<u>Functions</u> :-

A function is a subpart of the program which can be invoked from other parts of the program as and when required.

A function is a named unit of a group of program statements. There are three things associated with a function.

1**.** **<u>Function Definition</u>** :

Obviously, a function needs to be defined before it can be used anywhere. The general form of a function definition is :

```
return-type function-name ( argument-list )
{
        body of the function;
}
```

<u>Example</u> :

```
int sum(int a, int b)
{
int sum;
sum = a+b;
return sum;
}
```

This code will add the value of ‘a’ and ‘b’ and return it back to the function when it is called.

2. **<u>Function Prototype/Declaration</u> :**

The prototype is exactly the same as the function definition except that it has no body. It is just used to tell the complier that somewhere in the program, there is a function with these specifications.

<u>Example</u> : The function prototype for the above function will be :

```
int sum(int , int);
```

<u>Note</u> : (i)It is not necessary to include the variable names in the function prototype.

(ii)There is no need for a prototype if the function is defined before the main() function.

3. **Function Call :**
Whenever the function needs to be used, it has to be called by referring to it by it’s name and the arguments(if any).
The function call for the above function will look like : **sum(3,4);**
There are two ways of calling a function – ***Call by Value*** and ***Call by Reference***.
Call by Value : In this case, the values of actual arguments/parameters is copied to the formal arguments/parameters and the function is executed, i.e., the function creates its own copy of arguments and acts on them. Thus, the original values of the variables is not modified.
Call by Reference : In this case, the reference of the variables is passed to the function. A reference means an alias(a different name). Thus, the called function does not create its own copy of the variables, instead, it refers to the original ones by another name i.e., the function acts upon the original variables indirectly. Thus, any changes made to the values reflect back in the original data.

Recursion :-

When a function calls itself in it’s own body, it is called *Recursion*.
Example :

```
int fac(int n)
{
if(n==1) return 1;
else
return n*fac(n-1);
}
```

As is clear from this example, the function fac(int) calls itself again in the else part. This is called recursion.

Practice Problems :

1. Write a function to determine all Pythagorean triplets between 100 and 1000.
Hint : A Pythagorean triplet is a set of three integers i, j, k, such that
(i*i + j*j = k*k).
2. Write a menu driven program for taking the values of radius and printing the area and circumference of a circle.