

# Agenda

---

- Foarte important de citit
- Introducere Vectori/Array-uri
- Rezolvare exercitii tema

## Foarte important de citit

- Sa iti amintesti ca scopul a ceea ce facem nu este ca tu sa rezolvi cat mai multe probleme ci sa le intelegi.
- Daca tu ai o implementare care difera de a mea, nu inseamna ca e gresita. Orice problema se rezolva in mai multe feluri.
- Daca te blochezi, este ok sa cauti pe internet, dar nu cauta din prima. Incearca ce iti vine in cap, ruleaza, vezi ce eroare ai, cauta eroarea pe net, vezi ce intelegi din ea.
  - Pare ceva greoi dar sa stii ca cel mai mult vei invata din exercitiile care nu iti iasa din prima!
- Experimenteaza cu programul odata ce l-ai finalizat si functioneaza asa cum trebuie.
  - Sterge variabile, sterge punctul si virgula, pune ceva valori random
  - Asta te va ajuta pe viitor deoarece vei stii sa rezolvi imediat o eroarea care iti e familiara
- Cel mai bun sfat pe care ti-l pot da este sa fii cat mai curios si mereu sa te gandesti la toate cazurile posibile, sa incerci sa vezi problema si din alte puncte si...
  - Incearca
  - Greseste
  - Repeta doar asa inveti.
- Si aminteste-ti, aici nu e pe note 😊

## Introducere Vectori/Array-uri

1. Ce este un vector (sau tablou sau array cum zic oamenii normali 😊)?

- Un vector este o colecție de valori de același tip, stocate una după alta în memorie. Poți să te gândești la el ca la o cutie cu mai multe sertare, numerotate de la 0 la N-1.

```
int v[5]; // vector de 5 elemente întregi
```

2. Declararea unui vector

- Pentru a declara un vector, trebuie mai intai sa stim ce tip de date vrem sa stocam acolo. Deci putem generaliza ca declararea unui vector arata ceva de genul:

```
tip vector[nr_elemente];
```

- Si ca sa vedem mai multe exemple:

```
int v[100];      // vector de 100 de întregi
float note[30];  // vector de 30 de float-uri
char litere[10]; // vector de 10 caractere
```

### 3. Accesarea elementelor

- Cel mai important lucru de retinut este faptul ca vectorii se indexeaza de la 0!

```
v[0] = 7;      // primul element
v[4] = 10;     // al cincilea element
int x = v[2];  // x primește valoarea de pe poziția 2
```

### 4. Initializarea unui vector

- Completa

```
int v[5] = {1, 2, 3, 4, 5};
```

- Partiala (restul devine 0)

```
int v[5] = {1, 2}; // echivalent cu {1, 2, 0, 0, 0}
```

### 5. Citirea și afișarea elementelor (foarte frecvent la Bac!)

- Exemplu clasic:

```
#include <iostream>
using namespace std;

int main() {
    int n;
    cin >> n; // câte elemente vom citi
    int v[n];
    for (int i = 0; i < n; i++)
        cin >> v[i];

    for (int i = 0; i < n; i++)
```

```
        cout << v[i] << " ";

        return 0;
    }
```

## 6. Operatii de baza cu vectori

- Suma elementelor

```
int suma = 0;
for (int i = 0; i < n; i++) {
    suma += v[i];
}
```

- Cautarea unui element `x`:

```
int x;
cin >> x;
int gasit = 0;
for (int i = 0; i < n; i++){
    if (v[i] == x) {
        gasit = 1;
        break;
    }
}
if (gasit) {
    cout << "Elementul a fost gasit";
} else {
    cout << "Elementul nu a fost gasit";
}
```

- Numararea elementelor pare:

```
int cnt = 0;
for (int i = 0; i < n; i++) {
    if (v[i] % 2 == 0) {
        cnt++;
    }
}
```

## 7. Important de stiut.

- Atunci cand declari un vector de genul `int numere[10]`; si incerci sa afisezi primul element (`cout<<numere[0]`) sunt sanse foarte mari sa vezi un numar afisat chit ca tu nu ai apucat sa initializezi acel vector.

- Acest lucru se intampla pentru ca atunci cand tu declari un vector, sistemul de operare iti alocă in memorie spatiul pentru el insa nu face si curat. Este ca si cum tu ai inchiria o camera dar o primești exact așa cum a lasat-o fostul chirias. Poate fi curata, poate fi haos.
- De aceea, pentru a te asigura ca nu ai valori random din memorie este bine sa iti initializezi tu vectorul inainte de a-l folosi
- Pe viitor o sa avem nevoie de trick-ul de mai jos unde initializam toate elementele vectorului cu 0:

```
int frecventa[10] = {0}; // Toate elementele o sa aibe valoarea 0.
```

## - Rezolvare exercitii tema

### 1. Exerciitiu 1

- Link: <https://www.pbinfo.ro/probleme/633/paritate1>
- Solutie:

```
#include <iostream>
#include <cstring>

using namespace std;

int main() {

    int n;
    cin >> n;
    int numere[n];

    for (int i = 0; i < n; i++) {
        cin >> numere[i];
    }

    int contorPare = 0, contorImpare = 0;
    for(int i = 0; i < n; i++) {
        if (numere[i] % 2 == 0) {
            contorPare++;
        } else {
            contorImpare++;
        }
    }

    if (contorImpare > contorPare) {
        cout << contorImpare - contorPare;
    } else {
        cout << contorPare - contorImpare;
    }
}
```

```
        return 0;
    }
```

## 2. Exercițiu 2

- Link: <https://www.pbinfo.ro/probleme/546/afisare0>
- Soluție:

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int numere[n];
    for (int i = 0; i < n; i++) {
        cin >> numere[i];
    }

    int ultimulElement = numere[n-1];
    for (int i = 0; i < n; i++) {
        if (numere[i] % ultimulElement == 0) {
            cout << numere[i] << " ";
        }
    }

    return 0;
}
```

## 3. Exercițiu 3

- Link: <https://www.pbinfo.ro/probleme/4383/inlocuire6>
- Soluție:

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    int n, pozitieUltimPrim;
    cin >> n;
    int numere[n];
    for (int i = 0; i < n; i++) {
        cin >> numere[i];
    }

    // Varianta 1
    // Incepem de la primul numar si la fiecare prim gasit
```

```
// setam ultimulPrim=numarul respectiv
// asta inseamna ca la final, in variabila vom avea ultimul
// numar prim din sir
/*

for (int i = 0; i < n; i++) {
    int estePrim = 1;
    int numar = numere[i];
    for (int j = 2; j < numar/2;j++) {
        if (numar % j == 0) {
            estePrim = 0;
            break;
        }
    }
    if (estePrim) {
        pozitieUltimPrim = i;
    }
}
*/

// Varianta 2
// Incepem de la ultimul numar din vector si la
// primul numar prim gasit, setam variabila si iesim
// din loop.
for (int i = n-1; i >= 0; i--) {
    int estePrim = 1;
    int numar = numere[i];
    for (int j = 2; j < numar/2; j++) {
        if (numar % j == 0) {
            estePrim = 0;
            break;
        }
    }
    if (estePrim == 1) {
        pozitieUltimPrim = i;
        break;
    }
}

numere[pozitieUltimPrim] = 0;

for (int i = 0; i < n; i++) {
    cout << numere[i] << " ";
}

return 0;
}
```

#### 4. Exercițiu 4

- Link: <https://www.pbinfo.ro/probleme/4382/inlocuire5>
- Soluție:

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int numere[n];
    for (int i=0; i < n; i++) {
        cin >> numere[i];
    }

    for (int i = 0; i < n; i++) {
        int numar= numere[i];
        int estePrim = 1;
        for (int j = 2; j <= numar/2; j++) {
            if ( numar % j == 0) {
                estePrim = 0;
                break;
            }
        }
        if (estePrim == 1) {
            numere[i] = 0;
        }
    }

    for (int i = 0; i < n; i++) {
        cout << numere[i] <<" ";
    }

    return 0;
}
```

##### 5. Exercițiu 5:

- Link: <https://www.pbinfo.ro/probleme/488/afisare>
- Soluție:

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int numere[n];

    for (int i = 0; i < n; i++) {
        cin >> numere[i];
    }
}
```

```

//Parcurem pentru elementele cu indici pari in ordine
// crescatoare a indicilor
for (int i = 0; i < n; i++) {
    if ((i+1) % 2 == 0) {
        cout << numere[i] <<" ";
    }
}
cout << endl;
// Parcurem pentru elementele cu indici impari
// in ordine descrescatoare a indicilor
for (int i = n-1; i >=0; i--) {
    if ( (i+1) % 2 != 0) {
        cout << numere[i] <<" ";
    }
}
return 0;
}

```

## 6. Exercițiu 6

- Link: <https://www.pbinfo.ro/probleme/489/afisare1>
- Soluție:

```

#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int numere[n];
    for (int i = 0; i < n; i++) {
        cin >> numere[i];
    }

    for (int startIndex = 0, lastIndex = n-1; startIndex < lastIndex;
startIndex++, lastIndex--) {
        cout << numere[startIndex] <<" " << numere[lastIndex] << " ";
    }

    if ( n % 2 != 0) {
        cout << numere[n/2];
    }
    return 0;
}

```

## 7. Exercițiu 7

- Link: <https://www.pbinfo.ro/probleme/486/minmax0>



- Solutie:

```
#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    int numere[n];
    for (int i = 0; i<n;i++){
        cin >> numere[i];
    }
    int minim=numere[0], maxim=numere[0];
    for(int i = 1; i < n; i++) {
        if (numere[i] < minim) {
            minim = numere[i];
        }

        if (numere[i] > maxim) {
            maxim = numere[i];
        }
    }
    cout << minim << " " << maxim;
    return 0;
}
```

## 8. Exercițiu 8

- Link: <https://www.pbinfo.ro/probleme/553/pozminmax>
- Solutie:

```
#include <iostream>

using namespace std;
int main() {
    int n;
    cin >> n;
    int arr[n];
    for(int i =0; i < n; i++) {
        cin >> arr[i];
    }

    int maxPos = 0, minPos = 0;
    for(int i = 0; i < n; i++) {
        if(arr[i]> arr[maxPos]) {
            maxPos= i;
        }
        if(arr[i] < arr[minPos]) {
            minPos = i;
        }
    }
}
```

```
    cout <<minPos+1 << " " <<maxPos+1;

    return 0;
}
```

### 9. Exercițiu 9

- Link: <https://www.pbinfo.ro/probleme/547/numarare6>
- Soluție:

```
#include <iostream>
using namespace std;

// 1 citim elementele vectorului de la tastatura
// 2 aflam min/max
// 3 aflam diferenta max-min
// 4. Parcurgem vectorul si vedem cate elemente sunt egale cu diferenta

int main() {
    int n;
    cin >> n;
    int numere[n], minim, maxim, diferenta, contor = 0;
    for (int i = 0; i < n; i++) {
        cin >> numere[i];
    }
    minim = numere[0], maxim = numere[0];
    for (int i = 1; i < n; i++) {
        if (numere[i] < minim) {
            minim = numere[i];
        }
        if (numere[i] > maxim) {
            maxim = numere[i];
        }
    }

    diferenta = maxim - minim;
    for (int i = 0; i < n; i++) {
        if (numere[i] == diferenta) {
            contor++;
        }
    }
    cout << contor;
    return 0;
}
```

### 10. Exercițiu 10

- Link: <https://www.pbinfo.ro/probleme/490/afisareminmax>
- Soluție:

```
#include <iostream>
using namespace std;

// 1. Citim vectorul de la tastatura
// 2. Aflam pozitiile minim si maxim
// 3. Parcurgem vectorul intre [pozMinim, pozMaxim]

int main() {
    int n;
    cin >> n;
    int numere[n], pozitieMinim, pozitieMaxim, startInterval,
endInterval;
    for (int i = 0; i < n; i++) {
        cin >> numere[i];
    }
    pozitieMinim = 0, pozitieMaxim = 0;

    for (int i = 1; i < n; i++) {
        if (numere[i] < numere[pozitieMinim]) {
            pozitieMinim = i;
        }

        if (numere[i] > numere[pozitieMaxim]) {
            pozitieMaxim = i;
        }
    }

    if (pozitieMaxim > pozitieMinim) {
        startInterval = pozitieMinim;
        endInterval = pozitieMaxim;
    } else {
        startInterval = pozitieMaxim;
        endInterval = pozitieMinim;
    }

    for (int i = startInterval; i <= endInterval; i++) {
        cout << numere[i] << " ";
    }
    return 0;
}
```

## 11. Exercițiu 11

- Link: <https://www.pbinfo.ro/probleme/4151/amm>
- Soluție:

```
#include <iostream>
using namespace std;
```

```

// 1. Citim vectorul de la tastatura
// 2. Aflam pozitiile minim si maxim
// 3. Parcurgem vectorul intre [pozMinim, pozMaxim]

int main() {
    int n;
    cin >> n;
    int numere[n], pozitieMinim, pozitieMaxim, startInterval,
endInterval;
    for (int i = 0; i < n; i++) {
        cin >> numere[i];
    }
    pozitieMinim = 0, pozitieMaxim = 0;

    for (int i = 1; i < n; i++) {
        if (numere[i] < numere[pozitieMinim]) {
            pozitieMinim = i;
        }

        if (numere[i] > numere[pozitieMaxim]) {
            pozitieMaxim = i;
        }
    }

    if (pozitieMaxim > pozitieMinim) {
        startInterval = pozitieMinim;
        endInterval = pozitieMaxim;
    } else {
        startInterval = pozitieMaxim;
        endInterval = pozitieMinim;
    }

    for (int i = startInterval; i <= endInterval; i++) {
        cout << numere[i] << " ";
    }
    return 0;
}

```

## 12. Exercițiu 12

- Link: <https://www.pbinfo.ro/probleme/491/suma2>
- Soluție:

```

#include <iostream>
using namespace std;

// 1. Citim vectorul de la tastatura
// 2. Aflam pozitia primului numar par
// 3. Aflam pozitia ultimului numar par
// 4. Calculam suma elementelor intre primul si u ltimul numar par

```

```
inclusiv ele

int main() {
    int n;
    cin >> n;
    int numere[n], suma = 0, pozitiePrimPar=-1, pozitieUltimPar = -1;
    for (int i = 0; i < n; i++) {
        cin >> numere[i];
    }

    for (int i = 0; i < n; i++) {
        if (numere[i] % 2 == 0) {
            if (pozitiePrimPar == -1) {
                pozitiePrimPar = i;
            }
            pozitieUltimPar = i;
        }
    }

    if (pozitiePrimPar == -1) {
        cout << "NU EXISTA";
    } else {
        for (int i = pozitiePrimPar; i <= pozitieUltimPar; i++) {
            suma = suma + numere[i];
        }
        cout << suma;
    }

    return 0;
}
```

### 13. Exercițiu 13

- Link: <https://www.pbinfo.ro/probleme/487/numarare2>
- Soluție:

```
#include <iostream>
using namespace std;

// 1. Citim vectorul de la tastatura
// 2. Aflu media aritmetica a elementelor din vector
// 3. Numar cate elemente au valoarea > Ma

int main() {
    int n, suma = 0, contor = 0;
    float medieAritmetica;
    cin >> n;
    int numere[n];

    for (int i = 0; i < n; i++) {
```

```
        cin >> numere[i];
    }

    for (int i = 0; i < n; i++) {
        suma += numere[i];
    }
    medieAritmetica = suma / (n * 1.0);
    for (int i = 0; i < n; i++) {
        if (numere[i] > medieAritmetica) {
            contor++;
        }
    }

    cout << contor;

    return 0;
}
```

#### 14. Exercițiu 14

- Link: <https://www.pbinfo.ro/probleme/986/numarare7>
- Soluție:

```
#include <iostream>
using namespace std;

// 1. Citim vectorul de la tastatura
// 2. Aflam primul si ultimul element
// 3. Numaram cate numere sunt in afara intervalului [prim, ultimNumar]

int main() {
    int n;
    cin >> n;
    float numere[n];
    for(int i = 0; i < n; i++) {
        cin >> numere[i];
    }

    float primulNumar = numere[0];
    float ultimulNumar = numere[n-1];
    float startInterval, endInterval, contor = 0;
    if (primulNumar > ultimulNumar) {
        startInterval = ultimulNumar;
        endInterval = primulNumar;
    } else {
        endInterval = ultimulNumar;
        startInterval = primulNumar;
    }

    for (int i = 0; i < n; i++) {
```

```
        if ( numere[i] < startInterval || numere[i] > endInterval) {
            contor++;
        }
    }

    cout << contor;
    return 0;
}
```

### 15. Exercițiu 15

- Link: <https://www.pbinfo.ro/probleme/2858/pv>
- Soluție:

```
#include <iostream>

using namespace std;

int main()
{
    int n;
    cin >> n; // lungimea vectorului => cate elemente sunt in vector
    int numere[n];
    for (int i = 0; i < n; i++) {
        cin >> numere[i];
    }

    // afisare de la dreapta la stanga
    for (int i = n-1; i >= 0; i--) {
        cout << numere[i] << " ";
    }
    cout << endl;
    // suma valorilor pare din sir
    int suma = 0;
    for (int i = 0; i < n; i++) {
        if ( numere[i] % 2 == 0) {
            suma += numere[i];
        }
    }
    cout << suma << endl;

    // suma valorilor aflate pe pozitii pare in sir
    int sumaPozitiiPare = 0;
    for (int i = 0; i < n; i++) {
        // daca este pozitie para
        if ( (i+1) % 2 == 0) {
            sumaPozitiiPare += numere[i];
        }
    }
    cout << sumaPozitiiPare << endl;
}
```

```
// afisare numar de elemente divizibile cu 10;
int contorDiv10 = 0;
for (int i = 0; i < n; i++) {
    if (numere[i] % 10 == 0) {
        contorDiv10++;
    }
}

cout << contorDiv10 << endl;

// suma numerelor div cu 3 si aflate pe pozitii impare
int sumaDiv3PozImpare = 0;
for (int i = 0; i < n; i++) {
    if (numere[i] % 3 == 0 && (i+1) % 2 != 0) {
        sumaDiv3PozImpare += numere[i];
    }
}

cout << sumaDiv3PozImpare;
return 0;
}
```

## 16. Exercițiu 16

- Link: <https://www.pbinfo.ro/probleme/492/numarare3>
- Soluție:

```
#include <iostream>

using namespace std;
int gcd (int a, int b);
int main()
{
    int n;
    cin >> n;
    int numere[n];
    int contor = 0;
    for (int i = 0; i < n; i++)
    {
        cin >> numere[i];
    }

    // Varianta 1;
    for(int i = 0, j = n-1; i < j; i++,j--)
    {
        if (gcd(numere[i], numere[j]) == 1)
        {
            contor++;
        }
    }
}
```



```
        cout << contor;
        return 0;
    }

    // in engleza se zice GCD => greatest common divisor
    // in rommana se zice CMMDC => cel mai mare divizor comun
    int gcd (int a, int b)
    {
        while (b != 0)
        {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
}
```

### 17. Exercițiu 17

- Link: <https://www.pbinfo.ro/probleme/498/numararepie>
- Soluție:

```
#include <iostream>

using namespace std;
int gcd (int a, int b);
int main()
{
    int n;
    cin >> n;
    int numere[n];
    int contor = 0;
    for (int i = 0; i < n; i++)
    {
        cin >> numere[i];
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (j != i && gcd(numere[i], numere[j]) == 1) {
                contor++;
            }
        }
    }

    cout << contor/2;
    return 0;
}
```

```
// in engleza se zice GCD => greatest common divisor
// in rommana se zice CMMDC => cel mai mare divizor comun
int gcd (int a, int b)
{
    while (b != 0)
    {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}
```

### 18. Exercițiu 18

- Link: <https://www.pbinfo.ro/probleme/4381/numarareperechi2>
- Soluție:

```
#include <iostream>

using namespace std;
int main()
{
    int n;
    cin >> n;
    int numere[n];
    for (int i = 0; i < n; i++)
    {
        cin >> numere[i];
    }

    int contor = 0;
    for (int i = 0, j = n - 1; i < j; i++, j--) {
        int ultimaCifra1 = numere[i] / 10 % 10;
        int ultimaCifra2 = numere[j] / 10 % 10;
        if (ultimaCifra1 == ultimaCifra2) {
            contor++;
        }
    }

    cout << contor;

    return 0;
}
```

### 19. Exercițiu 19

- Link: <https://www.pbinfo.ro/probleme/4380/numarareperechi1>
- Solutie:

```
#include <iostream>

using namespace std;
int main()
{
    int n;
    cin >> n;
    int numere[n];
    for (int i = 0; i < n; i++)
    {
        cin >> numere[i];
    }

    int contor = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (i != j)
            {
                int ultimaCifra1 = numere[i] / 10 % 10;
                int ultimaCifra2 = numere[j] / 10 % 10;
                if (ultimaCifra1 == ultimaCifra2)
                {
                    contor++;
                }
            }
        }
    }

    // Impartim la 2 deoarece noi numaram atat perechea [a,b] cat si [b,
a]
    cout << contor/2;

    return 0;
}
```

## 20. Exercițiu 20

- Link: <https://www.pbinfo.ro/probleme/499/numarare5>
- Solutie:

```
#include <iostream>

using namespace std;
```

```
int sumaCifrelor(int numar);
int main()
{
    int n;
    cin >> n;
    int numere[n];
    for (int i =0; i < n; i++)
    {
        cin >> numere[i];
    }

    int contor = 0;
    for (int i =0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (i != j)
            {
                if ( sumaCifrelor( numere[i] ) == sumaCifrelor( numere[j] )
) {
                    contor++;
                }
            }
        }
    }

    // Impartim la 2 deoarece noi numaram atat perechea [a,b] cat si [b,
a]
    cout << contor/2;

    return 0;
}

int sumaCifrelor(int numar) {
    int suma = 0;
    while (numar > 0) {
        int ultimaCifra = numar % 10;
        suma += ultimaCifra;
        numar = numar /10;
    }
    return suma;
}
```