

Fibonacci Series Program in C Using Recursion

The Fibonacci sequence is a cornerstone concept in mathematics and computer science. It comprises a sequence of numbers where each number is a sum of its two immediate predecessors, usually starting from 0 and 1. This article examines the construction of a Fibonacci Series program in C using recursion and other alternative methods.

What is the Fibonacci Series in C?

In the realm of the C programming language, the Fibonacci series is a numerical sequence achieved through specific programming constructs. The series, conventionally, commences from 0 and 1, with subsequent numbers calculated as the sum of the prior two.

For instance, a Fibonacci series beginning with 0 and 1 appears as follows:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

How to Construct a Fibonacci Series in C Language?

There are several methodologies for constructing a Fibonacci series in C, each offering its unique programming constructs. These methods include:

- without recursion,
- with recursion, and
- using a for loop.

How Does a Program Compute Fibonacci Function?

To comprehend the mechanics of how a program calculates the Fibonacci series, it is first important to understand the fundamental principle behind the Fibonacci sequence. The Fibonacci sequence refers to a number series where every number is the sum of the two preceding numbers, starting from 0 and 1. Formally, the sequence can be defined by the relation:

$$F(n) = F(n-1) + F(n-2)$$

Where $F(n)$ is the n th term in the Fibonacci sequence.

Now, let's dissect how a program computes the Fibonacci function using both an iterative approach (for example, a 'for' loop) and a recursive approach.

Iterative Approach

Here's a step-by-step guide on how an iterative program calculates the Fibonacci series:

1. **Initialise Variables:** Two variables (t1 and t2) are initialised to 0 and 1, respectively, representing the first two numbers in the Fibonacci sequence.
2. **Input From User:** The program prompts users to input the number of terms they wish to calculate in the Fibonacci sequence. This value is stored in a variable n.
3. **Loop through the Sequence:** The program then enters a loop that will run n times.
4. **Calculate the Next Term:** Inside the loop, the next term is calculated as the sum of t1 and t2, and this sum is stored in a new variable (nextTerm).
5. **Update Variables:** The variables t1 and t2 are updated to be ready for the next iteration. t1 gets the current value of t2, and t2 gets the value of nextTerm.
6. **Print the Term:** The current term (t1) is printed, and the loop moves to the next iteration until it has run n times.

Recursive Approach

The recursive approach involves a function calling itself to calculate the Fibonacci sequence.

Here's how it works:

1. **Function Definition:** The Fibonacci function is defined to take an integer n as its argument. This argument represents the position in the Fibonacci sequence for which we want to find the number.
2. **Base Cases:** Inside the function, we define what should happen for $n = 0$ and $n = 1$ — our base cases. For $n = 0$, the function should return 0, and for $n = 1$, it should return 1.
3. **Recursive Call:** If n is neither 0 nor 1, the function calls itself twice: once with argument n-1 and once with n-2. The function then returns the sum of these two calls.
4. **Invoke the Recursive Function:** From our main program, we invoke our Fibonacci function with an argument corresponding to the position in the sequence for which we want to find the number. We repeat this for as many terms as we wish to calculate.
5. **Print the Result:** The result of each function call is then printed, giving us the Fibonacci series.

The iterative method directly implements the definition of the Fibonacci series in a straightforward manner. On the other hand, the recursive method leverages the nature of the Fibonacci series and the property of functions to call themselves to simplify the program's structure.

Calculating the Fibonacci Series in C

C programming language provides diverse ways to compute the Fibonacci series. These include:

- Fibonacci series in C Without Recursion
- Fibonacci series in C Using Recursion
- Fibonacci series in C Using For loop

Fibonacci Series in C Without Recursion

The Fibonacci series computation can be achieved without recursion—by employing a 'for' loop or while loop. This method, known as the iterative method, calculates the next number by summing up the previous two numbers.

Below is a basic Fibonacci series program in C without recursion, followed by an in-depth explanation:

```
#include <stdio.h>

int main()
int    0    1

printf "Enter the number of terms: "
scanf "%d"

printf "Fibonacci Series: "

for    1
printf "%d, "
return 0
```

This code begins by initialising two integer variables t1 and t2, to 0 and 1, respectively, representing the first two terms of the Fibonacci series. The user is then prompted to input the number of terms they wish to print.

The for loop runs n times, where n is the number of terms entered by the user. During each iteration, the code prints the current term, computes the next term by adding t1 and t2, and updates t1 and t2 to be the last two terms for the next round of calculation.

Fibonacci Series in C Using Recursion

Recursion in programming describes a process where a function calls itself within its code. The Fibonacci series in C can be computed using recursion, as shown below:

```
#include <stdio.h>

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

int main()
{
    int n;

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

    printf("Fibonacci Series: ");

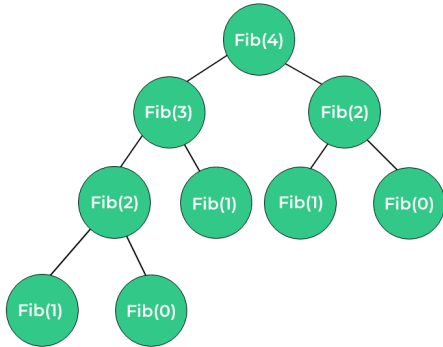
    for (int i = 0; i < n; i++)
        printf("%d ", fibonacci(i));

    return 0;
}
```

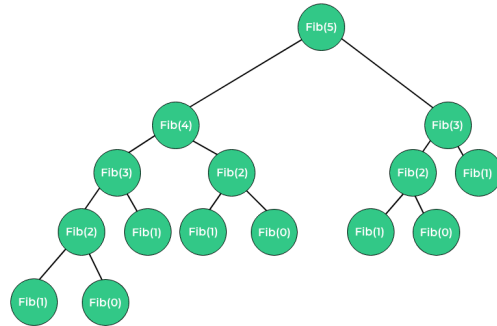
The function `fibonacci(int n)` is defined to compute the n th term of the Fibonacci sequence using recursion. If n equals 0 or 1, the function directly returns n . If n is greater than 1, the function calls itself with arguments $n-1$ and $n-2$ and returns the sum of these function calls, per the Fibonacci sequence's rule. Here's a graph that better illustrates how the different function calls are being made.

Recursion Graph for Fibonacci Series

Fib(4)



Fib(5)



Recursion Graph

The main function asks the user for the number of terms they want in the series. Then, a for loop runs n times to print the first n terms of the Fibonacci series. During each iteration, the fibonacci() function is called with the current term number to compute the corresponding Fibonacci number.

Program To Print Fibonacci Series In C Using Recursion

We can use recursion, a fundamental concept in various programming languages, to print the Fibonacci series. Recursion refers to the technique of making a function call itself directly or indirectly, making the solution to a problem easier to express.

A recursive function to display the Fibonacci series can be written as follows:

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

```
void printFibonacci(int n)
```

```
static int 0 1
```

```
if 0
```

```

}
printf "%d "
-1
}
}

int main()
int

printf "Enter the number of elements: "
scanf "%d"

printf "Fibonacci Series: "
printf "%d %d " 0 1 // printing 0 and 1
-2 // n-2 because 2 numbers are already printed

return 0
}

```

In this code, we use recursion to print the Fibonacci series up to n numbers. We first define a function `printFibonacci(int n)`, where n is the number of terms in the series.

In the function, we define three static variables, n1, n2, and n3, to hold the current number and the two preceding numbers in the series. n1 and n2 are initialised to the first two numbers in the series, 0 and 1.

If n is greater than 0, the function calculates n3 as the sum of n1 and n2, updates n1 and n2 for the next iteration, prints n3, and calls itself with n-1.

The main function asks users to input the number of terms they want in the series. It then prints the first two numbers and calls `printFibonacci(n-2)` to print the remaining numbers.

C Program To Print Fibonacci Series Using Iterative Method

We can also print the Fibonacci series using an iterative method, such as a 'for' loop. Here's how to do it:

```

#include <stdio.h>

void printFibonacci(int n)

```

```
int n1 = 0, n2 = 1;

printf("%d %d ", n1, n2) // printing 0 and 1

for (int i = 2; i < n; i++)
{
    printf("%d ", fib(i));
}

int main()
{
    int n;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Fibonacci Series: ");
    printFibonacci(n);

    return 0;
}
```

In this program, we define a function printFibonacci(int n), where n is the number of terms in the series. We initialise n1 and n2 to the first two numbers in the series and print them.

The function then enters a 'for' loop that runs n-2 times. In each iteration, it calculates n3 as the sum of n1 and n2, prints n3, and updates n1 and n2 for the next iteration.

The main function works the same as in the recursive version, asking the user to input the number of terms and then calling printFibonacci(n) to print the series.

Conclusion

The Fibonacci series, a fascinating concept in both mathematics and computer science, can be calculated using several methodologies in the C programming language. While recursion offers an elegant and intuitive solution, it may not be the most efficient method for larger series due to the function calls overhead. Conversely, an iterative method like a 'for' loop can execute more quickly, even though it may seem less straightforward.

Ultimately, the choice between recursion and iteration will depend on the specific requirements of your project, such as code readability, execution speed, and memory usage. Both methods, however, provide invaluable insight into how algorithms work and highlight the power of the C programming language.

To get a deeper understanding of these programming concepts and much more, consider enrolling in upGrad's [Master of Science in Computer Science](#) course. Leveraging in-demand skills, programming efficiencies and more, this course is designed to equip you with the knowledge and skills necessary for the ever-evolving tech world.

FAQs

1. What is a Fibonacci series?

The Fibonacci series is a sequence of numbers in which each number is the sum of the two preceding ones. It usually starts with 0 and 1.

2. How is the Fibonacci series calculated in C?

The Fibonacci series can be calculated in C using various methods, such as using recursion or an iterative method like a 'for' loop.

3. What is recursion in C?

Recursion in C is the process of a function calling itself from within its own code.

4. Why use recursion for the Fibonacci series in C?

Recursion provides a simple and clear way to calculate the Fibonacci series. However, for large inputs, the recursive method can be inefficient due to the overhead of function calls.
