INTRODUCTION OF BASIC

The BASIC stands for Beginner's All-purpose Symbolic Instruction Code and was first invented by Professors J. G. Kemeny and T.E. Kurtz of Dartmouth college, U.S.A. in 1964. As the name suggests right from the beginning BASIC was designed to be a beginner's language. It is very user-friendly English like language. Since then, this programming language has undergone various changes and modification. With each newer version it has become more user-friendly and useful for general, mathematical as well as commercial uses.

This language is not only easy to understand but you can also develop customized software packages for business application, music and games packages etc. using this language.

What are QBasic Constants?

The value that does not change during the time of execution of a program is called Constant.

Constants are of **two types**:

1. Numeric Constants:

Any numeric value, either a real number, positive or negative, or an integer is called a Numeric constant. ***For example***: 124, + 14, 0, – 7.2 are valid numeric constants. These constants are used for calculations, and should not include any special character.

2. Alphanumeric or String Constants:

A set of characters is called a String. An alphanumeric or string constant consists of a sequence of alphabet and numerical characters, A – Z, a – z, 0 – 9 and special symbols, like % ? & * ( ) ; # ^ " ' : / \, etc., which are written enclosed in double-quotes. String constants are used to represent non-numeric quantities, such as names, text, etc.

For example, "SHAM", "SUM = 44", "162", are some valid string constants.

What is QBasic Variable?

A variable can be defined as a location in the memory, which has been assigned a name and is used to store data temporarily. A variable will continue to hold the value until another value is assigned to it.

There are **two** types of variables:

1. Numeric Variables:

A numeric variable can hold only numeric value. It is represented by an alphabet followed by either another alphabet or a digit. It should not contain any space or symbols, like A, ?, \,

/, @, ;, : ,etc. Underscore can be used instead of space. For example, A, B, A3, ABC, A_4, etc., represent numeric variables in QBasic.

2. Alphanumeric or String Variables:

A string variable is represented by an alphabet followed by a dollar ($) sign. It can also contain letters, digits, underscore symbol, and the last character is always a dollar ($) sign. For example, A1$, RKL$, COMPU1$, etc., are valid qbasic string variables. A string variable can also store string constant in qbasic.

For Example, A2$ (**String Variable**) and “tom” (**String Constant**)

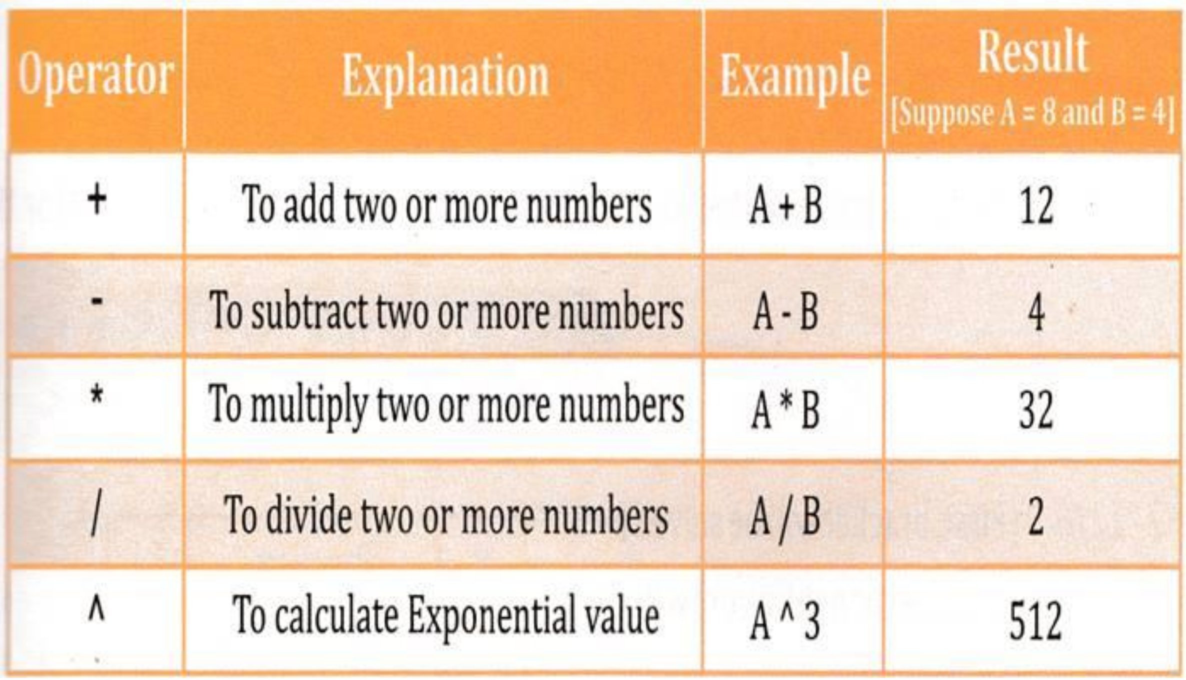
What is QBASIC Operator?

Operators in Qbasic are use to perform different mathematical, logical and comparison operations.

There are **three types of operators** in QBasic:

1. Arithmetic Operators:

A computer performs many arithmetic operations and calculations with the help of arithmetic operators. The basic arithmetic operators use in QBasic are given in the following table:



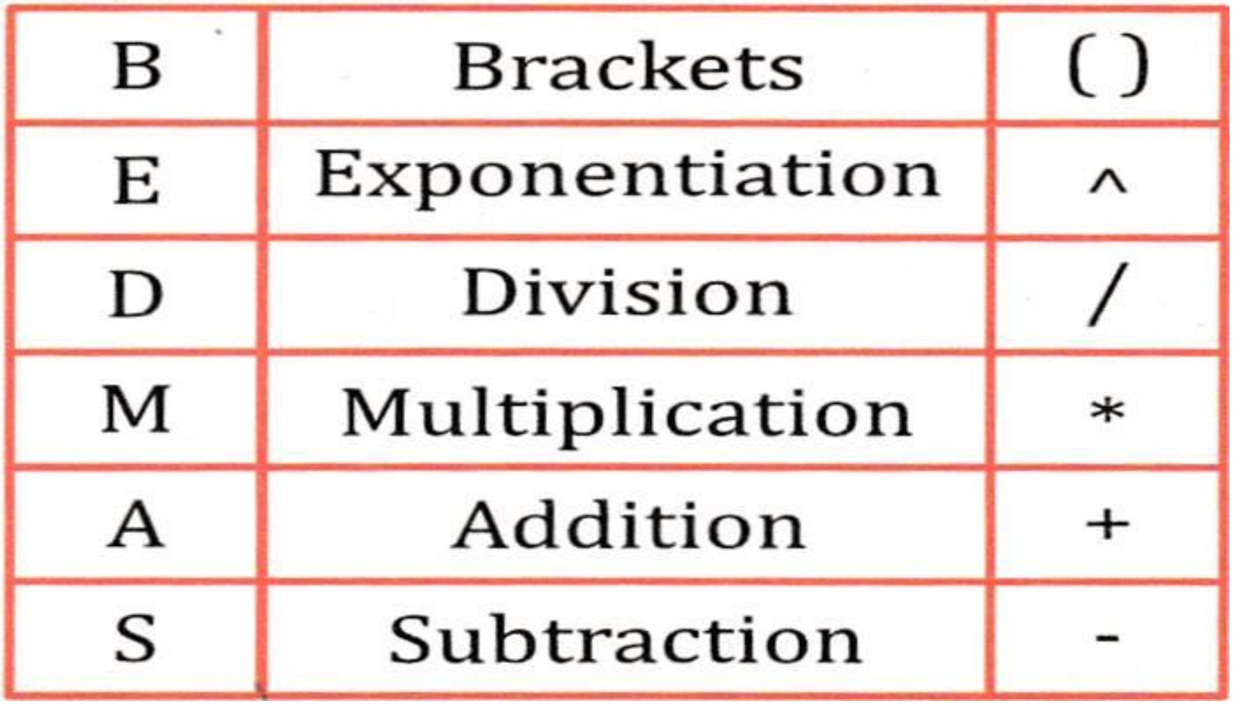
Operator	Explanation	Example	Result [Suppose A = 8 and B = 4]
+	To add two or more numbers	A + B	12
-	To subtract two or more numbers	A - B	4
*	To multiply two or more numbers	A * B	32
/	To divide two or more numbers	A / B	2
^	To calculate Exponential value	A ^ 3	512

2. Relational Operators

These operators are used to compare two values using any of the given relational operators. This can be called a QBasic comparison operator. Below is an image of all the qbasic relational operators:

HIERARCHY OF OPERATIONS in QBASIC

The hierarchy defines the order in which the operators are executed in any Qbasic expression. It uses BEDMAS rule for the hierarchy of operation.



B	Brackets	( )
E	Exponentiation	^
D	Division	/
M	Multiplication	*
A	Addition	+
S	Subtraction	-

STEPS OF SOLVING

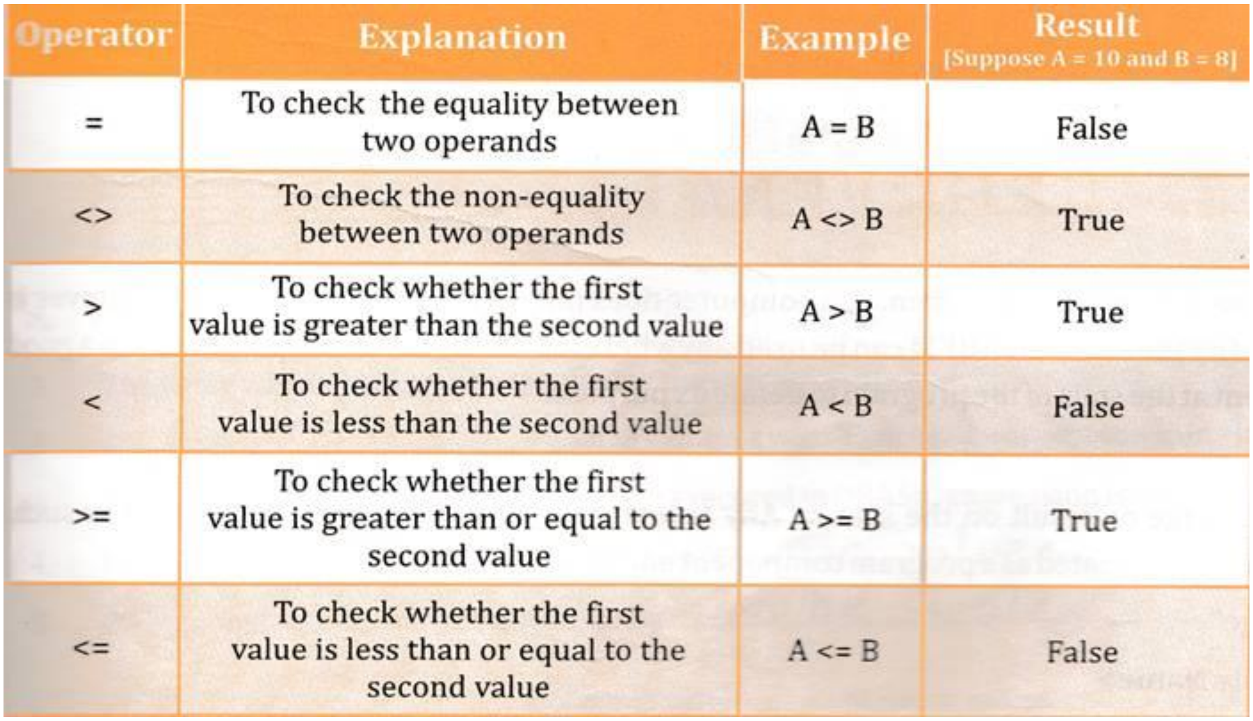
1. First, the bracket will be solved.
2. Then, the exponentiation will be performed.
3. Now multiplication and division will be performed.
4. Now, addition and subtraction will be performed.

Check the numerical expression below to know how it was solved using the BEDMAS rule in QBasic.

```
12*4+(14+4)+2^2-12/6   #the bracket will be solved.
12*4+18+2^2-12/6       #the exponentiation will be performed.
12*4+18+4-12/6         #multiplication and division will be performed.
48+18+4-2              #addition will be performed.
70-2                   #subtraction will be performed
```

What are arithmetic operators in Qbasic programming?

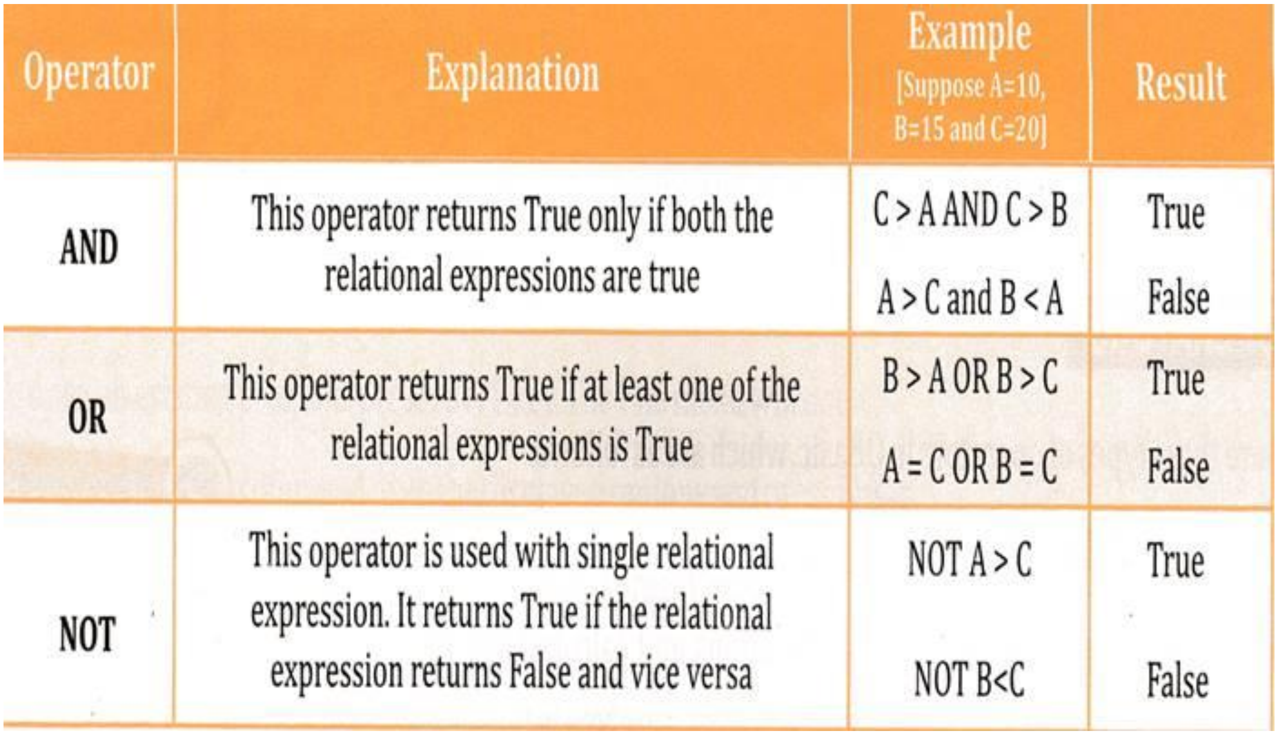
A computer performs many arithmetic operations and calculations with the help of arithmetic operators.



Operator	Explanation	Example	Result [Suppose A = 10 and B = 8]
=	To check the equality between two operands	A = B	False
<>	To check the non-equality between two operands	A <> B	True
>	To check whether the first value is greater than the second value	A > B	True
<	To check whether the first value is less than the second value	A < B	False
>=	To check whether the first value is greater than or equal to the second value	A >= B	True
<=	To check whether the first value is less than or equal to the second value	A <= B	False

3. Logical Operators

Logical operators in QBasic are used to perform logical operations on numeric values. QBasic Logical operators are used to combine two or more relational expressions and return a single value as TRUE or FALSE in a decision. The logical operators are: AND, OR, NOT.



Operator	Explanation	Example [Suppose A=10, B=15 and C=20]	Result
AND	This operator returns True only if both the relational expressions are true	C > A AND C > B A > C and B < A	True False
OR	This operator returns True if at least one of the relational expressions is True	B > A OR B > C A = C OR B = C	True False
NOT	This operator is used with single relational expression. It returns True if the relational expression returns False and vice versa	NOT A > C NOT B<C	True False

What are logical operators in Qbasic?

Logical operators are used to perform logical operations on numeric values. QBasic Logical operators are used to combine two or more relational expressions and return a single value as TRUE or FALSE in a decision. The logical operators are: AND, OR, NOT.

What do u mean by variable?

A variable can be defined as a location in the memory, which has been assigned a name and is used to store data temporarily. A variable will continue to hold the value until another value is assigned to it.

What are relational operators in basic?

These operators are used to compare two values using any of the given relational operators. This can be called a QBasic comparison operator.

o, let’s start with some of the **Qbasic statements**,

1. REM Statement

REM statement is used to give remarks in a QBasic program. The computer does not execute this statement. Whatever is written after REM is ignored by the computer. REM command can be used anywhere and many times in a program. It is always good practice to use the **REM statement** at the start of the program to define its purpose.

Example:

REM sum of two numbers

2. PRINT STATEMENT

The **PRINT statement** in QBasic is used to display any message or result on the screen. Any text written in double-quotes is printed as such. Anything written without quotes is treated as a program component and its value will be displayed.

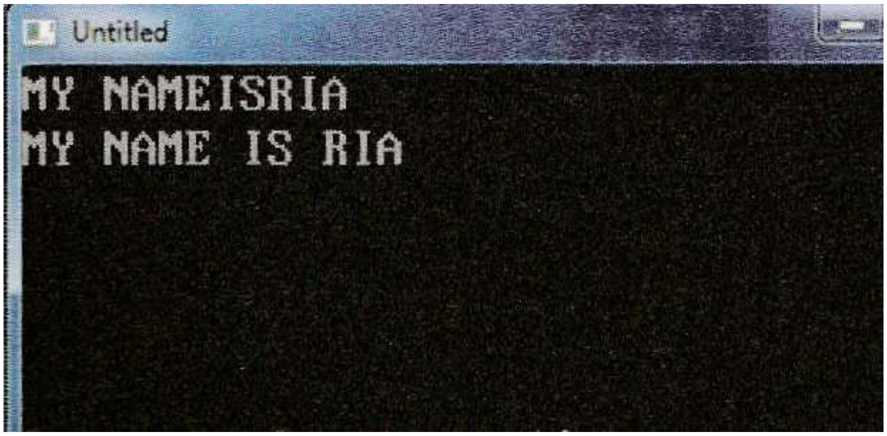
Syntax :
PRINT “Hello World”
PRINT<variable Name>
The **PRINT command** can be used in different variations to print with the output in different formats.

PRINT WITH SEMI-COLON(;)

PRINT with a semicolon is used to display the values one after another, without any space in between.

Syntax: **PRINT** (Variable);(Variable);(Variable);..
Example:

```
LET X$ = “MY NAME”
LET Y$ = “IS”
LET Z$ = “RIA”
PRINT X$;Y$;Z$
PRINT X$;””;Y$;””;Z$
END
```

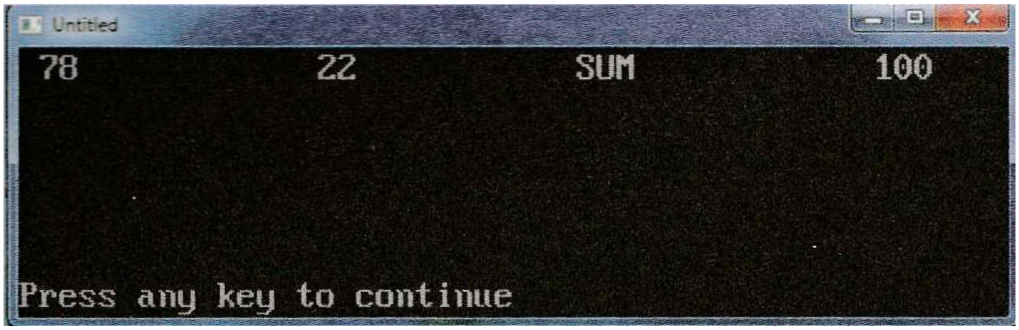


```
MY NAMEISRIA
MY NAME IS RIA
```

PRINT WITH COMMA(,)

The **PRINT statement** is used to display the values one after another with plenty of spaces(normally 14) in between. So, only five values can be printed in one line. In the case of more than five values, the remaining values will be printed on the next line.
Syntax: **PRINT** (Variable),(Variable),…
Example:

```
LET X= 78
LET Y= 22
Z = X + Y
PRINT X, Y, Z
```



```
 78               22               SUM               100




Press any key to continue
```

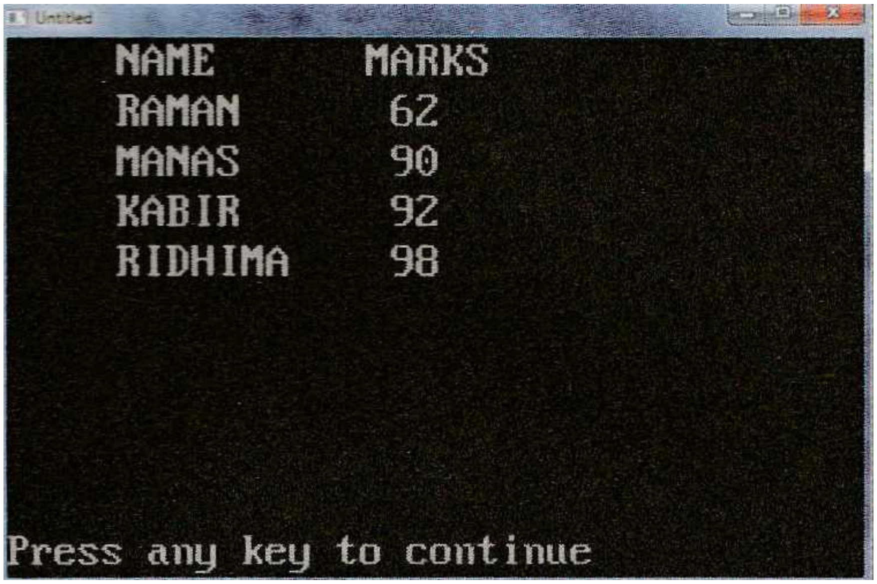
PRINT WITH TAB FUNCTION

The TAB functions are used to move the print position to the column indicated in its argument. This PRINT statement is quite suitable for printing tabular type of results as output.

Syntax:
PRINT TAB (C); “Constant/Variable/Expression”
Where C is the number of column number

PRINT TAB(18); “GOOD EVENING”
Ths message “GOOD EVENING” will be printed on the screen, 18th column onwards.
Example:

```
REM “TAB FUNCTION PROGRAM”
PRINT TAB(10); “NAME”; TAB(20);”MARKS”
PRINT TAB(10); “RAMAM”; TAB(20);62
PRINT TAB(10); “MANAS”; TAB(20);90
PRINT TAB(10); “KABIR”; TAB(20);92
PRINT TAB(10); “RIDHIMA”; TAB(20);98
END
```



```
     NAME        MARKS
     RAMAN        62
     MANAS        90
     KABIR        92
     RIDHIMA      98




Press any key to continue
```

3. INPUT STATEMENT

The **INPUT statement** in QB64 is used to accept a data item from the user and store it in a variable. This statement asks the user to enter data by displaying a question mark (?) during the execution of the program. The program execution is suspended until the user enters the required values and press the Enter key.

Syntax:

INPUT <Numeric or String Variable Name>

Example:

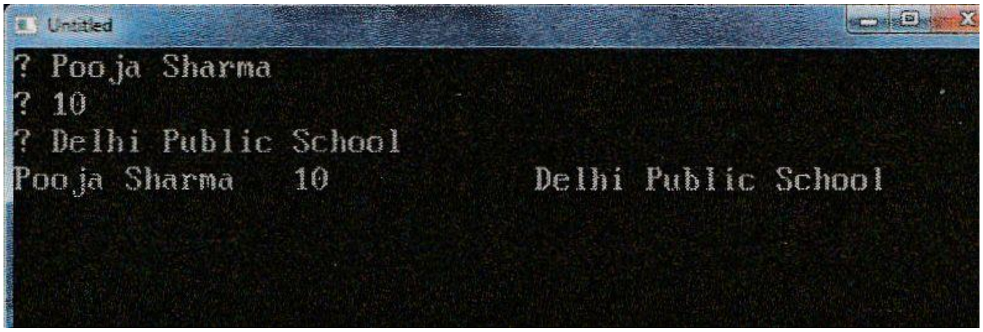
INPUT A

You can have a list of variables in an ***INPUT statement,*** but they should be separated by commas. The corresponding values which you key in should also be separated by commas.

Syntax:

INPUT <Variable1>,<Variable2>,<Variable3>

```
INPUT NAME$
INPUT CLASS
INPUT SCHOOL$
PRINT NAME$, CLASS, SCHOOL$
END
```



```
? Pooja Sharma
? 10
? Delhi Public School
Pooja Sharma    10          Delhi Public School
```

While using the **INPUT statement**, the computer performs the following tasks:

- It stops further execution of the program.
- It prints a question mark on the screen.
- It waits for the user to key in its response and presses the Enter key.
- It stores or assigns the same data item into the corresponding variable mentioned in the INPUT statement.

INPUT STATEMENT USING NUMERIC VARIABLES WITH MESSAGES

To make the input user friendly, the **INPUT statement** can also be used with an optional message, separated from the variable using a semicolon(;).

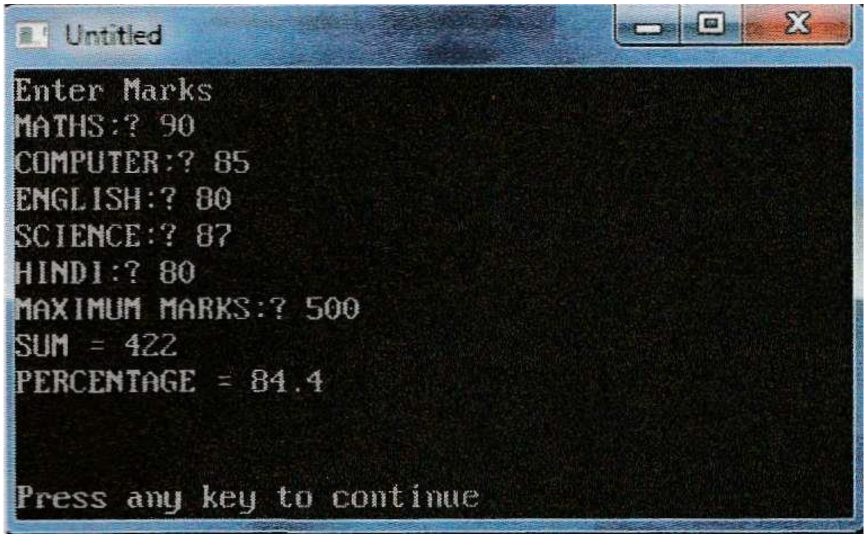
Syntax:

INPUT “MESSAGE”;<VARIABLE>

Examples:

```
CLS

PRINT “Enter Marks”
INPUT “MATHS”;M
INPUT “COMPUTER:”;C
INPUT “ENGLISH”;E
INPUT “SCIENCE”;S
INPUT “HINDI”;H
INPUT “MAXIMUM”;TM
SUM = M+C+E+S+H
PER = (SUM/TM)*100
PRINT “SUM=”;SUM
PRINT “PERCENTAGE=”;PER
END
```



```
Enter Marks
MATHS:? 90
COMPUTER:? 85
ENGLISH:? 80
SCIENCE:? 87
HINDI:? 80
MAXIMUM MARKS:? 500
SUM = 422
PERCENTAGE = 84.4


Press any key to continue
```

4. END STATEMENT

The **END statement** is used to indicate the end of a program. Any statement written after END will not be executed.

Example:

```
CLS
LET A = 2
LET B = 2
C = A+B
PRINT “THE ANSWER IS”,C
END
```

- **EXAMPLE 1:**
- **Write a QBasic program to enter your name and print it.**
- *CLS*
- *INPUT ‘Enter your name’;n$*
- *PRINT ‘The name is’;n$*
- *END*
- Here since the input data is a string then the variable name (n) in which it is to be stored is written after INPUT command followed by $ sign(n$).
- **EXAMPLE 2:**
- ***Write a qbasic program to enter your name, city, country, age and print them.***
- CLS
- INPUT ” Enter the name “;N$
- INPUT ” Enter the city”;C$
- INPUT ” Enter the country”;CO$

- INPUT " Enter the age";A
- PRINT " The name is ";N$
- PRINT " The city is ";C$
- PRINT " The country is ";CO$
- PRINT " The age is ";A
- END
- Here name, city, country are string so its variable name is written with sign **$** before it and since age is a number so the variable name is written directly.
- **Example 3:**
- **Write a program to find the area of a rectangle.**
- ***FORMULA:*** *Area of a Rectangle is length x breadth. Here length is variable **l** and breadth is variable **b**. And is the variable to store the value of the result and print it as output.*
- CLS
- INPUT " Enter the length " ;l
- INPUT " Enter the breadth " ;b
- LET A = l*b
- PRINT" The area of rectangle=" ;a
- END
- **Example 4:**
- **Write a program to find the area of the triangle.**
- ***FORMULA:*** *Area of triangle is ½ x base x height.*
- Here in this program we that variable **b** as base and **h** as height. T is the variable to store the result and print it.
- CLS
- INPUT " Enter the base" ;b
- INPUT " Enter the height" ;h
- LET T = 1/2*b*h
- PRINT" The area of triangle=" ;T
- END
- **Example 5:**
- **Write a program to find the area of the circle.**
- ***FORMULA*** *: Area of a circle is 22/7 x radius^2. Here we use variable **R** as Radius. And **C** is the variable where we store the result.*
- CLS
- INPUT" Enter the radius " ;R
- LET C=22/7*R^2
- PRINT " The area of circle =" ;C
- END
- **Example 6:**
- **Write a program to find the area of the square.**

- ***FORMULA:*** *Area of a square is = Side^2 square units. Here the result is stored in variable "**square**".*
- CLS
- INPUT" Enter the number" ;n
- LET square= n^2
- PRINT" The area of square=" ;Square
- END
- **Example 7:**
- **Write a program to find the volume of the box.**
- ***FORMULA****: Area of a box = length x breadth x height.*
- The variable are **l**, **b** , and **h** for length, breadth and height and the result will be stored in the variable **volume**.
- CLS
- INPUT " Enter the length " ;l
- INPUT " Enter the breadth " ;b
- INPUT " Enter the height " ;h
- LET volume= l*b*h
- PRINT" The volume of box =" ;volume
- END
- **Example 8:**
- **Write a program to find the circumference of the circle.**
- ***FORMULA*** *: circumference = 22/7 x Radius x 2*
- CLS
- INPUT" Enter the radius " ; R
- LET Circumference=22/7*R*2
- PRINT " The area of circle =" ; Circumference
- END
- **Example 9:**
- **Write a program to find out the Simple Interest.**
- *FORMULA : Simple Interest = Principle x Rate x Time / 100*
- CLS
- INPUT " Enter the Principal";P
- INPUT " Enter the Rate";R
- INPUT " Enter the Time";T
- LET I = P*T*R/100
- PRINT " The simple Interest = ";I
- END
- **Example 10:**
- **Write a program to find out the simple Interest and the Amount.**
- ***FORMULA*** *: Simple Interest = Principle x Rate x Time / 100*
- *Amount = Principle + Simple Interest*

- CLS
- INPUT " Enter the Principal";P
- INPUT " Enter the Rate";R
- INPUT " Enter the Time";T
- LET I = P*T*R/100
- LET A= P + I
- PRINT " The simple Interest = ";I
- PRINT " The amount=";A
- END
- **Example 11:**
- **Write a QBasic program to convert temperature from Fahrenheit to Celsius**
- ***Formula (Convert Fahrenheit to Celsius)****: C = (F – 32) * (5 / 9)*
- Here F is the variable that will store the INPUT value. And C is the variable where we will save the result in Celsius.
- CLS
- INPUT "ENTER TEMPERATURE IN FAHRENHEIT"; F
- C = (F – 32) * (5 / 9)
- PRINT "TEMPERATURE IN CELCIUS="; C
- END
- **Example 12:**
- **Write a QBasic program to convert temperature from Celsius to Fahrenheit**
- ***Formula(Convert Celsius to Fahrenheit):*** *F = (**C** * 9/5) + 32*
- Here C is the variable that will store the INPUT value of Celsius. And F is the variable where we will store the result in Fahrenheit, which will be used in a PRINT statement to print the results.
- CLS
- INPUT "ENTER TEMPERATURE IN CELSIUS"; C
- F = (**C** * 9/5) + 32
- PRINT "TEMPERATURE IN FAHRENHEIT ="; F
- END