//Calling Function

#include<stdio.h>

/\* function declaration \*/int addition();

int main()

{

/\* local variable definition \*/ int answer;

/\* calling a function to get addition value \*/ answer = addition();

printf("The addition of the two numbers is: %d\n",answer);

return 0;

}

/\* function returning the addition of two numbers \*/int addition()

{

/\* local variable definition \*/ int num1 = 10, num2 = 5;

return num1+num2;

}

#### Example of function with no return no argument

/\*\*

\* C program to generate Fibonacci series using function

\*/

#include <stdio.h>

/\* Function declaration \*/

void generateFibo();

int main()

{

generateFibo();

return 0;

}

/\* Function definition \*/

void generateFibo()

{

int a, b, c, i, terms;

/\* Input a number from user \*/

printf("Enter number of terms: ");

scanf("%d", &terms);

a = 0;

b = 1;

c = 0;

printf("Fibonacci terms: \n");

// Iterate through n terms

for(i=1; i<=terms; i++)

{

printf("%d, ", c);

a = b; // Copy n-1 to n-2

b = c; // Copy current to n-1

c = a + b; // New term

}

}

#### Example program of function with no return but with arguments

/\*\*

\* C program to print natural numbers using functions

\*/

#include <stdio.h>

/\* Function declaration \*/

void printNaturalNumbers(int start, int end);

int main()

{

int s, e;

printf("Enter lower range to print natural numbers: ");

scanf("%d", &s);

printf("Enter upper limit to print natural numbers: ");

scanf("%d", &e);

printNaturalNumbers(s, e);

return 0;

}

/\* Function definition \*/

void printNaturalNumbers(int start, int end)

{

printf("Natural numbers from %d to %d are: \n", start, end);

while(start <= end)

{

printf("%d, ", start);

start++;

}

}

#### Example program of function with return but no arguments

/\*\*

\* C program to print random prime numbers

\*/

#include <stdio.h>

#include <stdlib.h> // Used for rand() function

/\* Function declaration \*/

int randPrime();

int main()

{

int i;

printf("Random 5 prime numbers are: \n");

for(i=1; i<=5; i++)

{

printf("%d\n", randPrime());

}

return 0;

}

/\* Function definition \*/

int randPrime()

{

int i, n, isPrime;

isPrime = 0;

while(!isPrime)

{

n = rand(); // Generates a random number

/\* Prime checking logic \*/

isPrime = 1;

for(i=2; i<=n/2; i++)

{

if(n%i==0)

{

isPrime = 0;

break;

}

}

if(isPrime ==1)

{

return n;

}

}

}

#### Example program of function with return and arguments

/\*\*

\* C check even odd using function

\*/

#include <stdio.h>

/\* Function declaration \*/

int evenOdd(int num);

int main()

{

int num, isEven;

printf("Enter a number: ");

scanf("%d", &num);

/\* Function call \*/

isEven = evenOdd(num);

if(isEven == 0)

printf("The given number is EVEN.");

else

printf("The given number is ODD.");

return 0;

}

/\* Function definition \*/

int evenOdd(int num)

{

/\* Return 0 if num is even \*/

if(num % 2 == 0)

return 0;

else

return 1;

}

[1. Call by Value](https://www.w3schools.in/c-tutorial/function-arguments/#Call_by_Value)

[2. Call by Reference](https://www.w3schools.in/c-tutorial/function-arguments/#Call_by_Reference)

## Call by Value

Example:

#include<stdio.h>

/\* function declaration \*/int addition(int num1, int num2);

int main()

{

/\* local variable definition \*/ int answer;

int num1 = 10;

int num2 = 5;

/\* calling a function to get addition value \*/ answer = addition(num1,num2);

printf("The addition of two numbers is: %d\n",answer);

return 0;

}

/\* function returning the addition of two numbers \*/int addition(int a,int b)

{

return a + b;

}

Program Output:

The addition of two numbers is: 15

## Call by Reference

Example:

#include<stdio.h>

/\* function declaration \*/int addition(int \*num1, int \*num2);

int main()

{

/\* local variable definition \*/ int answer;

int num1 = 10;

int num2 = 5;

/\* calling a function to get addition value \*/ answer = addition(&num1,&num2);

printf("The addition of two numbers is: %d\n",answer);

return 0;

}

/\* function returning the addition of two numbers \*/int addition(int \*a,int \*b)

{

return \*a + \*b;

}

Program Output:

The addition of two numbers is: 15

**Example of recursion in C**

/\*\*

\* Recursive function to print n natural numbers in reverse

\*/

#include <stdio.h>

void print(int n);

int main()

{

print(5);

return 0;

}

void print(int n)

{

/\* Print the current value of n \*/

printf("%d ", n);

/\* Base condition to terminate recursion \*/

if(n <= 1)

{

/\* Return and make no more recursive call \*/

return;

}

/\* Call print() function recursively with n-1 \*/

print(n - 1);

}

The following image explains entire program flow control through recursive calls.

