

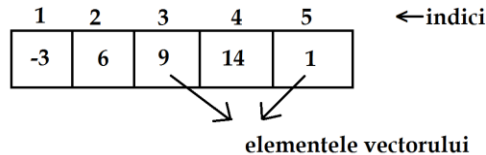
Vectori(tablouri unidimensionale)

- **Noțiunea de vector**

În practică, apare adesea necesitatea de a prelucra un set de valori de același tip, așezate într-o anumită ordine. O astfel de structură se numește **șir**, iar valorile respective se numesc **elementele șirului**.

Limbajul C/C++ oferă posibilitatea de a memora toate elementele unui șir într-o singură variabilă, în care *elementele sunt dispuse într-o anumită ordine și fiecare element ocupă o anumită poziție bine determinată*. O astfel de variabilă se numește **tablou unidimensional** sau **vector**.

Exemplu: Fie șirul de numere întregi – 3, 6, 9, 14, 1



Putem memora toate elementele șirului într-o variabilă numită **vector**, pe care o putem nota cu **v**. Elementele vectorului ocupă poziții succesive în vectorul **v**.

Valoare -3 este primul element din vector și pentru a ne referi la el vom spune **v[1]**. Valoare 6 este al doilea element din vector și pentru a ne referi la el vom spune **v[2]**.

Pe caz general, dacă variabila vector este **v**, atunci prin **v[i]** înțelegem elementul de pe poziția **i** în vector.

Observație: În C++ elementele vectorului pot începe de pe poziții de indice 0. De exemplu, fie vectorul de mai jos cu 6 elemente numere întregi(deci indicii vor fi de la 0 la 5):

i	0	1	2	3	4	5
v[i]	6	-3	5	0	2	7

- **Declararea unui vector**

Sintaxa declarării unui vector este:

tip identificador[DIM];

unde:

tip – tipul de date ordinar din care fac parte elementele vectorului(int, float, double, char, etc.)

identificador – numele variabilei asociat vectorului;

DIM – este numărul maxim de valori pe care vectorul le poate reține, care trebuie să fie un tip ordinal, cu un număr finit de valori.

Exemplu de declarații de vector:

int v[50];

Sarcină: Scrieți declarația unui vector **t** care să conțină maxim 25 de elemente, iar valorile pe care să le rețină acest vector să fie numere reale.

Observații:

În general, numărul de elemente precum și elementele vectorului se citesc de la tastatură.

Elementele unui vector se comportă ca niște variabile simple, deci pot fi citite, afișate și folosite în expresii.

Pentru exemplele de mai jos, explicați ce face fiecare secvență dată:

- cout<<v[2];
- x=(v[3] + v[5])/2;
- If (v[1]>0) cout<<"Primul element este pozitiv!";

- **Parcurea, citirea și afișarea unui vector**

Vom considera vectorul $v=(v[1],\dots,v[n])$, unde n este numărul de elemente citit de la tastatură

Prin parcurgerea vectorului se înțelege "vizitarea" elementelor pe rând și prelucrarea acestora.

Secvența de program folosită pentru **citirea** numărului de elemente a **vectorului** și a elementelor sale este:

```
int v[31];
cout<<"n="; cin>>n;
for(i=1;i<=n;i++) cin>>v[i];
```

Secvența folosită pentru **afișarea elementelor unui vector**(în ordine) este:

```
for(i=1;i<=n;i++) cout<<v[i];
```

Sarcină de lucru: Scrieți o secvență de program care să realizeze afișarea elementelor unui vector în ordine inversă.

Sarcină de lucru:

- Implementarea programului de parcurgere, citire și afișare a unui vector

Enunț: Se citește de la tastatură un număr natural în variabila n , reprezentând numărul de elemente a unui vector de tip întreg(cu maxim 30 de elemente). Scrieți un program C/C++ care citește elementele vectorului, apoi afișează elementele acestuia atât de la stânga la dreapta, dar și invers.

Exemplu: Fie $n=6$ și valorile 7, -1, 3, 6, 3, 5. Programul afișă: 7, -1, 3, 6, 3, 5 și 5, 3, 6, 3, -1, 7.

Probleme folosind vectori(tablouri unidimensionale)

1. **(Elemente de indice par, elemente de indice impar)** Se citește un număr natural nenul $n(1 \leq n \leq 30)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere reale, apoi afișează întâi elementele din șir de indice par și apoi elementele din șir de indice impar.
2. **(Câte elemente pare, câte elemente impare)** Se citește un număr natural nenul $n(1 \leq n \leq 30)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 4 cifre fiecare, apoi determină și afișează câte numere pare și câte numere impare există în șirul citit inițial. Dacă șirul citit inițial e format:
 - doar din numere pare atunci se va afișa mesajul „Nu există elemente impare”
 - doar din numere impare atunci se va afișa mesajul „Nu există elemente pare”
3. **(Câte elemente negative, câte elemente pozitive, câte zerouri)** Se citește un număr natural nenul $n(1 \leq n \leq 30)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 9 cifre fiecare, apoi determină și afișează câte numere negative, câte numere pozitive și câte valori de zero există în șirul citit inițial.
4. **(Elemente divizibile cu p și indicii acestor elemente)** Se citește un număr natural nenul $n(1 \leq n \leq 30)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi de maxim 9 cifre fiecare, apoi determină și afișează, separate prin câte un spațiu, elemente din vector care sunt divizibile cu p și pozițiile pe care acestea se găsesc în șir. Dacă nu există nici un element în șir divizibil cu p atunci se va afișa mesajul „NU EXISTĂ”.
5. **(Media aritmetică a elementelor)** Se citește un număr natural nenul $n(1 \leq n \leq 30)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere reale, apoi determină și afișează media aritmetică a elementelor șirului citit inițial.
6. **(Media aritmetică elemente pare și elemente impare)** Se citește un număr natural nenul $n(1 \leq n \leq 30)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 4 cifre fiecare, apoi

determină și afișează media aritmetică a elementelor pare și media aritmetică a elementelor impare din șirul citit inițial. Dacă șirul citit inițial e format:

- doar din numere pare atunci se va afișa mesajul „Nu există elemente impare”
- doar din numere impare atunci se va afișa mesajul „Nu există elemente pare”

7. (**Suma elemente pare, suma elemente impare**) Se citește un număr natural nenul $n(1 \leq n \leq 30)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 4 cifre fiecare, apoi determină și afișează suma elementelor pare și suma elementelor impare din șirul citit inițial. Dacă șirul citit inițial e format:
- doar din numere pare atunci se va afișa mesajul „Nu există elemente impare”
 - doar din numere impare atunci se va afișa mesajul „Nu există elemente pare”
8. (**Minimul într-un vector**) Se citește un număr natural nenul $n(1 \leq n \leq 50)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 9 cifre fiecare, apoi determină și afișează minimul din șirul de elemente citit inițial.
9. (**Nr. de apariții a minimului într-un vector**) Se citește un număr natural nenul $n(1 \leq n \leq 50)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 9 cifre fiecare, apoi determină și afișează numărul de apariții a minimului în șirul de elemente citit inițial.
10. (**Nr. de apariții a maximului într-un vector**) Se citește un număr natural nenul $n(1 \leq n \leq 50)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 9 cifre fiecare, apoi determină și afișează numărul de apariții a maximului în șirul de elemente citit inițial.
11. (**Minimul și poziția minimului într-un vector**) Se citește un număr natural nenul $n(1 \leq n \leq 50)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 9 cifre fiecare, apoi determină și afișează minimul și poziția minimului în șirul de elemente citit inițial.
12. (**Maximul și poziția maximului într-un vector**) Se citește un număr natural nenul $n(1 \leq n \leq 50)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 9 cifre fiecare, apoi determină și afișează maximul și poziția maximului în șirul de elemente citit inițial.
13. (**Cel mai mare număr negativ dintr-un vector**) Se citește un număr natural nenul $n(1 \leq n \leq 50)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 9 cifre fiecare, apoi determină și afișează cel mai mare număr negativ din șirul de elemente citit inițial.
14. (**Cel mai mic număr pozitiv dintr-un vector**) Se citește un număr natural nenul $n(1 \leq n \leq 50)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 9 cifre fiecare, apoi determină și afișează cel mai mic număr pozitiv din șirul de elemente citit inițial.
15. (**Interschimbarea minimului cu maximul**) Se citește un număr natural nenul $n(1 \leq n \leq 50)$ de la tastatură. Scrieți un program C++ care citește un șir de n elemente numere întregi, de maxim 9 cifre fiecare, apoi interschimbă valoarea minimului cu valoarea maximului, apoi afișează vectorul rezultatul în urma interschimbării.