

## CAPITOLUL II

### TABLOURI BIDIMENSIONALE

#### 2.1. Declararea matricelor

Să ne reamintim definiția tabloului:

**Definiție:**

*Un tablou este o colecție de date de același tip, memorate într-o zonă contiguă de memorie.*

Un tablou care conține două sau mai multe linii și două sau mai multe coloane, poartă numele de *tablou bidimensional* sau *matrice*.

#### Declararea unui tablou bidimensional:

Pentru un tablou bidimensional, la declarare trebuie precizate două dimensiuni (**numărul de linii** și **numărul de coloane**).

**tip** **nume**[**număr\_linii**][**număr\_coloane**]

unde: **tip** poate fi orice tip din C

**nume** este un identificator C reprezintă numele tabloului

**număr\_linii** este o expresie constantă întreagă pozitivă, reprezentând numărul de linii ale tabloului

**număr\_coloane** este o expresie constantă întreagă pozitivă, reprezentând numărul de coloane ale tabloului.

#### Accesarea elementelor unui tablou bidimensional:

Fiecare element al tabloului bidimensional este bine determinat de doi indici: **indicele de linie** și **indicele de coloană**.

**Indicele de linie** este o expresie întreagă pozitivă și poate lua valori în intervalul **[0 , număr\_linii - 1]**

**Indicele de coloană** este o expresie întreagă pozitivă și poate lua valori în intervalul **[0 , număr\_coloane - 1]**

**Exemplu:** Dacă dorim să utilizăm o matrice cu trei linii și două coloane, cu elemente numere întregi cu numele **a** vom declara:

```
int a[3][2];
```

<b>a</b>	Coloana 0	Coloana 1
linia 0	a[0][0]	a[0][1]
linia 1	a[1][0]	a[1][1]
linia 2	a[2][0]	a[2][1]

Dacă dorim să atribuim valoarea 3 elementului de pe linia 1 coloana 0, vom scrie:

```
a[1][0]=3.
```

## 2.2. Operații elementare cu matrice

### Citirea matricei:

```
//Citirea se realizează parcurgând matricea linie cu linie
void citire(int a[][10],int &n,int &m)
{ int i,j;
  cout<<"dati nr de linii:";cin>>n;
  cout<<"dati nr de coloane:";cin>>m;
  cout<<"dati elem:";
  for(i=0;i<n;i++)
    for(j=0;j<m;j++)
      {cout<<"a["<<i<<"["<<j<<"]=";
       cin>>a[i][j];
      }
}
```

### Afișarea matricei:

```
// afișarea se realizează linie cu linie
void afisare(int a[][10],int n,int m)
{ int i,j;
  for(i=0;i<n;i++)
    {for(j=0;j<m;j++)
     cout<<a[i][j]<<' ';
     cout<<endl;
    }
}
```

În operațiile de citire și afișare, s-a parcurs matricea pe linii, adică elementele de pe linia de indice 0 ( $a[0][0]$ ,  $a[0][1]$ ,  $a[0][2]$ , ...  $a[0][m-1]$ ), apoi elementele de pe linia de indice 1 ( $a[1][0]$ ,  $a[1][1]$ ,  $a[1][2]$ , ...  $a[1][m-1]$ ), .... elementele de pe linia de indice  $n-1$  ( $a[n-1][0]$ ,  $a[n-1][1]$ ,  $a[n-1][2]$ , ...  $a[n-1][m-1]$ ).

	coloana 0	coloana 1	...	coloana m-1
linia 0			...	
linia 1			...	
...	...	...	...	...
linia n-1			...	

Matricea poate fi parcursă și pe coloane (coloana 0, coloana 1, ... , coloana  $m-1$ ). În exemplul următor vom calcula sumele elementelor de pe coloanele matricei:

```

void sume_coloane(int a[][10],int n, int m)
{ int i,j,s;
for (j=0;j<m;j++)
  { s=0; // pentru fiecare coloana, se initializeaza suma s cu 0
  for(i=0;i<n;i++) // aici parcurgem o coloana
    s=s+a[i][j];
  cout<<" suma elementelor de pe coloana "<<j<<"="<<s<<endl;
  }
}

```

### 2.3. Matrice pătratică

#### Definiție:

O matrice cu proprietatea că numărul de linii este egal cu numărul de coloane se numește **matrice pătratică**

#### Diagonala principală a unei matrice pătratică

Elementele situate pe diagonala principală se caracterizează prin faptul că **indicele de linie este egal cu indicele de coloană** sau prin diferența indicilor egală cu 0.

	0	1	2	3	4
0	a[0][0]	a[0][1]	a[0][2]	a[0][3]	a[0][4]
1	a[1][0]	a[1][1]	a[1][2]	a[1][3]	a[1][4]
2	a[2][0]	a[2][1]	a[2][2]	a[2][3]	a[2][4]
3	a[3][0]	a[3][1]	a[3][2]	a[3][3]	a[3][4]
4	a[4][0]	a[4][1]	a[4][2]	a[4][3]	a[4][4]

**Diagonala principală împarte matricea în două zone:** zona situată deasupra diagonalei principale și zona situată dedesubtul diagonalei principale.

**Elementele matricei situate deasupra diagonalei principale** se caracterizează prin faptul că indicele de linie este strict mai mic decât indicele de coloană (dacă accesăm elementul matricei  $a[i][j]$ ,  $i < j$ ) sau  $i - j < 0$ .

**Elementele matricei situate sub diagonala principală** se caracterizează prin faptul că indicele de linie este strict mai mare decât indicele de coloană (dacă accesăm elementul matricei  $a[i][j]$ ,  $i > j$ ) sau  $i - j > 0$ .

În plus, observăm că **elementele situate pe aceeași „paralelă” la diagonala principală** se caracterizează prin faptul că **diferența indicilor este o constantă** ( $i - j = k$ ,  $k \in [-(n-1), n-1]$ ).

### Diagonala secundară a unei matrice pătratice

	0	1	2	3	4
0	$a[0][0]$	$a[0][1]$	$a[0][2]$	$a[0][3]$	$a[0][4]$
1	$a[1][0]$	$a[1][1]$	$a[1][2]$	$a[1][3]$	$a[1][4]$
2	$a[2][0]$	$a[2][1]$	$a[2][2]$	$a[2][3]$	$a[2][4]$
3	$a[3][0]$	$a[3][1]$	$a[3][2]$	$a[3][3]$	$a[3][4]$
4	$a[4][0]$	$a[4][1]$	$a[4][2]$	$a[4][3]$	$a[4][4]$

Elementele situate pe diagonala secundară se caracterizează prin faptul că **suma indicilor este egală cu  $n-1$**  ( $n$  este dimensiunea matricei).

Asemănător diagonalei principale, diagonala secundară **împarte elementele matricei în două zone.**

**Elementele matricei situate deasupra diagonalei secundare** se caracterizează prin faptul că suma indicilor  $i + j < n - 1$  iar **elementele situate sub diagonala secundară** se caracterizează prin faptul că suma indicilor  $i + j > n - 1$ .

Se observă că **elementele situate pe aceeași „paralelă” la diagonala secundară** au proprietatea că **suma indicilor este o constantă**. ( $i + j = k$ ,  $k \in [0, 2 \cdot (n - 1)]$ )

Cele două diagonale împart elementele matricei în patru zone, în funcție de poziția fiecărui element față de cele două diagonale, astfel:

ZONA N („nord”), elementele situate deasupra ambelor diagonale.

ZONA S („sud”), elementele situate dedesubtul ambelor diagonale.

ZONA E („est”), elementele situate deasupra diagonalei principale și dedesubtul diagonalei secundare.

ZONA V („vest”), elementele situate dedesubtul diagonalei principale și deasupra diagonalei secundare.

		ZONA N						
		0	1	2	3	4		
Z O N A  V	0	a[0][0]	a[0][1]	a[0][2]	a[0][3]	a[0][4]		
	1	a[1][0]	a[1][1]	a[1][2]	a[1][3]	a[1][4]	Z	O
	2	a[2][0]	a[2][1]	a[2][2]	a[2][3]	a[2][4]	N	A
	3	a[3][0]	a[3][1]	a[3][2]	a[3][3]	a[3][4]	E	
	4	a[4][0]	a[4][1]	a[4][2]	a[4][3]	a[4][4]		
		ZONA S						

## 2.4. Probleme rezolvate

1. Se dau două matrice A și B cu n linii și m coloane.

Să se calculeze și să se afișeze suma matricelor.

Datele se citesc din fișierul „mat.in” astfel: pe prima linie n și m , apoi de pe următoarele n linii câte m elemente ale matricei A și în continuare de pe următoarele n linii câte m elemente ale matricei B.

Rezultatul se va afișa pe ecran.

Pentru a putea însuma două matrice trebuie ca ele să aibă același număr de linii și același număr de coloane.

Suma a două matrice este tot o matrice, fiecare element al sumei fiind egal cu suma elementelor de aceiași indici din matricile ce se însumează.

## // SUMA A DOUĂ MATRICE

```
#include<conio.h>
#include<fstream.h>
void citire(int a[][10],int &n, int &m,char l, ifstream &f);
void afisare(int a[][10],int n, int m)
void suma(int a[][10],int b[][10],int c[][10],int n,int m)
void main()
{ int a[10][10],b[10][10],c[10][10],n,m;
  ifstream f("mat.in");
  f>>n>>m;
  cout<<"prima matrice"<<endl;citire(a,n,m,'a',f);
  cout<<"a doua matrice"<<endl;citire(b,n,m,'b',f);
  f.close();
  cout<<"suma celor doua matrice:"<<endl;
  suma(a,b,c,n,m);
  afisare(c,n,m);
  getch();
}

void citire(int a[][10],int &n, int &m,char l, ifstream &f)
{ int i,j;
  for(i=0;i<n;i++)
    for(j=0;j<m;j++)
      f>>a[i][j];
}

void afisare(int a[][10],int n, int m)
{ int i,j;
  for(i=0;i<n;i++)
    {for(j=0;j<m;j++)
      cout<<a[i][j]<<' ';
      cout<<endl;
    }
  cout<<endl;
}

void suma(int a[][10],int b[][10],int c[][10],int n,int m)
{ int i,j;
  for(i=0;i<n;i++)
    for(j=0;j<m;j++)
      c[i][j]=a[i][j]+b[i][j];
}
```

**Comment [A1]:** Am modificat ordinea functiilor

2. Să se scrie o funcție care calculează produsul a două matrice.

Cele două matrice A și B vor fi transmise ca și parametri, la fel numărul de linii și coloane iar matricea produs va fi returnată tot prin intermediul unui parametru. Parametri vor fi a,b,c (matrici), n,m,k numărul de linii și coloane, astfel  $A_{n \times m}$ ,  $B_{m \times k}$ ,  $C_{n \times k}$

Se știe de la matematică modalitatea de calcul al produsului a două matrice, și anume se face produsul „linie cu coloană”.

Pentru a putea efectua produsul  $A \times B$  este necesar ca numărul de coloane ale matricei A să fie egal cu numărul de linii ale matricei B.

Un element al produsului se calculează astfel:  $c_{i,k} = \sum_{j=0}^{m-1} a_{i,j} * b_{j,k}$ .

#### //PRODUSUL A DOUA MATRICE

```
void produs(int a[10][10],int b[10][10],int c[10][10],int n, int m, int l)
{ int i,j,k;
  for(i=0;i<n;i++)
    for(k=0;k<l;k++)
      { c[i][k]=0; // fiecare element al mat produs, fiind o sumă, se inițializează cu 0.
        for(j=0;j<m;j++)
          c[i][k]+=a[i][j]*b[j][k];
      }
}
```

3. Să se construiască și să se afișeze transpusa unei matrice cu n linii și m coloane. Datele se citesc din fișierul „mat.in” astfel: pe prima linie n și m , apoi de pe următoarele n linii câte m elemente ale matricei.

Reamintim faptul că transpusa unei matrice A cu n linii și m coloane este tot o matrice, B cu m linii și n coloane. Elementele matricei A se obțin prin „transformarea” liniilor matricei A în coloane în matricea B.

#### Exemplu:

A

1	2	3
4	5	6

B

1	4
2	5
3	6

**Comment [A2]:** In programul de mai jos am schimbat ordinea functiilor

```
                //MATRICEA TRANSPUSA
#include<conio.h>
#include<fstream.h>
#include<iostream.h>
void citire(int a[][10],int &n, int &m);
void afisare(int a[][10],int n, int m);
void transpusa(int a[][10],int na,int ma,int b[][10],int &nb,int &mb);

void main()
{ int a[10][10],b[10][10],na,ma,nb,mb;
  citire(a,na,ma);
  cout<<"matricea a:"<<endl;
  afisare(a,na,ma);
  transpusa(a,na,ma,b,nb,mb);
  cout<<"matricea transpusa:"<<endl;
  afisare(b,nb,mb);
  getch();
}

void citire(int a[][10],int &n, int &m)
{ int i,j;
  ifstream f("mat.in");
  f>>n>>m;
  for(i=0;i<n;i++)
    for(j=0;j<m;j++)
      f>>a[i][j];
  f.close();
}

void afisare(int a[][10],int n, int m)
{ int i,j;
  for(i=0;i<n;i++)
    { for(j=0;j<m;j++)
      cout<<a[i][j]<<' ';
      cout<<endl;
    }
  cout<<endl;
}

void transpusa(int a[][10],int na,int ma,int b[][10],int &nb,int &mb)
{ int i,j;
  nb=ma;mb=na;
  for(i=0;i<na;i++)
    for(j=0;j<ma;j++)
      b[j][i]=a[i][j];
}
```



4. Pentru o matrice cu n linii și m coloane, citită din fișier în același mod cu problemele precedente, pe baza unui meniu să se realizeze următoarele prelucrări:

- eliminarea unei linii l
- inserarea unei linii de indice l cu toate elementele egale cu 0
- interschimbarea elementelor a două linii l1 și l2 citite de la tastatură.

**Comment [A3]:** In programul de mai jos am schimbat ordinea functiilor

```

#include<conio.h> //INSERARE, STERGERE, INTERSCHIMBARE
#include<fstream.h>
#include<iostream.h>
void citire(int a[][10],int &n, int&m);
int meniu();
void afisare(int a[][10],int n, int m);
int cit_lin(int n);
void sterg_lin(int a[][10],int &n, int m,int l);
void ins_lin(int a[][10],int &n, int m,int l);

void main()
{ int a[10][10],n,m,l,l1,l2,o;
  citire(a,n,m);
  afisare(a,n,m);
  do{
    o=meniu();
    switch(o)
      { case 1:l=cit_lin(n);sterg_lin(a,n,m,l);afisare(a,n,m);break;
        case 2:l=cit_lin(n);ins_lin(a,n,m,l);afisare(a,n,m);break;
        case 3:l1=cit_lin(n);l2=cit_lin(n);
          schimb_lin(a,n,m,l1,l2);afisare(a,n,m);break;
        case 4:afisare(a,n,m);
        }
    } while (o!=5);
  getch();
}

void citire(int a[][10],int &n, int&m)
{int i,j;
  ifstream f("mat.in");
  f>>n>>m;
  for(i=0;i<n;i++)
    for(j=0;j<m;j++)
      f>>a[i][j];
  f.close();
}

int cit_lin(int n) // functia citeste si returneaza un indice valid pentru linie
{ int l;
  cout<<"linia (o valoare intre 0 si "<<(n-1)<<") :";
  do{ cin>>l; } while(l<0 ||l>n-1);
  return l;
}

```

```

int meniu()
{ int o;
  cout<<"1. stergerea unei linii"<<endl;
  cout<<"2. inserarea unei linii"<<endl;
  cout<<"3. interschimbarea a doua linii"<<endl;
  cout<<"4. afisarea matricei"<<endl;
  cout<<"5. Sfarsit"<<endl;
  cout<<endl<<"Alege:(1-5) ";
  do {cin>>o;} while (o<1||o>5);
  return o;
}

void afisare(int a[][10],int n, int m)
{ int i,j;
  for(i=0;i<n;i++)
  { for(j=0;j<m;j++)
    cout<<a[i][j]<<" ";
    cout<<endl;
  }
  cout<<endl;
}

void sterg_lin(int a[][10],int &n, int m,int l)
{ int i,j; //
  for(i=l+1;i<n;i++) // se deplaseaz? "în sus" liniile de dup? linia de
    for(j=0;j<m;j++) // indice l; o linie se deplaseaz? element cu element
      a[i-1][j]=a[i][j];
  n--;
}
// pentru a insera o linie, în primul rand "i se face loc" deplasând "în jos" toate
// liniile, începând cu ultima, până? la cea de indice l inclusiv, după care se atribuie
// valoarea 0 tuturor elementelor de pe linia l

void ins_lin(int a[][10],int &n, int m,int l)
{ int i,j;
  for(i=n-1;i>=l;i--)
    for(j=0;j<m;j++)
      a[i+1][j]=a[i][j];
  for(j=0;j<m;j++)
    a[l][j]=0;
  n++;
}

void schimb_lin(int a[][10],int n,int m,int l1,int l2)
{ int j,aux;
  for (j=0;j<m;j++)
  {aux=a[l1][j];a[l1][j]=a[l2][j];a[l2][j]=aux;}
}

```

5. Să se verifice dacă o matrice pătratică citită din fișierul text „mat.in” este simetrică față de diagonala principală.

Deoarece de data aceasta discutăm despre o matrice pătratică, atât indicele de linie cât și indicele de coloană iau valori în intervalul  $[0, n-1]$ .

Pentru a verifica simetria matricei, în funcția *simetric* se parcurg elementele matricei situate deasupra diagonalei principale și se verifică egalitatea acestora cu elementele simetrice situate sub diagonala principală; dacă cel puțin un element nu îndeplinește condiția cerută, funcția va returnează valoarea 0, iar dacă toate elementele sunt egale cu simetricele lor funcția va returna valoarea 1.

```

#include<conio.h>           //SIMETRIE
#include<fstream.h>
#include<iostream.h>
void citire(int a[][10],int &n);
void afisare(int a[][10],int n);
int simetrica(int a[][10],int n);

void main()
{ int a[10][10],n;
  cout<<"matricea "<<endl;citire(a,n);
  afisare(a,n);
  if(simetrica(a,n))
    cout<<"matricea este simetrica";
  else
    cout<<"matricea nu este simetrica";
  getch(); }
void citire(int a[][10],int &n)
{int i,j;
 ifstream f("mat.in");
 f>>n;
 for(i=0;i<n;i++)
   for(j=0;j<n;j++)
     f>>a[i][j];
 f.close();}
void afisare(int a[][10],int n)
{int i,j;
 for(i=0;i<n;i++)
   {for(j=0;j<n;j++)
     cout<<a[i][j]<<' ';
   cout<<endl;
 }
cout<<endl;}
int simetrica(int a[][10],int n)
{int i,j;
 for(i=0;i<n;i++)
   for(j=i+1;j<n;j++)
     if(a[i][j]!=a[j][i]) return 0;
return 1;}

```

**Comment [A4]:** Am modificat ordinea functiilor

6. Să se construiască în memorie și să se afișeze un tablou t cu n (n<20) linii și n coloane (n citit de la tastatură), cu elemente numere naturale, astfel încât pe diagonala principală să existe numai elemente egale cu 1, elementele de pe cele două „semidiagonale” paralele cu diagonala principală și alăturate diagonalei principale să fie toate egale cu 2, elementele de pe următoarele „semidiagonale” să fie egale cu 3, etc.

:

(Variantă Bacalaureat, 2002)

**Exemplu:** pentru  $n=4$  se va afișa tabloul

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

Observăm faptul că indicii elementelor au următoarele proprietăți:

- elementele situate pe diagonala principală au  $i-j=0$ ;
- elementele de pe prima „semidiagonală” de deasupra diagonalei principale au  $i-j=-1$
- elementele de pe prima „semidiagonală” de sub diagonala principală au  $i-j=1$ ;
- pentru cea de-a doua „semidiagonală” deasupra diagonalei  $i-j=2$ ,
- pentru cea de-a doua „semidiagonală” sub diagonala  $i-j=2$ .

Putem concluziona că relația dintre valorile elementelor și indicii lor este:

$a[i][j]=abs(i-j)$ .

**Comment [A5]:** Am modificat ordinea funcțiilor în programul de mai jos

```
#include<conio.h>
#include<math.h>
#include<iostream.h>
void afisare(int a[20][20],int n)
void main()
{int a[20][20],n,i,j;
cout<<"n=";cin>>n;
for(i=0;i<n;i++)
for(j=0;j<n;j++)
a[i][j]=abs(i-j)+1;
afisare(a,n);
getch();
}
void afisare(int a[20][20],int n)
{int i,j;
for(i=0;i<n;i++)
{for(j=0;j<n;j++)
cout<<a[i][j]<<' ';
cout<<endl;
}
}
```

7. Din fișierul „mat.in” se citește o matrice cu  $n$  linii și  $m$  coloane, astfel: de pe prima linie se citește  $n$  și  $m$  dimensiunile matricei, apoi de pe  $n$  linii se citesc  $m$  elemente.

Să se copieze elementele matricei, începând cu prima linie, într-un tablou unidimensional.

**Comment [A6]:** Am modificat ordinea functiilor

```
#include<conio.h> //LINIARIZARE
#include<fstream.h>
#include<iostream.h>
#define MAX 10
#define MAX1 100
void citire(int a[][MAX],int &n,int &m);
void afisare_mat(int a[][MAX],int n,int m);
void afisare_vect(int b[],int d);
void liniarizare(int a[][MAX],int b[MAX1],int n,int m);

void main()
{ int a[MAX][MAX],b[MAX1],n,m;
  citire(a,n,m);
  cout<<"matricea a:"<<endl;
  afisare_mat(a,n,m);
  liniarizare(a,b,n,m); //liniarizeaza matricea
  cout<<"vectorul b:"<<endl;
  afisare_vect(b,n*m);
  getch();
}

void citire(int a[][MAX],int &n,int &m)
{ int i,j;
  ifstream f("mat.in");
  f>>n>>m;
  for(i=0;i<n;i++)
    for(j=0;j<m;j++)
      f>>a[i][j];
  f.close();
}

void afisare_mat(int a[][MAX],int n,int m)
{ int i,j;
  for(i=0;i<n;i++)
    {for(j=0;j<m;j++)
      cout<<a[i][j]<<' ';
    cout<<endl;
  }
  cout<<endl;}

void afisare_vect(int b[],int d)
{ int i;
  for(i=0;i<d;i++)
    cout<<b[i]<<' ';
  cout<<endl;}

void liniarizare(int a[][MAX],int b[MAX1],int n,int m)
{ int i,j,k=0;
  for(i=0;i<n;i++)
    for(j=0;j<m;j++)
      b[k++]=a[i][j];
}
```

## 2.5. Evaluare

### TESTUL 1

1. Definiți noțiunea de tablou.
2. Declarați un tablou bidimensional cu numele *matrice*, cu 10 linii și 20 de coloane, care să conțină numere reale.
3. Se citește o matrice cu  $n$  linii și  $m$  coloane, cu elemente numere întregi. Să se determine media elementelor strict pozitive ale matricei.
4. Se citește o matrice cu  $n$  linii și  $m$  coloane din fișierul "*mat.in*" astfel, de pe prima linie se citesc dimensiunile matricei, apoi de pe  $n$  linii se citesc câte  $m$  elemente. Să se determine și să se afișeze valoarea minimă de pe fiecare coloană a matricei.

### TESTUL 2

1. Ce înțelegeți prin matrice pătratică?
2. Care este proprietatea indicilor elementelor situate pe diagonala principală a unei matrice pătratică? Dar a indicilor situați pe diagonala secundară a unei matrice?
3. Să se determine și să se afișeze de valoare minimă de pe diagonala principală a unei matrice pătratică de dimensiune  $n \times n$ , cu elemente reale și indicii acestor elemente.
4. Se citește o matrice  $A$  pătratică de dimensiune  $n \times n$ . Să se construiască matricea  $B$  de dimensiune  $2n \times 2n$  astfel:

$$\begin{pmatrix} A & O \\ I & A' \end{pmatrix}$$

Unde  $O$  este matricea nulă iar  $I$  este matricea unitate.

## 2.6. Probleme propuse

1. Fie  $a$  o matrice cu  $n$  linii și  $m$  coloane.
  - a. Calculați sumele elementelor de pe fiecare linie
  - b. Calculați sumele elementelor de pe fiecare coloană
  - c. Ordonăți crescător elementele de pe ultima coloană prin interschimbări de linii.
  - d. Stergeți o coloană  $c$  a matricei,  $c$  fiind citit de la tastatură
  - e. Interschimbați elementele a două coloane  $c_1$  și  $c_2$ , citite de la tastatură

2. Fie a o matrice pătratică cu  $n$  linii cu elemente întregi.
- Calculați suma elementelor de pe diagonala principală și de pe diagonala secundară
  - Numărați elementele strict pozitive situate deasupra diagonalei secundare
  - Numărați elementele strict negative situate sub diagonala secundară.

3. Scrieți un program care construiește în memorie un tablou  $T$  cu  $n$  linii și  $n$  coloane, cu elemente numere naturale, astfel încât pe ultima coloană și pe ultima linie a tabloului să se afle numai elemente egale cu 1, iar oricare alt element al tabloului să fie suma dintre elementul aflat imediat sub el și elementul aflat imediat în dreapta lui.

Valoarea lui  $n$  ( $n < 20$ ) se citește de la tastatură. Tabloul se va afișa pe ecran în mod standard: de la prima la ultima linie, elementele unei linii fiind scrise de la stânga la dreapta, cu spații între elementele iecărei linii.

Ex: pentru  $n=4$ , se va afișa tabelul următor:

```
20 10 4 1
10 6 3 1
4 3 2 1
1 1 1 1
```

(Variantă Bacalaureat, 2002)

4. Să se interschimbe liniile și coloanele unei matrice pătratice, astfel încât în final elementele situate pe diagonala principală să fie ordonate crescător.

Exemplu:

matricea inițială	matricea ordonare	după
2 9 7	-1 4	1
1 -1 4	6 1	8
8 6 1	9 7	2

5. Din fișierul „mat.in” se citește o matrice cu  $n$  linii și  $m$  coloane, astfel: de pe prima linie se citește  $n$  și  $m$  dimensiunile matricei, apoi de pe  $n$  linii se citesc  $m$  elemente.

Să se copieze elementele matricei, începând cu prima linie, într-un tablou unidimensional, apoi să se ordoneze crescător elementele vectorului.

Să se copieze elementele tabloului în matrice începând cu prima linie a matricei.

6. Din fișierul „mat.in” se citește o matrice cu  $n$  linii și  $m$  coloane, astfel: de pe prima linie se citește  $n$  și  $m$  dimensiunile matricei, apoi de pe  $n$  linii se citesc  $m$  elemente numere reale.

Să se determine maximele elementelor de pe fiecare linie și să se formeze cu ele un tablou unidimensional de numere reale.

7. Să se calculeze media aritmetică a elementelor aflate pe diagonala principală a unei matrice pătratice cu  $n$  elemente numere întregi.

8. Să se formeze un tablou unidimensional cu media aritmetică a elementelor aflate pe fiecare coloană a unei matrice cu  $n$  linii și  $m$  coloane, care conține numere întregi.



9. Să se scrie un program care însumează elementele aflate pe conturul unei matrice cu  $n$  linii,  $m$  coloane și elemente reale.

10. Să se scrie un program care realizează rotirea unei matrice cu 90 de grade la stânga (cu 90 de grade la dreapta).

11. Fie o matrice cu  $n$  linii și  $m$  coloane, ale cărei elemente sunt numere binare. Să se construiască un tablou unidimensional cu  $n$  numere întregi scrise în sistemul de numerație zecimal. Fiecare număr se obține din cele  $m$  cifre binare de pe o linie (o linie se interpretează ca un număr în baza 2 și se convertește în baza 10).

12. Se consideră o matrice care reprezintă un eisberg în plan, codificat astfel: 1 dacă este un „pătrățel” de gheață și 0 dacă este aer. Un pătrățel de gheață se topește într-o unitate de timp numai dacă este înconjurat din trei părți de aer sau dacă se află la marginea eisberg-ului. Să se precizeze în câte unități de timp se topește eisberg-ul.