

# tema1<sub>S</sub>I

Jessica Soare

October 2025

**Exercițiul 1 – Sa cream un script care afiseaza lista de argumente de o primeste de la tastatura;**

```
GNU nano 7.2                                         ex1.cpp
#include <iostream>
using namespace std;

int main(int argc, char **argv){
    int i;
    for(i=0; i<= argc; i++)
        cout<<argv[i]<<" ";
    cout<<"\n";

    return 0;
}
```

**Explicație:**

Am afisat pe rand folosind un **for** toate argumentele primite de la tastatura  
**Makefile-ul:**

```
GNU nano 7.2                                         Makefile
compilare:
    g++ -o exec ex1.cpp

rulare:
    ./exec 1 2 3
```

**Rezultatul rularii:**

```
jessica@DESKTOP-FIANKP5:~/tema1/ex1$ make compilare
g++ -o exec ex1.cpp
jessica@DESKTOP-FIANKP5:~/tema1/ex1$ make rulare
./exec 1 2 3
./exec 1 2 3 jessica@DESKTOP-FIANKP5:~/tema1/ex1$ m
```

**Exercițiul 2 – realizați un program în C sau C++ care primește ca argument număr întreg N (N ≤ 90) și tipărește primii N termeni ai secvenței Fibonacci.**

```
GNU nano 7.2                                         ex2.cpp
#include <iostream>
using namespace std;

int main(int argc, char **argv){
    long long a=0, b=1, c;
    int n = atoi(argv[1]);
    cout<<a<<" "<<b<<" ";
    for (int i=2; i<n; i++){
        c = a+b;
        a = b;
        b = c;
        cout<<c<<" ";
    }

    return 0;
}
```

**Explicație:**

Am salvat primii termeni ai sirului în variabilele **a** și **b** și le am afisat în alta variabilă **c** am retinut suma ultimilor 2 termeni ai sirului iar după aceea schimbam valorile din **a** și **b**.

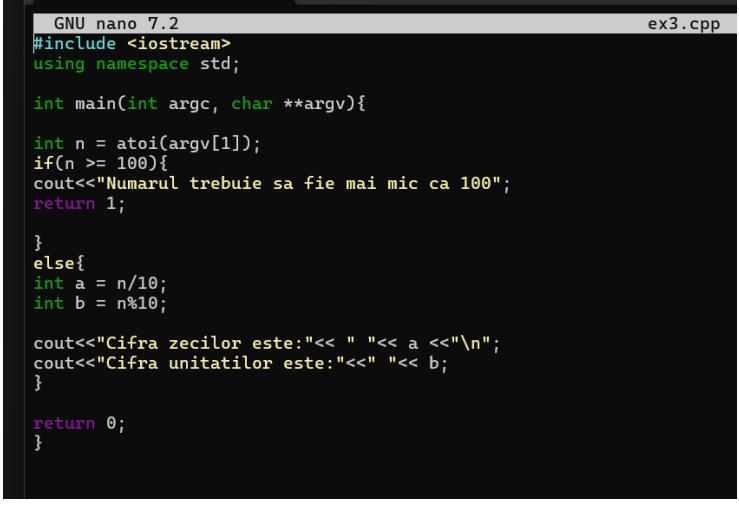
**Makefile-ul:**

```
GNU nano 7.2                                         Makefile
compilare:
    g++ -o exec ex2.cpp
rulare:
    ./exec 6
```

**Rezultatul rularii:**

```
jessica@DESKTOP-FIANKP5:~/tema1/ex2$ make compilare
g++ -o exec ex2.cpp
jessica@DESKTOP-FIANKP5:~/tema1/ex2$ make rulare
./exec 6
0 1 1 2 3 5 jessica@DESKTOP-FIANKP5:~/tema1/ex2$ |
```

**Exercițiul 3 – realizați un program în C/C++ care primește ca argument un număr întreg de N cifre (N ≤ 100) și tipărește lista cifrelor numărului.**



```
GNU nano 7.2 ex3.cpp
#include <iostream>
using namespace std;

int main(int argc, char **argv){
    int n = atoi(argv[1]);
    if(n >= 100){
        cout<<"Numarul trebuie sa fie mai mic ca 100";
        return 1;
    }
    else{
        int a = n/10;
        int b = n%10;

        cout<<"Cifra zecilor este:"<< " "<< a <<"\n";
        cout<<"Cifra unitatilor este:"<< " "<< b;
    }
    return 0;
}
```

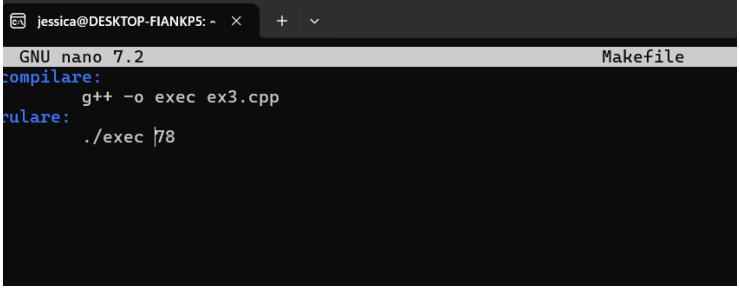
**Explicație:**

I-am atribuit lui n valoarea data ca si argument.

In variabila a am salvat cifra zecilor.

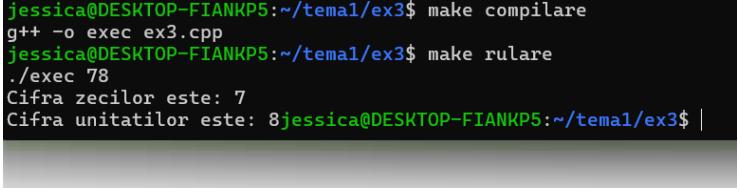
In variabila b am salvat cifra unitatilor

**Makefile-ul:**



```
GNU nano 7.2 Makefile
compilare:
    g++ -o exec ex3.cpp
rulare:
    ./exec 78
```

**Rezultatul rularii:**



```
jessica@DESKTOP-FIANKP5:~/tema1/ex3$ make compilare
g++ -o exec ex3.cpp
jessica@DESKTOP-FIANKP5:~/tema1/ex3$ make rulare
./exec 78
Cifra zecilor este: 7
Cifra unitatilor este: 8jessica@DESKTOP-FIANKP5:~/tema1/ex3$ |
```

**Exercițiul 4 – realizați un program în C/C++ care citește de la tastatură un sir de caractere format din unul sau mai multe cuvinte separate prin unul sau mai multe spați și tipărește cuvintele, unul pe linie, într-un cadru dreptunghiular.**

```

GNU nano 7.2
#include <iostream>
#include <string>
using namespace std;
int main(){
    string linie;
    cout<<"Introduceți un sir de caractere";
    getline(cin, linie);
    int lungime_cuv;
    int lung_max = 0;
    for(char c : linie){
        if(c == ' '){
            lungime_cuv++;
            if(lungime_cuv > lung_max)
                lung_max = lungime_cuv;
        }
        else{
            lungime_cuv = 0;
        }
    }
    for (int i=0; i<= lung_max+4; i++)
        cout<<"*";
    cout<<endl;

    string cuvant = "";
    for(char c:linie){
        if(c==' ')
            cuvant += cuvant + c;
        else if(cuivant == ""){
            int spatii_total = (lung_max+4) - cuvant.length();
            int spatii_s = spatii_total / 2;
            int spatii_d = spatii_total - spatii_s;
            cout<<"+";
            for(int i=0; i<spatii_s; i++) cout << " ";
            cout<<cuvant;
            for(int i=0; i<spatii_d; i++) cout << " ";
            cout<<"+<endl;
            cuvant= " ";
        }
        if(cuivant != ""){
            int spatii_tot = (lung_max+4) - cuvant.length();
            int spatii_s = spatii_tot / 2;
            int spatii_d = spatii_tot - spatii_s;
            cout<<"+";
            for(int i=0; i<spatii_s; i++) cout << " ";
            cout<<cuvant;
            for(int i=0; i<spatii_d; i++) cout << " ";
            cout<<"+<endl;
        }
    }
    for(int i=0; i<lung_max+4; i++)
        cout<<"*";
    cout<<endl;
}
return 0;
}

```

**Explicație:**

Am citit sirul de caractere de la tastatura;

Am aflat care este lungimea maxima a cuvintelor din sir;

Atât timp cat caracterul era diferit de spatiu, incrementam lungimea curenta iar apoi o comparăm cu lungimea maxima

Am afisat prima linie de stelute(+4 -; avem 2 stele in plus pentru capetele cuvantului si inca 2 pentru bordura)

Dupa care am salvat intr-un string fiecare cuvant din linie si am calculat cate spatii trebuie puse inaintea lui si dupa acesta

Am adaugat si steluta din stanga si din dreapta.

Am facut asta si pentru ultimul cuvant

Am afisat linia de final

**Makefile-ul:**

```
GNU nano 7.2                                     Makefile
compilare:                                         g++ -o exec ex4.cpp
rulare:                                           ./exec
```

Rezultatul rularii:

```
jessica@DESKTOP-FIANKP5:~/tema1/ex4$ make compilare
g++ -o exec ex4.cpp
jessica@DESKTOP-FIANKP5:~/tema1/ex4$ make rulare
./exec
Introduceti un sir de caractereana are mere
*****
* ana *
* are *
* mere *
*****
```