#### Function in C

A function is a block of code to perform a specific task. A whole program can be divided in to different functions. This will make a complex program to break down in to small and understandable parts or modules.

Before you can call a function, you must create it by declaring it. To declare a function you must put a return type such as void, int, float, or double followed by the name of function and parentheses. In the example below, i declare a function named ReadName.

Example:   
void ReadName()  
{   
 char name[20]; //array of characters to create a string   
 printf("%s","Enter your name: ");   
 scanf("%s",name);  
printf("%s\n","Your name is ",name);  
}   
   
Note: if you use void type, your function will return nothing. However, if you use other type, the function must return value of that type.

### Calling your function

Your function must be called from the main method.   
Example:   
  
void main()  
{  
ReadName();  
 return 0;  
}

### Parameters

Parameters are used to pass values to the function. Parameters must be put in parentheses after the name of the function.

void ReadName(char \*s)  
{  
cout << s;  
}  
  
void main()  
{  
ReadName(“Dara”);  
}

If you want to pass more than one value then you must separate parameters with commas.  
void PrintName(char \*fname, char \*lname)  
{  
printf("%s%s",fname,lname);  
}  
  
void main()  
{  
PrintName(“Dara”, “Yuk”);  
}

### Recursive Function

In C language, you can create a recursive function. A recursive function is a function that can call itself. Generally, a recursive function must have a base case. The call will terminate as the base case is met.

Example:  
int fact(int n){ //a recursive function to calculate a factorial value of n.  
 if(n==1) return 1;  
 return(n\*fact(n-1));  
}  
void main(){  
 int n;  
 printf("%s","Enter n:");  
 scanf("%d",&n);  
 printf("The factorial value of %d is %d",n,fact(n));  
return 0;  
}