

Laborator nr. 6

Instrucțiuni repetitive în C++ (partea II)

A. Probleme rezolvate:

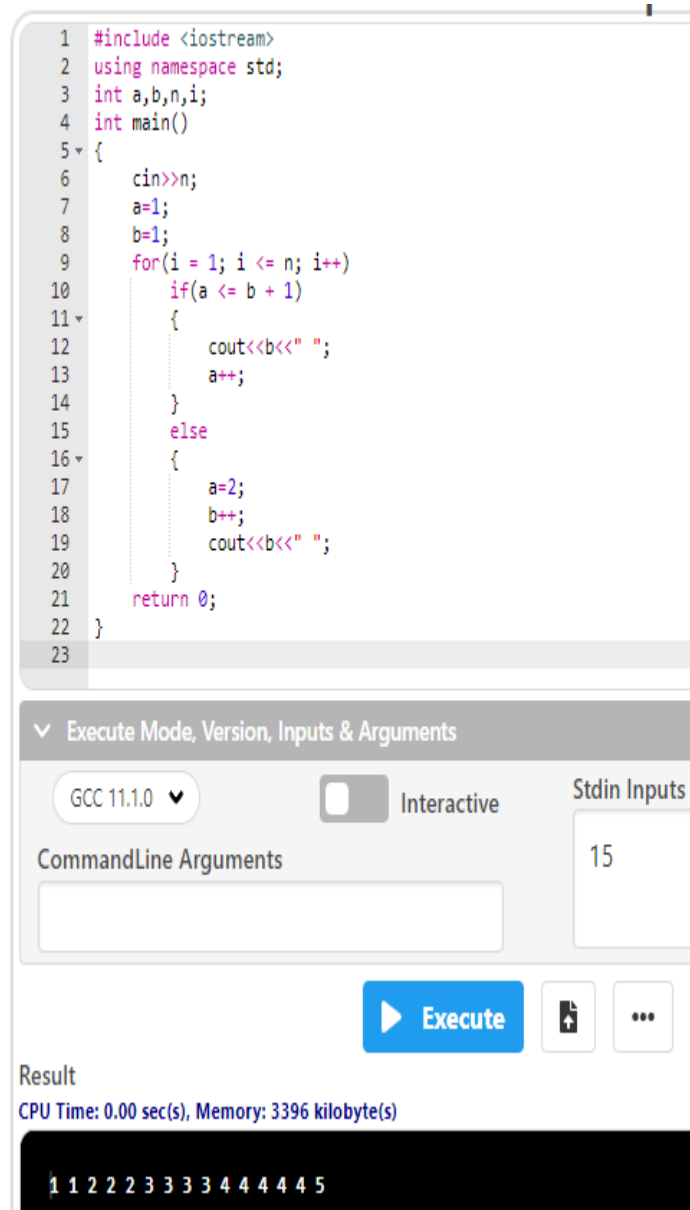
1) Sir generat. Să se scrie un program C++ care să afișeze primii n termeni ai șirului 1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 4,

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;
}
```

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, 4, 5**



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

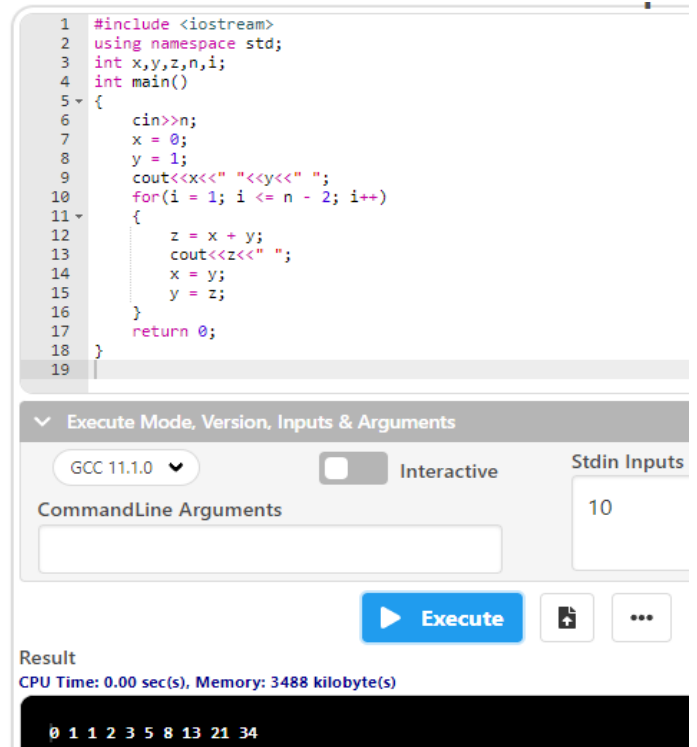
```
1 #include <iostream>
2 using namespace std;
3 int a,b,n,i;
4 int main()
5 {
6     cin>>n;
7     a=1;
8     b=1;
9     for(i = 1; i <= n; i++)
10         if(a <= b + 1)
11         {
12             cout<<b<<" ";
13             a++;
14         }
15         else
16         {
17             a=2;
18 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

Below the code editor, there are settings for the execution mode (GCC 11.1.0), an interactive checkbox, and stdin inputs (15). A blue "Execute" button is visible. The result section shows the output: "1 1 2 2 2 3 3 3 3 4 4 4 4 4 5".

2) Sirul lui Fibonacci. Să se scrie un program C++ care să afișeze primii n termeni ai **șirului 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;
}
```



The screenshot shows a code editor with the following C++ code:

```
1 #include <iostream>
2 using namespace std;
3 int x,y,z,n,i;
4 int main()
5 {
6     cin>>n;
7     x = 0;
8     y = 1;
9     cout<<x<<" "<<y<<" ";
10    for(i = 1; i <= n - 2; i++)
11    {
12        z = x + y;
13        cout<<z<<" ";
14        x = y;
15        y = z;
16    }
17    return 0;
18 }
19
```

Below the code editor, the execution mode is set to 'GCC 11.1.0' and 'Interactive' is checked. The 'Stdin Inputs' field contains the value '10'. The 'Execute' button is visible. The 'Result' section shows the output: '0 1 1 2 3 5 8 13 21 34'.

De exemplu:

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

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

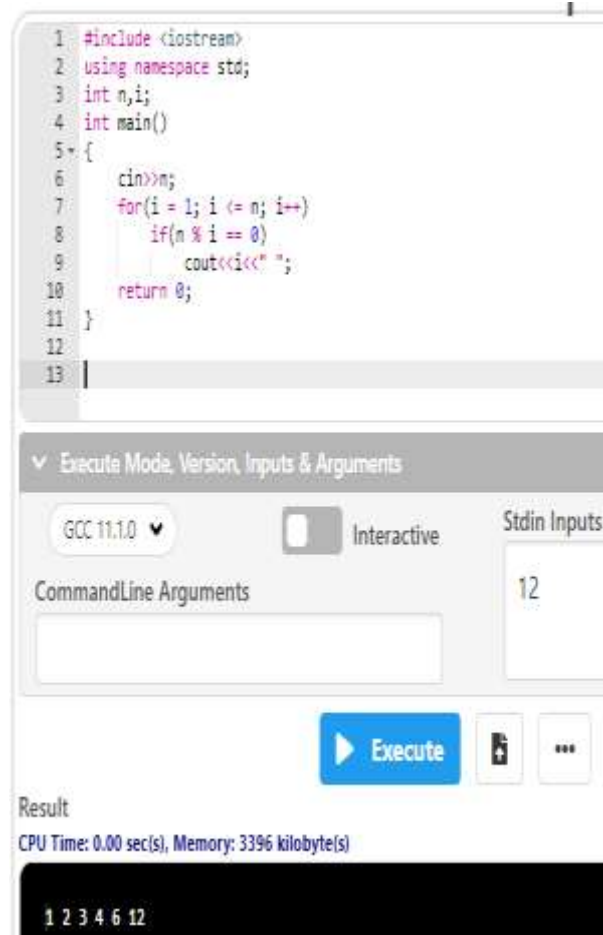
Soluție implementată în compilatorul online <https://www.jdoodle.com/online-compiler-c++/>:

```
#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

4) 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**.

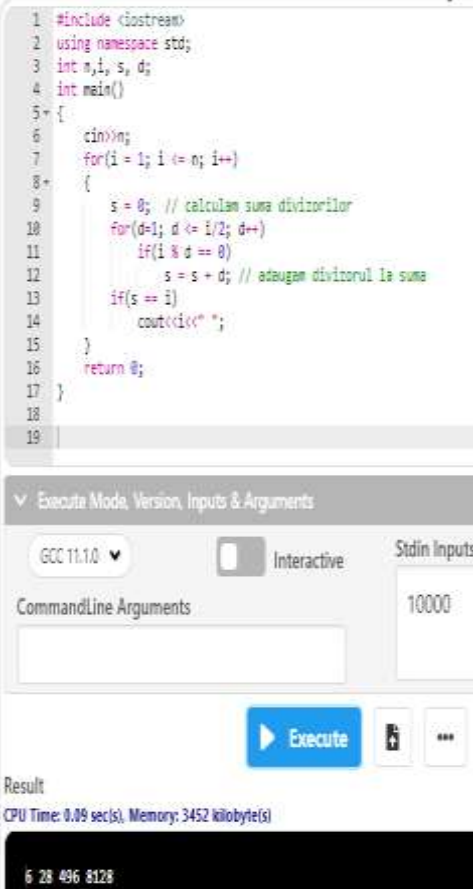
De exemplu dacă se citește pentru n valoarea 30 atunci algoritmul va afișa **6 28**, deoarece aceste două numere sunt singurele pentru care putem scrie:

$$6 = 1 + 2 + 3 \quad \text{și}$$

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

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;
}
```



```
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
```

Execute Mode, 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

Exemplu:

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

5) 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**”

```
1 #include <iostream>
2 using namespace std;
3 int a, b, i, suma_a=1, suma_b=1;
4 int main()
5 {
6     cin>>a>>b;
7     for(i = 2; i <= a/2; i++)
8         if(a % i == 0)
9             suma_a = suma_a + i; // suma divizorilor proprii numărului a
10    for(i = 2; i <= b/2; i++)
11        if(b % i == 0)
12            suma_b = suma_b + i; // suma divizorilor proprii numărului b
13    if( suma_a == b and suma_b == a )
14        cout<<"Numere prietene";
15    else
16        cout<<"NU sunt numere prietene";
17    return 0;
18 }
19
20
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0 Interactive Stdin Inputs
284 220

CommandLine Arguments

Execute

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

Numere prietene

6) Factorial. Se citește un număr întreg a . Să se scrie un program C++ care 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 $4! = 1 * 2 * 3 * 4 * 5 * 6 = 720$.

```
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 |
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0 Interactive Stdin Inputs

CommandLine Arguments

6

Execute

Result

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

Factorial de 6 este egal cu 720

7) Se dă un număr. Să se scrie un program C++ care să afișeze numărul dat, dacă se poate, ca sumă de două numere impare.

Exemple:

Date de intrare: 24

Date de ieșire:

24 = 1 + 23

24 = 3 + 21

24 = 5 + 19

24 = 7 + 17

24 = 9 + 15

24 = 11 + 13

Date de intrare: 33

Date de ieșire: Nu se poate

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

```
#include <iostream>
using namespace std;
long long n, ok, i, j;
int main()
{
    cin>>n;
    for(i = 1; i <= n; i = i + 2)
        for(j = i; j <= n; j = j + 2)
            if(i + j == n)
            {
                ok = 1;
                cout<<i<<' '<<j<<endl;
            }
    if(!ok)
        cout<<"Nu se poate scrie ca suma de doua numere impare";
    return 0;
}
```

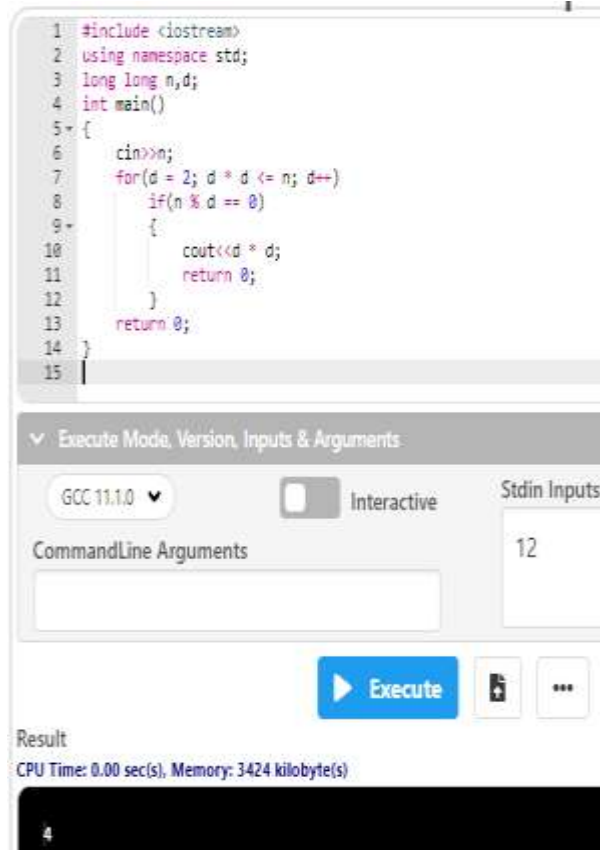
The screenshot shows an online C++ compiler interface. The code editor contains the same code as shown in the previous block. The execution settings are set to GCC 11.1.0, with the 'Interactive' checkbox unchecked. The 'Stdin Inputs' field contains the number 24. The 'Execute' button is highlighted in blue. Below the code editor, the 'Result' section shows the output: '1 23', '3 21', '5 19', '7 17', '9 15', and '11 13'. The CPU time is 0.00 sec(s) and the memory usage is 3428 kilobyte(s).

<https://www.pbinfo.ro/probleme/3663/ppdp>

8) Se dă un număr n. Să se scrie un program C++ care să calculeze pătratul celui mai mic divizor propriu al său.

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

```
#include <iostream>
using namespace std;
long long n, d;
int main()
{
    cin>>n;
    for(d = 2; d * d <= n; d++)
        if(n % d == 0)
        {
            cout<<d * d;
            return 0;
        }
    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 long long n,d;
4 int main()
5 {
6     cin>>n;
7     for(d = 2; d * d <= n; d++)
8         if(n % d == 0)
9         {
10            cout<<d * d;
11            return 0;
12        }
13    return 0;
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 number '12'. A blue 'Execute' button is visible. The 'Result' section shows the output '4' and performance metrics: 'CPU Time: 0.00 sec(s), Memory: 3424 kilobyte(s)'.

Exemplu:

Date de intrare: 12

Date de ieșire: 4

Explicație: Cel mai mic divizor al lui 12 este 2 iar pătratul acestei valori este 4.

<https://www.pbinfo.ro/probleme/408/divizorii-oginditului>

9) Se dă un număr n. Să se scrie un program C++ care să determine numărul de divizori ai oginditului lui n.

Exemplu:

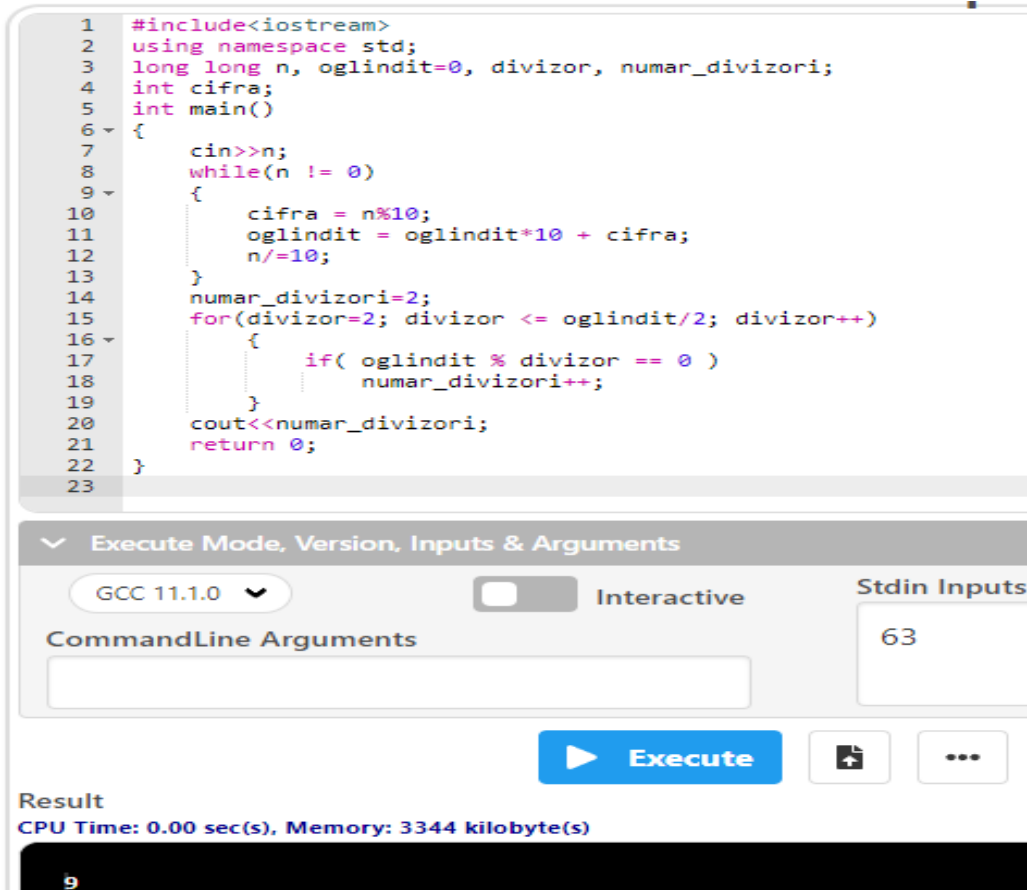
Date de intrare: 63

Date de ieșire: 9

Explicație: Oglinditul lui 63 este 36, care are 9 divizori.


```
#include<iostream>
using namespace std;
long long n, oglindit, divizor, numar_divizori;
int cifra;
int main()
{
    cin>>n;
    while(n != 0)
    {
        cifra = n % 10;
        oglindit = oglindit * 10 + cifra;
        n /= 10;
    }
    numar_divizori = 2;
    for(divizor=2; divizor <= oglindit/2; divizor++)
        if( oglindit % divizor == 0 )
            numar_divizori++;
    cout<<numar_divizori;
    return 0;
}
```

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



```
1  #include<iostream>
2  using namespace std;
3  long long n, oglindit=0, divizor, numar_divizori;
4  int cifra;
5  int main()
6  {
7      cin>>n;
8      while(n != 0)
9      {
10         cifra = n%10;
11         oglindit = oglindit*10 + cifra;
12         n/=10;
13     }
14     numar_divizori=2;
15     for(divizor=2; divizor <= oglindit/2; divizor++)
16     {
17         if( oglindit % divizor == 0 )
18             numar_divizori++;
19     }
20     cout<<numar_divizori;
21     return 0;
22 }
23
```

Execute Mode, Version, Inputs & Arguments

GCC 11.1.0 Interactive Stdin Inputs

CommandLine Arguments

63

Execute

Result

CPU Time: 0.00 sec(s), Memory: 3344 kilobyte(s)

9

<https://www.pbinfo.ro/probleme/429/cifbin>

10) Se dă un număr natural n. Să se scrie un program C++ care să afișeze câte cifre 0 și câte cifre 1 are reprezentarea în baza 2 a acestui număr.

Exemplu:

Date de intrare: 174

Date de ieșire: 3 5

Explicație: Reprezentarea în baza 2 a lui 174 este 10101110.

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

```
#include <iostream>
using namespace std;
long long n, nr_cifre0, nr_cifre1;
int main()
{
    cin>>n;
    while(n!=0)
    {
        if(n % 2 == 0) nr_cifre0++;
        else nr_cifre1++;
        n /= 2;
    }
    cout<<nr_cifre0<<" "<<nr_cifre1;
    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 long long n, nr_cifre0, nr_cifre1;
4 int main()
5 {
6     cin>>n;
7     while(n!=0)
8     {
9         if(n % 2 == 0) nr_cifre0++;
10        else nr_cifre1++;
11        n /= 2;
12    }
13    cout<<nr_cifre0<<" "<<nr_cifre1;
14    return 0;
15 }
16
```

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 '174'. The 'CommandLine Arguments' field is empty. A blue 'Execute' button is visible.

The 'Result' section shows the output: '3 5'. Below the result, the CPU time is 0.00 sec(s) and the memory usage is 3492 kilobyte(s).

B. Probleme propuse spre rezolvare:

L6.1) 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

L6.2) Se dă un număr. Să se scrie, dacă este posibil, ca sumă de două numere consecutive.

Exemple:

Date de intrare: 5

Date de ieșire: $5 = 2 + 3$

Date de intrare: 6

Date de ieșire: Nu se poate

L6.3) Dându-se un număr natural n , să se găsească toate posibilitățile de scriere a acestui număr ca sumă de numere consecutive.

Exemplu:

Date de intrare: 15

Date de ieșire:

$15 = 1 + 2 + 3 + 4 + 5$

$15 = 4 + 5 + 6$

$15 = 7 + 8$

L6.4) Se dă 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

L6.5) Se dă 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

L6.6) 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

L6.7) Se dă un număr întreg n și alte k numere întregi. Să se afle dacă, adunând toate cele k numere la n se obține o valoare egală cu valoarea inițială a lui n .

Exemplu:

Date de intrare:

25

3

16 -9 3

Date de ieșire: NU

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

L6.8) 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

<https://www.pbinfo.ro/probleme/122/cifrebinare>

L6.9) Se citesc două numere naturale. Să se afișeze numărul care are mai multe cifre egale cu 1 în reprezentarea în baza 2.

Exemplu:

Date de intrare: 125 1250

Date de ieșire: 125

Explicație: Reprezentarea lui 125 în baza 2 este 1111101, cu 6 cifre 1, iar reprezentarea lui 1250 este 10011100010, cu 5 cifre 1. Astfel, 125 are mai multe cifre 1 în reprezentarea în baza 2.

<https://www.pbinfo.ro/probleme/947/baza16>

L6.10) În baza 16, cifrele unui număr pot fi 0, 1, 2, ..., 9, A, B, C, D, E, F. Fiind dat n un număr natural, afișați scrierea lui n în baza 16.

Exemplu

Date de intrare: 3024

Date de ieșire: BD0

Explicație: $3024 : 16 = 189$ rest 0; $189 : 16 = 11$ rest 13; $11 : 16 = 0$ rest 11. Cum $11 = B$, $13 = D$, $0 = 0$, rezultă BD0.

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 implementati in limbajul C++. Volumul I – Algoritmi elementari*”, Editura Academica Brâncuși din Târgu Jiu, 2021, ISBN 978-606-9614-06-8