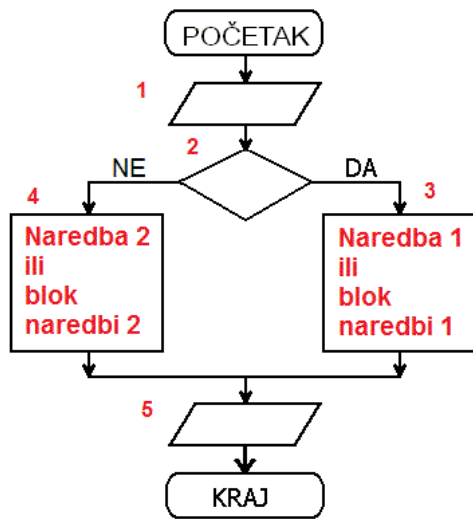


Kontrolne naredbe

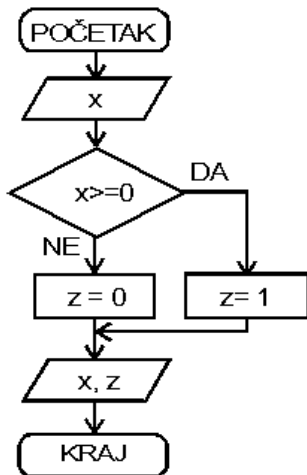
Primjeri:

Opšti oblik razgranate strukture (if sa else)



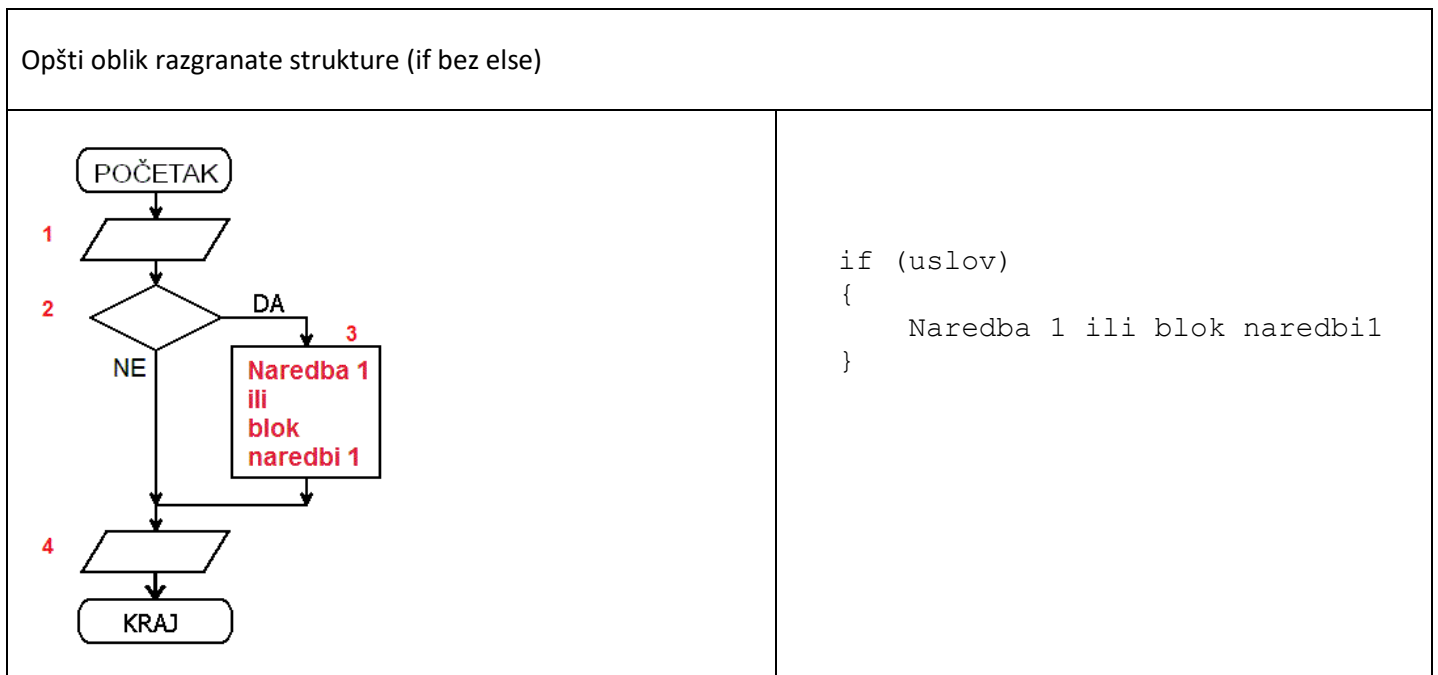
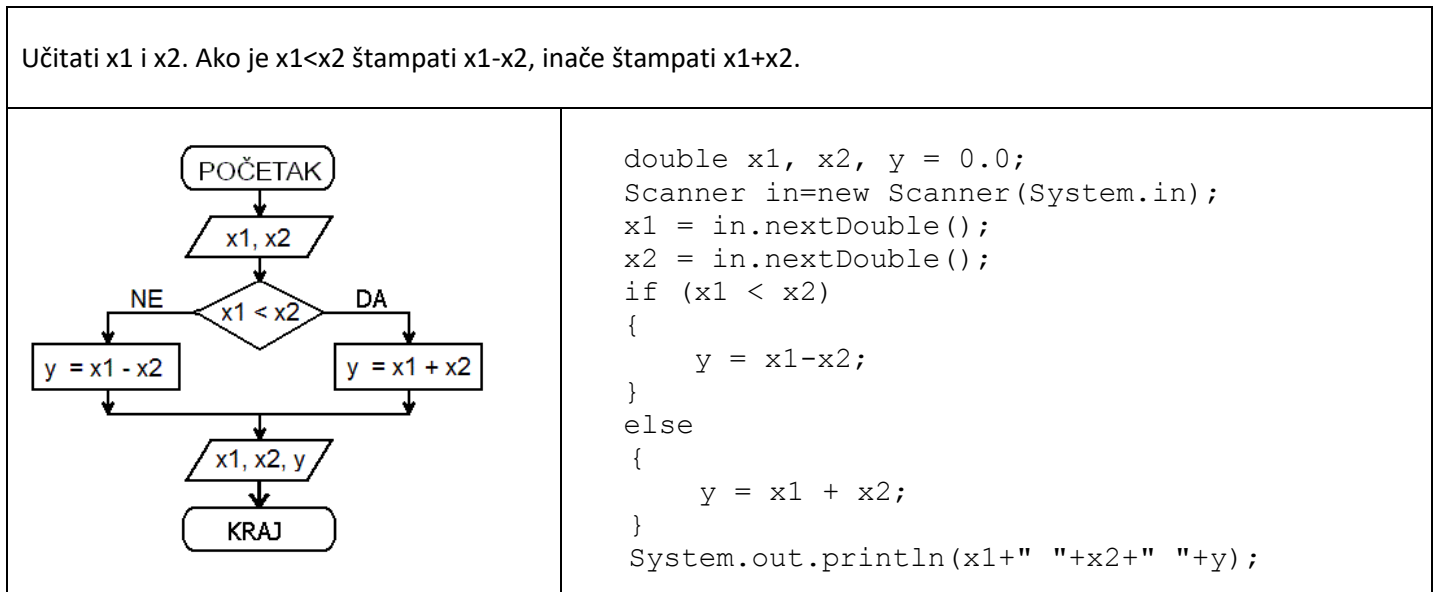
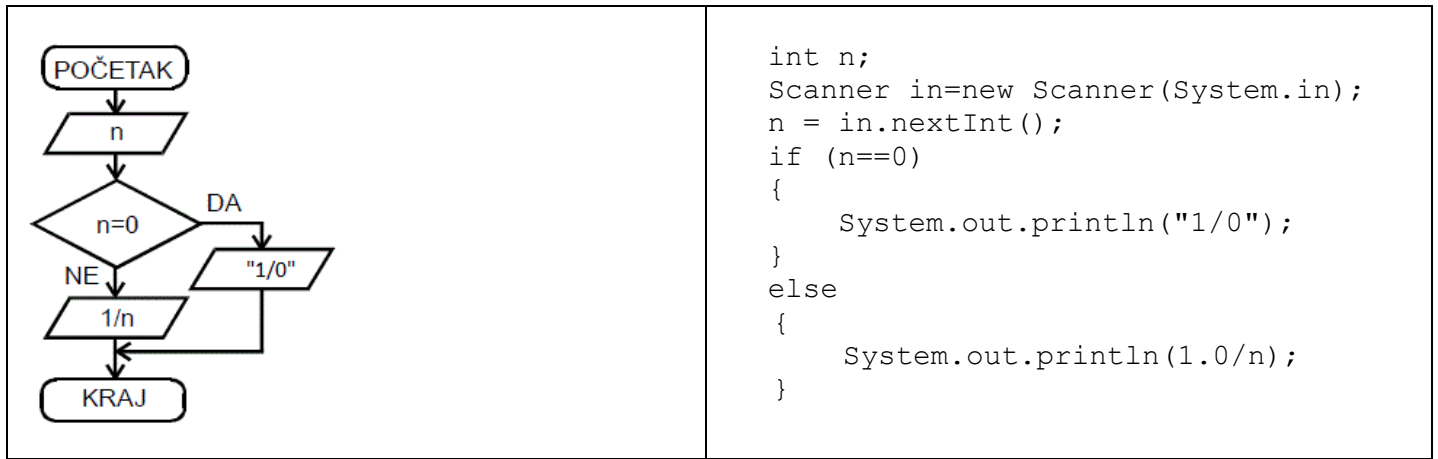
```
if (uslov)
{
    Naredba 1 ili blok naredbi1
}
else
{
    Naredba 2 ili blok naredbi2
}
```

Učitati broj x i štampati vrijednost $z = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$

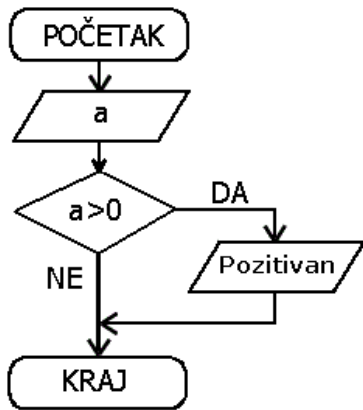


```
double x, z;
Scanner in=new Scanner(System.in);
x = in.nextInt();
if (x>=0)
{
    z = 1;
}
else
{
    z = 0;
}
System.out.println(x+" "+z);
```

Učitati cio broj n i štampati njegovu recipročnu vrijednost. Ako je učitani broj 0, štampati „1/0“.



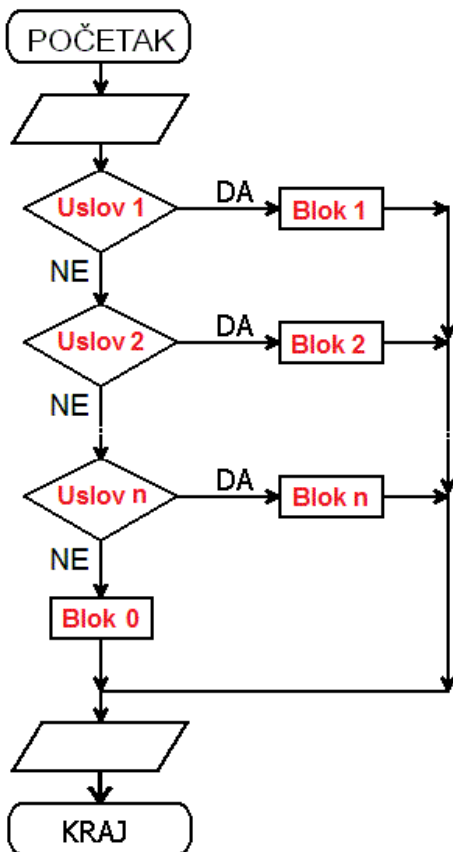
Učitati broj a. Ako je a pozitivan, štampati poruku "Pozitivan".



```

double a;
Scanner in=new Scanner(System.in);
a = in.nextDouble();
if (a > 0)
{
    System.out.println("Pozitivan");
}
  
```

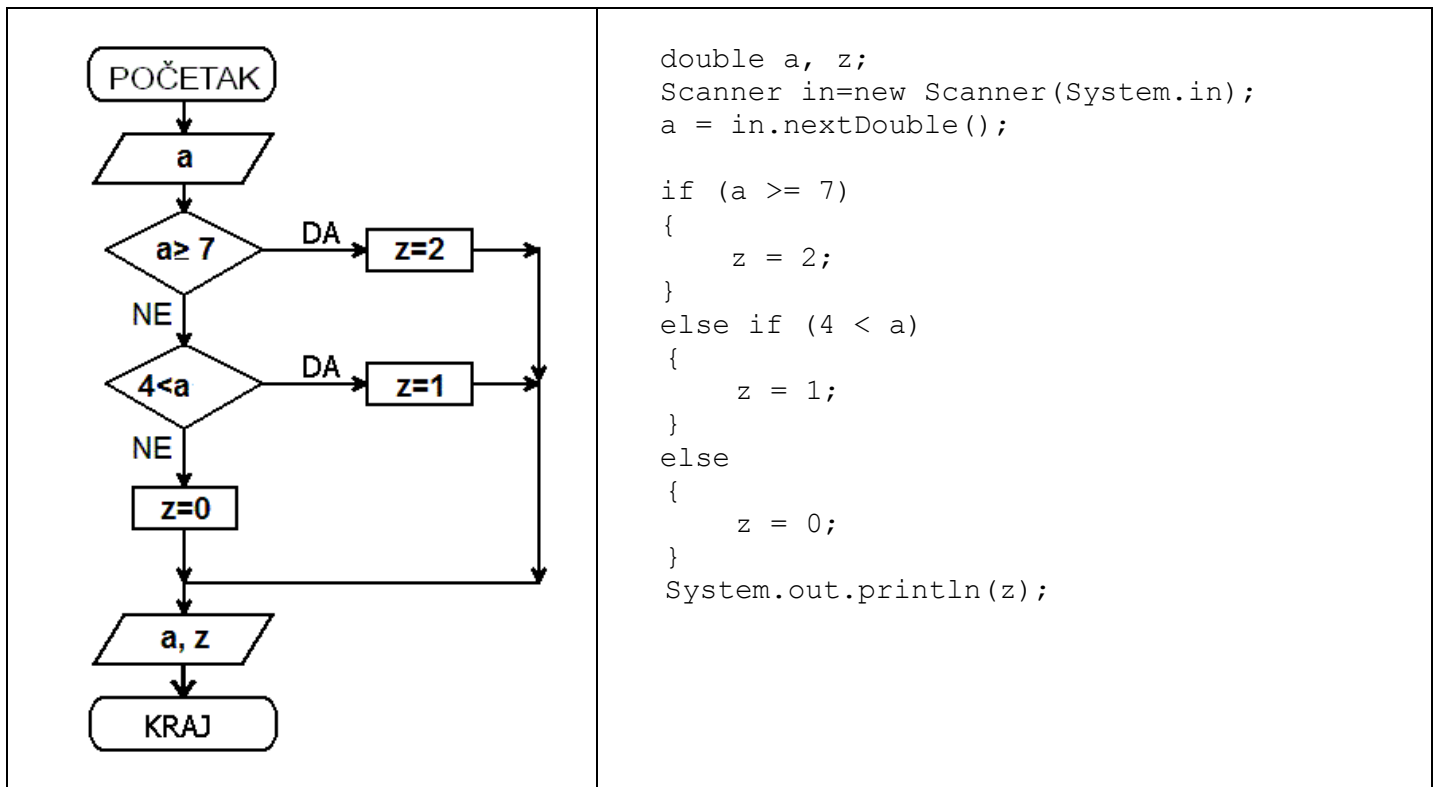
Opšti oblik višestruko razgranate strukture



```

if (Uslov 1)
{
    Blok 1
}
else if (Uslov 2)
{
    Blok 2
}
...
else if (Uslov n)
{
    Blok n
}
else
{
    Blok 0
}
  
```

Učitati broj a i štampati vrijednost izraza $z = \begin{cases} 2, & a \geq 7 \\ 1, & 4 < a < 7 \\ 0, & a \leq 4 \end{cases}$



Naredba switch

Opšti oblik naredbe switch

```

switch ( promjenljiva ) {
    case value_1:
        code_here_1;
        break;
    case value_2:
        code_here_2;
        break;
    ...
    case value_n:
        code_here_n;
        break;
    default:
        code_here_default;
}

```

Testira se vrijednost promjenljive. Ako je jednaka value_1, izvršava se kod code_here_1; ako je jednaka value_2, izvršava se kod code_here_2; ..., ako je jednaka value_n, izvršava se kod code_here_n. Ako nije jednaka nijednoj od vrijednosti value_1, ..., value_n, izvršava se kod code_here_default.

Ova naredba se koristi kao zamjena za više if-else naredbi.

Primjer 1: Na osnovu rednog broja mjeseca, štampati ime mjeseca:

```

int month = 8;
String monthString;
switch (month) {
    case 1: monthString = "January"; break;
    case 2: monthString = "February"; break;
    case 3: monthString = "March"; break;
    case 4: monthString = "April"; break;
    case 5: monthString = "May"; break;
    case 6: monthString = "June"; break;
    case 7: monthString = "July"; break;
}

```

```

        case 8: monthString = "August"; break;
        case 9: monthString = "September"; break;
        case 10: monthString = "October"; break;
        case 11: monthString = "November"; break;
        case 12: monthString = "December"; break;
        default: monthString = "Pogresan mjesec"; break;
    }
    System.out.println(monthString);

```

Prethodni primjer je ekvivalentan sa:

```

int month = 8;
if (month == 1) {
    System.out.println("January");
} else if (month == 2) {
    System.out.println("February");
}
... // i tako dalje, sve do decembra

```

Moguće je da kod za više slučajeva (case) bude isti – vidi primjer 2.

Primjer 2: Na osnovu rednog broja mjeseca u godini, štampati broj dana u mjesecu:

```

int month = 2;
int year = 2000;
int numDays = 0;
switch (month) {
    case 1: case 3: case 5:
    case 7: case 8: case 10:
    case 12:
        numDays = 31; // mjeseci sa 31 danom
        break;
    case 4: case 6:
    case 9: case 11:
        numDays = 30; // mjeseci sa 30 dana
        break;
    case 2: // februar
        if (((year % 4 == 0) && !(year % 100 == 0)) || (year % 400 == 0))
            numDays = 29; // prestupna godina
        else
            numDays = 28; // nije prstupna
        break;
    default: System.out.println("Pogresan mjesec."); break;
}
System.out.println("Number of Days = " + numDays);

```

Promjenljiva u naredbi switch mora biti cjelobrojna (byte, short, int, long), karakter (char) ili string (tipa String, od verzije Java SE 7)

Primjer 3: Upotreba tipa String u naredbi switch:

```

switch (month.toLowerCase()) {
    case "january": monthNumber = 1; break;
    case "february": monthNumber = 2; break;
    case "march": monthNumber = 3; break;
    case "april": monthNumber = 4; break;
    case "may": monthNumber = 5; break;
    case "june": monthNumber = 6; break;
    case "july": monthNumber = 7; break;
    case "august": monthNumber = 8; break;
    case "september": monthNumber = 9; break;
    case "october": monthNumber = 10; break;
    case "november": monthNumber = 11; break;
}

```

```
    case "december": monthNumber = 12; break;
    default: monthNumber = 0; break;
}
```

Ciklusi

1. Naredba while

- a. Opšti oblik naredbe while:

```
while(logicki uslov)
```

```
{
```

```
    Naredbel
```

```
}
```

- b. Logički uslov (engl. boolean expression) mora imati vrijednost `true` ili `false`. Blok `Naredbel`, se izvršava sve dok je logički uslov tačan (`true`); kada uslov postane netačan (`false`) izvršava se prva naredba iza naredbe `while`.
- c. Obratite pažnju da poslije `while` **nema** simbola tačka-zapeta.

2. Naredba do ... while

- a. Opšti oblik naredbe `do...while`:

```
do {
```

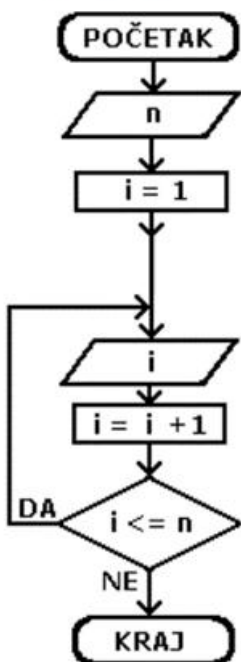
```
    Naredbel
```

```
} while(logicki uslov);
```

- b. Logički uslov (engl. boolean expression) mora imati vrijednost `true` ili `false`. Blok `Naredbel`, se izvršava sve dok je logički uslov tačan (`true`); kada uslov postane netačan (`false`) izvršava se prva naredba iza naredbe `while`.
- c. Obratite pažnju da poslije `while` **mora da stoji** simbol tačka-zapeta.

Učitati prirodan broj `n` i štampati sve prirodne brojeve od 1 do `n`.

Prvi način



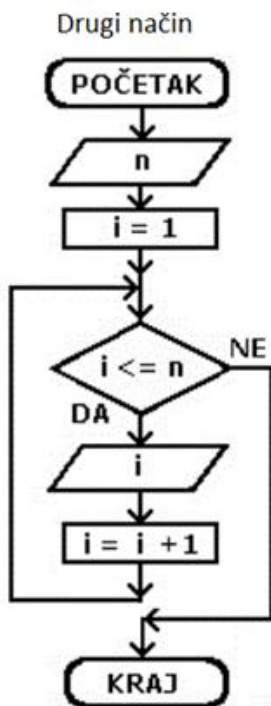
```
import java.util.Scanner;
public class Hello1 {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);
        int n = in.nextInt();

        int i = 1;
        while (i<=n)
        {
            System.out.println(i);
            i = i +1;
        }
    }
}
```

Učitati prirodan broj n i štampati sve prirodne brojeve od 1 do n.



```
import java.util.Scanner;

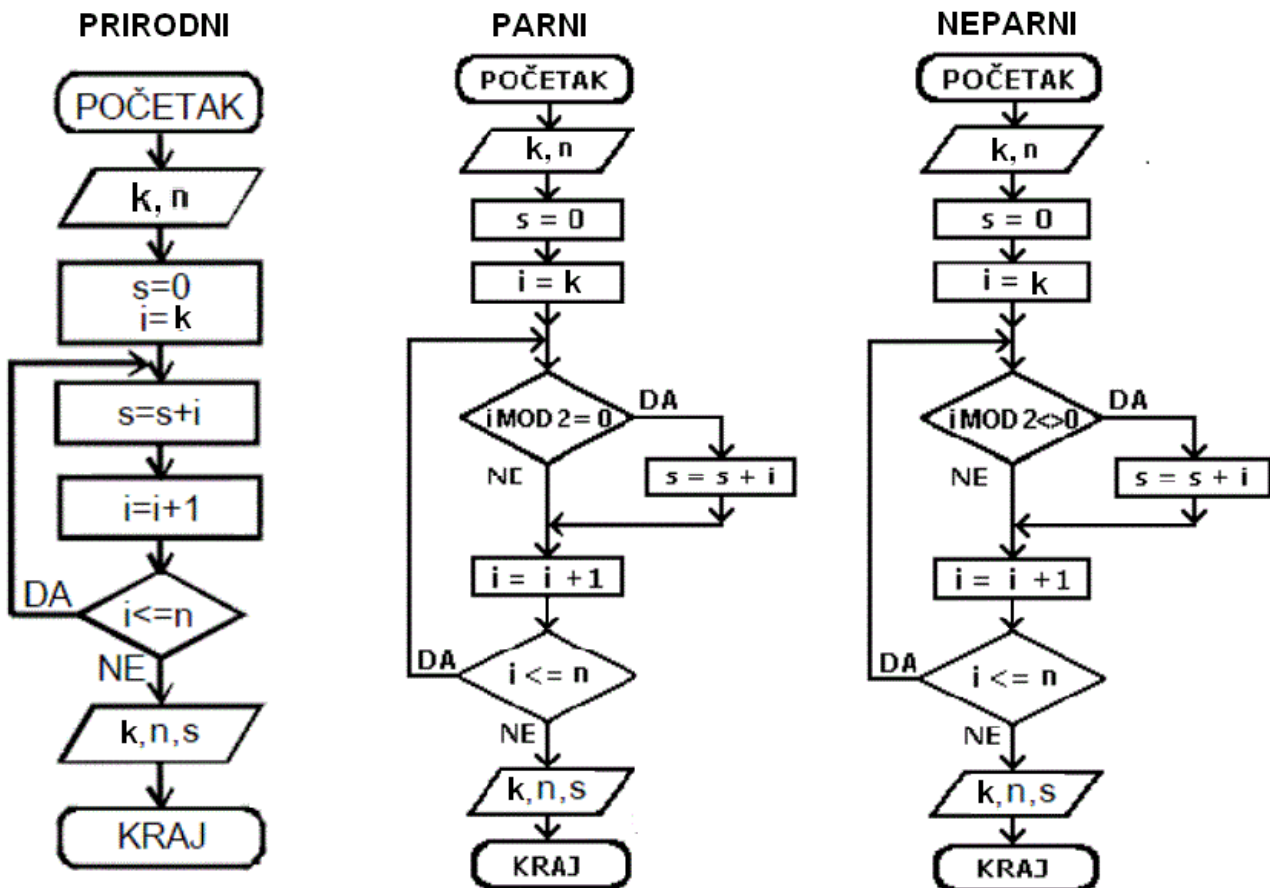
public class Hello2 {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);
        int n = in.nextInt();

        int i = 1;
        do
        {
            System.out.println(i);
            i = i + 1;
        }
        while (i<=n);
    }
}
```


Učitati prirodne brojeve k i n ($k \leq n$) i štampati: (a) brojeve k i n i zbir svih prirodnih brojeva od k do n ; (b) brojeve k i n i zbir svih neparnih prirodnih brojeva od k do n ; (c) brojeve k i n i zbir svih neparnih prirodnih brojeva od k do n .



Rješenje (a):

```
import java.util.Scanner;
public class Hello3 {

    public static void main(String[] args) {
        int k= 10, n = 20, s = 0, i = 1;
        Scanner in = new Scanner(System.in);
        k = in.nextInt();
        n = in.nextInt();
        for (i=k; i<= n; i = i+1)
        {
            s = s + i;
        }
        System.out.println(k + " " + n + " " + s );
    }
}
```

Rješenje (b):

```
import java.util.Scanner;
public class Hello4 {

    public static void main(String[] args) {
        int s = 0, i;
        Scanner in = new Scanner(System.in);
        int k = in.nextInt();
        int n = in.nextInt();

        i = k;
        while(i <= n)
```

```
{
    if (i%2 == 0)
    {
        s = s + i;
    }
    i = i+1;
}
System.out.println(k + " " + n + " " + s );
}
}
```

Rješenje (c):

```
import java.util.Scanner;
```

```
public class Hello5 {
```

```
    public static void main(String[] args) {
```

```
        int k= 10, n = 20, s = 0, i;
```

```
        Scanner in = new Scanner(System.in);
```

```
        k = in.nextInt();
```

```
        n = in.nextInt();
```

```
        i = k;
```

```
        do
```

```
        {
```

```
            if (i%2 != 0)
```

```
            {
```

```
                s = s + i;
```

```
            }
```

```
            i = i+1;
```

```
        }
```

```
        while(i <= n);
```

```
        System.out.println(k + " " + n + " " + s );
```

```
    }
```

```
}
```