

Laborator 3 - Tablou unidimensional - Vector

În limbajul C, tablourile de date reprezintă colecții de elemente de același tip care pot fi identificate prin unul sau mai mulți indecși (chei). Tablourile ocupă un spațiu contiguu de memorie.

Tablourile unidimensionale, denumite și vectori, au proprietatea că elementele acestora sunt identificate printr-un singur index, numit și poziție sau cheie. Vectorii au următoarele caracteristici:

1. identificatorul vectorului (denumirea variabilei de tip vector)
2. tipul de dată al elementelor
3. dimensiunea vectorului (numărul maxim de elemente).

De ce avem nevoie de vectori în programare?

Scenariu: trebuie să se calculeze media aritmetică a 100 de numere întregi introduse de utilizator.

În limbajul C, există două modalități de a face acest lucru:

1) Definim 100 de variabile cu tip de date `int` și apoi efectuăm 100 de operații de citire, pentru a stoca valorile introduse în variabile și apoi calculăm media acestora.

2) Avem un singur vector de 100 valori tip `int` pentru a stoca datele. Într-o instrucțiune repetitivă citim valorile pe care le stocăm în vector și apoi calculăm media.

Ce soluție este mai bună?

Cele trei aspecte de mai sus sunt definite la declararea vectorilor

tip_data identificator [dimensiune] ;

Indecșii sunt numere naturale consecutive de la 0 la (dimensiune vector - 1).

De exemplu, în urma declarării de mai jos:

```
float v[100];
```

se rezervă în memorie 100 de spații consecutive de memorie de mărimea `sizeof(float)`. Blocul de memorie se asociază variabilei `v`. Elementele vectorului `v` sunt `v[0]`; `v[1]`; ... ; `v[99]`.

Exemplul 3.1. Să se citească de la tastatură `n` numere întregi, cu $1 < n < 100$ și să se memoreze într-un vector. Calculați suma elementelor din vector.

```

1  #include <stdio.h>
2  int main()
3  {
4      int i, n, v[100], suma;
5      printf("Introdu n=");
6      scanf("%d", &n);
7      for(i=0; i<n; i++)
8      {
9          printf("v[%d] = ", i);
10         scanf("%d", &v[i]);
11     }
12     suma = 0;
13     for(i=0; i<n; i++)
14         suma = suma + v[i];
15     printf("Suma elementelor vectorului v = %d\n", suma);
16     return 0;
17 }
```

Exercițiul 3.2. Să se citească de la tastatură `n` numere întregi, cu $1 < n < 50$ și să se memoreze într-un vector. Calculați produsul elementelor aflate pe poziții pare în vector.

Exercițiul 3.3. Să se scrie un program care citește de la tastatură un șir de numere nenule (citirea se termină la introducerea primului 0), care vor fi stocate într-un vector, având dimensiunea maximă 50. Verificați dacă:

- (a) șirul este strict crescător
- (b) șirul este strict descrescător
- (c) șirul este o progresie aritmetică
- (d) șirul este o progresie geometrică
- (e) șirul conține toate valorile identice
- (f)* determinați care valoare nu respecta regula șirului. Spre exemplu, fie șirul: 1,1,2,2,4,4,5,5,6,7,7,8,8. Elementul cu valoarea 6 nu respecta regula de formare a șirului.

Exercițiul 3.4. Căutare secvențială. Fiind dat un șir de `n` numere întregi memorate într-un vector și o valoare `x`, să se verifice dacă `x` aparține vectorului și, în caz afirmativ, să se afișeze prima poziție pe care a fost găsit. Modificați programul pentru a afișa ultima poziție pe care este găsită valoarea de căutat `x`.

Nota: Exercițiile marcate cu * sunt optionale

Last modified: Tuesday, 18 October 2022, 1:34 PM

«	PREVIOUS ACTIVITY C3-Vector (Matrice)	NEXT ACTIVITY Tema de casa - Laborator 3	»
---	--	---	---

[Get the mobile app](#)