LAB 4

DATA STRUCTURES AND ALGORITHM

Hassaan Akbar Cheema

174351

TASK 1: Calculate the total time for the computation of the Fibonacci series for n by using both iterative and recursive methods.

ITERATIVE:

CODE:

// ConsoleApplication1.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <ctime>

using namespace std;

int main()

{

int n1 = 0, n2 = 1, n3, i, number;

cout << "Enter the number of elements: ";

cin >> number;

cout << n1 << " " << n2 << " "; //printing 0 and 1

clock\_t start = clock(); // start timer

for (i = 2; i<number; ++i) //loop starts from 2 because 0 and 1 are already printed

{

n3 = n1 + n2;

cout << n3 << " ";

n1 = n2;

n2 = n3;

}

clock\_t stop = clock();

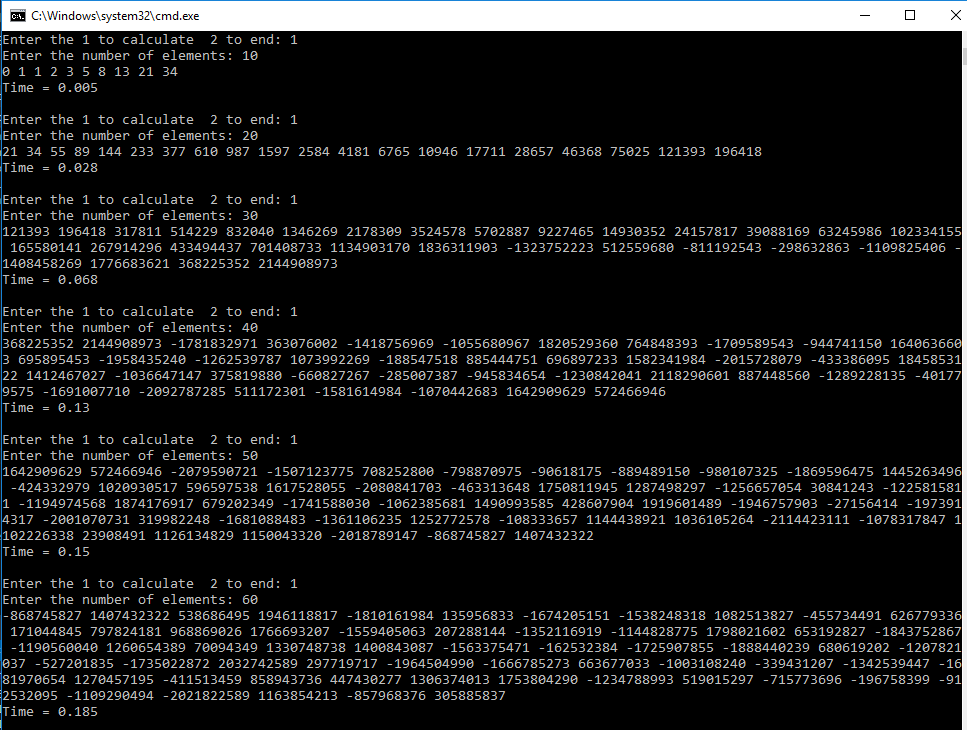
int cpu\_time\_used = float(stop - start) / CLOCKS\_PER\_SEC; //time in seconds

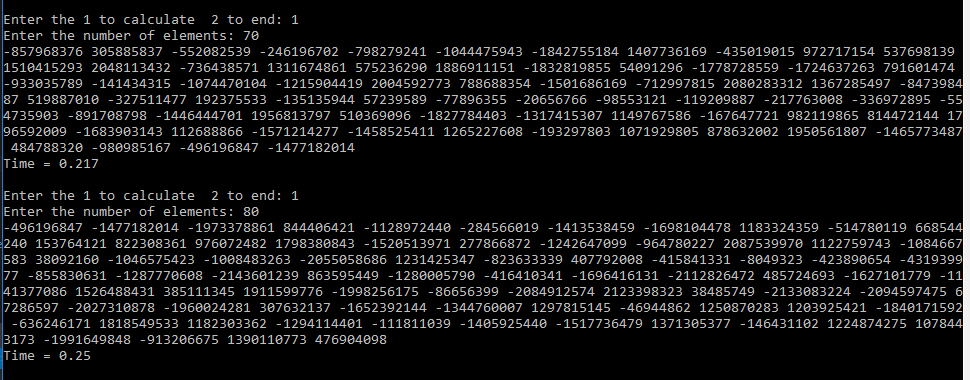
cout <<endl<< cpu\_time\_used <<endl;

return 0;

}

output:





Recursive:

CODE:

// ConsoleApplication1.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <ctime>

using namespace std;

int Fibonacci(int);

int main(void) {

int number;

cout << "Please enter a positive integer: ";

cin >> number;

if (number < 0)

cout << "That is not a positive integer.\n";

else{

clock\_t start = clock(); // start timer

cout << number << " Fibonacci is: " << Fibonacci(number) << endl;

clock\_t stop = clock(); // stop timer

int cpu\_time\_used = float(stop - start) / CLOCKS\_PER\_SEC; //time in seconds

cout << endl << cpu\_time\_used << endl;

}

}

int Fibonacci(int x)

{

if (x < 2)

{

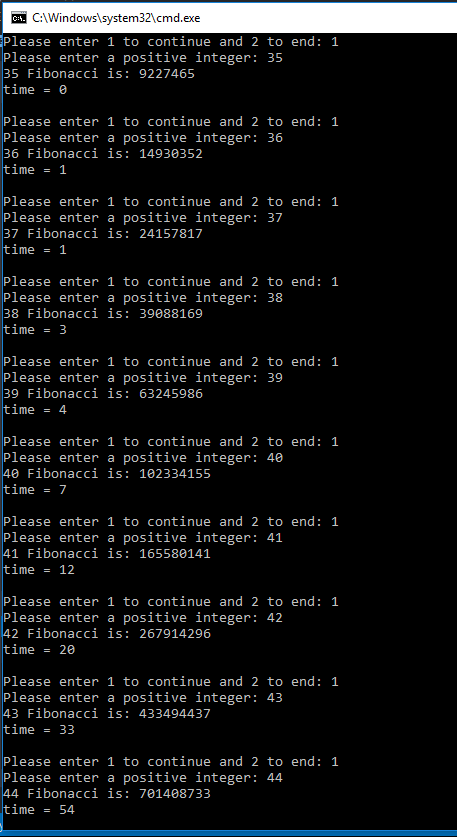
return x;

}

return (Fibonacci(x - 1) + Fibonacci(x - 2));

}

output:



TASK 1: Calculate the total time for the computation of the Factorial series for n by using both iterative and recursive methods.

ITERATIVE:

CODE:

#include "stdafx.h"

#include <iostream>

#include <stdlib.h>

#include <ctime>

using namespace std;

//Iterative Version

int factorial(int n)

{

int total = 1;

for (int i = 1; i <= n; i++)

{

total \*= i;

}

return total;

}

int main() {

int number = 0;

int choice = 0;

cout << "Enter 1 to continue and 2 to end : ";

cin >> choice;

while (choice == 1){

cout << "Enter number for factorial : ";

cin >> number;

clock\_t start = clock(); // start timer

factorial(number);

clock\_t stop = clock(); // stop timer

float cpu\_time\_used = float(stop - start) / CLOCKS\_PER\_SEC; //time in seconds

cout << "Time = " << cpu\_time\_used << endl;

cout <<endl<< "Enter 1 to continue and 2 to end : ";

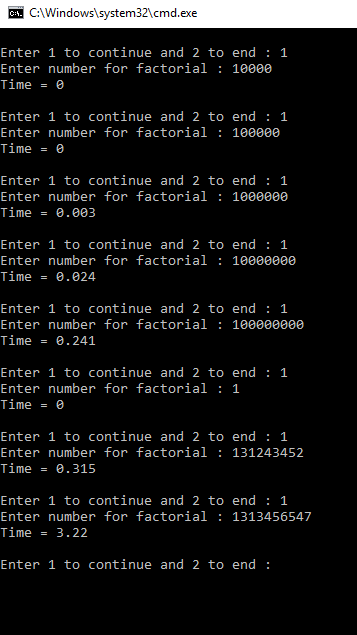
cin >> choice;

}

return 0;

}

output:



recursive:

CODE:

// Including header files.

#include <iostream>

#include <ctime>

using namespace std;

// Prototype

unsigned int factorial (int n);

// Main function

int main(int argc, char \*argv[]) {

int number;

cout << "Please enter a number to calculate factorial: " << endl;

cin >> number;

for (int i = 0; i < number; i++) {

clock\_t start = clock();

cout << factorial (i) << endl;

clock\_t stop = clock();

cout << "Time : "<< i << " recursive calculation is: " << (float(stop\_r - start\_r) / float(CLOCKS\_PER\_SEC)) << " sec" << endl;

}

return 0;

}

// Function definitions.

unsigned int factorial(int n) {

if (n == 0) {

return 1;

}

else if (n == 1) {

return 1;

}

else {

return n \* factorial (n - 1);

}

} // calculate factorial of n using recursion.

output:

