

## TABLOURI UNIDIMENSIONALE (VECTORI)

### 1. Declarația / definiția unui tablou

Tabloul este o mulțime ordonată de elemente la care ne putem referi cu ajutorul unor indici. Orice tablou are un nume, iar tipul comun al elementelor sale este tipul tabloului. În limbajul **C** există doar tablouri unidimensionale. Tablourile multidimensionale sunt considerate tot tablouri unidimensionale în care fiecare element este la rândul său un tablou. Declarația (care coincide cu definiția) unui tablou unidimensional este de forma:

**T** nume[dim];

unde :

**T** este unul din tipurile existente în limbaj sau definite de programator  
nume este numele tabloului  
dim este o valoare **constantă** care reprezintă numărul de elemente din tablou.

Orice declarație de tablou se termină cu ;. **După declarare, dacă nu au fost inițializate, elementele tabloului au valori nedefinite.**

La definirea unui tablou se alocă o zonă de memorie necesară pentru a memora elementele tabloului. La primul element ne vom referi prin nume[0], la al doilea prin nume[1], ..., la ultimul prin nume[dim-1], deci indicele de parcurgere a tabloului variază între 0 și dim-1.

Determinarea numărului de octeți rezervați tabloul identificat prin **nume** se face prin utilizarea expresiei:

sizeof(nume)

Rezultatul acestei expresii trebuie să coincidă cu valoarea dată de expresia:

dim \* sizeof(**T**)

#### Exemplu L4.1.

Definirea unui tablou unidimensional cu 10 elemente de tip întreg cu semn al cărui nume este **a**.

int a[10];

Pentru acest tablou se vor rezerva, la compilare, 10\*sizeof(int) octeți.

sizeof(a) va avea același rezultat.

Se pot declara mai multe tablouri cu același tip de elemente pe aceeași linie, caz în care se folosește separatorul virgulă.

#### Exemplu L4.2.

Definirea a trei tablouri cu elemente de tip real simplu (float).

float a[15], b[5], vector[20];

Se recomandă definirea unui singur tablou pe o linie pentru a da posibilitatea inserării de comentarii pentru buna documentare a programului.

## 2. Folosirea tablourilor ca parametri de intrare în funcții

Tablourile pot fi folosite ca parametri de intrare în funcții, dar funcțiile nu pot avea tipul tablou (nu pot returna tablouri).

La apelul unei funcții, programul apelant trimite către funcție adresa de început a zonei de memorie în care se găsește vectorul, iar orice operație cu elementele unui vector se face direct asupra locației de memorie corespunzătoare. Ca urmare, orice modificare făcută asupra unui element al unui tablou (unidimensional sau multidimensional) se păstrează și după ieșirea din funcție.

**Pentru rezolvarea corectă a problemelor ce urmează citiți cu ATENȚIE enunțul fiecărei probleme de la început până la sfârșit. Rezolvările se fac fără variabile globale.**

## TEMA 1

### Problema 1.1.

Să se scrie o funcție care citește un șir de  $n$  numere reale în simplă precizie (numerele sunt memorate într-un vector al cărui nume este **x1**).

Să se scrie o funcție care afișează  $n$  elemente dintr-un vector de numere reale în simplă precizie.

Să se scrie o funcție (numită **fcnt1**) care primește ca parametri de intrare un număr real în simplă precizie și un număr întreg. Funcția nu returnează nimic și face următoarele prelucrări: împarte numărul real la 3 și crește numărul întreg cu 2.

Să se scrie un program care:

- citește de la tastatură un număr întreg fără semn  $n$  (cel mult egal cu 10);
- un șir de  $n$  numere reale în simplă precizie stocate într-un vector (folosind o funcție de citire a unui vector);
- afișează vectorul citit (folosind o funcție de afișare);
- afișează valorile celui de-al doilea element din vector și a lui  $n$ ;
- apelează funcția **fcnt1** care primește ca parametri: al doilea element din vector și  $n$ ;
- afișează din nou valorile celui de-al doilea element din vector și a lui  $n$ ;
- apelează o funcție **fcnt2** (care trebuie și ea scrisă) care are ca parametri vectorul și valoarea  $n$ .

Funcția împarte al doilea element din vector la 3 și crește pe  $n$  cu 2.

- afișează din nou valorile celui de-al doilea element din vector și a lui  $n$ ;
- afișează din nou vectorul.

Discuție. Ce se observă?

### Problema 1.2.

Într-o stație meteorologică se măsoară, în fiecare zi la ora 7, temperatura la 1,5 m deasupra solului. La anumite intervale de timp, măsurate în zile, se realizează rapoarte care cuprind: temperatura minimă, temperatura maximă și temperatura medie din intervalul de timp considerat, precum și media geometrică a valorilor absolute ale temperaturilor măsurate în intervalul de timp dat (această valoare va fi folosită apoi în alte prelucrări necesare pentru o analiză meteorologică completă).

Intervalul de timp pentru care se fac prelucrările poate fi de maxim 31 de zile (raportul se face numai pentru zile din aceeași lună).

Să se scrie un program care citește informațiile necesare de la tastatură și generează raportul cerut (ca în exemplul de mai jos) afișat pe ecran.

Valorile temperaturilor măsurate (se iau în considerare valori întregi pentru temperaturile măsurate) se stochează într-un vector, iar, pentru individualizarea prelucrărilor, în raport se va indica și luna pentru care se fac prelucrările.

*Indicație: Pentru media geometrică se va folosi funcția **pow**, iar pentru determinarea valorii absolute se va folosi funcția **abs**.*

Se vor scrie funcții adecvate pentru citirea unui vector, scrierea unui vector, calcularea valorii maxime a elementelor unui vector, calcularea valorii minime a elementelor unui vector, calcularea mediei aritmetice a valorilor elementelor unui vector, calcularea mediei geometrice a valorilor elementelor unui vector.

**Exemplu de raport care trebuie generat de program** (textul scris cu **albastru** reprezintă mesajele scrise de program, textul scris cu **verde** reprezintă valorile datelor de intrare, iar textul scris cu **roșu** reprezintă valorile rezultatelor calculate de program).

```
Raport de temperatura pentru ultimele 3 zile din luna februarie.  
Temperaturile citite sunt: -7, 1, -5.  
    Valoarea maxima a temperaturii este: 1.  
    Valoarea minima a temperaturii este: -7.  
    Valoarea medie a temperaturii este: -3.667.  
    Media geometrica a temperaturilor este 3.2711.  
  
Final raport.
```

*Atenție:* în acest exemplu culorile sunt folosite doar a evidenția de unde provin informațiile citite. Programul nu va folosi culori diferite pentru scrierea raportului și nici nu va genera un chenar.

### Problema 1.3.

Firma "Trioda" comercializează componente electronice și are două magazine în **Pădurea cu alune**. Cele două magazine se numesc **Trioda1** și **Trioda2** și comercializează aceleași sortimente de componente electronice. La sfârșitul fiecărei luni se face un inventar pentru a determina valoarea stocurilor de componente electronice și valoarea totală a mărfii. După inventar trebuie generat un raport ca în exemplul următor (convențiile de culoare sunt cele de la problema 1.1):

```
Raport inventar pentru firma Trioda.
```

```
Firma Trioda comercializează 5 tipuri de componente electronice.
```

```
Pret/bucata pentru fiecare componenta electronica = (125.3, 3.15, 574, 98.35, 4.25)
```

```
Stocuri magazin Trioda1 = (23, 675, 12, 5, 170)
```

```
Stocuri magazin Trioda2 = (12, 25, 3, 25, 30)
```

```
Total stocuri firma = (35, 700, 15, 30, 200)
```

```
Valoare stocuri magazin Trioda1 = (2881.90, 2126.25, 6888.00, 491.75, 722.50)
```

```
Valoare stocuri magazin Trioda2 = (1503.60, 78.75, 1722.00, 2458.75, 127.50)
```

```
Valoare stocuri firma = (4385.50, 2205.00, 8610.00, 2950.50, 850.00)
```

```
Valoarea totală marfă = 19001.00 lei.
```

```
Sfarsit raport inventar.
```

Să se scrie un program care citește informațiile necesare și

- generează un raport de inventar pentru firma **Trioda** după modelul dat;
- calculează valoarea stocurilor firmei și valoarea totală a mărfii după o lună de la inventarul inițial considerând că stocurile au rămas aceleași (ceea ce s-a vândut a fost imediat completat), dar toate prețurile au crescut cu o anumită valoare procentuală, valoare care va fi citită de la tastatură (de exemplu, creșterea prețurilor a fost de 0.7%).

Firma **Trioda** poate avea pe stoc la un moment dat maxim 50 de tipuri de componente electronice.

Se vor scrie funcții adecvate pentru: citirea unui vector, afișarea unui vector (conform modelului dat), înmulțirea a doi vectori, suma a doi vectori, suma valorilor elementelor a doi vectori, înmulțirea unui vector cu un scalar.

Datele de intrare se citesc de la tastatură, iar rezultatele (raportul de inventar inițial și noile valori ale stocurilor) vor fi afișate pe ecran.

#### Problema 1.4.

Să se scrie un program care citește un tablou de maximum 30 de numere întregi, afișează pe monitor vectorul citit, ordonează crescător, prin metoda bulelor, elementele vectorului și afișează rezultatul pe monitor. Se vor scrie funcții pentru citirea unui vector de numere întregi, scrierea unui vector de numere întregi și ordonare prin metoda bulelor. (*Observație se pot folosi funcțiile scrise în problemele anterioare*).

Schema logică a programului este dată în **Figura 1**, iar schema logică a funcției pentru sortarea, în ordine crescătoare, a elementelor vectorului folosind metoda bulelor este dată în **Figura 2**.

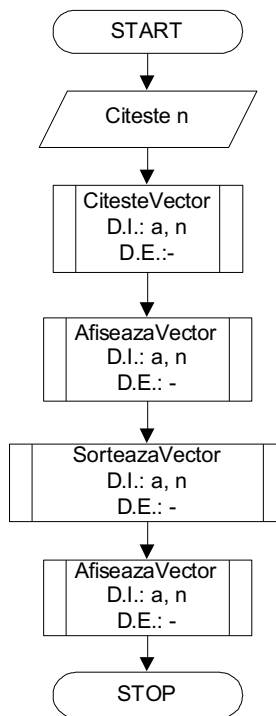


Figura 1. Schema logică a programului de ordonare în sens crescător a elementelor unui vector

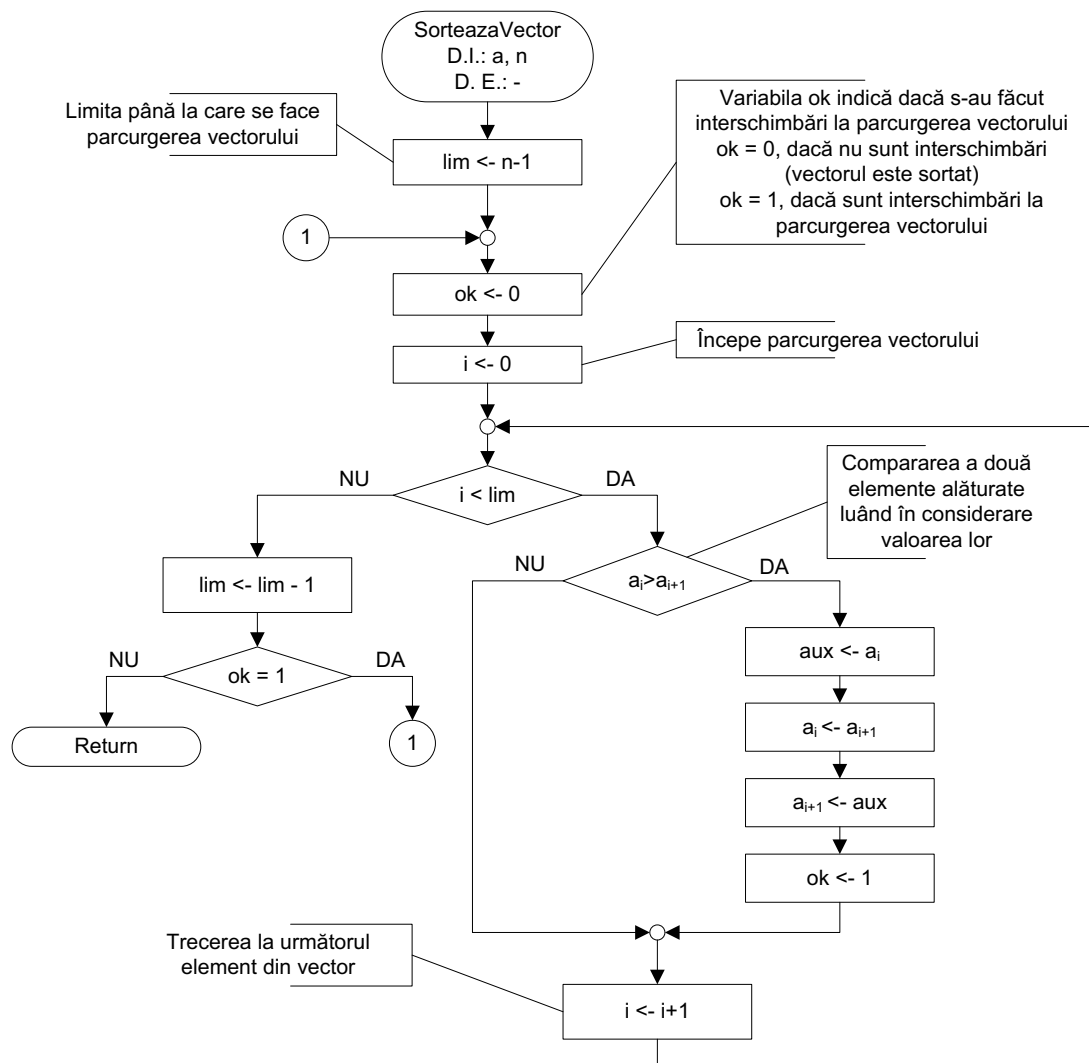


Figura 2. Schema logică a funcției de ordonare în sens crescător a elementelor unui vector

## TEMA 2

### Problema 2.1.

În urma unor măsurători făcute în cadrul unui experiment rezultă un șir de  $n$  date experimentale (de tip real) care se memorează într-un vector (tablou unidimensional)  $\mathbf{x}$ . Prelucrarea acestor date experimentale înseamnă:

- a. calcularea valorii medii cu formula

$$x_m = \frac{1}{n} \sum_{i=0}^{n-1} x_i$$

- b. calcularea abaterii medii pătratice cu formula:

$$x_p = \sqrt{\frac{\sum_{i=0}^{n-1} (x_i - x_m)^2}{n(n-1)}}$$

- c. afișarea numărului de componente care nu depășesc valoarea medie

- d. crearea unui alt vector  $\mathbf{y}$  cu elementele din  $\mathbf{x}$  mai mari decât valoarea medie și afișarea vectorului  $\mathbf{y}$  cu câte 5 elemente pe o linie.

Să se scrie un program pentru a putea prelucra datele experimentale obținute. Pentru rezolvarea problemei se vor folosi funcții pentru:

1. citirea unui șir de numere întregi, cu prototipul:

```
int citesteVectorReal2(double a[]);
```

```
// funcția returnează numărul de elemente efectiv citite
```

2. afișarea unui șir de numere, cu prototipul:

```
void scrieVectorReal2(double a[], int n);
```

3. fiecare din cele 4 tipuri de prelucrări (câte o funcție pentru fiecare prelucrare). Aceste funcții vor returna rezultatul calculat, scrierea rezultatelor făcându-se din funcția **main**.

Pentru punctul **d** se vor scrie două funcții: una pentru crearea vectorului  $\mathbf{y}$  având ca date de intrare vectorul  $\mathbf{x}$ , numărul de elemente din  $\mathbf{x}$  și vectorul  $\mathbf{y}$  și ca date de ieșire numărul de elemente din  $\mathbf{y}$  și cea de a doua pentru scrierea unui vector cu câte 5 elemente pe o linie.

Prototipurile funcțiilor se vor găsi într-un fișier header după modelul dat la curs.

Datele vor fi citite de la tastatură, iar rezultatele vor fi scrise pe monitor.

### Problema 2.2.

Se citește de la tastatură un șir de cel mult 100 numere întregi, până la întâlnirea valorii **0**. Șirul de numere se memorează într-un tablou.

Să se afișeze elementele distincte.

Exemplu: în șirul **2 2 5 4 5 1 2** elementele distincte sunt **2 5 4 1**.

Se vor scrie funcții pentru:

1. citirea unui șir de numere, cu prototipul:

```
int citireVectorIntreg(int a[]);
```

```
// funcția returnează numărul de elemente efectiv citite
```

2. afișarea unui șir de numere, cu prototipul (de la problema anterioară):

```
void afisareVectorIntreg(int a[], int n);
```

3. determinarea șirului cu elemente distincte, cu prototipul:

```
int distinct(int a[], int b[], int n);
// funcția are ca parametri de intrare șirul inițial (a), șirul final (cu elemente distincte b) și
numărul de elemente din șirul inițial n și returnează numărul de elemente din șirul final.
```

Prototipurile funcțiilor se vor găsi într-un fișier header după modelul dat la curs.

### Problema 2.3.

În cadrul unui laborator de analize medicale se fac măsurători care privesc modul în care evoluează în timp concentrația unei anumite componente în proba de laborator analizată. Pentru aceasta se notează pe o fișă concentrația și momentul de timp corespunzător (considerând momentul de început al analizei ca moment zero). Se obțin astfel două șiruri de valori: (i) șirul cu valorile momentelor de timp și (ii) șirul cu valorile concentrațiilor la aceste momente de timp.

Se cere să se scrie un program care citește cele două șiruri de valori (care sunt numere reale) și afișează coeficienții dreptei de regresie sau afișează un mesaj corespunzător dacă aceștia nu pot fi calculați.

Dreapta de regresie  $x = a \cdot t + b$  are proprietatea că pentru punctele din plan valoarea expresiei

$$E = \sum_{i=0}^{n-1} (x_i - a \cdot t_i - b)^2$$

este minimă, unde  $n$  este numărul de măsurători făcute.

Coeficienții  $a$  și  $b$  se determină prin rezolvarea sistemului:

$$\begin{cases} n \cdot a + b \cdot \sum_{i=0}^{n-1} (t_i) = \sum_{i=0}^{n-1} (x_i) \\ a \cdot \sum_{i=0}^{n-1} (t_i) + b \cdot \sum_{i=0}^{n-1} (t_i \cdot t_i) = \sum_{i=0}^{n-1} (t_i \cdot x_i) \end{cases}$$

Pentru rezolvarea problemei se vor scrie următoarele funcții:

- funcție pentru citirea unui vector
- funcție pentru afișarea unui vector

(Aceste două funcții sunt cele scrise pentru rezolvarea problemei 1)

- funcție pentru rezolvarea unui sistem de două ecuații cu două necunoscute. Această funcție are doi parametri:

- un tablou cu 6 elemente conținând coeficienții celor două ecuații;
- un tablou cu două elemente conținând soluțiile sistemului

și returnează rezultatul:

- 1 – dacă sistemul este compatibil determinat
- 2 – dacă sistemul este compatibil nedeterminat
- 3 – dacă sistemul este incompatibil

d. funcție care calculează produsul scalar a doi vectori  $a$  și  $b$ . Funcția are ca parametri cei doi vectori și numărul de elemente al unui vector și returnează valoarea produsului scalar. Pentru calculul sumei valorilor elementelor unui vector folosind funcția produs scalar unul din vectori va fi un vector inițializat cu unități.

Observație:

Toate sumele se vor calcula folosind funcția produs scalar.

Prototipurile funcțiilor se vor găsi într-un fișier header după modelul dat la curs.