

Programming
Simplified Search[Home](#) » [C programming](#) » [C programming examples](#) » [Fibonacci series in c](#)

Fibonacci series in c

india_hacks THE BIGGEST
HACKATHONWIN A TRIP TO
SAN
FRANCISCO

Title Sponsor

ALTIMETRIK

Co Sponsor



Platinum Sponsor



Fibonacci series in c programming: c program for Fibonacci series without and with [recursion](#). Using the code below you can print as many numbers of terms of series as desired. Numbers of Fibonacci sequence are known as Fibonacci numbers. First few numbers of series are 0, 1, 1, 2, 3, 5, 8 etc, Except first two terms in sequence every other term is the sum of two previous terms, For example $8 = 3 + 5$ (addition of 3, 5). This sequence has many applications in mathematics and Computer Science.

Fibonacci series in c using for loop

```
/* Fibonacci Series c Language */
#include<stdio.h>

int main()
{
    int n, first = 0, second = 1, next, c;

    printf("Enter the number of terms\n");
    scanf("%d",&n);

    printf("First %d terms of Fibonacci series are :-\n",n);

    for ( c = 0 ; c < n ; c++ )
    {
        if ( c <= 1 )
            next = c;
        else
        {
            next = first + second;
            first = second;
            second = next;
        }
        printf("%d\n",next);
    }

    return 0;
}
```

Output of program:

```
E:\programmingsimplified.com\c\fibonacci-series.exe
Enter the number of terms
5
First 5 terms of Fibonacci series are :-
0
1
1
2
3
Process returned 0 (0x0)   execution time : 5.828 s
Press any key to continue.
```

Fibonacci series program in c using recursion

```
#include<stdio.h>

int Fibonacci(int);

main()
{
    int n, i = 0, c;

    scanf("%d",&n);

    printf("Fibonacci series\n");

    for ( c = 1 ; c <= n ; c++ )
    {
        printf("%d\n", Fibonacci(i));
        i++;
    }

    return 0;
}

int Fibonacci(int n)
{
    if ( n == 0 )
        return 0;
    else if ( n == 1 )
        return 1;
    else
        return ( Fibonacci(n-1) + Fibonacci(n-2) );
}
```

Recursion method is less efficient as it involves function calls which uses stack, also there are chances of stack overflow if function is called frequently for calculating larger Fibonacci numbers.

