

CURS 09 – FP

Tablouri bidimensionale (matrice)

Problemă de modelare:

- n magazine identificate prin numerele $0, 1, \dots, n - 1$
- în fiecare magazin sunt aceleași m produse, identificate prin numerele $0, 1, \dots, m - 1$
- vrem să memorăm, pentru fiecare magazin, cantitatea din fiecare produs aflată în magazinul respectiv
- 1 magazin \Leftrightarrow un tablou unidimensional p cu m elemente, în care $p[i]$ ar reprezenta cantitatea din produsul i aflată în magazinul respectiv \Leftrightarrow folosim n tablouri unidimensionale cu m elemente:

```
int p0[100] = {20, 10, 30}, p1[100] = {10, 5, 0}, p2[100] = {15, 15, 5}, p3[100]
= {10, 0, 10};
int p4[100] = {7, 7, 10};
```

```
cantitate_totala_produs_1 = p0[1] + p1[1] + p2[1] + p3[1] + p4[1];
```

Soluția NU este scalabilă!!!

- vom grupa cele n tablouri cu câte m elemente, corespunzătoare celor n magazine, într-un tablou unidimensional cu n elemente, fiecare element fiind un tablou unidimensional cu m elemente, adică o **matrice** / un **tablou bidimensional** cu n linii și m coloane

Exemplu: $n = 4$ magazine și $m = 3$ produse

$$P = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 100 & 25 & 13 \\ 1 & 45 & 100 & 7 \\ 2 & 28 & 25 & 0 \\ 3 & 5 & 21 & 43 \end{pmatrix}$$

$P = ((100, 25, 13), (45, 100, 7), (28, 25, 0), (5, 21, 43))$

$P[2][0] = 28 \Leftrightarrow$ în magazinul 2 sunt 28 kg din produsul 0

$P[i][j]$ = cantitatea din produsul j aflată în magazinul i

- Declararea unui tablou bidimensional:

tip_de_date nume_tablou[nr_max_linii][nr_max_coloane];

Exemplu: `int P[4][3];`

- Accesarea elementului aflat la intersecția dintre linia *i* și coloana *j* (ambele indexate de la 0):

nume_tablou[i][j]

Exemple de modelare cu matrice:

- $P[i][j]$ = media elevului *i* la disciplina *j*
- $P[i][j]$ = nota studentului *i* la examenul *j*
- $P[i][j]$ = valorile asociate unor puncte in plan sau unor parcele pătratice (de exemplu, înălțimi unor puncte pe o hartă sau cantitatea de grâu recoltată de pe parcela respectivă)



- $P[i][j]$ = culoarea RGB a pixelului cu coordonatele (i, j)
- $P[i][j]$ = temperatura din orașul i în ziua j
- $P[i][j]$ = cursul în RON al valutei i în ziua j

Problema 1:

Considerăm o matrice care conține încasările zilnice pentru primele m zile ale unei luni în fiecare dintre cele n magazine ale unui lanț de magazine alimentare. Să se afișeze încasările totale din fiecare magazin, precum și încasările totale din fiecare zi.

Exemplu: $n = 4$ magazine și $m = 3$ zile

$$P = \begin{pmatrix} & \textcolor{red}{0} & \textcolor{red}{1} & \textcolor{red}{2} & \\ \textcolor{blue}{0} & 10000 & 2500 & 1300 & 13800 \\ \textcolor{blue}{1} & 4500 & 10000 & 700 & 15200 \\ \textcolor{blue}{2} & 2800 & 2500 & 0 & 5300 \\ \textcolor{blue}{3} & 5000 & 21000 & 43000 & 69000 \\ & \textcolor{red}{22300} & \textcolor{red}{36000} & \textcolor{red}{45000} & \end{pmatrix}$$

Încasările totale ale magazinului i sunt egale cu suma elementelor de pe linia i !

Încasările totale din ziua j sunt egale cu suma elementelor de pe coloana j !

```
#include <stdio.h>
```

```
int main()
{
```

```
    //n = numarul de magazine (numarate de la 0)
```

```
    //m = numarul de zile (numarate de la 0)
```

```
    int n, m, i, j, s;
```

```
    //consideram ca sunt maxim n = 100 de magazine
```

```
    //si maxim m = 31 de zile intr-o luna
```

```
    int P[100][31];
```

```
    printf("Numarul de magazine: ");
```

```
    scanf("%d", &n);
```

```
    printf("Numarul de zile: ");
```

```
    scanf("%d", &m);
```

```
    printf("\n");
```

```

//citim elementele matricei, unul cate unul, pe linii
for(i = 0; i < n; i++)
{
    printf("Incasarile magazinului %d:\n", i);
    for(j = 0; j < m; j++)
    {
        printf("\tZiua %d: ", j);
        scanf("%d", &P[i][j]);
    }

    printf("\n");
}

//afisam matricea
printf("\n\nSituatia incasarilor din magazine pe zile:\n");
for(i = 0; i < n; i++)
{
    for(j = 0; j < m; j++)
        printf("%6d ", P[i][j]);
    printf("\n");
}

printf("\nIncasarile din fiecare magazin:\n");
for(i = 0; i < n; i++)
{
    s = 0;
    for(j = 0; j < m; j++)
        s = s + P[i][j];
    printf("\tMagazinul %d: %d\n", i, s);
}

printf("\nIncasarile din fiecare zi:\n");
for(j = 0; j < m; j++)
{
    s = 0;
    for(i = 0; i < n; i++)
        s = s + P[i][j];
    printf("\tZiua %d: %d\n", j, s);
}

return 0;
}

```

TABLOURI BIDIMENSIONALE PĂTRATICE

- O *matrice pătratică* este o matrice în care numărul de linii și de coloane este același (de exemplu, pentru $n = 4$):

$$P = \begin{pmatrix} 0 & 0 & 1 & 2 & 3 \\ 0 & P[0][0] & P[0][1] & P[0][2] & P[0][3] \\ 1 & P[1][0] & P[1][1] & P[1][2] & P[1][3] \\ 2 & P[2][0] & P[2][1] & P[2][2] & P[2][3] \\ 3 & P[3][0] & P[3][1] & P[3][2] & P[3][3] \end{pmatrix}$$

- Diagonala principală:** $P[i][j]$ cu $i = j \Leftrightarrow P[i][i]$
- Diagonala secundară:** $P[i][j]$ cu $i+j = n-1 \Leftrightarrow P[i][n-1-i]$

$$P = \begin{pmatrix} 0 & 0 & 1 & 2 & 3 \\ 0 & P[0][0] & P[0][1] & P[0][2] & P[0][3] \\ 1 & P[1][0] & P[1][1] & P[1][2] & P[1][3] \\ 2 & P[2][0] & P[2][1] & P[2][2] & P[2][3] \\ 3 & P[3][0] & P[3][1] & P[3][2] & P[3][3] \end{pmatrix}$$

- Sub diagonala principală:** $P[i][j]$ cu $i > j$
- Deasupra diagonalei principale:** $P[i][j]$ cu $i < j$
- Paralelele sub diagonala principală:** $P[i][j]$ cu $i-j \in \{n-1, \dots, 1\}$
- Paralelele deasupra diagonalei principale:** $P[i][j]$ cu $j-i \in \{1, \dots, n-1\}$

$$P = \begin{pmatrix} 0 & 0 & 1 & 2 & 3 \\ 0 & P[0][0] & P[0][1] & P[0][2] & P[0][3] \\ 1 & P[1][0] & P[1][1] & P[1][2] & P[1][3] \\ 2 & P[2][0] & P[2][1] & P[2][2] & P[2][3] \\ 3 & P[3][0] & P[3][1] & P[3][2] & P[3][3] \end{pmatrix}$$

- Sub diagonala secundară:** $P[i][j]$ cu $i+j > n-1$
- Deasupra diagonalei secundare:** $P[i][j]$ cu $i+j < n-1$
- Paralelele sub diagonala secundară:** $P[i][j]$ cu $i+j \in \{n, \dots, 2n-2\}$
- Paralelele deasupra diagonalei secundare:** $P[i][j]$ cu $i+j \in \{0, \dots, n-2\}$

Problema 2:

Considerăm o matrice pătratică de dimensiune n . Să se afișeze elementele sale pe paralele la diagonala principală, respectiv pe paralele la diagonala secundară.

```
Matricea:
  1    2    3    4
  5    6    7    8
  9   10   11   12
 13   14   15   16

Paralelele la diagonala principala:
13
 9  14
 5  10  15
 1  6  11  16
 2  7  12
 3  8
 4

Paralelele la diagonala secundara:
 1
 5  2
 9  6  3
13 10  7  4
14 11  8
15 12
16
```

```
#include <stdio.h>

int main()
{
    int n, i, j, k;
    int P[100][100];

    printf("Dimensiunea matricei: ");
    scanf("%d", &n);

    printf("\n");

    //citim elementele matricei, unul cate unul, pe linii
    for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
        {
            printf("P[%d][%d] = ", i, j);
            scanf("%d", &P[i][j]);
        }
}
```

```

//afisam matricea
printf("\n\nMatricea:\n");
for(i = 0; i < n; i++)
{
    for(j = 0; j < n; j++)
        printf("%6d ", P[i][j]);
    printf("\n");
}

//afisam elementele pe paralele la diagonala principala
printf("\n\nParalelele la diagonala principala:\n");
for(i = n-1; i >= 0; i--)
{
    for(k = i, j = 0; k < n; k++, j++)
        printf("%6d ", P[k][j]);
    printf("\n");
}

for(j = 1; j < n; j++)
{
    for(i = 0, k = j; k < n; i++, k++)
        printf("%6d ", P[i][k]);
    printf("\n");
}

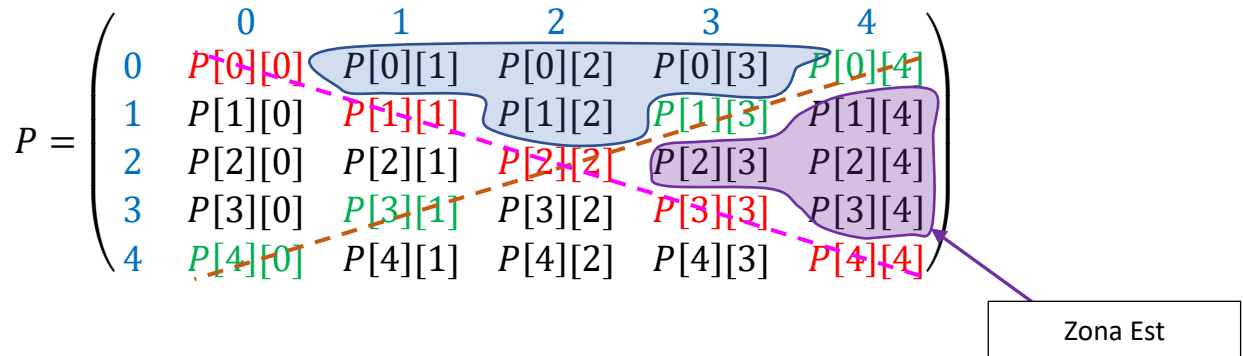
//afisam elementele pe paralele la diagonala secundara
printf("\n\nParalelele la diagonala secundara:\n");
for(i = 0; i < n; i++)
{
    for(k = i, j = 0; k >= 0; k--, j++)
        printf("%6d ", P[k][j]);
    printf("\n");
}

for(j = 1; j < n; j++)
{
    for(i = n-1, k = j; k < n; i--, k++)
        printf("%6d ", P[i][k]);
    printf("\n");
}

return 0;
}

```

Zonele unei matrice pătrate



- **Zona Nord:**
 - deasupra diagonalei principale și deasupra diagonalei secundare
 - $P[i][j]$ cu $i < j$ și $P[i][j]$ cu $i+j < n-1$
- **Zona Est:**
 - deasupra diagonalei principale și sub diagonalei secundare
 - $P[i][j]$ cu $i < j$ și $P[i][j]$ cu $i+j > n-1$
- **Zonele Sud și Vest: ...**