

بسم الله الرحمن الرحيم

University of Khartoum

Faculty of Mathematical Sciences and Informatics

Data Structures (C2033)

Lab 5 Assignment

(Recursion – Call Stack)

Content:

- ❖ Question 1: What is the call stack?
- ❖ Question 2: Call Stack of Fibonacci number of 5.

Written by:

Abobaker Ahmed Khidir Hassan

Computer Science

1: What is the call stack?

Each time a function is called, the system creates an activation record (also called an activation frame), that stores parameters and variables for the function and places the activation record in an area of memory known as a **call stack**.

When a function calls another function or itself recursively, the caller's activation record is kept intact, and a new activation record is created for the new function called.

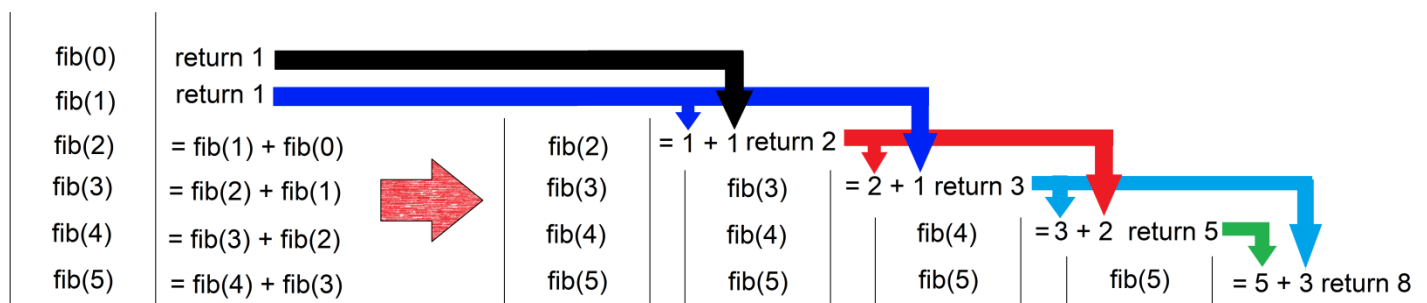
When a function finishes its work and returns to its caller, its activation record is removed from the call stack.

2: The Call Stack of Fibonacci number of n { $\text{fib}(n)$ } with $n = 5$

The code of the function that calculates Fibonacci number

```
int fib( int n )
{
    if( n < 0 ){
        Cout << "There is no Fibonacci number for negative values" << endl;
        return -1;
    } // end if
    else{
        if( n == 0 ) return 1;
        else if( n == 1 ) return 1;
        else { return ( fib( n - 1 ) + fib( n - 2 ) ) } //end en else
    } //end ex else
} // end fib
```

A Simple call stack of $\text{fib}(5)$



Real compiling steps to calculate Fibonacci of 5

fib(5):

(5 < 0) ? No.

(5 == 0) No. (5 == 1) No.

fib(5) = fib (4) + fib (3)

fib(4):

(4 < 0) ? No.

(4 == 0) No. (4 == 1) No

fib(4) = fib (3) + fib (2)

fib(3): // Calculate fib(3) for fib(4)

(3 < 0) ? No.

(3 == 0) No. (3 == 1) No.

fib(3) = fib (2) + fib (1)

fib(2): //Calculate fib(3)

(2 < 0) ? No.

(2 == 0) No. (2 == 1) No.

fib(2) = fib (1) + fib (0)

fib(1): // Calculate fib(1) for fib(2)

(1 < 0) ? No.

(1 == 0) No. (1 == 1) Yes

return 1.

fib(0): // Calculate fib(0) for fib(2)

(0 < 0) ? No.

(0 == 0) Yes.

return 1.

fib(2) = 1 + 1 = 2

return 2.

fib(1): //Calculate fib(1) for fib(3)

(1 < 0) ? No.

(1 == 0) No. (1 == 1) Yes

return 1.

fib(3) = 2 + 1 = 3

return 3.

fib(2): //Calculate fib(2) for fib(4)

(2 < 0) ? No.

(2 == 0) No. (2 == 1) No.

fib(2) = fib (1) + fib (0)

fib(1): //Calculate fib(1) for fib(2)

(1 < 0) ? No.

(1 == 0) No. (1 == 1) Yes.

return 1.

fib(0): //Calculate fib(0) for fib(2)

(0 < 0) ? No.

(0 == 0) Yes.

return 1.

fib(2) = 1 + 1 = 2

return 2.

fib(4) = 3 + 2 = 5

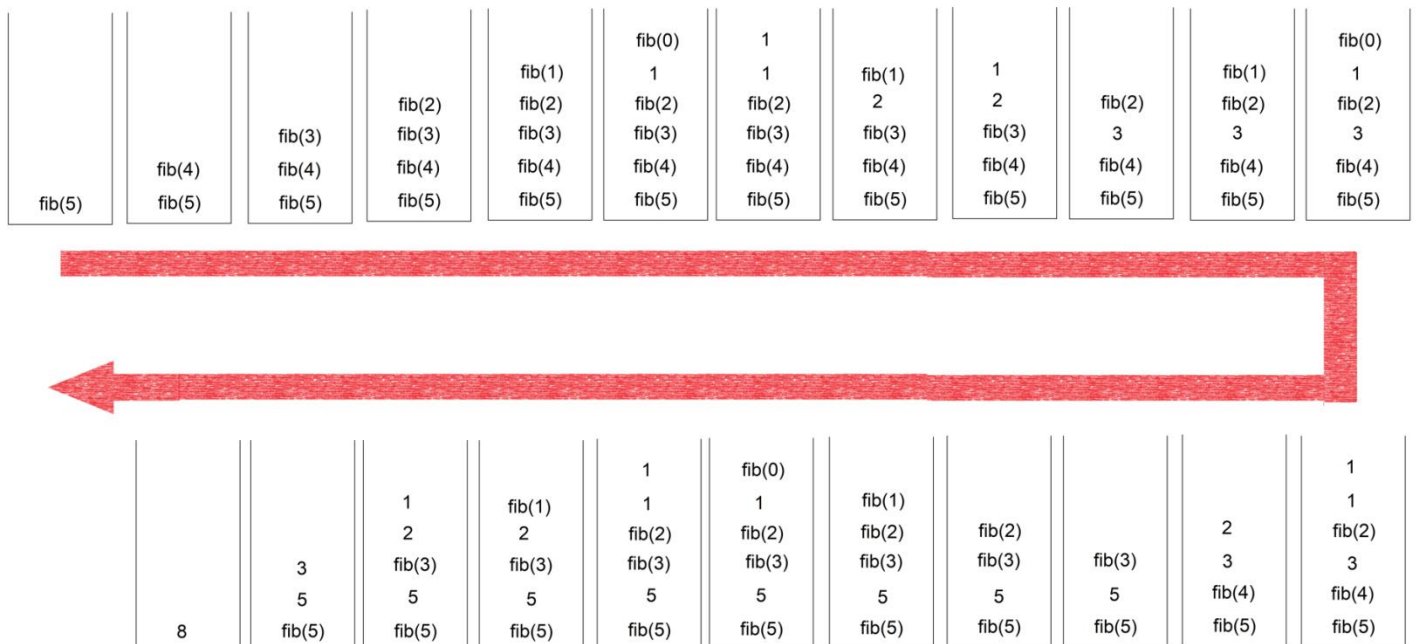
return 5.

```

fib(3): //Calculate fib(3) for fib(5)
    ( 3 < 0 ) ? No.
    ( 3 == 0 ) No. ( 3 == 1 ) No
    fib(3) = fib (2) + fib (1)
    fib(2): //Calculate fib(2) for fib(3)
        ( 2 < 0 ) ? No.
        ( 2 == 0 ) No. ( 2 == 1 ) No.
        fib(2) = fib (1) + fib (0)
        fib(1): //Calculate fib(1) for fib(2)
            ( 1 < 0 ) ? No.
            ( 1 == 0 ) No. ( 1 == 1 ) Yes
            return 1.
        fib(0): //Calculate fib(0) for fib(2)
            ( 0 < 0 ) ? No.
            ( 0 == 0 ) Yes.
            return 1.
        fib(2) = 1 + 1 = 2
    return 2.
    fib(1): //Calculate fib(0) for fib(3)
        ( 1 < 0 ) ? No.
        ( 1 == 0 ) No. ( 1 == 1 ) Yes
    return 1.
    fib(3) = 2 + 1 = 3
return 3.
fib(5) = 5 + 3 = 8
return 8.

```

Real Call Stack of Calculating Fibonacci Number of 5



Check that:

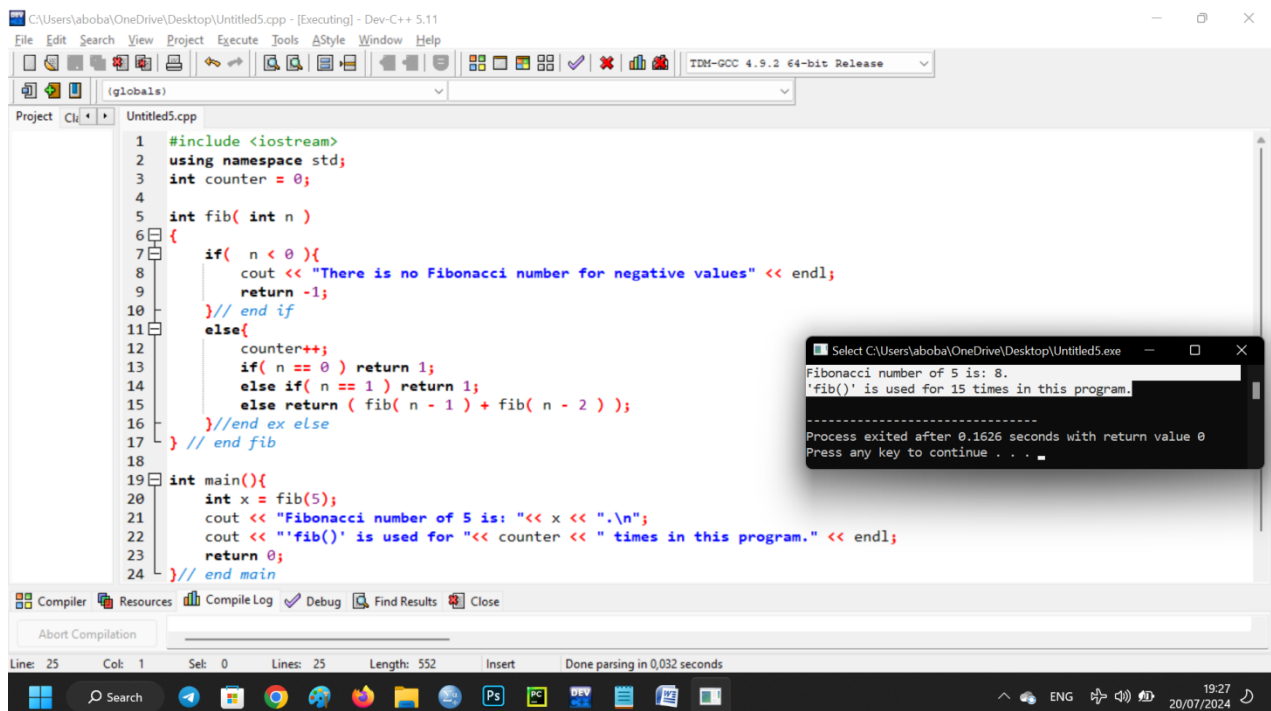
We can check that by adding a counter that calculates how many times the function is used as this code:

```
#include <iostream>
using namespace std;
int counter = 0;

int fib( int n )
{
    if( n < 0 ){
        cout << "There is no Fibonacci number for negative values" << endl;
        return -1;
    } // end if
    else{
        counter++;
        if( n == 0 ) return 1;
        else if( n == 1 ) return 1;
        else return ( fib( n - 1 ) + fib( n - 2 ) );
    } //end ex else
} // end fib

int main(){
    int x = fib(5);
    cout << "Fibonacci number of 5 is: "<< x << ".\n";
    cout << "'fib()' is used for "<< counter << " times in this program." << endl;
    return 0;
} // end main
```

Outputs:



```
C:\Users\aboba\OneDrive\Desktop\Untitled5.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools Style Window Help
(globals)
Project C:\Users\aboba\OneDrive\Desktop\Untitled5.cpp
1 #include <iostream>
2 using namespace std;
3 int counter = 0;
4
5 int fib( int n )
6 {
7     if( n < 0 ){
8         cout << "There is no Fibonacci number for negative values" << endl;
9         return -1;
10    } // end if
11    else{
12        counter++;
13        if( n == 0 ) return 1;
14        else if( n == 1 ) return 1;
15        else return ( fib( n - 1 ) + fib( n - 2 ) );
16    } //end ex else
17 } // end fib
18
19 int main(){
20     int x = fib(5);
21     cout << "Fibonacci number of 5 is: "<< x << ".\n";
22     cout << "'fib()' is used for "<< counter << " times in this program." << endl;
23     return 0;
24 } // end main

Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Line: 25 Col: 1 Sel: 0 Lines: 25 Length: 552 Insert Done parsing in 0,032 seconds
19:27 20/07/2024
```

Select C:\Users\aboba\OneDrive\Desktop\Untitled5.exe

Fibonacci number of 5 is: 8.
'fib()' is used for 15 times in this program.

.....
Process exited after 0.1626 seconds with return value 0
Press any key to continue . . .