What is modular programming?
ANS: Modular programming is a technique used to divide our program into many small logical, manageable and functional modules or blocks. Every modular program has one main module and may have many sub modules. The sub module of a program is also called as procedure.

2. Write some advantages of modular programming.
ANS: The advantages of modular programming are:

1. Easy for programmers: they can write the block or modules of program independently
2. It reduces program codes: Same procedures can be used in different numbers of places at any time
3. Increases the readability of program
4. Easy to debug the program
5. Easy to design program

3. Why QBASIC is called modular programming language?
Because:
It allows the programmers to divide the program into manageable and functional modules with the help of sub procedure and function procedure.

Procedure is an independent manageable and functional part or block that solves a particular problem given by the user independently. It is also known as a Sub Module or Sub program.

The types of sub modules in QBASIC are:

Sub Procedure
Function Procedure

5. Define Sub Procedure and write its some important features.

ANS:

A sub procedure is a small manageable and functional part of a program that performs specific tasks and does not return any value to the calling module. A CALL statement is used to call the sub procedure module. A sub program is written with SUB...END SUB statements.

Features:

- It does not return any value
- Sub procedure name can't have a type declaration characters
- Sub procedure name can't be used as a variable
- Sub procedure is called by CALL statement.
- Arguments can be passed to the subprogram by reference and by value method.

6. Define function procedure and write its some important features.

ANS:

A function procedure is a small manageable and functional part of a program that performs specific tasks and returns a value to the main program or calling module. It is written with Function....END FUNCTION statements.

Features:
Function procedure returns a value
Function procedure name has type declaration characters
A function name can be used in an expression
Function procedure can be recursive
arguments can be passed to the sub procedure by reference or by value method

7. Differentiate between sub procedure and function procedure.

ANS:

<table>
<thead>
<tr>
<th>Sub Procedure</th>
<th>Function Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub procedure does not return value to the calling module.</td>
<td>A function procedure returns a value to the calling module.</td>
</tr>
<tr>
<td>Sub procedure name can’t be used in an expression</td>
<td>Function name can be used in an expression.</td>
</tr>
<tr>
<td>Sub procedure is called using CALL statement</td>
<td>Function procedure can be called by statement method using print statement or expression method. (C$=CHECK$(N)</td>
</tr>
</tbody>
</table>

8. What are arguments and Parameters?

ANS:

The consonant and variable enclosed in parenthesis of procedure call statement and that are supplied to a procedure are known as arguments. It is also known as actual parameters.

Variables in a sub module or a procedure declaration which accept data or variable passed to them from the calling module are known as parameters. It is also known as formal parameters.

8. Write short notes on
A. Arguments passing by reference

Arguments are passed to a sub module by reference, which gives procedure access to the actual variable. In this method, the address of each variable is passed to the procedure. The changes made in the procedure's variable will affect the value of arguments. It is the default mode of passing arguments. For example, CALL TEST(A,B)

B. Passing by value method:

Arguments passed by value method do not make any effect to the value of the variable which are passed to the procedure even if they are changed in the procedure. To pass arguments by value method, each argument is enclosed in individual parenthesis in the calling statements.

Example CALL TEST ((A),(B))

C. Main Module and Sub Module

The top level module is known as the main module, which is located at the top of all procedures.

Sub module is a program which is written under the main module. A program may have one or more than one sub modules.

D. Formal and Actual Parameters:

Formal parameters are variables which are used to specify or declare types of data to be passed to the sub procedure either sub or function. For example, SUB MySub(A,B) or Declare Sub MySub(A,B)

Actual or real parameters are arguments which are used to pass real value to the procedure. Actual parameters may be variables or constants. For example, CALL MySub(A,B)
E. Local Variable and global Variable

A variable which is defined in a module and is not accessible to any other modules is known as local variables. It is only accessible to the module where it is defined.

A variable in main module which can be accessed from any module or procedure of a program is known as global variable.

Variable an be made global declaring them with DIM SHARED, COMMON SHARED or SHARED attribute.

For ex

DECLARE SUB TEST (A,B)

COMMON SHARED X

A=5
B=10

CALL TEST (A,b)

PRINT Y;

END

SUB TEST (P,Q)

SHARED Y

X=5
Y=10

PRINT P,Q

END SUB
9. Write the purpose of following statements and functions

A. CLS
   The CALL statement transfer control to sub procedure and statement of sub procedure executes. It is used to execute sub procedure from main module.

A. FUNCTION.......END FUNCTION
   It is a non executable statement that declares the name, parameters and the code that form the body of a function procedure.

B. SUB.....END SUB
   It is used to define sub procedure module. ex. sub test(a,b)

C. DECLARE
   It is used to declare procedures, such as function or sub in a modular programming. It should appear before any executable statements. ex. DECLARE FUNCTION SUM(A,B)

D. SHARED
   It is used to share variables among parts of a modules. It appears only in the sub module. eg shared x,y

E. DIM SHARED
   It makes the variable accessible or global to all modules. It appears in the main module
   DIM SHARED X,Y

F. COMMON SHARED
   It declares variable as global. So that can be shared between main module and sub modules. It appears in main module
   COMMON SHARED X,Y