COMPUTER – AIDED INSTRUCTION (CAI)

ON

NUMBER BASES

BY

JOSEPH ENEMONA IDOKO
PGD/MCS/181/96

A Project submitted to the Department of Mathematics and Computer Science, School of Science and Science Education, Federal University of Technology (FUT), MINNA, in partial fulfillment of the requirements for the award of Post Graduate Diploma (PGD) in Computer Science.

MARCH, 2000

CERTIFICATION / APPROVAL

This project titled "COMPUTER–AIDED INSTRUCTION ON NUMBER BASES" by JOSEPH ENEMONA IDOKO meets the regulations governing the award of POST GRADUATE DIPLOMA (PGD) in COMPUTER SCIENCE by the Federal University of Technology (FUT), MINNA, and is approved for its contribution to scientific knowledge and the study of Computer Science.

______________________ __________________

(DR. REJU S. A.) DATE

PROJECT SUPERVISOR

______________________ __________________

(DR. REJU S. A.) DATE

HOD

MATHS & COMP. SCIENCE DEPT

______________________ __________________

EXTERNAL EXAMINER DATE

DEDICATION

This project is dedicated to my MUM, who nursed me for bearly one year and passed on, and DAD, who then acted as both father and mother but also passed on when I was fourteen years old. May their souls rest with the Lord.

ACKNOWLEDGEMENT

To God Almighty be glory and honour for His infinity goodness, love, mercy and protection over my life and for the successful completion of this programme.

No individual can achieve anything good in life without some sort of assistance or support materially, morally or otherwise from others. I am therefore, so much indebted to the following and others who are too numerous to mention here.

My sincere thanks to Dr. S. A. Reju, who doubles as the HOD and my supervisor in the Department of Mathematics and Computer Science, Federal University of Technology (FUT), Minna for his good advice, tolerance, consideration and reception throughout the long period of this project work. I am equally grateful to all the staff of the Department, especially Mr John Echoga, my host in Minna, who was once my student and then my lecturer.

I am very grateful to the Authorities of Kaduna Polytechnic, Kaduna for allowing me undertake this course of study and I also thank the HOD and my other colleagues in the Department of Mathematics and Computer Studies for their encouragement and concern.

My profound gratitude goes to the following whose great assistance were very instrumental in the preparatory stage and the final production of this project. They include Mr. Dele Adeoye, Mr. Richard Idoko, Mr. Ajoge Sanni and Mr. Peter Ikani Idoko, who have one time and the other been students in Maths and Computer Studies Dept. or are staff of Kaduna Polytechnic, Kaduna.

I sincerely wish to acknowledge the encouragement, support and advice from some of my relations and close associates during the period of my study. They include; Barrister A.A. Idoko, Mr. J.O. Onoja, Mall. A.A Amana, Mr. S.O Paul, Mr, N.N Enebi and Engr S.O. Onoja.

Finally, my greatest thanks to my, wife, Mary and my children for their understanding, encouragement and for all that they had to sacrifice and bear during my course of study in F.U.T. Minna.

For every individual or group of individuals mentioned or not who contributed in one way or the other for my success and progress, I pray the Lord rewards them and grants them the grace to enjoy everlasting life in Jesus Christ.

ABSTRACT

In virtually every field of human endeavour today computer application is inevitable. The area of teaching and learning is not an exception in this fact.

Computer-Aided Instruction (or Computer Assisted Instruction), CAI involves the use of computer to provide either a full-scale course or a supplementary course in any given topic or subject. This can be achieved in one or all of the three forms, namely, drill-and-assistance, tutorials, and simulations. With the assistance of a well-written computer software a student goes through a drill-and-practice sequence that enables him or her learn at his/her own pace, with less difficulties or assistance from another person.

By using modular or structured programming, this project attempts to use computer to teach a learner some basic concepts and operations on Number Bases.

Conversion of fractions and Mathematical operations on fractions in different bases were not considered in the tutorial part of this work.

TABLE OF CONTENTS

CHAPTER ONE

CHAPTER TWO

CHAPTER THREE

CHAPTER FOUR

CHAPTER FIVE

CHAPTER ONE

1.0 INTRODUCTION TO COMPUTER-AIDED INSTRUCTION

1.1 INTRODUCTION

The art and science of teaching (or instructing) and learning are as old as man himself on earth. Each of these can be achieved in many different ways, formally or informally, sometimes intentionally or incidentally, and mostly between a teacher (or an instructor) and a learner (or student).

Instruction is an undertaking whose purpose is to help people learn. It is a set of events that affects learners in such a way that learning is facilitated. Instruction may include events that are generated by a page of print, by a picture, by a T.V. programme or by a combination of some physical objects, among other things.

Modern techniques and developments have contributed immensely to instructional processes. Computer-aided instruction is aimed at providing effective instruction, utilising the seemingly unlimited data processing powers of the electronic computer and in overcoming some of the deficiencies of instructions associated with conventional classroom teaching. The revolutionary impact of the new technology even leads to the erroneous impression by some people that the human teacher may eventually be replaced or put out of job.

1.2 INSTRUCTIONAL AND LEARNING PROCESSES

Instruction is a deliberately arranged set of events designed purposely, however it may be done, to support learning processes.

The ultimate goal of all teaching or instructional activities is learning. Learning can be described as a relatively permanent change in behaviour arising from experience. This occurs when an individual's behaviour has been influenced, modified or changed in such a way as to make the individual act or think differently as a result of his/her newly acquired skill, knowledge, fact, principle and information.

According to Robert M. Gagne (1988), learning can be described as a change in human disposition or capability which persists over a period of time, and which is not simply ascribable to processes of growth. The kind of change called learning exhibits itself as a change in behaviour, and the inference of learning is made by comparing what behaviour was possible before the individual was placed in a "learning situation" and what behaviour can be exhibited after such treatment.

Theoretically, every good learning environment is expected to satisfy some requirements which include:

i. the individual is actively involved in the learning process. This facilitates learning.

ii. the learner is permitted to proceed at his/her own pace;

iii. reinforcement of learning is immediate and systematised. This results in more effective learning according to some established theories of instructions. With good software available, CAI appears to satisfy many of such requirements as above.

1.2.1 WHAT IS CAI?

CAI - Computer - Aided Instruction can be defined as the use of computer to provide either a full-scale course or a supplementary course in a given subject in one or all of the following three forms: drill - and - practice, tutorials and simulations.

CAI is a relatively new discipline in this our part of the world but it has been in existence in the United State of America for quite some time now. CAI has not been fully introduced into the Nigerian school curriculum because of some problems. These problems include the high cost of the computer systems and the non-availability of effective and efficient software developed for the African child. A quality software development for teaching is necessarily culture-bound. So software developed for American child cannot be expected to be suitable, without adaptation, for the average African child. This is because their background experiences and resources differ greatly.

EXAMPLES OF CAI

Except for some few cases of project write-ups by some students, the author found it difficult to lay hands on some examples of locally developed software for CAI and even foreign ones are not readily available. An example of a student's project on CAI is "Computer Aided Learning" by Joy Efionayi Solomon (1996) - a student project submitted in the Department of Maths and Computer Studies, Kaduna Polytechnic. The project deals on how to use computer to administer quiz to students on some elementary topics in various subjects.

1.2.2 **INSTRUCTIONAL METHODS**

The mode of instructions for achieving the teaching or instructional objectives is known as method. So instructional methods refer to the processes of and techniques of transmitting facts, skills, information and knowledge to the learners so as to facilitate the accomplishment of some set objectives. The instructional methods are many and vary in nature and CAI is one of the modern ones.

For effective learning to take place, Obiwusi (1981) contended that a good teaching (or instructional) method should have, among others, the following characteristics:

1. It must proceed from known to the unknown, that's from simple (concrete) to complex (general).
2. A good method should help the students (learners) to make discoveries and solve problems on their own.

3. It must be related to "real-life" so that students can see that their work is beneficial and a real achievement.

4. It should provide students ample opportunity to participate actively in the lesson. It should satisfy their basic need for activity.

5. It should be related to the subject matter in question.

6. It should allow for learners' individual difference, for instance, slow learners going at their own pace just like the fast learners.

7. It should make learning more effective and prevent fatigue and boredom.

8. It should appeal to two or more senses in order to aid retention and transfer of learning.

9. It must satisfy the needs and dominant interests of the learners.

10. It should be structured around learners' experiences and help them understand the subject matter.

1.3 COMPUTER AS AN INSTRUCTIONAL AID

The word "computer" is becoming a daily language in every home and office because of the fact that the use of computer is becoming inevitable in every sphere of life. Simply put, computer is that machine that carries out arithmetic and logical operations upon command. But conventional usage restricts the term computer to that class of machines that not only performs arithmetic and logical operations on command but also has facilities for being programmed in one of the programming languages. Computer is particularly useful to people because of its ability to perform tasks with a high degree of speed and accuracy.

Presently, there are many educational problems of great magnitude which the traditional methods of teaching and learning alone cannot solve. Prominent among these problems, according to Adewoyin (1991), are:

1. School population explosion
2. Shortage of fund
3. Knowledge (information) explosion
4. Shortage of qualified teachers and,
5. Inadequate school facilities

Hence, one should also look towards the technological media, especially the use of computers and make use of their potentialities. Through this, students in various spheres of education, be it primary, secondary or tertiary, can share valuable experiences relevant to their level with immediate impact, irresistible attention and in similar and realistic environment.

To achieve the above objectives, one of the best ways is by computer-aided instruction and learning. According to Osisiogu U.A. (1987) in "Computer Aded

Instruction in Science and Technology", the major attributes of a computer as an instructional aid includes the following:

(a) It can create learning situations that cannot be realised by the conventional teaching methods;

(b) it can be used to present information in a variety of ways that can best meet particular learning objectives;

(c) It can make learning more effective by increasing the realism, the dynamism and the attitudinal impact of information; it can thus increase the motivation to learn;

(d) It can allow for individualisation of curricula by presenting the same information, or variations of it, to different students at different times;

(e) It can, in some cases, be employed to direct information to some people in shorter-than conventional periods of time.

(f) The use of computers forces educators to examine their goals and objectives more closely than before (UNESCO, 1979).

Also, for any CAI package to be effective, it must meet some standards. Awele Maduemezaia (1987), quoting what Mr John K. Lanrian suggested in "Computer Aided Instruction in Science and Technology", said that, "a good CAI package must conform with the following:

1. the user must be able to use less time and have a better retention than when using any other medium;

2. it must be self-motivated and self-directed to each individual's responsiveness, and there must be academic respectability. It must not be boring for the intelligent or difficult to use or understand for the less privileged;

3. it should be such as can be used as an integral part of the classroom course or as a supplement to the individual's own effort to study;

4. the target user (or a range of users) must be well defined, for the educational usefulness of any of such package will depend on the statistics of its use. The assessment of attitudes of teachers and students will eventually determine the life-span of the usefulness of the package;

5. though simulation is best written in BASIC through the use of the capabilities provided by the language of the host machine, cost and transportability could not totally be sacrificed to the computing system. Producers should have this in mind so that the objective of making CAI packages available to a wide-range of compatible machines will not be completely forgotten. This will also help in keeping an eye on the running cost of maintenance at every stage".

1.4 ADVANTAGES AND DISADVANTAGES OF CAI

1.4.1 Advantages

The following are some of the advantages of CAI for teaching/learning activities:

(a) If a learner is slow in learning and the teacher does not have the requisite experience to cope with the situation, then there is the danger of fear-induced psychological blockage. With a sympathetic computer in front of such learner, problems like this are eliminated.

(b) The student is offered the opportunity to learn at his/her own pace. The average or less than average student is not burdened (and indeed slowed down) by the wrath of an impatient teacher as obtained in the traditional tutorial class with a teacher. CAI provides remedial instruction for those who need it and provides realistic interaction between the learner and the learning task.

(c) The extraordinary versatility and speed of the computer enables it to be integrated into most instructional activities. It can play a specific role as demonstrator, illustrator, graphic artist, information retriever and personal tutor. It can offer assistance in learning at all the age and ability levels of learning institutions.

(d) With good software, reinforcement of learning is immediately and systematically offered and this results in effective learning.

(e) The learner is directly and actively involved in the learning process and by this, learning is facilitated.

(f) The computer is a very good and useful private tutor. Unlike the human teacher, it is infinitely patient and permits the learner to make mistakes without necessarily impairing his/her self concept.

1.4.2 Disadvantages

Whatever has advantages, it is said, has disadvantages. Even with the many positive attributes of CAI, there are some few negative ones which include the following:

(a) Computers, which are the major instruments of CAI, are very expensive and are out of reach of many schools and individuals. Only very few individuals possess them.

(b) Every effective and efficient software developed for teaching is necessarily culture-bound. Thus, software developed for the American child, for instance, cannot be expected to be suitable, without adaptation, for the average African child. Their background experiences and their resources differ markedly. So, software development must be matched with the local environment.

(c) Availability of inappropriate software or poor quality ones can have frustrating negative effect. A microcomputer is only as good as the software available for it.

(d) The cost of proper installation, management and high levels of maintenance of the computers may not be easy to bear.

(e) A computer is just an instructional aid; not a substitute for the teacher, so, for instance, paper and biro will still be needed.

1.5 PURPOSE OF THIS PROJECT

The main purpose or immediate objective of this project is to develop a computer software program that can be used by a learner (who may know little or nothing about number bases) to learn and understand some basic concepts and arithmetic operations on number bases. Specifically, this project work is aimed at achieving the following objectives:

1. To teach the user number bases with less or no human assistance;
2. To encourage the user on the use of computer to solve other problems;
3. To involve the learner actively in the learning process which really facilitates learning;

4. The individual learner is allowed to progress at his/her own pace;

5. The learner, during his/her recreational or remedial period, can use this to improve on his/her understanding on the subject matter;

6. The learner will be able to expose himself/herself to areas which may have not been covered in the conventional class setting for some reasons.

This software should be very useful especially to students in the senior classes of the post-primary institutions and those in the preliminary or first year in the post-secondary institutions where concepts of number bases are taught and used, if they have access to computers.

CHAPTER TWO

2.0 NUMBER SYSTEMS

2.1 INTRODUCTION

A number system is a set of symbols and rules intended to represent numbers. It is like a language system consisting of ordered set of symbols called digits with rules of addition, multiplication, and any other mathematical operation.

The number systems may be classed into two broad groups:- positional and non-positional. The oldest of all, the non-positional number systems have nearly completely been replaced with systems using positional notation, that is, every digit in a multi-digit word has a weight attached to it. For example, the decimal number 3856 can be expressed as the sum of weighted coefficients:

$3856 = 3\times10^3 + 8\times10^2 + 5\times10^1 + 6\times10^0$.

2.2 NUMBER BASES

2.2.1 The Denary Number System (Number Base Ten)

From early childhood, most individuals are trained to represent quantities in the decimal (denary) number system. This system counts in units of tens and was probably developed because man has ten fingers. The base (also called radix) of this number is ten, since ten basic digits (or symbols) are used. The basic digits used are 0, 1, 2, ..., 8 and 9. By giving to these digits different positions, or weights, we can express numbers larger than 10. In positional notation, a number is represented by a sequence of digits. In general, a number $(N)_b$ in a fixed, positive, integral base (or radix) b is represented in positional notation as:

$(N)_b = (P_nP_{n-1}P_{n-2}...P_1P_0.P_{-1}P_{-2}...)$ ------ (2.1).

The number b is the base (or radix) of the system. The base is a positive integer and it is fixed throughout the system. The positional digits p_i are integers such that $0 <= P_i <= b-1$, i =..., -2, -1, 0, 1, 2, 3, ..., n. The value of each digit depends on its position in the sequence. The value of each digit in a number is known as its positional coefficient or weight.

The radix of a given number system gives the number of distinguishable symbols, or digits, used in the representation. It also gives the number by which the weight of a position in the representation must be multiplied to obtain the weight of the next more significant position. For example, 4036 is made up of:

$$\begin{array}{l}
4 \times 10^3 = 4000 \\
0 \times 10^2 = 0 \\
3 \times 10^1 = 30 \\
6 \times 10^0 = \underline{6} \\
4036
\end{array}$$

A simple decimal weighting table is given as:

.... 10^4 10^3 10^2 10^1 10^0.10^{-1} 10^{-2} 10^{-3}....

In general, in any base b system, the successive digit positions, left to right, have the weights ...b^5 b^4 b^3 b^2 b^1 b^0 ● b^{-1} b^{-2} b^{-3} b^{-4}....

The symbol `●' is called the radix point. The position of the number to the right of the radix point is called the fractional part of the number, and the position to the left is called the integral part. In the decimal number system, the radix point is called the decimal point.

There are other bases in which numbers can be expressed apart from base ten, the decimal system. In general, any number N_b that consists of n digits given in any base b can be expressed as the sum of its weighted coefficients:

$$N_b = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + a_{n-3}b^{n-3} + \ldots + a_1b^1 + a_0b^0 + a_{-1}b^{-1} + a_{-2}b^{-2} + \ldots + a_{-m}b^{-m}$$

$$= \sum_{i=-m}^{n-1} a_i b^i , \qquad \ldots\ldots \quad (2.2)$$

where a_i are the different digits in N_b and m is the number of digits after the radix point. In positional notation, the weights are only implied thus:

$$N_b = a_{n-1}\, a_{n-2}\, a_{n-3} \ldots a_1\, a_0\, a_{-1}\, a_{-2} \ldots a_{-m} \text{ - - - - - - - -(2.3)}$$

Among all other number systems, three are particularly of interest because of their importance in computer and their relationship with one another. These are the binary, octal and hexadecimal number systems.

2.2.2 The Binary Number System (Number Base Two)

Binary numbers use the base two and are often denoted by a subscript. For example, 1001_2 expresses a number in binary notation that is equivalent to the decimal 9_{10}. This system uses only 0's and 1's as digits. The rules of arithmetic operations are similar to those of the decimal system, except that the binary number system uses a set of two binary digits {0,1} instead of the set of ten decimal digits {0,1,2...,9}. The binary number system, because of the binary nature of computer circuits, has found very wide acceptance.

2.2.3 The Octal Number System (Number Base Eight)

In this system, the base $b = 2^3 = 8_{10}$; thus eight different symbols, 0 through 7, are required to express any number in base eight. So an octal number N_8 consisting of n integers and m fractional digits can be represented by the sum:

$$N_8 = a_{n-1}8^{n-1} + a_{n-2}8^{n-2} + \ldots + a_1 8^1 + a_0 8^0 + a_{-1}8^{-1}a_{-2}8^{-2} + \ldots + a_{-m}8^{-m}$$

$$= \sum_{i=-m}^{n-1} a_i 8^i \qquad \ldots\ldots \qquad (2.4)$$

2.2.4 The Hexadecimal Number System (Number Base Sixteen)

This number system has the base $b = 2^4$, thus 4 bits (binary digits) are required to express one hexadecimal digit and there are $2^4 = 16_{10}$ symbols. These symbols are 0, 1,2..., 9, A, B, ..., F; where A represents ten, B represents eleven, and so on until F which represents fifteen. The different combinations (symbols) can be represented by exactly 4 bits. Hexadecimal numbers are often identified by a subscript h rather than 16, for example the number $B7D_{16}$ is the same as $B7D_h$.

As a result of the special relationship between the binary number system and the octal and hexadecimal number systems, the octal or hexadecimal numbers can be conveniently used as alternatives for binary numbers. This is because three binary digits representing the numbers 0 to 7 can be indicated by a single octal digit; while four binary digits representing the numbers 0 to 15 can be shown by a single hexadecimal digit. This relationship is used in converting from binary system to octal or hexadecimal system or vice versa. (See 2.3.3).

2.3 CONVERSION OF NUMBER BASES

A number given in any one base can be expressed by an equivalent numerical value in another base by the application of a conversion algorithm. Any number in decimal system can be converted to an equivalent value in any other base and vice-versa. Conversion between other number bases apart from the system with base ten may not be easy without first converting to base ten. Binary code can be directly converted into octal and hexadecimal notations because the radices in the octal and hexadecimal notations are integer powers of two.

2.3.1 **Conversion From Base Ten to other Bases**

The decimal number system acts like a `bridge' when a number is to be converted from any base p to another base q, since it is easier to first convert the given number in base p to base ten before the equivalent number in base ten is then converted to base q.

Generally, the most common method of converting an integer number in base ten to another base q is as described in section 2.3.1(a) below and section 2.3.1(b) equally describes how a decimal fraction can be converted to an equivalent fractional number in the q-nary system. In each case, care must be taken when the required base is greater than ten so that every two-digit number in the base is represented by the appropriate alphabet as in the case of hexadecimal number system (section 2.2.4 above) and any other number system whose base is greater than two.

It should be noted that the conversion of integer numbers requires repeated division while that of fractional numbers requires repeated multiplication by the desired radix.

(a) **Conversion of Integers**

In order to convert an integer number, N, expressed in the radix p to a number system in the radix q, the number must first of all be converted to base ten (section 2.3.2). The equivalent number in base ten is then divided by the radix q consecutively until the last quotient is less than q. Then in the system to the radix q, the number N will be represented as an ordered set of residuals, the most significant digit (msd) in N (to the radix q) being the last quotient, followed by the last residual in the reverse order, the first residual being the least significant digit (lsd). For instance, to convert the numbers 23_{10} and 3562_{10} to base 2 and base 8, respectively, the steps are shown below:

(i) To convert 23_{10} to binary

23 / 2 = 11	Remainder 1	(lsd)
11 / 2 = 5	Remainder 1	↑
5 / 2 = 2	Remainder 1	│
2 / 2 = 1	Remainder 0	│
1 / 2 = 0	Remainder 1	(msd)

Writing down (from bottom to top) the residuals or remainders from the msd to lsd in that order, the equivalent of 23_{10} in binary is 10111. Alternatively, the working can be set out in a neater way as follows:

2	23		
2	11	R 1	Lsd
2	5	R 1	↑
2	2	R 1	↑
2	1	R 0	↑
	0	R 1	Msd

Therefore, $23_{10} = 10111_2$

(b) Similarly, 3562_{10} is converted to base 8 as follows:

8	3562		
8	445	R 2	Lsd
8	55	R 5	↑
8	6	R 7	↑
	0	R 6	Msd

The remainders from bottom to the top give 6752.

Thus, $3562_{10} = 6752_8$.

(b) Conversion of Fractions

n converting a fraction expressed in the radix p to a number system in the radix q, the fraction is consecutively multiplied by the radix q, only the fractional part being multiplied at each step. In the q-nary system, the fraction will be represented by an ordered sequence of the integer parts of the product, where the msd is the first digit of the product and the last integer product is the lsd of the converted number in the new base. An example of this is as follows:

To convert 0.8125_{10} to an equivalent number in base 2; we proceed as follows:

	Integer Part	Fractional Part
		.8125
		x $\underline{\quad 2}$
(msd)	1	.625
		x $\underline{\quad 2}$
↓	1	.25
		x $\underline{\quad 2}$
	0	.5
		x $\underline{\quad 2}$
(lsd)	1	0

Thus; $0.8125_{10} = 0.1101_2$

It should be noted that conversion of a fraction from one base to another is done, just like that of integers, through base ten. This is because conversion between any other two bases apart from base ten is not easy. Equally to be noted is the fact that not every fraction can be converted from one system into another completely. The process of multiplication can be terminated after a desired number of radix points is reached and normally a round-off error is made by cutting off **E**, where **E** is less than q^{-n}, q and n being the radix and the no. of radix points required, respectively. For instance, 0.3564_{10} is converted to an equivalent number in base 5 as follows:

Integer Part	Fractional Part
	.3564
	x 5
(msd) 1	.782
	x 5
3	.91
	x 5
4	.55
	x 5
↓ 2	.75
	x 5
(lsd) 3	.75 etc.

Thus, $0.3564_{10} = 0.1343_5$, correct to 4 radix places which has a round-off error or more accurately, $0.3564_{10} = 0.13423_5$, a non-terminating fraction in base 5 with

3 recurring. Note that 0.13423_5 is rounded up to 0.1343_5 to 4 radix places (since 3 is greater than half of 5, 1 is added to 2 to give 0.1343_5 to 4 radix places).

(c) Conversion of Mixed Numbers

When the conversion from one system into another involves a mixed number, the number is first separated into integer and fractional parts and then the conversion of each part is effected by the respective conversion algorithms and the two results are finally merged appropriately. So to convert 57.8125_{10} to an equivalent number in binary, we convert 57_{10} and 0.8125_{10} separately (as shown in 2.3.1(a) and 2.3.1(b), respectively) to their binary equivalents. Thus, $57_{10} = 111001_2$ and $0.8125_{10} = 0.1101_2$ which means that

$57.8125_{10} = 111001.1101_2$.

2.3.2 Conversion from other Bases to Base Ten

Equation (2.2) gives a general method of converting any number, N in base b to an equivalent number in base ten. This equation gives a general method of converting any mixed number in any base to base ten. If the number is wholly an integer then we have:

$N_b = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + a_{n-3}b^{n-3} + \ldots + a_1b^1 + a_0b$ --------(2.5)

and if the number is wholly a fraction with m digits in the fraction

$N_b = a_{-1}b^{-1} + a_{-2}b^{-2} + a_{-3}b^{-3} + \ldots + a_{-m+1}b^{-m+1} + a_{-m}b^{-m}$ ------(2.6)

For example, the conversion of each of:

(i) 111001.1101_2, (ii) 3241_5, (iii) 0.232_4 (iv) 2C3B9 to base ten can be carried out as follows:

(i) $111001.1101_2 = 1x2^5 + 1x2^4 + 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 + 1x2^{-1} + 1x2^{-2} + 0x2^{-3} + 1x2^{-4}$

$= 32 + 16 + 8 + 0 + 0 + 1 + 0.5 + 0.25 + 0 + 0.0625$

$= 57.8125_{10}$

(ii) $3241_5 = 3x5^3 + 2x5^2 + 4x5^1 + 1x5^0$

$= 3x125 + 2x25 + 4x5 + 1x1$

$= 375 + 50 + 20 + 1$

$= 446_{10}$

(iii) $0.232_4 = 2x4^{-1} + 3x4^{-2} + 2x4^{-3}$

$= 2x0.25 + 3x0.0625 + 2x0.015625$

$= 0.5 + 0.1875 + 0.03125$

$= 0.71875_{10}$

(iv) $2C3B9 = 2x16^4 + Cx16^3 + 3x16^2 + Bx16^1 + 9x16^0$

But $C_{16} = 12$, $B_{16} = 11$

$2C3B9 = 2x65536 + 12x4096 + 3x256 + 11x16 + 9x1$

$= 131072 + 49152 + 768 + 176 + 9$

$= 181177_{10}$

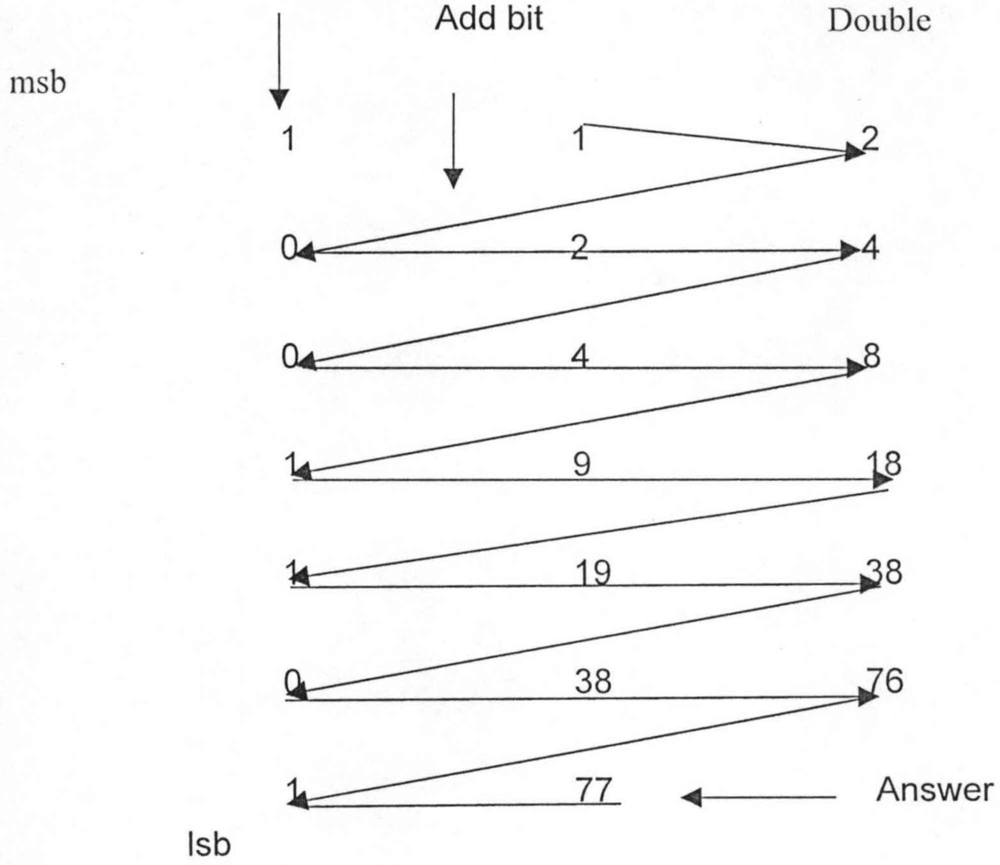
Another way to convert a binary number to a decimal number which is applicable only to integer numbers is the "double-dabble" method. It is based on the following nested representation of binary numbers:

$N_2 = \{[(a_{n-1}.2 + a_{n-2}).2 + a_{n-3}].2 + ... + a_1\}.2 + a_0$ ... (2.6)

Which is the conversion procedure. The most significant bit (msb) is doubled and then added to the coefficient of the next bit (binary digit), which may be 0 or 1. The result is doubled again, and the procedure is terminated when the least significant bit (lsb) has been finally added.

For example, the binary number 1001101 can be converted to its equivalent as follows:

Bits of Binary numbers



Above is the illustration of binary conversion using the "Double-Dabble" method. Alternatively, this can be carried out as follows:

1001101_2 .

$1 \times 2 =$ 2 (1st digit times base)

(add 0) $+\underline{0}$

$2 \times 2 =$ 4 (sum times base)

(add 0) $+\underline{0}$

$4 \times 2 = 8$ (sum times base)

(add 1) $+\underline{1}$

$9 \times 2 = 18$ (sum times base)

(add 1) $+\underline{1}$

$19 \times 2 = 38$ (sum times base)

(add 0) $+\underline{0}$

(sum times base) $38 \times 2 = 76$

(add last digit) $+\underline{1}$

Answer = 77_{10}

This method can be generalised to convert any number system (integers only) to decimal number system. The following nested representation of a number in any base b can be used:

$N_b = \{[(a_{n-1}.b + a_{n-2}).b + a_{n-3}].b + ... + a_1\}.b + a_0$ .... (2.7)

The most significant digit (msd) of the number N_b is multiplied by the base b and then added to the next digit, which can be any of the digits from 0 to b-1. The result is multiplied by b again, and the procedure is terminated when the least significant digit (lsd) has been finally added.

For example, $2C3B9_{16}$ can be converted to an equivalent number in the decimal system as follows:

$2C3B9_{16}$:
2x16 = 32 (1st digit times base)
+ $\underline{12}$ (add next digit)
44 x 16 = 704 (sum times base)
+ $\underline{3}$ (add next digit)
707 x 16 = 11312 (sum times base)
(add next digit) + $\underline{11}$
(sum times base) 11323 x 1 6 = 181168
(add last digit) + $\underline{\quad 9}$
Answer = $\mathbf{181177_{10}}$

Similarly, any number given in any base can be converted to an equivalent number in the decimal system using the above procedure.

2.3.3 CONVERSION BETWEEN BINARY AND OCTAL / HEXADECIMAL NUMBER SYSTEMS

Conversion from binary number system to Octal/Hexadecimal number systems and verse-visa plays an important part in digital computers. This is because octal and hexadecimal numbers are convenient alternatives for binary numbers. Since $2^3 = 8$ and $2^4 = 16$, three binary digits (or bits) representing the numbers 0 to 7 can be indicated by a single octal digit; while four binary digits representing the numbers 0 – 15 can be shown by a single hexadecimal digit. So, one octal digit corresponds to three binary digits (bits) and each hexadecimal digit corresponds to four bits. Thus, a binary number can be converted to octal equivalent by grouping the bits in threes from the right and converting each individual group. Similarly, the conversion to hexadecimal is achieved by splitting into four bit groups from the right and converting each individual group.

For example, 10 001 111 011 000 011_2 = 217303_8 .
and similarly, 1 0001 110 1100 0011_2 = $11EC3_{16}$.
That is 1000111101100011_2 = 217303_8 or $11EC3_{16}$
.

Also 1011111101100010_2 is converted as follows:

1 3 7 5 4 2 (octal)

1 0 1 1 1 1 1 1 0 1 1 0 0 0 1 0 (binary)
B F 6 2 (hexadecimal)

Therefore, $1011111101100010_2 = 137542_8$ Or $BF62_{16}$

In order to convert a number in octal or hexadecimal system to its equivalent in binary, every digit in the octal/hexadecimal number is separately converted into binary first, expressing writing each converted number as a three/four-digit number, possibly filling any empty space with a zero. Then the three/four-digit numbers in binary can be combined to give the desired equivalent number in binary. For example, to convert 573_8 to its binary equivalent, we proceed as follows:

$5_8 = 101_2$; $7_8 = 111_2$ and $3_8 = 011_2$

Therefore, $573_8 = 101111011_2$.

Also to convert $1AF_{16}$ to its binary equivalent we have:

$1_{16} = 0001_2$;$A_{16} = 10_{10} = 1010_2$ and $F_{16} = 15_{10} =1111_2$

So, $1AF_{16} = 110101111_2$ (or 000110101111)

Note that the leading zeros (in 0001_2) before a non-zero digit are ignored in the final answer. So, the equivalent of $1AF_{16}$ in binary, correctly written, is 11010101111_2.

2.4 ARITHMETIC OPERATIONS ON NUMBER BASES

Irrespective of any number base being used at any given time, the four basic arithmetic operations of addition, subtraction, multiplication and division are possible. The set of symbols (or digits) used in each case depends on the particular base of the number system. If the base of the system is, say, b then there are b different digits including the digit '0' and all other ones that are less than or equal to b-1.

For instance, if the base is two, the set of digits is {0,1} and if the base is eight, the set of digits used is {0,1,2,...7}. To avoid confusion, if any base greater than ten is used, every 2-digit-symbol under that particular number system from the symbol `10' are represented by the letters of the alphabets A, B, C, etc. For example under the hexadecimal system, the set of digits is {0,1,2,...,9,A,B,C,D,E,F}; where A = 10, B = 11, C = 12, ...; and F = 15.

In every number system, the set of the digits is said to be closed under the four rules of arithmetic operations; that is, if any of the four operations is used on any two or more of the digits in any given base, the result also is a number that contains digits found under the number system.

2.4.1 **Addition and Subtraction Operations**

The operations of addition and subtraction are carried out in every number system in a similar manner as the operations are carried out in decimal number system. The same idea of `carrying' to the left in the case of addition and `borrowing' from the left in the case of subtraction are used. One only needs to bear in mind the very base of the system being used just like when the decimal system is used. For instance, just like:

$7_{10} + 5_{10} = 12_{10}$ (i.e. write down the remainder 2 and `carry' the quotient 1 to the left), so also we have 5 + 6 = 14 in base seven (i.e. 4 is the remainder when $11_{10}$ is divided by 7 and the quotient 1 is `carried' to the left). Similarly, C + 9 = 15 in base sixteen, i.e C + 9 gives 21_{10} and the remainder when 21 is divided by 16 is 5 with a quotient of 1 which is `carried' to the left to give the sum of C and 9 to be 15_{16}.

In the binary number system, addition yields the sums of only four combinations of 0's and 1's which are:

0 + 0 = 0; 0 + 1 = 1; 1 + 0 = 1; and 1 + 1 = 0 with a carry of 1, giving 10. Additions

in binary or any other number system can be performed column by column as in the decimal system, with carries being handled in the same manner. The only difference is the base and the digits or symbols involved. In the case of binary number system, the only digits involved are 0 and 1 and there are four entries for subtraction also like in the case of addition. The possible four differences that can be obtained are:

0 - 0 = 0; 1 - 0 = 1; 1 - 1 = 0; and 0 - 1 = 1, with a borrow of 1 from the next more significant digit on the left in the minuend. The `borrow' is handled as with the decimal numbers. For instance, the difference between 25_8 and 6_8 can be found as follows: since 5 is less than 6, 5-6 is not possible so 1 is borrowed from 2 on the left side and getting to the next position on the right the 1 takes the value of 1 x 8. Now, adding 8 and 5 we get 13_{10} and 13 - 6 gives 7. Bringing the remaining 1 on the left down we get 17. Thus $25_8 - 6_8 = 17_8$.

Subtraction in the other number bases are based on the same principles as explained above in the case of decimal and binary systems. The operation of subtraction is carried out column-wise and with the idea of 'borrowing' handled in the same manner as in the denary system. The following are some examples on how subtractions are done in different number systems.

(1)
$$\begin{array}{r} 1\,0\,1\,1\,1\,0\,1_2 \\ -\,1\,0\,0\,1\,1\,1_2 \\ \hline 1\,1\,0\,1\,1\,0_2 \end{array}$$

(2)
$$\begin{array}{r} 6\,3\,5\,2_7 \\ -\;1\,4\,2\,5_7 \\ \hline 4\,6\,2\,4_7 \end{array}$$

$$\begin{array}{lr} (3) & 5\,A\,8\,D\,6_{16} \\ & -\quad E\,7\,B\,9_{16} \\ \hline & 4\,C\,1\,1\,D_{16} \end{array} \qquad\qquad \begin{array}{lr} (4) & 3\,2\,1\,0_{4} \\ & -\,1\,3\,2\,1_{4} \\ \hline & 1\,2\,2\,3_{4} \end{array}$$

It should be noted that addition or subtraction between any two numbers in different bases is not possible unless the two numbers are converted into the same base, either into one of the two given bases or into base 10, and when the operation is carried out, the result can be converted into the desired base. For instance, the sum of 3124_5 and 6517_8 can be found in any of the following ways:

Either $3124_5 + 6517_8 = 3124_5 + 102112_5 = 110241_5$

or $3124_5 + 6517_8 = 636_8 + 6517_8 = 7355_8$

Or $3124_5 + 6517_8 = 414_{10} + 3407_{10} = 3821_{10}$, which can now be converted to any other desired base, say $3821_{10} = 2265_{12}$.

This same idea is applied when the difference between any two numbers that are not in the same base is to be found.

2.4.2 Multiplication and Division Operations

The knowledge of how to add and subtract numbers using any given base is a pre-requisite for the understanding of multiplication and division, respectively, in any number system. This is because multiplication and division are, respectively, just like repeated addition and subtraction. The process of writing down the remainder and `carrying' the quotient to the left (when the sum of two numbers is greater than the base) used in the case of addition, is normally employed when multiplying. Equally used in the case of division is the idea of `borrowing' from the left to the right when a larger number is to be deducted from a smaller number.

Just like the operations of addition and subtraction, multiplication and division in any other base are carried out in a similar way as in the denary number system. The only thing to bear in mind at each stage is the base in which the operations are carried out and the different digits involved in that system. It should also be noted that multiplication, like addition in binary operation, is both commutative and associative, that is, the order in which any given two numbers are multiplied do not matter.

Similar to what is obtained when denary numbers are multiplied, if the product of any two digits (in any given base) is greater than the base, the product is divided by the base, the remainder is written down and the quotient is `carried' to the next more significant position on the left. For instance, the product of 2 and 4 in base 5 is obtained as follows: since the product of 2 and 4 is 8 (in base ten which is greater than five, the base in use), and the figure 8 can not be used in base 5 system, 8 is divided by 5 giving a quotient of 1 and a remainder of 3. Writing down the remainder and `carrying' the quotient to the left, we obtain 13, i.e. 2x4 = 13 (in base 5).

Products of other numbers in any other base are obtained in the same manner.

In binary number system, multiplication is simple if one understands binary addition very well. This is because at any given time, the product of any two digits is either 0 or 1. It also follows that understanding and multiplication of numbers in other number systems depends greatly on the knowledge of addition of numbers in such systems. It should, however, be noted that two numbers can only be multiplied if they are in the same base. Below are some examples on how to multiply numbers in different number bases.

Examples:

(1)
$$\begin{array}{r}
11011_2 \\
\text{x}\ \ 111_2 \\ \hline
11011 \\
11011 \\
+\ \ 11011 \\ \hline
1011110 1_2
\end{array}$$

(2)
$$\begin{array}{r}
4132_5 \\
\text{x}\ \ 43_5 \\ \hline
+\ \ 12301 \\
32133 \\ \hline
334131_5
\end{array}$$

(3)
$$\begin{array}{r}
3675_8 \\
\text{x}\ \ 213_8 \\ \hline
13467 \\
3675 \\
+\ \ 7572 \\ \hline
1031637_8
\end{array}$$

(4)
$$\begin{array}{r}
B398_{16} \\
\text{x}\ \ 2D_{16} \\ \hline
91ED2 \\
+\ 16734 \\ \hline
1F9212_{16}
\end{array}$$

Division in other number systems is done very much the same way as it is done in the denary system with the only major difference being the base used which is not ten. The familiar trial-and-error technique is used when division is performed. In long divisions, the knowledge of multiplication and subtraction is very important.

In the case of binary number system, each quotient digit can only be a 0 or 1 which greatly simplifies the process. Some examples are given below on how to perform division in some number systems.

Examples

(1) $1110011_2 \div 101_2$

$$
\begin{array}{r|l|l}
101 & 1110011_2 & 10111_2 \\
\hline
 & \underline{101} & \\
 & \quad 1000 & \\
 & \quad \underline{101} & \\
 & \qquad 111 & \\
 & \qquad \underline{101} & \\
 & \qquad\quad 101 & \\
 & \qquad\quad \underline{101} & \\
 & \qquad\qquad - &
\end{array}
$$

Therefore $1110011 \div 101_2 = 10111_2$

(2) $15013_6 \div 23_6$

$$
\begin{array}{r|l|l}
23_6 & 15013_6 & 423_6 \\
\hline
 & \underline{140} & \\
 & \quad 101 & \\
 & \quad \underline{\;\;50} & \\
 & \qquad 113 & \\
 & \qquad \underline{113} & \\
 & \qquad\quad - &
\end{array}
$$

$15013_6 \div 23_6 = 423_6$

(3) $8D530_{16} \div 34_{16}$

$$
\begin{array}{r|l|l}
34_{16} & 8D530_{16} & 2B7C_{16} \\
\hline
 & \underline{68} & \\
 & 255 & \\
 & \underline{23C} & \\
 & \quad 193 & \\
 & \quad \underline{16C} & \\
 & \qquad 270 & \\
 & \qquad \underline{270} &
\end{array}
$$

Therefore $8D530_{16} \div 34_{16}$ = $\mathbf{2B7C_{16}}$

(4) $16633_8 \div 65_8$

$$
\begin{array}{r|l|l}
65_8 & 16633_8 & 217_{16} \\
\hline
 & \underline{152} & \\
 & \quad 143 & \\
 & \quad \underline{\;\;65} & \\
 & \quad 563 & \\
 & \quad \underline{563} & \\
 & \qquad - &
\end{array}
$$

$16633_8 \div 65_8$ = $\mathbf{217_8}$

2.5 USES OF NUMBER BASES

The conventional decimal (or the base ten) number system is commonly used virtually in every sphere of life for counting and otherwise. Apart from this, computing machines and other modern electronic systems can be built on the basis of any number system. Particularly useful is the binary (or base two) number system. This is because all modern digital computers are based on the binary system since it is easier to distinguish between two entities than when other bases are used.

Moreover, most physical quantities all have only two states: a switch is on or off; a light bulb is on or off; a current is positive or negative; a material is magnetised or demagnetised; etc. Most computers perform their operations in binary system. This means that all data are represented by series of only two symbols: 1 and 0. The Central Processing Unit (CPU) of a computer uses a binary system in storing and handling data and so its job is simplified as it has to recognise only 1's and 0's. If it had to deal with our conventional symbols (i.e 26 alphabetical characters, 10 different digits and many special symbols), the CPU would need much more complex and costly circuits. Computers use several variations of binary system; some can switch from one binary system to another, under program control. Examples of some variations include BCD-Binary Coded Decimal; EBCDIC - Extended Binary Coded Decimal Interchange Codes; and ASCII - American Standard Code for Information Interchange.

Other important number systems are the Octal and Hexadecimal systems. Octal and Hexadecimal numbers are of particular importance in computers since they can be used as convenient alternatives for binary numbers. (See conversion from octal/hexadecimal to binary). An important use of octal system is in listings of programs and for memory 'dumps' for binary machines, thus making the printouts more compact. Also in keeping track of the contents of different registers and in orally conveying the contents of a binary register to someone, it is more convenient to use octal characters rather than binary. For instance to send the binary 11101111 over the telephone, it is easier and less conducive to error to say "three - five - seven in octal" than to say "one - one - one - zero - one - one - one - one in binary". The hexadecimal number system is useful for similar reasons. In many computers, especially the minicomputers, the memories are organised into sets of "bytes" consisting of 8 binary digits.

CHAPTER THREE

3.0 C.A.I SYSTEM DESIGN

3.1 INTRODUCTION

A system can be regarded as a set of interacting elements responding to input in order to generate output. In other words, a system can be said to be a collection of components, either physical or non-physical in nature that are interconnected, with each one carrying out a specific function and which interact with one another towards achieving a common objective.

A computer is an electronic device that is capable of accepting data (input), processing the data and producing information (output), fast, accurately and more efficiently than human efforts. A computer system, which has a goal of solving problems for the user, consists of three major components, namely:

(i) the hardware

(ii) the human-ware (life ware) and

(iii) the software.

For a computer to exist, all the above three components must be present. The hardware consists of the physical components of a computer which is made up of the mechanical, magnetic, electrical and electronic devices. The input, processing, storage and controlling devices constitute the hardware.

The human-ware is made up of people engaged in the use of computers for one purpose or the other. Software is the general term used to denote all forms of programs that control the activities of a computer and which enables us to exploit the capabilities of a computer.

A computer program is a set or a sequence of instructions, which informs a computer of the steps required for achieving a defined task.

A computer programming language is simply a mode of communicating to the computer. There are three main classes of computer languages – machine language, assembly language, and high level language.

High level language includes BASIC, FORTRAN, COBOL, Pascal and DBASE. The main advantage of the high level language is that they are machine independent and they are written in English-like manner which can be easily understood. However, they require a translator for a computer to execute programs written in them.

3.2 CHOICE OF PROGRAMMING LANGUAGE

BASIC, an acronym for "Beginners All Purpose Symbolic Instruction Code", is the programming language used in this project work. BASIC language has many versions but the difference between the versions is minimal. Every higher version is an improvement on the lower one and therefore can accept instructions in the lower version in most cases but the reverse does not hold. The version of the BASIC language used in this project is the QBASIC (Quick BASIC).

The major reason for using BASIC language in this program is because BASIC is found in almost every available personal computer (PC), it is easy to understand and programs written in BASIC language are easy to debug. Other advantages of using the BASIC language include the following:

1. BASIC language is people oriented. It is easy to learn, fun to use and does not necessarily need extensive background knowledge in mathematics.

2. The language is machine-independent, therefore a BASIC program can be run on many different computers.

3. BASIC language is very flexible. It therefore allows the programmer to alter an existing program or instruction with very little efforts.

4. QBASIC supports instant syntax checking as instructions are entered and it gives instant help on errors.

5. The language is suitable for use on time-sharing Systems and this offers the programmer the opportunity to use large computers at a minimum expense.

The BASIC language is commonly found in almost every computer system so this program can easily be run by anybody who can afford a personal computer and who is interested in using the program.

3.3 STRUCTURED OR MODULAR PROGRAMMING

Structured (or modular) programming is a new method of programming in which programs have easy-to-follow logic and are divided into subprograms or modules, each designed to perform a specific task. In BASIC language, subprograms or modules are referred to as SUBROUTINES.

Structured programming avoids large and complex programs in favour of more manageable subprograms, each designed to perform a specific task. Because the logic of structured programs are easier to follow than that of the unstructured programs, they are more likely to be free of errors and easier to modify at a later period.

The following advantages are derived from using modular or structured approach:

1. Since modules are small, they are easier to create and maintain.
2. Problem areas are simple to pinpoint. If every module performs a specific task, the programmer can more easily keep track of the elements of his program. Programming errors can easily be found and corrected quickly with this system.
3. When each module is tested and found working then they can all be linked together to form a complete system.

3.4 PROGRAM DESIGN

The BASIC program in this project work has been designed to be as user-friendly as possible. The program has been divided into modules or sections, which can be assessed by choosing from a list of options in a menu presented to the user.

After being loaded, this program can be run on any computer that is installed with the QBASIC version of BASIC language. The menu in this program, from where choices are made from letters A to I, comprises of the following:

A Introduction to Number Bases.

B Conversion of Numbers from One Base to Another

C Addition in Various Number Bases

D Subtraction in Various Number Bases

E Multiplication in Various Number Bases

F Division in Various Number Bases

G Conversion Program from base ten

H Conversion Program to base ten

I Quit

The choice of any of the letters A through F takes the user to the indicated module or section with the corresponding topic. Options G and H are programs that can convert numbers from one base to another as indicated while option I allows the user to quit the program if and when desired.

The modules are designed in such a way that at every screen-full (page) the user may choose to move to the next page (or next screen) or to the previous page of the text, or he may choose to go back to the menu at any point in time within the same module.

Each module which treats a particular mathematical operation on number bases contains some texts, which are followed by some examples, and it finally provides some exercises which the user is expected to do. The user is expected to work out the answer to each of the exercises and feed the answer to the computer.

There is always an immediate feedback to each of the exercises. If the answer to any of the given exercises is wrong at the first attempt, the user is advised by the computer to try again. If the answer is still wrong after the second trial the user would then be supplied with the correct answer and asked to go over that particular module again before the exercises can be re-tried. But if the answer to the exercise is right in the first or second trial, then the response: "That is correct; Very Good" or a similar remark is shown on the screen.

At the end of each module, the user is taken back to the Menu where he/she may choose another module or quit the program. Each of the options A through F in the Menu treats a particular concept on number bases as indicated below.

OPTION A: INTRODUCTION TO NUMBER BASES.

This option contains the text that introduces the user to the idea or concept of number system generally.

OPTION B: CONVERSION OF NUMBER FROM ONE BASE TO ANOTHER.

Conversion between any two number systems is treated in this module. After the user must have gone through all the examples and exercises in this section, options G and H can be chosen if he/she wishes to convert more numbers from one base to another. This enables the user to practice more on his/her own and checks the correctness or otherwise of any conversion done by the user.

OPTIONS C - F: ADDITION, SUBTRACTION, MULTIPLICATION AND DIVISION OF NUMBER BASES.

Each of these options treats the indicated mathematical operation on Number Bases with examples and at the end gives exercises for the user to practice.

OPTION G: CONVERSION FROM BASE TEN TO OTHER BASES.

This option takes the user to the program that can convert any positive decimal integer not greater than 2^{16} = 65536 to any desired base not greater than 16.

OPTION H: CONVERSION FROM OTHER BASES TO BASE TEN.

The program in this section accepts any positive integer in any base (not greater than 16) and converts it to its decimal equivalent. The converted equivalent in base ten should not, however, be greater than 65,536. (ie 2^{16}).

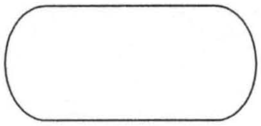
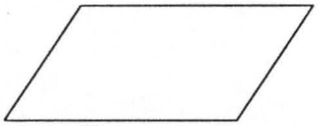
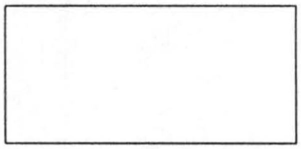
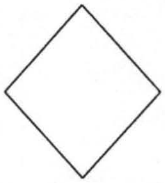
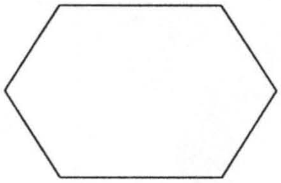
OPTION I: QUIT

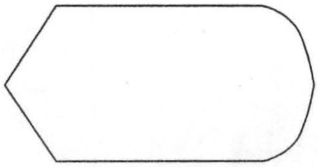
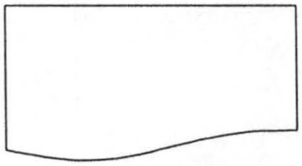
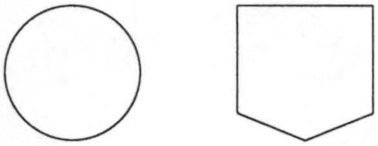
The choice of this option enables the user terminate the tutorial and, if the user so wishes, quits the QBASIC environment.

3.5 FLOWCHART

A flowchart is a graphic representation of the basic steps to solve a well-defined problem. It helps to indicate the flow of events and their relationship. A flowchart can equally be defined as a diagrammatical representation of algorithms.

In general, some symbols frequently used in flowcharts' and their meanings or uses are as follows:

	SYMBOL	USES
1.	 Terminal (or start/stop) symbol	Used to indicate the start or end of a program.
2.	 Input/output symbol	Represents either inputting data or outputting results.
3.	 Process Symbol	Used to represent calculations or other processing operations.
4.	 Decision Symbol	Represents a comparison. The action taken next depends on the result of the comparison.
5.	 Preparation Symbol	Indicates the dimensions of arrays or represents initialization procedures

6.	 Display Symbol	Used to display information
7.	 Document Symbol	This indicates a document or a text
8.	 (i) (ii) Connection Symbols	(i) Indicates exit from or entry to another part of the flowchart on the same page. (ii) Indicates exit from or entry to another flowchart on another page.

TYPES OF FLOWCHARTS

There are two major types of standard flowcharts. They are: system and program flowcharts. The program flowchart is always more detailed than the systems flowchart. Also, the systems flowchart is the systems analysts' tool used for systems designs, while the program flowchart is the programmers' tool used for program development. These are the only differences between the two types.

SYSTEMS FLOWCHART:

The systems flowchart is sometimes called the Operational Systems Chart. It shows the actual flow of processing from input document to output, and clearly portrays the usage of files and the relationship between the programs in the system.

PROGRAM FLOWCHART

This is used as a documentation of the algorithm design. The graphic display permits an easier and more effective grasp of the algorithms flow and, more importantly the location of errors in logic. The program flowcharts used in this project are as shown in Appendix A.

CHAPTER FOUR

4.0 CAI SYSTEM EXPERIMENTATION

4.1 INTRODUCTION

The immediate aim of this work is to provide a suitable means of instruction to any body who wishes to learn about Number Bases through the use of a computer without much human assistance.

In the production of an output from any system, some of the following points are considered:

1. How often are the outputs needed?
2. Who requires the output and in what form is it needed?
3. Are the copies required in multiple form for internal or external circulations?
4. Are pre-printed forms needed?

Generally, the major users of CAI outputs are the learners of a particular topic or subject covered by the CAI. The users are expected to benefit personally and directly from the materials provided by the computer as they interact with the computer.

4.2 OUTPUT OF CAI

Apart from the texts in form of instructions and notes in the tutorial, the main outputs in this system are obtained in the sections that contain the exercises. The user may be required to provide an answer to a given exercise, after which the computer may respond by giving remarks based on the answer given.

4.2.1 Output on Conversion

This package enables the user to convert different numbers from one base to another and gives the correct results of such conversions. Appendix – B – are examples of program outputs on number conversion.

4.2.2 OUTPUT ON MATHEMATICAL OPERATIONS

This tutorial contains exercises on each of the mathematical operations – addition, subtraction, multiplication and division – on number bases. As the user follows the instructions in the tutorial and supplies the required information, the outputs are viewed on the screen.

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATIONS

5.1 SYSTEMS REQUIREMENTS

The computer system configuration that can effectively execute the application software developed in this project work include the following:

(i) A QBASIC Interpreter

(ii) A visual display unit (V.D.U), preferably a VGA 14" coloured/monochrome monitor for good outputs of the screen designs.

(iii) At least a 101 enhanced keyboard which is compatible with most software.

(iv) Hard disk drive with at least 40 megabytes memory.

(v) Any DOS based (at least 5.0) operating system.

(vi) At least a 3.5" floppy disk drive.

(vii) A system unit with at least four megabytes of RAM to run the QBASIC package effectively.

5.2 USER'S MANUAL INSTRUCTIONS

This tutorial (CAI) on number bases consists of a menu with nine different items (A – I) which can be selected, one at a time, by entering through the keyboard any of the indicated upper case letters (A – I) on the menu to signify the desired section of the tutorial the user is interested in. The menu consists of the following options:

A. Introduction.
B. Conversion of Number Bases.
C. Addition.
D. Subtraction.
E. Multiplication.
F. Division.
G. Conversion Program from Base ten.
H. Conversion Program to Base ten.
I. Quit

This menu is activated by typing 'MAINBASE' from the QBASIC directory or environment and pressing the enter key. Since this package is a tutorial, no password is required to gain access to use it. The menu will be displayed on the screen immediately the word 'MAINBASE' is typed and entered. From here the user can now select any of the options, depending on what he/she intends to do, by entering any of the upper case letters (A to I) as earlier mentioned. In the process, the user is instructed on what steps to take at every stage.

The choice of option I (Quit) in the menu terminates the tutorial and enables the user to quit the QBASIC environment if so desired.

5.3 COST AND BENEFIT ANALYSIS

The growth in the use of microcomputers or personal computers (PCs) in schools, homes and offices today has made a lot of things easier and simpler to achieve which would have hitherto been difficult or almost impossible. Utilising the seemingly unlimited data processing powers of the electronic computers enables the computer-aided instruction to overcome some of the deficiencies of instruction associated with the conventional classroom teaching. Without any assistance from the teacher and without much extra costs, if any at all, a student can use a CAI like this to help himself/herself overcome some of the difficulties encountered on a given topic or subject at his/her own time and pace.

It does not require much to install and use this CAI software in a place where there is a computer system already. Thanks to God, the number of individuals or institutions acquiring computer systems even in a developing country like ours today is growing so rapidly that by God's grace the microcomputer will soon be a common tool in most schools, offices and homes in the near future.

Another benefit is that the more the students use this software to improve in their knowledge of the particular topic covered, the more they get familiar with the use of computer itself.

With all these benefits, however, the use of other writing materials such as paper and pen and some assistance from the teacher should always be encouraged. The help from the computer can hardly totally eliminate the human assistance provided by the teacher to the learner. The user is also advised to get a good textbook handy for further and more exercises since the tutorial cannot possibly contain everything.

5.4 LIMITATIONS OF THE PROJECT

This project work is not intended to be used by any students as a substitute for their normal class work on number base systems. Moreover, the tutorial does not cover every aspect of number bases. For instance, conversion of fractions or arithmetic operations on fractions are not considered in this tutorial.

Although this tutorial can be used by any interested persons, it is only possible for those who have access to computers and who can, at least, basically or minimally operate a microcomputer.

No number bases less than two (2) or greater than sixteen (16) and no number whose decimal (converted) equivalent is greater than 2^{16} (= 65,526) can be accepted for conversion in the example and exercises contained in this tutorial.

The bases of numbers in the tutorial are written in words as it is difficult to indicate the bases as subscripts in the program environment.

5.5 RECOMMENDATIONS

This project work is a simple tutorial package on CAI that can be useful to students, ranging from those in their senior classes (SS1 and above) in post primary institutions to those in their preliminary or first year in the tertiary institutions, who are interested in or are expected to have some knowledge of number bases.

This package is, therefore, recommended for schools and even homes that can afford personal computers as the students/users can freely use the tutorial even during their leisure periods in the school or at home.

It should be noted that the materials in this tutorial can only be used as supplements to students' work in the class but does not completely replace the teacher's class teachings or recommended textbooks on the topic covered. The teacher can equally recommend this tutorial to the students to be used, where possible, during their own free times.

5.5.1 HARDWARE ACQUISITION

Any institution interested in installing this software for their students should purchase, if there is none already, a computer system as specified in the systems requirements (see 5.1).

5.5.2 **INSTALLATION**:-

For this package to run in a computer system, **QBASIC** has to be resident in that system. Then the package can be installed on the hard disk from the source diskette (enclosed) as follows:

From the C:\> (prompt) Type ***CD QBASIC***

C:\QBASIC> Type ***MD MAINBASE***

C:\QBASIC> Type **CD MAINBASE**

Now insert Diskette in **Drive A:**

C:\MAINBASE> Type **Copy A:*.***

C:\MAINBASE> Type **MAINBASE**

From here, the user follows the instructions on the screen and selects the required menu.

REFERENCES

1. ADEWOYIN, J. E. (1991): Introduction to Educational Technology; John - LAD Publishers LTD, Lagos.

2. AYENI, J. O A. (1992): Fundamentals of Computing; University of Lagos Press, Lagos.

3. AYO, C. K. (1994): Computer Literacy (Operations & Appreciation); Alanukitan Commercial Press (NIG.) LTD, Egbe, Kogi State.

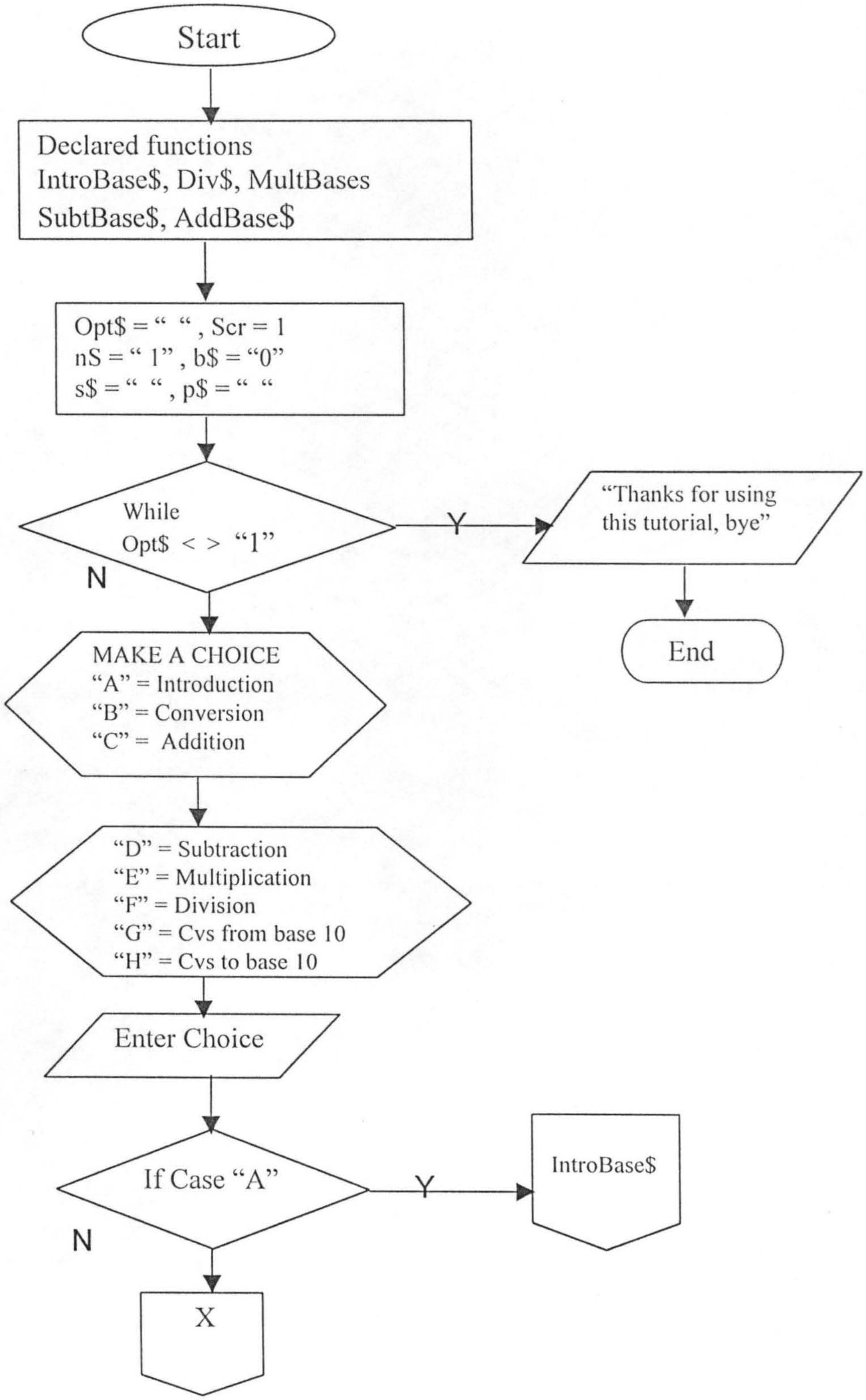
4. CURZON, L. B. (1985: Teaching In Further Education: An Outline of Principles and Practice (3rd edition), Cassell Educational LTD, London.

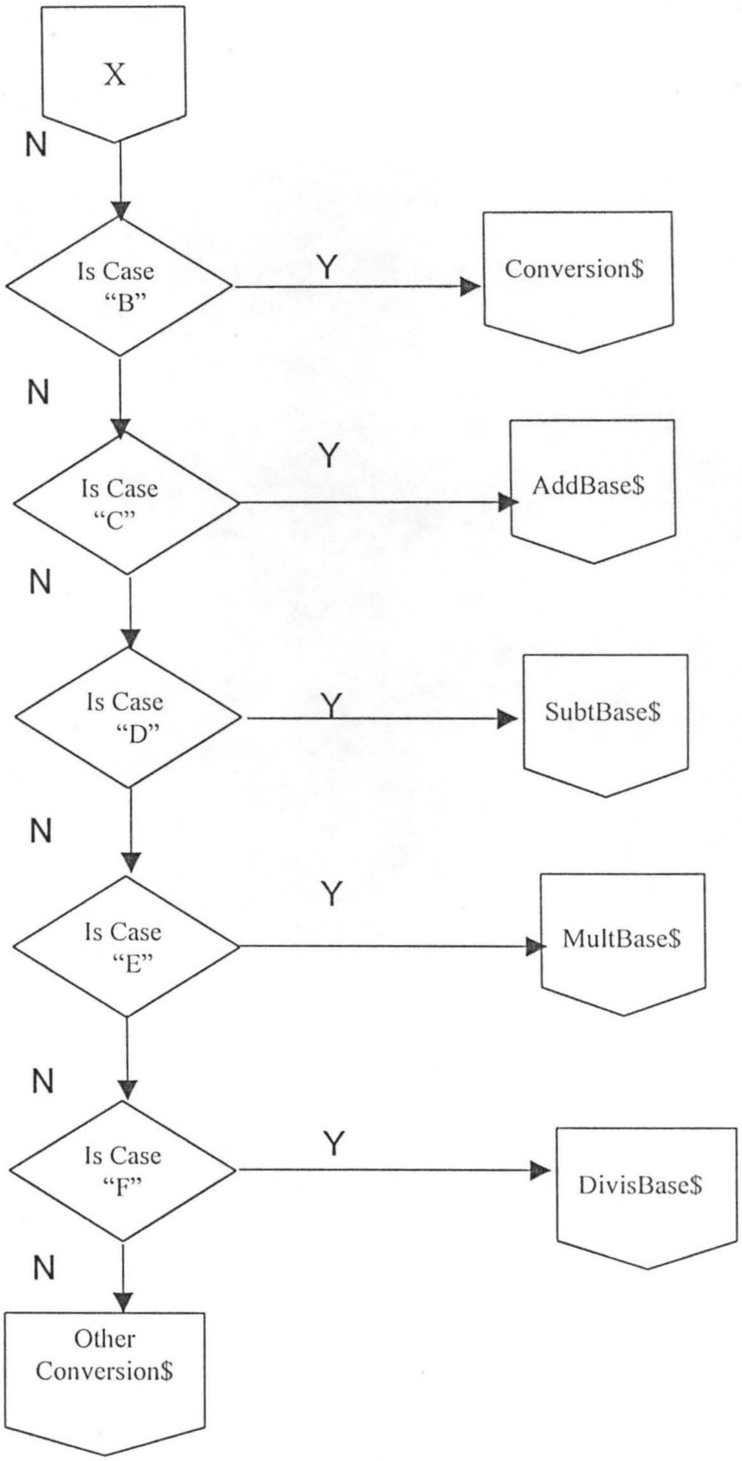
5. ICSU/COSTED (1987): Computer - Aided Instruction in Science & Technology (Proceedings of a Sub-Regional Workshop held in Lagos, Nigeria, 4th - 7th May, 1987, edited by Awele Maduemezia); ICSU/COSTED, Ibadan.

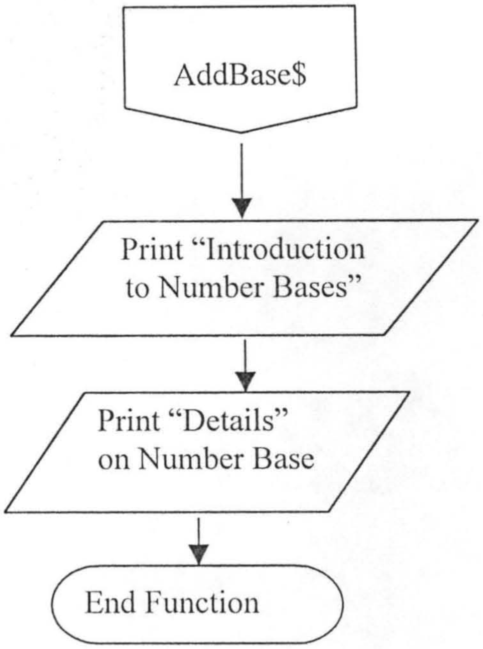
6. KRAMER, A. D, (1986): Mathematics for Computers; McGraw-Hill Inc.; New York.

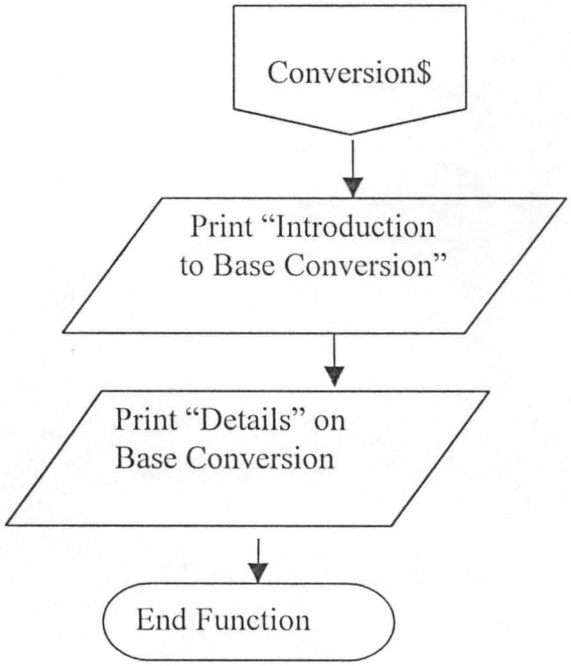
7. MANDEL, S. L. (1986): Basic Programming Today - A Structured Approach; West Publishing Company, Minnesota.

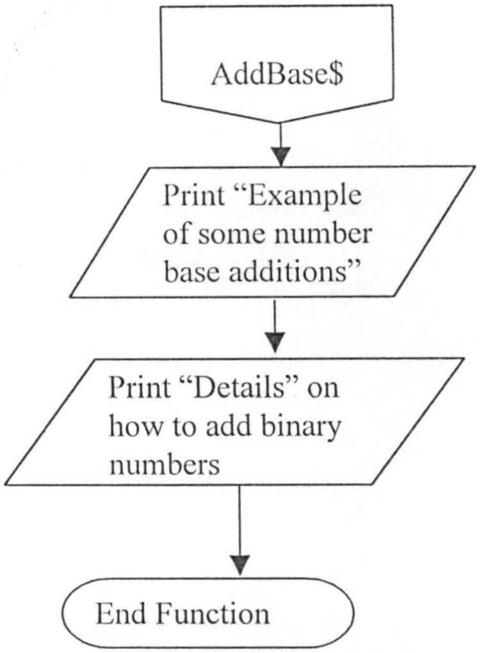
APPENDIX A: - FLOW CHARTS

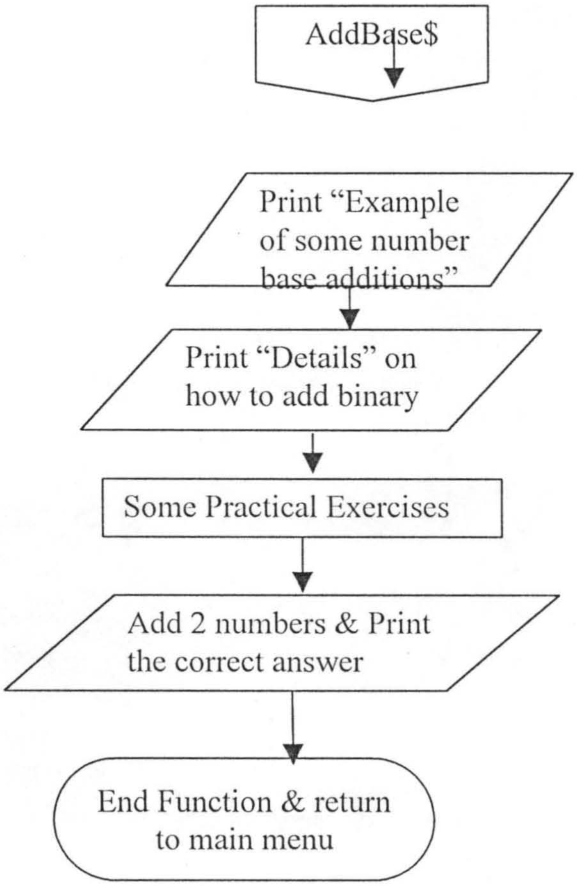


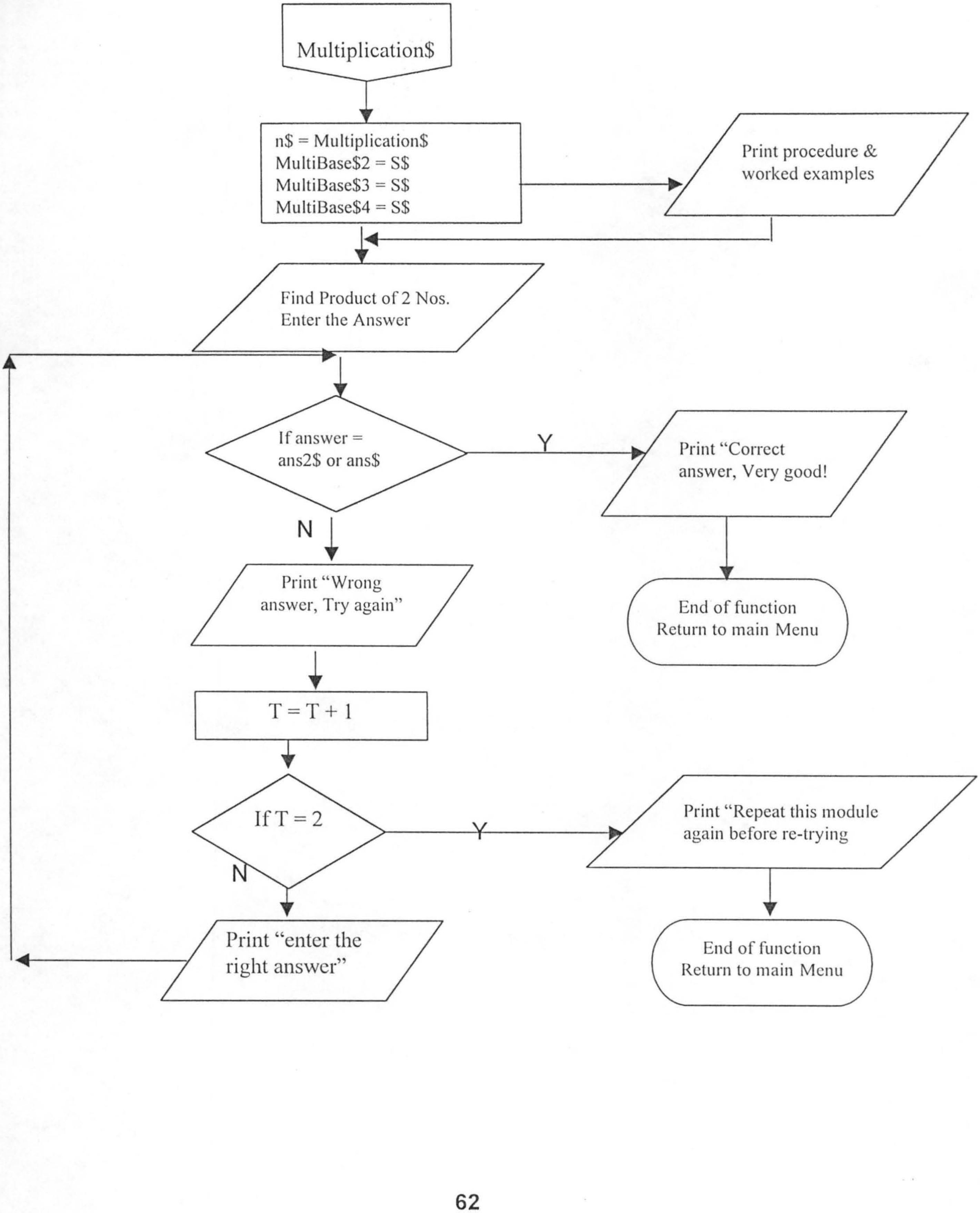

X
N
Is Case "B"
Y
Conversion$
N
Is Case "C"
Y
AddBase$
N
Is Case "D"
Y
SubtBase$
N
Is Case "E"
Y
MultBase$
N
Is Case "F"
Y
DivisBase$
N
Other Conversion$

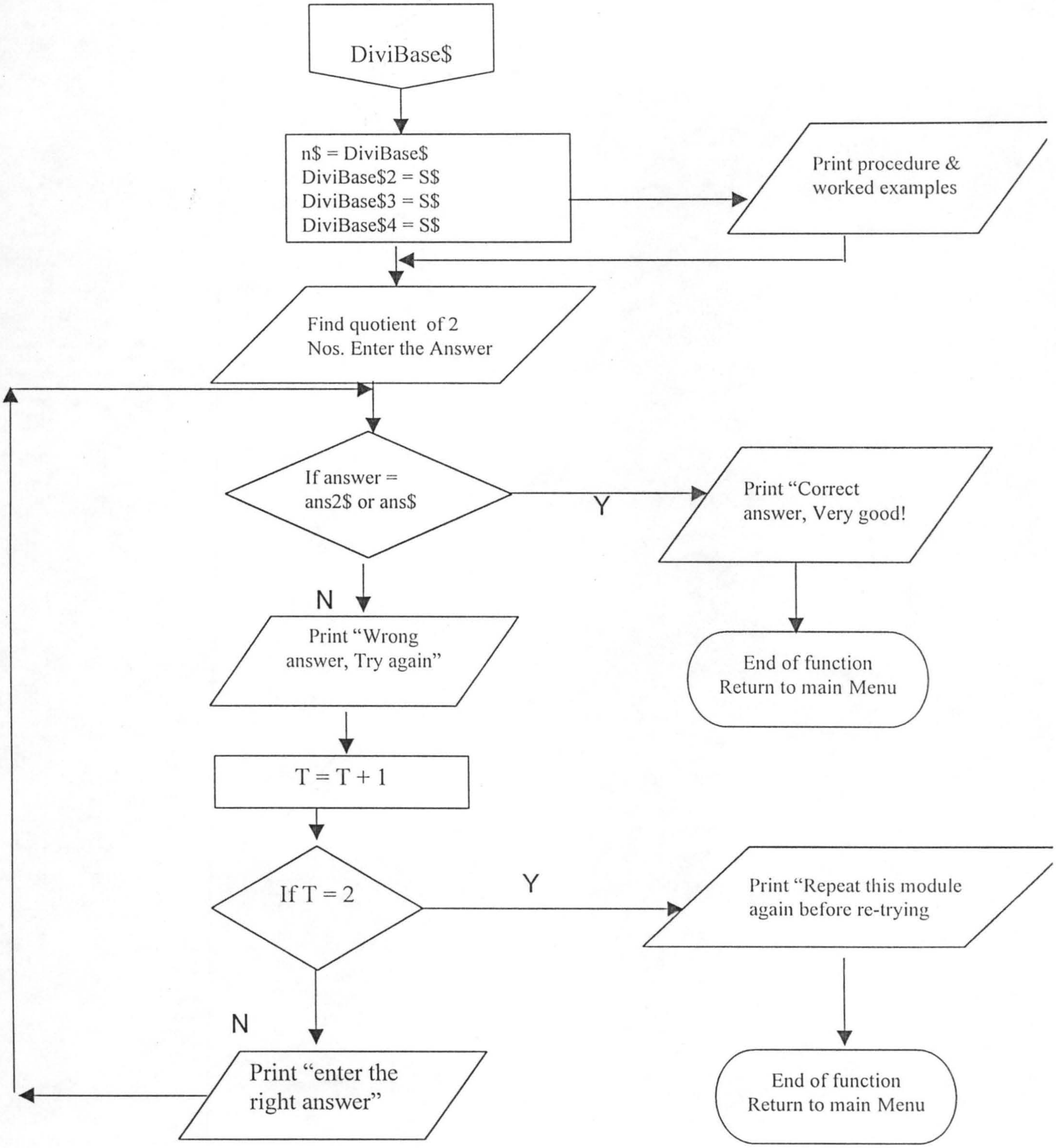

AddBase$
Print "Introduction to Number Bases"
Print "Details" on Number Base
End Function

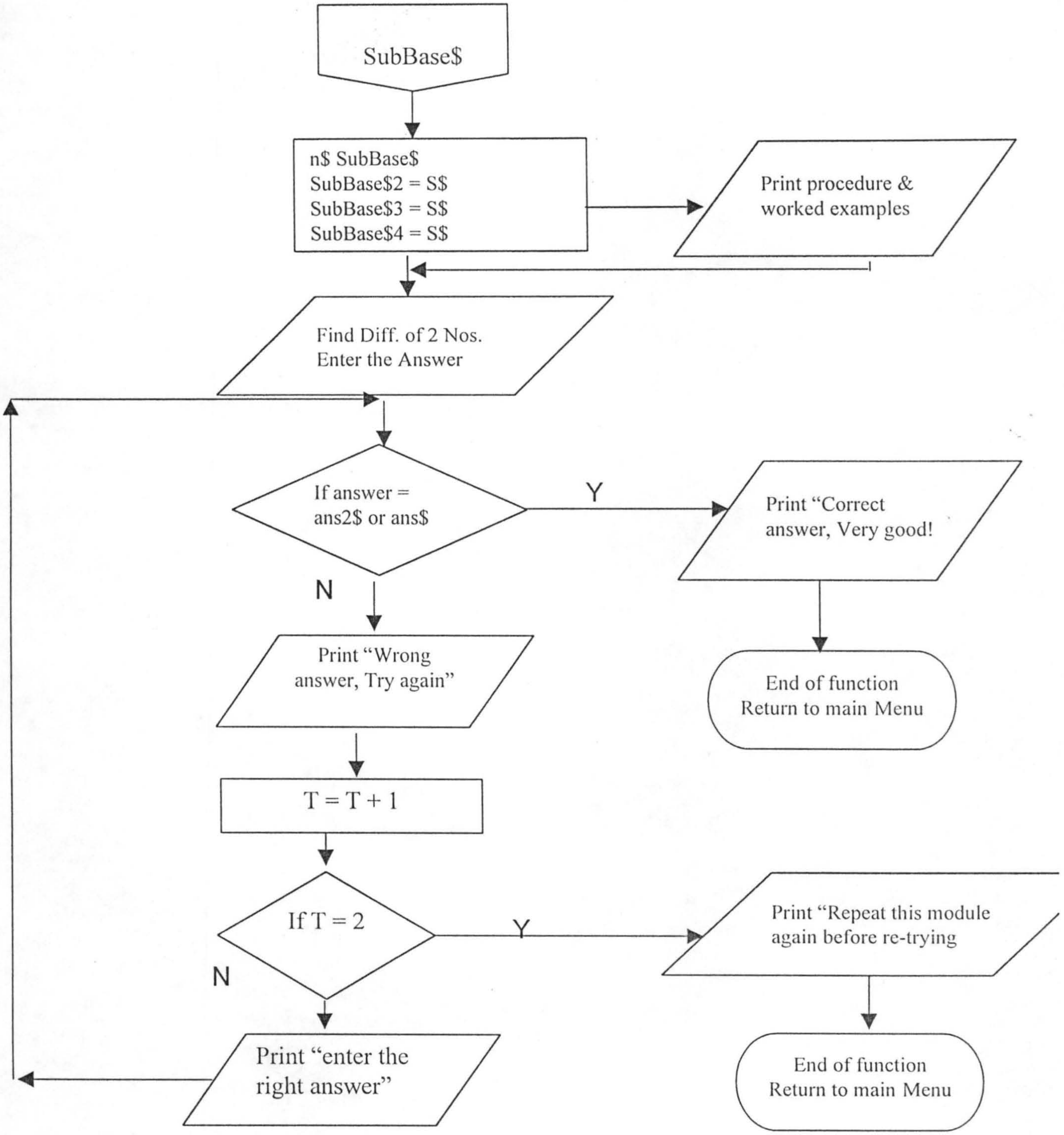

Conversion$
Print "Introduction to Base Conversion"
Print "Details" on Base Conversion
End Function

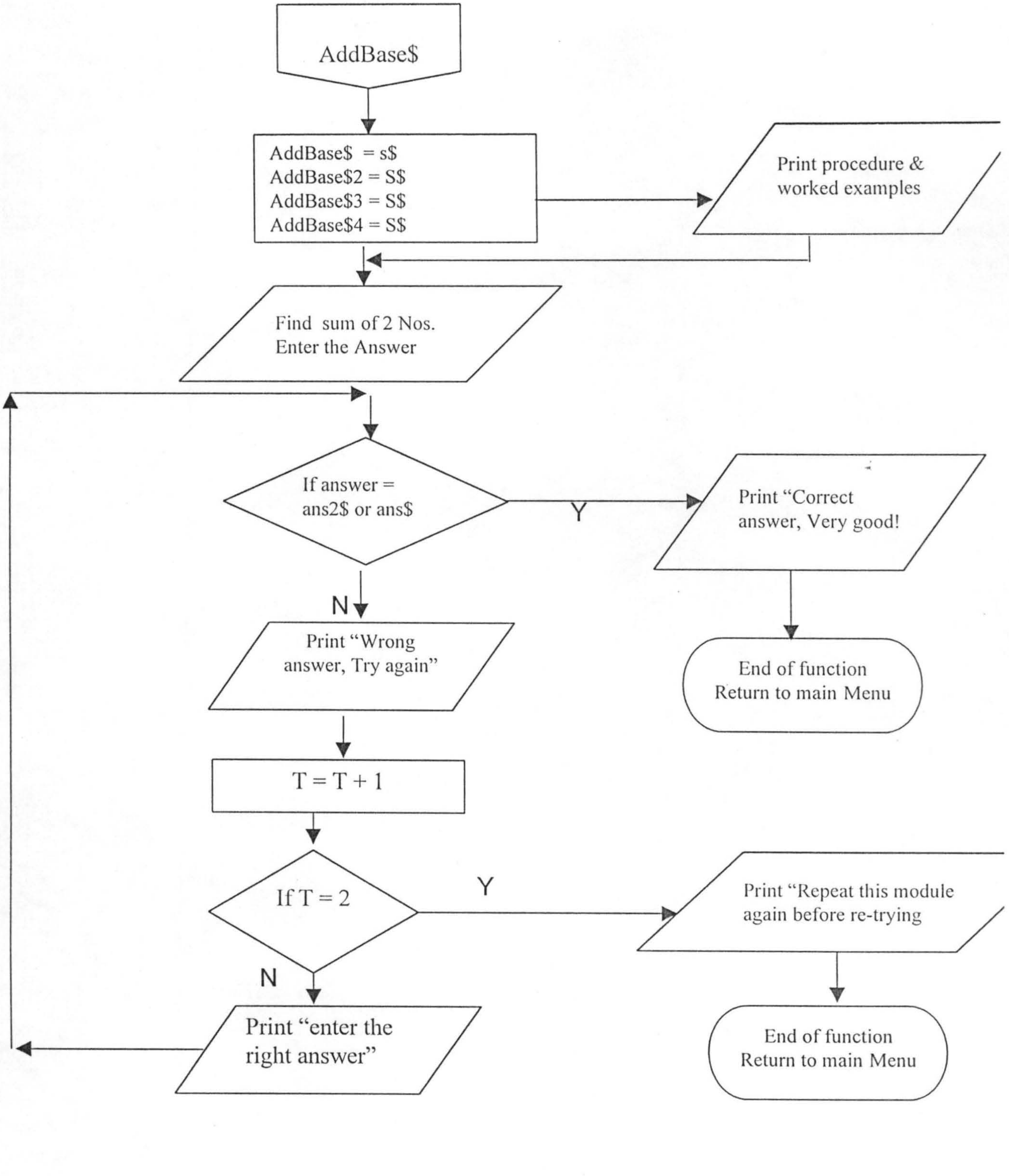

AddBase$
Print "Example of some number base additions"
Print "Details" on how to add binary numbers
End Function


AddBase$
Print "Example of some number base additions"
Print "Details" on how to add binary
Some Practical Exercises
Add 2 numbers & Print the correct answer
End Function & return to main menu


Multiplication$
n$ = Multiplication$
MultiBase$2 = S$
MultiBase$3 = S$
MultiBase$4 = S$
Print procedure & worked examples
Find Product of 2 Nos. Enter the Answer
If answer = ans2$ or ans$
Y
Print "Correct answer, Very good!
N
Print "Wrong answer, Try again"
End of function Return to main Menu
T = T + 1
If T = 2
Y
N
Print "Repeat this module again before re-trying
Print "enter the right answer"
End of function Return to main Menu


DiviBase$
n$ = DiviBase$
DiviBase$2 = S$
DiviBase$3 = S$
DiviBase$4 = S$
Print procedure &
worked examples
Find quotient of 2
Nos. Enter the Answer
If answer =
ans2$ or ans$
Y
Print "Correct
answer, Very good!
End of function
Return to main Menu
N
Print "Wrong
answer, Try again"
T = T + 1
If T = 2
Y
Print "Repeat this module
again before re-trying
N
Print "enter the
right answer"
End of function
Return to main Menu


SubBase$
n$ SubBase$
SubBase$2 = S$
SubBase$3 = S$
SubBase$4 = S$
Print procedure & worked examples
Find Diff. of 2 Nos. Enter the Answer
If answer = ans2$ or ans$
Y
Print "Correct answer, Very good!
End of function Return to main Menu
N
Print "Wrong answer, Try again"
T = T + 1
If T = 2
Y
Print "Repeat this module again before re-trying
End of function Return to main Menu
N
Print "enter the right answer"


AddBase$
AddBase$ = s$
AddBase$2 = S$
AddBase$3 = S$
AddBase$4 = S$
Print procedure & worked examples
Find sum of 2 Nos. Enter the Answer
If answer = ans2$ or ans$
Y
Print "Correct answer, Very good!
End of function Return to main Menu
N
Print "Wrong answer, Try again"
T = T + 1
If T = 2
Y
Print "Repeat this module again before re-trying
N
Print "enter the right answer"
End of function Return to main Menu

APPENDIX B

Program Listing

```
DECLARE FUNCTION CONPROG1$ ()
DECLARE FUNCTION Conversion$ ()
DECLARE FUNCTION Convprog10$ ()
DECLARE FUNCTION conprog10$ ()
DECLARE FUNCTION MultBase3$ ()
DECLARE FUNCTION AddBase6$ ()
DECLARE FUNCTION AddBase5$ ()
DECLARE FUNCTION AddBase4$ ()
DECLARE FUNCTION AddBase3$ ()
DECLARE FUNCTION AddBase2$ ()
DECLARE FUNCTION AddBase1$ ()
DECLARE FUNCTION SubtBase3$ ()
DECLARE FUNCTION SubtBase2$ ()
DECLARE FUNCTION SubtBase1$ ()
DECLARE FUNCTION SubtBase4$ ()
DECLARE FUNCTION subtBase5$ ()
DECLARE FUNCTION SubtBase6$ ()
DECLARE FUNCTION MultBase1$ ()
DECLARE FUNCTION MultBase2$ ()
DECLARE FUNCTION MultBase3$ ()
DECLARE FUNCTION MultBase4$ ()
DECLARE FUNCTION DiviBase1$ ()
DECLARE FUNCTION DiviBase2$ ()
DECLARE FUNCTION DiviBase3$ ()
DECLARE FUNCTION DiviBase4$ ()
DECLARE FUNCTION Addition$ ()
DECLARE FUNCTION Subtraction$ ()
DECLARE FUNCTION Multiplication$ ()
DECLARE FUNCTION Divi$ ()
DECLARE FUNCTION IntroBase1$ ()
DECLARE FUNCTION Intro$ ()
DECLARE FUNCTION IntroBase2$ ()
DECLARE FUNCTION IntroBase3$ ()

opt$ = " "
scr = 1
n$ = "1"
b$ = "0"
s$ = " "
p$ = " "

WHILE UCASE$(opt$) <> "I"

Mainmenu:
CLS
LOCATE 2, 35: PRINT "M A I N  M E N U"
LOCATE 3, 35: PRINT STRING$(9, CHR$(196))
LOCATE 4, 30: PRINT "NUMBER BASE TUTORIAL"
LOCATE 5, 30: PRINT STRING$(20, CHR$(196))
LOCATE 6, 31: PRINT "A. Introduction to Number Bases"
LOCATE 7, 31: PRINT "B. Conversion of Numbers "
LOCATE 8, 31: PRINT "C. Addition in Bases"
LOCATE 9, 31: PRINT "D. Subtraction in Bases"
LOCATE 10, 31: PRINT "E. Multiplication of Numbers"
LOCATE 11, 31: PRINT "F. Division of Numbers"
LOCATE 12, 31: PRINT "G. Conversion from Base Ten"
LOCATE 13, 31: PRINT "H. Conversion to Base Ten"
LOCATE 14, 31: PRINT "I. Quit"

LOCATE 17, 27: PRINT "Enter Choice [A - I]:"
LOCATE 17, 54: INPUT opt$

'WEND
opt$ = UCASE$(opt$)

SELECT CASE opt$

        CASE "A"
                p$ = Intro$
        CASE "B"
              p$ = Conversion
        CASE "C"
                p$ = Addition
        CASE "D"
                p$ = Subtraction
        CASE "E"
                p$ = Multiplication
        CASE "F"
                p$ = Divi
        CASE "G"
                p$ = conprog10
        CASE "H"
                p$ = CONPROG1

END SELECT

 WEND
 PRINT "Thank you for using this TUTORIAL.  Bye!"
 END
 ' SHELL

FUNCTION AddBase1$
CLS
STATIC s$
s$ = ""
LOCATE 1, 19: PRINT "ADDITION IN VARIOUS NUMBER BASES         SCREEN 1/6"
PRINT : PRINT "Addition of numbers in any base b is done in similar manner as
in"
PRINT "base ten.  The only difference is that is the base is less than ten,
the the "
PRINT "only digits that can be involved are 0, 1, ..., b - 1, where b is the
base."
PRINT "But when the base is greater than ten, then the digits involved are 0
and 9,"
PRINT "and A for 10, B for 11, etc, until the last alphabet representing the"
PRINT "number b - 1.  If base b is used, there is a total of b different
symbols"
PRINT "(or figures) involved.  For example, in base 8, there is a total of 8
symbols"
PRINT "involved:  0, 1, ..., 7 are the figures.  The idea of 'carrying' used"
PRINT "in base 10 is also applied if the sum of any two digits is equal to or
greater"
PRINT "than the given base.  For instance, just like:"
PRINT "      7 + 5 = 12 in base 10   so also is"
PRINT "      4 + 6 = 13 in base 7    or"
PRINT "      C + 9 = 15 in base 16."
PRINT : PRINT "   With the above idea in mind, any given set of numbers can be
added"
PRINT "in a given base.  Since addition is a binary operation, that is, at
any"
PRINT "point in time only two figures are added and if there are more than two
figures"
PRINT "to be added, their addition is done in pairs before the subsequent
results are "
PRINT "similarly added."

LOCATE 23, 5: PRINT "N for next screen; P for previous screen; Q to Quit to
main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
AddBase1 = s$

END FUNCTION

'
FUNCTION AddBase2$
CLS
STATIC s$
PRINT "                                              screen 2/6"
PRINT "For example, in any base the sum a + b + c = (a + b) + c or"
PRINT "a + (b + c), since addition is associative. NOTE that, addition
between"
PRINT "any two numbers is possible only if the two numbers are in the same
base. "
PRINT "So, to add any which are not in the same base, first convert them to
the same base."
PRINT : PRINT "Example 1:  Addition in base 2:": PRINT
PRINT "         1 0 0 1 0 1 1           Note: 0 + 0 = 0; 0 + 1 = 1"
PRINT "           + 1 1 0 1 1           1 + 0 = 1 and 1 + 1 = 0 with a carry
of 1"
PRINT "         -------------           to the left giving 1 0."
PRINT "         1 1 0 0 1 1 0           "
PRINT "         =============                   "
PRINT : PRINT "       1 1 1 <--- 'carried digits'"

LOCATE 23, 5: PRINT "N for next screen; P for previous screen;Q to Quit to
main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
AddBase2$ = s$

END FUNCTION

FUNCTION AddBase3$
CLS
STATIC s$
s$ = ""
PRINT "Example 2:  Addition in base 8                        screen 3/6"
PRINT
PRINT "                 7 5 3           "
PRINT "            +  6 4 3 5                   "
PRINT "            ----------           "
PRINT "               7 4 1 0                   "
PRINT "            ==========                 "
PRINT "                1 1 1  <---- 'carried digits"
PRINT
PRINT " Note:  3+ 5 gives 0 with a carry of 1 to the left, 5 + 3 + 1 (carried)"
PRINT "gives 1 with a carry of 1 to the left and 7 + 4 + 1 (carried) gives 4 with"
PRINT "a carry of 1 added to 6 to give 7."
PRINT : PRINT "Example 3:  Addition in base 6."
PRINT
PRINT "              4 B 9 2 C"
PRINT "            + 2 5 4 D 7    "
PRINT "             ----------    "
PRINT "              7 0 E 0 3       "
PRINT "             ==========              "
PRINT "                1 1 1 <--- 'carried digits'       "

LOCATE 23, 5: PRINT "N for next screen; P for previous screen;Q to Quit to main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
AddBase3$ = s$


END FUNCTION

FUNCTION AddBase4$
CLS
PRINT "                                                       Screen 4/6"
PRINT "  To add 3124 in base 5 abd 6517 in base 8"
PRINT "Step 1:  If the answer is needed in base 5, then convert 6517 to base 5"
PRINT "(but remember to first of all convert 6517 to base 10 before the decimal"
PRINT "equivalent is converted to base 5) or if the answer is desired in base 8"
PRINT "then convert 3124 to base 8 as stated conversion to base 5 above."
PRINT "Step 2:  Add the two numbers to get the final answer in the desired base."
PRINT "But if the answer is desired in a base different from the two bases give, then"
PRINT "a)  Convert each of the numbers into its equivalent in base 10"
PRINT "b)  Add the two numbers in base 10; and"
PRINT "  Convert the sum in base 10 to the desired base to get the final answer"
PRINT "  So, 3124 (in base 5) + 6517 (in base 8) equals 110241 in base 5"
PRINT "3124 (in base 5) + 6517 (in base 8) equals 7355 in base 8 and 2265 in base 12"

LOCATE 23, 5: PRINT "N for next screen; P for previous screen;Q to Quit to main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
AddtBase4$ = s$


END FUNCTION
```

```
FUNCTION AddBase5$
CLS
PRINT "                                        Screen 5/6"
PRINT "Note that:"
PRINT "3124 (base 5) = 141 (base 10) or 636 (base 8) and 6517 (base 8) = 3407"
PRINT "(base 10) or 102112 (base 5).  So, 3124 (base 5) + 6517 (base 8) =
110241"
PRINT "(base 5) or 7355 (base 8) 2265 (base 12)."
PRINT : PRINT "Example 5:  Study the following examples that are given without
further"
PRINT "explanations.  Add the following in the given base:"
PRINT
PRINT "a)  Base 3                b)  Base 5               c)  Base 15 "
PRINT "    2 1 1 2                   3 4 1 2 0                B 4 5 C "
PRINT "  + 1 2 0 2                +  2 0 4 3 4              + 8 D 3 4        "
PRINT "  ----------               -------------             ----------   "
PRINT "  1 1 0 2 1                  1 1 0 1 0 4             1 5 2 9 1"
PRINT "  =========                =============             ==========        "

LOCATE 23, 5: PRINT "N for next screen; P for previous screen;Q to Quit to
main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
AddBase5$ = s$

END FUNCTION

FUNCTION AddBase6$
CLS : T = 0
ans2$ = ""
PRINT "                                                       screen 6/6"
PRINT "Exercises:"
PRINT : PRINT "Exercise 1:  Add in base two:  1 0 1 1 0 + 1 0 1 1 1 "
10      INPUT "Result is ..."; ans
IF ans = 100001 THEN
        PRINT "That is correct!"
ELSE
T = T + 1
IF T = 2 THEN
        PRINT "Repeat this module again before re-trying the exercises"
        PRINT "The correct answer is 100001."
ELSE
PRINT "Please, try again"
GOTO 10
END IF
END IF
PRINT

PRINT "Exercise 2.  Add in base 16:  3F94 + DE2D"
T = 0

20 INPUT "Result is ..."; ans2$
IF ans2$ = "FDC1" THEN
        PRINT "That is the correct answer!"
ELSE
T = T + 1
IF T = 2 THEN
        PRINT "Repeat this module again before re-trying the exercise"
        PRINT " The correct answer is FDC1."
ELSE
        PRINT "Please, try again."
        GOTO 20
        END IF
        END IF
PRINT : PRINT
'EXIT FUNCTION
'END SELECT


LOCATE 23, 5: PRINT "P for previous screen;Q to Quit to main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
AddBase6 = s$



END FUNCTION

FUNCTION Addition$
prevA1:
n$ = AddBase1$
SELECT CASE n$
        CASE "M"        'Main Menu
        PRINT "You selected main menu ..."
        CASE "Q"        'Q
        Addition = "Q"
        EXIT FUNCTION
END SELECT

prevA2:
n$ = AddBase2$
SELECT CASE n$
        CASE "M"        'Main Menu
        PRINT "You selected main menu ..."
        CASE "P"
        GOTO prevA1
        CASE "Q"        'Q
        Addition = "Q"
        EXIT FUNCTION
END SELECT

prevA3:
n$ = AddBase3$
SELECT CASE n$
        CASE "M"        'Main Menu
        PRINT "You selected main menu ..."
        CASE "P"
        GOTO prevA2
        CASE "Q"        'Q
        Addition = "Q"
        EXIT FUNCTION
END SELECT

prevA4:
n$ = AddBase4$
```

```
SELECT CASE n$
        CASE "M"        'Main Menu
        PRINT "You selected main menu ..."
        CASE "P"
        GOTO prevA3
        CASE "Q"        'Q
        Addition = "Q"
        EXIT FUNCTION
END SELECT

prevA5:
n$ = AddBase5$
SELECT CASE n$
        CASE "M"        'Main Menu
        PRINT "You selected main menu ..."
        CASE "P"
        GOTO prevA4
        CASE "Q"        'Q
        Addition = "Q"
        EXIT FUNCTION
END SELECT

n$ = AddBase6$

IF n$ = "P" THEN
        GOTO prevA5
ELSE
        Addition = "Q"
END IF
'END SELECT
END FUNCTION

FUNCTION CONPROG1$
19

CLS
' Bas16t10.bas
LOCATE 5, 5: PRINT " *** CONVERSION FROM ANY BASE NOT GREATER THAN 16 TO BASE
10 ***"
PRINT : PRINT : PRINT
'LOCATE 5, 15: PRINT "***  PROGRAM OUTPUT ***   "
'
'
cval$ = " "
DIM x(50)
btensum = 0
lenval = 0
bnum = 0
 PRINT : PRINT

INPUT "Enter Number to be converted to base ten :"; cval$
DO
        INPUT "Enter Base of the Number to be converted: "; bnum
        IF bnum <= 1 OR bnum > 16 THEN
PRINT : PRINT "Base must be greater than ONE but less or equal to 16"
        END IF

LOOP UNTIL bnum > 1 AND bnum <= 16

cval$ = LTRIM$(RTRIM$(cval$))
```

```
lenval = LEN(cval$)

FOR i = 1 TO lenval
        SELECT CASE ASC(UCASE$(MID$(cval$, i, 1)))
                CASE 65
                        x(i) = 10
                CASE 66
                        x(i) = 11
                CASE 67
                        x(i) = 12
                CASE 68
                        x(i) = 13
                CASE 69
                        x(i) = 14
                CASE 70
                        x(i) = 15
                CASE ELSE
                        x(i) = VAL(MID$(cval$, i, 1))
        END SELECT
NEXT i

j = 1
FOR i = lenval - 1 TO 0 STEP -1
    btensum = btensum + (x(j) * (bnum ^ i))
    j = j + 1
NEXT i

PRINT :

LOCATE 16, 2: PRINT "The number "; ; cval$; ; "in base"; ; bnum; ; "converted
to base ten is "; btensum


LOCATE 22, 10: INPUT "Any more number to be converted"; s$
IF UCASE$(s$) = "Y" THEN 19



END FUNCTION

FUNCTION conprog10$
15

'cdecbase.bas
CLS

PRINT "**** THIS MODULE OF PROGRAM CONVERTS NUMBERS FROM BASE TEN TO OTHER
BASES ****": PRINT
DIM x(25) AS STRING * 2
n% = 0
b% = 0
i% = 1
j% = 1
DO
        INPUT "ENTER NUMBER TO BE CONVERTED"; n%
        IF (n% <= 0) THEN
                PRINT "Number must be greater than ZERO "
        END IF

LOOP UNTIL n% > 0
```

```
c% = n%

DO
        INPUT "Enter the Base to be converted to : "; b%
        IF (b% <= 1) OR (b% = 10) OR (b% > 16) THEN

PRINT "The Base should be greater than ONE, should not be TEN and must be less
or equal to 16"

        END IF

LOOP UNTIL (b% > 1) AND (b% <> 10) AND (b% <= 16)

DO
        x(i%) = STR$(c% MOD b%)
        SELECT CASE c% MOD b%
                CASE 10
                        x(i%) = "A"
                CASE 11
                        x(i%) = "B"
                CASE 12
                        x(i%) = "C"
                CASE 13
                        x(i%) = "D"
                CASE 14
                        x(i%) = "E"
                CASE 15
                        x(i%) = "F"
        END SELECT
        IF b% > c% THEN
                i% = i% + 1
                x(i%) = "0"
                EXIT DO
        ELSE
                c% = c% \ b%
        END IF

        i% = i% + 1
LOOP UNTIL c% <= 0
i% = i% - 1

PRINT USING "Number in Base 10 is #####: "; n%: PRINT
PRINT USING "Converted value in base ## is "; b%: PRINT
FOR j% = i% TO 1 STEP -1
        PRINT x(j%);
NEXT j%


LOCATE 22, 10: INPUT "Any more number to be converted"; s$
IF UCASE$(s$) = "Y" THEN 15


END FUNCTION

FUNCTION Conversion$

CLS
PRINT " ****** CONVERSION OF NUMBERS FROM ONE BASE TO ANOTHER ******"
PRINT : PRINT "Any number given in any base can be expressed by an equivalent
value"
```

```
PRINT "in another base.  Any number in decimal system can be converted to any
othetr"
PRINT "base and vice - versa.  However, conversion between any other two bases
may not"
PRINT "be easy without first converting to base ten.  Binary numbers can be
numbers"
PRINT "becasue the bases or radices in the octal and hexadecimal numbers are
integer "
PRINT "powers of 2."
PRINT
PRINT "Conversion can be:": PRINT
PRINT "1.  From base ten to other bases.": PRINT "2.  From other bases to base
ten."
PRINT "3.  Betwen Binary and Octal/Hexadecimal number system."
SLEEP (1): CLS

GOSUB PRESS
MENU: CLS
     COLOR 28: LOCATE 11, 21: PRINT "1": LOCATE 13, 21: PRINT "2": LOCATE 15,
21: PRINT "3"
     LOCATE 17, 21: PRINT "4": LOCATE 18, 21: COLOR 7, 0
     M$ = " XXXXX  BASE TEN TO OTHER BASES  XXXXX   ": UR$ = " XXXXX  OTHER
BASE TO TEN  XXXXXX "
     T$ = " XXXXX  BINARY & OCTAL/HEXADECIMAL  XXXXX": 'P$ = " XXXXX RECORD
DELETING  XXXX "
     E$ = " XXXX EXIT TO MAIN MENU  XXXX    ": FOR PP = 1 TO LEN(M$)
     LOCATE 11, 25: PRINT LEFT$(M$, DD): LOCATE 13, 25: PRINT RIGHT$(UR$, DD):
     LOCATE 15, 25: PRINT LEFT$(T$, DD): DD = DD + 1
     LOCATE 17, 25: PRINT LEFT$(E$, DD): DD = DD + 1
     FOR FF = 1 TO 300: NEXT: NEXT
    a$ = " CONTROL MENU   ": FOR a = 1 TO 35: LOCATE 7, a: PRINT a$
    FOR TT = 1 TO 200: NEXT: NEXT
    COLOR 16, 7: LOCATE 7, 35: PRINT a$: COLOR 7, 0
    FOR Q = 6 TO 22: LOCATE Q, 20: PRINT CHR$(186): LOCATE Q, 22: PRINT
CHR$(179)
    LOCATE Q, 70: PRINT CHR$(179): LOCATE Q, 72: PRINT CHR$(186): NEXT
    LOCATE 6, 20: PRINT CHR$(201) + STRING$(51, 205) + CHR$(187): LOCATE 22,
20: PRINT CHR$(200) + STRING$(51, 205) + CHR$(188)
    LOCATE 8, 22: PRINT CHR$(218) + STRING$(47, 196) + CHR$(191): LOCATE 20,
22: PRINT CHR$(192) + STRING$(47, 196) + CHR$(217)
    LOCATE 21, 35: INPUT " YOUR OPTION PLS  "; opt
             IF opt = 1 THEN GOSUB TENBASE
             IF opt = 2 THEN GOSUB OTHERBASE
             IF opt = 3 THEN GOSUB BOHBASES
             IF opt = 4 THEN CLS : SYSTEM
GOTO MENU

TENBASE:

CLS
PRINT " ****** CONVERSION OF NUMBERS FROM BASE TEN TO OTHERS ******": PRINT
PRINT "This conversion is done by repeatedly dividing the given number in
base"
PRINT "ten by the base of the required new system, writing down the remainder"
PRINT "after each division.  This is continued until the quotient is equal to"
PRINT "zero."
PRINT TAB(4); "The equivalent number in the new base is the member formed by
all the"
PRINT "remainders, starting from the last remainder to the first.  The last
remainder"
```

```
PRINT "forms the most significant digit (MSD) while the first reminder is the
least "
PRINT "significant digit (LSD) of the number in the new base."
SLEEP (2): COLOR 20: PRINT : PRINT "Please wait for Examples": COLOR 7: SLEEP
(3)
PRINT
PRINT "Example 1:  23 (base 10) is converted to binary as follows:"

GOSUB PRESS
4 GOSUB cont:
GOTO MENU

SLEEP (2)
LOCATE 23, 15: PRINT "Press any key to continue"
IF INPUT$(1) = "" THEN
END IF

RETURN

cont:
CLS
PRINT "No."; TAB(8); "New base Division"; TAB(34); "Quotient"; TAB(50);
"Reminder"
PRINT : PRINT "23  /"; TAB(8); "2"; TAB(34); "11"; TAB(50); "1          [LSD]"
PRINT "11  /"; TAB(8); "2"; TAB(34); "5"; TAB(50); "1"
PRINT "5   /"; TAB(8); "2"; TAB(34); "2"; TAB(50); "1"; TAB(62); CHR$(24)
PRINT "2   / "; TAB(8); "2"; TAB(34); "1"; TAB(50); "0"
PRINT "1   / "; TAB(8); "2"; TAB(34); "0"; TAB(50); "1        [MSD]"
PRINT : PRINT "The result written in the direction of the arrow gives 10111"
PRINT ":- 23 (base 10) = 10111 (base 2)."

PRINT : PRINT "This working can be set out in a neater way as:"
PRINT
PRINT "                2 ³ 23  ³  R                              "
PRINT "               ÄÄÄ³ÄÄÄÄÄ³ÄÄÄÄÄ                        "
PRINT "                2 ³ 11  ³  R 1          [LSD]             "
PRINT "                2 ³ 5   ³  R 1                            "
PRINT "                2 ³ 2   ³  R 1            "; CHR$(24)
PRINT "                2 ³ 1   ³  R 0                            "
PRINT "                2 ³ 0   ³  R 1          [MSD]             "

PRINT : PRINT "This yield the same result as 23 (base 10) = 10111 (base 2)"

GOSUB PRESS

CLS
PRINT "Example 2.  3562 (base 10) converted to base 7."
PRINT
PRINT "                2 ³3562 ³  R                              "
PRINT "               ÄÄÄ³ÄÄÄÄÄ³ÄÄÄÄÄ                        "
PRINT "                2 ³ 508 ³  R 6           [LSD]            "
PRINT "                2 ³ 72  ³  R 4                            "
PRINT "                2 ³ 10  ³  R 2             "; CHR$(24)
PRINT "                2 ³ 1   ³  R 3                            "
PRINT "                2 ³ 0   ³  R 1           [MSD]            "
PRINT
PRINT ": -  3562 (base 10) = 13246 (base 7)"

GOSUB PRESS
```

```
CLS
PRINT "Example 3.  44699 (base 10) converted to base 16."
PRINT
PRINT "                16 ³44699³  R                                  "
PRINT "                ÄÄÄ³ÄÄÄÄÄ³ÄÄÄÄÄ                                "
PRINT "                16 ³ 2793³  R 11  i.e B         [LSD]          "
PRINT "                16 ³ 174 ³  R 9                                 "
PRINT "                16 ³ 10  ³  R 14  i.e E          "; CHR$(24)
PRINT "                   ³ 0   ³  R 10  i.e A                         "
PRINT "                   ³ 0   ³                        [MSD]           "
PRINT
PRINT ": -  44699 (base 10) = AE9B  (base 16)."
PRINT
PRINT "EXERCISES:-  Convert the following decimal numbers to the required
base:"
PRINT : PRINT "1.  42 to binary number": PRINT "     ANSWER - 101010 base 2."
PRINT : PRINT "2.  5678 to Octal number": PRINT "     ANSWER - 13056 base 8."
PRINT : PRINT "Now if you want to convert any integer number in base ten not
greater than"
PRINT "65,500 to its equivalent value in any other base then ..... ?":

GOSUB PRESS
RETURN

OTHERBASE:
CLS
PRINT TAB(10); " *** CONVERSION FROM OTHER BASES TO BASE TEN ****"
PRINT : PRINT "This Conversion can be done using any of the following two
methods:"
COLOR 22: PRINT "METHOD 1:": PRINT : COLOR 7
PRINT "If the given number to be converted to base ten is in base `b' and has"
PRINT "`n' digits, the following steps are taken:": PRINT
PRINT "1st digit [msd]  * b ^ (n-1)"
PRINT "2nd digit        * b ^ (n-2)"
PRINT " :                   :     "
PRINT " :                   :     "
PRINT "last digit [lsd] * b ^ (n-n)": PRINT

PRINT "Then sum up all the results got in each step above to get the required"
PRINT "equivalent value in base ten."
SLEEP (2): COLOR 20: PRINT : PRINT "Please wait for Examples:": COLOR 7:

CLS
PRINT "Example 1:  110101 (base 2) is to be converted to base ten."
PRINT "                  Where n = 6 number of digits and b = 2 given base."
PRINT "            :- the equivalent number in base ten is": PRINT
PRINT "1*2**(6-4) + 1*2**(6-2) + 0*2**(6-3) + 1*2**(6-4) + 0*2**(6-5) +
1*2**(6-6)"
PRINT : PRINT "i.e    2**(5) + 2**(4)+ 0 + 2**(2)+ 0 + 2**(0)": PRINT
PRINT "       = 32 + 16 + 0 + 4 + 0 + 1    =   53 (base 10).": PRINT
PRINT "           : - 110101 (base 2)  =  53 (base 10)": PRINT : PRINT

PRINT "Example 2:  To converted 6B2E (base 16) to Decimal."
PRINT "where n = 4 number of digits, b  = 16 given base, so the equivalent "
PRINT "number is given by:": PRINT
PRINT "6*16**(4 - 1)+ B*16**(4 - 2) + 2*16**(4 - 3)+E*16**(4 - 4)"
PRINT : PRINT "Notice that, B & E (base 16) are 11 & 10 (base 10)
respectively."

GOSUB PRESS
```

```
CLS : PRINT "The equivalent number is:-": PRINT
PRINT "6*16**(3) + 11*16**(2) + 2*16**(1) + 14"
PRINT "     = 6*4096 + 11*256 + 2*16 + 1"
PRINT "       = 24576 + 2816 + 32 + 14        =   27438              "
PRINT : PRINT ":. 6B2E (base 16)  =  27425 (base 10)"

GOSUB PRESS

CLS
COLOR 22: PRINT "METHOD 2:": COLOR 7:
PRINT "This method is applied as follows:":
PRINT "   The first digit is [msd] applied in the unmber to be converted to
base ten"
PRINT "is multiplied by the base, then the next digit is added to the product.
 The"
PRINT "sum is agian multiplied by the base and the next digit is added again.
 This"
PRINT "process continues until the last digit [lsd] in the number finnaly
added"
PRINT "which will give the required equivalent number in base ten.": PRINT

PRINT "Example 1:  To convert 10111 (base 2) to base ten."
PRINT "1st digit [msd] * 2 i.e 1 * 2 = 2     "
PRINT "add next digit                  0      "
PRINT "                              ÄÄÄ      "
PRINT "sum multiply by 2              2 * 2 = 4"
PRINT "add next digit                          1       "
PRINT "                                      ÄÄÄ     "
PRINT "sum multiply by 2                      5 * 2 = 10    "
PRINT "add next digit                                  1   "
PRINT "                                               ÄÄÄ    "
PRINT " sum multiply by 2                             11 * 2 = 22"
PRINT "add last digit [lsd]                                    1"
PRINT "                                                        ÄÄÄ "
PRINT "Equivalent value (base 10)               =               23 "

GOSUB PRESS

CLS
   PRINT "Example 2:  To convert A6E3 (base 16) to denary equivalent.":
   PRINT "                10 * 16 = 160           "
   PRINT "add next digit              6                       "
   PRINT "                         ÄÄÄÄ              "
   PRINT "                         166 * 16  = 2656           "
   PRINT "add next digit                         14  (i.e E) "
   PRINT "                                     ÄÄÄÄ           "
   PRINT "                                      2670 * 16 = 42720   "
   PRINT "                                                      3   "
   PRINT "                                                  ÄÄÄÄÄ   "
   PRINT "                                                  42723    "
   PRINT : PRINT "      :.  A6E3 (base 16) = 42723 (base 10).   "

PRINT "EXERCISES:  Convert the following numbers to base 10:": PRINT
PRINT "1.  11101 (base 2)                      Ans  =  29      ": PRINT
PRINT "2.  3241 (base 5)                       Ans  =  446      ": PRINT

GOSUB PRESS
RETURN
```

```
BOHBASES:
CLS
PRINT TAB(10); "**** CONVERSION BETWEEN BINARY AND OCTAL/HEXADECIMAL ****"
PRINT TAB(10); "********************* NUMBER SYSTEMS ******************":
PRINT
PRINT "  It is possible to convert from binary number system to octal or
hexadecimal"
PRINT "system and vice-versa without going through base ten.   Since 2^3 = 8
and"
PRINT "4^3 = 16, three binary digits (or bits) representing the numbers 0 to 7
can be"
PRINT "indicated by a single octal digit, and four bits representing the
numbers 0 to"
PRINT "15 can be shown by a single hexadecimal digit.  So each octal digit
corresponds"
PRINT "to three binary digits and each hexadecimal digit corresponds to four
binary"
PRINT "digits.  Thus, a binary number can be converted to octal equivalent by
grouping"
PRINT "the bits in threes from the right and converting each individaul group.
"
PRINT "Similarly, the conversion to hexadecimal is done by splitting into four
- bit"
PRINT "groups from the right and converting each individual group.  For
example,"
PRINT "10001111011 (base 2) can be converted as follows:": PRINT
PRINT "10     001      111      011      000     011        (base 2)"
PRINT " 2      1        7        3        0       3         (octal equivalent) ":
PRINT "or"
PRINT "1      0001     1110     1100     0011               (base 2)"
PRINT "1       1         E        C        3                (hexadecimal equivalent)
  "
PRINT : PRINT "That is, 10001111011000011 (base 2)  =  217303 (base 8) or
11EC3 (base 16)"

GOSUB PRESS

CLS
PRINT "Also, 1011111101100010 (base 2) is converted as follows:": PRINT
PRINT "  1          3          7          5          4          2    - Octal"
PRINT "ÄÄÄÄÄ   ÄÄÄÄÄÄ     ÄÄÄÄÄÄ     ÄÄÄÄÄÄ     ÄÄÄÄÄÄ     ÄÄÄÄÄÄ   - binary "
PRINT "  1        011        111        101        100        010           "
PRINT "ÄÄÄÄÄ   ÄÄÄÄÄÄ     ÄÄÄÄÄÄ     ÄÄÄÄÄÄ     ÄÄÄÄÄÄ     ÄÄÄÄÄÄ   -
(hexadecimal)"
PRINT : PRINT "that is, 1011111101100010 (base 2) = 137542 (base 8) or "
PRINT "BF62 (base 16)."
PRINT : PRINT "Note: B = 11 (base 10) and F = 15 (10).": PRINT : PRINT

PRINT "To convert from octal or hexadecimal system to its binary equivalent,"
PRINT "every digit in the octal or hexadecimal number is separately converted
"
PRINT "into binary first, expressing each converted number as a three-digit"
PRINT "number (in the case of octal) or a four-digit number (in the case of"
PRINT "hexadecimal), possibly filling any empty space with a zero.  Then the"
PRINT "three/four-digit numbers in binary are combined to give the desired "
PRINT "equivalent numbers in binary equivalent, we proceed as follows:"
PRINT : PRINT "5 = 101; 7 = 111; and 3 = 011,  (all in base 2)."

GOSUB PRESS
```

```
CLS
PRINT "Similarly, to convert 3AF (base 16 to its binary equivalent we have:"
PRINT : PRINT "3 = 001;  A = 10 (base 10) = 1010 (base 2) and"
PRINT "F = 15 (base 10) = 1111 (base 2)"
PRINT : PRINT ":. 3AF (base 16) = 001110101111 or 1110101111 (base 2) ignoring
the "
PRINT "leading zeros."

PRINT : PRINT "EXERCISES:": PRINT "1.  Convert 567 (base 8) to its binary
equivalent."
PRINT "                ANS:           101110111)": PRINT
PRINT "2.  Convert 1A9 (base 16) to binary number.     (ANS:  110101001))":
PRINT
PRINT "3.  Convert 110011010 to its octal equivalent.    (ANS:   632))": PRINT
PRINT "4.  Convert 11011110111110 to hexadecimal unmber.   (ANS:  37BE)"

PRESS: SLEEP (4): COLOR 27

LOCATE 23, 15: PRINT "Press any key to continue"
   IF INPUT$(1) = "" THEN
      END IF

COLOR 7
RETURN

END FUNCTION

FUNCTION Divi$
prevD1:
n$ = DiviBase1$
SELECT CASE n$
        CASE "M"          'Main Menu
        PRINT "You selected main menu ..."
        CASE "Q"         'Q
        Divi = "Q"
        EXIT FUNCTION
END SELECT
prevD2:
n$ = DiviBase2$
SELECT CASE n$
        CASE "M"          'Main Menu
        PRINT "You selected main menu ..."
        CASE "P"
        GOTO prevD1
        CASE "Q"         'Q
        Divi = "Q"
        EXIT FUNCTION
END SELECT
prevD3:
n$ = DiviBase3$
SELECT CASE n$
        CASE "M"          'Main Menu
        PRINT "You selected main menu ..."
        CASE "P"
        GOTO prevD2
        CASE "Q"         'Q
        Divi = "Q"
        EXIT FUNCTION
END SELECT
```

```
prevD4:
n$ = DiviBase4$
IF n$ = "P" THEN
        GOTO prevD3
ELSE
        Divi = "Q"
       END IF

END FUNCTION

FUNCTION DiviBase1$
CLS
LOCATE 1, 19: PRINT "Division in Various Number Bases           Screen 1/4"
PRINT
PRINT "Division of numbers in any number system is performed exactly like in
the"
PRINT "denary number system.  The base of any unmber system in which division
is"
PRINT "to be performed must be used exactly as 10 is used in the denary
system."
PRINT "In the case of binary number system each quotient digit is either a 0
or a"
PRINT "1 and this simplifies division based two."
PRINT "   When dividing, the familar trial and error technique is used.  To be
able"
PRINT "to divide effectively in any number base, especially to use the long"
PRINT "division method, you need to be conversant with multiplication and"
PRINT "subtraction in that base."

LOCATE 23, 10: PRINT "Press N for next scree, Q to return to main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(81)
DiviBase1 = s$


END FUNCTION

FUNCTION DiviBase2$
CLS
PRINT "                                              Screen 2/4"
PRINT "Examples:  Below are some examples of how to divide in various number
base."
PRINT "a) Base 2                             b) Base 6"
PRINT "       10111                   4231     "
PRINT "      ÄÄÄÄÄÄÄÄÄ               ÄÄÄÄÄÄÄ        "
PRINT "  101³ 1110011            23 ³ 150153            "
PRINT "  ÄÄÄ  101                ÄÄÄ  104                 "
PRINT "       ____                    ___               "
PRINT "       1000                    101         "
PRINT "        101                     50           "
PRINT "        ___                     ___          "
PRINT "        101                     115       "
PRINT "        101                     113    "
PRINT "                                 ___   "
PRINT "        11100011 divide by 101 = 1011    23      "
PRINT "       (in base 2                        23        "
PRINT
PRINT "                 150153 divide by 23 = 4231"
PRINT "                         (in base 6)"
```

```
PRINT

LOCATE 22, 10: PRINT "Press N for next screen, P forprevious Screen, M to
return to manin menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
DiviBase2$ = s$

END FUNCTION

FUNCTION DiviBase3$
CLS
PRINT "                                                     Screen 3/4"
PRINT "Examples continue ..."
PRINT
PRINT " c) Base 16                          d) Base 8"
PRINT "         2B7C                   216     "
PRINT "      ÄÄÄÄÄÄÄÄÄ              ÄÄÄÄÄÄÄ        "
PRINT "  34 ³ 8D530              65 ³ 16633      "
PRINT "  ÄÄÄ  68                 ÄÄÄ  152              "
PRINT "       ___                     ___              "
PRINT "       255                     143      "
PRINT "       23C                      65         "
PRINT "       ___                     ___         "
PRINT "       193                     563      "
PRINT "       16C                     563   "
PRINT "       ___                     ___   "
PRINT "       270                      -   "
PRINT "       270                      -"
PRINT "       ___                                        "
PRINT "8D530 divide by 34 = 2B7C            166333 divide by = 217"
PRINT "(in base 16)                         (in base 8)"

LOCATE 22, 10: PRINT "Press N for Next Screen, P for Previous Screen, M to
return to main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
DiviBase3$ = s$

END FUNCTION

FUNCTION DiviBase4$
CLS : T = 0
ans2$ = " "
PRINT "                                                     Screen 4/4"
PRINT
PRINT "Exercise 1.  What is 11001011 divide by  111 in base 2 ?"
70 INPUT "Result is"; ans
  IF ans = 11101 THEN
  PRINT "That is correct!"
  ELSE
  T = T + 1
  IF T = 2 THEN
    PRINT " Repeat this module again before re-trying the exercise"
    PRINT "The correct answer is 11101"
    ELSE
    PRINT "Please, try again"
    GOTO 70
    END IF
    END IF
    PRINT

    PRINT " Exercise 2.  Divide 77612 by 31 in base 8": T = 0
80 INPUT "Result is"; ans2$
   IF ans2$ = "2432" THEN
   PRINT "Very good! that is the correct answer.  Up you!"
   ELSE
   T = T + 1
   IF T = 2 THEN
   PRINT "Repeat this module again before re-trying the exercise"
   PRINT "The correct answer is 2432"
   ELSE
   PRINT "Please, try again"
   GOTO 80
END IF
END IF
PRINT
  LOCATE 22, 5: PRINT "P for Previous Screen; Q to return to the main menu"
  DO
  s$ = INKEY$
  LOOP UNTIL s$ = CHR$(80) OR s$ = CHR$(81)
  DiviBase4 = s$


END FUNCTION

FUNCTION Intro$
PrevE1:
n$ = IntroBase1$
SELECT CASE n$
        CASE "M"        'Main Menu
        PRINT "You selected main menu ..."
        CASE "Q"        'Quit
        Intro = "Q"
        EXIT FUNCTION
END SELECT

PrevE2:
n$ = IntroBase2$
SELECT CASE n$
        CASE "M"        'Main Menu
        PRINT "You selected main menu ..."
        CASE "P"
        GOTO PrevE1
        CASE "Q"        'Quit
        Intro = "Q"
        EXIT FUNCTION
END SELECT
prevE3:
n$ = IntroBase3$
SELECT CASE n$
        CASE "M"        'Main Menu
        PRINT "You selected main menu ..."
        CASE "P"
        GOTO PrevE2
        CASE "Q"        'Quit
        Intro = "Q"
END SELECT
END FUNCTION

FUNCTION IntroBase1$
CLS
LOCATE 1, 19: PRINT "Introduction To Number Bases                    Screen1/3"
PRINT
PRINT "   A number System is a set of symbols and rules intendded to"
PRINT "represent mumber.  It is likely consisting of ordered set of symbol"
PRINT "called DIGITS with rules of addition, subtraction, multiplication and"
PRINT "division.  Examples of number systems include decimal or denary number"
PRINT "system, binary, octal and hexadecimal number systems.  These are
systems"
PRINT "with ten, two, eigth and sixteen as their bases respectively."
PRINT "  The Base or Radix of a number system is the number of istinguishable"
PRINT "symbols or digits used in representing that system.  The base (radix)"
PRINT "is always a positive integer and it is fixed through out a given
number"
PRINT "system.  For example, the base of the decimal or denary number"
PRINT "system is 10 because there are ten distinct symbols or digits (0 and
9)"
PRINT "use in the system.  Also, the base of binary number system is 2 since"
PRINT "there are two distinct digits (0 and 1) to use.  In the decimal"
PRINT "system, any number in that system, whatever may be the value, can be"
PRINT "written with the combination of the digits, 0 to 9.  Also, in the
binary"
PRINT "system any number in that system, no matter how huge large, can be
expressed"
PRINT "with the combination of 0 and 1.  This applies to all other number"
PRINT "systems, whatever the base may be."
PRINT "   Two figures are added and if there are more than two figures to
added"
PRINT "their addition is done in pairs before the subsequent results are"
PRINT "similarly added."

LOCATE 23, 10: PRINT "Press N for Next Screen, Q to Return to Main Menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(81)
IntroBase1 = s$

END FUNCTION

FUNCTION IntroBase2$
CLS
PRINT "                                                 screen 2/3"
PRINT "   Other important number systems include the Octal and the
Hexadecimal"
PRINT "number systems.  These are numbers in base 8 and 16 respectively.  it
should"
PRINT "be noted that if a number system has a base b, then the greatest digit"
PRINT "used in that system is b-1.  For instance, in base ten the greatest
digit is"
PRINT "10-1 = 9; in bas two the greatest digit is 2-1 = 1; in base five the
gratest"
PRINT "digit is 5-1 = 4 and in base sixteen it is 16-1 = 15 which is normally"
PRINT "represented by letter F.  This is because if the base of a number is"
PRINT "greater than nine, to avoid confusion in the representation of of two-
digit"
```

```
PRINT "symbol '101', '11', '12', etc, letters of alphabets are used to
represent"
PRINT "them.  So, 101 = A; 11 = B; 12 = C; 13 = D; etc."
PRINT "   The base of every given number system is always indicated as a"
PRINT "subscript in the numbers except where the context is very clear.  For
example,"
PRINT "213 and 4752 respectively indicate that the numbers are in base 8 and
10"
PRINT "When only decimal numbers are used, the base may not be indicated if"
PRINT "the context is clear."

LOCATE 22, 10: PRINT "Press N for Next Screen, P for Previous Screen, M to
return to Main Menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
IntroBase2$ = s$

END FUNCTION

FUNCTION IntroBase3$
CLS
PRINT "                                             screen 3/3"
PRINT "Conversion of numbers from one number system/base to an equivalent
value"
PRINT "in another system/base is possible.  This can be achieved by using some
methods"
PRINT "of conversion as explained under the conversion module."
PRINT "   Any number in decimal system can convert to an eqvivalent value in
any other"
PRINT "base abd vice - visa.  But conversion between any other two bases or
systems"
PRINT "different from base ten is always easily done through base ten.  That
is,"
PRINT "to convert from, say, base 3 to base 8, the number in base 3 is first
converted"
PRINT "to base ten 10 and then its equivalent in base 10 is converted to the
required"
PRINT "base 8"
PRINT "   Arithmetic Operations - addition, subtraction, multiplication and
division"
PRINT "can be carried out in any given number system with any base in a
similar"
PRINT "manner as it is obtained in the decimal system.  However, one must
always bear"
PRINT "in mind whateve is the base being used and the digits associated with
that base"
PRINT "It shoul be noted that none of the arithmetic operations can be carried
out"
PRINT "upon any set of numbers unless the numbers are in the same base."

LOCATE 22, 10: PRINT "Press P for Previous Screen, Q to Return to Main Menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(80) OR s$ = CHR$(81)
IntroBase3$ = s$
END FUNCTION

FUNCTION MultBase1$
```

```
CLS
LOCATE 1, 19: PRINT "Multiplication in Various Number Bases          Screen
1/4"
PRINT
PRINT "   The idea of Multiplication of numbers in any given base is based on"
PRINT "the knowledge of addition in that base, since multiplication is just
like"
PRINT "repeated addition.  Just like addition, multiplication in a binary
operation"
PRINT "and it is both commutative and associative, that is, the order in which
any"
PRINT "given two numbers are multiplied do not matter."
PRINT "   Similar to what is done when denary numbers are multiplied, if the"
PRINT "product of any given two digits (in any given base) is greater than
the"
PRINT "base, the product 0, 1, ..., 7 are the figures involved.  is divide by
the"
PRINT "base, the reminder is written down and the quotients is carried to the"
PRINT "next digit position on the left.  For instance, the product of 2 and 4"
PRINT "8 (in base 10, which is greater than 5, the base in use) and the figure
8"
PRINT "cannot be used in base 5 number, 8 is divided by 5 giving a quotient
of1"
PRINT "and a reminder of 3.  Wrting down the reminder and carrying the
quotient"
PRINT "to the left we obtain 13.  That is, 2 x 4 = 13 (in base 5).  Products
of other"
PRINT "base are found in the same manner."

LOCATE 23, 10: PRINT "Press N for next Screen; Q to return to main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(81)
MultBase1 = s$


END FUNCTION

FUNCTION MultBase2$

CLS
PRINT "                                              Screen 2/4"
PRINT "  In the case of binary number multiplication is simple if one
understands"
PRINT "binary addition very well.  This is because as t any point in time the
"
PRINT "product of any two digits is either '0' or '1'.  It should also be
noted"
PRINT "that numbers can be multiplied if the are in the same base.  "
PRINT : PRINT "Exa,ples:  The following multiplication is done in the
indicated bases:"
PRINT : PRINT "a)  Base 2                                   b) Base 8       "
PRINT "           1 1 0 1 1                                3 6 7 5         "
PRINT "           x   1 1 1                              x 2 1 3         "
PRINT "                                                                  "
PRINT "           1 1 0 1 1                       +   1 3 4 6 7     "
PRINT "       + 1 1 0 1 1                            3 6 7 5           "
PRINT "       1 1 0 1 1                              7 5 7 2           "
PRINT "                                                                     "
PRINT "      1 0 1 1 1 1 0 1                      1 0 3 1 6 3 7            "
PRINT "     ===============                         ==============     "
LOCATE 22, 10: PRINT "Press N for Next screen, P, for previous screen; M to
return to main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
MultBase2$ = s$


END FUNCTION

FUNCTION MultBase3$
PRINT "                                                    Screen 3/4"
PRINT "More Exercises in Multiplication:"
PRINT
PRINT "       c) Base 5                       d) Base 16        "
PRINT "       4 1 3 2                            B 3 9 A        "
PRINT "        x  4 3                            x   2 D         "
PRINT "                                                       "
             _______                             _______
PRINT "       1 2 3 0 1                          9 1 E D 2"
PRINT "       3 2 1 3 3                          1 6 7 3 4          "
PRINT "       _________                          _________    "
PRINT
PRINT "       3 3  4 1 3 1                       1 F 9 2 1 2    "

LOCATE 22, 5: PRINT "Press N for next Screen; P for previous Screen; Q to
quit"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
MultBase3$ = s$

END FUNCTION

FUNCTION MultBase4$
CLS : T = 0
ans2$ = " "
PRINT "                                         Screen 4/4"
PRINT "Exercises:                                                      "
        PRINT "Exercise 1.  Find the product of 10101 and 101 in base 2"
50 INPUT "Result is"; ans
  IF ans = 1101001 THEN
  PRINT "Very good!  This is correct"
ELSE
T = T + 1
IF T = 2 THEN
        PRINT "Repeat this module again before re-trying the exercise": PRINT
        PRINT "The correct answer is 1101001"
ELSE
        PRINT "Pleaser, try again"
        GOTO 50
  END IF
END IF
        PRINT "Exercise 2.  What is 5EB3 x 9D in hexadecimal system ?": T = 0
60 INPUT "Result is "; ans2$
  IF ans2$ = "2A13C7" THEN
        PRINT "That is a good answer!  Good trial, keep it up.  Ok ..."
ELSE
T = T + 1
IF T = 2 THEN
```

```
        PRINT "Repeat this module again before re-trying the exercise"
        PRINT "The corect answer is 3A13C7"
ELSE
        PRINT "Please, try again"

        GOTO 60
END IF
  END IF

  LOCATE 22, 5: PRINT "P for previous Screen; Q to return to the main menu"
  DO
  s$ = INKEY$
  LOOP UNTIL s$ = CHR$(80) OR s$ = CHR$(81)
  MultBase4 = s$

END FUNCTION

FUNCTION Multiplication$
prevC1:
n$ = MultBase1$
SELECT CASE n$
        CASE "M"        'Main Menu
        PRINT "You selected main menu ..."
        CASE "Q"        'Q
        Multiplication = "Q"
        EXIT FUNCTION
END SELECT

prevC2:
n$ = MultBase2$
SELECT CASE n$
        CASE "M"        'Main Menu
        PRINT "You selected main menu ..."
        CASE "P"
        GOTO prevC1
        CASE "Q"        'Q
        Multiplication = "Q"
        EXIT FUNCTION
END SELECT

PrevC3:
n$ = MultBase3$
SELECT CASE n$
        CASE "M"        'Main Menu
        PRINT "You selected main menu ..."
        CASE "P"
        GOTO prevC2
        CASE "Q"        'Q
        Multiplication = "Q"
        EXIT FUNCTION
END SELECT

n$ = MultBase4$
IF n$ = "P" THEN
        GOTO PrevC3
ELSE
        Multiplication = "Q"
END IF

END FUNCTION
```

```
FUNCTION SubtBase1$
CLS
'STATIC$ s$
s$ = ""
LOCATE 1, 19: PRINT "SUBTRACTION IN VARIOUS NUMBER BASES        SCREEN 1/6"
PRINT "  Just like in the case of addition, subtraction in any base is carried
out"
PRINT "out in the same way as it is done in base ten.  Depending on the base
used, the"
PRINT "number of symbols involved is b, where b is the base or radix of the
number"
PRINT "system.  If b < 10 then the figures involved are 0, 1, 2, ..., b-1 and"
PRINT "if b < 10 then the figures involved are 0, 1, 2, ... 9 and A for 10,"
PRINT "B for 11, etc, until the last alphabet representing the number b - 1."
PRINT "  Subtraction in any other base is also carried out in a similar
manner"
PRINT "as in base ten.  The idea of borrowing from the left is applied if a
large"
PRINT "number is to be subtracted from a smaller one, bearing in mind the very
base in use"
PRINT "   For instance, to subtract 17 from 34 (both in base 8 ), since 4 is
less than 7,"
PRINT "borrow 1 from 3 to the right (which takes the positional value of 1 x
8).  Taking"
PRINT "3 from 8 (to add to 4 to give 7) we are now left with 5 (in base 8)"
PRINT "Since 1 has been borrowed from 3, we are left with 2 and subtracting 1"
PRINT "from 2 gives 1 in base 8 position.  The final answer is therefore 15.
That is;"
PRINT "              3 4    "
PRINT "            - 1 7        "
PRINT "            -------           "
PRINT "              1 5     "
PRINT "            =======      "

LOCATE 23, 5: PRINT "N for next screen; Q to Quit to main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(81)
SubtBase1 = s$


END FUNCTION

FUNCTION SubtBase2$
STATIC s$
PRINT "                                              screen 2/6"
PRINT "   Subtraction in any base is carried out in a similar way.  If the
operation"
PRINT "is to be performed in the binary notation, only four entries are
involved:"
PRINT "0 -0 = 0; 0 - 1 = 1 with a borrow of 1 from the next more significant
digit"
PRINT "on the left; 1 - 0 = 1; and 1 - 1 = 0.  The 'borrowing' is handled as
with decimal"
PRINT "numbers.  Subtraction is also a binary operation that is, at any point
one"
PRINT "number is subtracted from another one at a time.  So, if there are more
than"
```

```
PRINT "two numbers whose difference is to be found, the subtraction is carried
out in pairs,"
PRINT "following the normal rules of algebra.  It should be noted that
subtraction between"
PRINT "any two numbers is possible if the numbers are in the same base, or
else"
PRINT "the numbers need to be converted (see module on conversion) into a
common base"
PRINT "of addition."

LOCATE 23, 5: PRINT "N for next screen; P for previous screen;Q to Quit to
main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
SubtBase2$ = s$

END FUNCTION

FUNCTION SubtBase3$
CLS
PRINT "                                                          Screen 3/6"
PRINT "   The following are some examples of how to subtract numbers in a
given"
PRINT "base."
PRINT "Example 1.    Subtraction in base 2                 "
PRINT : PRINT "     1 0 1 1 1 0 1      (Note:  1 - 1 = 0;  0 - 1 is not
possible"
PRINT "            -  1 0 0 1 1 1      so 1 is 'borrowed' from the left which
       "
PRINT "            --------------      gives 1 x 2 as the positional valve to
   "
PRINT "               1 1 0 1 1 0      the right and 2 - 1 = 1; the 3rd digit
from        "
PRINT "            ==============      right now becomes 0 so, since 0 -1 is
not "
PRINT "possible same method is applied to give 1 as the 3rd digit in the
answer"
PRINT "from the right.  Then 0 - 0 = 0 (since the digit 1 in the 4th position
from the"
PRINT "right has been borrowed).  Continuing like this, the remaining digits
in the answer"
PRINT "are obtained."
PRINT : PRINT "Example 2:  Subtraction in base 7."
PRINT "      6 3 5 2         Note: Just as it is done in decimal numbers or base
 "
PRINT "     -1 4 2 5         ten numbers, 2 - 5 is not possible so, 1 is
borrowed  "
PRINT "      -------         from 5 and on moving to the next position on the
right   "
PRINT "      4 6 2 4         the value becomes 1 x 7 = 7 and taking 3 from this"
PRINT "      ========       (to add 2 to give the required 5) the remaining
becomes "
PRINT "4.  Now 5 (the 2nd digit from the right) is less by 1 (already
borrowed)"
PRINT "and we have 4 - 2 = 2."

LOCATE 23, 5: PRINT "N for next screen; P for previous screen;Q to Quit to
main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
SubtBase3$ = s$



END FUNCTION

FUNCTION SubtBase4$

CLS
PRINT "                                              Screen 4/6"
PRINT "   3 - 4 is also not possible and using the same method as above we"
PRINT "obtain 6 as the next digit in the answer and finally 5 (since 1 has"
PRINT "been borrowed from 6) less 1 gives 4.  This is how the answer 4624 is
obtained"
PRINT : PRINT "Example 3.  Subtraction in base 16."
PRINT "      5 A 8 D 6       (note that A, B, C, D and E repectively
represent"
PRINT "    -   E 7 B 9        10,11 12 13 and 14 in base 16"
PRINT "    -----------"
PRINT "      4 C 1 1 D       "
PRINT "    ============          "
PRINT "   Since 6 - 9 is not possible, 1 is borrowed from D and moving to the
next"
PRINT "positon on the right, the value becomes 1 x 16 and (16 + 6) - 9 = 13
(in base 10)"
PRINT "which is D (in base 16).  Since 1 has been borrowed from the 2nd digit
D, we"
PRINT "have C out of which we subtract B (i.e 12 -11 in decimal) giving us 1."
PRINT "   Now 8 - 7 gives 1 and to subtract E from A similar steps as in the
case"
PRINT "of subtracting 9 from 6 are taken since E is greater than A and
therefore"
PRINT "A - E is not possible.  If 1 is borrowed from 5 then we have 1 x 16 + A
- E"
PRINT "(or 16 + 10 - 14 in base 10) which gives C (i.e 12 in base 10)."

LOCATE 23, 5: PRINT "N for next screen; P for previous screen;Q to Quit to
main menu"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
SubtBase4$ = s$


END FUNCTION

FUNCTION subtBase5$
CLS
PRINT "                                                       Screen 5/6"
PRINT "   Finally, since 1 has been borrowed from 5, we now have 4 - 0"
PRINT "which gives 4, the first digit in the answer from the left.  Based on"
PRINT "the above examples and explanations more examples are given below"
PRINT "without further explanations."

PRINT : PRINT "Example 4:  Subtraction in:"
PRINT "a)  Based 8   b)  Base 12      c)  Base 3      d)  Base 9 ": PRINT
PRINT "  3 7 4 1       4 3 B 9        2 1 1 0 2       6 8 3 2 1
 "
PRINT "- 2 6 7 5     - 2 8 6 A      - 1 2 0 2 1     - 3 8 5 0 7    "
PRINT "----------    ----------     ------------     ----------- "
PRINT "  1 0 4 4       1 7 4 B          1 0 0 1       2 8 7 1 3  "
PRINT "=========     ==========     ============     =============        "
PRINT : PRINT "Note that in borrowing any number from the left, the number is"
PRINT "multiplied by the base in use before any number in that position is
added"
PRINT "and then the number to be subtracted is deducted."

LOCATE 22, 5: PRINT "pree N for Next screen; P for Previous screen; Q to Quit"
DO
s$ = INKEY$
LOOP UNTIL s$ = CHR$(78) OR s$ = CHR$(80) OR s$ = CHR$(81)
subtBase5$ = s$

END FUNCTION

FUNCTION SubtBase6$

CLS : T = 0
ans2$ = ""
PRINT "Exercises:                                        Screen 6/6"
PRINT : PRINT "Exercise 1.  Subtract 25473 from 63247 (both in base 8)"

30 INPUT "Result is"; ans
        IF ans = 35554 THEN
        PRINT "That is correct!"
ELSE
T = T + 1
        IF T = 2 THEN
        PRINT "Repeat this module again before re-trying the exercise"
        PRINT "The correct answer is... 35554"
ELSE
        PRINT "Please, try again"
GOTO 30
END IF
END IF
        PRINT "Exercise 2.  Find the difference between 110001010 and 1011101
in base 2": T = 0
40  INPUT "Result is... "; ans$
        IF an2$ = "100101101" THEN
        PRINT "The is the correct answer, very good!"
ELSE
T = T + 1
IF T = 2 THEN
        PRINT "Repeat this module again before re-trying the exercise"
        PRINT "The correct answer is 100101101"
ELSE
        PRINT "Please, try again"
        GOTO 40
END IF
END IF
PRINT

        LOCATE 22, 5: PRINT "P for previous screen; Q to return to the main
menu"
        DO
        s$ = INKEY$
        LOOP UNTIL s$ = CHR$(80) OR s$ = CHR$(81)
        SubtBase6 = s$


END FUNCTION

FUNCTION Subtraction$
PrevBl:
n$ = SubtBase1$
SELECT CASE n$
        CASE "M"                'Main Menu
        PRINT "You selectd main menu ..."
        CASE "Q"                'Q
        Subtraction = "Q"
        EXIT FUNCTION
END SELECT
prevB2:
n$ = SubtBase2$
SELECT CASE n$
        CASE "M"                'Main Menu
        PRINT "You selectd main menu ..."
        CASE "P"
        GOTO PrevB1
        CASE "Q"                'Q
        Subtraction = "Q"
        EXIT FUNCTION
END SELECT
prevB3:
n$ = SubtBase3$
SELECT CASE n$
        CASE "M"                'Main Menu
        PRINT "You selectd main menu ..."
        CASE "P"
        GOTO prevB2
        CASE "Q"                'Q
        Subtraction = "Q"
        EXIT FUNCTION
END SELECT
prevB4:
n$ = SubtBase4$
SELECT CASE n$
        CASE "M"                'Main Menu
        PRINT "You selectd main menu ..."
        CASE "P"
        GOTO prevB3
        CASE "Q"                'Q
        Subtraction = "Q"
        EXIT FUNCTION
END SELECT
prevB5:
n$ = subtBase5$
SELECT CASE n$
        CASE "M"                'Main Menu
        PRINT "You selectd main menu ..."
        CASE "P"
        GOTO prevB4
        CASE "Q"                'Q
        Subtraction = "Q"
        EXIT FUNCTION
END SELECT
```

```
n$ = SubtBase6$
IF n$ = "P" THEN
        GOTO prevB5
ELSE
        Subtraction = "Q"
END IF
END FUNCTION
```

APPENDIX C

* * * SAMPLE OUTPUT FROM CONVERSION PROGRAM * * *

```
Number in Base 10 is 2356:
Converted value in base 5 is
33411


Number in Base 10 is 566:
Converted value in base 3 is
202222

Number in Base 10 is 878:
Converted value in base 7 is

2363

Number in Base 10 is 234:
Converted value in base 20 is

The base should be greater than ONE, should not be TEN and must be less or
equal to 16.
```

* * * SAMPLE OUTPUT FROM CONVERSION PROGRAM * * *

```
The number 1010111 in Base 2 converted to BASE Ten is 87


The number 2303 in Base 6 converted to BASE Ten is 207


The number 2AB3 in base 16 converted to BASE TEN is 10931

The number 28191 in BASE 18 converted to BASE TEN is:

Base must be greater than ONE but less or equal to 16.
```