

## Tipuri structurate de date în C++.

### Structuri de date pentru memorarea șirurilor de numere (*VECTORI*)

#### Ce este un vector și când este necesară reprezentarea datelor cu ajutorul lui?

În foarte multe probleme apare necesitatea de a prelucra repetat un șir cu multe valori de același tip, care nu pot fi denumite prin variabile obișnuite (simple), datorită numărului prea mare de elemente.

Exemplu:

- 1) La supermarket, trebuie analizate un șir de  $n$  prețuri de produse ( $n < 10000$ ) codificate cu numere de la 1 la  $n$ . Produselor cu prețul maxim trebuie să li se aplice o reducere de 50%. Se va afișa prețul final și codul produselor pentru care a fost aplicată reducerea.
- 2) Rezultatele celor  $n$  candidați la un concurs ( $n < 300$ ) trebuie afișate în ordinea descrescătoare a punctajelor obținute.

VECTOR = ansamblu de valori de același tip, depuse în memorie pe poziții consecutive și recunoscute sub nume unic.

Pozițiile pe care sunt memorate valorile sunt distincte și numerotate crescător de la 0 la  $n-1$  și nu pot fi modificate de către utilizator. Aceste poziții se numesc **indici**.

**Valorile componentelor din vector** nu coincid obligatoriu cu pozițiile pe care se situează și pot fi modificate prin program.

De exemplu, acceleratul A106 este un tren cu vagoane de călători (componentele vectorului, toate la fel), numerotate de la 1 la 10 (indicii vagoanelor). În fiecare vagon se află un număr de călători la un moment dat. Trenul nu-și modifică structura dar numărul de călători din fiecare vagon se schimbă în fiecare stație în care urcă și coboară călători.

#### Declararea unui vector. Referirea unei componente.

**tip\_valori nume\_structura[nr\_max\_elemente+1];**

ex: int a[101];

//declararea unui vector **a** cu cel mult 101 numere intregi, numerotate de la 0 la 100

	75	22	1	24	1234	.....	a[i]		a[100]
0	1	2	3	4	5		i		100

- 0,1, 2,...i se numesc indici (indica pozitia valorii in sir)
- 75, 22, 1,24,1234,....., **a[i]** sunt valorile efective de pe pozitiile date (ex: numărul de cuvinte de pe pagina de carte numarul i, numărul de călători din vagonul i, prețul produsului cu codul i etc.)
- a[i] = valoarea de pe pozitia i (referirea componenteii i)

## Operații specifice

### 1. Citire (doar prin parcurgerea componentă cu componentă)

```
int a[101], n, i;
fin>>n;
for (i=1; i<=n; i++)
    fin>>a[i];
```

### 2. Scriere (doar prin parcurgerea componentă cu componentă)

#### - de la stânga la dreapta

```
for (i=1; i<=n; i++)
    fout<<a[i]<<' ';
```

ex: pe sirul de mai sus, pentru  $n=5$  se va afișa  
**75 22 1 24 1234**

#### - de la dreapta la stânga

```
for (i=n; i>=1; i--)
    fout<<a[i]<<' ';
```

ex: pe sirul de mai sus, pentru  $n=5$  se va afișa  
**1234 24 1 22 7**

### 3. Parcurgere (în scopul prelucrării valorilor din vector)

```
for (i=1; i<=n; i++)
    prelucrare (a[i]);
```

ex: determinarea minimului unui șir de numere

```
int mini;
....
mini=a[1];
for (i=2; i<=n; i++)
    if (a[i]<mini) mini=a[i];
....
```



## Vectori – APLICAȚII

1. Se citesc cele  $n$  numere naturale ( $n < 100$ ). Afișați câte numere din șir sunt divizibile cu ultimul element al șirului.
2. Gigel s-a documentat și a făcut o listă cu prețurile ultimelor  $n$  smartphone-uri apărute și altă listă cu prețurile celor mai vandute  $m$  gadgeturi. Ajutați-l să determine cel mai scump smartphone și cel mai ieftin gadget. Dacă sunt mai multe, sa afișeze toate pozițiile din listele corespunzătoare.
3. Date fiind un șir cu  $n$  numere naturale ( $n < 1000$ ) și un  $0 < k < n$ , calculați și afișați produsul ultimelor  $k$  numere diferite de 0, dacă există. Dacă nu sunt  $k$  astfel de numere în șir, afișați mesajul IMPOSIBIL.

4. Duda are 2 albume cu poze pe Facebook. In primul album are  $n$  poze, în cel de al doilea are  $m$  poze. Fiecare poză are un anumit numar de like-uri. Calculați câte poze din primul album au mai puține like-uri decât fiecare poză din cel de-al doilea și care sunt acestea. Pozele sunt numerotate cu numere de la 1 la  $n$  respectiv de la 1 la  $m$  ( $1 < n, m < 50$ ).

**Exemplu:**

```

primul album:      5                //5 poze in primul album (de la 1 la 5)
                  3 6 18 5 9        //numarul de like-uri pentru fiecare
al doilea album:  7                //5 poze in primul album (de la 1 la 7)
                  13 9 10 22 8 7 12 // numarul de like-uri pentru fiecare
Se va afișa:      3
                  1  2  4

```

adica: sunt 3 poze in primul album care au mai putine like-uri decât orice poză din al doilea iar pozele căutate sunt prima (3 like-uri), a doua (6 like-uri) și a patra (5 like-uri).

Scrieți câte un program C++ pentru fiecare dintre cerințele de mai jos.

5. Se citește de la tastatură  $n$ , un număr natural nenul,  $n < 100$ , apoi se citesc cele  $n$  elemente ale vectorului  $a$ . Să se afișeze pe ecran:
- Pozițiile din vector pe care se află numere divizibile cu 5
  - Elementele vectorului care au ultimele două cifre egale, în ordinea inversă apariției în vector
  - Numărul de valori egale cu 0 din vectorul dat
  - Suma numerelor de exact 3 cifre din vectorul dat
  - Numererele care sunt egale cu media aritmetică a elementelor din vector
  - Perechile de numere alăturate din vector care au aceeași paritate (sunt ambele pare sau ambele impare)
6. Se citește de la tastatură  $n$ , un număr natural nenul,  $n < 100$ , apoi se citesc cele  $n$  elemente ale vectorului  $a$ . Să se afișeze pe ecran:
- produsul numerelor mai mari decât 100
  - numărul de valori negative
  - numerele care sunt egale cu media aritmetică a vecinilor lor (numărul din stânga+numărul din dreapta lui)/2
  - mesajul DA, dacă toate valorile sunt în ordine crescătoare și NU altfel
7. Se citesc din fișierul text **medii.in**  $n$ , numărul de elemente ( $n < 40$ ) apoi cele  $n$  medii la informatică ale unei clase de elevi  $a_1, a_2, \dots, a_n$ . Să se scrie în fișierul **medii.out**:
- Indicii elevilor cu media 10
  - Numărul de elevi corigenți (cu media < 5)
  - Media clasei
  - Media cea mai mică și media cea mai mare din clasă
8. Se citesc din fișierul text **produse.in**,  $S$ , o sumă de bani disponibilă,  $n$ , numărul de produse ( $n < 1000$ ) apoi cele  $n$  prețuri ale unor produse  $p_1, p_2, \dots, p_n$  și cantitățile în care sunt disponibile  $c_1, c_2, \dots, c_n$ . Să se scrie în fișierul **produse.out**:
- Mesajul **DA**, dacă toate produsele pot fi cumpărate cu suma  $S$  și **NU**, altfel
  - Cel mai mare număr de produse la fel care pot fi cumpărate cu suma dată
  - Cele mai ieftine produse și numărul lor
  - Lista produselor în ordinea crescătoare a prețurilor și, la același preț, în ordinea descrescătoare a cantităților