

Tablouri unidimensionale

vectori

Exemplu

- Presupunem că vrem să citim un șir de numere pe care să-l afișăm în ordine inversă citirii. Dacă șirul ar fi format din puține valori (ex. 5) atunci am putea folosi ceva de genul:
int a1, a2, a3, a4, a5;
- Dar dacă numărul de elemente e mare avem nevoie de o *structură de date* capabilă să memoreze informațiile dorite de noi.

Tablouri - noțiuni introductive

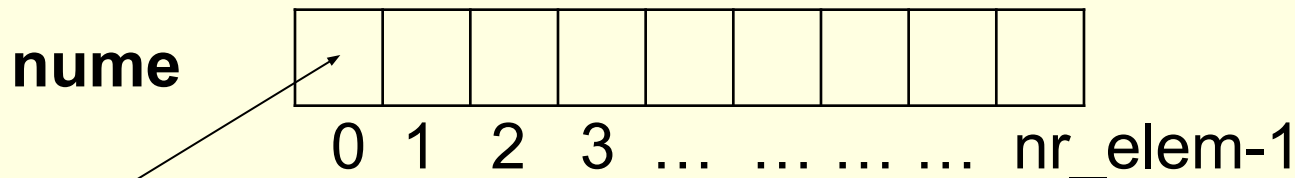
- Programele, în general, operează cu volume mari de date ce trebuie memorate “unde va”: în structuri de date.
- **Definiție:** o *structură de date* reprezintă un ansamblu de date organizate după anumite reguli, reguli ce depind de tipul de structură.

Tablouri

- **Definiție:** Un tablou este o colecție de date de același tip, memorate într-o zonă de memorie continuă, reunite sub un nume comun (numele tabloului)
- **Declarația:**
tip nume[Nr_elemente]

Structura internă

Un tablou declarat: tip **nume**[nr_elem]



tip - îmi spune ce pot să pun în tablou

nume – mă ajută să identific tabloul

nr_elem – îmi spune câte elemente are tabloul

- Deoarece elementele sunt în ordine, unul după altul, un element poate fi accesat prin intermediul numărului său de ordine (indice), primul element având numărul de ordine 0.

Exemplu

`int a[10]; // am declarat un tablou cu 10 elemente`

Elementele acestuia sunt: `a[0]`, `a[1]`, `a[2]`, ..., `a[9]`

a	3		5				9			
	0	1	2	3	4	5	6	7	8	9

Dacă facem atribuirile:

`a[2]=5;`

`a[6]=9;`

`a[0]=3;`

Observații

- Dimensiunea tabloului se calculează la compilarea programului: `dimensiunea=nr_elem*sizeof(tip)`.
- Dimensiunea maximă a unui tablou este limitată de compiler.
- Putem inițializa un tablou chiar de la declarare:
 - `int p3[5] = {1, 3, 9, 27, 81} ; //puterile lui 3`
 - `int p3[] = {1, 3, 9, 27, 81} ;`
 - cele două declarații de mai sus sunt echivalente

Vector - matrice

- Un tablou pentru care la declarare este specificată o singură dimensiune se numește *tablou unidimensional* sau *vector*.

Ex. `int a[100];` // vector cu 100 de elemente

- Un tablou pentru care la declarare sunt specificate două dimensiuni se numește *tablou bidimensional* sau *matrice*.

- Ex. `int x[3][5];` // matrice cu 3 linii și 5 coloane

Citirea/afișarea elementelor unui vector

```
#include <iostream>
using namespace std;
int a[20], i, n;
int main()
{ cout<<"Dati nr. de elemente:"; cin>>n;
  for(i=0;i<n;i++) cin>>a[i];
  cout<<"Elementele vectorului sunt:";
  for(i=0;i<n;i++) cout<<a[i];
  return 0;
}
```

Ordonarea elementelor unui vector

```
#include <iostream>
using namespace std;
int n, i, ok;
double a[100], aux;
int main()
{ cin>>n; for(i=1;i<=n;i++)cin>>a[i]; // citim elementele vectorului
  do{ok=1; // presupunem că vectorul este ordonat
    for(i=1;i<n;i++) // parcurgem vectorul
      if(a[i]>a[i+1]) // verificăm dacă 2 elem. nu resp. regula
        {aux=a[i]; a[i]=a[i+1]; a[i+1]=aux; ok=0;} // le interschimbăm
  }while(ok==0);
  for(i=1;i<=n;i++)cout<<a[i]<<" ";
  return 0;
}
```

Interclasarea a doi vectori

- Avem 2 vectori ordonați pe care vrem să-i unim obținând un al treilea vector ordonat.

```
#include <iostream>
using namespace std;
int a[100], b[100], c[200], na, nb, nc, ia, ib, ic, INF=32000;
int main()
{ cin>>na; for(ia=1;ia<=na; ia++)cin>>a[ia]; a[na+1]=INF;
  cin>>nb; for(ib=1;ib<=nb; ib++)cin>>b[ib]; b[nb+1]=INF;
  nc=na+nb; ia=1; ib=1;
  for(ic=1;ic<=nc;ic++)
    if(a[ia]<b[ib])c[ic]=a[ia++]; else c[ic]=b[ib++];
  for(ic=1;ic<=nc;ic++)cout<<c[ic]<<" ";
  return 0;
}
```