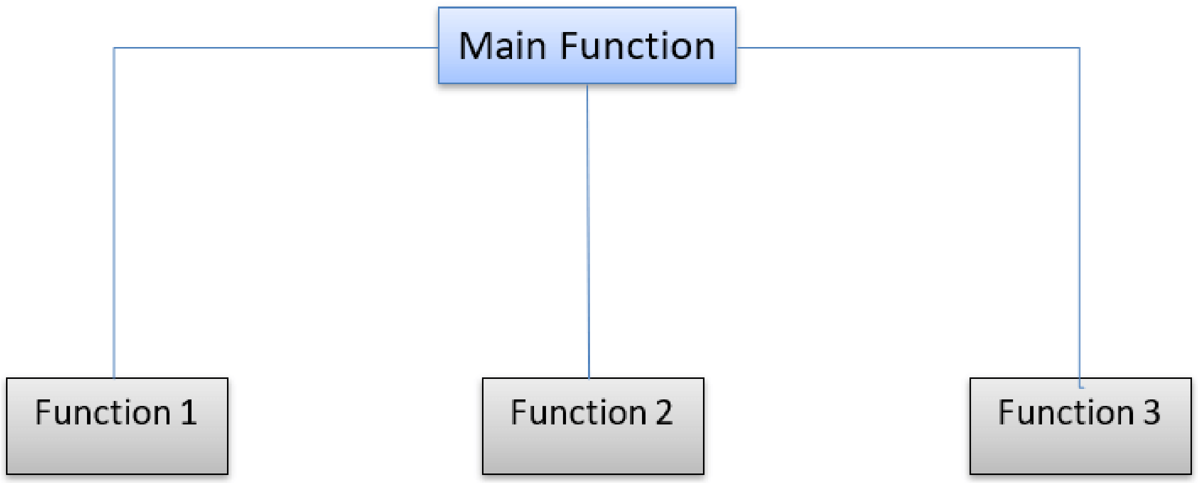
What is a Function in C?

Function in C programming is a reusable block of code that makes a program easier to understand, test and can be easily modified without changing the calling program.
Functions divide the code and modularize the program for better and effective results.In short, a larger program is divided into various subprograms which are called functions.

A function can be called multiple times to provide reusability and modularity to the C program. In other words, we can say that the collection of functions creates a program. The function is also known as *procedure or subroutine in* other programming languages.



When you divide a large program into various functions, it becomes easy to manage each function individually.
Whenever an error occurs in the program, you can easily investigate faulty functions and correct only those errors. You can easily call and use functions whenever they are required which automatically leads to saving time and space.

Advantages of Function

The advantages of using functions are:

- ☐ Avoid repetition of codes.
- ☐ Increases program readability.
- ☐ Divide a complex problem into simpler ones.

☐ Reduces chances of error.
☐ Modifying a program becomes easier by using a function.

Types of function

There are two types of function in C programming:

1.Standard Library Functions

C Standard library functions or simply C Library functions are inbuilt functions in C programming.

The prototype and data definitions of these functions are present in their respective header files. To use these functions we need to include the header file in our program. For example,

If you want to use the `printf()` function, the header file `<stdio.h>` should be included.

Advantages of Using C library functions

1. They work
2. The functions are optimized for performance
3. It saves considerable development time
4. The functions are portable

2.User-defined function

You can also create functions as per your need. Such functions created by the user are known as user-defined functions.

A function created by a user is known as a user defined function.

C programming functions are divided into three activities such as,

1. ***Function declaration***
2. ***Function definition***
3. ***Function call***

Function declaration means writing the name of a function.
In a function declaration, we just specify the name of a function that we are going to use in our program like a variable declaration.
A function declaration is also called "Function **prototype**."

The function declarations (called prototype) are usually done above the main () function and take the general form:

return_data_type function_name (data_type arguments);
int sum(int a , int b);

The **return_data_type**: is the data type of the value function returned back to the calling statement.
The **function_name**: is followed by parentheses
Arguments names with their data type declarations optionally are placed inside the parentheses.

Function Definition

Function definition means just writing the body of a function. A body of a function consists of statements which are going to perform a specific task. A function body consists of a single or a block of statements.

```
return_type function_name (argument list)
{
        function body;

}
```

Function call

A function call means calling a function whenever it is required in a program. Whenever we call a function, it performs an operation for which it was designed.

```
function_name (argument_list);
```

Type of User-defined Functions in C

There can be 4 different types of user-defined functions, they are:

1. ***Function with no arguments and no return value***
2. ***Function with no arguments and a return value***
3. ***Function with arguments and no return value***
4. ***Function with arguments and a return value***

1.***Function with no arguments and no return value***

```
//Function with no arguments and no return value
#include<stdio.h>
void area_of_rect(); // function declaration or prototype
int main()
{
	area_of_rect(); // function call
	return 0;
}
void area_of_rect() // function definition
{
	int l,b,area;
	printf("Enter the length and breadth\n");
	scanf("%d %d", &l, &b);
	area=l*b;
	printf("The area of the rectangle is: %d",area);
}
```

Output:

```
Enter the length and breadth
4
5
```

```
The area of the rectangle is: 20
--------------------------------
Process exited after 2.584 seconds with return value 0
Press any key to continue . . .
```

2.*Function with no arguments and a return value*

```c
//Function with no arguments and a return value
#include<stdio.h>
int area_of_rect(); // function declaration
int main()
{
	int result;
	result = area_of_rect(); // function call
	printf("The area of the rectangle is: %d", result);
	return 0;
}
int area_of_rect() // function definition
{
	int l,b,area;
	printf("Enter the length and breadth\n");
	scanf("%d %d", &l, &b);
	area=l*b;
// returning the result
	return area;
}
```

Output:

```
Enter the length and breadth
2
3
The area of the rectangle is: 6
--------------------------------
```

```
Process exited after 2.021 seconds with return value 0
Press any key to continue . . .
```

3.*Function with arguments and no return value*

```
//Function with arguments and no return value
#include<stdio.h>
void area_of_rect(int x, int y ); // function declaration
int main()
{
	int l, b;
	printf("Enter the length and breadth\n");
	scanf("%d %d", &l, &b);
	area_of_rect(l, b); // function call
	return 0;
}
void area_of_rect(int x, int y) // function definition
{
	int area;
	area=x*y;
	printf("The area of the rectangle is: %d",area);
}
```

Output:

```
Enter the length and breadth
5
6
The area of the rectangle is: 30
--------------------------------
Process exited after 1.738 seconds with return value 0
Press any key to continue . . .
```

4.*Function with arguments and a return value*

```
//Function with arguments and a return value
#include<stdio.h>
```

```
int area_of_rect(int x, int y ); // function declaration
int main()
{
	int l,b,result;
	printf("Enter the length and breadth\n");
	scanf("%d %d", &l, &b);
	result = area_of_rect(l, b); // function call
	printf("The area of the rectangle is: %d", result);
	return 0;
}
int area_of_rect(int x, int y) // function definition
{
	int area;
	area=x*y;
	return area;
}
```

Output:

```
Enter the length and breadth
5
9
The area of the rectangle is: 45
--------------------------------
Process exited after 2.751 seconds with return value 0
Press any key to continue . . .
```

Programs:

1.Wap to find the area of a triangle using user-defined functions

```
#include <stdio.h>
void area_of_triangle();
int main()
{
   area_of_triangle();
   return 0;
}
void area_of_triangle()
{
   float base,height,area;
   printf("Enter base ");
   scanf("%f", &base);
   printf("Enter height ");
   scanf("%f", &height);
   area=(base*height)/2;
   printf("Area of the triangle= %f sq. units", area);
}
```

Output:

```
Enter base 5
Enter height 5
Area of the triangle= 12.500000 sq. units
-------------------------------
```

2.Wap to find the area of a circle using user-defined functions

```
#include <stdio.h>
void area_of_circle();
int main()
{
       area_of_circle();
   return 0;
}
void area_of_circle()
{
       float r,area;
   printf("Enter radius ");
   scanf("%f", &r);
       area=3.14*r*r;
   printf("Area of the circle= %f sq. units", area);
}
```

```
//Another method
```

//Write a C program with a user-defined function to find the area of a circle using a user-defined function.

```
#include<stdio.h>

// function declaration
float circleArea(float r);

int main()
{
  float radius, area;

  printf("Enter the radius of the circle: ");
  scanf("%f", &radius);

  area = circleArea(radius); //function calling
  printf("Area of circle = %.2f\n",area);

  return 0;
}

// function definition
float circleArea(float r)
{
  float area;
  area= 3.14 * r * r;
  return area; // return statement
}
```

3.Wap to find the volume of the sphere using user-defined functions

```
#include <stdio.h>
```

```
void volume_of_sphere();
int main()
{
   volume_of_sphere();
    return 0;
}
void volume_of_sphere()
{
   float r,volume;
    printf("Enter radius ");
    scanf("%f", &r);
   volume=(float)4/3*3.14*r*r*r;
    printf("Volume of the sphere= %f", volume);
}
```

Output:

```
Enter radius 3
Volume of the sphere= 113.040001
```

4.Wap to check whether the given number is odd or even using user-defined functions.

```
#include <stdio.h>
void number();
int main()
{
    number();
    return 0;
}
void number()
{
          int n;
    printf("Enter  number ");
    scanf("%d", &n);
    if (n%2==0)
    printf("%d is even",n);
```

```
   else
   printf("%d is odd",n);
}
```

Output:

```
Enter  number 5
5 is odd
```

5.Wap to find the cube of a number using user-defined function

```
#include<stdio.h>
// function prototype, also called function declaration
float cube ( float x );
int main( )
{
   float m, n ;
   printf ( "\nEnter some number for finding cube \n");
   scanf ( "%f", &m ) ;
   n = cube ( m ) ;   //function call
   printf ( "\nSquare of the given number %f is %f",m,n );
}

float cube ( float x )   // function definition
{
   float p ;
   p = x * x * x ;
   return ( p ) ;
}
```

Output:

```
Enter some number for finding cube
3

Square of the given number 3.000000 is 27.000000
```

6.Wap to find the square of a number using user-defined functions

```
#include<stdio.h>
// function prototype, also called function declaration
float square ( float x );
int main( )
{
   float m, n ;
   printf ( "\nEnter some number for finding square \n");
   scanf ( "%f", &m ) ;
   n = square ( m ) ;  //function call
   printf ( "\nSquare of the given number %f is %f",m,n );
}

float square ( float x )   // function definition
{
   float p ;
   p = x * x ;
   return ( p ) ;
}
```

Output:

```
Enter some number for finding square
3

Square of the given number 3.000000 is 9.000000
```

7.Wap to find the greater among two numbers using user-defined functions.

```
#include <stdio.h>

/* function declaration */
int max(int num1, int num2);
int main ()
{
  /* local variable definition */
  int a,b,ret;
  printf("Enter the two numbers\n");
  scanf("%d %d",&a,&b);
  /* calling a function to get max value */
  ret = max(a, b);
  printf( "Max value is : %d\n", ret );
  return 0;
}

/* function returning the max between two numbers */
int max(int num1, int num2)
{
  /* local variable declaration */
  int result;
  if (num1 > num2)
    result = num1;
  else
    result = num2;
  return result;
}
```

Output:

```
Enter the two numbers
4
6
Max value is : 6
```

8. Wap to find the greatest among 3 numbers using user-defined functions.

```
#include <stdio.h>
```

```
void max();
int main ()
{
max();
return 0;
}
void max()
{
int a,b,c;
printf("Enter the 3 numbers");
scanf("%d %d %d",&a,&b,&c);
if (a>b&&a>c)
{
printf("The greatest number is %d",a);
}
else if(b>a && b>c)
{
printf("The greatest number is %d",b);
}
else if(c>a && c>b)
{
printf("The greatest number is %d",c);
}
else
{
printf("All numbers are equal");
}
}
```

Output:

```
Enter the 3 numbers4
5
6
The greatest number is 6
```

----

9. Wap to find the sum of N natural numbers

```
#include<stdio.h>
void sum();
int main()
{
        sum();
        return 0;
}
void sum()
{
   int n, result,sum=0;
   printf("Upto which number you want to find sum: ");
   scanf("%d", &n);
   for(int i=1; i<=n; i++)
   {
     sum += i;
   }
   printf("The sum of the given n natural numbers is %d",sum);
}
```

Output:

```
Upto which number you want to find sum: 10
The sum of the given n natural numbers is 55
```

10.wap to find the sum of N even natural numbers

```
#include<stdio.h>
void sum();
int main()
{
        sum();
        return 0;
}
void sum()
{
  int n, result,sum=0;
  printf("Upto which number you want to find sum: ");
  scanf("%d", &n);
  for(int i=2; i<=n; i+=2)
  {
    sum += i;
  }
  printf("The sum of the given n natural even numbers is %d",sum);
}
```

Output:

```
Upto which number you want to find sum: 10
The sum of the given n natural even numbers is 30
```

11.wap to find the sum of N odd natural numbers

```
#include<stdio.h>
void sum();
int main()
{
        sum();
        return 0;
}
void sum()
{
  int n, result,sum=0;
  printf("Upto which number you want to find sum: ");
  scanf("%d", &n);
  for(int i=1; i<=n; i+=2)
  {
    sum += i;
  }
  printf("The sum of the given n natural odd numbers is %d",sum);
}
```

Output:

```
Upto which number you want to find sum: 10
The sum of the given n natural odd numbers is 25
```

----

12. Wap to find the factorial of a number using user-defined functions

```
#include <stdio.h>
void fact();
int main()
{
        fact();
        return 0;
}
void fact()
{
   int i,n,f=1;
   printf("Enter a number to calculate its factorial\n");
   scanf("%d",&n);
   for(i=1;i<=n;i++)
   {
      f=f*i;
   }
   printf("Factorial of the num %d = %d\n",n,f);
}
```

Output:

```
Enter a number to calculate its factorial
5
Factorial of the num 5 = 120
```

13.Wap to find the factors of a number using user-defined functions.

```
#include<stdio.h>
void factors();
int main()
{
	factors();
	return 0;
}
void factors()
{
	int i,n;
	printf("Enter the number:\n");
	scanf("%d",&n);
	for(i=1;i<=n;i++)
	{
		if(n%i==0)
		printf("%d is a factor of %d\n",i,n);
	}
}
```

Output:

```
Enter the number:
15

1 is a factor of 15
3 is a factor of 15
5 is a factor of 15
15 is a factor of 15
```

14.wap to find the reverse of a number using udf.

```
#include<stdio.h>
void reverse();
int main()
{
	reverse();
	return 0;
}
void reverse()
{
	int n, rev=0,r,temp;
	printf("enter a number\n");
	scanf("%d",&n);
	temp=n;
	while(n!=0)
	{
		r=n%10;
		rev=rev*10+r;
		n=n/10;
	}
	printf("the reverse of %d is %d",temp,rev);
}
```

Output:

```
enter a number
123
the reverse of 123 is 321
--------------------------------
Process exited after 1.7 seconds with return value 0
Press any key to continue . . .
```

15.wap to check whether the given number is palindrome or not.

```
#include <stdio.h>
void check_palindrome();
int main()
```

```
{
check_palindrome();
return 0;
}
void check_palindrome()
{
        int n;
        int reverse_num=0, remainder,temp;
        printf("Enter an integer: ");
        scanf("%d", &n);
        temp=n;
        while(temp!=0)
        {
                remainder=temp%10;
                reverse_num=reverse_num*10+remainder;
                temp=temp/10;
        }
        if(reverse_num==n)
                printf("%d is a palindrome number",n);
        else
        printf("%d is not a palindrome number",n);
        }
```

Output:

```
Enter an integer: 1111
1111 is a palindrome number
--------------------------------
Process exited after 3.04 seconds with return value 0
Press any key to continue . . .
```

16.Wap to check whether the given number is armstrong or not for N digits.

```
#include<stdio.h>
#include<math.h>
void armstrong();
int main()
{
        armstrong();
        return 0;
}
void armstrong()
{
        int n, rev=0,r,temp,c;
        printf("enter the number of digits");
        scanf("%d",&c);
        printf("enter a number");
        scanf("%d",&n);
        temp=n;
        while(n!=0)
        {
                r=n%10;
                rev=pow(r,c)+rev;
                n=n/10;
        }
        if (rev == temp) {
                printf("%d is armstrong", temp);
        }
        else {
                printf("%d is not armstrong", temp);
        }
}
```

Output:

```
enter the number of digits4
enter a number1634
1634 is armstrong
```

```
Process exited after 3.394 seconds with return value 0
Press any key to continue . . .
```

17.Wap to check whether the given number is positive, negative or zero.

```
#include<stdio.h>
void pos_neg();
int main()
{
	pos_neg();
	return 0;
}
void pos_neg()
{
	int n;
	printf("enter the number:");
	scanf("%d",&n);
	if(n>0)
	{
		printf("the number %d is positive",n);
	}
	else if(n<0)
	{
		printf("the number %d is negative",n);
	}
	else
	{
		printf("the number %d is zero",n);
	}

}
```

Output:

```
enter the number:6
the number 6 is positive
--------------------------------
Process exited after 1.881 seconds with return value 0
Press any key to continue . . .
```

18.Wap to check whether the given number is prime or composite.

```
#include<stdio.h>

void check_primeorcomposite();
int main()
{
     check_primeorcomposite();
     return 0;
}
void check_primeorcomposite()
{
   int i,count=0;
   int num;
   printf("Enter a number to check whether it is    prime or not\n");
   scanf("%d",&num);
   for(i=1;i<=num;i++)
   {
      if(num%i==0)
      {
        count++;
      }
   }
   if(count==1)
   {
     printf("The given number is neither prime nor composite");
   }
  else if(count==2)
```

```
        {
    printf("The given number %d is Prime Number\n",num);
        }
        else
        {
    printf("The given number %d is not Prime Number\n",num);
     }
}
```

Output:

```
Enter a number to check whether it is    prime or not
5
The given number 5 is Prime Number
```

19.WAP to Display Fibonacci series in C within a range using a function

```
#include<stdio.h>
void fibonacciSeries();

int main()
{
  fibonacciSeries();
  return 0;
}
void fibonacciSeries()
{
  int a=0, b=1, temp,range;
  printf("Enter range upto which you want to display: ");
  scanf("%d", &range);
  printf("\nThe Fibonacci series is: \n");
  while (a<=range)
  {
    printf("%d\t", a);
    temp = a+b;
```

```
     a = b;
     b = temp;
   }
}
```

Output:

```
Enter range upto which you want to display: 23

The Fibonacci series is:
0      1      1      2      3      5      8      13     21
```

20.WAP TO Display fibonacci series upto Nth term using function.

```
#include<stdio.h>
void fibonacciSeries();
int main()
{
   printf("The fibonacci series is: \n");
   fibonacciSeries();
   return 0;
}

void fibonacciSeries()
{
   int a=0, b=1, c,i;
   int term;
   printf("Enter the term: ");
   scanf("%d", &term);
   for(i=0; i<term; i++)
   {
     printf("%d\t", a);
     c = a+b;
     a = b;
     b = c;
   }
}
```

Output:

```
The fibonacci series is:
Enter the term: 9
0     1     1     2     3     5     8     13     21
```

21.WAP TO Find nth term of Fibonacci series in C using function

```c
#include<stdio.h>
void fibonacciTerm();
int main()
{
  fibonacciTerm();
  return 0;
}
void fibonacciTerm()
{
  int a=0, b=1, c,i;
  int term;
  printf("Enter term to find: ");
  scanf("%d", &term);
  for(i=1; i<term; i++)
  {
    c = a+b;
    a = b;
    b = c;
  }
  printf("%d",a);
}
```

Output:

```
Enter term to find: 12
89
```

22.Write a C program to print the multiplication table of a number using user-defined functions.

```
#include<stdio.h>
void tables();
int main()
{
   tables();
   return 0;
}
void tables()
{
   int count,num;
   printf("Enter a positive number\n");
   scanf("%d", &num);
   printf("\nMultiplication Table for %d is:\n", num);
   for(count = 1; count <= 10; count++)
   {
      printf("%d x %d = %d\n", num, count, num*count);
   }
}
```

Output:

```
Enter a positive number
5

Multiplication Table for 5 is:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
```

```
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

--------------------------------
```