A Program:
A set of instructions to be executed by a computer to perform any particular task is called as a program. A number of program compiled together to carry out any big task to be executed by a computer is now termed as a software.

A program or instructions are written in a language which is well understood by computer. Machine Language or Binary Codes are most easy to understand by a computer system. So the first language used to write program was machine language. Machine language has some short comings as given below.

1- Very difficult to write programs because binary codes are not easy to remember.
2- A Program made on Machine Language is Machine Dependent and can not be run on other machines.

As every program is specially made for particular type of machine and one program made on a particular machine can not be run on any other machine because different machines have different electronic circuitry and sets of registers. So, need for a new type of language was felt and a new version of programming language came into existence called as "Assembly Language".

Assembly Language:
The first evolution in programming language was the development of "Assembly Language". In assembly language; mnemonics are used to represent operation codes and string of characters to represent the address. Hence machine language coding has one to one correspondence coding in assembly language.

As it is clear that mnemonics and each address replaces the machine code one to one, therefore a program in assembly language is also for a particular machine and hence it is machine dependent. Also, since a computer understands only binary coding so assembly language program has to be converted into machine code. This procedure is done through a special program which acts as a translator to translate assembly language program into machine language. The translator is termed as "Assembler".

Now after the action of assembler we have two end of a program in assembly language.
Source Program
Object Program
Source program is the instructions written in assembly language and object program is the converted program in machine language. Processor gets the coding in machine language and then it processes the program to obtain the final output.
Assembly language also has shortcomings like machine language:

All the mnemonics and address are to be remembered which is a tedious task to do.
Program is still machine dependant.

Classification of Programming Languages:

Low Level Languages

- Machine Language
- Assembly Language

High Level Language

- FORTRAN
- PASCAL
- COBOL
- BASIC etc.

High Level Language:

High Level Languages are type of programming language which uses English as a language to write programs in its own grammar called as "Syntax". Though in programming every word or notations has its specific meaning which may appear inappropriate to us as far as English grammar is concerned but meaning of these words are well defined for respective computer languages. As we have assembler to convert assembly language codes into machine codes, high level language too has a special program called "Compiler" to convert the coding into a code which is easily understood by processor for execution. A compiler firstly compiles all the programs at a time and checks for errors. It displays the errors for debugging.
The main difference between a compiler and assembler is that compiler compiles and checks for errors the entire program at a time whereas an assembler performs this task line by line instruction of the program. It moves to next line only when the first line is free from errors.

Middle Level Language:

As we know that program made on machine language are machine dependant and very difficult to write whereas programs made on high level languages take more time to execute. Hence a new language was developed which eliminated the shortcomings of high level languages and comprised the benefits of low level language. This language was termed as middle level language. C and C++ are the examples of Middle Level Language.

A computer Program:

A computer program is a set of step by step procedural instruction to be given to computer to execute any task on computer. Let's take a common example to execute a task. Suppose we have to prepare tea, then the step by step procedure to make tea would be as under:
Step 1: Collect utensils, tea filter, milk, water, tea leaves, sugar and stove.
Step 2: Ignite the stove.
Step 3: Put the utensil on the stove.
Step 4: Pour requisite quantity of water and milk into the utensil.
Step 5: Add appropriate quantity of sugar and tea leaves.

Step 6: Boil the mixture for 5 minutes.
Step 7: Filter and pour the prepared tea into cups.
Step 8: Serve the tea.

Similarly while programming we have to provide step by step procedural instructions to carry out the operations while taking the syntax and hierarchy of steps in consideration.
A sample program to add two digits and to obtain its sum in BASIC Language is given as under:

```
LET A= 10
LET B= 30
C= A+B
PRINT C
END
RUN
```

BASIC Language:
(Beginners All Purpose Symbolic Instructions Code)
BASIC language comes into the category of High Level Language which can be used for simplex, complex and graphical computer programming. It's easy to write and compile. Errors finding and rectification of the same is also easy. A program made on BASIC occupies comparatively less space in the memory and they take less time for execution.

Its text editor gives us a CUI interface; however mouse pointer gives us user friendly environment while writing a program.
Before using BASIC as programming language one must have to be familiar with the components of the language. A programming language consists of some components such as variable, constant etc, reserve keyword, reserved symbols etc.

Reserve Keywords are certain words which have their specific meaning and are used in programming as per the syntax. If these words are used in program at any place other than the appropriate place, the possibility of errors are likely.

Similarly Reserve Symbols are symbols which are used in programming with their own specific meaning. Reserve symbols are symbols which are used in a program for logical or arithmetical operations. Such as:
(,), ‟, ‟, +, -, %, *, /, <, > etc.
Some Reserve Keywords and Symbols:

Variables:
Variables are those values which can be changed during the execution of the program. Variable are also known as identifiers as they identify any location in memory. In fact a variable acts as a container in memory where different valued data can be stored during execution of the program.

Constant:
Constant are those values which can not be changed during the execution of the program.
Note:- *Though variable are those values which can not be changed during execution , however we can provide a condition for a variable for never to go beyond its limit and this action of applying the condition is called as initialization.*

Type of variables:
In BASIC programming mainly there are two types of variables:
Numeric Variables
String Variables
Numeric variables have no dollar sign and can store only numeric values.
e.g. A, B, A1, A_B etc.
String variables have dollar sign and can store alphabets and alphanumeric values.
e.g. A$, A_B$, XY$, A2$, etc.

Arrays: It is a subscripted variable which represents memory location where memory is divided into many cells and each cell is capable of storing one constant.

Some important commands (Keywords) of BASIC:

REM:	Specifies a non- executable remark statement.
CLS:	Clear the screen.
PRINT:	Display a message on the screen.
LPINT:	Print on Paper.
INPUT:	Gets information from the user and stores it in a variable.
END:	End the Program.
GOTO:	Sends the control to another point in the program. It could take the control back or jump ahead.
LOCATE:	Specifies the Position / Location of display.
FOR-NEXT:	It used to make the computer repeat commands several times. A delay loop is an empty FOR-NEXT loop with no instruction in it.
IF-THEN-ELSE:	To check a condition and take suitable action.
RND:	It is used to generate random number. REN generates a random number between 0 and 1. To generate a random number between 1 and 100, multiply by 100, i.e. RND*100. To generate integral random numbers, say 1 and 100 the command is INT(RND*100).
LEN:	Gives the number of characters in a string.
INKEYS$:	This command accepts a character from the keyboard but unlike the INPUT command does not wait for the enter key to the pressed.

SCREENn, CIRCLE, PSET, LINE and PAINT are all graphic commands,

Example1
To input the name of five fruits, and then display them on the screen at the same time by using any array:

```
REM "Display name of fruits"
DIM Fruit$(5)
FOR count = 1 to 5
INPUT "Type the name of seasonal fruit" Fruit$(count)
NEXT count
CLS
FOR n = 1 to 5
LOCATEn+2.20
PRINT Fruit$(n)
NEXT n
END
```

The second statement contains a DIM command. The DIM command informs the computer how much memory to reserve for an array. Here DIM fruit$(5) implies that there are five elements in this array. If we want to input names of say seven fruits, what will the DIM command be?
In the FOR-NEXT loop we have used 'count' as a variable. Fruit$(count) will mean Fruit$(1), Fruit$(2),…, Fruit$(5) depending on the value of count variable.
For printing we have used another variable 'n'.n+2 in the LOCATE command is simply to space out the printing of names of fruits on the monitor. We could, however, have used the same variable for both the FOR-NEXT loops.

Example2
Write a program to randomly generate integer 1 or 2, 50 times, and see how many times we get number '1' and '2'.

The program for this is as follows:

```
Label for the program
Clear the screen
Initialize random number generation
Define the array size
Generate random numbers
And store in num(1) and in num(2)
Display count
End the program
```

```
REM "Find the number of 1s and 2s"
CLS
RANDOMIZE TIMER
DIM Num(2)
```

```
FOR count 1 TO 50
LET n = INT(RND*2+1)
LET num(n) = num(n)+1
NEXT count
PRINT “Number of 1s were”,num(1)
PRINT “Number of 2s were”,num(2)
END
```

The array in the above program contains only two elements – num(1) and num(2).
Note: We have used a new command RANDOMIZE TIMER in the above program. Let us understand the significance of the command. You know that the RND command generates random numbers. If you give the RND command to generate a sequence of number, say 50 times, you will get the same sequence of random numbers. A way out is o initialize the random number generation y giving the RANDOMIZE TIMER command, after generating each sequence. RANDOMIZE TIMER resets everything, so that you will not get the same sequence of random numbers every time you give the RND command.

Example3

Write a program to calculate the total score of student in five subjects, and to print ‘Good performances’ if the percentage exceeds 90.

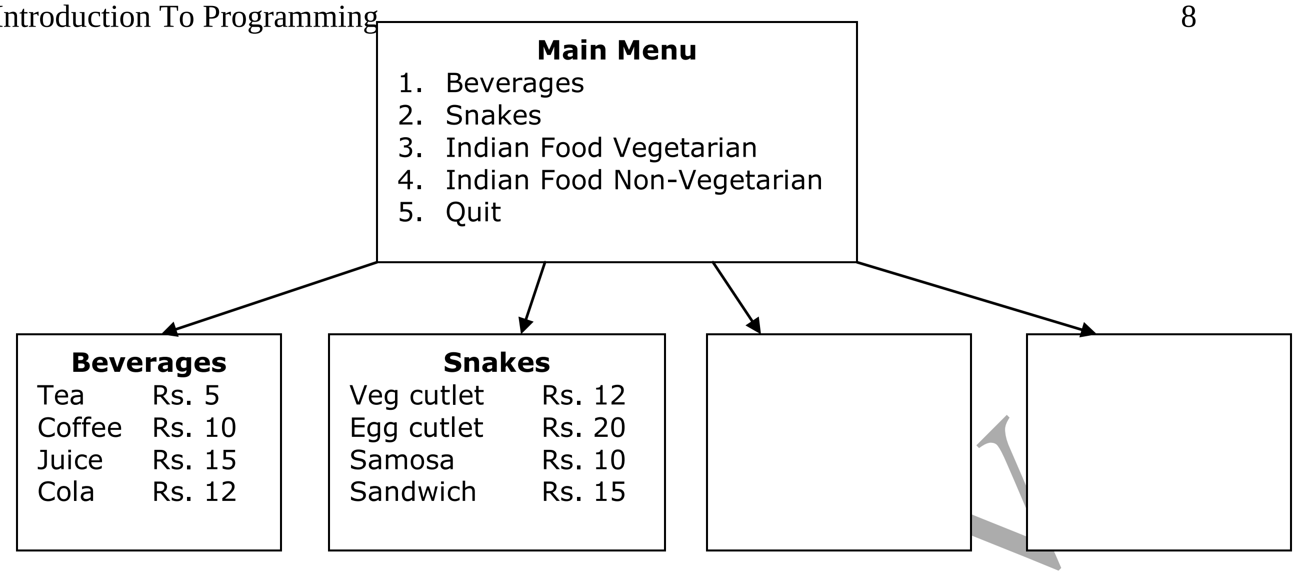
```
REM “Calculate total score”
CLSe
DIM score(5), Subject$(5)
Subject$(1) = “English”
Subject$(2) = “Mathematics’”
Subject$(3) = “Hindi”
Subject$(4) = “Science”
Subject$(5) = “Social Studies”
INPUT “Name of students”,Name$
FOR n = 1 to 5
PRINT Subject$(n);“:”;
INPUT “Enter marks out of 100”,score(n)
Total = total + score(n)
NEXT n
PRINT Name$; “Has score a total of”,total,“marks.”
IF total/5>90 THEN PRINT “Good Performance”
END
```

The ON-GOSUB command is a variation of the GOSUB command. This command allows the program control to go to a certain label, depending on the value that is input in the concerned variable.

The NO-GOSUB command works in exactly the same way. The only different is that the RETURN command in the subroutine returns the control back to the main program automatically, and there is no need to give the 'GOTO main' command.

The given program does the following.
Display the main menu which consist of the categories of foods available (Beverages, Snacks, Indian Food Vegetarian, Indian Food Non-Vegetarian).
Depending on the customer's choice of category (which gets saved in the variable 'choice'), the program control is transferred to one of the subroutines by the ON-GOTO command.
In the subroutine, the INKEY$ command makes the computer wait till a key is pressed by the customer. Ones a key is pressed, the control goes to the mane menu.
Given below is the main menu program and two subroutines and snakes.

```
REM "Menu program"
Main: CLS
PRINT "1. Beverages"
PRINT "2. Snakes"
PRINT "3. Indian Food Vegetarian"
PRINT "4. Indian Food Non-Vegetarian"
PRINT "7. Quit"
INPUT "Type the number of your choice", choice
ON choice GOTO Beverages, Snakes, Indian Food Vegetarian, Indian
Food Non-Vegetarian
Beverages: CLS
PRINT "Tea              Rs. 5"
PRINT "Coffee           Rs. 10"
PRINT "Juice        Rs. 15"
PRINT "Cola         Rs. 12"
PRINT "Press any key to go back to main menu"
Waiting: A$=INKEY$
IF A$ = " "THEN GOTO waiting ELSE GOTO main
Snacks: CLS
PRINT "Veg cutlet       Rs. 12"
PRINT "Egg cutlet       Rs. 20"
PRINT "Samosa           Rs. 10"
PRINT "Sandwich         Rs. 15"
PRINT "Press any key to go back to main menu"
Waiting: A$=INKEY$
IF A$ = " "THEN GOTO waiting ELSE GOTO main
Quit: CLS
END
```



Graphics Command:

The graphics commands are SCREEN n, PSET, LINE, CIRCLE, and PAINT.
SCREEN ➔ A graphics statement that sets the specifications for the display screen and n: is the mode of the screen. Values of n varies from 0-4 and 7-13. eg.
Screen 12
➔ 640X480 Graphics
➔ 80X30 or 80X80 text format.
PSET ➔ A graphics statement that draws a point on the screen.
Syntax : PSET (X1,Y1) [,Color]
LINE ➔ A graphics statement that draws a Line or Box on the screen.
Syntax : LINE (x1,y1)-(x2,y2)[[,Color],[B[F]]]
CIRCLE ➔ A graphics statement that draws a Circle on the screen.
Syntax: CIRCLE(x1,y1),radius[[,color][,[starta][,[enda][,aspect]]]]
Aspect: The ratio of y radius to the x radius
PAINT ➔ A graphics statement that fills an area with the color.
Syntax: PAINT(x1,y1)[,[paint][,[bordercolor][,[backgroundcolor]]]

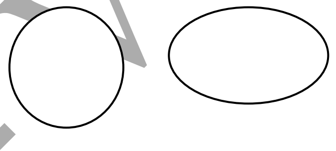
Program:
```
SCREEN 12
PAINT (10,10),3,7
PSET (90,100),10
LINE(50,50)-(200,200),5
LINE(100,100)-(200,200),5,B
LINE(150,150)-(250,250),5,BF
CIRCLE(200,200),200,9,,,1.5
CIRCLE(200,200),200,4,22/14,22/7
END
```
CIRCLE (XI,Y1),r
Draw a circle with center at (X1, Y1) and radius r. The command
CIRCLE(X1,Y1),r,c
Draw the same circle in the colour specified by the number ‘c’.
Drawing an Arc:

Till now have been drawing only complete circles. A position of a circle is an arc. To draw an arc, the starting and ending angles of the arc have to be given, however, these are not given in degrees, but in measure called radius. The relation ship between degrees and radius as follows:
Thus 90 = 3.14/2 radians = 1.57radians, and 360=2x3.14 radians =6.28
So the general formula for converting from degrees to radians is to multiply degrees by 3.14/180. Look at this circle command.

```
CIRCLE(100,200),50,4,0,1.57
```

This draws an arc with center at (100, 200), radius = 50, in red colour (4), with starting angle of 0` and ending angle of 90`.

Making an Ellipse:
An Ellipse is actually a circle stretched horizontally or vertically. The CIRCLE command can be used to draw an ellipse by altering the ratio between the y-axis and X-axis (known as the aspect ratio) as 0.5, the horizontally to form an ellipse (See Figure)


Circle stretched horizontally to make an ellipse

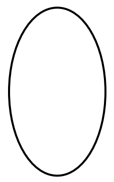
This ratio can be specified after the ending angle, as in the command below. Since we are drawing a circle and not an arc, the angles have not been given. However commas have to be inserted for the missing angles.

```
Circle(100,200),20,3,,,0.5
```

This should give you a horizontally stretched circle. Now try the following with the aspect ratio being greater then 1.

```
Circle(100,200),20,3,,,1.5
```

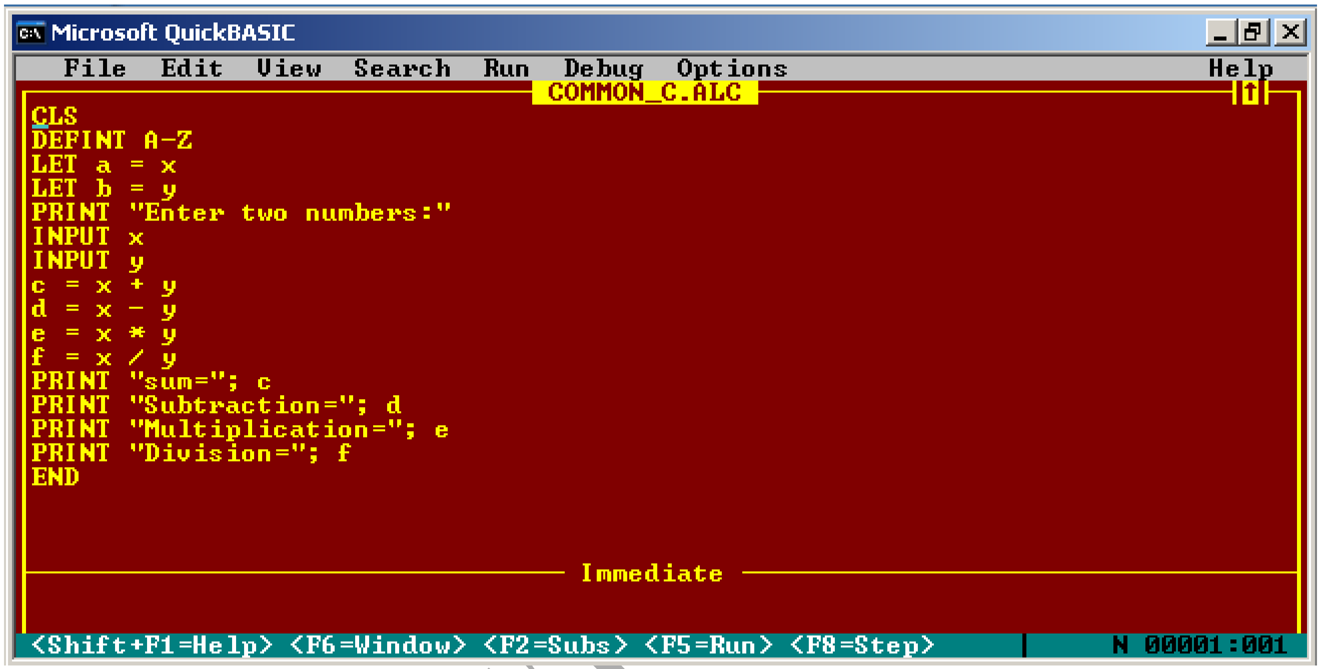
This should give you a vertically stretched circle. (See Figure)



Three-Dimensional effects:
The Circle Command can be used cleverly to create three dimensional (3-D) graphics. Type and execute the following program and see the result.

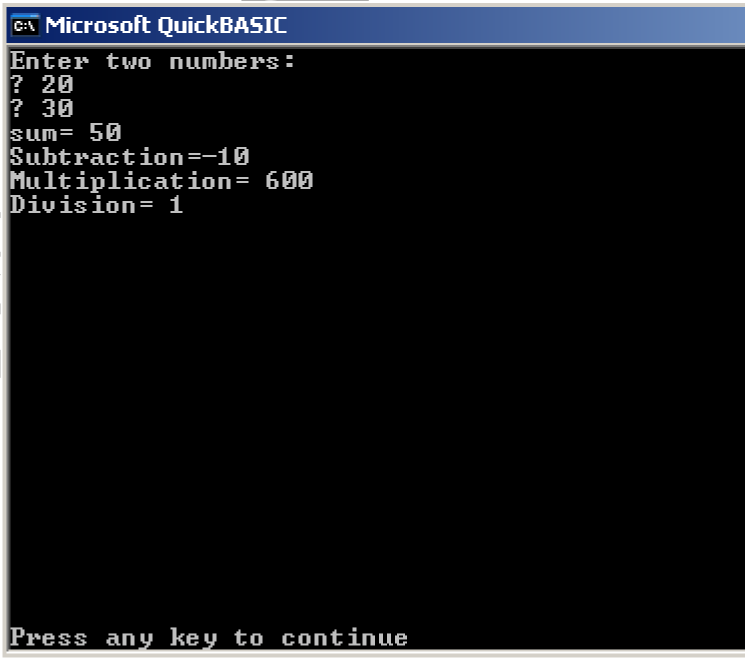
```
REM "3D Effect"
CLS
SCREEN 12
FOR Y=300 to 240 STEP-4
CIRCLE(240,y), y-200,3,,,0.5
NEXT y
END
```

Some Simple QBASIC Programs:
1-Program for Common Calculation, to input two digits and to get their Sum, Subtraction, Multiplication & Division.


Microsoft QuickBASIC
File Edit View Search Run Debug Options Help
COMMON_C.ALC
Immediate
<Shift+F1=Help> <F6=Window> <F2=Subs> <F5=Run> <F8=Step>
N 00001:001

```
CLS
DEFINT A-Z
LET a = x
LET b = y
PRINT "Enter two numbers:"
INPUT x
INPUT y
c = x + y
d = x - y
e = x * y
f = x / y
PRINT "sum="; c
PRINT "Subtraction="; d
PRINT "Multiplication="; e
PRINT "Division="; f
END
```

Code Window

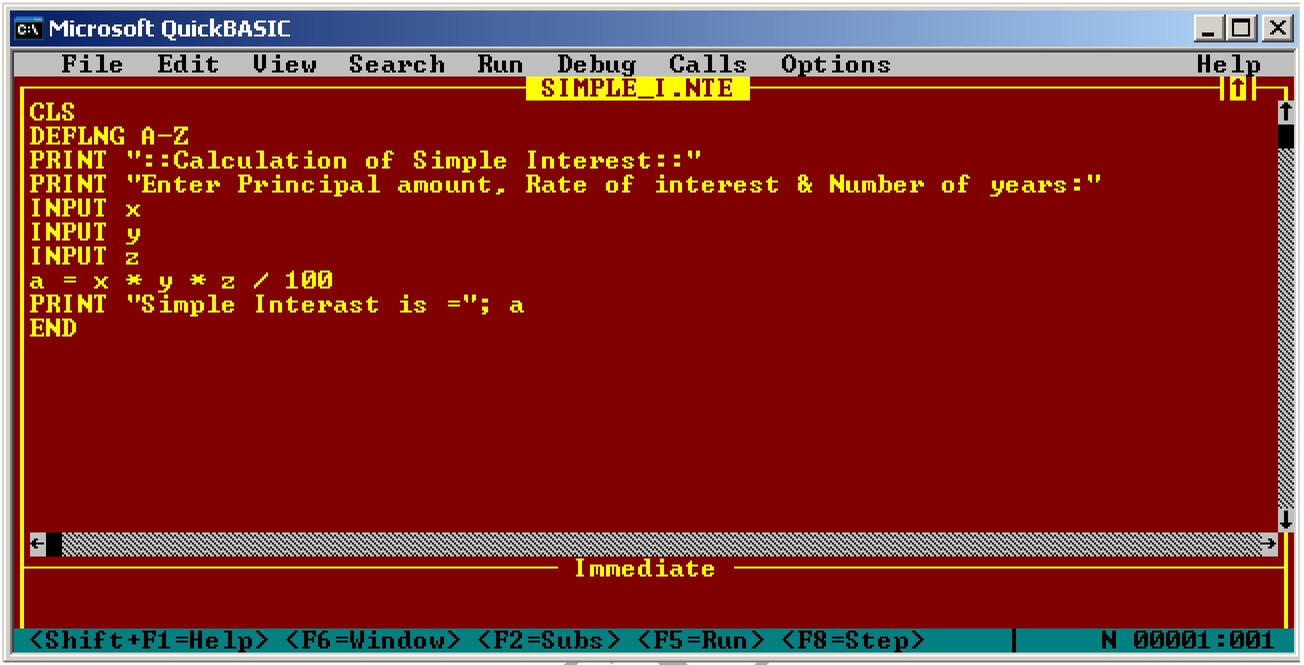


```
Enter two numbers:
? 20
? 30
sum= 50
Subtraction=-10
Multiplication= 600
Division= 1

Press any key to continue
```

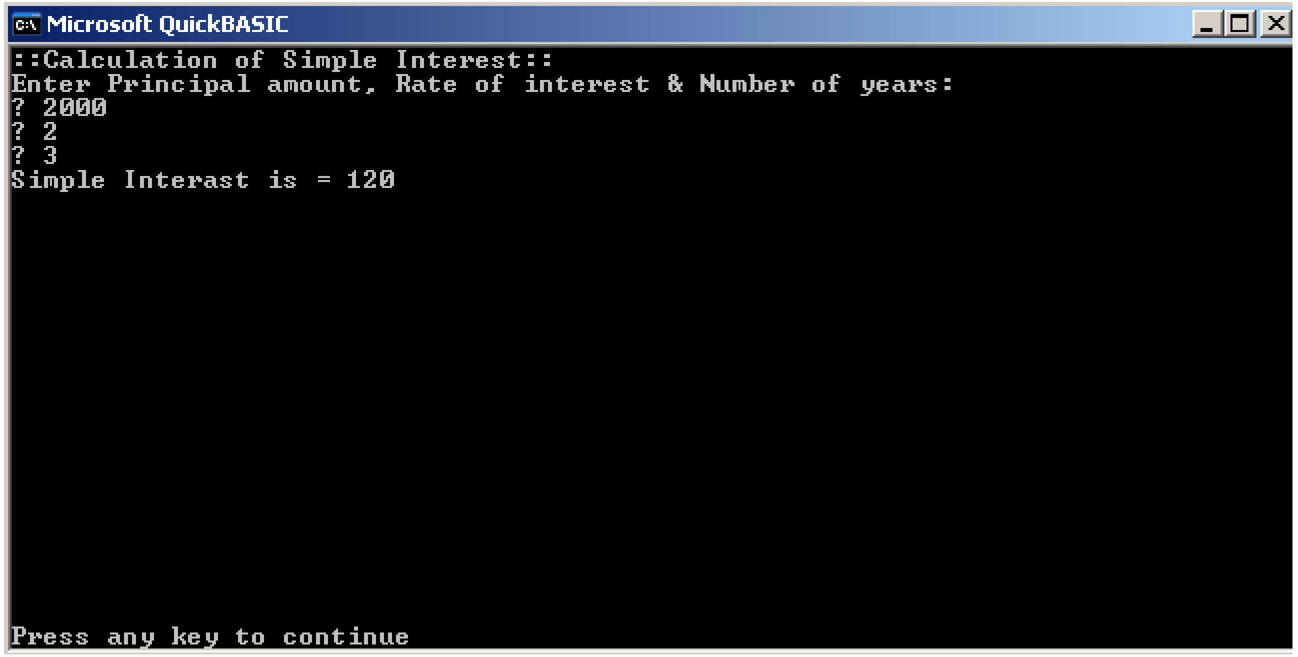
Run Time Window

2- Program to obtain simple interest while entering “Principle amount”, Rate of interest” and Number of years”.


Microsoft QuickBASIC
File Edit View Search Run Debug Calls Options Help
SIMPLE_I.NTE
Immediate
<Shift+F1=Help> <F6=Window> <F2=Subs> <F5=Run> <F8=Step>
N 00001:001

```
CLS
DEFLNG A-Z
PRINT "::Calculation of Simple Interest::"
PRINT "Enter Principal amount, Rate of interest & Number of years:"
INPUT x
INPUT y
INPUT z
a = x * y * z / 100
PRINT "Simple Interast is ="; a
END
```

Code Window



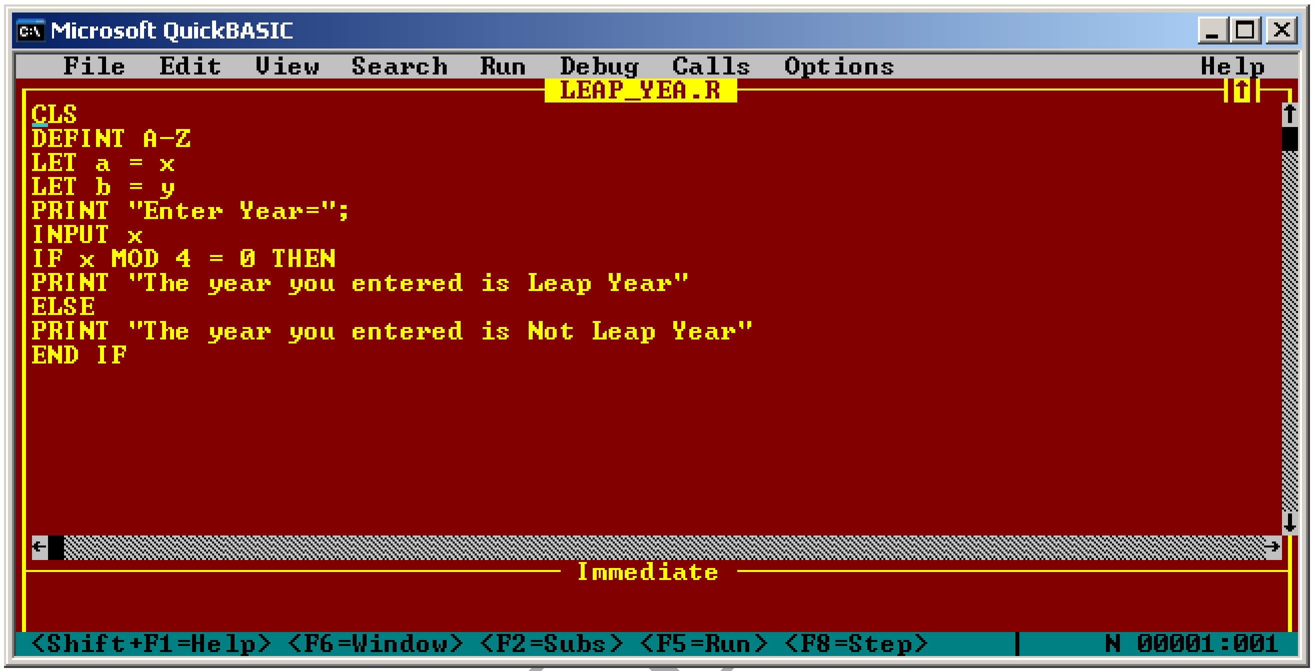
```
::Calculation of Simple Interest::
Enter Principal amount, Rate of interest & Number of years:
? 2000
? 2
? 3
Simple Interast is = 120




Press any key to continue
```

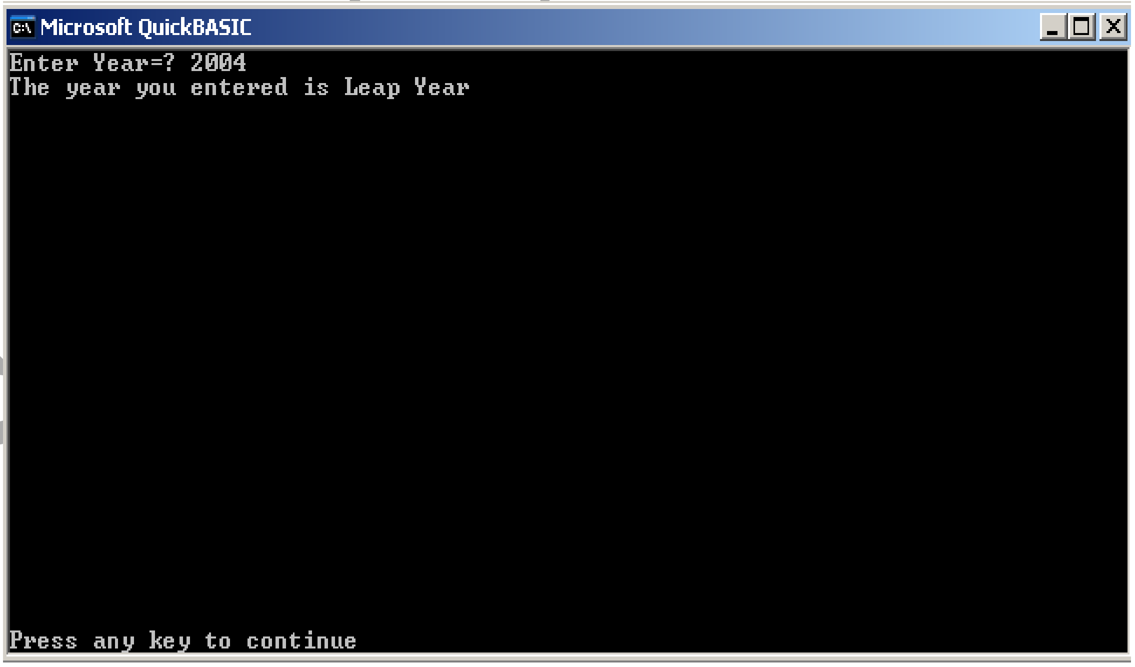
Run Time Window

3- Program to enter "Year" and to obtain whether this year is Leap Year or not?


Microsoft QuickBASIC
File Edit View Search Run Debug Calls Options Help
LEAP_YEA.R
Immediate
<Shift+F1=Help> <F6=Window> <F2=Subs> <F5=Run> <F8=Step>
N 00001:001

```
CLS
DEFINT A-Z
LET a = x
LET b = y
PRINT "Enter Year=";
INPUT x
IF x MOD 4 = 0 THEN
PRINT "The year you entered is Leap Year"
ELSE
PRINT "The year you entered is Not Leap Year"
END IF
```

Code Window

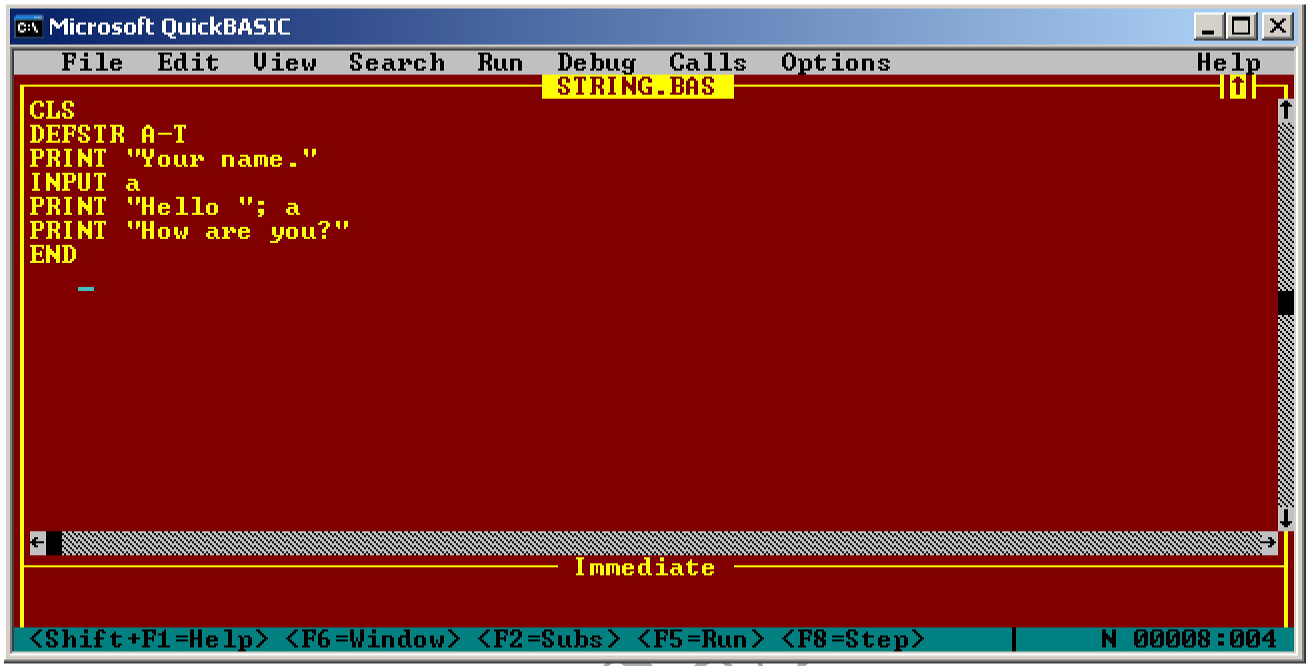


```
Enter Year=? 2004
The year you entered is Leap Year

Press any key to continue
```

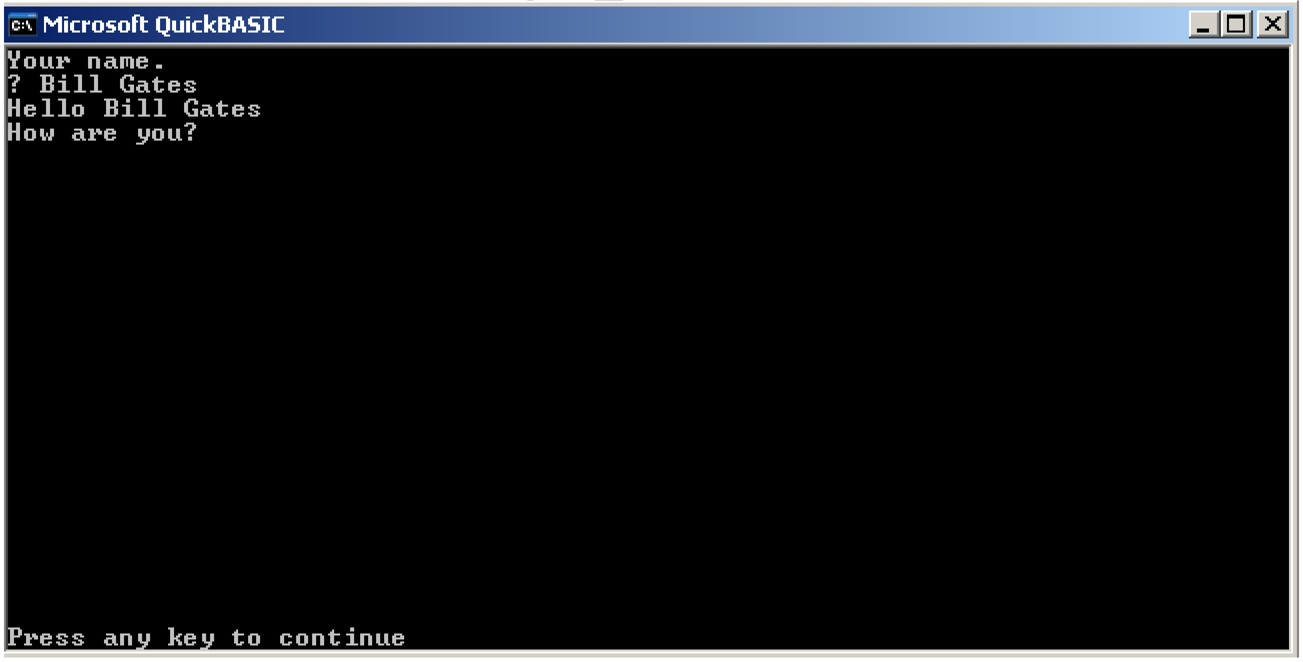
Run Time Window

4- Program to enter any string and obtain any message including that string.


Microsoft QuickBASIC
File Edit View Search Run Debug Calls Options Help
STRING.BAS
Immediate
<Shift+F1=Help> <F6=Window> <F2=Subs> <F5=Run> <F8=Step>
N 00008:004

```
CLS
DEFSTR A-T
PRINT "Your name."
INPUT a
PRINT "Hello "; a
PRINT "How are you?"
END
```

Code Window



```
Your name.
? Bill Gates
Hello Bill Gates
How are you?

Press any key to continue
```

Run Time Window