

## Матрици

1. Дадена е квадратна матрица  $A=[a_{ij}]N \times N$ . Да се напише програма која ќе ја пресмета вредноста:

$$f = \prod_{i=1}^n \left[ \prod_{j=1}^n a_{ij} + \prod_{j=1}^n a_{ji} \right]$$

```
#include "stdio.h"
#define Max 100
int main() {

    int n, i, j;
    float a[Max][Max];
    float first, second, product;

    printf("Vnesi red na matricata N=");
    fflush(stdout);
    scanf("%d", &n);
    printf("Vnesi ja matricata...\n");
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
        {
            printf("a[%d,%d]=", i, j);
            fflush(stdout);
            scanf("%f", &a[i][j]);
        }
    product=1;

    for (i=0; i<n; i++)
    {
        first=1;
        second=1;
        for (j=0; j<n; j++)
        {
            first *= a[i][j];
            second *= a[j][i];
        }
        product*=(first+second);
    }

    printf("Presmetanata vrednost f=%f\n", product);
    return 0;
}
```

2. Квадратна матрица со ист број редици и колони се нарекува симетрична ако  $A_{ij} = A_{ji}$  за секое  $i \neq j$ . На пример:

$$\begin{bmatrix} 0 & 3 & 5 & 1 \\ 3 & 7 & 6 & 9 \\ 5 & 6 & 2 & 4 \\ 1 & 9 & 4 & 2 \end{bmatrix}$$

Да се напише програма која проверува дали матрицата A NxN е симетрична.

```
#include "stdio.h"
#define Max 100
int main()
{
    int n,i,j,s;
    float a[Max][Max];

    printf("Vnesi red na matricata N=");
    fflush(stdout);
    scanf("%d", &n);
    printf("Vnesi ja matricata...\n");
    fflush(stdout);
    for (i=0; i<n; i++)
        for (j=0; j<n; j++){
            printf("a[%d,%d]= ", i, j);
            fflush(stdout);
            scanf("%f", &a[i][j]);
        }

    s=1;
    for (i=0; i<n-1; i++)
        for (j=i+1; j<n; j++)
            if (a[i][j] != a[j][i])
                s=0;

    if (s)
        printf("Matncata e simetricna\n");
    else
        printf("Matricata ne e simetricna\n");

    return 0;
}
```

3. Да се напише програма која елементите на матрица  $A_{m \times n}$  огледално ги пресликува во однос на хоризонталата.

$$\begin{bmatrix} 2 & 3 \\ 1 & 7 \\ 0 & 5 \\ 4 & 6 \end{bmatrix} \Rightarrow \begin{bmatrix} 4 & 6 \\ 0 & 5 \\ 1 & 7 \\ 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 5 \\ 4 & 3 & 8 \\ 9 & 0 & 6 \end{bmatrix} \Rightarrow \begin{bmatrix} 9 & 0 & 6 \\ 4 & 3 & 8 \\ 1 & 2 & 5 \end{bmatrix}$$

```

#include <stdio.h>
#define Max 100

int main()
{
    int m,n,i,j;
    float t, a[Max][Max];

    printf("Vnesi gi dimenziite na matricata\n");
    fflush(stdout);
    scanf("%d %d", &m, &n);
    printf("Vnesi ja matricata...\n");
    fflush(stdout);
    for (i=0; i<m; i++)
        for (j=0; j<n; j++){
            printf("a[%d,%d]= ", i, j);
            fflush(stdout);
            scanf("%f", &a[i][j]);
        }

    for (i=0; i<m/2; i++)
        for (j=0; j<n; j++) {
            t=a[i][j];
            a[i][j]=a[m-i-1][j];
            a[m-i-1][j]=t;
        }

    for (i=0; i<m; i++){
        printf("\n");
        for (j=0; j<n; j++)
            printf("a[%2d,%2d]=%5.2f\t", i, j, a[i][j]);
    }

    return 0;
}

```

4. Да се напише програма која елементите од дадена матрица ќе ги помести на следниот начин:

Да не се користи if.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \Rightarrow \begin{bmatrix} 9 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}$$

```

#include <stdio.h>
#define Mx 100

int main(){
    int m,n,i,j,k;
    float a[Mx][Mx], t;

    printf("Vnesi gi dimenziite na matricata\n");
    fflush(stdout);
    scanf("%d %d", &m, &n);
    printf("Vnesi ja matricata...\n");
    fflush(stdout);
    for (i=0; i<m; i++)
        for (j=0; j<n; j++) {

```

```

        printf("a[%d,%d]=" , i, j);
        fflush(stdout);
        scanf("%f", &a[i][j]);
    }

    t=a[m-1][n-1];
    for(k=n*m-1; k>0; k--)
        a[k/n][k%n]=a[(k-1)/n][(k-1)%n];

    a[0][0]=t;

    for (i=0; i<m; i++){
        printf("\n");
        for (j=0; j<n; j++)
            printf("%5.2f\t", a[i][j]);
    }

    return 0;
}

```

(2D) матрица  $[m \times n] \rightarrow$  (1D) поле  $[m*n]$        $M[i,j] \leftrightarrow P[k]$

$i,j \rightarrow k: \quad k=i*n+j$

$k \rightarrow m,n: \quad i=((k-1)/n)$

$j=((k-1) \% n)$

5. Да се напише програма за работа со квадратни матрици (најмногу 100x100). Од тастатура се чита димензијата на квадратната матрица и потоа се внесуваат елементите на матрицата. Програмата треба да провери дали матрицата го задоволува следниов услов: секој елемент од **споредната** дијагонала да биде **еднаков** на збирот од елементите лево (во истата редица) од тој елемент. На екран да се испечати порака дали матрицата го задоволува условот.

1	2	3	6
2	5	7	7
8	8	9	1
6	0	1	8

1	2	3	⇒ 6
2	5	⇒ 7	7
8	⇒ 8	9	1
6	0	1	8

```

#include <stdio.h>
#define MAX 100

int main(int argc, char *argv[])
{
    int A[MAX][MAX], i, j, n, suma, prov=1;
    printf("Vnesi go redot na kvadratnata matrica");
    scanf("%d", &n);
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            printf("mat[%d][%d]=" , i, j);
            scanf("%d", &A[i][j]);
        }
    }

    for(i=0; i<n-1; i++){
        suma=0;
        for(j=0; (i+j)<n-1; j++){
            suma+=A[i][j];
        }
    }
}

```

```

    }
    if(suma!=A[i][j]) {
        prov=0;
        break;
    }
}
if(prov)
    printf("Matricata go zadovoluva uslovot\n");
else
    printf("Matricata NE go zadovoluva uslovot\n");

return 0;
}

```

6. Да се напише програма која чита матрица со целобројни елементи. Димензиите на матрицата се внесуваат од тастатура, максимални димензии се 10x10. Програмата треба да ги прикаже на екран оние елементи од матрицата кои се појавуваат во секој ред. Доколку не постојат такви броеви, на екран да се прикаже порака за грешка.

Пример:

$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 1 & 7 \\ 8 & 2 & 9 & 1 \\ 6 & 0 & 1 & 8 \end{bmatrix}$	На екран ќе се прикаже пораката: Elementot 1 go zadovoluva uslovot
--	--

```

#include <stdio.h>
#define MAX 10
int main(){
    int A[MAX][MAX], i, j, k, m, n, prov, kolku;

    printf("Vnesi go redot na matricata");
    scanf("%d%d", &m, &n);
    for(i=0; i<m; i++){
        for(j=0; j<n; j++){
            printf("mat[%d][%d]= ", i, j);
            scanf("%d", &A[i][j]);
        }
    }

    kolku = 0;
    for(k=0; k<n; k++){
        for(i=1; i<m; i++){
            prov=0;
            for(j=0; j<n; j++){
                if(A[0][k]==A[i][j]) {prov=1; break;}
            } //for(j)
            if(prov==0)
                break;
        } //for(i)
        if(prov==1) {
            printf("Elementot %d go zadovoluva uslovot\n", A[0][k]);
            kolku++;
        }
    } //for(k)

    if(kolku==0)
        printf("Greska!\n");

    return 0;
}

```