Preluat din:

https://sites.google.com/site/elearningalinabonto/tablouri-unidimensionale---vectori/1-tablouri-unidimensionale---vectori

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1. Tablouri unidimensionale - vectori**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | ***Definiţie*** Tabloul unidimensional este o structură de date căreia i se atribuie un nume. Este format dintr-o colecție de elemente de același tip, dispuse contiguu într-un bloc de memorie. Elementele pot fi accesate individual prin indici sau ca un tot unitar. Toate elementele au un predecesor (excepție primul) și un succesor (excepție ultimul)  ***Declarare***  tip nume\_tablou [dimensiune\_max];  tip - precizează tipul datelor (întregi, real, caracter, etc.)  nume\_tablou – identificator, precizează numele dat tabloului  Dimensiune max – numărul maxim de componente (o constantă întreagă)  Dimensiune max = memoria fizică alocată. Dimens. logică ≤ dimens.max  Se observă că este obligatorie folosirea parantezelor drepte care să încadreze dimensiunea maximă pe care o alege utilizatorul pentru acel tablou unidimensional.  *Exemple*  int a[25];   // declararea unui tablou unidimensional cu maxim 25 de elemente, fiecare de tip întreg  float x[30];   // declararea unui tablou unidimensional cu maxim 30 de elemente, fiecare de tip real simplă precizie  char s[40];  // declararea unui tablou unidimensional cu maxim 40 de elemente, fiecare de tip caracter  int a[99], b[99] ;  // au fost declarați doi vectori a, b  Un element al unui tablou poate fi utilizat ca orice alta variabilă. Adresarea unei componente se face prin indicele ei, trecut între paranteze drepte. **Se pot efectua operaţii asupra fiecărui element al tabloului, nu asupra întregului tablou.**  Compilatorul C++ *alocă un spaţiu de memorie egal cu numărul maxim de elemente ale tabloului*, rezervând octeţi în funcţiede tipul de bază al fiecărui tablou.  **Inițializarea tablourilor**  Tablourile, ca şi variabilele simple pot fi iniţializate. Tablourile globale se iniţializează prin definiţiile lor la compilare, iar tablourile locale se iniţializează prin declaraţiile lor la execuţie.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | ***a. prin introducerea datelor de la tastatură/fișier***  int v[100],n;  cin>>n;  for (int i=0; i<n; ++i)  cin >> v[i]; | ***b. prin folosirea instrucţiunii de atribuire***   int v[5],n;  for (int i=0; i<n; ++i)  v[i] = 0;  /\* toate componentele    vectorului v sunt iniţializate cu 0.  \*/ | ***c. prin utilizarea unei constante de tip tablou***  int v[5] = {1,2,3,4,5};   |  |  | | --- | --- | | int t[10] = {1,2,3};  /\* t[0] = 1, t[1] = 2, t[2] = 3; celelalte componente ale tabloului t[3], …, t[9] au  valoarea 0 dacă tabloul este declarat global sau o valoare iniţială imprevizibilă dacă tabloul este declarat local. \*/ |  | |   **Accesul la elementele tabloului**   • **Accesul la fiecare element al tabloului se face prin numele acestuia urmat între paranteze, de indicele său (adică poziţia pe care o ocupă în tablou).**Ex. nume[indice]   • În limbajul C++, ***indicii tablourilor încep numărătoarea de la valoarea 0* *şi se termină cu limita superioară minus 1 (n-1).***  *Exemplu*  int  vec [4] = {12, 23, 34, 45};  Referirea la elemente  [https://sites.google.com/site/elearningalinabonto/_/rsrc/1487523297368/tablouri-unidimensionale---vectori/tablouri-unidimensionale--Vectori/1.png?height=92&width=200](https://sites.google.com/site/elearningalinabonto/tablouri-unidimensionale---vectori/tablouri-unidimensionale--Vectori/1.png?attredirects=0)  Primul element din vector are numărul de ordine (indicele) zero !  Astfel, elementul cu număr de ordine 0 din vector este 12, elementul al 2-lea este 23 și elementul 45 este al 4-lea element din vector (are indicele 3). | |

[Si](https://accounts.google.com/ServiceLogin?continue=https://sites.google.com/site/elearningalinabonto/tablouri-unidimensionale---vectori/1-tablouri-unidimensionale---vectori&service=jotspot)

**Afișarea tuturor elementelor unui tablou**

|  |  |
| --- | --- |
| ***a. afișarea tuturor elementelor unui tavlou***  for (int i=0; i<n; ++i)  cout<<v[i]<<” ”; | ***b. afișarea unui element de tablou***    cout<<v[5];  /\* toate componentele    vectorului v sunt iniţializate cu 0.  \*/ |