

4. Да се напише програма која за дадена низа A од реални броеви и N , број на елементи во A , ќе ја пресмета најголемата разлика меѓу два соседни елементи од оваа низа.

```
#include "stdio.h"
#include "math.h"
#define MAXELEM 10
void main(void)
{
    float a[MAXELEM];
    int i,n;
    float raz,maxr;

    printf("Vnesi kolku elementi ke ima nizata A\n");
    scanf("%d",&n);
    printf("Vnesi gi elementite na nizata A\n");
    for (i=0; i<n; i++)
    {
        printf("A[%d]=",i);
        scanf("%f",&a[i]);
    }
    maxr=fabs(a[1]-a[0]);
    for (i=2; i<n; i++)
    {
        raz=fabs(a[i]-a[i-1]);
        if(raz>maxr)
            maxr=raz;
    }
    printf("Najgolemata razlika medju dva sosedni elementi
e:%f\n",maxr);
}
```

5. Да се напише програма која врши табелирање на полиномот $P(x)$ во интервалот $[a,c]$, со чекор $(c-a)/m$.

$$P(x) = b_0 + b_1x + b_2x^2 + K + b_nx^n$$

$$P(x) = b_0 + x(b_1 + x(b_2 + K + x(b_{n-1} + xb_n)K))$$

```
#include "stdio.h"

#define MAXELEM 100

void main(void)
{
    float b[MAXELEM];
    int i,n,j,m,a,c;
    float x,cekor,clen;

    printf("Red na polinomot:"); scanf("%d",&n);
    printf("Koeficienti na polinomot\n");
```

```
    for (j=0; j<n; j++)
        scanf("%f", &b[j]);
    printf("Interval\n");
    scanf("%d %d", &a, &c);
    scanf("%d", &m);
    cekor=(float) (c-a)/m;
    x=(float) a;
    while (x <= c) /* ili for (x=(float)a; x<=c; x+=cekor)
    {
        clen=b[n-1];
        for (i=n-2; i>=0; i--)
            clen=b[i]+x*clen;
        printf("x=%f P(x)=%f\n", x, clen);
        x+=cekor;
    }
}
```

8 МАТРИЦИ

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

$$f = \prod_{i=0}^{n-1} \left[\prod_{j=0}^{n-1} a_{ij} + \prod_{j=0}^{n-1} a_{ji} \right]$$

```
#include "stdio.h"

#define MAX 100

void main(void)
{
    int n,i,j;
    float a[MAX][MAX];
    float first,second,product;

    printf("Vnesi red na matricata N=");
    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);
            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);
}
```

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 $A_{N \times N}$ е симетрична.

```
#include "stdio.h"

#define MAX 100

void main(void)
{
    int n,i,j,s;
    float a[MAX][MAX];

    printf("Vnesi red na matricata N=");
    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);
            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");
}
```

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

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

    printf("Vnesi gi dimenziite na matricata\n");
    scanf("%d %d",&m, &n);
    printf("Vnesi ja matricata...\n");
    for (i=0; i<m; i++)
        for (j=0; j<n; j++)
        {
            printf("a[%d,%d]=",i,j);
            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-1-i][j];
            a[m-1-i][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]);
    }
}
```

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

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

Да не се користи if.

```
#include "stdio.h"
#define MAX 100

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

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

(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. Дадена е квадратна матрица A. Да се напише програма која го пресметува збирот на елементите:

- од, над и под главната дијагонала;
- од, над и под споредната дијагонала.

```
#include "stdio.h"
#define MAX 100
```

```
void main(void)
{
    int n,a[MAX][MAX], i, j;
    int gOd,gNad,gPod,sOd,sNad,sPod;

    gOd=gNad=gPod=sOd=sNad=sPod=0;

    printf("Dimenzija: ");
    scanf("%d",&n);
    printf("Vnesi gi elementite\n");
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            scanf("%d",&a[i][j]);

    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
        {
            if (i<j) gNad+=a[i][j];
            else if (i>j) gPod+=a[i][j];
            else gOd+=a[i][j];

            if (i+j<n-1) sNad+=a[i][j];
            else if (i+j>n-1) sPod+=a[i][j];
            else sOd+=a[i][j];
        }
    printf("Glavna dijagonala - nad %d od %d pod %d\n",gNad,gOd,gPod);
    printf("Sporedna dijagonala - nad %d od %d pod %d\n",sNad,sOd,sPod);
}
```

6. Да се напише логичка функциска потпрограма која ќе провери дали матрицата $X=AB+C$ е долнотриаголна. Матриците $A_{n \times k}$, $B_{k \times n}$ и $C_{n \times n}$ се влезни параметри во функцијата. Напомена: Сите елементи над главната дијагонала на долнотриаголна матрица се еднакви на нула.

```
#include "stdio.h"
#define MAX 100

void main(void)
{
    int
    a[MAX][MAX],b[MAX][MAX],c[MAX][MAX],x[MAX][MAX],i,j,l,k,n,dt;

    printf("Dimenzii: ");
    scanf("%d %d",&n, &k);
    printf("Vnesi gi elementite na A\n");
    for (i=0; i<n; i++)
        for (j=0; j<k; j++)
```

```
        scanf("%d",&a[i][j]);
printf("Vnesi gi elementite na B\n");
for (i=0; i<k; i++)
    for (j=0; j<n; j++)
        scanf("%d",&b[i][j]);
printf("Vnesi gi elementite na C\n");
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        scanf("%d",&c[i][j]);

// mnozi
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
    {
        x[i][j]=0;
        for (l=0; l<k; l++)
            x[i][j]+=a[i][l]*b[l][j];
    }
// soberi
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        x[i][j]+=c[i][j];
// dolno triagolna
dt=1;
for (i=0; i<n-1; i++)
    for (j=i+1; j<n; j++)
        if (x[i][j]) dt=0;

if (dt)
    printf("matricata x e dolnotriagolna\n");
else printf("matricata x ne e dolnotriagolna\n");
}
```