Universitatea Tehnică a Moldovei
Facultatea Calculatoare Informatică și Microelectronică
Departamentul Ingineria Software și Automatică

RAPORT

La lucrarea de laborator nr. 1

TEMA: „Analiza diferitor algoritmi, care rezolvă aceiași problemă șirul lui Fibonacci”

Disciplina: Analiza și proiectarea algoritmilor

A efectuat studentul: gr.SI-201 Ivanova Evghenia
A verificat: asistentul universitar Buldumac Oleg

Chișinău 2021

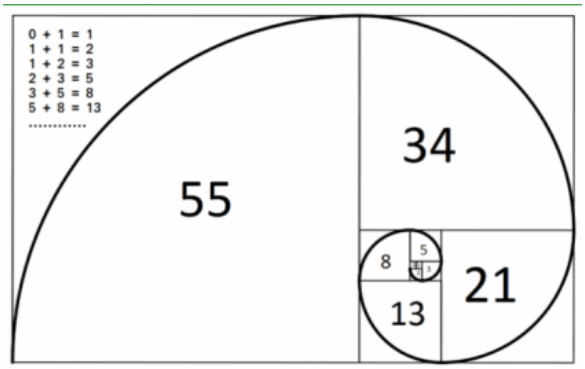
Scopul lucrării : Analiza diferitor algoritmi ce returnează al n-lea număr din șirul lui Fibonacci . Și anume prin 3 metode: iterativă, recursivă și prin formulă utilizând secvența de aur (golden raison). Eficiența acestora în dependență de numărul de iterații și timpul de execuție.

Sarcina : Executarea unui program care va afișa prin cele 3 metode elementul din șirul lui Fibonacci pentru diferite valori ale lui n. Elaborarea unui tabel cu datele de ieșire respectiv pentru cele 3 metode.

Considerații teoretice :

Italianul Leonardo of Pisa (cunoscut în matematică drept Fibonacci) a descoperit un șir de numere destul de interesant: **0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144…**

Primul lucru pe care îl observăm în acest șir este că dacă împărțim un element al șirului Fibonacci la precedentul său obținem: **1.61803.** Acest număr a fost denumit Φ (sau „phi" – în limba greacească) fiind considerat încă din antichitate raportul de aur.



Una din cele mai spectaculoase caracteristici a șirului lui Fibonacci și a numărului de aur este frecvența foarte mare cu care îl regăsim în natură : cochiliile melcilor, numărul petalelor florilor…

Termenul F_n este calculat prin următoarea relație de recurență: $F_n = F_{n-1} + F_{n-2}$ cu valorile inițiale $F_1=0$ **și** $F_2=1$

Analizând complexitatea algoritmilor metoda iterativă are complexitatea liniară O(n) aceasta fiind în dependență de valoarea lui n. Metoda recursivă are complexitatea exponențială $O(2^n)$. Și metoda ce calculează prin formulă are complexitatea O(n).

Codul programului :

```
#include <iostream>
#include <cmath>
#include <ctime>
using namespace std;
unsigned long long k = -1;
void fi(int n){
   unsigned long long a = 0, b = 1, c, k = 0;
```

```
    if(n == 1) c = 0;
    if(n == 2) c = 1;
    for(int i=3; i<=n; i++){
        k++;
        c = a + b;
        a = b;
        b = c;
    }
    cout<<c<<"\tNr de iteratii "<<k;;
}
unsigned long long fr(int n){
    k++;
    if(n == 1) return 0;
    if(n == 2) return 1;
    else return fr(n-1) + fr(n-2);
}
void ff(int n){
    cout<<(unsigned long long)round(pow((1+sqrt(5))/2,n-1)/sqrt(5));
}
int main (){
            int n;
a:      cout<<"\tIntroduceti raza: "; cin>>n;
        if(n<=0){ cout<<"\n !!! raza > 0"; goto a; }
        cout<<"\n M. Iterativa\t"; clock_t begini = clock(); fi(n); clock_t endi = clock();
        cout<<"\tTimpul de executie: "<<(double)(endi-begini)/CLOCKS_PER_SEC<<"s";
        cout<<"\n\n M. Recursiva\t"; clock_t beginr = clock(); cout<<fr(n); clock_t endr = clock();
cout<<"\tNr de iteratii "<<k<<"\tTimpul de executie: "<<(double)(endr-beginr)/CLOCKS_PER_SEC<<"s";
cout<<"\n\n Prin Formula\t"; clock_t beginf = clock(); ff(n); clock_t endf = clock();
cout<<"\tNr de iteratii -\tTimpul de executie: "<<(double)(endf-beginf)/CLOCKS_PER_SEC<<"s\n";
}
```

Execuția programului :

```
        Introduceti raza: 5

 M. Iterativa    3       Nr de iteratii 3        Timpul de executie: 1.8e-05s

 M. Recursiva    3       Nr de iteratii 8        Timpul de executie: 1e-06s

 Prin Formula    3       Nr de iteratii -        Timpul de executie: 2.2e-05s


...Program finished with exit code 0
Press ENTER to exit console.
```

```
        Introduceti raza: 25

 M. Iterativa    46368   Nr de iteratii 23       Timpul de executie: 1.3e-05s

 M. Recursiva    46368   Nr de iteratii 150048   Timpul de executie: 0.000666s

 Prin Formula    46368   Nr de iteratii -        Timpul de executie: 2.4e-05s


...Program finished with exit code 0
Press ENTER to exit console.
```

```
        Introduceti raza: 35

 M. Iterativa    5702887 Nr de iteratii 33       Timpul de executie: 1.8e-05s

 M. Recursiva    5702887 Nr de iteratii 18454928 Timpul de executie: 0.086743s

 Prin Formula    5702887 Nr de iteratii -        Timpul de executie: 2.9e-05s


...Program finished with exit code 0
Press ENTER to exit console.
```

```
        Introduceti raza: 49

 M. Iterativa    4807526976      Nr de iteratii 47       Timpul de executie: 0.000113s

 M. Recursiva    4807526976      Nr de iteratii 15557484096      Timpul de executie: 45.4465s

 Prin Formula    4807526976      Nr de iteratii -        Timpul de executie: 3.2e-05s


...Program finished with exit code 0
Press ENTER to exit console.
```

```
        Introduceti raza: 54

 M. Iterativa    53316291173     Nr de iteratii 52       Timpul de executie: 1.4e-05s

 M. Recursiva    53316291173     Nr de iteratii 172535142542     Timpul de executie: 506.207s

 Prin Formula    53316291173     Nr de iteratii -        Timpul de executie: 3.9e-05s


...Program finished with exit code 0
Press ENTER to exit console.
```

Tabelul cu datele de ieșire :

	5	25	35	49	62
M.Iterativă Iterații Timpul	**3** 3 1.8e-05	**46368** 23 1.3e-05	**5702887** 33 1.8e-05	**4807526976** 47 0.000113	**2504730781961** 60 1.6e-05
M.Recursivă Iterații Timpul	**3** 8 1e-06	**46368** 150048 0.000666	**5702887** 18454928 0.086743	**4807526976** 15557484096 45.4465	**2504730781961** 4052739537880 72776.333
Prin formula Iterații Timpul	**3** - 2.2e-05	**46368** - 2.4e-05	**5702887** - 2.9e-05	**4807526976** - 3.2e-05	**2504730781961** - 0.001

72 : 308061521170129 (m.iterativă) / 308061521170130 (prin formula)

94: 1220016045121876738 (m.iterativă) /12200160415121913856 (prin formula)

95: 19740274219868223167 (șirul lui Fibonacci) / 1293530146158671551 (m.iterativă) / 0 (prin formula)

Concluzia :

Pentru a efectua o analiză empirică nu este suficient un singur set de date de intrare ci mai multe, care să pună în evidenţă diferitele caracteristici ale algoritmului. În urma acestei lucrări de laborator am analizat 3 metode care afișează al n-lea număr din șirul lui Fibonacci. La final am ajuns la concluzia că metoda iterativă este mai eficientă pentru că are o execuție într-un timp mai scurt și într-un număr de iterații mai mic pentru numere mari în comparație cu metoda recursivă. Metoda ce calculează prin formulă evident este fără un număr de iterații și într-un timp și mai scurt în schimb aceasta dă greș începând cu termenul 72 , aceasta fiind din cauza rotungirilor care se fac în calculul aproximativ. Dar pentru termenul de pe poziția 95 acesta returnează valoare zero. Am utilizat tipul de date unsigned long long dar am observat că termenul de pe poziția 95 este mai mare decât limita maximă acestui tip de date , astfel începând cu acest număr ele sunt afișate incorect. Pentru a face o analiză complexă am utilizat 2 compilatoare și am observat că compilatorul online ne dă un timp minimal de execuție dar acesta nu se poate executa pentru termenii de pe poziția >= 55. Astfel afișând la ecran Killed.