

## Laborator nr. 5

### Instrucțiuni repetitive în C++ (partea I)

**1) Descompunerea în factori primi ai unui număr a.** Se citește un număr întreg a. Să se realizeze un algoritm care să afișeze factorii primi și puterile lor pentru numărul citit.

Soluție:

```
#include <iostream>
using namespace std;
int a, divizor, putere;
int main()
{
    cin>>a;
    divizor = 2;
    while(a > 1)
    {
        putere = 1;
        while(a % divizor == 0)
        {
            putere = putere + 1;
            a = a / divizor;
        }
        if(putere != 0)
            cout<<divizor<<"^"<<putere<<", ";
        divizor = divizor + 1;
    }
    return 0;
}
```

De exemplu:

Dacă se citește pentru **a** valoarea 36 atunci algoritmul va afișa „**2<sup>2</sup>, 3<sup>2</sup>**”.

36		2
18		2
9		3
3		3
1		

Solutie implementata in compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:


```
1 #include <iostream>
2 using namespace std;
3 int a, divizor, putere;
4 int main()
5 {
6     cin>>a;
7     divizor = 2;
8     while(a > 1)
9     {
10        putere = 1;
11        while(a % divizor == 0)
12        {
13            putere = putere + 1;
14            a = a / divizor;
15        }
16        if(putere != 0)
17            cout<<divizor<<"^"<<putere<<", ";
18        divizor = divizor + 1;
19    }
20    return 0;
21 }
22
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0  Interactive Stdin Inputs

CommandLine Arguments

36

**Execute**  

Result

CPU Time: 0.00 sec(s), Memory: 3500 kilobyte(s)

2^3, 3^3,

**2) Cel mai mare divizor comun între 2 numere întregi a și b.** Se citesc două numere întregi a și b. Să se realizeze un algoritm care să afișeze `cmmdc(a,b)`.

Soluție:

```
#include <iostream>
using namespace std;
int a, b, rest;
int main()
{
    cin>>a>>b;
    rest = a % b; // rest reține restul împărțirii lui a la b
    while(rest != 0)
    {
        a = b;
        b = rest;
        rest = a % b;
    }
    cout<<"cmmdc = "<<b;
    return 0;
}
```

De exemplu:

Dacă se citește a = 36 și pentru b = 24 atunci algoritmul va afișa **cmmdc = 12**

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2 using namespace std;
3 int a, b, rest;
4 int main()
5 {
6     cin>>a>>b;
7     rest = a % b; // rest reține restul împărțirii lui a la b
8     while(rest != 0)
9     {
10         a = b;
11         b = rest;
12         rest = a % b;
13     }
14     cout<<"cmmdc = "<<b;
15     return 0;
16 }
17
18
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0  Interactive Stdin Inputs

CommandLine Arguments 36 24

Execute

Result

CPU Time: 0.00 sec(s), Memory: 3500 kilobyte(s)

```
cmmdc = 12
```

**3) Numărul invers.** Se citește un număr întreg a. Să se realizeze un algoritm care să afișeze numărul invers. Numim *număr invers* (sau **oglintit**) numărul format cu cifrele citite de la dreapta la stanga.

Soluție:

```
#include <iostream>
using namespace std;
int a, invers;
int main()
{
    cin>>a;
    invers = 0; // initial valoarea in variabila invers este zero
    while(a != 0)
    {
        invers = invers * 10 + a % 10;
        a = a / 10;
    }
    cout<<"Numarul oglindit = "<<invers;
    return 0;
}
```

De exemplu:

Dacă se citește  $a = 327$  atunci algoritmul va afișa **Numarul oglindit este = 723**

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:




```
1 #include <iostream>
2 using namespace std;
3 int a, invers;
4 int main()
5 {
6     cin>>a;
7     invers = 0; // initial valoarea in variabila invers este zero
8     while(a != 0)
9     {
10         invers = invers * 10 + a % 10;
11         a = a / 10;
12     }
13     cout<<"Numarul oglindit = "<<invers;
14     return 0;
15 }
16
17
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0  Interactive Stdin Inputs

CommandLine Arguments

327

**Execute**   

Result

CPU Time: 0.00 sec(s), Memory: 3448 kilobyte(s)

**Numarul oglindit = 723**

**4) Numărul palindrom.** Se citește un număr întreg a. Să se realizeze un algoritm care să verifice dacă numărul citit este sau nu palindrom. Numim **palindrom** *un număr care este egal cu oglinditul său.*

Soluție:

```
#include <iostream>
using namespace std;
int a, invers, aux;
int main()
{
    cin>>a;
    aux = a; // copiezi valoarea inițială a lui a în aux
    invers = 0; // inițial valoarea în variabila invers este zero
    while(aux != 0)
    {
        invers = invers * 10 + aux % 10;
        aux = aux / 10;
    }
    if(invers == a)
        cout<<"Numarul este PALINDROM";
    else
        cout<<"Numarul NU este PALINDROM";
    return 0;
}
```

De exemplu:

Dacă se citește pentru **a** valoarea 323 atunci algoritmul va afișa „**numarul este PALINDROM**”, iar dacă va citi 123 va afișa „**numarul NU este PALINDROM**”.

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2 using namespace std;
3 int a, invers, aux;
4 int main()
5 {
6     cin>>a;
7     aux = a; // copiez valoarea inițială a lui a în aux
8     invers = 0; // initial valoarea in variabila invers este zero
9     while(aux != 0)
10    {
11        invers = invers * 10 + aux % 10;
12        aux = aux / 10;
13    }
14    if(invers == a)
15        cout<<"Numarul este PALINDROM";
16    else
17        cout<<"Numarul NU este PALINDROM";
18    return 0;
19 }
20
21
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0  Interactive Stdin Inputs

CommandLine Arguments 323

**Execute**

Result  
CPU Time: 0.00 sec(s), Memory: 3496 kilobyte(s)

**Numarul este PALINDROM**

**5) Suma cifrelor unui numar.** Se citește un număr întreg n. Să se realizeze un algoritm care să calculeze suma cifrelor numărului dat.

Soluție:

```
#include <iostream>
using namespace std;
int n, suma, rest;
int main()
{
    cin>>n;
    suma = 0;
    while(n != 0)
    {
        rest = n % 10;
        suma = suma + rest;
        n = n / 10;
    }
    cout<<"Suma cifrelor = "<<suma;
    return 0;
}
```

De exemplu:

Dacă se citește  $n = 12345$  atunci algoritmul va afișa **Suma cifrelor = 15**

Iar dacă va citi  $n = 789$  va afișa **Suma cifrelor = 24**

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
1 #include <iostream>
2 using namespace std;
3 int n, suma, rest;
4 int main()
5 {
6     cin>>n;
7     suma = 0;
8     while(n != 0)
9     {
10        rest = n % 10;
11        suma = suma + rest;
12        n = n / 10;
13    }
14    cout<<"Suma cifrelor = "<<suma;
15    return 0;
16 }
17
18
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0  Interactive Stdin Inputs

CommandLine Arguments

789

**Execute**

Result

CPU Time: 0.00 sec(s), Memory: 3500 kilobyte(s)

Suma cifrelor = 24



**6) Aflarea primei cifre a numărului natural n.** Se citește un număr întreg n. Să se realizeze un algoritm care să afișeze prima cifră a numărului dat.

Soluție:

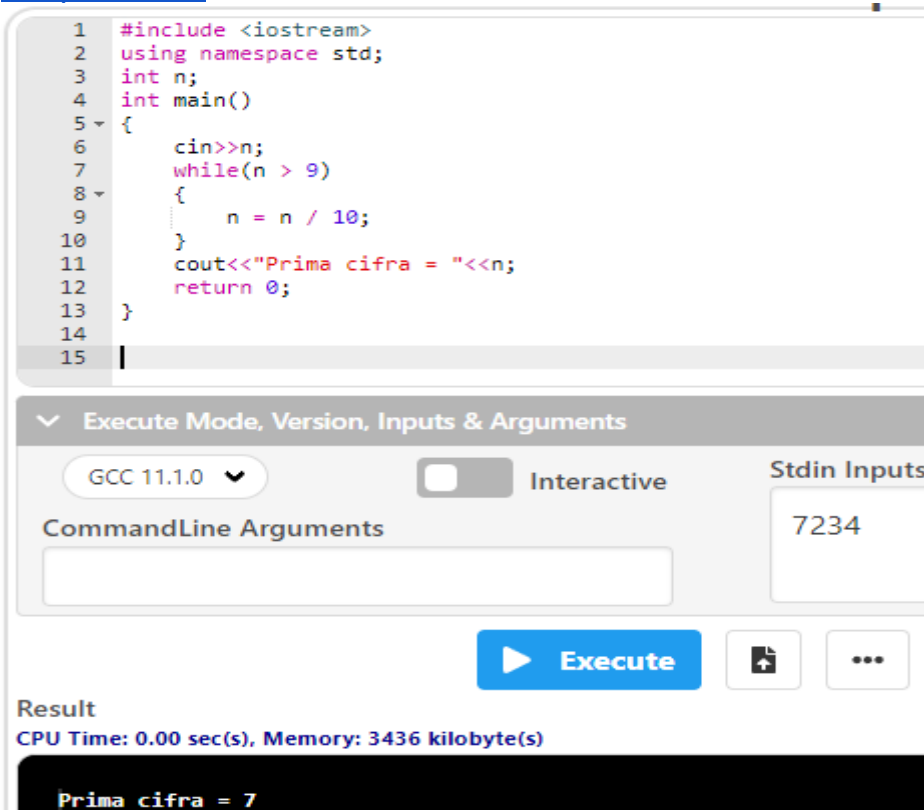
```
#include <iostream>
using namespace std;
int n;
int main()
{
    cin>>n;
    while(n > 9)
    {
        n = n / 10;
    }
    cout<<"Prima cifra = "<<n;
    return 0;
}
```

De exemplu:

Dacă se citește n = 7234 atunci algoritmul va afișa **Prima cifra = 7**

Iar dacă va citi n = 1789 va afișa **Prima cifra = 1**

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:



The screenshot shows an online C++ compiler interface. The code editor contains the following code:

```
1 #include <iostream>
2 using namespace std;
3 int n;
4 int main()
5 {
6     cin>>n;
7     while(n > 9)
8     {
9         n = n / 10;
10    }
11    cout<<"Prima cifra = "<<n;
12    return 0;
13 }
14
15
```

Below the code editor, there are settings for the compiler (GCC 11.1.0), an "Interactive" checkbox, and a "Stdin Inputs" field containing the value "7234". There is also a "CommandLine Arguments" field. A blue "Execute" button is visible.

The "Result" section shows the output: "Prima cifra = 7". Below the result, it displays "CPU Time: 0.00 sec(s), Memory: 3436 kilobyte(s)".

**7) Aflarea cifrei maxime dintr-un număr n.** Se citește un număr întreg n. Să se realizeze un algoritm care să afișeze cifra maximă a numărului dat.

Soluție:

```
#include <iostream>
using namespace std;
int n, suma, cifra_maxima, ultima_cifra;
int main()
{
    cin>>n;
    cifra_maxima = 0;
    suma = 0;
    while(n != 0)
    {
        ultima_cifra = n % 10;
        if(ultima_cifra > cifra_maxima)
            cifra_maxima = ultima_cifra;
        n = n / 10;
    }
    cout<<"Cifra maxima = "<<cifra_maxima;
    return 0;
}
```

De exemplu:

Dacă se citește n = 7284 atunci algoritmul va afișa **Cifra maxima = 8**,

Iar dacă va citi n = 9789 va afișa **Cifra maxima = 9**

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:



The screenshot shows an online C++ compiler interface. The code editor contains the following code:

```
1 #include <iostream>
2 using namespace std;
3 int n, suma, cifra_maxima, ultima_cifra;
4 int main()
5 {
6     cin>>n;
7     cifra_maxima = 0;
8     suma = 0;
9     while(n != 0)
10    {
11        ultima_cifra = n % 10;
12        if(ultima_cifra > cifra_maxima)
13            cifra_maxima = ultima_cifra;
14        n = n / 10;
15    }
16    cout<<"Cifra maxima = "<<cifra_maxima;
17    return 0;
18 }
19
20
```

Below the code editor, there is a section for execution settings. The compiler is set to GCC 11.1.0. The 'Interactive' checkbox is unchecked. The 'Stdin Inputs' field contains the value 7584. The 'CommandLine Arguments' field is empty. There is an 'Execute' button with a play icon.

The 'Result' section shows the output: **Cifra maxima = 8**. Below the result, it displays performance metrics: CPU Time: 0.00 sec(s), Memory: 3504 kilobyte(s).

**8) Aflarea cifrei minime dintr-un număr n.** Se citește un număr întreg n. Să se realizeze un algoritm care să afișeze cifra minimă a numărului dat.

Soluție:

```
#include <iostream>
using namespace std;
int n, suma, cifra_minima, ultima_cifra;
int main()
{
    cin>>n;
    cifra_minima = 10;
    suma = 0;
    while(n != 0)
    {
        ultima_cifra = n % 10;
        if(ultima_cifra < cifra_minima)
            cifra_minima = ultima_cifra;
        n = n / 10;
    }
    cout<<"Cifra minima = "<<cifra_minima;
    return 0;
}
```

De exemplu:

Dacă se citește n = 7284 atunci algoritmul va afișa **Cifra minima = 2**

Iar dacă va citi n = 9781 va afișa **Cifra minima = 1**

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:



The screenshot shows an online C++ compiler interface. The code editor contains the following code:

```
1 #include <iostream>
2 using namespace std;
3 int n, suma, cifra_minima, ultima_cifra;
4 int main()
5 {
6     cin>>n;
7     cifra_minima = 10;
8     suma = 0;
9     while(n != 0)
10    {
11        ultima_cifra = n % 10;
12        if(ultima_cifra < cifra_minima)
13            cifra_minima = ultima_cifra;
14        n = n / 10;
15    }
16    cout<<"Cifra minima = "<<cifra_minima;
17    return 0;
18 }
19
20
```

Below the code editor, the "Execute Mode, Version, Inputs & Arguments" section is visible. The compiler version is set to "GCC 11.1.0". The "Stdin Inputs" field contains the value "9781". The "Execute" button is highlighted in blue.

The "Result" section shows the output: "Cifra minima = 1". The CPU Time is 0.00 sec(s) and Memory is 3448 kilobyte(s).

**9) Numărul de cifre pare dintr-un număr n.** Se citește un număr întreg n. Să se realizeze un algoritm care să afișeze câte cifre pare are numărul dat.

Soluție:

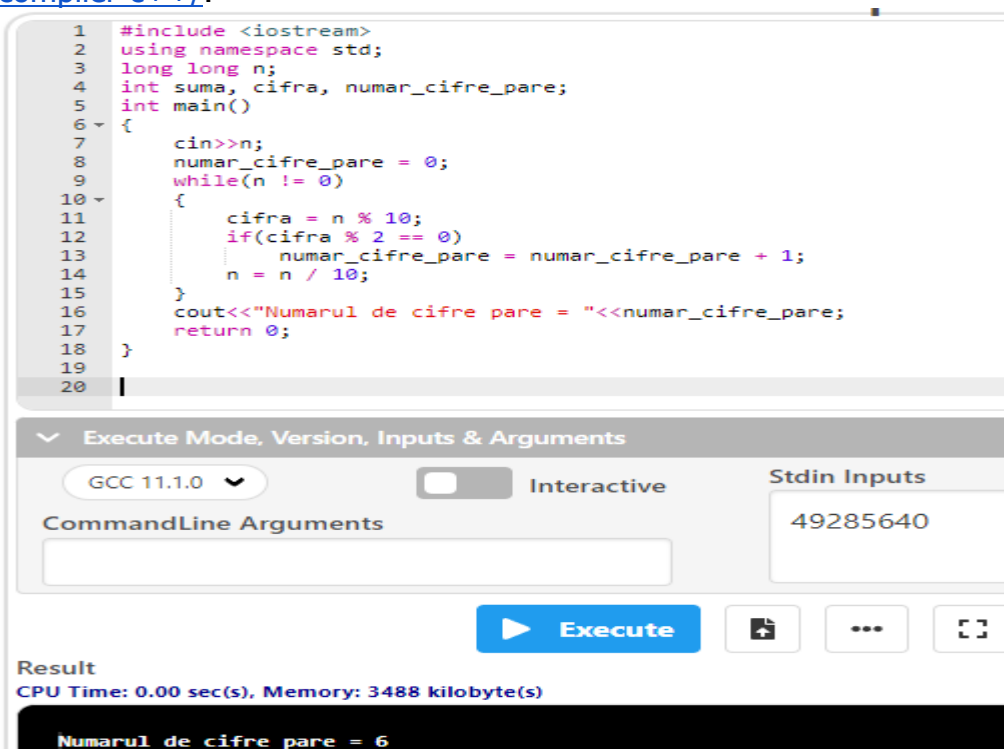
```
#include <iostream>
using namespace std;
long long n;
int suma, cifra, numar_cifre_pare;
int main()
{
    cin>>n;
    numar_cifre_pare = 0;
    while(n != 0){
        cifra = n % 10;
        if(cifra % 2 == 0)
            numar_cifre_pare = numar_cifre_pare + 1;
        n = n / 10;
    }
    cout<<"Numarul de cifre pare = "<<numar_cifre_pare;
    return 0;
}
```

De exemplu:

Dacă  $n = 5428$ , atunci se afișează **'Numarul de cifre pare = 3'** (4, 2 și 8)

iar dacă  $n = 49285640$ , atunci se afișează **'Numarul de cifre pare = 6'** (4, 2, 8, 6, 4, 0)

Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:



The screenshot shows an online C++ compiler interface. The code editor contains the following code:

```
1 #include <iostream>
2 using namespace std;
3 long long n;
4 int suma, cifra, numar_cifre_pare;
5 int main()
6 {
7     cin>>n;
8     numar_cifre_pare = 0;
9     while(n != 0)
10    {
11        cifra = n % 10;
12        if(cifra % 2 == 0)
13            numar_cifre_pare = numar_cifre_pare + 1;
14        n = n / 10;
15    }
16    cout<<"Numarul de cifre pare = "<<numar_cifre_pare;
17    return 0;
18 }
19
20
```

Below the code editor, the 'Execute Mode, Version, Inputs & Arguments' section is visible. The compiler version is set to 'GCC 11.1.0'. The 'Stdin Inputs' field contains the value '49285640'. A blue 'Execute' button is present.

The 'Result' section shows the output: 'CPU Time: 0.00 sec(s), Memory: 3488 kilobyte(s)' and a black box containing the text 'Numarul de cifre pare = 6'.

**10) Numar prim.** Sa se verifice daca un numar este numar *prim* sau nu. Spunem ca un numar  $n$  este numar prim daca are ca divizori numai valorile 1 si  $n$ .

Solutie:

```
#include <iostream>
using namespace std;
int n,i,prim;
int main(void)
{
    cin>>n;
    prim = 1;
    for(i = 2; i <= n / 2; i++)
        if(n % i == 0) prim = 0;
    if(prim == 1)
        cout<<"Numarul "<<n<<" este numar PRIM";
    else
        cout<<"Numarul "<<n<<" NU este numar PRIM";
    return 0;
}
```

De exemplu:

Dacă se citește  $n = 13$  atunci algoritmul va afișa **Numarul 13 este numar PRIM**

Iar dacă va citi  $n = 22$  va afișa **Numarul 22 NU este numar PRIM**

Solutie implementata in compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

The screenshot shows an online C++ compiler interface. The code editor contains the following code:

```
1 #include <iostream>
2 using namespace std;
3 int n,i,prim;
4 int main(void)
5 {
6     cin>>n;
7     prim = 1;
8     for(i = 2; i <= n / 2; i++)
9         if(n % i == 0) prim = 0;
10    if(prim == 1) cout<<"Numarul "<<n<<" este numar PRIM";
11    else cout<<"Numarul "<<n<<" NU este numar PRIM";
12    return 0;
13 }
14
```

The compiler settings show GCC 9.1.0. The command line arguments are empty. The stdin inputs are 13. The result of the execution is displayed in a black box:

```
Numarul 13 este numar PRIM
```

## Probleme propuse spre rezolvare:

**L5.1)** *Aflarea ultimei cifre a numărului natural n.* Se citește un număr întreg n. Să se scrie un program C++ care să le afișeze ultima cifră a numărului dat.

Exemplu:

Pentru  $n = 12345$  se va afișa valoarea 5

Iar pentru  $n = 789$  se va afișa valoarea 9

**L5.2)** *Cifre distincte.* Se citește un număr întreg n. Să se scrie un program C++ care să spună dacă un număr are toate cifrele distincte.

Exemplu:

Dacă  $n = 545471236$ , atunci se va afișa mesajul **"Numarul NU are toate cifrele distincte"** (deoarece cifrele 5 și 4 apar de câte două ori în număr)

Dacă  $n = 4321$ , atunci se va afișa mesajul **"Numarul are toate cifrele distincte"** (deoarece nici o cifră a sa nu se repeat)

**L5.3)** *Ecuția de gradul 1.* Fie ecuația  $a \cdot x = b$  cu a și b numere reale. Să se scrie un program C++ care să calculeze x. Precizare: Ecuția poate avea multiple soluții, o soluție sau niciuna!

Exemplu:

Dacă avem valorile  $a = 2$  și  $b = 3$ , atunci  $x = 3/2 = 1.5$

Dacă avem valorile  $a = 0$  și  $b = 7$ , atunci x nu are valori

Dacă avem valorile  $a = 0$  și  $b = 0$ , atunci x poate lua orice valori

**L5.4)** *Cel mai mare divizor propriu al unui număr.* Se citește un număr n. Să se scrie un program C++ care să afișeze cel mai mare divizor propriu al lui n (strict mai mic decât n).

Exemplu:

Dacă  $n = 24$  cel mai mare divizor propriu este 12.

Dacă  $n = 7$  cel mai mare divizor propriu este 1.

Dacă  $n = 125$  cel mai mare divizor propriu este 25.

Dacă  $n = 175$  cel mai mare divizor propriu este 35.

**L5.5)** *Numere prime până la n.* Se citește un număr n. Să se scrie un program C++ care să afișeze toate numerele prime mai mici sau egale cu n.

Exemplu:

Dacă  $n = 11$  atunci se va afișa 2, 3, 5, 7, 11

Dacă  $n = 20$  atunci se va afișa 2, 3, 5, 7, 11, 13, 17, 19

**L5.6)** <https://www.pbinfo.ro/probleme/3166/vas1>

Într-un vas sunt x litri de apă ( $x > 0$ ). După fiecare t minute, x/i din cantitatea de apă rămasă se evaporă. Să se scrie un program C++ care să determine după câte minute vor rămâne în vas cel mult y litri de apă.

Exemplu:

Dacă se dau valorile 100 15 20 5, atunci se va afișa 120

Explicație

$x=100$ ,  $t=15$ ,  $y=20$ ,  $i=5$ . Sunt necesare  $T=120$  minute pentru ca în vas să rămână cel mult  $y=20$  litri de apă.

**L5.7)** <https://www.pbinfo.ro/probleme/1650/acelasinumar>

Se dă un număr întreg  $n$  și alte  $k$  numere întregi. Să se scrie un program C++ care să afle dacă, adunând toate cele  $k$  numere la  $n$  se obține o valoare egală cu valoarea inițială a lui  $n$ .

Exemplu:

Dacă se dau valorile  $n = 25$ ,  $k = 3$  și următoarele  $n$  valori: 3, 16, -9, 3, atunci se va afișa mesajul NU

Explicație

$25 + 16 - 9 + 3 = 35$ , număr diferit de cel inițial (25).

**L5.8)** <https://www.pbinfo.ro/probleme/2601/sumapatratecifre>

Să se scrie un program C++ care să afișeze suma patratelor cifrelor unui număr natural de trei cifre citit de la tastatură.

Exemplu:

Intrare

Dacă  $n = 123$  atunci se va afișa 14 ( $1^2+2^2+3^2 = 1 + 4 + 9 = 14$ )

**L5.9)** <https://www.pbinfo.ro/probleme/1681/power>

Să se scrie un program C++ care să calculeze  $a^b$ .

Exemplu:

Dacă  $a = 5$  și  $b = 4$  atunci se va afișa 625 ( $5^4 = 5 * 5 * 5 * 5 = 625$ )

## **Bibliografie**

[1] <http://www.pbinfo.ro> Descrierea site-ului: "*www.pbinfo.ro îți propune să rezolvi probleme de informatică, cu evaluator automat. Știi pe loc dacă soluția ta este corectă sau dacă trebuie să mai lucrezi la ea.*

*Problemele sunt grupate după programa de informatică pentru liceu. Dar nu trebuie să fii la liceu ca să rezolvi aceste probleme. Poți fi elev de gimnaziu, student, profesor sau pur și simplu pasionat de informatică. De fapt, trebuie doar să vrei!!"*

[2] <https://www.runceanu.ro/adrian>

[3] Adrian Runceanu – „*Programarea și utilizarea calculatoarelor*”, Editura Academica Brâncuși din Târgu-Jiu, 2003, ISBN 973-8436-44-3

[4] Adrian Runceanu, Mihaela Runceanu, „*Noțiuni de programare – limbajul C++*”, Editura Academica Brâncuși din Târgu-Jiu, 2012, ISBN 978-973-144-550-2

[5] Adrian Runceanu, Mihaela Runceanu - „*Algoritmi implementați în limbajul C++*. Volumul I – *Algoritmi elementari*”, Editura Academica Brâncuși din Târgu Jiu, 2021, ISBN 978-606-9614-06-8