

Laborator nr. 4

Instrucțiuni repetitive în C++

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 „2^2, 3^2”.

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 $\text{cmmdc}(a,b)$.

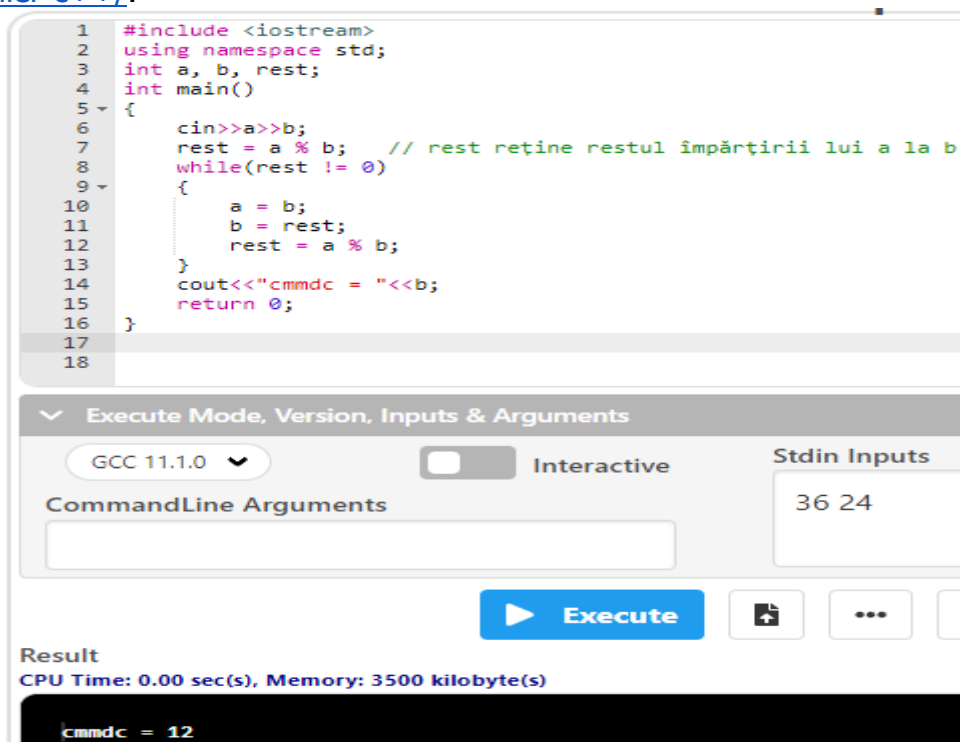
Soluție:

```
#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++/>:



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

```
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
```

Below the code editor, the "Execute Mode, Version, Inputs & Arguments" section shows the compiler version as "GCC 11.1.0" and the "Stdin Inputs" as "36 24". The "Execute" button is visible.

The "Result" section shows the output: "cmmdc = 12". Below the result, the CPU time is "0.00 sec(s)" and the memory usage is "3500 kilobyte(s)".

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

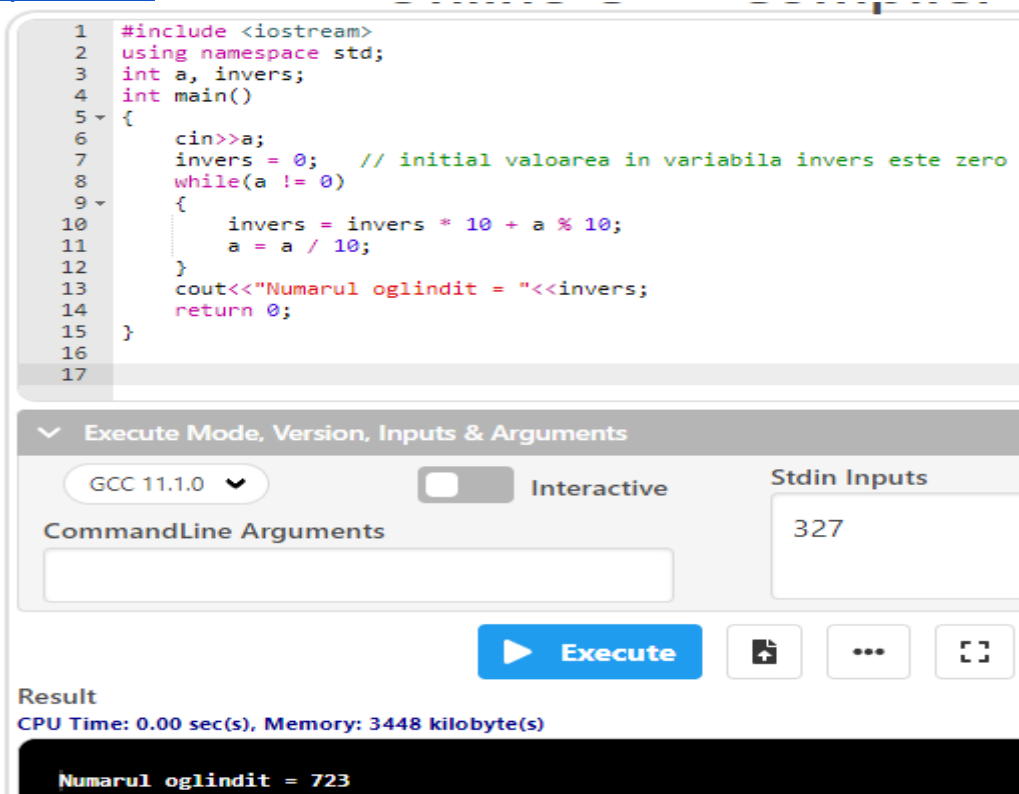
Solutie:

```
#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**

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 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
```

Below the code editor, the execution settings are shown: GCC 11.1.0, Interactive mode is disabled, and Stdin Inputs is set to 327. The CommandLine Arguments field is empty. A blue 'Execute' button is visible.

The Result section shows the output: CPU Time: 0.00 sec(s), Memory: 3448 kilobyte(s). The output text is: **Numarul oglindit = 723**

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

Solutie:

```
#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**

Solutie implementata in 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

5) 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:

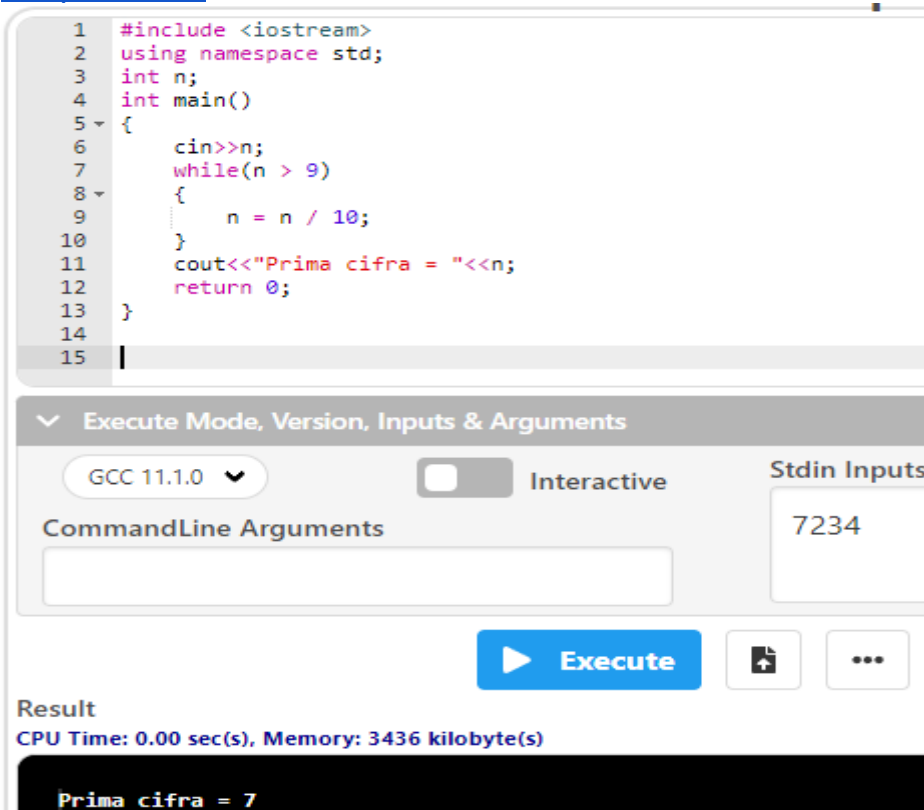
```
#include <iostream>
using namespace std;
int n;
int main()
{
    cin>>n;
    while(n > 9)
    {
        n = n / 10;
    }
    cout<<"Prima cifra = "<<n;
    return 0;
}
```

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), a checkbox for "Interactive" mode, and a text input field for "Stdin Inputs" containing the value "7234". There is also a "CommandLine Arguments" field which is currently empty. A blue "Execute" button is visible.

The "Result" section shows the output of the program: "Prima cifra = 7". The CPU time is 0.00 sec(s) and the memory usage is 3436 kilobyte(s).

6) 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 of the program: **Cifra maxima = 8**. Below the result, it displays performance metrics: CPU Time: 0.00 sec(s), Memory: 3504 kilobyte(s).

7) 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 same C++ code as shown in the previous block. Below the 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 number '9781'. An 'Execute' button is present. Below the execution controls, the 'Result' section shows the output: 'Cifra minima = 1'. The CPU time is 0.00 sec(s) and memory usage is 3448 kilobyte(s).

8) 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:

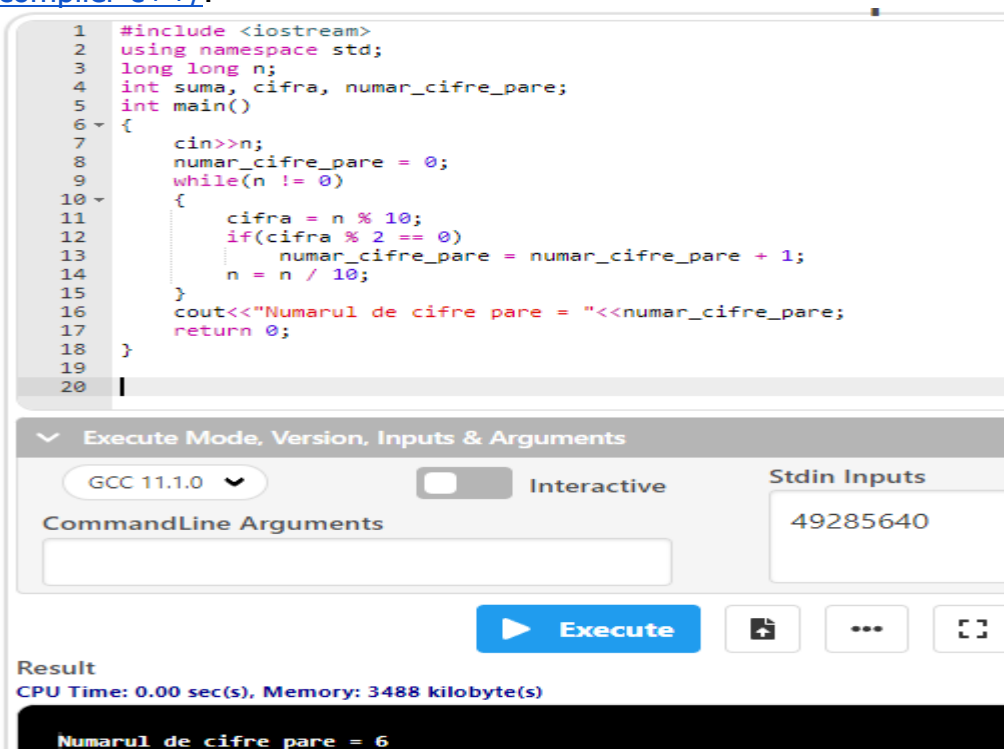
```
#include <iostream>
using namespace std;
long long n;
int suma, cifra, numar_cifre_pare;
int main()
{
    cin>>n;
    numar_cifre_pare = 0;
    while(n != 0){
        cifra = n % 10;
        if(cifra % 2 == 0)
            numar_cifre_pare = numar_cifre_pare + 1;
        n = n / 10;
    }
    cout<<"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 same C++ code as shown in the previous block. Below the 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 number '49285640'. A blue 'Execute' button is present. Below the execution controls, the 'Result' section shows the output: 'Numarul de cifre pare = 6'. The CPU time is 0.00 seconds and memory usage is 3488 kilobytes.

9) 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 same C++ code as shown in the previous block. Below the code editor, the 'Execute Mode, Version, Inputs & Arguments' section shows the compiler version as 'GCC 9.1.0'. The 'Stdin Inputs' field contains the value '13'. The 'Execute' button is highlighted in blue. Below the execution controls, the 'Result' section shows the output: 'Numarul 13 este numar PRIM'. The CPU time is 0.00 seconds and memory usage is 3372 kilobytes.

10) Sir generat. Să se scrie un program C++ care să afișeze primii n termeni ai șirului 1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4,

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

```
#include <iostream>
using namespace std;
int a,b,n,i;
int main()
{
    cin>>n;
    a=1;
    b=1;
    for(i = 1; i <= n; i++)
        if(a <= b + 1)
            {
                cout<<b<<" ";
                a++;
            }
        else
            {
                a=2;
                b++;
                cout<<b<<" ";
            }
    return 0;
}
```

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

```
1 #include <iostream>
2 using namespace std;
3 int a,b,n,i;
4 int main()
5 {
6     cin>>n;
7     a=1;
8     b=1;
9     for(i = 1; i <= n; i++)
10         if(a <= b + 1)
11             {
12                 cout<<b<<" ";
13                 a++;
14             }
15         else
16             {
17                 a=2;
18                 b++;
19                 cout<<b<<" ";
20             }
21     return 0;
22 }
23
```

Below the code editor, there is a control panel with the following settings:

- Compiler: GCC 11.1.0
- Interactive:
- Stdin Inputs: 15
- CommandLine Arguments: (empty)

The "Execute" button is highlighted in blue. Below the code editor, the "Result" section shows the output of the program:

```
1 1 2 2 2 3 3 3 3 4 4 4 4 5
```

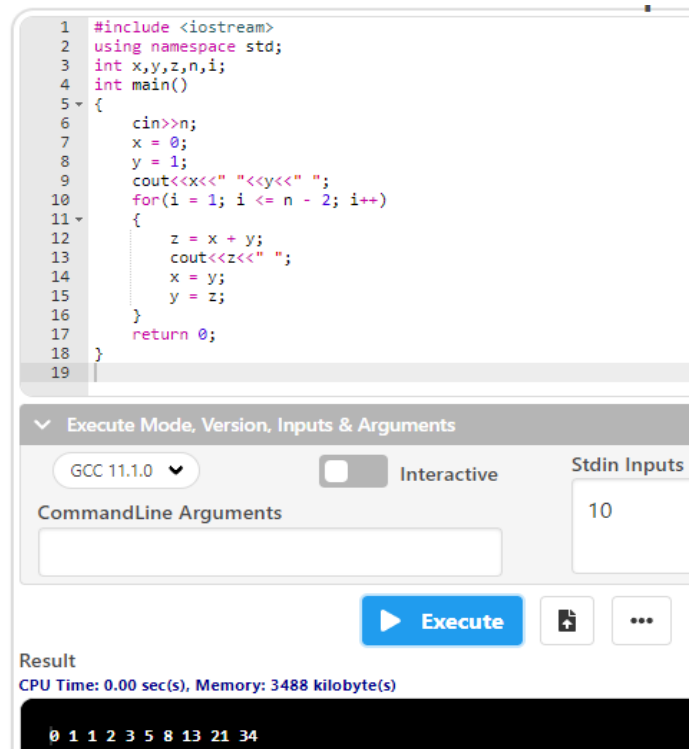
De exemplu:

Dacă se citește pentru **n** valoarea **15** atunci algoritmul va afișa **1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5**

11) Sirul lui Fibonacci. Să se scrie un program C++ care să afișeze primii n termeni ai **sirului lui Fibonacci**: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,

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

```
#include <iostream>
using namespace std;
int x,y,z,n,i;
int main()
{
    cin>>n;
    x = 0;
    y = 1;
    cout<<x<<" "<<y<<" ";
    for(i = 1; i <= n - 2; i++)
    {
        z = x + y;
        cout<<z<<" ";
        x = y;
        y = z;
    }
    return 0;
}
```



De exemplu:

Dacă se citește pentru n valoarea **10** atunci algoritmul va afișa **0, 1, 1, 2, 3, 5, 8, 13, 21, 34**

12) Divizorii unui număr n. Se citește un număr întreg n. Să se scrie un program C++ care să afișeze toți divizorii numărului n.

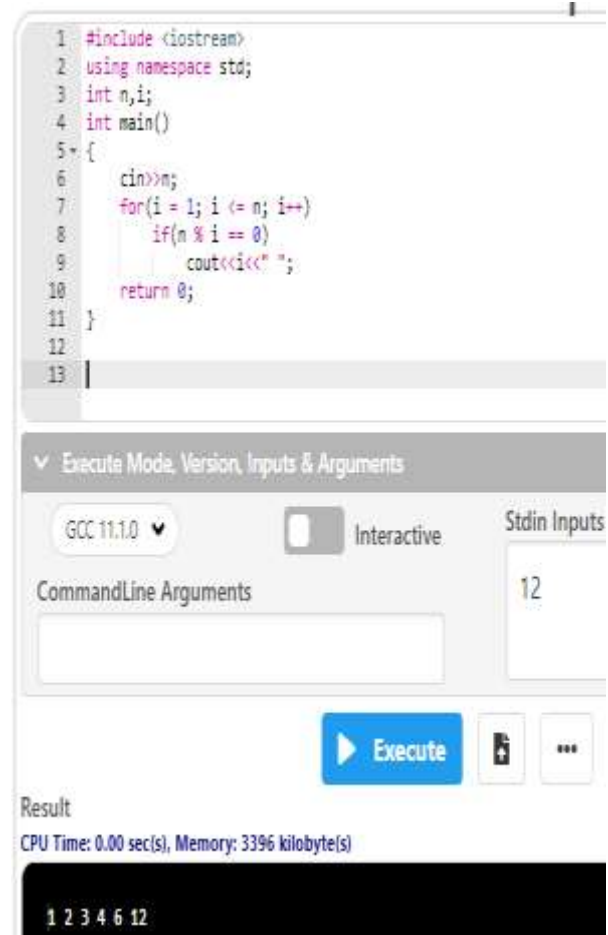
Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
#include <iostream>
using namespace std;
int n,i;
int main()
{
    cin>>n;
    for(i = 1; i <= n; i++)
        if(n % i == 0)
            cout<<i<<" ";
    return 0;
}
```

Exemplu:

Dacă se citește pentru n = 12 atunci algoritmul va afișa „1 2 3 4 6 12”

Iar în cazul în care se citește n = 13 se va afișa „1 13”.



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

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0 Interactive Stdin Inputs

CommandLine Arguments 12

Execute

Result

CPU Time: 0.00 sec(s), Memory: 3396 kilobyte(s)

1 2 3 4 6 12

13) Numere perfecte. Se citește un număr întreg n. Să se scrie un program C++ care să afișeze toate numerele perfecte mai mici sau egale cu n. Spunem că un număr este **perfect** dacă este **egal cu suma divizorilor săi, fără el însuși**.

De exemplu dacă se citește pentru n valoarea 30 atunci algoritmul va afișa 6 28, deoarece aceste două numere sunt singurele pentru care putem scrie:

$$6 = 1 + 2 + 3 \text{ și}$$

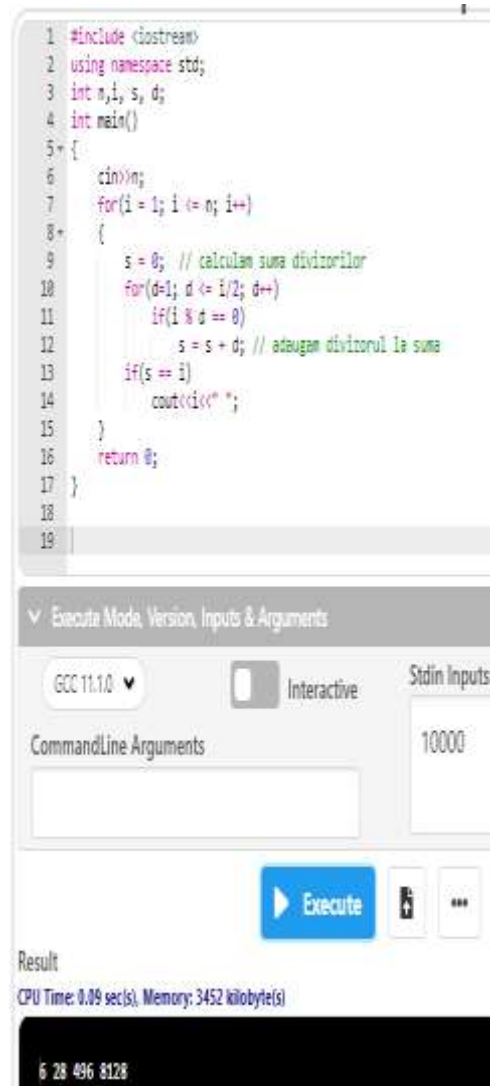
$$28 = 1 + 2 + 4 + 7 + 14.$$

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

```
#include <iostream>
using namespace std;
int n,i, s, d;
int main()
{
    cin>>n;
    for(i = 1; i <= n; i++)
    {
        s = 0; // calculam suma divizorilor
        for(d = 1; d <= i/2; d++)
            if(i % d == 0)
                s = s + d; // adaugam divizorul la suma
        if(s == i)
            cout<<i<<" ";
    }
    return 0;
}
```

Exemplu:

Dacă se citește pentru $n = 10000$, atunci se va afișa: 6 28 496 8128



```
1 #include <iostream>
2 using namespace std;
3 int n,i, s, d;
4 int main()
5 {
6     cin>>n;
7     for(i = 1; i <= n; i++)
8     {
9         s = 0; // calculam suma divizorilor
10        for(d=1; d <= i/2; d++)
11            if(i % d == 0)
12                s = s + d; // adaugam divizorul la suma
13        if(s == i)
14            cout<<i<<" ";
15    }
16    return 0;
17 }
18
19
```

Executa Modă, Version, Inputs & Arguments

GCC 11.1.0 Interactive Stdin Inputs

CommandLine Arguments 10000

Execute

Result

CPU Time: 0.09 sec(s), Memory: 3452 kilobyte(s)

6 28 496 8128

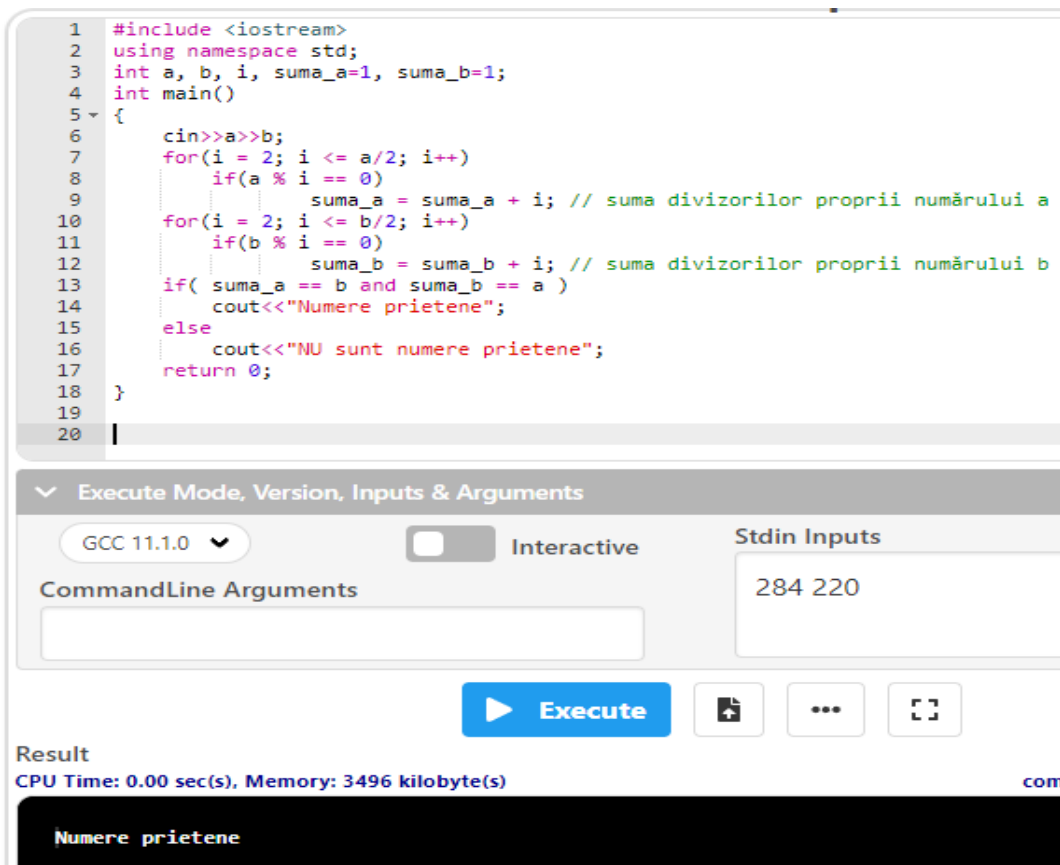
14) Numere prietene. Se citesc două numere întregi a și b . Să se scrie un program C++ care să verifice dacă cele două numere sunt prietene. Spunem ca două numere sunt **prietene** dacă **suma divizorilor proprii ai unui număr este egală cu celalalt și invers**.

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

```
#include <iostream>
using namespace std;
int a, b, i, suma_a=1, suma_b=1;
int main()
{
    cin>>a>>b;
    for(i = 2; i <= a/2; i++)
        if(a % i == 0)
            suma_a = suma_a + i; // suma divizorilor proprii numărului a
    for(i = 2; i <= b/2; i++)
        if(b % i == 0)
            suma_b = suma_b + i; // suma divizorilor proprii numărului b
    if( suma_a == b and suma_b == a )
        cout<<"Numere prietene";
    else
        cout<<"NU sunt numere prietene";
    return 0;
}
```

Exemplu:

Dacă se citește a = 284 și b = 220 atunci algoritmul va afișa mesajul „**numere prietene**”



The screenshot shows a C++ IDE interface. The code editor contains the same code as shown in the previous block. Below the editor, the 'Execute Mode, Version, Inputs & Arguments' section is visible. The compiler version is set to 'GCC 11.1.0'. The 'Interactive' checkbox is unchecked. The 'Stdin Inputs' field contains '284 220'. The 'CommandLine Arguments' field is empty. A blue 'Execute' button is present. Below the execution controls, the 'Result' section shows 'CPU Time: 0.00 sec(s), Memory: 3496 kilobyte(s)' and the output 'Numere prietene'.

15) Factorial. Se citește un număr întreg a. Să se scrie un program C++ care să afișeze să afișeze **n!**. Factorial de n (notat n!) este produsul numerelor de la 1 la n.

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

```
#include <iostream>
using namespace std;
long long n, i, p;
int main()
{
    cin>>n;
    p = 1;
    for(i = 1; i <= n; i++)
        p = p * i;
    cout<<"Factorial de "<<n<<" este egal cu "<<p;
    return 0;
}
```

Exemplu:

Dacă se citește a = 6 atunci algoritmul va afișa 24, deoarece $4! = 1 * 2 * 3 * 4 * 5 * 6 = 720$.

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, i, p;
4 int main()
5 {
6     cin>>n;
7     p = 1;
8     for(i = 1; i <= n; i++)
9         p = p * i;
10    cout<<"Factorial de "<<n<<" este egal cu "<<p;
11    return 0;
12 }
13
```

Below the code editor, there are settings for the execution mode (GCC 11.1.0), an interactive checkbox, and a text input field for Stdin Inputs containing the number 6. A blue 'Execute' button is visible.

The result section shows the output: **Factorial de 6 este egal cu 720**.

Probleme propuse spre rezolvare:

L4.1) *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

L4.2) *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)

L4.3) *Ecuția de gradul 1.* Fie ecuația $a * x = b$ 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

L4.4) *Cel mai mare divizor propriu al unui numar.* 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.

L4.5) *Numere prime pana 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

L4.6) <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:

Daca se dau valorile 100 15 20 5, atunci si va afisa 120

Explicatie

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

L4.7) <https://www.pbinfo.ro/probleme/1650/acelasinumar>

Se da 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:

Daca se dau valorile $n = 25$, $k = 3$ si urmatoarele n valori: 3, 16, -9, 3, atunci se va afisa mesajul NU

Explicatie

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

L4.8) <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 afisa 14 ($1^2+2^2+3^2 = 1 + 4 + 9 = 14$)

L4.9) <https://www.pbinfo.ro/probleme/1681/power>

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

Exemplu:

Dacă $a = 5$ si $b = 4$ atunci se va afisa 625 ($5^4 = 5 * 5 * 5 * 5 = 625$)

L4.10) Să se calculeze suma numerelor naturale cuprinse între două numere date (dintr-un interval).

Exemplu:

Date de intrare: capetele intervalului 3 6

Date de ieșire: suma = 9

L4.11) Se da un număr natural n . Afișați în ordine crescătoare primele n numere naturale nenule.

Exemplu:

Date de intrare: 5

Date de ieșire: 1 2 3 4 5

L4.12) Se da un număr natural n . Afișați pe o linie primele n numere naturale nenule în ordine crescătoare, iar pe linia următoare aceleași numere, dar în ordine descrescătoare.

Exemplu:
Date de intrare: 5
Date de ieșire:
1 2 3 4 5
5 4 3 2 1

L4.13) Se dau n numere naturale nenule. Calculați suma celor n numere date.

Exemplu:
Date de intrare:
5
6 2 0 4 1
Date de ieșire: 13

L4.14) Se dau n numere naturale. Determinați primul număr par dintre cele n numere.

Exemplu:
Date de intrare
5
7 4 2 5 8
Date de ieșire: 4

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