Fibonacci Without Using Recursion

Asked 1 year, 2 months ago Modified 1 year ago Viewed 1k times

1

How I can print the (n)th element of the Fibonacci sequence, using and without using recursion, I've solved the first half of the question [revFibo() using recursion], but it seems that I can't see the problem with my iterative answer (Fibo() Function), it keeps printing the same parameter that I gave it, but throw a trash value when I put m = 3.

```c
#include <stdio.h>

int revfibo(int n);
int fibo(int m);

int main() {
    int n;
    printf("Give n: \n");
    scanf("%d",&n);
    printf("The Result is %d\n", fibo(n));
    return 0;
}

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

int fibo(int m){
    int T[m+1];
    int i;
    if (m == 0){
        return 0;
    }
    else {
        if (m == 1) {
            return 1;
        } else {
            T[0] = 0;
            T[1] = 1;
            for (i = 2; i <=m; i++) {
                T[i] = T[i - 2] + T[i - 1];
            }
            return T[m];
        }
    }
}
```

c algorithm recursion

Share Improve this question Follow

edited Jan 24, 2022 at 17:35

Jabberwocky
47k 17 59 111

asked Jan 24, 2022 at 16:51

oussama101
21 7

5 You really don't need any arrays. You only need to keep two last elements to derive the next one. – Eugene Sh. Jan 24, 2022 at 16:54

1 FWIW, your recursive solution is very suboptimal too. `revfibo(n-1)` and `revfibo(n-2)` are doing the exact same work up to `n-2` . – Eugene Sh. Jan 24, 2022 at 16:58

1 In addition to the first comment: Imagine `m` is 10. What is `T[m - 2] + T[m - 1]` on the very first iteration of the for loop? Are you sure you need to use `m` in the expression `T[m] = T[m - 2] + T[m - 1]` ? – Jabberwocky Jan 24, 2022 at 16:58

Please note that patterns like: `if (A) { ... } else { if (B) { ... } else { ... } }` can be expressed as `if (A) { ... } else if (B) { ... } else { ... }` – Chris Jan 24, 2022 at 17:20

Here is a non-recursive, non-iterative, Closed form: stackoverflow.com/questions/40966711/... – abelenky Jan 24, 2022 at 19:28

2 Answers

Sorted by: Highest score (default)

0

Your code is correct, in that it's calculating the sequence correctly, it's just returning the wrong part of it.

The commenters are correct that it's not very elegant code, but that's not really the point of your question.

it helps to rewrite your main function as follows so you can compare values and see how the sequence develops. As you'll see your two functions produce the same result:

```
int main() {
    int n;
    printf("Give n: \n");
    scanf("%d",&n);

    for (int i = 3; i < n; i++)
        {
        printf("%i recursive: %i, iterative: %i\n", i, revfibo(i), fibo(i));
        }

    return 0;
}
```

here's an interactive compiler online you can test it with the fully modified source.

https://onlinegdb.com/P_DDaqCpw

```
Give n:
10
3 recursive: 2, iterative: 2
4 recursive: 3, iterative: 3
5 recursive: 5, iterative: 5
6 recursive: 8, iterative: 8
7 recursive: 13, iterative: 13
8 recursive: 21, iterative: 21
9 recursive: 34, iterative: 34
```

The expected sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Share Improve this answer Follow edited Jan 24, 2022 at 19:30

answered Jan 24, 2022 at 19:25

Al Ro
410 2 11

This one is in javascript, but it meets what you are asking for:

```
const fibonacci = (num) => {
    if (num < 2) {
        return num;
    }
    let prev1 = 1, prev2 = 0;
    for (let i = 2; i < num ; i ++)
    {
        const tmp = prev1;
        prev1 = prev1 + prev2;
        prev2 = tmp;
    }
    return prev1 + prev2;
}

console.log(fibonacci(1)) // 1
console.log(fibonacci(2)) // 1
console.log(fibonacci(3)) // 2
console.log(fibonacci(4)) // 3
console.log(fibonacci(5)) // 5
console.log(fibonacci(6)) // 8
console.log(fibonacci(7)) // 13
console.log(fibonacci(8)) // 21
console.log(fibonacci(9)) // 34
```

Share Improve this answer Follow

answered Mar 16, 2022 at 20:48

JuanG
41 5