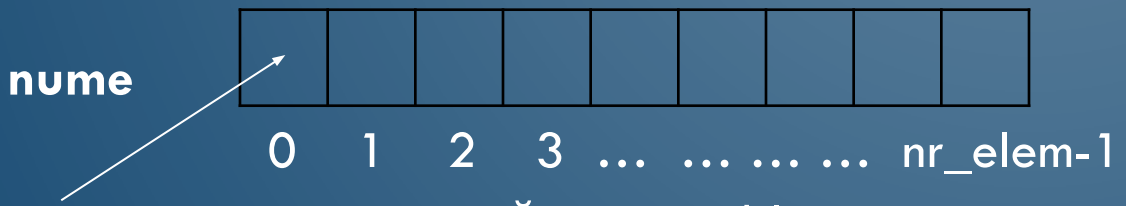
TABLOU/MATRICE BIDIMENSIONALĂ

TABLOU UNIDIMENSIONAL - VECTOR

Un tablou unidimensional se declară astfel: 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.

TABLOU BIDIMENSIONAL

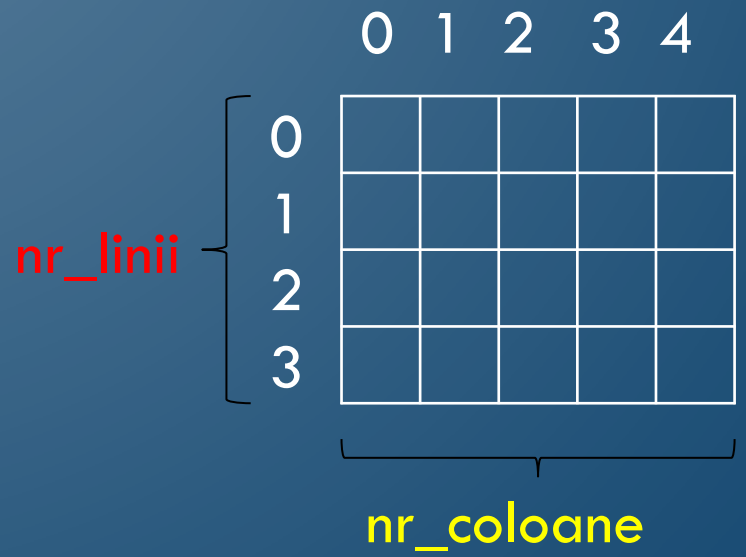
- Declararea unui tablou bidimensional: tip nume[nr_linii][nr_coloane]

tip - îmi spune ce pot să pun în tablou

nume – mă ajută să identific tabloul

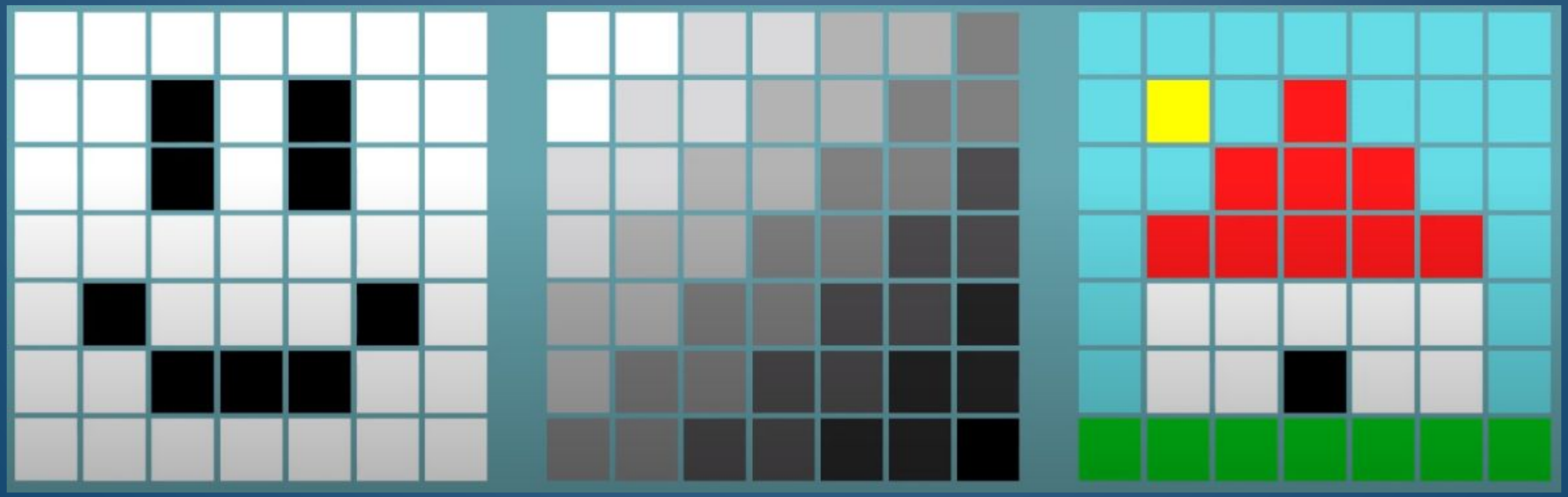
nr_linii – îmi spune câte linii are tabloul

nr_coloane - Îmi spune câte coloane are tabloul



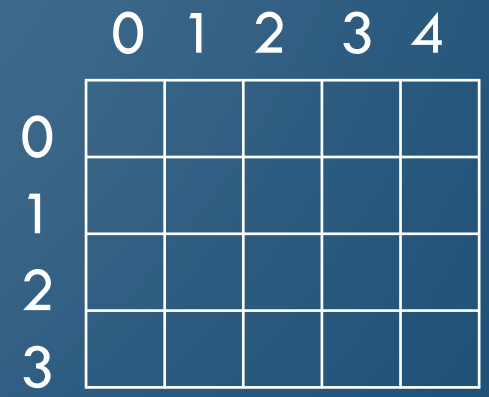
DE CE AȘ FOLOSI TABLOURI BIDIMENSIONALE ?

- O imagine poate fi memorată într-un tablou bidimensional



EXERCIȚIU

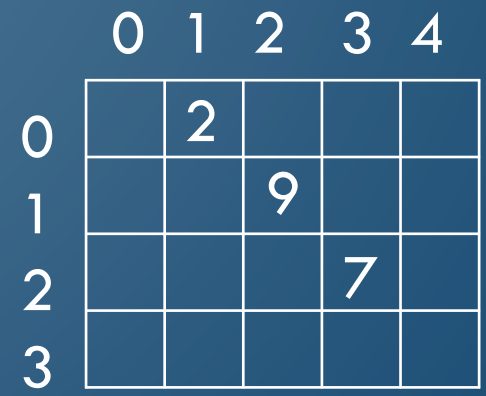
- Trebuie să declar un tablou a, ca cel din imagine, ce scriu?
- int a[4][5];
- De ce ?
- Deoarece are 4 linii și 5 coloane.



ACCESAREA UNUI ELEMENT

Presupunem că am declarat tabloul int a[4][5];

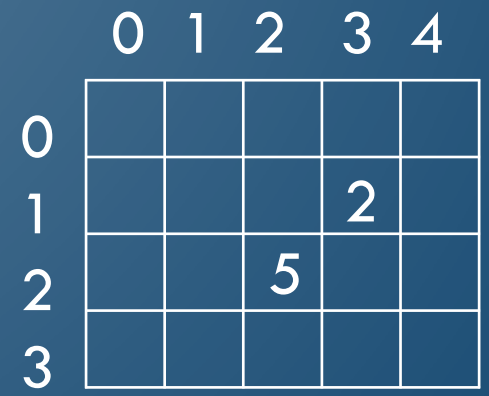
- Pentru accesarea unui element scriem a[linie][coloana]
- Exemplu:
- Atribuirea a[0][1] = 2 va modifica elementul de pe linia 0, coloana 1
- a[2][3] = 7
- a[1][2] = a[0][1] + a[2][3]



EXERCIȚIU

Presupunem că am declarat tabloul int a[4][5];

- Ce trebuie să scriu pentru a pune valoarea 2 în celula din imagine?
- a[1][3] = 2;
- Dar pentru a pune valoarea 5?
- a[2][2] = 5;

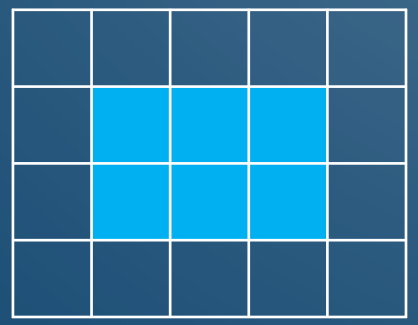


CITIREA ELEMENTELOR UNUI TABLOU BIDIMENSIONAL LINIE CU LINIE

- Pas 1: Citești numărul de linii și de coloane ale zonei în care faci citirea
- Pas 2: Parcurgi liniile și pentru fiecare linie citești elementele de pe aceasta
- Vei avea nevoie de două structuri repetitive imbricate

EXEMPLUL 1

- Pentru n=2, m =3 se citește zona din matrice colorată în albastru

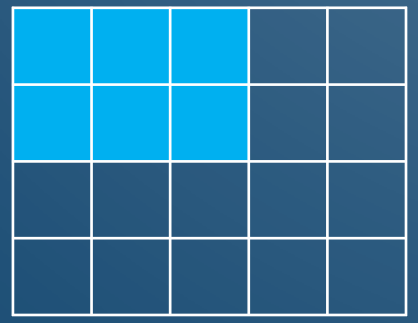


```
#include <iostream>

using namespace std;
int a[4][5], i, j, n, m;
int main()
{
    cin >> n >> m;
    for(i=1; i<=n; i++)
        for(j=1; j<=m; j++)
            cin >> a[i][j];
    return 0;
}
```

EXEMPLUL 2

- Pentru n=2, m =3 se citește zona din matrice colorată în albastru



```
#include <iostream>

using namespace std;
int a[4][5], i, j, n, m;
int main()
{
    cin >> n >> m;
    for(i=0; i<n; i++)
        for(j=0; j<m; j++)
            cin >> a[i][j];
    return 0;
}
```

OBSERVAȚII

- În ambele cazuri se citește o matrice cu 2 linii și 3 coloane
- Ceea ce diferă este locul în care sunt memorate datele în matricea declarată
- Pentru declararea corectă a matricei trebuie avut în vedere numărul maxim de linii și coloane pe care aceasta trebuie să le memoreze.

EXEMPLU

- Să presupunem că trebuie să rezolvi problema: *Fiind dată o matrice cu **n** linii și **m** coloane (1< n, m <100), determină suma elementelor de pe fiecare linie.*
- Pas 1: Declari variabilele necesare.
- Cum declari matricea?
- int a[100][100]
- Ce alte variabile mai ai nevoie?
- int i, j, n, m, s;

EXEMPLU

- Care sunt pașii rezolvării problemei?
- Pas 1: Citesc dimensiunile matricei
- Pas 2: Citesc matricea
- Pas 3: Parcurg matricea linie cu linie. Pentru fiecare linie inițializez suma cu 0 și parcurg elementele liniei, calculând suma.

PROGRAM

```cpp
#include <iostream>

using namespace std;
int a[100][100], i, j, n, m, s;
int main()
{
    cin >> n >> m;
    for(i=0; i<n; i++)
        for(j=0; j<m; j++)
            cin >> a[i][j];
    for(i=0; i<n; i++)
    {
        s = 0;
        for(j=0; j<m; j++) s = s + a[i][j];
        cout << s << " ";
    }
    return 0;
}
```

AFIȘAREA MATRICELOR BIDIMENSIONALE

- Fie problema: *Scrieți un program care citește de la tastatură două numere naturale nenule n și m (0<n, m<50), apoi construiește în memorie și afișează o matrice A cu n linii (numerotate de la 1 la n) şi m coloane (numerotate de la 1 la m) cu proprietatea că fiecare element A_{ij} memorează cea mai mică dintre valorile indicilor i şi j (1≤i≤n, 1≤j≤m).*
- Exemplu: pentru n=4, m = 5 se afișează

1 1 1 1 1
1 2 2 2 2
1 2 3 3 3
1 2 3 4 4

AFIȘAREA MATRICELOR BIDIMENSIONALE

- Ce variabile aș avea nevoie pentru scrierea programului?
- int a[51][51], i, j, n, m;
- Care sunt pașii de rezolvare a problemei?
- Pas 1: Citesc dimensiunile matricei n, m.
- Pas 2: Parcurg matricea și completez valorile conform cerinței
- Pas 3: Afișez matricea

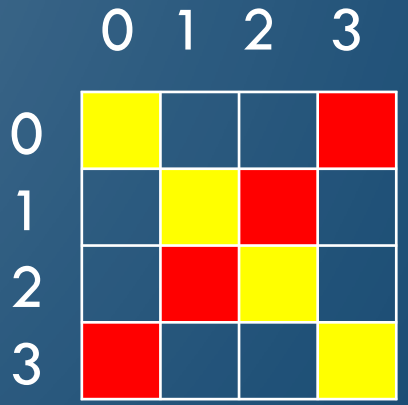
AFIȘAREA MATRICELOR BIDIMENSIONALE

```cpp
#include <iostream>

using namespace std;
int a[51][51], i, j, n, m;
int main()
{
    cin >> n >> m;
    for(i=1; i<=n; i++)
        for(j=1; j<=m; j++)
            a[i][j] = min(i, j);
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=m; j++) cout << a[i][j] << ' ';
        cout << '\n';
    }
    return 0;
}
```

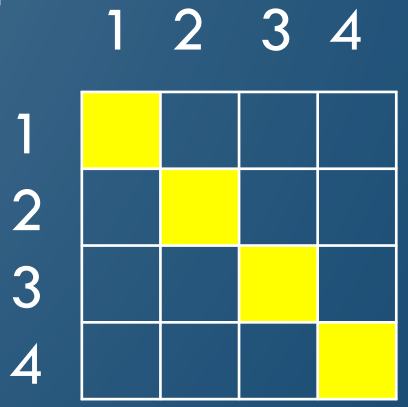
MATRICE PĂTRATICE

- Dacă pentru o matrice numărul de linii este egal cu numărul de coloane, atunci avem o matrice pătratică.
- Elementele marcate cu galben sunt situate pe diagonala principală
- Elementele marcate cu roșu sunt situate pe diagonala secundară.



EXERCIȚIU

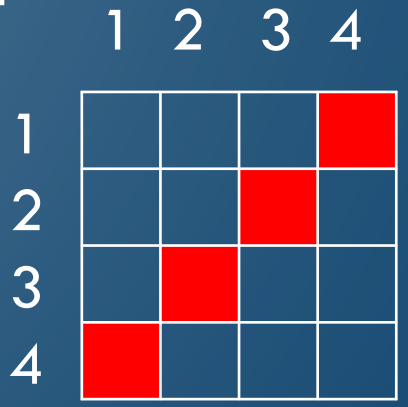
- Afișează elementele de pe diagonala principală a unei matrice pătratice.
- Ce coordonate au elementele de pe diagonala principală?
- (1, 1), (2, 2), (3, 3), (4,4). Ce observați la coordonate?
- De câte structuri repetitive avem nevoie pentru a le afișa?
- Una.
- Soluție:




```
for(i=1; i<=n; i++) cout << a[i][i] <<' ';
```

EXERCIȚIU

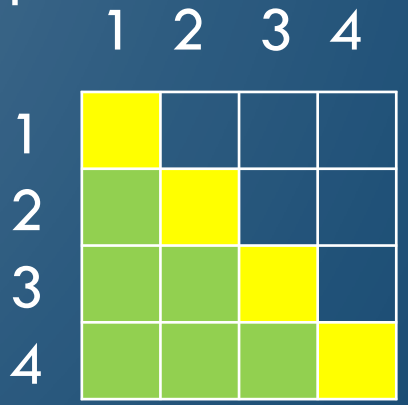
- Afișează elementele de pe diagonala secundară a unei matrice pătratice.
- Ce coordonate au elementele de pe diagonala secundară?
- (1, 4), (2, 3), (3, 2), (4, 1). Ce observați la coordonate?
- De câte structuri repetitive avem nevoie pentru a le afișa?
- Una.
- Soluție:




```
for(i=1; i<=n; i++) cout << a[i][n-i+1] <<' ';
```

EXERCIȚIU

- Afișează elementele de sub diagonala principală a unei matrice pătratice.
- De câte structuri repetitive avem nevoie pentru a le afișa?
- Două. Avem de a face cu o suprafață.
- Soluție:




```
for(i=2; i<=n; i++)
{
    for(j=1; j<i; j++) cout << a[i][j] << ' ';
    cout << '\n';
}
```

TEMĂ

- Probleme de pe pbinfo.ro
- GenMat2: https://www.pbinfo.ro/probleme/207/genmat2
- CmmdcSum: https://www.pbinfo.ro/probleme/780/cmmdcsum
- Diagonale1: https://www.pbinfo.ro/probleme/783/diagonale1
- SumElPare: https://www.pbinfo.ro/probleme/662/sumelpare