

Fundamentele programării

Curs 6

Şef lucr.dr.ing. GENGE Béla

Universitatea “Petru Maior”, Departamentul de Informatică
Tîrgu Mureş, România
bela.genge@ing.upm.ro

- Reprezintă un tip structurat de date cu elemente componente de același tip, ce ocupă o zonă de memorie contiguă
- Elementele sunt accesate după un index
- Tablouri: unidimensionale și multidimensionale (de regulă bidimensionale)

Tablouri unidimensionale (vectori)

- Declaraire:

```
tip_baza nume_tablou[dimensiune]
```

- Dimensiunea constantă
- Index primul element: 0
- Index ultimul element: dimensiune-1
- Numărul de octeți alocați: dimensiune*sizeof(tip_baza)
- nume_tablou denotă adresa de început a tabloului
- ATENȚIE! Compilatorul nu face verificări asupra depășirii limitelor (vezi atacuri buffer overflow)

Exemple

Exemplu1

```
int a[100];
```

Exemplu2

```
int n = 10;  
float a[n];
```

Exemplu3

```
const int n = 10;  
char a[n];
```

Exemple

Exemplu1

```
int a[100];
```

Exemplu2

```
int n = 10;  
float a[n];
```

Exemplu3

```
const int n = 10;  
char a[n];
```

Exemple

Exemplu1

```
int a[100];
```

Exemplu2

```
int n = 10;  
float a[n];
```

Exemplu3

```
const int n = 10;  
char a[n];
```

Exemple - citire, afișare

Exemplu1

```
int a[100];  
for ( int i = 0 ; i < n ; ++i ){  
    cin >> a[i];  
}
```

Exemplu2

```
int a[100];  
for ( int i = 0 ; i < n ; ++i ){  
    scanf("%d", &a[i]);  
}
```

Exemplu3

```
for ( int i = 0 ; i < n ; ++i ){  
    cout << a[i] << endl;  
}
```

Exemple - citire, afișare

Exemplu1

```
int a[100];  
for ( int i = 0 ; i < n ; ++i ){  
    cin >> a[i];  
}
```

Exemplu2

```
int a[100];  
for ( int i = 0 ; i < n ; ++i ){  
    scanf("%d", &a[i]);  
}
```

Exemplu3

```
for ( int i = 0 ; i < n ; ++i ){  
    cout << a[i] << endl;  
}
```


Exemple - citire, afișare

Exemplu1

```
int a[100];  
for ( int i = 0 ; i < n ; ++i ){  
    cin >> a[i];  
}
```

Exemplu2

```
int a[100];  
for ( int i = 0 ; i < n ; ++i ){  
    scanf("%d", &a[i]);  
}
```

Exemplu3

```
for ( int i = 0 ; i < n ; ++i ){  
    cout << a[i] << endl;  
}
```

Exemple

- Întrebare: pentru `int a[10];`, adresa de început 0, ce adresă va avea `a[5]`?
- Aceeași întrebare pentru:
 - `char a[10];`
 - `short int a[10];`
 - `unsigned short int a[10];`

Tablouri unidimensionale (vectori)

- Inițializare:
 - `int a[] = {1, 2, 3, 4, 5}; //5 elemente`
 - `int a[10] = {1, 2, 3, 4, 5}; //10 elemente`
- Care va fi valoarea inițială fără inițializare?

- Se citesc n numere întregi de la tastatură într-un tablou unidimensional. Să se afișeze numerele în ordinea introducerii și în ordine inversă.

Problemă

- Se citesc n numere întregi de la tastatură într-un tablou unidimensional. Să se determine numerele care apar o singură dată.

- Un alpinist își notează pe parcursul drumului în N locații diferite altitudinea la care se află în metri (peste nivelul mării). Implementați un program care determină:
 - Altitudinea maximă și locația în care a fost măsurată
 - Altitudinea minimă și locația în care a fost măsurată
 - Altitudinea medie
 - TEMĂ: numărul văilor și vârfurilor parcurse

Tablouri bidimensionale (matrici)

- Declaraire:

```
tip_baza nume_tablou[dimensiune1][dimensiune2]
```

- Dimensiunea constantă
- Index primul element: 0,0
- Index ultimul element: dimensiune1-1,dimensiune2-1
- Numărul de octeți alocați:
`dimensiune1*dimensiune2*sizeof(typ_baza)`
- `nume_tablou` denotă adresa de început a tabloului
- Cu toate că tabloul este bidimensional, acesta va ocupa tot o zonă contiguă de memorie

Exemple

Exemplu1

```
int a[10][100];
```

Exemplu2

```
int n = 10;  
float a[n][100];
```

Exemplu3

```
const int n = 10;  
char a[n][n];
```


Exemple

Exemplu1

```
int a[10][100];
```

Exemplu2

```
int n = 10;  
float a[n][100];
```

Exemplu3

```
const int n = 10;  
char a[n][n];
```

Exemple

Exemplu1

```
int a[10][100];
```

Exemplu2

```
int n = 10;  
float a[n][100];
```

Exemplu3

```
const int n = 10;  
char a[n][n];
```

Exemple - citire, afișare

Exemplu1

```
int a[10][10];  
for ( int i = 0 ; i < n ; ++i )  
    for ( int j = 0 ; j < m ; ++j )        cin >> a[i][j];
```

Exemplu2

```
int a[10][10];  
for ( int i = 0 ; i < n ; ++i )  
    for ( int j = 0 ; j < m ; ++j )        scanf("%d", &a[i][j]);
```

Exemplu3

```
for ( int i = 0 ; i < n ; ++i ){  
    for ( int j = 0 ; j < m ; ++j )  
        printf("%5d", a[i][j]);  
    printf("\n");  
}
```

Exemple - citire, afișare

Exemplu1

```
int a[10][10];
for ( int i = 0 ; i < n ; ++i )
    for ( int j = 0 ; j < m ; ++j )        cin >> a[i][j];
```

Exemplu2

```
int a[10][10];
for ( int i = 0 ; i < n ; ++i )
    for ( int j = 0 ; j < m ; ++j )        scanf("%d", &a[i][j]);
```

Exemplu3

```
for ( int i = 0 ; i < n ; ++i ){
    for ( int j = 0 ; j < m ; ++j )
        printf("%5d", a[i][j]);
    printf("\n");
}
```

Exemple - citire, afișare

Exemplu1

```
int a[10][10];
for ( int i = 0 ; i < n ; ++i )
    for ( int j = 0 ; j < m ; ++j )        cin >> a[i][j];
```

Exemplu2

```
int a[10][10];
for ( int i = 0 ; i < n ; ++i )
    for ( int j = 0 ; j < m ; ++j )        scanf("%d", &a[i][j]);
```

Exemplu3

```
for ( int i = 0 ; i < n ; ++i ){
    for ( int j = 0 ; j < m ; ++j )
        printf("%5d", a[i][j]);
    printf("\n");
}
```

Exemple

- Întrebare: pentru `int a[10][10];`, adresa de început 0, ce adresă va avea `a[5][0]`?
- Aceeași întrebare pentru:
 - `char a[10][10];`
 - `short int a[10][10];`
 - `unsigned short int a[10][10];`

Tablouri bidimensionale (matrici)

- Inițializare:

- `int a[][2] = {{1, 2}, {3, 4}, {5, 6}}; //3 "rânduri", 2 "coloane"`
- `int a[3][] = {{1, 2}, {3, 4}, {5, 6}}; //eroare (calcularea memoriei alocate necesită specificarea numărului de elemente pe linie, prima dimensiune este doar un divizor!!!)`
- `int a[][] = {{1, 2}, {3, 4}, {5, 6}}; //eroare`

- Care va fi valoarea inițială fără inițializare?

Problemă

- Se citesc $n \times m$ numere întregi de la tastatură într-un tablou bidimensional. Să se afișeze numerele pentru care $i=j$ (diagonala principală).

- L veverițe adună alune. Alunele sunt strânse în locații strategice la coordonatele i, j (date de utilizator) într-un plan bidimensional $n \times m$. Să se implementeze o aplicație ce determină pentru un număr total de X alune:
 - Distribuția finală a alunelor pentru fiecare veveriță
 - Vecinii direcți pentru fiecare veveriță