

Șiruri de numere – Fibonacci

Pentru a implementa șirul Fibonacci în C++, ne vom folosi de trei variabile, a , b și c , cu următorul algoritm:

- La început, a și b sunt egale cu 1 (primele două elemente Fibonacci);
- La fiecare pas, c primește $a + b$ (devenind astfel noul termen Fibonacci generat)
- a ia valoarea lui b , iar b devine ultima sumă calculată.

$a = 1, b = 1, c = a + b(2)$

$a = 1, b = 2, c = a + b(3)$

$a = 2, b = 3, c = a + b(5)$

$a = 3, b = 5, c = a + b(8)$

$a = 5, b = 8, c = a + b(13)$

”[Limbajul C++](#) este un limbaj compilat. Etapele scrierii unui program în C++ sunt:

- editarea programului C++; se obține fișierul sursă, cu extensia **.cpp**
- compilarea fișierului sursă; aici se verifică corectitudinea sintactică a programului (corectitudinea cuvintelor folosite, prezența semnelor de punctuație etc.); dacă programul este corect sintactic, se va obține fișierul obiect, cu extensia **.o** sau **.obj**
- editarea de legături; se stabilesc legături între fișierul obiect curent și alte fișiere obiect, ale programatorului sau incluse în compilator; în urma acestei etape se obține programul executabil. În Windows, fișierele executabile au extensia **.exe**;
- programul executabil poate fi lansat în execuție (rulat).

În C++, identificatorii sunt grupați în spații de nume – *namespaces*. Există un spațiu de nume predefinit, cu numele *std*, din care fac parte toți identificatorii din biblioteca C++ standard; **cout**, ca și **endl**, este un identificator din spațiul de nume *std*.

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin >> n; //Numărul de elemente nenule de afișat
    int a = 1, b = 1, c;
    cout << "0 " << a << " " << b << " "; //Afișăm primii 3 termeni
    for(int i = 3; i <= n; i++) { //Afișăm restul de termeni, de la 3 la n
        c = a + b; //Conform algoritmului descris
        a = b;
        b = c;
        cout << c << " "; //Afișăm noul element generat
    }
    cout << endl << "In total sunt " << n+1 << " elemente afișate din șirul Fibonacci";
    return 0;
}
```