Tablouri bidimensionale – Matrice

Un tablou bidimensional sau matricea este o structură de date care are un număr dat de elemente, de același tip, dispuse pe linii și coloane.

$A_{3,2} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{pmatrix}$ este o matrice cu 3 linii și 2 coloane.

Declararea unui tablou bidimensional în C++ se realizează astfel:

tip_date nume [nr_linii][nr_coloane];

Ex: int a[4][3];

Obs: În C++ numerotarea liniilor/coloanelor începe de la 0 (0 fiind și primul indice al liniei/coloanei), astfel, la declarare se adaugă o linie/coloană în plus. Nu este obligatorie crearea/prelucrarea de la 0 la n-1.

1. Citirea unei matrice cu 3 linii și 2 coloane - - >

```
#include <iostream>
using namespace std;
int i,j,m,n,a[4][3];
int main()
{
    cin>>m>>n;
    for (i=1;i<=m;i++)
        for (j=1;j<=n;j++)
        cin>>a[i][j];
    return 0;
}
```

2. Afișarea unei matrice:

```
for (i=1;i<=m;i++)
    {
        for (j=1;j<=n;j++)
        {
            cout<<a[i][j]<<" ";
        }
        cout<<endl;
    }

return 0;
```

Obs. În cazul în care numărul de linii este egal cu numărul de coloane, spunem că avem o matrice pătratică. Astfel, vom declara o singură dimensiune, n, și parcurgerile vor fi făcute de la 1 la n sau de la 0 la n-1.

În cazul matricelor pătratice putem vorbi de diagonale: **principală**(indicele liniei este egal cu al coloanei) și **secundară** (suma indicilor liniei și coloanei este egală cu n+1).

Totodată, se pot accesa și prelucra elementele de deasupra diagonalei principală/secundară, respectiv, de sub diagonala principală/secundară, astfel:

1. Elementele de deasupra diagonalei principale au indicii liniei strict mai mici decât cei ai coloanei. (i < j)
2. Elementele de sub diagonala principală au indicii liniei strict mai mici decât cei ai coloanei. (i > j)

3. Elementele de deasupra diagonalei secundare au suma indicilor strict mai mici decât n+1. (i + j < n + 1)
4. Elementele de sub diagonala secundară au suma indicilor strict mai mare decât n+1. (i + j > n + 1)

Ex: 1. Construiți și afișați o matrice pătratică cu n elemente (n<=10), astfel: Elementele de pe diagonala principală sunt egale cu 1, elementele aflate deasupra diagonalei principale sunt egale cu dublul sumei indicilor, iar cele aflate sub diagonala principală sunt egale cu diferența indicilor dintre linie și coloană.

```
#include <iostream>
using namespace std;
int i,j,n,a[11][11];
int main()
{
    cin>>n;
    for (i=1;i<=n;i++)
        for (j=1;j<=n;j++)
        {
            if (i==j) a[i][j]=1;
            if (i<j) a[i][j]=2*(i+j);
            if (i>j) a[i][j]=i-j;
        }
    for (i=1;i<=n;i++)
        {
            for (j=1;j<=n;j++)
            {
                cout<<a[i][j]<<" ";
            }
            cout<<endl;
        }
    return 0;
}
```

Ex: 2. Construiți și afișați o matrice pătratică cu n elemente (n<=6), astfel: Elementele de pe diagonala secundară sunt egale cu 0, elementele aflate deasupra diagonalei secundare sunt egale cu suma indicilor, iar cele aflate sub diagonala secundară sunt egale cu diferența dintre numărul maxim de elemente și indicele liniei.

```
#include <iostream>
using namespace std;
int i,j,n,a[7][7];
int main()
{
    cin>>n;
    for (i=1;i<=n;i++)
        for (j=1;j<=n;j++)
        {
            if (i+j==n+1) a[i][j]=0;
            if (i+j<n+1) a[i][j]=i+j;
            if (i+j>n+1) a[i][j]=n-i;
        }
    for (i=1;i<=n;i++)
        {
            for (j=1;j<=n;j++)
            {
                cout<<a[i][j]<<" ";
            }
            cout<<endl;
        }
    return 0;
}
```