

Појам дводимензионог низа

До сада смо имали прилике да се упознамо са једнодимензионим низовима чији су елементи скаларне величине (цели бројеви, реални бројеви, знакови,..). Овакви низови се могу шематски приказати као хоризонтална или вертикална листа података, што можемо видети на следећим сликама.



x
4
8
3
11
6
5
5
17
13
6

Слика 1. Колона

x									
4	8	3	11	6	5	5	17	13	6

Слика 2. Врста

Међутим, у великом броју реалних проблема је погодније податке представити у табеларној форми.

	x		
x[0]	4	12	3
x[1]	8	2	5
x[2]	3	2	7
x[3]	11	5	2
x[4]	6	1	6
x[5]	5	4	5
x[6]	5	9	10
x[7]	17	12	7
x[8]	13	13	1
x[9]	6	8	11

Слика 3. Низ низова

	x		
	4	12	3
	8	2	5
	3	2	7
	11	5	2
	6	1	6
	5	4	4
	5	9	10
	17	12	7
	13	13	1
	6	8	11

Слика 4. Матрица

Табела на слици 3, висине 10 и ширине 3 садржи 10 врста и 3 колоне.

Произвољна табела висине t и ширине n , може се представити као низ дужине t чији су **елементи низови** дужине n , у ствари имамо t врста и n колоне. Зато дводимензионе низове најчешће називамо **матрице** (слика 4).

Сада се можда питате како би могли да приступимо одређеном елементу матрице?!

Појединачној врсти се може приступити коришћењем записа $aMatrixa[i]$, где је i редни број врсте.

Елементу у i -тој врсти и j -тој колони се може приступити коришћењем записа $aMatrixa[i, j]$.

У случају да наша матрица има велики број елемената најбоље је унети вредности са тастатуре у једној *for* петљи.



```
//Definisemo matricu intMatrixa dimenzija 3x4
int [ , ] intMatrixa = new int[3, 4];
int i,j;

Console.WriteLine("Unesi clanove matrice:");

/*Spoljasnja for petlja prolazi po vrstama. Svaka vrsta po i predstavlja jedan niz,
ciji se elementi unose po j.*/
for (i = 0; i < 3; i++)
{
    /*Unutrasnja for petlja prolazi po kolonama. Kada je jedna vrsta fiksirana, npr.
za i=0,
unose se elementi po promenljivoj j, npr. intMatrixa[0,j]*/
    for (j = 0; j < 4; j++)
    {
        /*Sledeca naredba ispisuje naziv svakog elementa pojedinačno,pre ispisa
vrednosti
intMatrixa[i,j]*/
        Console.Write("clan[" + i + "," + j + "] = ");
        intMatrixa[ i, j ] = int.Parse(Console.ReadLine());
    }
}

//Ispisujemo unete clanove po vrstama i kolonama
for(i = 0; i < 3; i++)
{
    for (j = 0; j < 4 ; j++)
    {
        //Navodnici ostavljaju prazan prostor posle svakog unetog elementa
        Console.Write(" " + intMatrixa[i ,j]);
    }
    //Kada se ispise jedna vrsta, prelazi se u sledecu i tako se formira matricni
oblik
    Console.WriteLine();
}
}
```

Стартовањем овог програма и уносом вредности чланова добија се конзолни екран приказан на следећој слици:

```
Unesi članove matrice:
clan[0,0] = 1
clan[0,1] = 2
clan[0,2] = 3
clan[0,3] = 4
clan[1,0] = 5
clan[1,1] = 6
clan[1,2] = 7
clan[1,3] = 8
clan[2,0] = 9
clan[2,1] = 10
clan[2,2] = 11
clan[2,3] = 12
 1  2  3  4
 5  6  7  8
 9 10 11 12
Press any key to continue . . .
```

Слика 6. Конзола

Напомена: Метод *GetLength* има увек повратну вредност типа *int* и враћа број елемената некога низа. У нашој итерацији учествује при провери услова да се бројачи увек крећу по индексима редова и врста матрице.



Пример 1. Написати програм који учитава матрицу целих бројева a димензије $m*n$ и рачуна суму свих елемената матрице.

Решење:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace suma_elemenata_matrice
{
    class Program
    {
        static void Main(string[] args)
        {
            //Rezervisemo proizvoljan broj vrsta i kolona u memoriji
```

```

int[,] a = new int[ 30,40 ];
int i, j;

//m i n kao dimenzije matrice
int m, n;
int s = 0;

//Unosimo dimenzije matrice
Console.WriteLine("Unesite broj vrsta:");
m = int.Parse(Console.ReadLine());

Console.WriteLine("Unesite broj kolona:");
n = int.Parse(Console.ReadLine());

//Unosimo elemente matrice po vrstama
Console.WriteLine("Unesi clanove matrice:");

for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        Console.Write("clan[" + i + "," + j + "] = ");
        a [i, j] = int.Parse(Console.ReadLine());
    }
}

//Racunamo sumu svih elemenata matrice
for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        //Iteracijom po promenljivoj j kroz svaku i-tu vrstu dodajemo
        elemente u zbir s
        s = s + a[i, j];
    }
}

Console.WriteLine("Suma elemenata matrice je:" + s);
}
}
}

```

Пример 2. Написати програм који учитава квадратну матрицу a целих бројева димензије $n*n$ и израчунава највећи и најмањи елемент у матрици као и збир елемената на главној дијагонали.

Напомена: Главну дијагоналу матрице чине елементи на дијагонали која спаја горњи леви и доњи десни угао матрице.



Решење:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace max_min_matrice
{
    class Program
    {
        //Funkcija koja racuna min i max matrice
        static void min_max(int[,] a, int n)
        {
            int i, j;
            //Postavljamo promenljive max i min na pocetne vrednosti
            //Element a[0,0] se nalazi u gornjem levom uglu date matrice a
            int max = a[0, 0];
            int min = a[0, 0];
            //Spoljasnjom for petljom prolazimo kroz matricu a po vrstama
            for (i = 0; i < n; i++)
            {
                //Unutrasnjom for petljom prolazimo kroz matricu a po kolonama
                for (j = 0; j < n; j++)
                {
                    //Proveravamo da li je a[i,j] manji od tekuceg minimuma
                    if (a[i, j] < min)
                    {
                        //Ako je manji, onda njegovu vrednost dodeljujemo promenljivoj
                        min = a[i, j];
                    }
                    /*U slucaju da je a[i,j] veci od tekuceg maksimuma, njegovu
                    vrednost dodeljujemo
                    promenljivoj max*/
                    if (a[i, j] > max)
```

```

        {
            max = a[i, j];
        }
    }
}

Console.WriteLine("Najveci element matrice je:", max, "a najmanji je:",
min);
}

//Funkcija dijagonala izracunava zbir elemenata matrice na dijagonali
static void dijagonala(int[,] a, int n)
{
    int i, suma = 0;

    //Kako je i=j dovoljna nam je jedna for petlja po vrstama i kolonama
matrice a
    for (i = 0; i < n; i++)
    {
        //Clanovi na dijagonali su oblika a[i,i]
        suma += a[i, i];
    }

    Console.WriteLine("Suma elemenata na glavnoj dijagonali je {0}", suma);
}

static void Main(string[] args)
{
    int[,] aMatrica = new int[10, 10];
    int i, j, max, min, dim;

    Console.WriteLine("Unesite dimenziju matrice");
    dim = int.Parse(Console.ReadLine());

    Console.WriteLine("Unesite elemente matrice");
    for (i = 0; i < dim; i++)
    {
        for (j = 0; j < dim; j++)
        {
            Console.Write("clan[", i, j, "] = ");
            aMatrica[i, j] = int.Parse(Console.ReadLine());
        }
    }

    //Standardno ispisujemo elemente unete matrice iteracijom po vrstama i
kolonama
    for (i = 0; i < dim; i++)
    {
        for (j = 0; j < dim; j++)

```



```

    {
        Console.WriteLine(aMatrica[i, j]);
    }
    Console.WriteLine('\n');
}

//Pozivamo funkciju koja racuna min i max matrice aMatrica
min_max(aMatrica, dim);

//Pozivamo funkciju koja racuna zbir elemenata na dijagonali matrice
aMatrica

//Ispis rezultata se vrši pri pozivu funkcija
dijagonala(aMatrica, dim);

}

}

}

```

Пример 4. Написати програм који учитава квадратну матрицу A димензије $n*n$ и транспонује унету матрицу.

Напомена: Транспонована матрица се добија када одговарајуће врсте и колоне замене места.



Решење:

```

using System;
using System.Collections.Generic;
using System.Text;

namespace kvadrat_matrice
{
    class Program
    {
        //Funkcija kvadriraj izracunava vrednost matrice A*A
        static void kvadriraj(int[,] A, int n)
        {
            /*Definisemo kvadratnu matricu B dimenzija n, koja ce predstavljati
            rezultat kvadriranja matrice A*/
            int[,] B = new int[n, n];
            int i, j, k;

            //Matricu B dobijamo tako sto mnozimo odgovarajuce vrste i kolone matrice
            A
            for (i = 0; i < n; i++)
                for (j = 0; j < n; j++)

```

```

    {
        /*Pocetnu vrednost elementa matrice B na preseku i-te vrste i j-te
kolone
        postavljamo na 0, kako tekuci element dobijamo sabiranjem n
        proizvoda
        elemenata matrice A*/
        B[i, j] = 0;
        /*Pomocna for petlja po promenljivoj k omogucava da se istovremeno
pomnoze
        elementi u i-toj vrsti i j-toj koloni matrice A*/
        for (k = 0; k < n; k++)
        {
            /*Proizvoljni element B[i,j] dobijamo tako sto medjusobno
pomnozimo elemente
            i-te vrste i j-te kolone matrice A*/
            B[i, j] += A[i, k] * A[k, j];
        }
    }
    //Ispisujemo elemente rezultujuce matrice B
    Console.WriteLine("Matrica B = A*A:");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            Console.Write(" " + B[i, j]);
        }
        Console.WriteLine();
    }
}

static void Main(string[] args)
{
    //Definisemo matricu A proizvoljno izabrane dimenzije
    int[,] A = new int [20,20];
    int dim, i, j;
    Console.WriteLine("Unesite dimenziju kvadratne matrice");
    dim = int.Parse(Console.ReadLine());

    //Unosimo elemente matrice A
    for (i = 0; i < dim; i++)
    {
        for (j = 0; j < dim; j++)
        {
            Console.Write("clan[" + i + "," + j + "] = ");

```

```

        A[i, j] = int.Parse(Console.ReadLine());
    }
}

//Pozivamo funkciju kvadriraj koja ce ispisati dobijenu kvadriranu
matricu A*A
    kvadriraj(A,dim);
}
}
}

```

Пример 5. Написати програм који учитава матрицу димензије $n*m$, сортира елементе у колонама и врстама унете матрице и исписује тако сортиране матрице.

Решење:

```

using System;
using System.Collections.Generic;
using System.Text;

namespace sortiranje_elemenata
{
    class Program
    {
        static void sortiranje_kolona(int[,] matrica, int n, int m)
        {
            int i, j;
            int pom, t;
            for (j = 0; j < m; j++)
            {
                //Sortiramo kolone uz pomoc insertion sort algoritma koji nam je
                poznat od ranije
                for (i = 0; i < n; i++)
                {
                    t = i;
                    while (t > 0 && matrica[t, j] < matrica[t-1, j])
                    {
                        pom = matrica[t, j];
                        matrica[t, j] = matrica[t - 1, j];
                        matrica[t - 1, j] = pom;
                        t--;
                    }
                }
            }
        }
    }
}

```

```

        //Ispisujemo matricu ciji su elementi u svakoj koloni sortirani
        Console.WriteLine("Matrica cije su kolone sortirane");
        for (i = 0; i < n; i++)
        {
            for (j = 0; j < m; j++)
            {
                Console.Write(matrica[i, j]);
            }
            Console.WriteLine('\n');
        }
    }

    static void sortiranje vrsta(int[,] matrica, int n, int m)
    {
        int i, j, t;
        int pom;
        for (i = 0; i < n; i++)
        {
            for (j = 0; j < m; j++)
            {
                t = j;
                while(t > 0 && matrica[i, t]< matrica[i, t-1])
                {
                    //Clan na poziciji [i,t] menja mesto sa clanom na poziciji
                    [i,t-1]
                    pom = matrica[i, t];
                    matrica[i,t] = matrica[i, t-1];
                    matrica[i, t-1] = pom;
                    t--;
                }
            }
        }

        //Ispisujemo matricu kod koje su elementi u svakoj vrsti sortirani
        Console.WriteLine("Matrica cije su vrste sortirane");
        for (i = 0; i < n; i++)
        {
            for (j = 0; j < m; j++)
            {
                Console.Write(matrica[i, j]);
            }
            Console.WriteLine('\n');
        }
    }

    static void Main(string[] args)

```

```

{
    int[,] A = new int[50, 50];
    int i, j, m, n;
    Console.WriteLine("Unesite broj vrsta matrice");
    n = int.Parse(Console.ReadLine());
    Console.WriteLine("Unesite broj kolona matrice");
    m = int.Parse(Console.ReadLine());
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
        {
            Console.Write("clan[{0},{1}] = ", i, j);
            A[i, j] = int.Parse(Console.ReadLine());
        }
    }
    sortiranje_vrsta(A, n, m);
    sortiranje_kolona(A, n, m);
}
}
}

```