

Razgranate linijske algoritamske šeme

Razgranate linijske algoritamske šeme

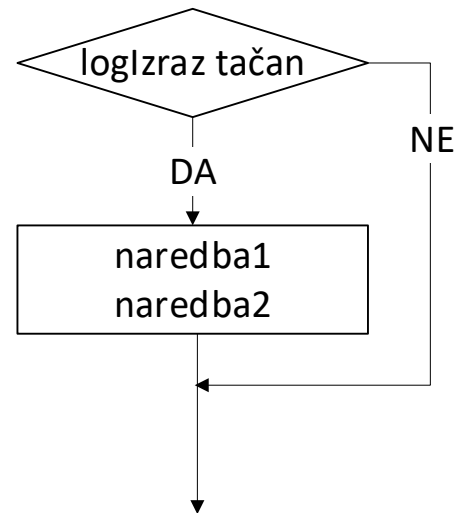
- Svaki algoritamski korak se izvršava **najviše jednom** (znači jednom ili nijednom) i obavezno sadrži bar jedan uslovni algoritamski koraka
- Ako je uslov ispunjen, izlaz iz algoritamskog koraka biće označen sa 'DA', a ako uslov nije ispunjen izlaz će biti označen sa 'NE', ili će biti bez oznake

Naredba if

Naredba if sa blokom naredbi.

```
if (logIzraz)
{
    naredba1;
    naredba2;
}
```

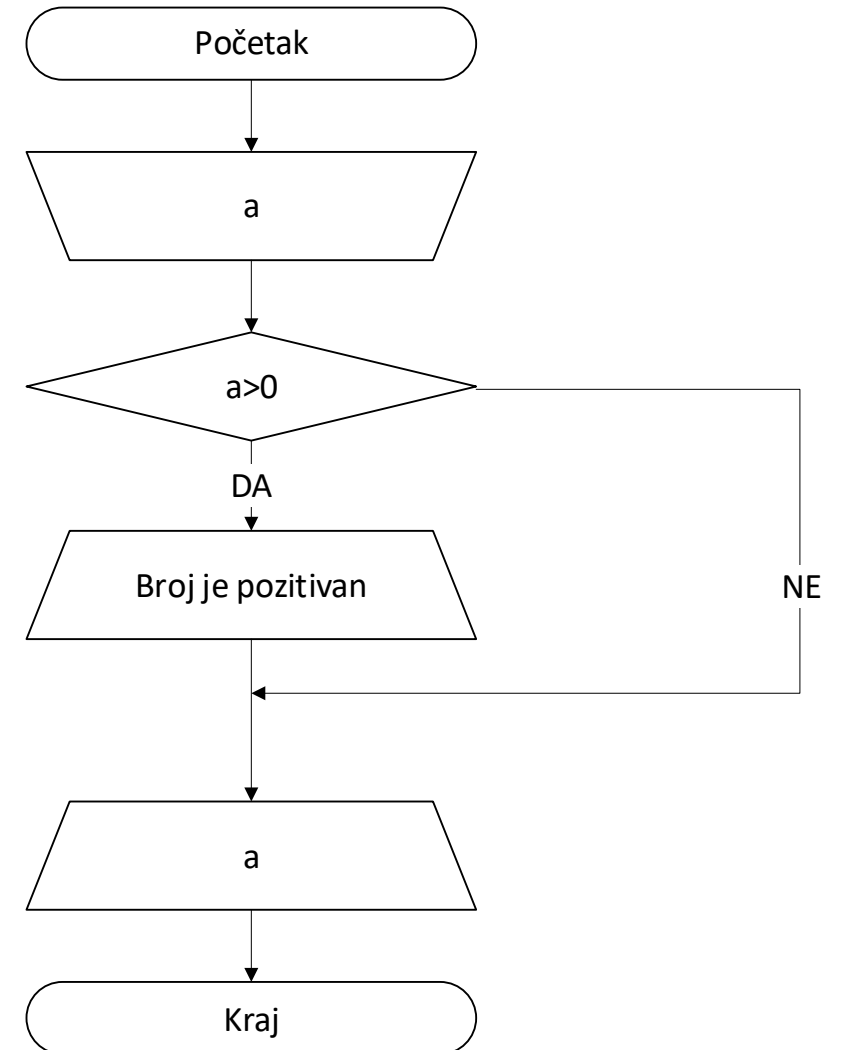
```
if (logizraz)
naredba1;
```



Primer upotrebe naredbe if

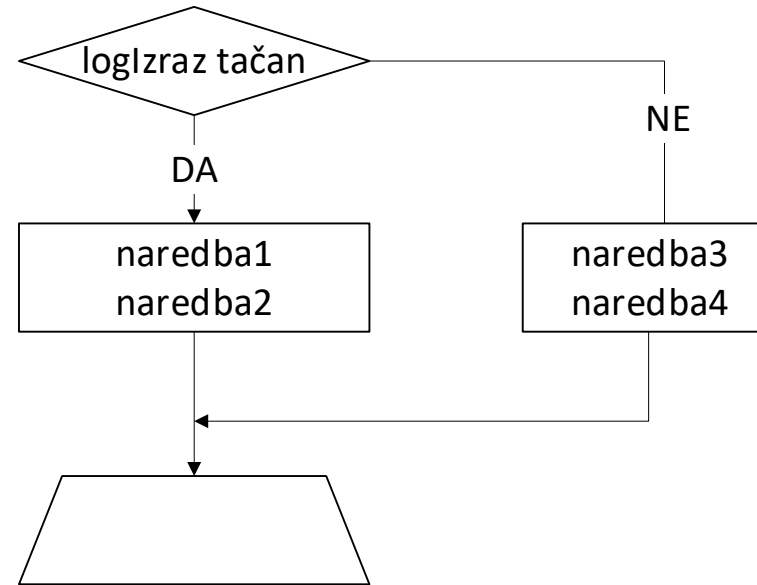
```
static void Main(string[] args)
{
    Random rnd = new Random();
    // -7 do 7
    int a = rnd.Next(-7,8);

    if (a > 0)
    {
        Console.WriteLine("Generisan je pozitivan broj");
    }
    Console.WriteLine($"a={a}");
    Console.ReadLine();
}
```



Naredba if-else

```
if (logizraz)
{
    naredba1;
    naredba2;
}
else
{
    naredba3;
    naredba4;
}
```



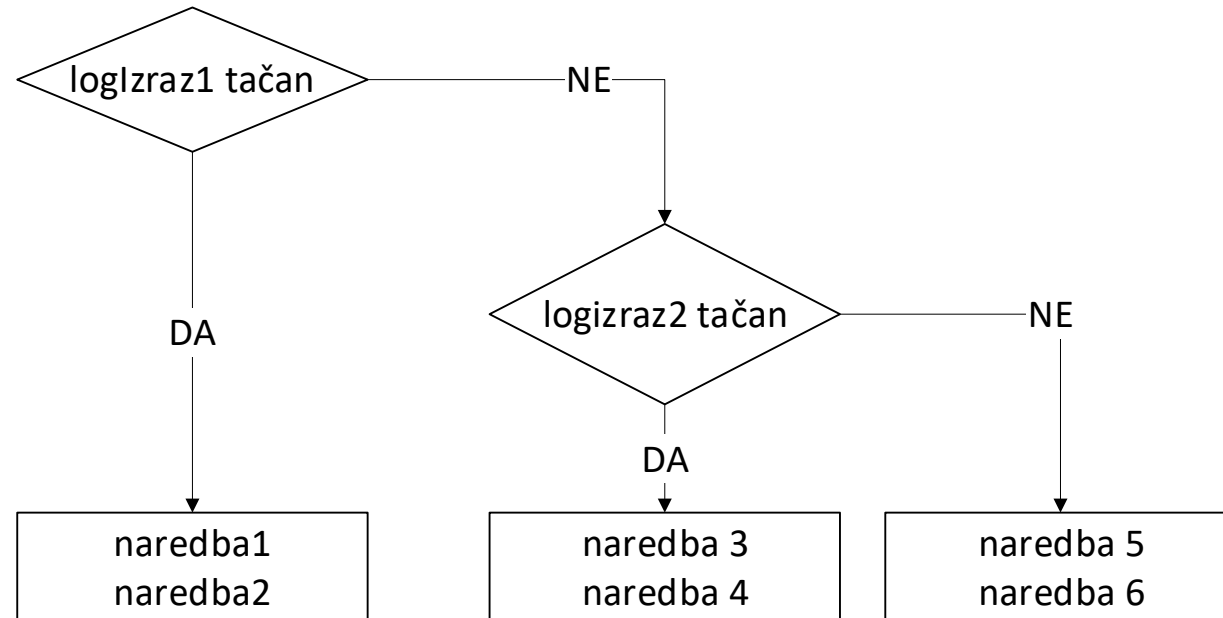
Primer upotrebe if-else naredbe

```
static void Main(string[] args)
{
    Random rnd = new Random();
    // -7 do 7
    int a = rnd.Next(-7, 8);

    if (a > 0)
    {
        Console.WriteLine("Generisan je pozitivan broj");
    }
    else
    {
        Console.WriteLine("Generisan je broj koji nije pozitivan");
    }
    Console.WriteLine($"a={a}");
    Console.ReadLine();
}
```

if-else if

```
if (logizraz1)
{
    naredba1;
    naredba2;
}
else if (logizraz2)
{
    naredba3;
    naredba4;
}
else
{
    naredba5;
    naredba6;
}
```



Primer upotrebe if-else if naredbe

```
static void Main(string[] args)
{
    Random rnd = new Random();
    // -7 do 7
    int a = rnd.Next(-7, 8);

    if (a > 0)
    {
        Console.WriteLine("Generisan je pozitivan broj");
    }
    else if (a == 0)
    {
        Console.WriteLine("Generisan je broj 0");
    }
    else
    {
        Console.WriteLine("Generisan je negativan broj");
    }
    Console.WriteLine($"a={a}");
    Console.ReadLine();
}
```


Primer

Sastaviti algoritam za izračunavanje vrednosti Y po formuli:

$$Y = \begin{cases} X1 + X2, & X1 < X2 \\ X1 - X2, & X1 \geq X2 \end{cases}$$

```
static void Main(string[] args)
{
    // Y = X1 + X2, ako je X1 < X2
    // Y = X1 - X2, ako je X1 >= X2

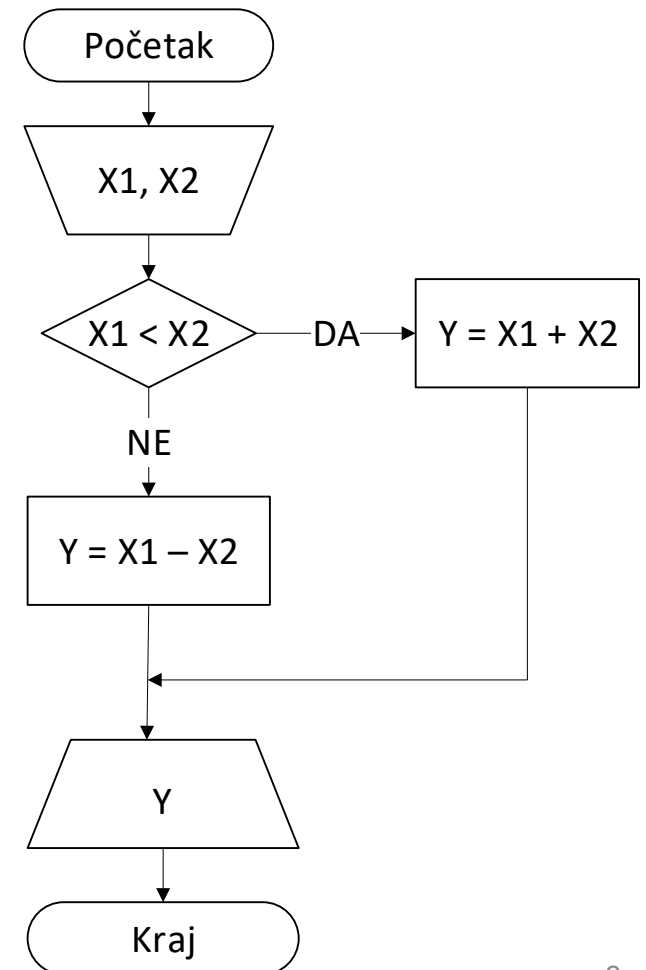
    Console.WriteLine("Ucitaj X1");
    double x1 = double.Parse(Console.ReadLine());

    Console.WriteLine("Ucitaj X2");
    double x2 = double.Parse(Console.ReadLine());

    double y = 0;

    if (x1 < x2)
    {
        y = x1 + x2;
    }
    else
    {
        y = x1 - x2;
    }

    Console.WriteLine($"x1 = {x1}, x2 = {x2}, y = {y}");
    Console.ReadLine();
}
```



Operator ?:

```
static void Main(string[] args)
{
    Random rnd = new Random();
    int a = rnd.Next(-3, 4);

    int b = a >= 0 ? a : -a;

    Console.WriteLine($"a={a}, |a| = {b}");
    Console.WriteLine("Pritisni ENTER za izlazak");
    Console.ReadLine();
}
```

Pitanje 1

Šta je rezultat izvršavanja sledećeg koda:

```
static void Main(string[] args)
{
    int x = 2;
    if (x>1)
    {
        x = 1;
    }
    else
    {
        x = 0;
    }
    Console.WriteLine(x);
}
```

- a. 0
- b. 1
- c. 2

Odgovor: b

Pitanje 2

Šta je rezultat izvršavanja sledećeg koda:

```
static void Main(string[] args)
{
    float x = 3.14f;
    if (x>4)
    {
        x = 4.1f;
    }
    else if(x>3)
    {
        x = 3.5f;
    }
    else
    {
        x = 0;
    }
    Console.WriteLine(x);
}
```

- a. 4.1
- b. 3.5
- c. 0

Odgovor: b

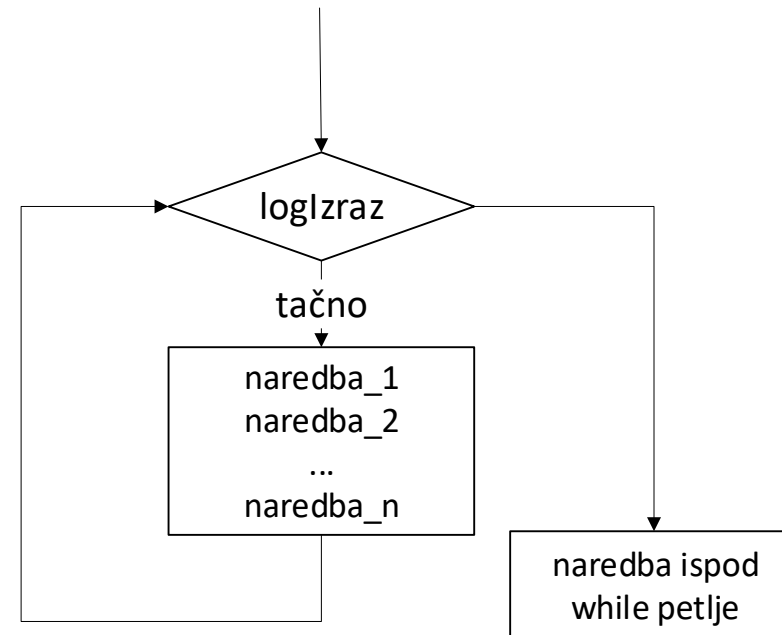
Ciklične algoritamske šeme

Definicija

- Ciklična algoritamska šema je ona šema kod koje se jedan ili više algoritamskih koraka može izvršavati više od jedanput u toku jednog izvršavanja algoritma
- Algoritamski koraci koji se ponavljaju čine ciklus

While petlja

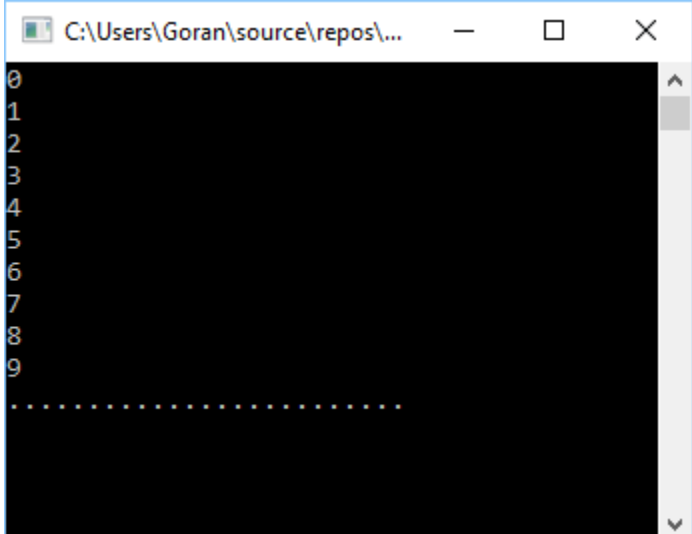
```
while (logIzraz)
{
    naredba_1;
    ....
    naredba_n;
}
```



Brojanje unapred

```
static void Main(string[] args)
{
    int brojac = 0;
    while (brojac < 10)
    {
        Console.WriteLine(brojac);
        brojac++;
    }

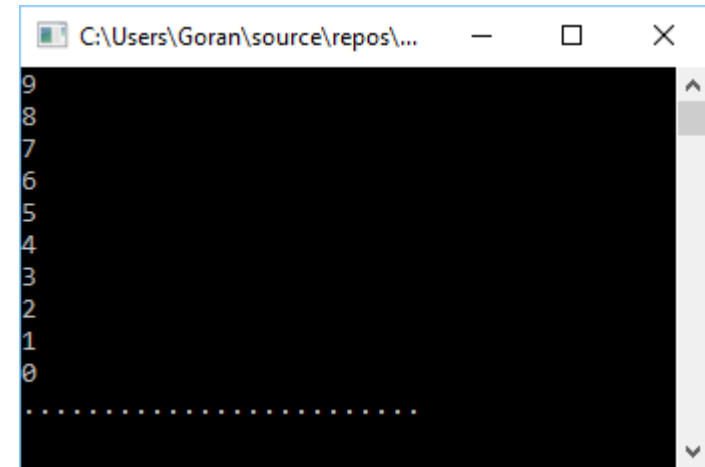
    Console.WriteLine(".....");
    Console.ReadLine();
}
```



```
C:\Users\Goran\source\repos\...
0
1
2
3
4
5
6
7
8
9
.....
.....
```


Brojanje unazad

```
static void Main(string[] args)
{
    int brojac = 9;
    while (brojac >= 0)
    {
        Console.WriteLine(brojac);
        brojac--;
    }
    Console.WriteLine(".....");
    Console.ReadLine();
}
```



The screenshot shows a Windows console window with the title bar "C:\Users\Goran\source\repos\...". The console output displays a reverse count from 9 to 0, with each number on a new line. Below the number 0, there is a line of ten dots. The window has a scrollbar on the right side.

Primer beskonačne while petlje

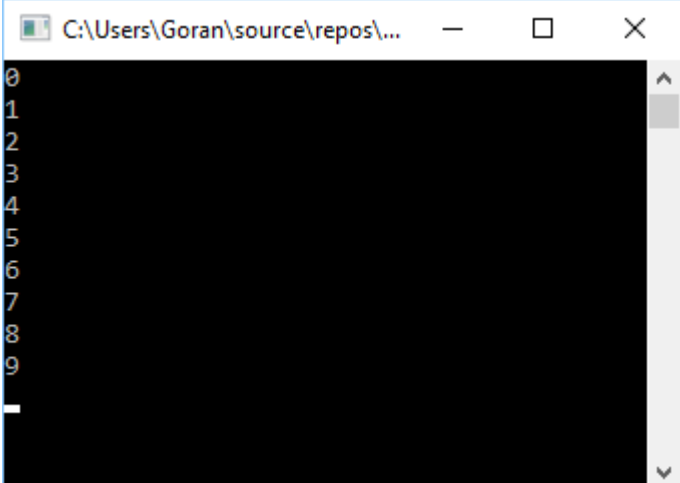
```
static void Main(string[] args)
{
    int brojac = 0;
    while (true)
    {
        Console.WriteLine(brojac);
        brojac++;
    }
}
```

Naredba break

```
static void Main(string[] args)
{
    int brojac = 0;
    while (true)
    {
        Console.WriteLine(brojac);
        brojac++;

        if (brojac > 9)
        {
            break;
        }
    }

    Console.ReadLine();
}
```



```
C:\Users\Goran\source\repos\...
0
1
2
3
4
5
6
7
8
9
_
```

Suma brojeva od 1 do n

```
static void Main(string[] args)
{
    // suma = 1 + 2 + 3 + .... + n

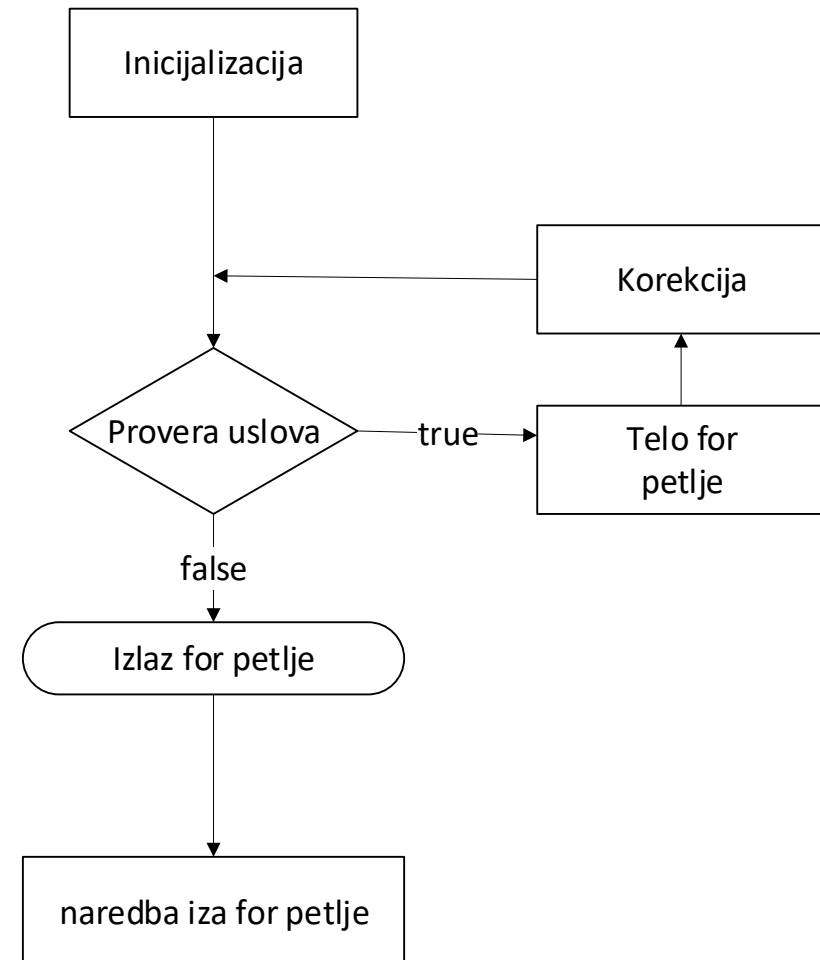
    Console.WriteLine("Unesi pozitivan ceo broj");
    int n = int.Parse(Console.ReadLine());

    int i = 1;
    double suma = 0;
    while (i <= n)
    {
        suma += i;
        i++;
    }

    Console.WriteLine($"Zbir brojeva je: {suma}");
    Console.ReadLine();
}
```

Petlja for

```
for (inicijalizacija; provera_uslova; korekcija)
{
    naredba_1;
    naredba_2;
    .....
    naredba_n;
}
```



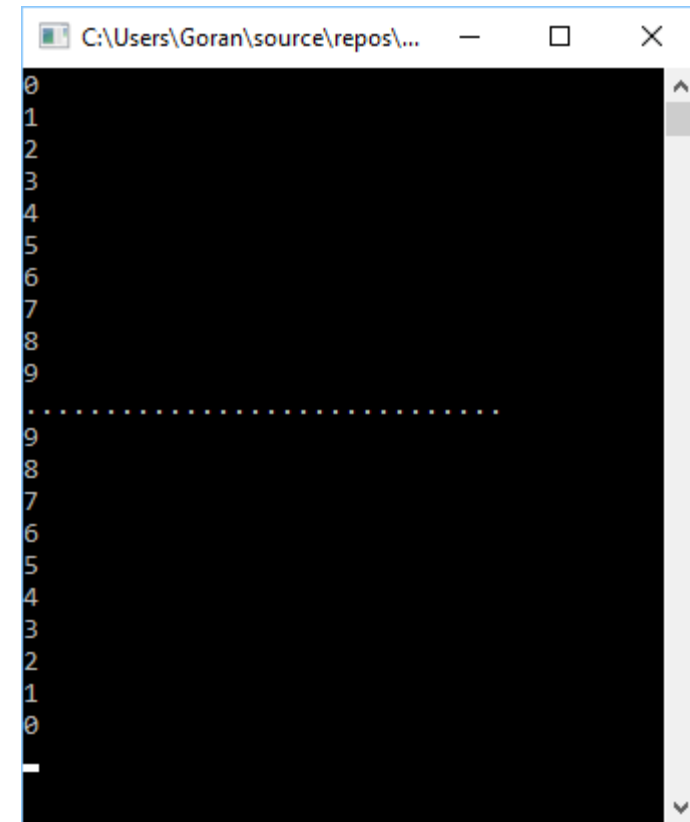
Primeri upotrebe for petlje

```
static void Main(string[] args)
{
    for (int i = 0; i < 10; i++)
    {
        Console.WriteLine(i);
    }

    Console.WriteLine(".....");

    for (int i = 9; i >= 0; i--)
    {
        Console.WriteLine(i);
    }

    Console.ReadLine();
}
```



```
C:\Users\Goran\source\repos\...
0
1
2
3
4
5
6
7
8
9
.....
9
8
7
6
5
4
3
2
1
0
_
```

Specijalni slučajevi ciklusa for

```
static void Main(string[] args)
{
    int i = 0;
    for (;i < 10; )
    {
        Console.WriteLine(i);
        i++;
    }
    Console.ReadLine();
}
```

Beskonačna for petlja

```
static void Main(string[] args)
{
    int i = 0;
    for (;;)
    {
        Console.WriteLine(i);
        i++;
        if (i > 9)
        {
            break;
        }
    }
    Console.ReadLine();
}
```


Stepen celog broja

Napisati program kojim se sa tastature učitavaju ceo broj a kao i stepen n. Program treba da izračuna i prikaže n-ti stepen celog broja a primenom petlje for. Ne koristiti klasu Math.

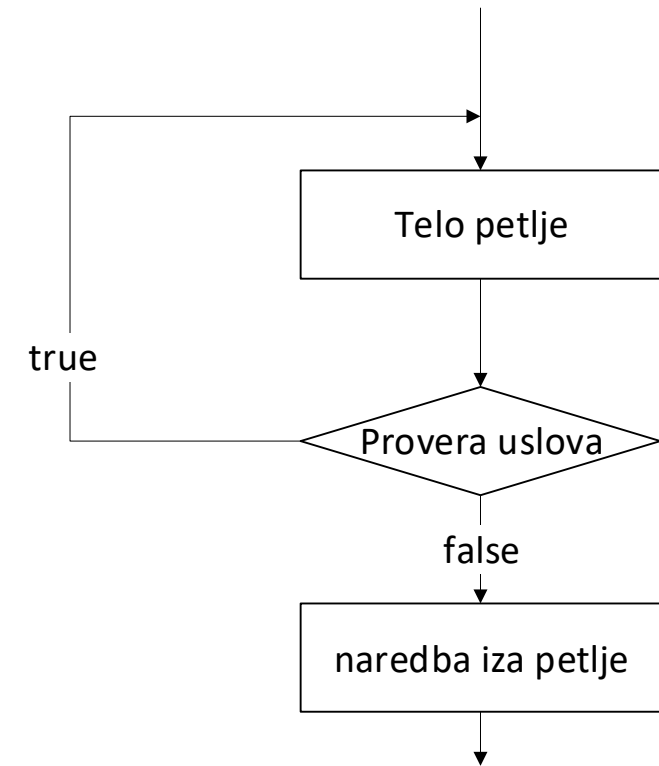
```
static void Main(string[] args)
{
    Console.WriteLine("Unesite ceo broj a:");
    int a = int.Parse(Console.ReadLine());
    Console.WriteLine("Unesite stepen n: ");
    int n = int.Parse(Console.ReadLine());

    int an = 1;

    for (int i = 0; i < n; i++)
    {
        an *= a;
    }
    Console.WriteLine($"Broj {a} na stepen {n} je: {an}");
    Console.ReadLine();
}
```

Opšti oblik petlje do-while

```
do
{
    naredba_1;
    naredba_2;
    ...
    naredba_n;
}
while (logickiIzraz);
```



Primer upotrebe do-while petlje

```
static void Main(string[] args)
{
    string odgovor = "n";
    do
    {
        Console.WriteLine("Unesite prvi broj");
        int a = int.Parse(Console.ReadLine());

        Console.WriteLine("Unesite drugi broj");
        int b = int.Parse(Console.ReadLine());

        Console.WriteLine($"{a} + {b} = {a + b}");
        Console.WriteLine("Da li zelite novo racunanje d - Da, bilo koji drugi taster - Ne");
        odgovor = Console.ReadLine();

    } while (odgovor == "d" || odgovor == "D");

    Console.WriteLine("ENTER za izlazak");
    Console.ReadLine();
}
```

Suma brojeva od 1 do n

```
static void Main(string[] args)
{
    // suma = 1 + 2 + 3 + .... + n
    int n;
    do
    {
        Console.WriteLine("Unesi pozitivan ceo broj: ");
        n = int.Parse(Console.ReadLine());
    }
    while (n <= 0);

    int suma = 0;

    for (int i = 1; i <= n; ++i)
    {
        suma += i;    // suma = suma+i;
    }

    Console.WriteLine($"Zbir brojeva je: {suma}");
    Console.ReadLine();
}
```

Faktorijel broja

```
static void Main(string[] args)
{
    // faktorijel = 1 * 2 * 3 * .... * n
    int n;
    Console.WriteLine("Unesi pozitivan ceo broj: ");
    n = int.Parse(Console.ReadLine());

    if (n < 0)
    {
        Console.WriteLine("Ne postoji faktorijel negativnog broja");
    }
    else
    {
        long faktorijel = 1;
        for (int i = 1; i <= n; ++i)
        {
            faktorijel *= i;    // faktorijel = faktorijel*i;
        }
        Console.WriteLine($"faktorijel od {n} = {faktorijel}");
    }
    Console.ReadLine();
}
```

Pitanje 1

Petlja while je petlja:

- a. Sa izlaskom na vrhu
- b. Sa izlaskom na dnu
- c. Beskonačna petlja

Odgovor: a

Pitanje 2

Petlja do-while je petlja sa :

- a. Sa izlaskom na vrhu
- b. Sa izlaskom na dnu
- c. Beskonačna petlja

Odgovor: b

Pitanje 3

Koliko se puta ispisiuje tekst Zdravo primenom sledeće for petlje:

```
for (int i = 0; i < 5; i++)  
{  
    Console.WriteLine("Zdravo");  
}
```

a.4

b.5

c.6

Odgovor: b