

## Laborator nr. 5

### Tablouri unidimensionale – vectori

#### A. Probleme rezolvate:

**Problema 1:** Se considera un tablou unidimensional cu n numere reale și se cere sa se afișeze cel mai mare și cel mai mic element din tablou.

**Solutie:**

```
#include <iostream>
using namespace std;
int i, n;
float x[50], minim, maxim;
int main(void)
{
    cin>>n;
    for(i = 1; i <= n; i++)
        cin>>x[i];
    // initializam minimul si maximul cu primul element
    din vector
    minim = x[1];
    maxim = x[1];
    for(i = 2; i <= n; i++)
        if(minim > x[i]) minim = x[i];
        else if(maxim < x[i]) maxim = x[i];
    cout<<endl<<"Minimul este "<<minim;
    cout<<endl<<"Maximul este "<<maxim;
    return 0;
}
```

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

The screenshot shows an online C++ compiler IDE. The code in the editor is as follows:

```
1: #include <iostream>
2: using namespace std;
3: int i, n;
4: float x[50], minm, maxm;
5: int main(void)
6: {
7:     cin>>n;
8:     for(i = 1; i <= n; i++)
9:         cin>>x[i];
10:    // initializam minimul si maximul cu primul element din vector
11:    minm = x[1];
12:    maxm = x[1];
13:    for(i = 1; i <= n; i++)
14:        if(minm > x[i]) minm = x[i];
15:        else if(maxm < x[i]) maxm = x[i];
16:    cout<<endl<<"Minimul este "<<minm;
17:    cout<<endl<<"Maximul este "<<maxm;
18:    return 0;
19: }
```

The IDE interface includes a dropdown menu for the compiler version (GCC 9.1.0), an 'Interactive' checkbox, a 'Stdin Inputs' field with the value '7' and a list of numbers '12.34 -78 -89.23 34.0 1234.6 -64.5 11', and an 'Execute' button. Below the code, the 'Result' section shows the output: 'Minimul este -89.23' and 'Maximul este 1234.6'. The CPU time is 0.00 seconds and memory is 3368 kilobytes.

**Problema 2:** Fie tabloul unidimensional x cu n elemente numere reale și numerele întregi a și b. Sa se calculeze media aritmetică a elementelor din tablou cuprinse între valorile a și b.

**Solutie:**

```
#include <iostream>
using namespace std;
int a, b, i, n, suma, nr_elemente;
float x[50], media;
int main(void)
{
    cin>>n;
    for(i = 1; i <= n; i++) cin>>x[i];
    cin>>a>>b;
    for(i = 1; i <= n; i++)
        if( (a <= x[i]) && (x[i] <= b) ) {
            suma += x[i];
            nr_elemente++;
        }
    media = (1.0*suma) / nr_elemente;
    cout<<"Media aritmetica a elementelor din
    tablou, aflate intre "<<a<<" si "<<b<<" este
    "<<media;
    return 0;
}
```

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

Online C++ Compiler IDE

```
1 #include <iostream>
2 using namespace std;
3 int a, b, i, n, suma, nr_elemente;
4 float x[50], media;
5 int main(void)
6 {
7     cin>>n;
8     for(i = 0; i <= n; i++)
9         cin>>x[i];
10    cin>>a;
11    cin>>b;
12    for(i = 0; i <= n; i++)
13        if( (a <= x[i]) && (x[i] <= b) )
14        {
15            suma += x[i];
16            nr_elemente++;
17        }
18    media = (1.0*suma) / nr_elemente;
19    cout<<"Media aritmetica a elementelor din tablou, aflate între "<<a<<" și "<<b<<" este "<<media;
20    return 0;
21 }
```


Execute Mode, Version, Inputs & Arguments

GCC9.10  Interactive

CommandLine Arguments

Stdin Inputs

```
7
12.34 -78 -89.23 34.0 1234.6 -64.5 11
-300 2000
```

**Execute**   

Result

CPU Time: 0.00 sec(s). Memory: 3436 kilobyte(s)

```
Media aritmetica a elementelor din tablou, aflate între -300 și 2000 este 151.286
```

**Problema 3:** Se citesc doua tablouri unidimensionale cu componente numere naturale. Fiecare tablou are elementele sortate crescator. Se cere sa se construiască un al treilea tablou care contine elementele celor două în ordine crescătoare. (*Problema interclasarii*)

**Solutie:**

```
#include <iostream>
using namespace std;
int i, j, n, m, k;
float x[50], y[50], z[50];
int main(void)
{
    cin>>n;
    for(i = 1; i <= n; i++) cin>>x[i];
    cin>>m;
    for(i = 1; i <= m; i++) cin>>y[i];
    // parcurgem cei doi vectori care au elementele în ordine crescătoare
    i = 1; j = 1; k = 0;
    while( (i <= n) && (j <= m) )
        // dacă un element din primul vector este mai mic decat un alt
        // element din cel de-al doilea vector, atunci il copiem in vectorul cu
        // elementele interclasate
        if(x[i] < y[j]){
            k++;    z[k] = x[i];    i++;
        }
        // în caz contrar copiem elementul din cel de-al doilea vector
        else {
            k++;    z[k] = y[j];    j++;
        }
        // dacă mai sunt elemente în primul vector, atunci le copiem in vectorul
        // interclasat
        if(i <= n)
            for(j = i; j <= n; j++) {
                k++;
                z[k]=x[j];
            }
        // dacă mai sunt elemente în al doilea vector, atunci le copiem in vectorul
        // interclasat
        else
            for(i = j; i <= m; i++) {
                k++;
                z[k] = y[i];
            }
        cout<<"\nVectorul Z cu elementele interclasate este: ";
        for(i = 1; i <= k; i++) cout<<z[i]<<" ";
        return 0;
}
```

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

Online C++ Compiler IDE

```
1) #include <iostream>
2) using namespace std;
3) int i, j, n, m, k;
4) float a[50], y[50], z[50];
5) int main(void)
6) {
7)     cin>>n;
8)     for(i = 1; i <= n; i++)
9)         cin>>a[i];
10)    cin>>m;
11)    for(i = 1; i <= m; i++)
12)        cin>>y[i];
13)    // parcurgem cel mai vectori care au elementele in ordine crescatoare
14)    i = 1; j = 1; k = 0;
15)    while( (i <= n) && (j <= m) )
16)        // daca un element din primul vector este mai mic decat un alt element din cel de-al doilea vector, atunci il copiem in vectorul cu elementele interclasate
17)        if(a[i] < y[j])
18)        {
19)            k++;
20)            z[k] = a[i];
21)            i++;
22)        }
23)        // in caz contrar copiem elementul din cel de-al doilea vector
24)        else
25)        {
26)            k++;
27)            z[k] = y[j];
28)            j++;
29)        }
30)    // daca mai sunt elemente in primul vector, atunci le copiem in vectorul interclasat
31)    if(i <= n)
32)        for(j = 1; j <= n; j++)
33)        {
34)            k++;
35)            z[k]=a[j];
36)        }
37)    // daca mai sunt elemente in al doilea vector, atunci le copiem in vectorul interclasat
38)    else
39)        for(j = 1; j <= m; j++)
40)        {
41)            k++;
42)            z[k] = y[j];
43)        }
44)    cout<<"Vectorul Z cu elementele interclasate este: ";
45)    for(i = 1; i <= k; i++) cout<<z[i]<<" ";
46)    return 0;
47) }
```

Execute Mode, Version, Inputs & Arguments

GCC 8.1.0  Interactive Still Inputs

CommandLine Arguments

5  
1 4 7 12 34  
7  
2 5 6 11 15 21 35

Execute

Result

CPU Time: 0.00 sec(s), Memory: 3692 kilobyte(s)

Vectorul Z cu elementele interclasate este: 1 2 4 5 6 7 11 12 15 21 34 35

**Problema 4:** Se da un tablou unidimensional cu n componente numere intregi si un numar intreg A. Sa se numere cate elemente sunt mai mari decat a si sa se construiasca un vector cu aceste elemente.

**Solutie:**

```
#include <iostream>
using namespace std;
int i, n, j, a;
float x[50], y[50];
int main(void)
{
    cin>>n;
    for(i = 1; i <= n; i++) cin>>x[i];
    cin>>a;
    j = 0;
    for(i = 1; i <= n; i++)
        if (a <= x[i]) {
            j++;
            y[j] = x[i];
        }
    cout<<"Sunt "<<j<<" numere mai mari decat
"<<a<<endl;
    cout<<"Elementele sunt: \n";
    for(i = 1; i <= j; i++) cout<<y[i]<<" ";
    return 0;
}
```

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

The screenshot shows an online C++ compiler IDE interface. At the top right, it says "Online C++ Compiler IDE". The main area contains a code editor with the following code:

```
1 #include <iostream>
2 using namespace std;
3 int i, n, j, a;
4 float x[50], y[50];
5 int main(void)
6 {
7     cin>>n;
8     for(i = 1; i <= n; i++)
9         cin>>x[i];
10    cin>>a;
11    j = 0;
12    for(i = 1; i <= n; i++)
13        if (a <= x[i]) {
14            j++;
15            y[j] = x[i];
16        }
17    cout<<"Sunt "<<j<<" numere mai mari decat "<<a<<endl;
18    cout<<"Elementele sunt: \n";
19    for(i = 1; i <= j; i++) cout<<y[i]<<" ";
20    return 0;
21 }
22
23
```

Below the code editor, there are settings for the compiler (GCC 9.1.0) and a checkbox for "Interactive". The "Stdin Inputs" section shows the input values: 5, 1 4 7 12 34, and 7. There is an "Execute" button and a "Clear" button.

The "Result" section shows the output of the program:

```
Result
CPU Time: 0.00 sec(s). Memory: 2664 kilobyte(s)

Sunt 3 numere mai mari decat 7
Elementele sunt:
7 12 34
```

**Problema 5:** Se da un tablou unidimensional cu  $n$  ( $1 \leq n \leq 100$ ) componente numere naturale. Sa se calculeze suma componentelor divizibile cu  $a$ , pentru un număr  $a$  citit de la tastatura.

**Solutie:**

```
#include <iostream>
using namespace std;
int i, n, a, suma;
int x[50];
int main(void)
{
    cin >> n;
    for(i = 1; i <= n; i++)
        cin >> x[i];
    cin >> a;
    for(i = 1; i <= n; i++)
        if ( x[i] % a == 0) suma = suma + x[i];
    cout << "Suma elementelor din vector care sunt
divizibile cu " << a << " este egala cu " << suma;
    return 0;
}
```

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

The screenshot shows an online C++ compiler interface. At the top right, it says "Online C++ Compiler IDE". The code editor contains the C++ code from the solution. Below the code editor, there is a control panel with "GCC 9.1.0" selected, an "Interactive" checkbox, and a "Stdin Inputs" section with the values "5", "1 4 7 12 34", and "2". A blue "Execute" button is visible. Below the control panel, the "Result" section shows the output: "Suma elementelor din vector care sunt divizibile cu 2 este egala cu 50".

## B. Probleme propuse spre rezolvare:

**L5\_1).** Se citește un vector cu  $n$  elemente, numere naturale. Să se afișeze elementele din vector care sunt multipli ai ultimului element.

Date de intrare: Programul citește de la tastatură numărul  $n$ , iar apoi  $n$  numere naturale, reprezentând elementele vectorului.

Date de ieșire: Programul afișează pe ecran elementele cerute, separate prin exact un spațiu, în ordinea din vector.

Exemplu

Intrare

5

7 4 9 6 2

Ieșire

4 6 2

**L5\_2).** Se dă un vector cu  $n$  elemente, numere naturale și un număr  $k$ . Ordonăți crescător primele  $k$  elemente ale vectorului și descrescător ultimele  $n-k$  elemente.

Date de intrare: Programul citește de la tastatură numerele  $n$  și  $k$ , iar apoi  $n$  numere naturale, reprezentând elementele vectorului.

Date de ieșire: Programul va afișa pe ecran elementele vectorului, separate prin exact un spațiu, după efectuarea operațiilor cerute.

Exemplu

Intrare

7 3

13 1 10 15 3 7 11

Ieșire

1 10 13 15 11 7 3

**L5\_3).** Se dă un șir cu  $n$  elemente, numere naturale. Să se verifice dacă în șir există elemente prime.

Date de intrare: Programul citește de la tastatură numărul  $n$ , iar apoi cele  $n$  elemente ale șirului.

Date de ieșire: Programul afișează pe ecran mesajul DA, dacă șirul conține elemente prime, respectiv NU în caz contrar.

Exemplu

Date de intrare

5

21 8 6 10 8

Date de ieșire

NU



**Laborator - Programarea Calculatoarelor si Limbaje de Programare (2022)**  
**Limbajul C++**

**Adrian Runceanu**

**L5\_4).** Se dă un șir cu n elemente, numere naturale. Să se verifice dacă în șir există elemente impare.

Date de intrare: Programul citește de la tastatură numărul n, iar apoi cele n elemente ale șirului, separate prin spații.

Date de ieșire: Programul afișează pe ecran mesajul DA, dacă șirul conține elemente impare, respectiv NU în caz contrar.

Exemplu

Date de intrare

5

2 8 6 10 8

Date de ieșire

NU

**L5\_5).** Se dă un vector cu n numere naturale. Să se determine câte dintre elementele vectorului sunt prime cu ultimul element.

Date de intrare: Programul citește de la tastatură numărul n, iar apoi n numere naturale, separate prin spații, reprezentând elementele vectorului.

Date de ieșire: Programul afișează pe ecran numărul C, reprezentând valoarea cerută.

Exemplu

Intrare

6

50 64 15 28 35 75

Ieșire

2

Explicație: Elementele din vector prime cu 75 sunt 64 și 28.

**L5\_6).** Se considera un vector  $\mathbf{x} = \{ x_1, x_2, x_3, \dots, x_n \}$ . Sa se scrie un program C++ care sa modifice vectorul astfel încât în final sa avem:

a)  $\mathbf{x} = \{ x_2, x_3, \dots, x_n, x_1 \}$

b)  $\mathbf{x} = \{ x_n, x_1, x_2, x_3, \dots, x_{n-1} \}$

c)  $\mathbf{x} = \{ x_2, x_1, x_4, x_3, \dots, x_n, x_{n-1} \}$

**L5\_7).** Se considera doi vectori  $\mathbf{x}$  și  $\mathbf{y}$  cu câte n elemente numere întregi fiecare. Sa se scrie un program C++ care sa calculeze următoarele valori:

a)  $e = (x_1+y_1) \cdot (x_2+y_2) \cdot \dots \cdot (x_n+y_n)$

b)  $e = x_1y_1 + x_2y_2 + \dots + x_ny_n$

c)  $e = x_1^2 y_1 + x_2^2 y_2 + \dots + x_n^2 y_n$

**L5\_8).** Se considera un vector  $\mathbf{x}$  care are elemente numere reale. Sa se scrie un program C++ care sa introduca media geometrica intre oricare doua elemente vecine din vector. Vectorul se va afișa înainte și după modificare.

**L5\_9).** Sa se scrie un program C++ care sa realizeze inversarea unui vector:

a) in acelasi vector si fara a utiliza un vector suplimentar

b) intr-un alt vector.

**L5\_10).** Se considera un vector  $\mathbf{x}$  cu  $n$  numere întregi. Să se copieze într-un alt vector  $\mathbf{y}$  elementele pozitive ale vectorului inițial.

## 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