Andrew Tran

Lab Experience 11

**Lab11Part1 - Machine Tracing**

#include <iostream>

using namespace std;

//-- PUT THE GLOBAL VARIABLE DECLARATIONS HERE

int indent = -2;

// PUT PROTOTYPE OF recFibonacci HERE

int f(int n);

int main()

{

int n;

cout << "\nPlease enter a positive integer: ";

cin >> n;

cout << "The function equates to " << f(n) << endl;

}

int f(int n)

{

indent += 2;

for (int i = 0; i < indent; i++)

{

cout << " ";

}

cout << "-> entering f(" << n << ")" << endl;

if (n < 2) //2

{

indent -= 2;

cout << "<- f(" << n << ") returns 0" << endl;

return 0; //3

}

else

{

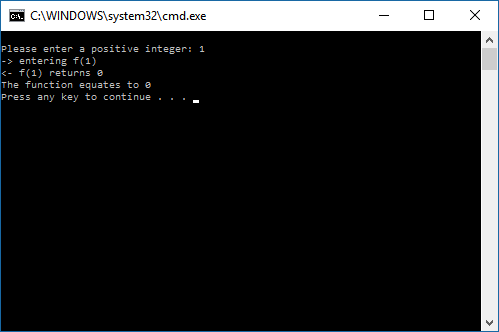
cout << "<- f(" << n << ") returns " << (1 + f(n / 2)) << endl;

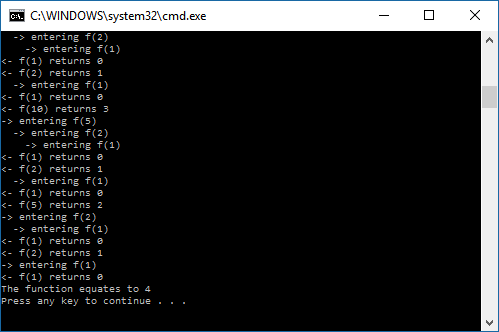
indent -= 2;

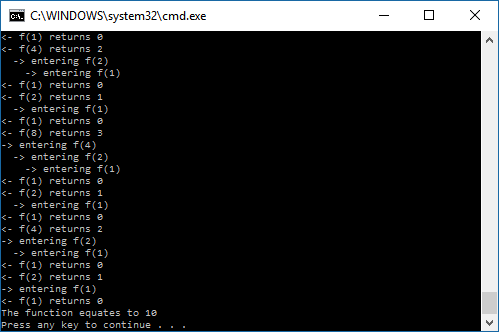
return 1 + f(n / 2);//4

}

}







**Lab11Part2 – Fibonacci**

#include <iostream>

using namespace std;

//-- PUT THE GLOBAL VARIABLE DECLARATIONS HERE

int numCalls = 0;

int indent = -2;

// PUT PROTOTYPE OF recFibonacci HERE

int recFibonacci(int number);

/\*-----------------------------------------------------------

recFibonacci is a recursive Fibonacci number calculator.

Precondition: The parameter nis a positive integer.

Postcondition: The n-th Fibonacci number is returned.

------------------------------------------------------------\*/

int main()

{

int n;

for (;;)

{

cout << "\nPlease enter a positive integer (or 0 to stop): ";

cin >> n;

if (n <= 0) break;

cout << "The " << n << "-th Fibonacci number is "

<< recFibonacci(n) << endl;

cout << "Number of function calls: " << numCalls << endl;

}

}

// DEFINE recFibonacci() HERE

int recFibonacci(int n) //1

{

numCalls++;

indent += 2;

for (int i = 0; i <= indent; i++)

{

cout << " ";

}

cout << "-> entering f(" << n << ")" << endl;

if (n == 0) //2

{

indent =- 2;

cout << "<- f(" << n << ") returns 0" << endl;

return 0;

}

if (n == 1) //3

{

indent =- 2;

cout << "<- f(" << n << ") returns 1" << endl;;

return 1;

}

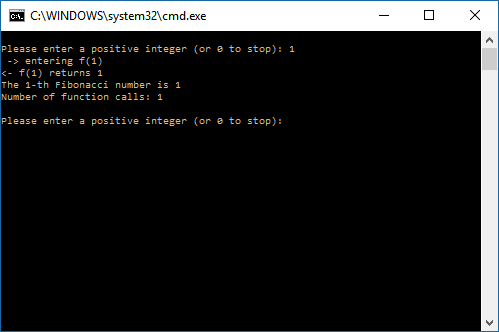
else

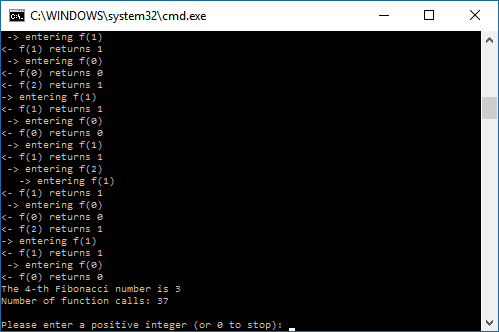
cout << "<- f(" << n << ") returns " << recFibonacci(n - 1) + recFibonacci(n - 2) << endl;

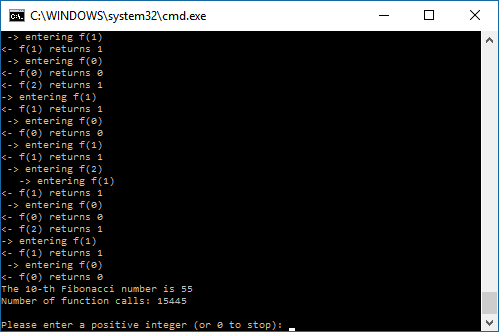
indent -= 2;

return recFibonacci(n - 1) + recFibonacci(n - 2); //4

}







**Lab11Part3 – GCD**

#include <iostream>

using namespace std;

int indent = -2;

int GCD(int a, int b);

int main()

{

int a=0, b=0;

cout << "Enter the first number for the GCD: " << endl;

cin >> a;

cout << "Enter the second number for the GCD: " << endl;

cin >> b;

cout << "The GCD of the two numbers is " << GCD(a, b) << endl;

}

int GCD(int a, int b)

{

indent += 2;

for (int i = 0; i < indent; i++)

{

cout << " ";

}

cout << "-> entering f(" << a << ", " << b << ")" << endl;

if (a < 0)

a = -a;

if (b < 0)

b = -b;

if (b == 0)

{

cout << "<- f(" << a << ", " << b << ") returns " << a << endl;

indent -= 2;

return a;

}

else

{

cout << "<- f(" << a << ", " << b << ") returns " << GCD(b, a % b) << endl;

indent -= 2;

return GCD(b, a % b);

}

}

