,mPersRow-6,35,21,70,Len(mLocation),;
   "[Select Location Code:]")}
   bCondition := {|| !SEEKEY(CID(aCODE(),"STAFCATEID")+;
   mLocation,"CODEFIL")}
   cMsg := [ Location Code Not Found ]
   bDisp:= {||  DEVPOS(mPersRow,35),QQout( mLocation+[    ]+;
   CODELOOK(CID(aCODE(),"STAFCATEID")+mLocation,"CODEFIL",3))}
   If !PersGet(bGet,bCondition,cMsg, bDisp); Loop; EndIf
     mPersRow+=2
     GETSAY(mPersRow,35,@mShell,[@!K])
   IF ACTION = 2
     CODEID:=ACODE(.T.)
     FOR I = 1 TO 4
       AADD(MLIST, CHR(I+64)+[ - ]+CODEid[I,3])
     NEXT
     SETCURSOR(3)
     Option:=0
     aa:=SaveScreen(0,0,maxRow(),MaxCol())
     DO WHILE .T.
       OPTION:=MCHOICE(mList,4,30,nil,nil,"[ SELECT: ]")
       IF OPTION > 0;EXIT;ELSE;LOOP;ENDIF
     ENDDO
     SS_SLIDELEFT(0,0,MaxRow(),MaxCol(),aa)
     C:=CODEID[OPTION,2]
     cTmpCode:=c
     CodeItem:=ItemCode
     MLIST:={}
   ENDIF
   IF LOTUSMENU(24,10,[ Save ],[ Modify ]) == 1
     IF ACTION = 1 .AND. !ADDREC(); LOOP; ENDIF
     IF !HOLD(1); LOOP; ENDIF
     SAVEPERS(Action)
     mItemNo:=NUMTOSTR(VAL(AllTrim(mItemNo))+1,Len(AllTrim(mItemNo)))+;
     Space(9-Len(AllTrim(mItemNo)))
   ENDIF
   EXIT
ENDDO
RETURN
*
PROC PERSHOWREC
Local mRow := 6
SELECT([BINCARDS])
```

```
@ 4,35 Say mItemNo
@ mRow,35 SAY mDesc
mRow+=2
@ mRow,35 SAY mQuantity pict "999,999"
mRow+=2
@ mRow,35 SAY mUnitCost pict "999,999.99"
mRow+=2
@ mRow,35 SAY mLocation +[   ]+CODELOOK(CID(aCODE(),"STAFCATEID")+;
LOCATION,"CODEFIL",3)
mRow+=2
@ mRow,35 SAY mShell pict "@!"
*
Proc PersCls
@ 24,0; @ 4,35
@ 6,35 CLEAR TO 22,MaxCol()
SetColor(Sls_NormMenu())
@ 2,53 SAY [*** ]+MACTION+[ ***]
DispRec(Str(LastRec(),5),3,52,[Record Count])
SetColor(Sls_PopCol())
return
*
Function PersGet(bGet,bCondition,cMsg, bDisp)
Local nTmp := 0
Do While .T.
   Eval(bGet)
   IF Eval(bCondition)
      Buzz()
      nTmp := Horizontal(Urow(),cMsg,[ Retry ],[ Next ],[ Previous ])
      If nTmp == 1
         Loop
      ElseIf nTmp == 3
         Return(.f.)
      Else
         Exit
      Endif
   ENDIF
   If bDisp != Nil; Eval(bDisp); Endif
   Exit
Enddo
Return(.t.)
*
function memorize(lFlag)
default lFlag to .f.
public mItemCode,mQuantity,C,mShell
public mUnitCost,mLocation,mDesc
if lFlag
  mDesc:=space(30);mQuantity:=0;mUnitCost:=0.00;mLocation:=Space(9);
  mShell:=space(5)
else
  mDesc:=Desc;mItemCode:=ItemCode;mQuantity:=Quantity
  mUnitCost:=UnitCost;mLocation:=Location;mShell:=Shell
endif
return .t.

PROC BINPRN(Action)
public start:=.t.,COUNTDOWN:=1
SupplierCode:={};ItemCode:={};LocationCode:={};MoveCode:={};DeptCode:={}
PLSWAIT(.T.,"Opening Bincards Data Files")
OPENFILES(1,3)
Plswait(.f.)
LOADCODES()
MLIST:={}
```

```
mPageLength:=PageLength
ASORT(aCodeType)
SELECTFILE([BINCARDS],[NONITEMNO])
For CodeCnt = 1 To Len(aCodeType)
MSN := 1
start:=.t.
C := Chr(aCodeType[CodeCnt]+64)
DBGOTOP(); LOCA FOR FLAG = C
IF !FOUND();LOOP;ENDIF
xCodeHead( PadC([*** ]+xStockExtra(c))+[ ***], 77)
For CountDOWN=1 to len(LocationCode)
DbGoTop()
LOCATE FOR FLAG=C .AND. substr(LocationCode[COUNTDOWN],1,9)=;
Bincards->Location
IF !FOUND()
  LOOP
ENDIF
IF !START
  ? repl([-],80)
ENDIF
? [Location : ]+substr(LocationCode[COUNTDOWN],14)
? repl([-],80)
pagelength-=IF(START,3,2)
start:=.f.
DbGoTop()
DO WHILE !EOF()
  if flag=c
    IF substr(LocationCode[COUNTDOWN],1,9)==ALLTRIM(Bincards->Location)
      ? xCodeLine()
      IF PRNSTOP(); RETURN; ENDIF
      IF PrinterRow() > PageLength-1 .AND. !EOF();xCodeHead( PadC([*** ]+
      xStockExtra(c))+[ ***], 77) ; ENDIF
    ENDIF
  endif
  DBSKIP()
ENDDO
PAGELENGTH:=mPagelength
next
Next
PrintEnd()
*
PROC xCODEHEAD(LineHead)
EJECT()
set date format to "dd-mm-yyyy"
? PADC(CN,80);?
  ? padc(xstockextra(C),80);?
? PADR([Page: ]+NUMTOSTR(PAGE++,4)+Space(80-53)+[Date: ]+;
FullDate(DATE()),80)
? REPL([-],80)
xPrnCodeHead()
? REPL([-],80)

Func xPrnCodeHead()
? "      "+SPACE(1)+"Item  "+SPACE(4)+"Item        "+space(20)+;
"Opening"+space(1)+"      Unit"
? "S/No "+space(1)+"Number"+space(4)+"Description"+SPACE(20)+;
"Balance"+space(1)+"      Cost"+space(1)+"Sub-Location"
*
Function xstockextra(mlFlag)
  do case
   case mlFLAG == "A"
     nCategory := "Clothing & Accessories Store"
   case mlFLAG == "C"
     nCategory := "Accommodation & General Store"
   case mlFLAG == "D"
     nCategory := "Publication & Stationery Store"
  endcase
return (nCategory)

Func xCodeLine()
Return ( STR(SNo++,5)+SPACE(1)+ItemNo+Desc+space(1)+;
TRANS(Quantity,"999,999")+space(1)+TRANS(UnitCost,"999,999.99")+;
space(1)+SHELL )

FUNC XCODETYPE()
LOCAL MLIST:={}
LOCAL CODEID:=ACODE(.T.)
FOR I = 1 TO 4
  AADD(MLIST, CHR(I+64)+[ - ]+CODEid[I,3])
NEXT
Option:=0
aa:=SaveScreen(0,0,maxRow(),MaxCol())
Option := TagArray(mList,"Mark desired code(s):")
SS_SLIDELEFT(0,0,MaxRow(),MaxCol(),aa)
RETURN (OPTION)
```

```
PROGRAM : PURCHASE.PRG
AUTHOR  : AHMED TIJJANI IBRAHIM

#include "all.ch"
PROC P_Purchase(Action)
PUBLIC cInScreen := savescreen(0,0,MaxRow(),MaxCol()),mFlag
SupplierCode:={};ItemNo:={};LocationCode:={};DeptCode:={};MoveCode:={}
MLIST:={}
PUBL XX:={},OPTION,ACODETYPE:={},nCategory:=" ",C,ItemSample:={}
mDocumentNo:=space(9)
LOADCODES()
Loader()
DbCloseAll()
PLSWAIT(.T.,"Opening Bincards Data Files")
OPENFILES(1,2)
Plswait(.f.)
SELECT([BINCARDS])
SetCursor(3)
DO WHILE .T.
   setcolor(dataentcolor)
   DO CASE
   CASE Action = 1; MACTION := [RECORD WILL BE ADDED]
     pPersAmd(1)
   CASE Action = 2; MACTION := [RECORD WILL BE CHANGED]
     pPersAmd(2)
   CASE Action = 3; MACTION := [RECORD WILL BE DELETED]
     pPersAmd(3)
   CASE Action = 4; MACTION := [RECORD VISUALIZATION]
     pPersAmd(4)
   ENDCASE
   EXIT
enddo
*
PROC pPersAmd(Action)
MNEXTSENT := .F.
DO WHILE .T.
  Setcolor(DataEntColor)
  pPersForm()
  @ 24,0
  pPersCls()
  IF !MNEXTSENT
    IF !pPersHead(@mDocumentNo,Action)
      ss_IMPLODE(0,0,MaxRow(),MaxCol(),cInscreen,2500)
      RETURN
    ENDIF
  ENDIF
   If(ACTION=1,pMemorize(.T.),pMemorize())
    DO CASE
      CASE ACTION = 1
        pPersEdit(Action)
      CASE ACTION = 2
        pPerShowrec(); pPersEdit(Action)
      CASE ACTION = 3
        pPerShowrec()
        Buzz()
        If Password(1,"PAPASSWD")
          DelRecord("BINCARDS",.t.)
        EndIf
        Select("BINCARDS")
        Loop
      CASE ACTION = 4
        pPerShowrec()
    MM = NEXTPREV(24)
    IF(MM != 3,MNEXTSENT := .T.,MNEXTSENT := .F.)
      IF MM = 6
        RETURN
      ENDIF
  ENDIF
ENDDO
*
PROC pPersForm
setcolor(dataentcolor)
CLEAR
SET DATE FORMAT TO "DD-MM-YYYY"
CENTMSG(0,CN+[ - ]+[STOCK CONTROL SYSTEM])
@ 01,0 TO 1,79
@ 02,0 SAY [DATE: ]+DTOC(DATE())+SPACE(3)+;
[Store Receipt Voucher]+SPACE(32)
@ 03,0 TO 3,79
@ 04,0 SAY PADl([Document #],30)
@ 05,0 TO 5,79
@ 06,0 SAY padl([Document Date],30)
@ 08,0 SAY padl([L. P. O. Number],30)
@ 10,0 SAY padl([Invoice Number],30)
@ 12,0 SAY padl([Item Code],30)
@ 14,0 SAY padl([Quantity Received],30)
@ 16,0 SAY padl([Unit Cost],30)
@ 18,0 SAY padl([Supplier Code],30)
@ 23,0 TO 23,79 DOUBLE
RETURN
*
FUNC pPersHead(mDocumentNo,Action)
WHILE .T.
   selectfile([bincards],[documentno])
   pPersCls()
   CENTMSG(24,[Enter Document # or Press ESC to Exit])
   GetSay(4,35,@mDocumentNo,[@!]); @ 24,0
   If LASTKEY() = 27; RETURN(.F.); ENDIF
   IF SEEKEY(mDocumentNo) .AND. ACTION = 1
     BUZZ()
     MSG("Document # Already Exist")
     LOOP
   ENDIF
   IF !SEEKEY(mDocumentNo) .AND. ACTION != 1
     buzz()
     MSG("Document # does not exist.")
     LOOP
   ENDIF
   RETURN .T.
ENDDO
*
PROC pSavePers(Action)
If Hold(1)
   repl DocumentNo with mDocumentNo, Flag with mFlag
   REPL Date with mDate,Balance with mQuantity
   repl LpoNo with mLpoNo,invoice with mInvoice
   repl ItemNo with mItemNo,Quantity with mQuantity
   repl UnitCost with mUnitCost,Supplier with mSupplier
   Release()
   AutoSave(1)
EndIf
RETURN
*
PROC pPersEdit(Action)
```

```
  mPersRow:=6
  While .t.
    getsay(mPersRow,35,@mDate,[@D])
     IF EMPTY(mDate)
       BUZZ()
       MSG([Document Date Cannot be Empty])
      LOOP
    ENDIF
     Exit
  Enddo
  @ mPersRow,35 SAY mDate Pict [@D]
  mPersRow+=2
  GETSAY(mPersRow,35,@mLpoNo,[@!]); @ 24,0
  mPersRow+=2
  GETSAY(mPersRow,35,@mInvoice,[@!])
  mPersRow+=2
  mRecordNo:=RecNo()
  GetOrPick(-1,24,0,"[F2]",mPersRow,35,@mItemNo,[@!],;
  ItemSample,mPersRow-6,25,21,70,Len(mItemNo),"[Select Item Code:]")
  SELECTFILE("bincards","itemno")
  IF !SEEKEY(mItemNo)
    BUZZ()
    MSG([ Item Code Not Found ])
    LOOP
  ENDIF
  @ mPersRow,35 SAY bincards->ItemNo+[   ]+bincards->DESC
  mFlag:=Bincards->Flag
  mPersRow+=2
  GETSAY(mPersRow,35,@mQuantity,[999,999])
  mPersRow+=2
  GETSAY(mPersRow,35,@mUnitCost,[999,999.99])
  mPersRow+=2
  bGet := {|| GetOrPick(-1,24,0,"[F2]",mPersRow,35,;
  @mSupplier,[@!],SupplierCode,mPersRow-5,35,21,70,;
  Len(mSupplier),"[Select Supplier Code:]")}
  bCondition := {|| !SEEKEY(CID(aCODE(),"PAYMODEID")+;
  mSupplier,"CODEFIL")}
  cMsg := [ Supplier Code Not Found ]
  bDisp:= {||  DEVPOS(mPersRow,35),QQout( mSupplier+[   ]+;
  CODELOOK(CID(aCODE(),"PAYMODEID")+mSupplier,"CODEFIL",3))}
  If !PersGet(bGet,bCondition,cMsg, bDisp); Loop; EndIf
  IF LOTUSMENU(24,10,[ Save ],[ Modify ]) == 1
    selectfile([bincards],[documentno])
    DbGoto(mRecordNo)
    IF ACTION = 1 .AND. !ADDREC(); LOOP; ENDIF
    IF !HOLD(1); LOOP; ENDIF
    pSavePers(Action)
  ENDIF
  EXIT
ENDDO
RETURN
*
PROC pPerShowrec
Local mRow := 6
SELECT([BINCARDS])
@ mRow,35 CLEAR TO 22,79
@ 4,35 Say mDocumentNo
@ mRow,35 SAY mDate
mRow+=2
@ mRow,35 SAY mLpoNo
mRow+=2
@ mRow,35 SAY mInvoice
```

```
SUPPLIER,'CODEFIL',3)
mRow+=2
@ mRow,35 SAY mQuantity pict "999,999"
mRow+=2
@ mRow,35 SAY mUnitCost pict "999,999.99"
mRow+=2
@ mRow,35 SAY mSupplier +[   ]+CODELOOK(CID(aCODE(),"PAYMODEID")+;
SUPPLIER,'CODEFIL',3)

Proc pPersCls
@ 24,0; @ 4,35
@ 6,35 CLEAR TO 22,MaxCol()
SetColor(Sls_NormMenu())
@ 2,53 SAY [*** ]+MACTION+[ ***]
DispRec(Str(LastRec(),5),3,52,[Record Count])
SetColor(Sls_PopCol())
return
*
function pMemorize(lFlag)
default lFlag to .f.
public mDate,mLpoNo,mInvoice,mItemNo,mQuantity
public mUnitCost,mSupplier
if lFlag
  mDate:=date();mLpoNo:=space(9);mInvoice:=space(9);mItemNo:=space(9)
  mQuantity:=0;mUnitCost:=0.00;mSupplier:=Space(9)
else
  mDate:=Date;mLpoNo:=LpoNo;minvoice:=Invoice
  mItemNo:=ItemNo;mQuantity:=Quantity
  mUnitCost:=UnitCost;mSupplier:=Supplier
endif
return .t.
```

```
#include "all.ch"
PROC P_movement(Action)
PUBLIC cInScreen := savescreen(0,0,MaxRow(),MaxCol())
SupplierCode:={};ItemSample:={};LocationCode:={};DeptCode:={};
MoveCode:={}
mDocumentNo:=space(10)
mEntryNo:="00001"
LOADCODES()
loader()
DbCloseAll()
SetCursor(3)
DO WHILE .T.
   setcolor(dataentcolor)
   DO CASE
   CASE Action = 1; MACTION := [RECORD WILL BE ADDED]
   xPersAmd(1)
   CASE Action = 2; MACTION := [RECORD WILL BE CHANGED]
   xPersAmd(2)
   CASE Action = 3; MACTION := [RECORD WILL BE DELETED]
   xPersAmd(3)
   CASE Action = 4; MACTION := [RECORD VISUALIZATION]
   xPersAmd(4)
   ENDCASE
   EXIT
enddo
*
PROC xPersAmd(Action)
PLSWAIT(.T.,"Opening Data Files")
OPENFILES(1,3)
Plswait(.f.)
SELECT([movement])
MNEXTSENT := .F.
DO WHILE .T.
  Setcolor(DataEntColor)
  xPersForm()
  @ 24,0
  xPersCls()
  IF !MNEXTSENT
    IF !xPersHead(@mDocumentNo,Action)
      ss_IMPLODE(0,0,MaxRow(),MaxCol(),cInscreen,2500)
      RETURN
    ENDIF
  ENDIF
   If(ACTION=1,xMemorize(),xMemorize(.T.))
    DO CASE
      CASE ACTION = 1
        xPersEdit(Action)
      CASE ACTION = 2
        xPersShowrec(); xPersEdit(Action)
      CASE ACTION = 3
        xPersShowrec()
        Buzz()
        If Password(1,"PAPASSWD")
          DelRecord("movement",.t.)
        EndIf
        Select("movement")
        Loop
      CASE ACTION = 4
        xPersShowrec()
    ENDCASE
  IF ACTION != 1
```

```
        RETURN
      ENDIF
  ENDIF
ENDDO
*
PROC xPersForm
setcolor(dataentcolor)
CLEAR
SET DATE FORMAT TO "DD-MM-YYYY"
CENTMSG(0,CN+[ - ]+[STOCK CONTROL SYSTEM])
@ 01,0 TO 1,79
@ 02,0 SAY [DATE: ]+DTOC(DATE())+SPACE(3)+[Materials Requisition Form]+;
SPACE(32)
@ 03,0 TO 3,79
@ 04,0 SAY PADl([Document #],30)
@ 06,0 SAY padl([Serial #],30)
@ 07,0 TO 7,79
@ 08,0 SAY padl([Document Date],30)
@ 10,0 SAY padl([Requesting Department],30)
@ 12,0 SAY padl([Item Code],30)
@ 14,0 SAY padl([Quantity Required],30)
@ 16,0 SAY padl([Quantity Issued],30)
@ 23,0 TO 23,79 DOUBLE
RETURN
*
FUNC xPersHead(mDocumentNo,Action)
WHILE .T.
   select([movement])
   xPersCls()
   CENTMSG(24,[Enter Document # or Press ESC to Exit])
   GetSay(4,35,@mDocumentNo,[@!]); @ 24,0
   If LASTKEY() = 27; RETURN(.F.); ENDIF
   CENTMSG(24,[Enter Entry # or Press ESC to Exit])
   GetSay(6,35,@mEntryNo,[@!]); @ 24,0
   If LASTKEY() = 27; RETURN(.F.); ENDIF
   IF SEEKEY(mDocumentNo+mEntryNo) .AND. ACTION = 1
     BUZZ()
     MSG("Document # and Serial # Already Exist")
     LOOP
   ENDIF
   IF !SEEKEY(mDocumentNo+mEntryNo) .AND. ACTION != 1
     buzz()
     MSG("Document # or Serial # does not exist.")
     LOOP
   ENDIF
   RETURN .T.
ENDDO
*
PROC xSavePers(Action)
If Hold(1)
   repl DocumentNo with mDocumentNo,Date with mDate
   repl ItemCode with mItemCode,Quantity with mQuantity
   repl UnitCost with mUnitCost,Issued with mIssued
   repl Department with mDepartment,Movement with mMovement
   repl EntryNo with mEntryNo
   Release()
   AutoSave(1)
EndIf
RETURN
*
PROC xPersEdit(Action)
DO WHILE .T.
```

```
  mPersRow:=8
    getsay(mPersRow,35,@mDate,[@D])
     IF EMPTY(mDate)
       BUZZ()
       MSG([Document Date Cannot be Empty])
       LOOP
     ENDIF
     Exit
  Enddo
  @ mPersRow,35 SAY mDate Pict [@D]
  mPersRow+=2
  bGet := {|| GetOrPick(-1,24,0,"[F2]",mPersRow,35,@mDepartment,;
  [@!],DeptCode,mPersRow+1,35,21,70,Len(mDepartment),;
  "[Select Department Code:]")}
  if lastkey()=27;exit;endif
  bCondition := {|| !SEEKEY(CID(aCODE(),"DEPARTID")+mDepartment,;
  "CODEFIL")}
  cMsg := [ Department Code Not Found ]
  bDisp:= {||  DEVPOS(mPersRow,35),QQout( mDepartment+[    ]+;
  CODELOOK(CID(aCODE(),"DEPARTID")+mDepartment,"CODEFIL",3))}
  If !xPersGet(bGet,bCondition,cMsg, bDisp); Loop; EndIf
  mPersRow+=2
  GetOrPick(-1,24,0,"[F2]",mPersRow,35,@mItemCode,[@!],;
  ItemSample,mPersRow-6,25,21,70,Len(mItemCode),"[Select Item Code:]")
  SELECTFILE("bincards","itemno")
  IF !SEEKEY(mItemCode)
    BUZZ()
    MSG([ Item Code Not Found ])
    LOOP
  ENDIF
  @ mPersRow,35 SAY bincards->ItemNo+[    ]+bincards->DESC
  mPersRow+=2
  DO WHILE .T.
    GETSAY(mPersRow,35,@mQuantity,[99,999])
    mPersRow+=2
    GETSAY(mPersRow,35,@mIssued,[99,999])
    IF mQuantity >= mIssued
      exit
    else
      buzz()
      msg([Quantity Issued Cannot be greater than Quantity Required])
      mPersRow-=2
      loop
    endif
  enddo
  mPersRow+=2
  mUnitCost:=CODELOOK(mItemCode,"Bincards",8)
  mPersRow+=2
  IF LOTUSMENU(24,10,[ Save ],[ Modify ]) == 1
    SELECT([movement])
    IF ACTION = 1 .AND. !ADDREC(); LOOP; ENDIF
    IF !HOLD(1); LOOP; ENDIF
    xSavePers(Action)
    mEntryNo:=NUMTOSTR(VAL(AllTrim(mEntryNo))+1,Len(AllTrim(mEntryNo)))+
    Space(5-Len(AllTrim(mEntryNo)))
  ENDIF
 EXIT
ENDDO
RETURN
*
PROC xPersShowrec
```

```
@ 4,35 Say mDocumentno
@ 6,35 SAY mEntryNo
mRow+=2
@ mRow,35 SAY mDate
mRow+=2
@ mRow,35 say mDepartment+[ ]+CODELOOK(CID(aCODE(),"DEPARTID")+;
Department,'CODEFIL',3)
mRow+=2
@ mRow,35 SAY mItemCode+[ ]+martha(mItemCode)
mRow+=2
@ mRow,35 SAY mQuantity pict "9999"
mRow+=2
@ mRow,35 SAY mIssued pict "9999"
return
*
Proc xPersCls
@ 24,0
@ 8,35 CLEAR TO 22,MaxCol()
SetColor(Sls_NormMenu())
@ 2,53 SAY [*** ]+MACTION+[ ***]
DispRec(Str(LastRec(),5),3,52,[Record Count])
SetColor(Sls_PopCol())
Return
*
Function xPersGet(bGet,bCondition,cMsg, bDisp)
Local nTmp := 0
Do While .T.
   Eval(bGet)
   IF Eval(bCondition)
      Buzz()
      nTmp := Horizontal(Urow(),cMsg,[ Retry ],[ Next ],[ Previous ])
      If nTmp == 1
         Loop
      ElseIf nTmp == 3
         Return(.f.)
      Else
         Exit
      Endif
   ENDIF
   If bDisp != Nil; Eval(bDisp); Endif
   Exit
Enddo
Return(.t.)
*
function xmemorize(lFlag)
default lFlag to .f.
public mDate,mIssued,mItemCode,mQuantity
public mUnitCost,mDepartment,mIssued,mMovement
if !lFlag
  mDate:=date();mItemCode:=space(9);mQuantity:=0;mMovement:=space(9)
  mUnitCost:=0.00;mDepartment:=Space(9);mIssued:=0.00
else
  mDate:=Date;mItemCode:=ItemCode;mQuantity:=Quantity;mMovement:=Movement
  mUnitCost:=UnitCost;mDepartment:=Department;mIssued:=Issued
endif
return .t.

Function Martha(klm)
Public Kazzah
selectfile([bincards],[itemno])
DbSeek(klm)
kazzah:=Desc
```

```
* PROGRAM : CODE.PRG
* AUTHOR  : AHMED TIJJANI IBRAHIM

#include "\fp\Common.ch"
PROC P_CODE(Action)
Local nRow,cInScreen:=SaveScreen(0,0,MaxRow(),MaxCol())
publ c,nCategory:="",ZXCV:=""
MCODE := SPACE(9); MDESC := SPACE(30)
nSelect:=0
RECDEL := .F.
UseFile("default.prl",nil,"NoIndex")
PLSWAIT(.T.);OPENFILES(2,2); PLSWAIT(.F.); SELECT("CODEFIL");
 DBGOTOP()
publ CODEID := aCODE()
publ cTmpCode := C
If Action = 5
   CODEPRN(Action)
   Close data; ss_FALL(0,0,MaxRow(),MaxCol(),cInscreen,2500)
   return
EndIf
If Action < 4
   C:=CODETYPE(Action)
   If C=="Dummy"; Close Data; ss_closeh(0,0,MaxRow(),MaxCol(),;
   cInscreen,2000) ; RETURN;EndIf
Else
   aCodeType:=CodeType(Action,.T.)
EndIf
If ACTION = 4
   CODESCROLL(aCodeType)
   Close data; ss_closev(0,0,MaxRow(),MaxCol(),cInscreen,2000)
   return
EndIf
nRow:=5; MSN := 1
@ 20,1 TO 20,77
aTmp := {0,Space(8),"  ",Space(15),0,0}
cTmpCode:=C
DO WHILE .T.
   Inverse(); a:=BlinkOn();CENTMSG(24,;
   [Enter Code or Press ESC to EXIT]); BlinkOff(a); Normal()
   @ 22,27 Get MCODE Pict [@!K] valid !Empty(mCode); Read
   @ 22,27 Say MCODE
   @ 24,0; IF LASTKEY() = 27; Close data;IF(ACTION=2,ss_FOLD(0,0,;
   MaxRow(),MaxCol(),cInscreen,1),IF(ACTION=3,ss_IMPLODE(0,0,MaxRow(),;
   MaxCol(),cInscreen,2000),ss_RISE(0,0,MaxRow(),MaxCol(),cInscreen,;
   2000)));RETURN; ENDIF
   IF SEEKEY(C+MCODE) .AND. ACTION = 1; MSG([CODE ALREADY EXIST]);
   LOOP; ENDIF
   IF !SEEKEY(C+MCODE) .AND. ACTION != 1; MSG([CODE NOT FOUND !]);
   LOOP; ENDIF
   if action <> 1
    mdesc:=desc
   endif
   @ 22,19 SAY STR(MSN,4)
   IF ACTION != 3
      GetDesc(22,37,@MDESC,[@!K],24,30)
       If Empty(mDesc); Msg([Description Cannot Be Blank]); Loop; Endif
    ENDIF
    IF ACTION != 3
       IF LOTUSMENU(24,10,[ Save ],[ Change ]) == 1
          IF ACTION=1 .AND. !ADDREC(); LOOP; ENDIF
          IF !HOLD(1); LOOP; ENDIF
          REPL CODE WITH C, CODENAME WITH (MCODE),&(FIELD(3)) WITH MDESC
          AllTrim(MDESC)
          MCODE:=NUMTOSTR(VAL(AllTrim(MCODE))+1,Len(AllTrim(mCode)))+;
          Space(6-Len(AllTrim(mCode)))
          MDESC:=SPACE(30); nRow++
          IF nRow == 20; nRow := 5; @ 5,1 CLEAR TO 19,77; ENDIF
          @ 22,1 Clear to 22,77
       ELSE
          LOOP
       ENDIF
    ELSE
       @ 22,44 SAY &(FIELD(3))
       DelRecord()
       packfiles(2)
      @ 22,1 Clear to 22,77
   ENDIF
ENDDO
*
PROC CODESCROLL(aCodeType)
LI := 5; MSN := 1;Sno:=1
@ 5,1 CLEAR TO 19,77
If Empty(aCodeType); Return; EndIf
For I = 1 To Len(aCodeType)
   aCodeType[I] += 4
Next
ASORT(aCodeType)
For CodeCnt = 1 To Len(aCodeType)
C := Chr(aCodeType[CodeCnt]+64)
DBGOTOP(); LOCA FOR CODE = C
mExtHead := ""
IF !FOUND();LOOP;ENDIF
sp := CodeSpace(C)
CodeExt(sp,@mExtHead)
DispCodeHead(4,Sp,mExtHead)
DO WHILE !EOF() .AND. CODE = C
   mTitle := "RLMN_"+ALLTRIM(Str(ASC(C)-64))
   @ 3,1 Say [CODE-TYPE: ]+CID(aCODE(),C,2,3)
   IF CODE = C
      @ LI++,1 SAY CodeLine(sp)
      IF LI > 22
         LI := 5
         OPTION = LOTUSMENU(24,10,[ NEXT-PAGE ],[ EXIT ])
         IF OPTION = 2 .OR. OPTION = 0
            RETURN
         ENDIF
         @ 5,1 CLEAR TO MaxRow()-2,77
      ENDIF
   ENDIF
   SKIP
ENDDO
OPTION = LOTUSMENU(24,10,[ NEXT-CODE-TYPE ],[ EXIT ])
IF OPTION = 2 .OR. OPTION = 0
   RETURN
ENDIF
MSN := 1; LI := 5; @ 5,1 CLEAR TO MaxRow()-2,77
Next
a := BlinkOn()
ENDOFSTM(24,[END OF LIST....PRESS ANY KEY TO EXIT])
BlinkOff(a)
*
Func CodeLine(sp)
Return ( SPACE(sp[1])-STR(SNo++,6)+SPACE(sp[2]-2)+PADR(ALLTRIM(CODENAME),
9)+SPACE(sp[3]-2)+AllTrim((&(FIELD(3))))+CodeExt(sp))
```

```
Local mExt := ""
Do Case
   Case C = "A" .or. C = "B" .or. C = "C"
   mExtHead := mExt := ""
EndCase
Return(mExt)
*
FUNC CODETYPE(Action,Flag)
Local aActionDesc := {[ *** Code Will Be Added *** ],;
                      [ *** Code Will Be Changed *** ],;
                      [ *** Code Will Be Deleted *** ],;
                      [ *** Codes Listing *** ]}
local mList := {}
Default Flag To .F.
SETCURSOR(3)
FOR I = 1 TO 3 //LEN(aCode())
   AADD(MLIST, CHR(I+64)+[ - ]+CODEid[I,3])
NEXT
If Flag .And. Action = 5
   Return( TagArray(mList,"Mark desired code(s):") )
Endif
If Action < 4
   Option:=0
   aa:=SaveScreen(0,0,maxRow(),MaxCol())
   OPTION:=MCHOICE(mList,4,30,nil,nil,"[ SELECT: ]")
   SS_SLIDELEFT(0,0,MaxRow(),MaxCol(),aa)
   If Option ==0; Return("Dummy"); EndIf
Else
   Option := TagArray(mList,"Mark desired code(s):")
EndIf
Cls(8,Chr(177))
xbxx(1,1,MaxRow()-2,MaxCol()-2,standard(),3,SLS_shadatt(),'          ',;
76)
@ 1,3 Say PadC(Stretch("Stock Control Codes File"," ",1),75)
a:=BlinkOn()
@ 3,45 say PadL(aActionDesc[Action],33)
BlinkOff(a)
If Action < 4
   DispCodeHead(4,{18,4,4})
   DispCodeHead(21,{18,4,4})
   @ 3,3 Say [CODE-TYPE: ]+SUBS(MLIST[OPTION],5,30)
EndIf
RETURN( If(Action<4,CODEID[OPTION,2],Option) )
*
PROC CODEPRN(Action)
aCodeType := CodeType(Action,.T.)
IF !PrintSet(); RETURN; ENDIF
QAZ:={}
For I = 1 To Len(aCodeType)
  aCodeType[I] +=4
Next
ASORT(aCodeType)
For CodeCnt = 1 To Len(aCodeType)
MSN := 1
C := Chr(aCodeType[CodeCnt]+64)
DBGOTOP(); LOCA FOR CODE = C
IF !FOUND();LOOP;ENDIF
mExtHead := ""
sp := CodeSpace(C)
CodeExt(sp,@mExtHead)
mTitle := PadC([*** ]+QWERTY(Chr(aCodeType[CodeCnt]+64) )+;
[ ***], 77)
CodeHead(mTitle)
DO WHILE !EOF() .AND. CODE = C
```

```
   IF CODE = C
     ? CodeLine(sp)
     IF PRNSTOP(); RETURN; ENDIF
     IF PrinterRow() > PageLength-1  .AND. !EOF(); CODEHEAD(mTitle);
     ENDIF
   ENDIF
   DBSKIP()
ENDDO
IF EOF()
 EXIT
ENDIF
Next
PrintEnd()
*
PROC CODEHEAD(LineHead)
Local aSp := PrnCodeHead(Sp,mExtHead,.T.)
EJECT()
set date format to "dd-mm-yyyy"
? Space(aSp[2])+PADC(CN,aSp[1]); ?
? Space(aSp[2])+PADC([STOCK CONTROL:- ]+AllTrim(LineHead),aSp[1]);?
? Space(aSp[2])+PADR([Page: ]+NUMTOSTR(PAGE++,4)+Space(aSp[1]-26)+;
[Date: ]+PADL(DTOC(DATE()),10),aSp[1])
? Space(aSp[2])+REPL([-],aSp[1])
PrnCodeHead(Sp,mExtHead)
? Space(aSp[2])+REPL([-],aSp[1])
*
Proc DispCodeHead(R,Sp,Ext)
Default Ext to ""
Inverse()
@ R,1 SAY Padr(SPACE(Sp[1])+[S/NO]+Space(sp[2])+[<CODE>]+;
Space(sp[3])+[<---------DESCRIPTION-------->]+Ext,77)
Normal()
*
Func PrnCodeHead(Sp,Ext,Flag)
Default Ext to "", Flag to .f.
If !Flag
   ? Padr(SPACE(Sp[1])+[S/NO]+Space(sp[2])+[<CODE>]+;
   Space(sp[3])+[<---------DESCRIPTION-------->]+Ext,77)
Else
   mTmp := Len(Alltrim( Padr(SPACE(Sp[1])+[S/NO]+Space(sp[2])+;
   [<CODE>]+Space(sp[3])+[<---------DESCRIPTION-------->]+Ext,77)))
   Return( {mTmp,Sp[1]} )
Endif
*
Func CodeSpace(C)
Do Case
   Case C = "A" .or. C = "B" .or. C = "C"
     aSp := {19,4,4}
   Otherwise
     aSp := {19,4,4}
EndCase
Return(aSp)

FUNC QWERTY(U)
DO CASE
 CASE U = "E"
  ZXCV:="Location Code"
 CASE U = "F"
  ZXCV:="Suppliers Code"
 CASE U = "G"
  ZXCV:="Department Code"
ENDCASE
```

```
PROGRAM : PRL.PRG
AUTHOR  : AHMED TIJJANI IBRAHIM
*
#Define WKEYLOGIC      1
#Define WKEYCODE       2
#Define RLNDX          5
#define RECPICT       19
#define SUMPICT       20
#define CNTPICT       21
#define NREPLEVEL_1    1
#define NREPLEVEL_2    2
#define NREPLEVEL_3    3
*
Func aCODE(xlFlag)
Default xlFlag to .f.
if xlFlag
  Return(      { {"ORGCODEID",   "A","Clothing & Accessories Items"},;
                 {"BANKCODEID",  "B","Equipment & Training Materials"},;
                 {"POSTCODEID",  "C","Accommodation & General Store"},;
                 {"EMOCODEID",   "D","Publication & Stationery Store"},;
               } )
else
  Return(      { {"STAFCATEID", "E","Location Codes"}, ;
                 {"PAYMODEID",   "F","Supplier Codes"}, ;
                 {"DEPARTID", "G","Department Codes"}, ;
               } )
endif
*
Func Papasswd()
Local aList := {[A - MAIN MENU                    ],;
                [B - SUPERVISOR'S KEY             ],;
                [R - DATA FILES BACKUP            ],;
                [S - DATA FILES RESTORE           ],;
                [V - REMOVE MARKED RECORDS        ]}
Return(aList)
*
Proc CodesArrays(lFlag)
If lFlag
   Publ LocationCode:={},SupplierCode := {},ItemCode:={},;
   DeptCode:={},MoveCode:={}
   LOADCODES()
Else
   Release LocationCode, SupplierCode, ItemCode,DeptCode,MoveCode
EndIf
Return
*
PROC PERCENT(SOFAR,TOTAL,RECID)
SET CONSOLE ON
PERCENT = (SOFAR/TOTAL) * 100
PERCTADJ = PERCENT * 0.60
@ 21,02 SAY REPL(CHR(254),PERCTADJ)+REPL(CHR(250),60-PERCTADJ)
@ 15,01 SAY TRAN(SOFAR,[##,###.##])
@ 15,16 SAY TRAN(TOTAL,[##,###.##])
@ 21,66 SAY TRAN(PERCENT,[###.##])
INVERSE()
IF RECID != NIL; @ 13,15 SAY RECID; ENDIF
@ 23,43 SAY Time12()
NORMAL()
SET CONSOLE OFF
RETURN
*
```

```
//  Return(.f.)
//EndIf
DO CASE
CASE X = 1;  Aadd(aTmp,USEFILE("BINCARDS"))
CASE X = 2;  Aadd(aTmp,USEFILE("CODEFIL"))
CASE X = 3;  Aadd(aTmp,USEFILE("movement"))
CASE X = 4;  Aadd(aTmp,USEFILE("movehist"))
ENDCASE
Return( If(Ascan(aTmp,.f.)#0,.f.,.t.) )
*
FUNCTION FILETOCLOS(X)
DO CASE
CASE X = 1;  CLOSEAREA("BINCARDS")
CASE X = 2;  CLOSEAREA("CODEFIL")
CASE X = 3;  CLOSEAREA("movement")
ENDCASE
RETURN("")
*
Proc PayAudit(AuditRec)
Local a:=Select()
Select("PayAudit")
If AppendLock()
   For I = 1 To Len(AuditRec); Repl &(Field(I)) with AuditRec[I]; Next
   Release()
EndIf
Select(A)
Return
*
Proc Removedele
Local MLIST:={"A - Bincards File",;
              "B - Codes File",;
              "C - Movement File"}

OPTION:=CHOICELIST(3,45,19,75,MLIST,"[REMOVE RECORDS:]")
PACKFILES(OPTION)
REORG()
Return
*
PROC REORG
LOCAL mTiTle := "[ Progress Monitor ]"
Close DataBases
ERASEFILES(MDATAPATH,[*.CDX])
SET SAFETY OFF
USE CODEFIL Exclusive
IndexFile(MDATAPATH+[CODEFIL],"Code","CODE+CODENAME",.T.,.T.,mTitle)
IndexFile(MDATAPATH+[CODEFIL],"CodeNAME","CODENAME",.F.,.T.,mTitle)
USE;USE BINCARDS Exclusive
IndexFile(MDATAPATH+[BINCARDS],"DocumentNo","DocumentNo",.T.,.T.,mTitle)
IndexFile(MDATAPATH+[BINCARDS],"ITEMNO","ITEMNO",.t.,.T.,mTitle)
IndexFile(MDATAPATH+[BINCARDS],"NONITEMNO","ITEMNO",.f.,.T.,mTitle)
IndexFile(MDATAPATH+[BINCARDS],"LOcation","ITEMNO+Location",;
.f.,.T.,mTitle)
USE;USE movement Exclusive
IndexFile(MDATAPATH+[movement],"DOCUMENTNO","DOCUMENTNO+;
entryno",.T.,.T.,mTitle)
USE;USE movehist Exclusive
IndexFile(MDATAPATH+[movehist],"DOCUMENTNO","DOCUMENTNO+;
entryno",.T.,.T.,mTitle)
use
SET SAFE ON
CLOSE DATA
```

```
DO CASE
   CASE OPTION = 1; PACK([bincards])
   CASE OPTION = 2; PACK([CODEFIL])
   CASE OPTION = 3; PACK([movement])
   CASE OPTION = 4; PACK([movehist])
ENDCASE
Return
*
PROC P_DUPDELE
local aF := {}
IF !MESSYN([FILE CLEANING OPERATION. DO YOU WANT TO CONTINUE ?])
   RETURN
ENDIF
Aadd(aF,1);  Aadd(aF,2); DupDele(aF,[bincards]); Asize(aF,0)
Aadd(aF,1); Aadd(aF,2);  DupDele(aF,[CODEFIL]) ; Asize(aF,0)
Aadd(aF,1); Aadd(aF,2);  DupDele(aF,[movement]) ; Asize(aF,0)
REORG()
Return
*
Func LOADCODES(CodeId)
Local aTmp := {}
PLSWAIT(.T.,"Loading Stock Control Popup Codes... Please Wait")
lFileNotOpen := .f.
If Select("Codefil") == 0
   lFileNotOpen := .t.
   If !OPENFILES(2,2); Return(""); EndIf
EndIf
Select("CodeFil")
DbGoTop()
If CodeId # Nil
   Locate For CODE = CID(aCODE(),CodeId)
   Do While !Eof() .And. CODE = CID(aCODE(),CodeId)
      AADD(aTmp,CODENAME+[ : ]+&(Field(3)))
      Skip
   Enddo
   If lFileNotOpen; Use; EndIf
   PLSWAIT(.F.)
   Return(aTmp)
Else
  WHILE !EOF()
    DO CASE
      CASE CODE = CID(aCODE(),"STAFCATEID"); AADD(LOCATIONCODE,;
      CODENAME+[ : ]+&(Field(3)))
      CASE CODE = CID(aCODE(),"PAYMODEID"); AADD(SUPPLIERCODE,;
      CODENAME+[ : ]+&(Field(3)))
      CASE CODE = CID(aCODE(),"DEPARTID"); AADD(DeptCode,;
      CODENAME+[ : ]+&(Field(3)))
    ENDCASE
    SKIP
  ENDDO
EndIf
PLSWAIT(.F.)
RETURN
/*
If lFileNotOpen; Use; EndIf
lFileNotOpen := .f.
If Select("SalTable") == 0
   lFileNotOpen := .t.
   If !OPENFILES(5,5); Return(""); EndIf
EndIf
While !Eof()
```

```
Skip
End
If lFileNotOpen; Use; EndIf
PLSWAIT(.F.)
Return("")
*/

Func LastRecord(cFile)
Local AA := Select(), lTmp := .F.
Default cFile To Alias()
Select(cFile)
DbSkip()
If Eof()
   lTmp := .t.
EndIf
DbSkip(-1)
Select(aa)
Return (lTmp)

Function Loader()
openfiles(1,1)
selectfile([bincards],[itemno])
DbGotop()
do while !eof()
   if .not. empty(desc)
     AADD(ItemSample,ITEMNO+[ : ]+&(Field(2)))
   endif
   dbskip()
enddo
return (ItemSample)

function UpdateStock()
local Flag:=.t.,temp:={},nRecord,nCurrentRecord,nCount:=0
public mItemNo,mIssued
plswait(.t.,"Please wait... Opening Data Files")
openfiles(1,1);openfiles(3,4)
plswait(.f.)
selectfile([movement],[DocumentNo])
DbGoTop()
if Reccount() = 0
  Buzz()
  msg([Sorry, No Records to Post])
  return
endif
ProgOn("[ Progress Monitor ]")
do while .not. eof()
  nRecord:=RecNo()
  mItemNo:=ItemCode
  mIssue:=Issued
  selectfile([bincards],[nonitemno])
  if DbSeek(mItemNo)
    nCurrentRecord:=RecNo()
    if QuantityOk(@mItemNo,@mIssue)
      DbGoto(nCurrentRecord)
      do while .not. eof()
        if mItemNo == bincards->ItemNo
          if Bincards->Balance # 0
            if Bincards->Balance >= mIssue
              mBalance:=Bincards->Balance-mIssue
              repl Bincards->Balance with mBalance
              PostBlock()
              flag:=.f.
              mIssue:=Abs(Bincards->Balance-mIssue)
              repl Bincards->Balance with 0
            endif
          endif
        endif
        DbSkip()
      enddo
    endif
  endif
  selectfile([movement],[DocumentNo])
  DbGoto(nRecord)
  if !Flag
    DbDelete()
  endif
  PROGDISP(++nCOUNT,LASTREC(),{|| ALLTRIM(STR(nCOUNT))+[ Of ]+;
  ALLTRIM(STR(LASTREC()))},{|| INKEY()#27} )
  DbSkip()
enddo
ProgOff()
selectfile([movement],[DocumentNo])
pack([Movement])
set date format to "dd-mm-yyyy"
if RecCount() # 0
  buzz()
  IF multimsgyn({"    Some Items were not Posted,",;
                 " Because their Stock Balances were",;
                 "    Less than what is Requested.",;
                 " ",;
                 "     Do you want to view them ?"},"   Yes   ",;
                 "   No   ")
    aFlds := {"DocumentNo","EntryNo","Date","Department","ItemCode",;
    "Quantity","Issued","UnitCost","Movement"}
    aDesc := {"Document #","Entry #","Date","Department","Item #",;
    "Quantity","Issued","Unit Cost","Movement Code"}
    editdb(.f.,aFlds,aDesc,.t.,.F.)
  endif
endif
return .t.

Function QuantityOk(m,n)
local k:=0
do while .not. eof()
  k=k+Balance
  DbSkip()
  if AllTrim(Bincards->ItemNo) == AllTrim(m)
    loop
  else
    exit
  endif
enddo
return ( if(k >= n,.t.,.f.))

Function PostBlock()
select movehist
DbAppend()
repl EntryNo with Movement->EntryNo
repl DocumentNo with Movement->DocumentNo
repl Date with Movement->Date
repl Department with Movement->Department
repl ItemCode with Movement->ItemCode
repl Quantity with Movement->Quantity
```

```
Function StockSummary(bnm)
public reply,nCategory:=""
reply:=bnm
Plswait(.t.)
DbCloseAll()
OPENFILES(1,1)
Plswait(.f.)
TotalValue:=TotalIssue:=TotalQtyValue:=IssueValue:=TotalQuantity:=0
aCodeType := XCodeType()
IF !PrintSet(); RETURN; ENDIF
ASORT(aCodeType)
SELECTFILE([BINCARDS],[nonITEMNO])
For CodeCnt = 1 To Len(aCodeType)
MSN := 1
C := Chr(aCodeType[CodeCnt]+64)
DBGOTOP(); LOCA FOR FLAG = C
IF !FOUND();LOOP;ENDIF
BalHead(PadC(vStockExtra(c)+IF(reply=2,[ Balance],[ Movement])+;
[ Report],if(reply=2,80,132)))
m->desc:="1234567890"
GrandQuantity:=GrandQtyValue:=GrandIssue:=GrandIValue:=;
GrandBalance:=GrandValue:=0
do while !EOF()
  IF FLAG=C
    m->ItemNo:=Bincards->ItemNo
    m->desc:=if(!empty(Bincards->desc),Bincards->desc,m->desc)
    TotalQuantity:=TotalQuantity+Quantity
    TotalQtyValue:=TotalQtyValue+(Quantity*UnitCost)
    if Balance # 0
      TotalIssue:=TotalIssue+(Quantity-Balance)
      if Quantity = balance
        TotalValue:=TotalValue+(UnitCost*Balance)
      else
        TotalValue:=TotalValue+(UnitCost*(Quantity-Balance))
        IssueValue:=IssueValue+(UnitCost*(Quantity-Balance))
      endif
    else
      TotalIssue:=TotalIssue+Quantity
      IssueValue:=IssueValue+(UnitCost*Quantity)
    endif
    DbSkip()
    if m->ItemNo == Bincards->ItemNo
      loop
    else
      TotalBalance:=TotalQuantity-TotalIssue
      GrandQuantity+=TotalQuantity
      GrandQtyValue+=TotalQtyValue
      GrandIssue+=TotalIssue
      GrandIValue+=IssueValue
      GrandBalance+=TotalBalance
      GrandValue+=(TotalQtyValue-IssueValue)
    ENDIF
    ? str(sno++,4)+space(1)+m->ItemNo+space(1)+m->Desc+space(3)+;
    if(REPLY=2,"",trans(TotalQuantity,"9,999,999")+space(1)+;
    trans(TotalQtyValue,"999,999,999.99")+space(5)+;
    trans(TotalIssue,"9,999,999")+space(1)+trans(IssueValue,;
    "999,999,999.99")+space(5))+trans(TotalBalance,"9,999,999")+;
    space(1)+trans(TotalQtyValue-IssueValue,"999,999,999.99")
    TotalValue:=TotalIssue:=TotalQtyValue:=IssueValue:=TotalQuantity:=0
    IF PRNSTOP(); RETURN; ENDIF
    IF PrinterROW() > PageLength-1
  ELSE
    DBSKIP()
  ENDIF
ENDDO
IF PrinterROW() > PageLength
  eject()
endif
? REPL([-],if(REPLY=2,80,132))
? SPACE(6)+[Total ==========>]+space(26)+if(REPLY=2,"",;
trans(GrandQuantity,"9,999,999")+space(1)+;
trans(GrandQtyValue,"999,999,999.99")+space(5)+;
trans(GrandIssue,"9,999,999")+space(1)+trans(GrandIValue,;
"999,999,999.99")+space(5))+trans(GrandBalance,"9,999,999")+;
space(1)+trans(GrandValue,"999,999,999.99")
? REPL([-],if(REPLY=2,80,132))
next
USE
PRINTEND()
RETURN .T.
*
PROC BALHEAD(ASDFG)
EJECT()
set date format to "dd-mm-yyyy"
? PADC(CN,if(REPLY=2,80,132));?
? ASDFG;?
? PADR([Page: ]+NUMTOSTR(PAGE++,4)+if(REPLY=2,space(80-53),;
Space(132-55))+PADL("Date : "+FullDate(DATE()),45),if(REPLY=2,80,132))
? REPL([-],if(REPLY=2,80,132))
if REPLY=1                                                  ;
  ? [                                                          ======= ;
  RECEIPT =======      ======= ISSUED ========      ======= ;
  BALANCE =======]
  ? [S/No Item #     Item Description                          Quantity;
            Value      Quantity           Value      Quantity;
                      Value]
else
  ? [S/No Item #     Item Description                          Quantity;
           Value]
endif
? REPL([-],if(REPLY=2,80,132))

*
Function vstockextra(mlFlag)
  do case
   case mlFLAG == "A"
     nCategory := "Clothing & Accessories Store"
   case mlFLAG == "B"
     nCategory := "Equipment & Training Material"
   case mlFLAG == "C"
     nCategory := "Accommodation & General Store"
   case mlFLAG == "D"
     nCategory := "Publication & Stationery Store"
  endcase
return (nCategory)

function ReversePosting()
local Flag:=.t.,temp:={},nRecord,nCurrentRecord,nCount:=0
public mItemNo,mIssued:=0,k
plswait(.t.,"Please wait... Opening Data Files")
openfiles(1,1);openfiles(3,4)
plswait(.f.)
StockView()
```

```
  msg([Sorry, No records that match filter condition])
  return
endif
selectfile([movement],[DocumentNo])
DbGoTop()
ProgOn("[ Progress Monitor ]")
do while !eof()
  nRecord:=RecNo()
  mItemNo:=ItemCode
  mIssued:=Issued
  selectfile([bincards],[nonitemno])
  DbGoTop()
  if DbSeek(mItemNo)
    nCurrentRecord:=RecNo()
    k:=1
    do while .not. eof()
      DbSkip()
      if AllTrim(mItemNo) == AllTrim(bincards->ItemNo)
        k++
      else
        exit
      endif
    enddo
    for a = 1 to k
      DbSkip(-1)
      if Bincards->Quantity = Bincards->Balance
        loop
      else
        mIssued:=(Bincards->Quantity-(Bincards->Balance+mIssued))
        if mIssued = 0
          Repl Balance with Quantity
          exit
        else
          if mIssued < 0
            Repl Balance with Quantity
            mIssued:=abs(mIssued)
          else
            Repl Balance with mIssued
            exit
          endif
        endif
      endif
    next
    selectfile([movement],[DocumentNo])
    DbGoto(nRecord)
  endif
  PROGDISP(++nCOUNT,LASTREC(),{|| ALLTRIM(STR(nCOUNT))+[ Of ]+;
  ALLTRIM(STR(LASTREC()))},{|| INKEY()#27} )
  DbSkip()
enddo
ProgOff()
REORG()
return .t.

function StockView()
public StockFrom:="000000000",StockTo:="999999999",DocFrom:=DATE()
public DocTo:=Date(),EF:="0000",ETo:="9999",reply:=1
SET DATE FORMAT TO "DD-MM-YYYY"
POPREAD(.f.,"Document Number From   :",@StockFrom,"@!",;
            "Document Number To     :",@StockTo,"@!",;
            "Document Date From     :",@DocFrom,"@D",;
            "Document Date To       :",@DocTo,"@D")
```

```
do while !eof()
  Plswait(.t.,"Please wait... Building Database information")
  if DocumentNo >= StockFrom .and. DocumentNo <= StockTo ;
    .and. Date >= DocFrom .and. Date <= DocTo .AND. !DELETED()
    select Movement
    DbAppend()
    repl EntryNo with Movehist->EntryNo
    repl DocumentNo with Movehist->DocumentNo
    repl Date with Movehist->Date
    repl Department with Movehist->Department
    repl ItemCode with Movehist->ItemCode
    repl Quantity with Movehist->Quantity
    repl Issued with Movehist->Issued
    repl Movement with Movehist->Movement
    select MoveHist
    DbDelete()
    reply:=2
  endif
  DbSkip()
enddo
IF REPLY=2
  PACKFILES(4)
ENDIF
return (reply)
```

Ahmed Tijjani Ibrahim

Clothing & Accessories Store Movement Report

Page: 0001 Date : Thursday November 16, 2000 5:58

			======= RECEIPT =======		======= ISSUED =======		======= BALANCE ===	
S/No	Item #	Item Description	Quantity	Value	Quantity	Value	Quantity	V
1	1234	CAMOUFLAGE	23	2,300.00	0	0.00	23	2,30
2	1235	OG GREEN	34	782.00	0	0.00	34	78
3	456	LANTERN	1	150.00	0	0.00	1	15
4	457	MOUSE	5	2,500.00	0	0.00	5	2,50
	Total ==========>		63	5,732.00	0	0.00	63	5,73

```
                    Ahmed Tijjani Ibrahim

            Clothing & Accessories Store Balance Report

Page: 0001                    Date : Thursday November 16, 2000   6:01PM
-------------------------------------------------------------------------
S/No Item #     Item Description                  Quantity        Value
-------------------------------------------------------------------------
   1 1234       CAMOUFLAGE                              23      2,300.00
   2 1235       OG GREEN                                34        782.00
   3 456        LANTERN                                  1        150.00
   4 457        MOUSE                                    5      2,500.00
-------------------------------------------------------------------------
     Total ==========>                                  63      5,732.00
-------------------------------------------------------------------------
```

Ahmed Tijjani Ibrahim

Equipment & Training Material Movement Report

Page: 0002 Date : Thursday November 16, 2000 5:5

			======= RECEIPT =======		======= ISSUED =======		======= BALANCE ==	
S/No	Item #	Item Description	Quantity	Value	Quantity	Value	Quantity	
5	121	BINOCULARS	23	1,288.00	20	1,120.00	3	1
6	122	COMPASS	34	26,826.00	0	0.00	34	26,8
7	123	TETRON	787	618,582.00	0	0.00	787	618,5
8	126	TENT	234	184,626.00	0	0.00	234	184,6
	Total ==========>		1,078	831,322.00	20	1,120.00	1,058	830,2

Ahmed Tijjani Ibrahim

Equipment & Training Material Balance Report

Page: 0002 Date : Thursday November 16, 2000 6:01PM

S/No	Item #	Item Description	Quantity	Value
5	121	BINOCULARS	3	168.00
6	122	COMPASS	34	26,826.00
7	123	TETRON	787	618,582.00
8	126	TENT	234	184,626.00
	Total ==========>		1,058	830,202.00

Ahmed Tijjani Ibrahim

Accommodation & General Store Movement Report

Page: 0003 Date : Thursday November 16, 2000 5:!

			======= RECEIPT =======		======= ISSUED ========		======= BALANCE =:	
S/No	Item #	Item Description	Quantity	Value	Quantity	Value	Quantity	
9	987	AIRCONDITIONER	567	6,804.00	0	0.00	567	6,8
10	988	TYPEWRITER	89	8,722.00	0	0.00	89	8,7
11	989	COMPUTER	566	188,478.00	0	0.00	566	188,4
	Total ==========>		1,222	204,004.00	0	0.00	1,222	204,0

Ahmed Tijjani Ibrahim

Accommodation & General Store Balance Report

Page: 0003 Date : Thursday November 16, 2000 6:01PM

S/No	Item #	Item Description	Quantity	Value
9	987	AIRCONDITIONER	567	6,804.00
10	988	TYPEWRITER	89	8,722.00
11	989	COMPUTER	566	188,478.00
		Total ==========>	1,222	204,004.00

Ahmed Tijjani Ibrahim

Publication & Stationery Store Movement Report

Page: 0004 Date : Thursday November 16, 2000 5:5

			======= RECEIPT =======		======= ISSUED ========		======= BALANCE ==	
S/No	Item #	Item Description	Quantity	Value	Quantity	Value	Quantity	
12	876	DUPLICATING INK	90	11,880.00	5	170.00	85	11,7
13	877	EXERCISE BOOK	45	3,915.00	0	0.00	45	3,9
14	878	PHOTOSTATING INK	34	2,652.00	0	0.00	34	2,6
	Total ==========>		169	18,447.00	5	170.00	164	18,2

Ahmed Tijjani Ibrahim

Publication & Stationery Store Balance Report

Page: 0004 Date : Thursday November 16, 2000 6:01PM

S/No	Item #	Item Description	Quantity	Value
12	876	DUPLICATING INK	85	11,710.00
13	877	EXERCISE BOOK	45	3,915.00
14	878	PHOTOSTATING INK	34	2,652.00
		Total ==========>	164	18,277.00