

## ȘIRURI NUMERICE, ȘIRURI DE CARACTERE

Scrie un program care să citească un număr natural nenul  $n$  și să afișeze, pe prima linie, separați printr-un spațiu, primii  $n$  termeni din șirul *primelor cuburi perfecte nenule*. Pe a doua linie să afișeze suma termenilor oglinziți ai șirului. Calculează valoarea fiecărui termen din șir numai în funcție de poziția sa în șir.

Exemplu:  $n=6$

- se vor afișa pe linia 1: 1 8 27 64 125 216 (numerele oglindite  $\rightarrow$  1 8 72 46 521 612);

- se vor afișa pe linia 2: 1260.

Explicație:  $1260=1+8+72+46+521+612$ .

```
#include <iostream>
using namespace std;
int main()
{
    int i, n, ci, ogl, suma=0;
    cin>>n;
    for(i=1; i<=n; i++)
    {
        cout<<i*i*i<<' ';
        ci=i*i*i;
        ogl=0;
        //urmeaza rasturnarea cifrelor numarului ci
        while(ci!=0)
        {
            /*Exemplu pentru 81 ogl=ogl*10+81%10, adica ogl=0*10+1, deci ogl=1, apoi ogl=1*10+8*/
            ogl=ogl*10+ci%10;
            ci=ci/10;
        }
        suma= suma+ogl;
        cout<<ogl<<' ';
    }
    cout<<endl<<' '<<suma;
    return 0;
}
```

Compiler: [https://www.onlinegdb.com/online\\_c++\\_compiler;](https://www.onlinegdb.com/online_c++_compiler;)

\*\*\*\*

<https://www.pbinfo.ro/articole/5537/Sirul-lui-fibonacci>

Numerele Fibonacci sunt numere naturale care fac parte din următorul șir, în care fiecare număr este egal cu suma celor două de dinainte:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

Uneori, șirul este extins cu încă un termen, la început:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

Termenul  $F_n$  este calculat prin următoarea relație de recurență:

$$F_n = F_{n-1} + F_{n-2}$$

cu valorile inițiale  $F_1=1$ ,  $F_2=1$  sau  $F_0=0$  și  $F_1=1$ .

## Algoritm

Cum determinăm primii  $N$  termeni din șirul lui Fibonacci? Vom folosi trei variabile simple  $a$ ,  $b$ ,  $c$ . Două dintre ele vor reprezenta termenii anteriori  $F_{n-1}$  și  $F_{n-2}$ , iar a treia va reprezenta termenul curent  $F_n$ :

```
a ← 1
b ← 1
scrie a, b
pentru i ← 3, n execută
    c ← a + b
    scrie c
    a ← b
    b ← c
sfarsit_pentru
```

Se dă un număr natural  $n$ . Să se afișeze în ordine crescătoare, primii  $n$  termeni ai șirului lui Fibonacci.

```
#include<iostream>
using namespace std;

int main()
{
    int fib1 = 0, fib2 = 1, fib3 = 1;

    cout << "Termenii sirului Fibonacci mai mici decat 1000 sunt : " << endl << fib1 << " " << fib2 << " ";
    while (fib1 + fib2 < 1000)
    {
        fib3 = fib1 + fib2;
        fib1 = fib2;
        fib2 = fib3;
        cout << fib3 << " ";
    }
    cout << endl;

    return 0;
}
```

## ȘIRURI DE CARACTERE

<https://www.online-cpp.com/>

```
/*
C++ Program care lipeste/concateneaza doua siruri de caractere- Folosim Strings si strcat()
*/
```

```
#include <iostream>
#include <string.h>
using namespace std;
int main ()
{
    char str1[100], str2[50];
    cout << "Scrie, la tastatura, sirul 1: ";
    gets(str1);
    cout << "Scrie, la tastatura, sirul 2: ";
```

26.02-1.03, clasele a VIII-a

```
    gets(str2);
    strcat(str1, str2);
    cout << "Concatenated String : " << str1;
    return 0;
}
După rulare:
```

```
Scrie, la tastatura, sirul 1:
gkjgkjgkj
Scrie, la tastatura, sirul 2:
bgkjbbbjkkk
Concatenated String : kjgkjgkjgkjbgkjbbbjkkk
** Process exited - Return Code: 0 **
/*
 * C++ Program - concatenarea a doua siruri de caractere cu append()
 */
```

## VARIANTA 2

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string str1, str2;
    cout << "Primul sir de caractere: ";
    getline(cin, str1);
    cout << "Al doilea sir de caractere: ";
    getline(cin, str2);
    str1.append(str2);
    cout << "Concatenated String: " << str1;
    return 0;
}
```

Sursa: <https://www.sanfoundry.com/cpp-program-illustrate-cstyle-string-functions/#method-1>