

Osnovi programiranja



2024/25



Matrice



Matrica

- Matrica, ili dvodimenzionalni nizovi dimenzijsi $m \times n$ je šema brojeva koja se zapisuje kao:

$$A = \begin{bmatrix} a_{00} & a_{01} & \dots & a_{0n-1} \\ a_{10} & a_{11} & \dots & a_{1n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m-10} & a_{m-11} & \dots & a_{m-1n-1} \end{bmatrix}$$

Diagram illustrating the structure of a matrix A . The matrix is shown as a grid of elements a_{ij} . Arrows point from the labels "vrste" (rows) and "kolone" (columns) to the appropriate parts of the matrix. "Vrste" points to the vertical arrangement of rows, and "kolone" points to the horizontal arrangement of columns.

- Matrica kod koje je $m = n$ zove se kvadratna matrica.
- Elementu i -te vrste i j -te kolone se pristupa korišćenjem zapisa $a[i][j]$.

Matrica

- S obzirom da se u većini programskih jezika elementi matrice u memoriji pakuju vrsta po vrsta, uobičajeno je da prvi indeks predstavlja vrstu, a drugi kolonu. Vrste u matricama često nazivamo i redovima.

`typedef int matrica[10][3];`

`matrica a;`

`ili direktno kao`

`ili`

`int a[10][3];`

- Na sličan način mogu se definisati i višedimenzionalni nizovi. Tako, na primer, možemo definisati i trodimenzionalni niz koji bi predstavljao niz matrica. Sledeći kod pokazuje kako bi se definisao niz od 5 matrica dimenzija 10x3:

`int a[5][10][3];`

`a[t][i][j]` – t je redni broj matrice, i redni broj vrste, j redni broj kolone

Primer 1

- Napisati program koji učitava matricu realnih brojeva a dimenzija $m \times n$ i računa sumu svih elemenata matrice.

```
#include <stdio.h>
main()
{
    float a[100][100];
    int m, n, i, j;
    float suma;
    printf("Unesite broj vrsta i kolona:\n");
    scanf("%d%d", &m, &n);
    printf("Unesite elemente u matricnom obliku:\n");
    for(i = 0; i < m; i++)
        for(j = 0; j < n; j++)
            scanf("%f", &a[i][j]);
```

Primer 1

```
suma = 0.0;
for(i = 0; i < m; i++)
    for(j = 0; j < n; j++)
        suma += a[i][j];

printf("Suma elemenata matrice je %f\n", suma);
}
```

Primer 2

- Napisati program koji pozivanjem odgovarajućeg potprograma vrši transponovanje kvadratne matrice celih brojeva a dimenzija $n \times n$. Da se podsetimo, transponovanje matrice podrazumeva zamenu vrsta i kolona, tj. zamenu mesta elementima $a[i][j]$ i $a[j][i]$, za svako i i j .

```
#include <stdio.h>
typedef float matrica[100][100];
void transponuj(matrica a, int n)
{
    int i, j;
    float pom;
    for(i = 0; i < n; i++)
        for(j = 0; j < i; j++)
    {
        pom = a[i][j];
        a[i][j] = a[j][i];
        a[j][i] = pom;
    }
}
```

Primer 2

```
void stampaj(matrica a, int n)
{
    int i, j;
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
            printf("%10.2f", a[i][j]);
        printf("\n");
    }
}
```

```
main()
{
    matrica a;
    int n, i, j;
    printf("Unesite velicinu matrice:\n");
    scanf("%d", &n);
    printf("Unesite elemente u matricnom obliku:\n");
    for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
            scanf("%f", &a[i][j]);
    transponuj(a, n);
    stampaj(a, n);
}
```

Primer 3

- Napisati funkciju koja transformiše datu matricu celih brojeva A , dimenzija mxn ($n, m \leq 100$) tako što izbacuje:
 - a) sve vrste od k -te do l -te, pri čemu je $0 \leq k \leq l < m$
 - b) sve kolone od k -te do l -te, pri čemu je $0 \leq k \leq l < n$

```
#include <stdio.h>
typedef float matrica[100][100];
void unos(matrica a, int m, int n)
{
    int i, j;
    for(i = 0; i < m; i++)
        for(j = 0; j < n; j++)
            scanf("%f", &a[i][j]);
}
void stampaj(matrica a, int m, int n)
{
    int i, j;
    for(i = 0; i < m; i++)
    {
        for(j = 0; j < n; j++)
            printf("%10.2f", a[i][j]);
        printf("\n");
    }
}
```

```
int izbaci_vrste(matrica a, int m, int n, int k, int l) {  
    int i,j;  
    // Pomeri preostale vrste na njihovo novo mesto  
    for ( i = l + 1; i < m; i++) {  
        for ( j = 0; j < n; j++) {  
            a[i - (l - k + 1)][j] = a[i][j];  
        }  
    }  
    return m-(l - k + 1); // Ažuriraj broj vrsta  
}  
int izbaci_kolone(matrica a, int m, int n, int k, int l) {  
    int i,j;  
    // Pomeri preostale kolone na njihovo novo mesto  
    for ( i = 0; i < m; i++) {  
        for ( j = l + 1; j < n; j++) {  
            a[i][j - (l - k + 1)] = a[i][j];  
        }  
    }  
    return n-(l - k + 1); // Ažuriraj broj kolona  
}
```

```
main()
{
    matrica a;
    int m, n, i, j;
    int k=1,l=2;
    printf("Unesite velicinu matrice:\n");
    scanf("%d%d", &m,&n);
    printf("Unesite elemente u matricnom obliku:\n");
    unos(a, m, n);
    stampaj(a, m, n);
    printf("Nakon izbacivanja vrsta:\n");
    m=izbaci_vrste(a,m,n,k,l);
    stampaj(a, m, n);
    printf("Nakon izbacivanja kolona:\n");
    k=0;l=2;
    n=izbaci_kolone(a,m,n,k,l);
    stampaj(a, m, n);
}
```