

Osnove programiranja

Nizovi

The slide features a light blue gradient background. The main title 'Osnove programiranja' is centered in a large, dark blue serif font. Below it, the word 'Nizovi' is centered in a smaller, black serif font. On the right side of the slide, there are several thick, light gray wavy lines that curve downwards and to the right, creating a decorative, abstract pattern.

Sadržaj

- Definicija niza
- Vrste i elementi nizova
- Deklarisanje nizova
- Dodele (početne) vrednosti nizovima
- Petlja foreach – štampanje niza
- Jednodimenzionalni i višedimenzionalni nizovi

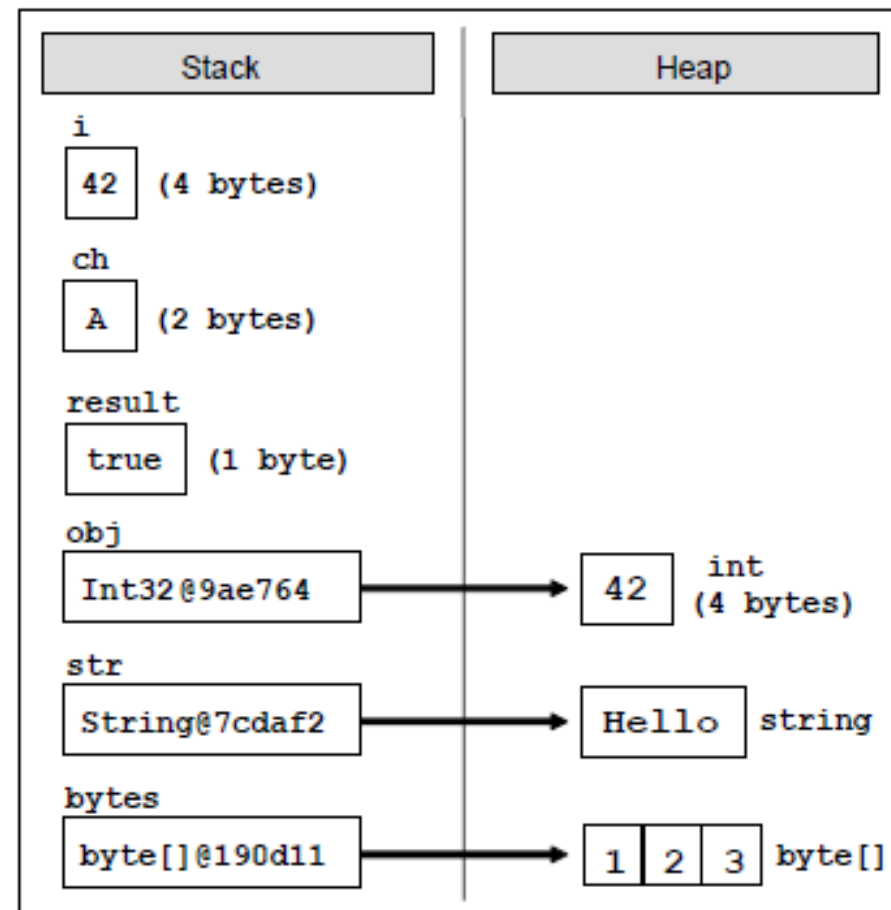
Nizovi - definicija

- Potreba za postojanjem većeg broja podataka istog tipa koje predstavljaju jednu celinu.
- Uvodi pojam niza ili u opštem slučaju pojam polja.
- Niz (eng. array) je indeksirani skup objekata koji su **istog** tipa.

Vrednosne i referentne promenljive

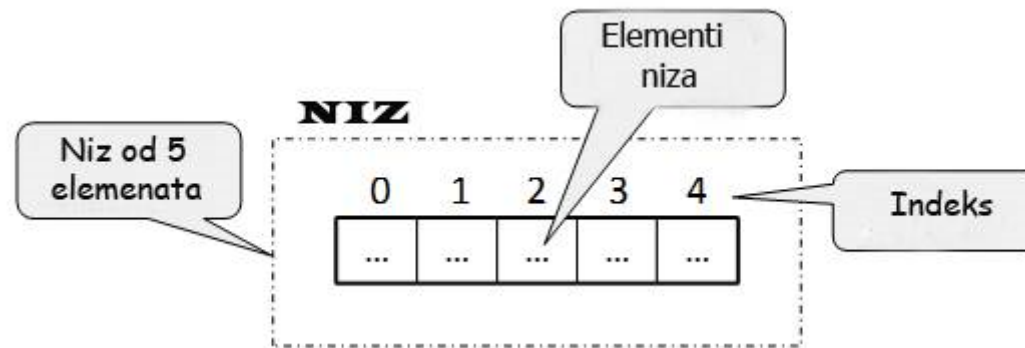
```
int i = 42;  
char ch = 'A';  
bool result = true;  
object obj = 42;  
string str = "Hello";  
byte[] bytes = { 1, 2, 3 };
```

Nizovi su uvek referentni tipovi.



Elementi i vrste

- Elementi niza numerisani su sa $0, 1, 2, \dots, N-1$.
- Ovi brojevi su indeksi elemenata niza.
- Broj elemenata u nizu predstavlja njegovu dužinu.
- Nizovi mogu biti različitih dimenzija.
- Najčešće se koriste jednodimenzionalni nizovi (vektori), dvodimenzionalni nizovi (matrice) i ugnježdjeni (nizovi nizova).

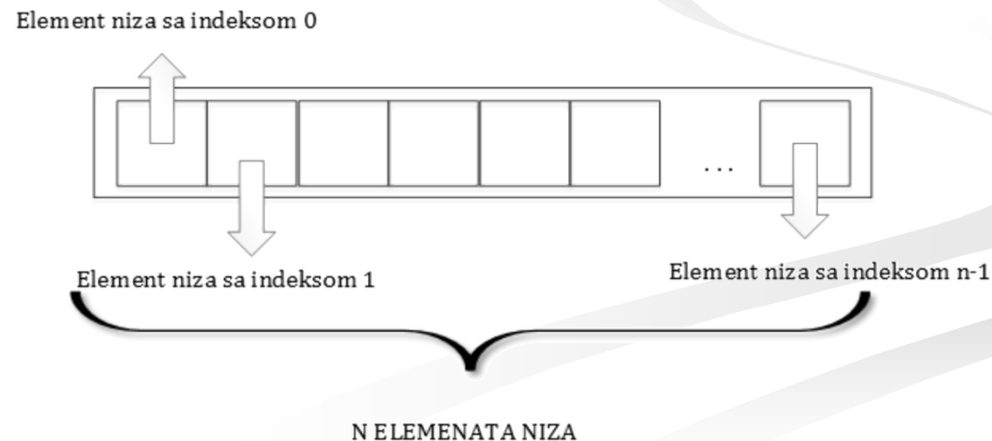


Indeksi

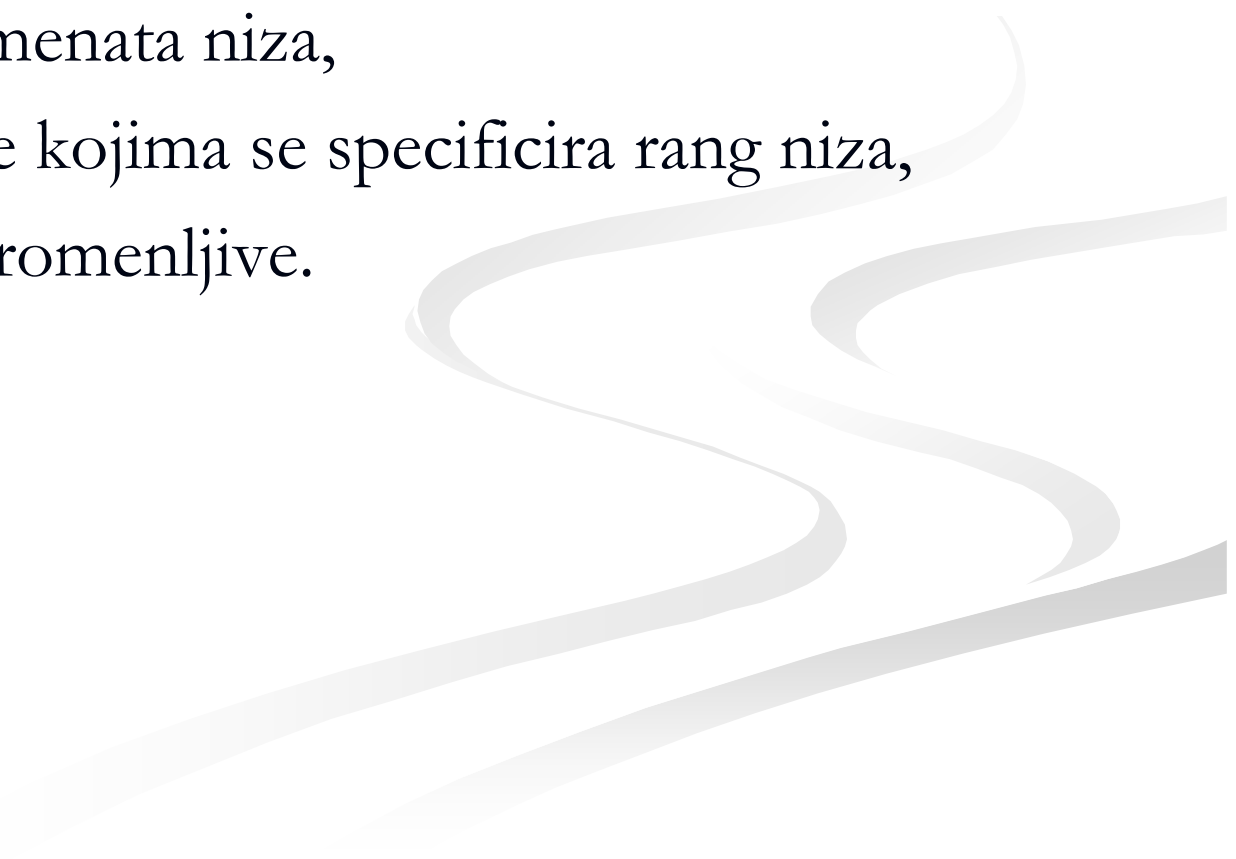
- Elementima niza se pristupa preko zajedničkog imena (naziv nizovne promenljive) i fiksnog broja indeksa.
- Element jednodimenzionalnog niza ima jedan indeks, a n-dimenzionalni niz ima n indeksa.
- U slučaju jednodimenzionalnih nizova veličina niza se poistovećuje sa brojem njegovih elemenata, a indeks elementa se svodi na redni broj elementa u nizu.
 - Prvi indeks je nula.

Elementi niza

- istog su tipa (tip elemenata niza);
- smešteni su u susednim memorijskim lokacijama;
- brzina pristupa bilo kom elementu niza je ista;
- pristup im je omogućen preko indeksa;



Deklarisanje nizova

- Da bi se deklarirala nizovna promenljiva potrebno je:
 - navesti tip elemenata niza,
 - uglaste zagrade kojima se specificira rang niza,
 - ime nizovne promenljive.
- 
- A decorative graphic consisting of several overlapping, wavy, light gray lines that flow from the bottom left towards the top right, positioned in the lower right quadrant of the slide.

Neka svojstva i metode System.Array klase

- Nizovni tipovi se implicitno izvode iz System.Array klase
- *Length* – svojstvo koje vraća dužinu niza.
- *Rank* – svojstvo koje vraća broj – dimenziju niza.
- ...

```
int[] niz = new int [5];  
Console.WriteLine(niz.Rank);
```

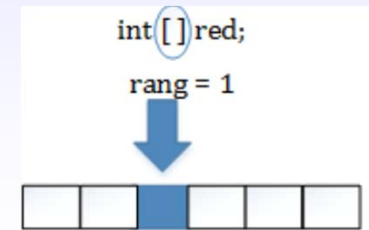
1

```
int[] niz = new int[5];  
Console.WriteLine(niz.Length);
```

Deklarisanje jednodimenzionalnih nizova

- Primer deklarisanja niza:

```
int[] red;
```



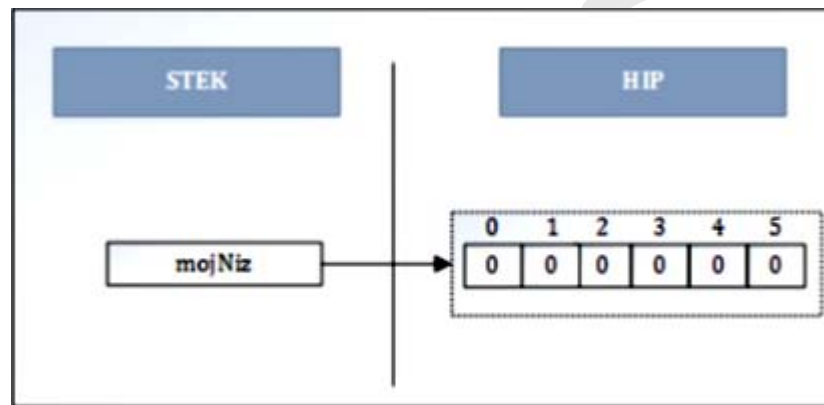
- U primeru promenljiva **red** je jednodimenzionalni niz (**int []**) koji int tipa.
- Elementi niza kreiraju se pozivom **metode new** (alokacija memorije) za svaki elemenat niza:

Primeri:

```
int[] mojNiz = new int[6];  
string [] imeStudenta = new string[5];
```

Kreiranje niza

- `int[] mojNiz = new int[6];`
- Alocira se niz dužine 6 čiji su elementi tip **int**.
- U dinamičkoj memoriji (hip) alocira se prostor za 6 celih brojeva koji su inicijalno postavljeni na 0.



Počtetne vrednosti

- Nakon deklaracije, elementi niza kao standardne vrednosti automatski dobijaju:
 - **0**, ako je niz numeričkog tipa
- Npr. ako deklarišemo niz: `ocena = new int[3];`
 - U memoriji će se rezervisati prostor za 3 ocene, `ocena[0]`, `ocena[1]`, `ocena[2]`, a svaka će imati početnu vrednost 0, tj. `ocena[0]=0`, `ocena[1]=0`, `ocena[2]=0`.

Početne vrednosti

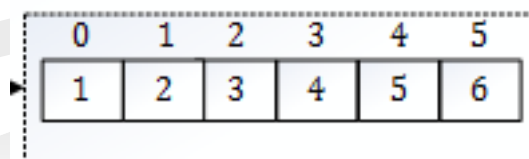
- Nakon deklaracije, elementi niza kao standardne vrednosti automatski dobijaju:
 - **null** ako su referentnog tipa.
 - Ovo uključuje tipove kao što su:
 - string tipovi
 - objekti kreirani pomoću klasa itd.
- null znači da promenljiva ne pokazuje ni na koji objekat, odnosno da nije inicijalizovana.

Dodela vrednosti – jednodimenzionalni nizovi numeričkog tipa

1.

```
int[] mojNiz = new int[6] { 1, 2, 3, 4, 5, 6 };  
//int[] mojNiz = new int[] { 1, 2, 3, 4, 5, 6 };  
//int[] mojNiz={ 1, 2, 3, 4, 5, 6 };  
Console.WriteLine(mojNiz[4]);
```
2. **Direktna dodela** vrednosti članova niza.

```
int[] mojNiz = new int[6];  
mojNiz[0] = 1;  
mojNiz[1] = 2;  
mojNiz[2] = 3;  
mojNiz[3] = 4;  
mojNiz[4] = 5;  
Console.WriteLine(mojNiz[5]);
```



Jednodimenzionalni nizovi - int

3. Ove vrednosti mogu se pridružiti primenom **for ciklusa**

```
int[] mojNiz = new int[6];  
for (int i = 0; i < 6; i++)  
{  
    mojNiz[i] = i+1;  
    Console.WriteLine(mojNiz[i]);  
}
```

4. Članovi niza mogu se uneti i **sa tastature:**

```
int[] mojNiz = new int[6];  
Console.WriteLine("Unesi vrednosti članova niza:");  
for (int i = 0; i < 6; i++)  
{  
    mojNiz[i] = int.Parse(Console.ReadLine());  
}
```

Jednodimenzionalni nizovi - int

5. Slučajne vrednosti se mogu pridružiti primenom **Random** klase i **Next** metode.

```
int[] mojNiz = new int[6];
Random r = new Random();
    for (int i = 0; i < mojNiz.Length; i++)
    {
        mojNiz[i] = r.Next(1, 6);
    }
// Ispis elemenata niza
foreach (int broj in mojNiz)
{
    Console.WriteLine(broj);
}
```


Primer 1.

- Deklarisati niz tipa int, čiji će članovi biti kvadrati prvih 10 brojeva i ispisati ih.

```
int[] niz = new int[10];
for (int i = 0; i < niz.Length; i++)
{
    niz[i] = (i + 1) * (i + 1);
    Console.WriteLine($"{i+1}x{i+1}="+niz[i]);
}
```

* niz.Length je dužina niza – ovde je to 10

Petlja foreach

```
string[] radniDani = { "ponedeljak", "utorak", "sreda",  
    "četrvtak", "petak" };  
    Console.WriteLine("Radni dani u nedelji su:\n");  
foreach (string dan in radniDani)  
{  
    Console.WriteLine(dan);  
}  
  
//for (int i=0;i<radniDani.Length;i++)  
    {  
        Console.WriteLine(radniDani[i]);  
    }
```

Primer 2.

Generisati članove niza i naći maksimalnu vrednost niza.

```
Console.WriteLine("Unesite broj članova niza:");
int n = int.Parse(Console.ReadLine());
int[] mojNiz = new int[n]; // Kreiranje niza sa 'n' elemenata
Random random = new Random();
for (int i = 0; i < mojNiz.Length; i++)
    mojNiz[i] = random.Next(1, 10);
    //Array.Sort(mojNiz);
Console.WriteLine("Elementi niza su:\n"); // Štampanje niza
foreach (int element in mojNiz)
{
    Console.WriteLine(element);
}

int max = mojNiz[0]; //Nalaženje najvećeg elementa
for (int j = 1; j < mojNiz.Length; j++)
    if (mojNiz[j] > max)
        max = mojNiz[j];
Console.WriteLine("\nNajveći član je {0}", max);
```

Jednodimenzionalni nizovi - string

■ 1. način

```
string[] imeStudenta;
```

```
imeStudenta = new string[5];
```

```
imeStudenta[0] = "Marko";
```

```
imeStudenta[1] = "Janko";
```

```
imeStudenta[2] = "Vuk";
```

```
imeStudenta[3] = "Ana";
```

```
imeStudenta[4] = "Jana";
```

Deklaracija niza od 5
elemenata

Dodeljivanje
vrednosti
elementima niza
pojedinačno

```
Console.WriteLine("Treci element niza je {0}", imeStudenta[2]);
```

■ 2. način: sve u jednoj liniji koda

```
string[] imeStudenta = { "Marko", "Janko", "Vuk", "Ana", "Jana" };
```

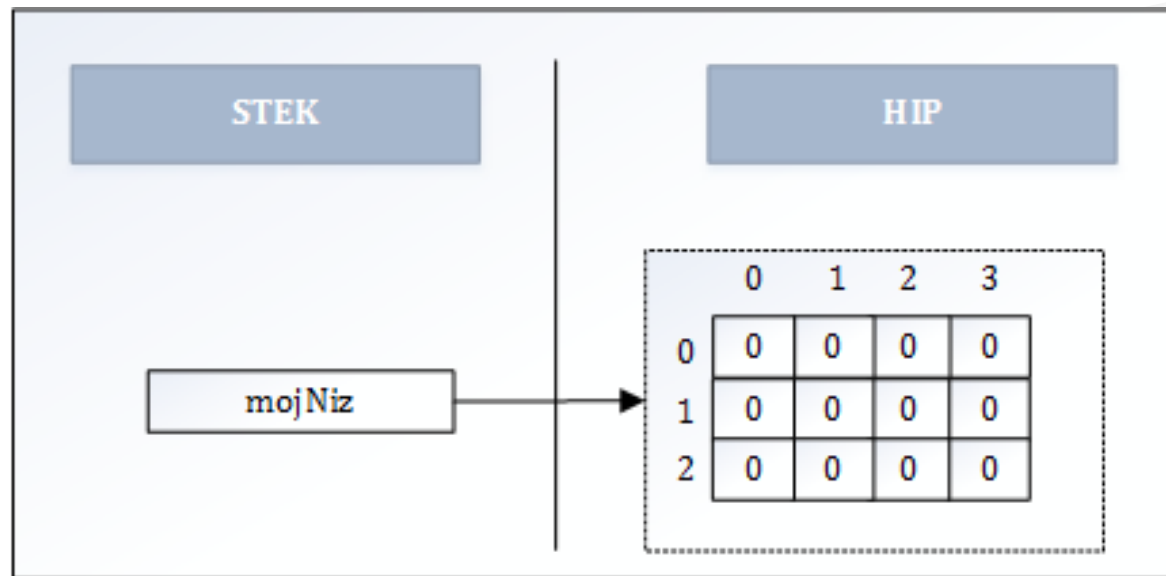
```
Console.WriteLine("Treci element niza je {0}", imeStudenta[2]);
```

Jednodimenzionalni nizovi - char

```
char[] samoglasnik = new char[] { 'a', 'e', 'i', 'o', 'u' };  
    for (int i = 0; i < 5; i++)  
        Console.WriteLine(samoglasnik[i]);  
  
//foreach (char znak in samoglasnik)  
    //Console.WriteLine(znak);
```

Deklarisanje dvodimenzionalnih nizova

- `int[,] mojNiz=new int [3,4];`
- Elementi niza kreiraju se pozivom metode **new** (alokacija memorije) za svaki element niza.



Kreiranje i pridruživanje početnih vrednosti (1)

```
int[,] matrica = new int [3,4]
{
    {1, 2, 3, 4}, // red 0.
    {5, 6, 7, 8}, // red 1.
    {9, 10, 11, 12}, // red 2.
};
```

Dimenzija se može izostaviti ako se iza deklaracije niza članovima dodele vrednosti.

`new int[,]`

```
Console.WriteLine("Prvi član matrice u prvom  
redu: {0}",matrica[0,0]);
```

Statičko dodeljivanje elemenata i štampanje svih članova (1)

```
int[,] matrica =  
{  
    {1, 2, 3, 4},  
    {5, 6, 7, 8},  
    {9, 10, 11, 12},  
};  
  
    for (int i = 0; i < 3; i++)  
    {  
        for (int j = 0; j < 4; j++)  
        { Console.Write(matrica[i, j] + " "); }  
        Console.WriteLine();  
    }
```


Kreiranje i pridruživanje početnih vrednosti (2)

```
int[,] matrica = new int [3,4]
matrica[0,0] = 1; matrica[0,1] = 3;
matrica[0,2] = 6; matrica[0,3] = 2;
matrica[1,0] = 8; matrica[1,1] = 5; ...
```

```
Console.WriteLine("Prvi član matrice u prvom  
redu: {0}",matrica[0,0]);
```

	0	1	2	3
0	1	3	6	2
1	8	5	9	1
2	4	7	3	0

Korisnički unos elemenata i štampanje svih članova (3)

```
int[,] a = new int[3, 4];
for (int i = 0; i < 3; i++)//formiranje matrice
{
    Console.WriteLine("unos "+i+"."+ "reda\n");
    for (int j = 0; j < 4; j++)
    {
        a[i,j] = int.Parse(Console.ReadLine());
    }
    Console.WriteLine();
}
for (int i = 0; i < 3; i++)//štampanje elemenata matrice
{
    for (int j = 0; j < 4; j++)
        Console.Write(a[i, j]+" ");
    Console.WriteLine();
}
```

Varijanta prethodnog primera

```
int m, n;
Console.WriteLine("Unesite broj redova:");
m = int.Parse(Console.ReadLine());
Console.WriteLine("Unesite broj kolona:");
n = int.Parse(Console.ReadLine());
int[,] a = new int[m, n];
for (int i = 0; i < m; i++)//formiranje matrice
{
    Console.WriteLine("unos " + i + "." + "reda\n");
    for (int j = 0; j < n; j++)
    {
        a[i, j] = int.Parse(Console.ReadLine());
    }
    Console.WriteLine();
}
for (int i = 0; i < a.GetLength(0); i++)//štampanje elemenata
{
    for (int j = 0; j < a.GetLength(1); j++)
        Console.Write(a[i, j] + " ");
    Console.WriteLine();
}
```

→Vraća broj redova

→Vraća broj kolona

Unos elemenata korišćenjem klase Random i štampanje svih članova (4)

```
Console.WriteLine("Upišite broj redova");
int m = int.Parse(Console.ReadLine());
Console.WriteLine("Upišite broj kolona");
int n = int.Parse(Console.ReadLine());
int[,] matrica = new int[m, n];
Random r = new Random();
Console.WriteLine();
for (int i = 0; i < m; i++)
{
    for (int j = 0; j < n; j++)
    {
        matrica[i, j] = r.Next(1, 50);
        Console.Write(matrica[i, j] + " ");
    }
    Console.WriteLine();
}
```

Primer 1.

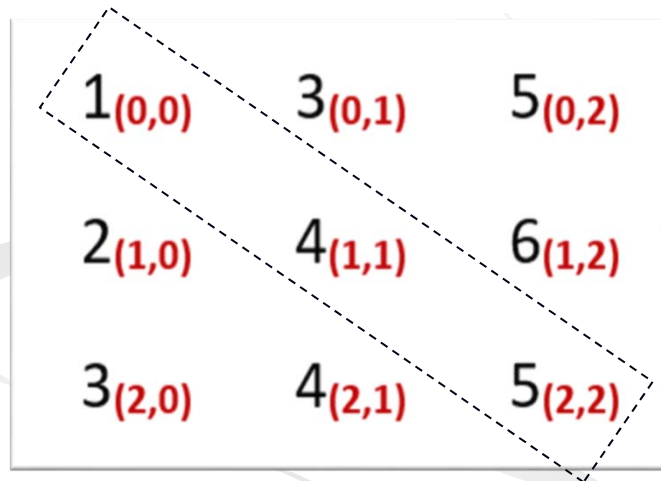
```
Console.WriteLine("Upišite broj el.1. niza");
int m = int.Parse(Console.ReadLine());
Console.WriteLine("Upišite broj el.2. niza");
int n = int.Parse(Console.ReadLine());
int[] a = new int[m];
int []b=new int[n];
int[,] matrica = new int[m, n];
Random r = new Random();
Console.WriteLine();
for (int i = 0; i < m; i++)//formiranje 1.niza
{
    a[i] = r.Next(1, 10);
    Console.WriteLine(a[i]);
}
Console.WriteLine();
for (int j = 0; j < n; j++)//formiranje 2.niza
{
    b[j] = r.Next(1, 10);
    Console.Write(b[j] + " ");
}
Console.WriteLine("\n");
for(int i=0;i<m;i++) //formiranje matrice
{
    for (int j = 0; j < n; j++)
    {
        matrica[i, j] = a[i] * b[j];
        Console.Write(matrica[i, j]+" ");
    }Console.WriteLine();
}
```

Za zadate nizove a i b, kreirati matricu čiji se svaki elemenat dobija kao proizvod odgovarajućih elemenata nizova a i b.

Primer 2.

```
int[,] a = new int[5, 5];
    Random r = new Random();
    for (int i = 0; i < 5;
i++)//formiranje kvadratne matrice
    {
        for (int j = 0; j < 5; j++)
        {
            a[i, j] = r.Next(1, 10);
            Console.Write(a[i, j] + " ");
        }
        Console.WriteLine();
    }
int max = a[0, 0], m = 0;
for (int i = 1; i < 5; i++)
{
    if (a[i, i] > max)
    {
        max = a[i, i];
        m = i;
    }
}
Console.WriteLine("Maksimalni element
na dijagonali je " + max);
Console.WriteLine($"Indeks tog
elementa je ({m},{m})" );
```

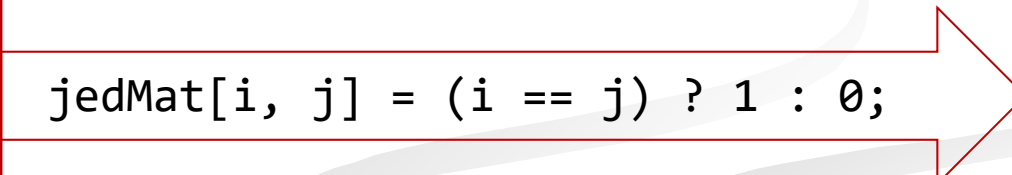
Pronaći indeks najvećeg elementa na glavnoj dijagonali **kvadratne** matrice.



J-na matrica nxn

if i inline if

```
Console.WriteLine("Unesi n:");
int n = int.Parse(Console.ReadLine());
int[,] jedMat = new int[n, n];
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        if (i == j)
            jedMat[i, j] = 1;
        else
            jedMat[i, j] = 0;
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        Console.Write(jedMat[i, j] + " ");
    }
    Console.WriteLine("\n");
}
```



`jedMat[i, j] = (i == j) ? 1 : 0;`