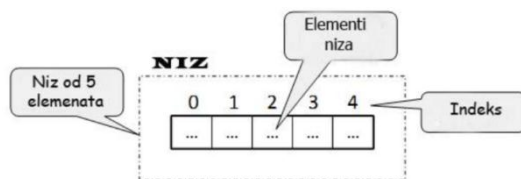


ЈЕДНОДИМЕНЗИОНАЛНИ НИЗОВИ

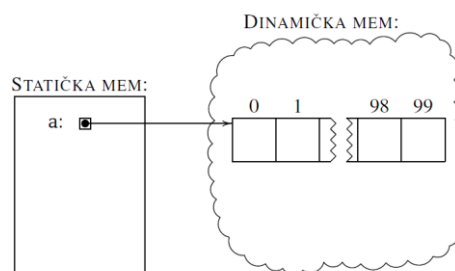
Низови (енглески **arrays**) представља фиксни број елемената одређеног типа.



Декларисање низа:

Низ се означава угластим заградама иза типа елемената. Пример дефинисања низа од 100 елемената типа **int**.

```
int[] a = new int[100];
```



Приступање елементима низа:

```
a[1] = 5;
```

другом елементу низа, тј. првом индексираним
додељујемо вредност 5

Иницијализација низа:

```
int [] niz1 = new int[]{3,4,7,9,1};
int [] niz2 = {3,4,9,4,1};
```

НАПОМЕНА: niz1 има 5 елемената
низ, могли смо да напишемо и овако:
`int [] niz1 = new int[5]{3,4,7,9,1};`

Пример: Анализирати програмски код и објаснити шта ће бити приказано на екрану.

```
using System;
class Program
{
    static void Main(string[] args)
    {
        int[] a = new int[100];
        Console.WriteLine(a[2]);
        Console.ReadKey();

        a = null;
        Console.WriteLine(a[2]);
        Console.ReadKey();
    }
}
```

НАПОМЕНА: Како се назива врста грешке која се јавља у програму?

Пример: Написати програм за израчунавање аритметичке средине бројева : 5, 1, 4, 5 и 2. Бројеви су дефинисани у форми низа.

```
using System;
class Program
{
    static void Main(string[] args)
    {
        int [] niz = new int[]{5,1,4,5,2};
        int suma =0;
        double sredina;
        for (int i = 0; i < niz.Length; i++)
            suma += niz[i];

        sredina = suma / 5;
        Console.WriteLine(sredina);
        Console.ReadKey();
    }
}
```

НАПОМЕНА: Како се назива врста грешке која се јавља у програму?

Пример: Анализирати програмски код и објаснити шта ће бити приказано на екрану.

```
using System;
class Program
{
    static void Main(string[] args)
    {
        int[] niz = new int[5];
        double sredina;
        for (int i = 0; i < niz.Length; i++)
            niz[i] = i;
        Console.WriteLine(a[i] + " ");
        Console.ReadKey();
    }
}
```

НАПОМЕНА: Како се назива врста грешке која се јавља у програму?

АНАЛИЗИРАТИ И ДАТИ ТАЧНЕ ОДГОВОРЕ ЗА НАВЕДЕНА ПИТАЊА. ЗА СВАКИ ОДГОВОР ДАТИ ОБЈАШЊЕЊЕ!

180. Дат је код програма у програмском језику C# који формира и штампа елементе низа **a**. Анализирати дати код и проценити шта ће се догодити након његовог извршавања.

```
namespace TestPrimer {
class Program{
static void Main(string[] args){
int[] a = new int[5];
for (int i = 0; i < a.Length; i++) a[i] = i;
Console.Write(a[i] + " ");
}
}
}
```

Заокружити број испред тачног одговора:

1. Програм приказује бројеве 0 1 2 3 4 на екрану.
2. Програм има грешку, јер ће у последњој наредби **Console.Write** метода **Main** покушати приступ непостојећем елементу **a[5]**.
3. Програм приказује број 5 на екрану.
4. Програм има грешку, јер променљива **i** у последњој наредби **Console.Write** у методи **Main** неће имати дефинисану вредност.

НАПОМЕНА: Разликовати појмове “непостојећи елемент” и “дефинисану вредност”. Када кажемо да елемент нема дефинисану вредност значи да постоји али да нема дефинисану вредност. У нашем случају покушавамо да приступимо непостојећим елементима, па је **ОДГОВОР ПОД 2.**

233.	Са леве стране наведене су врсте грешака које се могу десити у коду, а са десне описи тих грешака. На линију испред описа уписати редни број под којим је наведен одговарајући тип грешке:		
1.	Синтаксне грешке (syntax error)	Откривају се у фази компилације и најлакше су за кориговање	1,5
2.	Логичке грешке (logical error)	Не морају, али се могу десити током извршења и тада доводе до „пуцања“ програма	
3.	Грешке у време извршења (run-time error)	Програм функционише, нема „пуцања“, али се добијају погрешни резултати	
153.	Дата је наредба кода у програмском језику C# која представља дефиницију низа niz. Која је од доле наведених дефиниција низа коректна:		1
1.	int[] niz = new int[20];		
2.	int[] niz = new int(20);		
3.	int niz = new int[20];		
4.	int niz() = new int[20];		
208.	Дата је наредба кода у програмском језику C# која представља декларацију низа. Које су декларација тачне:		1,5
1.	int niz = new int(30);		
2.	double[] niz = new double[30];		
3.	int[] niz = { 3, 4, 3, 2 };		
4.	char[] niz = new char[];		
5.	char[] niz = new char { 'a', 'b', 'c', 'd' };		
6.	char[] niz = new char[] { 'a', 'b' };		

ВИШЕДИМЕНЗИОНАЛНИ НИЗОВИ

Разликујемо две типа вишедимензионалних низова: правоугаони и назубљени.

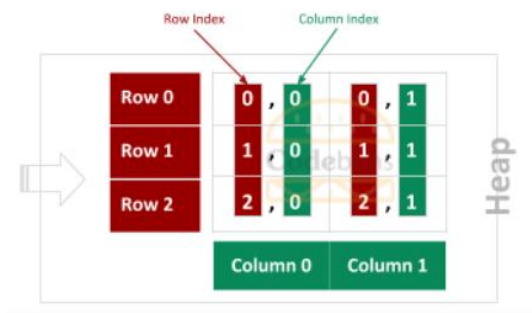
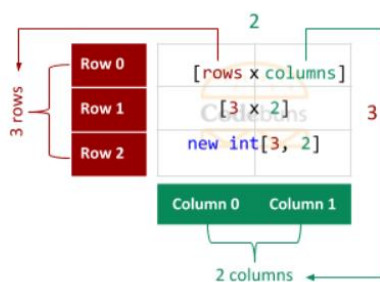
Правоугаони низови представљају n – димензионални блок меморије.

```
int [,] niz1 = new int[3,2];    //dvodimenzionalni niz
int [,,] niz2 = new int[4,2,3]; //trodimenzionalni niz
```

Пример: Дводимензионалног низа, димензије **3 x 2** (3 врсте и 2 колоне).

```
int[,] niz1 = new int[,]{ {1,3},{5,6},{4,7} };

//mogli smo i ovako da zapisemo
int[,] niz2 = new int[3,2] { { 1, 3 }, { 5, 6 }, { 4, 7 } };
```



Назубљени низови или низови низова су низови чији унутрашњи низови могу бити различитих величина, тј. унутрашњи низ може бити произвољне дужине.

Следећи пример показује како се декларише, иницијализује и приступа назубљеним низовима.

```
using System;
class Program
{
    static void Main(string[] args)
    {
        //deklarisanje niza
        int [][] niz = new int[4][]; //najveca spoljna dimenzija 4

        //inicijalizacija niza
        niz[0] = new int[] { 1, 2, 3, 4, 5, 6 };
        niz[1] = new int[] { 5, 6 };
        niz[2] = new int[] { 9, 10, 11 };
        niz[3] = new int[] { 13, 14, 15, 16, 17, 18, 19 };

        //pristup elementu
        Console.WriteLine(niz[2][2]); //sta ce prikazati na ekranu?
        Console.ReadKey();
    }
}
```