

## Zadaci – 8b, Petlja „FOR” (drugi dio)

### Zadatak 89:

Napravite dijagram toka, i na osnovu njega napišite C++ kôd za sljedeći zadatak:

Napravite program koji će pitati korisnika koliko želi unijeti brojeva. Zatim program treba zahtijevati unos toliko brojeva (mogu biti i decimalni brojevi). Nakon što je korisnik unio sve brojeve, program treba ispisati njihovu sumu.

Primjer rada program da je na sljedećoj slici.

Riješen dijagram toka i C++ kôd se nalaze na kraju dokumenta.

Pomoć:

- od korisnika zahtijevajte unos broja  $n$  (pokazuje koliko brojeva se unosi)
- varijablu  $s$  za sumu postavite na nulu
- u *for*-petlju dodajte:
  - o unos broja u varijablu  $x$  (koja je tipa *float*)
  - o sumu uvećajte za vrijednost  $x$
- naredbe u *for*-petlji ponovite  $n$ -puta
- na kraju programa (izvan *for*-petlje) ispišite sumu  $s$

```
Koliko brojeva zelite unijeti: 5
Unesite 1. broj: 7
Unesite 2. broj: 4
Unesite 3. broj: 3
Unesite 4. broj: 2
Unesite 5. broj: 6
Suma = 22
```

### Zadatak 90:

Napišite C++ kôd za sljedeći zadatak:

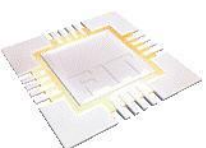
Napravite program koji će pitati korisnika koliko želi unijeti brojeva. Zatim program treba zahtijevati unos toliko brojeva (mogu biti i decimalni brojevi). Nakon što je korisnik unio sve brojeve, program treba ispisati aritmetičku sredinu kvadrata unesenih brojeva. Program trebate formatirati prema sljedećem primjeru:

```
Koliko brojeva zelite unijeti: 4

Unesite 1. broj: 3
..... + 9
Unesite 2. broj: 5
..... + 25
Unesite 3. broj: 4
..... + 16
Unesite 4. broj: 2
..... + 4
..... =====
..... = 54

aritmeticka sredina kvadrata je 13.5
```

Riješen dijagram toka i C++ kôd se nalaze na kraju dokumenta.



**Pomoć:**

- u *for*-petlju dodajte:
  - o unos broja u varijablu *x* (koja je tipa *float*)
  - o sumu uvećajte za vrijednost *x\*x*
  - o ispišite na ekran čitav red tačaka
  - o ispišite na ekran vrijednost *x\*x*
- izvan *for*-petlje izračunajte prosječnu vrijednost
- ispišite sumu kvadrata i prosječnu vrijednost

**Zadatak 91:**

Napravite dijagram toka, i na osnovu njega napišite C++ kôd za sljedeći zadatak:

Napravite program koji će pitati korisnika koliko želi unijeti brojeva. Zatim program treba zahtijevati unos toliko cijelih brojeva. Nakon što je korisnik unio sve te brojeve, program treba ispisati koliko ima neparnih brojeva i koliko ima negativnih brojeva.

Riješen dijagram toka i C++ kôd se nalaze na kraju dokumenta.

```
Koliko brojeva želite unijeti: 5
Unesite 1. broj: 3
Unesite 2. broj: -5
Unesite 3. broj: 4
Unesite 4. broj: 9
Unesite 5. broj: -15
neparni brojevi: 4
negativni brojevi: 2
```

**Pomoć:**

- od korisnika zahtijevajte unos varijable *n* da znate koliko ćete brojeva zahtijevati za unos od korisnika
- deklarirati ćemo dva dodatna brojača *b1* i *b2* (ovo nisu brojači *for*-petlje)
- brojač *b1* će brojati sve neparne brojeve – njegova početna vrijednost prije *for*-petlje mora biti nula
- brojač *b2* će brojati sve negativne brojeve (i njega ćemo inicijalizovati na nulu)
- u *for*-petlju dodajte:
  - o unos broja u varijablu *x* koja je tipa **int**
  - o if-iskaz: ako je broj *x* neparan, onda: povećajte brojač *b1* za 1, inače: ništa
  - o if-iskaz: ako je broj *x* pozitivan, onda: povećajte brojač *b2* za 1, inače: ništa
- naredbe u *for*-petlji ponovite *n*-puta
- ispišite *b1* i *b2*

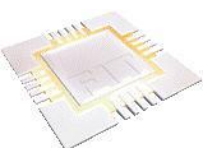
**Zadatak 92:**

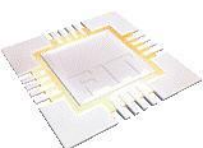
Napravite dijagram toka, i na osnovu njega napišite C++ kôd za sljedeći zadatak:

Napravite program koji će pitati korisnika koliko želi unijeti brojeva. Zatim program treba zahtijevati unos toliko cijelih brojeva. Nakon što je korisnik unio sve te brojeve, program treba ispisati aritmetičku sredinu parnih brojeva, kao i aritmetičku sredinu kubova svih unesenih brojeva.

Riješen dijagram toka i C++ kôd se nalaze na kraju dokumenta.

```
Koliko brojeva želite unijeti: 5
Unesite 1. broj: 2
Unesite 2. broj: 4
Unesite 3. broj: 7
Unesite 4. broj: 33
Unesite 5. broj: 19
Arit. sr. parnih brojeva: 3
Arit. sr. kubova: 8642.2
```





**Pomoć:**

- deklarirate samo jedan dodatni brojač  $b$  za parne brojeve
- deklarirate varijablu  $s1$  za sumu parnih brojeva
- deklarirate varijablu  $s2$  za sumu kubova svih brojeva
- u *for*-petlju dodajte:  
...
- naredbe u *for*-petlji ponovite  $n$ -puta
- ispišite vrijednost  $s1/b$  (možete koristiti i dodatnu varijablu za prosjek)
- ispišite vrijednost  $s2/n$

**Zadatak 93:**

Programi ponuđeni u rješenjima za zadatak 90 i 92 će pasti ako korisnik unese broj 0 na pitanje koliko on želi brojeva unijeti u računar. *Zašto?*

Ako korisnik unese vrijednost 0 za varijablu  $n$ , *for*-petlja se neće izvršiti. Greška nastaje prilikom računanja prosjeka kod dijeljenja sa brojem  $n$ .

Prepravite ovaj nedostatak u programima 90 i 92.

**Pomoć za program 92:**

Dodajte samo jedan *if-else*-iskaz koji će računati i ispisivati prosjek samo ako je  $n$  različito od nule.

**Pomoć za program 92:**

U ovom zadatku imate i brojač  $b$  koji ne smije biti nula prilikom računanja prosjeka. Dodajte jedan *if-else*-iskaz za brojač  $b$  i jedan *if-else*-iskaz za broj  $n$ .

**Zadatak 94:**

Napišite C++ kôd za sljedeći zadatak:

Napravite program koji će računati aritmetičku sredinu brojeva od 1 do 100, ali izuzimajući brojeve od 40 do 60.

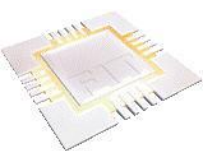
Znači, potrebno je izračunati aritmetičku sredinu za brojeve:

1, 2, 3, ..., 37, 38, 39, 61, 62, 63, ..., 99, 100

Riješen dijagram toka i C++ kôd se nalaze na kraju dokumenta.

**Pomoć:**

- koristite varijablu  $s$  za sumu
- koristite brojač  $b$  koji će brojati brojeve koje zadovoljavaju dati uslov
- neka *for*-petlja mijenja svoj brojač  $i$  od 1 do 100
- u *for*-petlju dodajte *if*-iskaz koji će, ako je broj trenutni broj  $i$  zadovoljio uslov, uvećati brojač i povećati sumu
- kako glasi uslov za *if*-iskaz? kad je broj  $i$  zadovoljio uslov?
- vi trebati samo provjeriti da li je trenutno  $i$  manje od 40 i veće 60
- provjeru da li je  $i$  manje od 1 ili veće od 100 ne trebate vršiti, jer vrijednost  $i$  može biti samo između 1 i 100, nikako van toga



- da li trenutno broj  $i$  zadovoljio zahtijevani uslov ako vrijedi  $(i \leq 40 \ \&\& \ i \geq 60)$  ?
  - nije, zbog dva razloga:
    1. Granične vrijednosti ne smijemo uzeti u računanje!  
Novi prepravljn uslov glasi:  
 $(i \leq 39 \ \&\& \ i \geq 61)$
    2. Ni prethodno napisan uslov nije ispravan. *Zašto?*  
Pogledajte taj uslov matematički. Nabrojte nekoliko brojeva koji zadovoljavaju taj uslov!  
  
Npr:
      - broj 50 ne zadovoljava uslov jer nije veći od 60, a nije ni manji od 40
      - broj 70 ne zadovoljava uslov jer nije manji od 40
      - broj 10 ne zadovoljava uslov jer nije veći od 60
$$(i \leq 40) \cap (i \geq 60) = \emptyset$$
- Ni jedan broj ne zadovoljava gornji uslov. Znači, trebat ćete uslov „ $(i \leq 39 \ \&\& \ i \geq 61)$ “ malo prepraviti. Pokušajte sami.

### Zadatak 95:

Napravite dijagram toka, i na osnovu njega napišite C++ kôd za sljedeći zadatak:

Napravite program koji će pitati korisnika koliko želi unijeti brojeva. Zatim program treba zahtijevati unos toliko cijelih brojeva. Nakon što je korisnik unio sve te brojeve, program treba ispisati najveći uneseni broj.

Riješen dijagram toka i C++ kôd se nalaze na kraju dokumenta.

*Pomoć:*

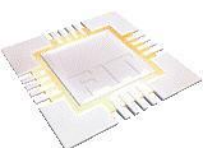
- koristit ćemo varijablu  $m$  koja će pamtit koji je do sada najveći uneseni broj
- u *for*-petlju ćemo dodati
  - o unos broja  $x$  sa tastature
  - o *if*-iskaz koji će, *ako je*  $m$  manje od  $x$ , varijabli  $m$  dodijeliti vrijednost  $x$ , *inače* neće se ništa izvršiti

Mi smo do sada brojačima i sumama postavljali početnu vrijednost na nulu. Sada pitanje glasi: *Koja je početna vrijednost varijable  $m$ ?*

*Prvi način:*

Ako bi se dogovorili da naš program može raditi samo sa brojevima od -10 000 do +10 000, onda bi početna vrijednost varijable  $m$  mogla biti -10 001. U tom slučaju bi *for*-petlja radila sljedeće (slijedi primjer):

- 
- korisnik unosi broj  $x$  (npr.  $x = -123$ ) koji bi trebao biti veći od početne vrijednosti
  - pošto je  $m$  manje od  $x$  ( $-10000 < -123$ ) uslov *if*-iskaza je zadovoljen pa će izvršiti naredba  $m = x;$  ( $m = -123$ )



- 
- zatim korisnik unosi opet broj  $x$  (neka to bude  $-200$ )
  - uslov  $m < x$  ( $-123 < -200$ ) nije zadovoljen, varijabla  $m$  ostaje neizmijenjena
- 

- zatim korisnik unosi opet broj  $x$  (neka to bude  $33$ )
  - uslov  $m < x$  ( $-123 < 33$ ) je zadovoljen, varijabla  $m$  dobija vrijednost  $33$
- 

...

- na kraju će vrijednost  $m$  biti najveći broj od svih unesenih brojeva

#### Drugi način:

Bolje rješenje ovog zadatka bi bilo kada bi početna vrijednost varijable  $m$  bila vrijednost prvog unesenog broja  $x$ . To možemo učiniti na dva načina:

- a) Dodat ćemo još jedan unos sa tastature broja  $m$  prije *for*-petlje, a *for*-petlju ćemo ponavljati samo  $(n-1)$  puta.  
 Probajte program riješiti i na ovakav način!

- b) Kad se petlja bude **prvi put** izvršavala, varijabli  $m$  ćemo dodijeliti vrijednost  $x$ . To možemo implemetirati na dva načina:

- b1) Koristit ćemo **if-else**-iskaz (dvostruki izbor), koji će, ako se petlja prvi put izvršava ( $i==1$ ):

- varijabli  $m$  dodijeliti vrijednost  $x$ ,

inače (ako se petlja ne izvršava prvi put):

- izvršit će se isti *if*-iskaz kao u prethodnim rješenjima

- b2) Koristit ćemo **if**-iskaz (jednostruki izbor), koji će, ako se petlja prvi put izvršava ( $i==1$ ):

- varijabli  $m$  dodijeliti vrijednost  $x$

Poslije tog *if*-iskaza će se, bez obzira da li se petlja izvršava prvi put ili ne, izvršiti isti *if*-iskaz kao u prethodnim rješenjima. To se smije izvršiti prvi put kada se petlja ponavlja, jer će varijabla  $m$  imati neku vrijednost.

#### Zadatak 96:

Napravite dijagram toka, i na osnovu njega napišite C++ kôd za sljedeći zadatak:

Napravite program koji će zahtijevati unos broja  $n$ . Program treba ispisati i izbrojiti sve brojeve od  $2$  do  $10$  sa kojima je broj  $n$  djeljiv. Program formatirajte na sljedeći način:

```
Unesi broj n: 18

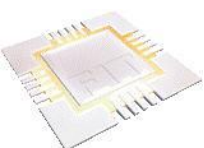
1. Broj 18 je djeljiv sa 2
2. Broj 18 je djeljiv sa 3
3. Broj 18 je djeljiv sa 6
4. Broj 18 je djeljiv sa 9

Broj 18 je djeljiv sa ukupno puta: 4
```

Riješen dijagram toka i C++ kôd se nalaze na kraju dokumenta.

#### Pomoć:

- koristite brojač  $b$  kojeg trebate povećati svaki put kada broj  $n$  bude djeljiv sa  $i$



**Zadatak 97:**

Napravite dijagram toka, i na osnovu njega napišite C++ kôd za sljedeći zadatak:

Napišite program koji će za uneseni broj provjeriti je li riječ o prostom ili složenom broju:

Unesi broj n:

Unesi broj n:

Riješen dijagram toka i C++ kôd se nalaze na kraju dokumenta.

Šta je to prost broj? Pomoć:

Primjer prostih brojeva: 3, 5, 7, 11, 13, 17, 19, 23, 31, 37, ...

Definicija prostog broja glasi:

Broj **n** je prost ako je djeljiv **SAMO** sa 1 i sa samim sobom (**n**).

U programiranju ne postoji iskaz kojim možemo reći 'SAMO', ali možemo reći djeljiv ili nije djeljiv sa tim i tim i tim... S toga morat ćemo našu definiciju promijeniti tako da ne koristimo riječ 'samo'. Evo korisnije definicije:

Broj **n** je prost ako **NIJE** djeljiv ni sa jednim brojem osim sa **1** i sa **n**.

Ista definicija ali bez riječi 'osim':

Broj **n** je prost ako **NIJE** djeljiv ni sa jednim brojem u intervalu od **2** do (**n-1**).

Kako riješiti zadatak?

Ako je broj **n** djeljiv makar sa jednim brojem od 2 do (**n-1**), onda on nije prost.

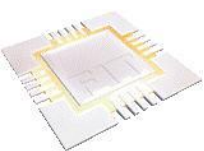
Na osnovu gornjeg zaključka mi ćemo praviti naš program. Koristit ćemo *for*-petlju da bi izbrojili sa koliko brojeva je broj **n** djeljiv.

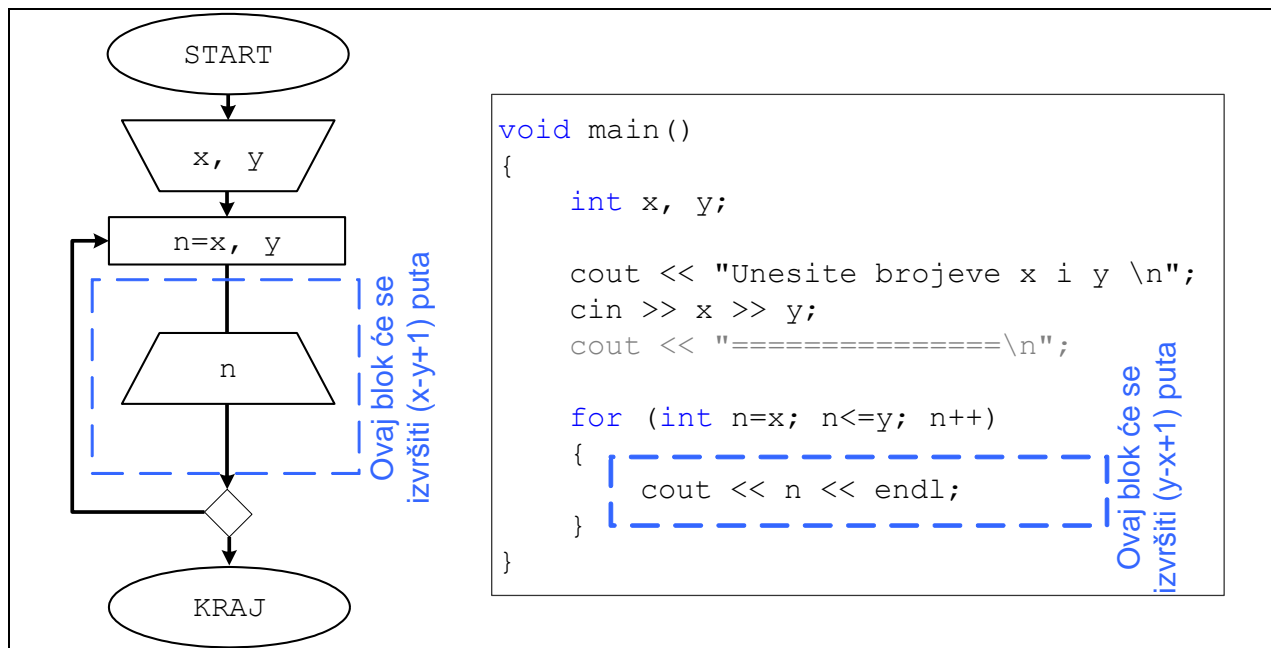
Ovaj program je sličan prethodnom programu (zadatak 98). U njemu smo brojili sa koliko brojeva (u intervalu od 2 do 10) je broj **n** djeljiv. To ćemo činiti i ovdje ali u intervalu od 2 do (**n-1**). Ako bi takav program ispisao da je broj **n** djeljiv 0 puta, to bi značilo da je broj **prost**, inače, u slučaju da program ispiše 1, 2, 3 ili bilo koji drugi broj osim nule, značilo bi da uneseni broj **n nije prost**. I ovakav program bi bio koristan, ali ne bi bio *user-friendly*. S toga trebamo program prepraviti da umjesto nekog broja ispiše jasnu poruku: '...je prost' ili '...NIJE prost'. Iz prethodnog program trebat ćemo još izbaciti nepotrebnu *cout*-naredbu za ispis poruke "Broj ... je djeljiv sa ...".

Napravit ćemo dijagram toka i C++ kôd za program koji će ispisivati brojeve od  $x$  do  $y$ . Korisnik bi trebao unijeti vrijednosti  $x$  i  $y$ .

Sad ćemo koristiti varijablu  $n$  umjesto  $i$  kao brojač *for*-petlje. Vidjet ćemo kasnije zašto.

Evo *for*-petlje.:





Kad bi tijelo ove *for*-petlje (naredba "cout << n << endl") izbacili a ubacili program za provjeru prostog broja (iz zadatka 97), dobili našu prvu ugniježdenu *for*-petlju. Takav program bi trebao ispisivati odgovor (na pitanje da li je riječ o prostom broju) za svaki broj iz datog intervala.

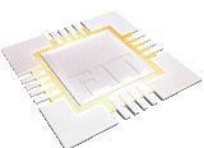
Program bi trebao raditi na sljedeći način:

```

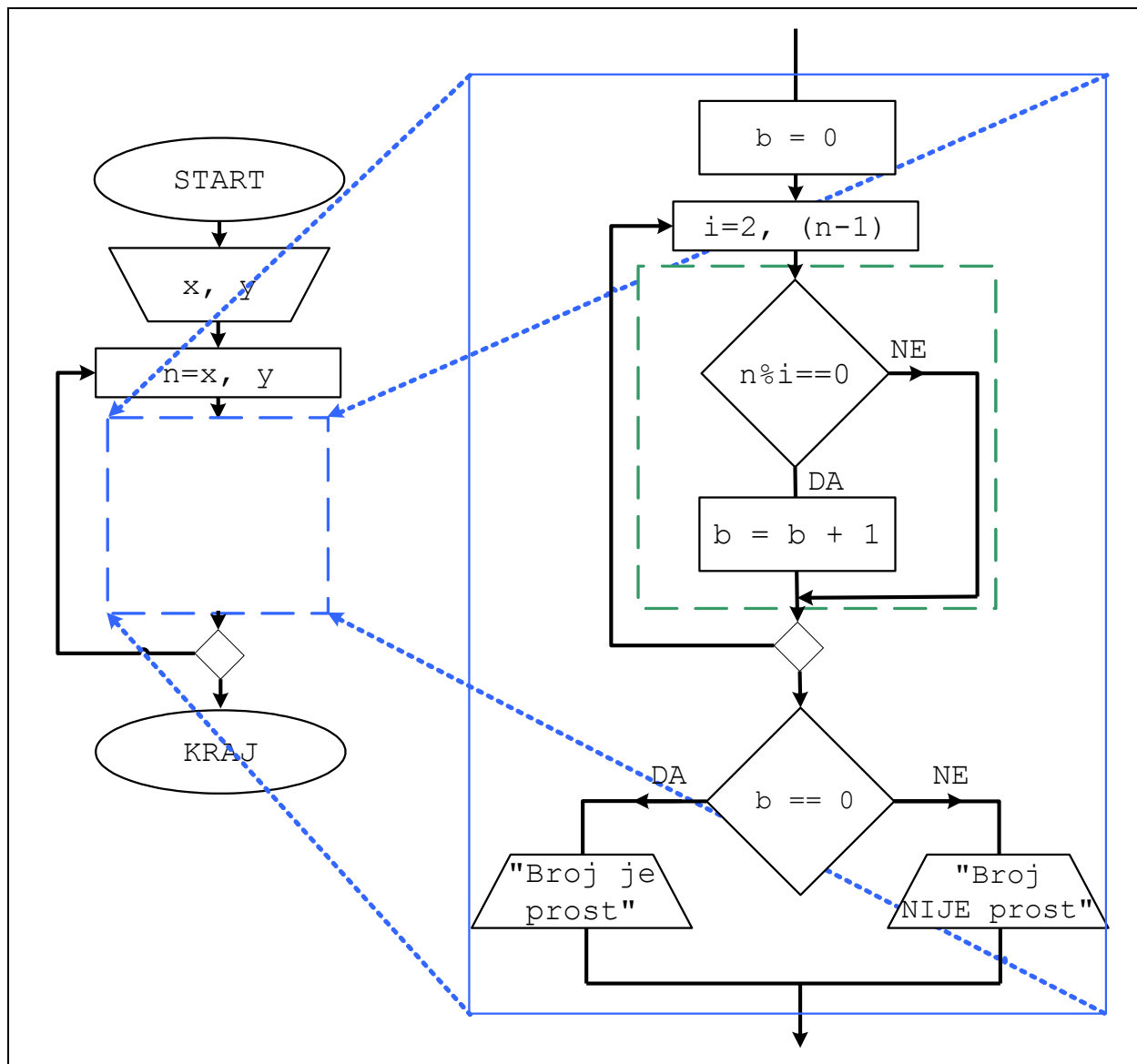
Unesite brojeve x i y
2
6
=====
Broj 2 je prost.
Broj 3 je prost.
Broj 4 NIJE prost
Broj 5 je prost.
Broj 6 NIJE prost

```

Dijagram toka datog programa je prikazan na sljedećoj slici.







### Zadatak 98:

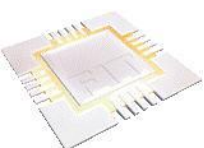
Na osnovu gore datog dijagrama toka napišite C++ kôd!

Cilj ovog zadatka je da napišite kôd i da budete čak prije kompajliranja prilično sigurni da će ovaj vaš veoma komplikovani program raditi (nećemo gledati na sintaksne greške koje svako mora napraviti).

Nakon što napišete ovaj C++ kôd koji je za vas sada veoma jednostavan i razumljiv, pokažite ga kolegi(ci) bez dijagrama toka i pitajte da li je taj kôd jednostavan :)

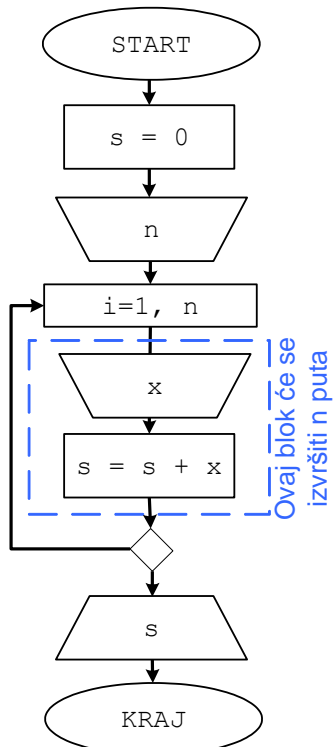
Nadamo se da ste sada shvatili važnost dijagrama toka.

*Riješen dijagram toka i C++ kôd se nalaze na kraju dokumenta.*



## Rješenja

### Rješenje zadatka br. 91:



```

1:  #include <iostream>
2:  using namespace std;
3:
4:  void main()
5:  {
6:      int n;
7:      float s, x;
8:      s = 0;
9:      cout << "Koliko brojeva zelite unijeti: ";
10:     cin >> n;
11:
12:     for (int i=1; i<=n; i++)
13:     {
14:         cout << " Unesite " << i << ". broj: ";
15:         cin >> x;
16:         s = s + x;
17:     }
18:
19:     cout << "Suma = " << s << endl;
20: }

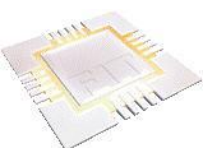
```

### Rješenje zadatka br. 92:

```

4:  void main()
5:  {
6:      int n;
7:      float s, x, pr;
8:      s = 0;
9:      cout << "Koliko brojeva zelite unijeti: ";
10:     cin >> n;
11:
12:     cout << endl;
13:
14:     for (int i=1; i<=n; i++)
15:     {
16:         cout << " Unesite " << i << ". broj: ";
17:         cin >> x;
18:         cout << "..... + " << x*x << endl << endl;
19:         s = s + x*x;
20:     }
21:
22:     cout << "..... =====\n\n";
23:     cout << "..... = " << s << endl << endl;
24:
25:     pr = s / n;
26:     cout << "Aritmeticka sredina kvadrata je " << pr << endl;
27: }

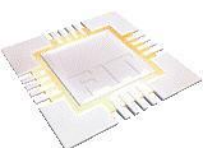
```

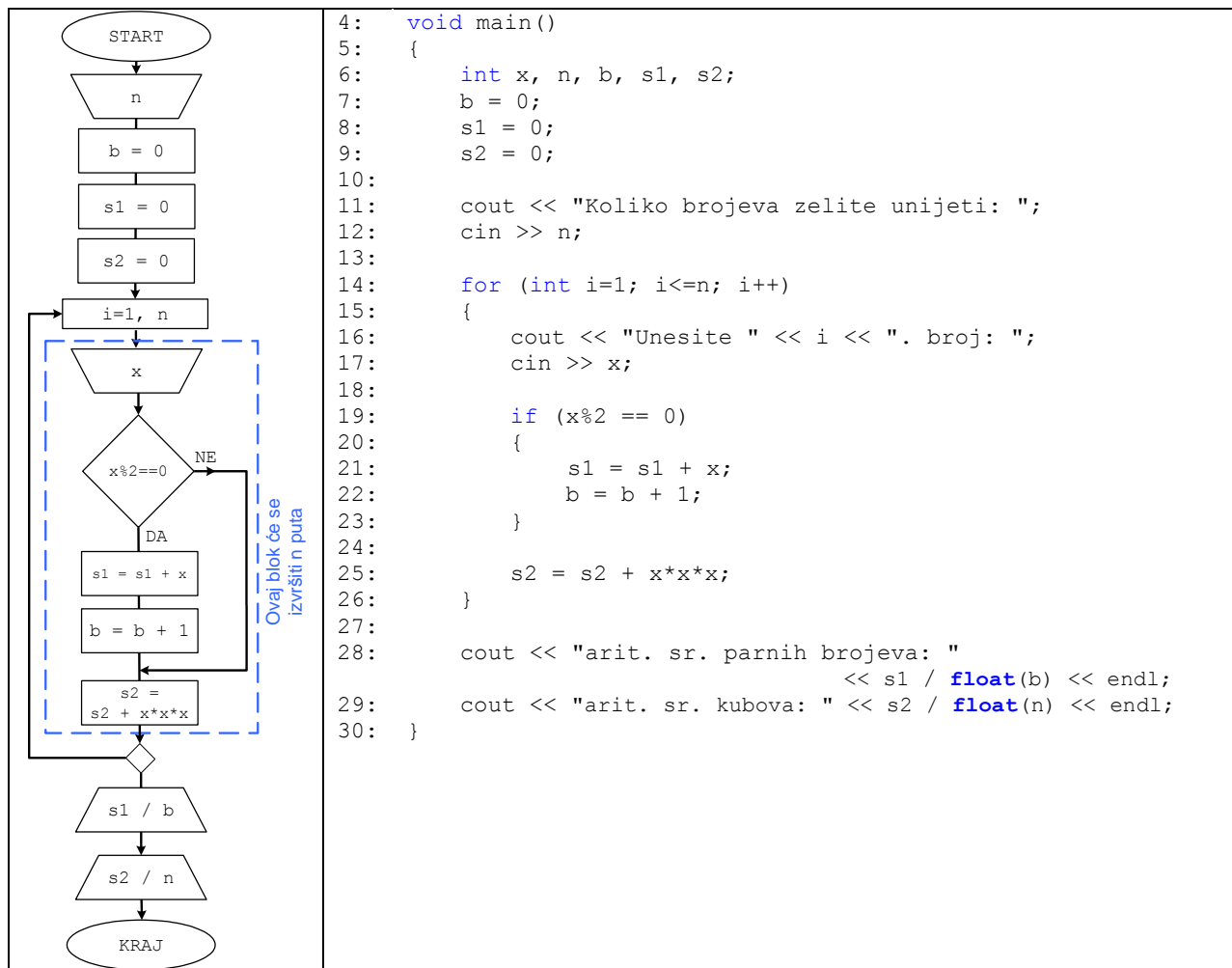


Rješenje zadatka br. 93:

```
4: void main()
5: {
6:     int n, b1, b2, x;
7:
8:     b1 = 0;
9:     b2 = 0;
10:
11:     cout << "Koliko brojeva zelite unijeti: ";
12:     cin >> n;
13:
14:     for (int i=1; i<=n; i++)
15:     {
16:         cout << " Unesite " << i << ". broj: ";
17:         cin >> x;
18:
19:         if (x%2 != 0)
20:             b1 = b1 + 1;
21:
22:         if (x < 0)
23:             b2 = b2 + 1;
24:     }
25:
26:     cout << "Neparni brojevi: " << b1 << endl;
27:     cout << "Negativni brojevi: " << b2 << endl;
28: }
```

Rješenje zadatka br. 94:





#### Rješenje zadatka br. 96:

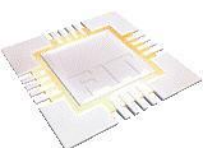
```

4: void main()
5: {
6:     int b, s;
7:     b = 0;
8:     s = 0;
9:
10:    for (int i=1; i<=100; i++)
11:    {
12:        if (i<=39 || i>=61)
13:        {
14:            b = b + 1;
15:            s = s + i;
16:        }
17:    }
18:
19:    cout << "aritmeticka sredina: " << s / float(b) << endl;
20: }

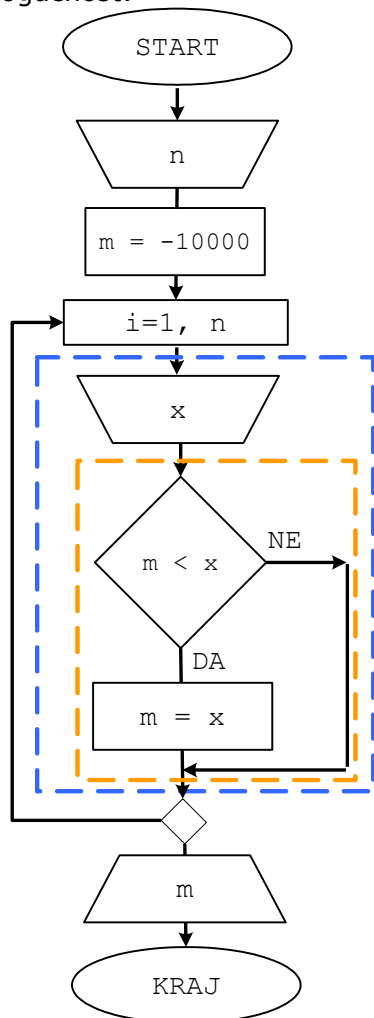
```

Ovaj program treba ispisati: 'aritmeticka sredina: 50.6329'

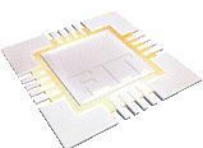
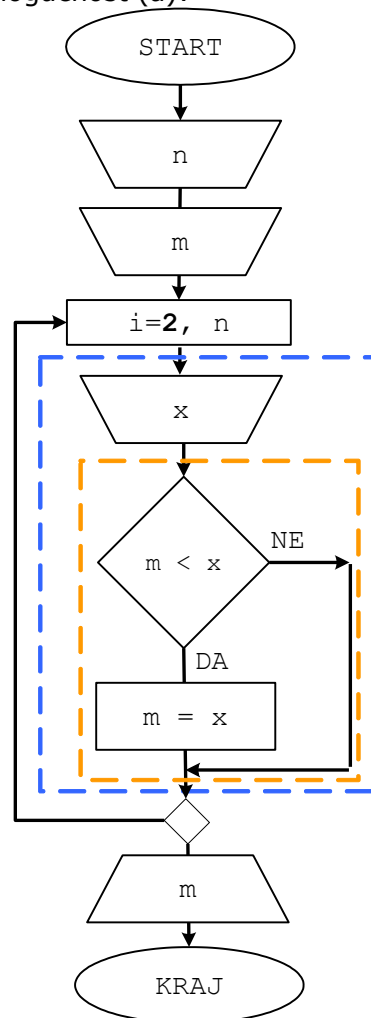
#### Rješenje zadatka br. 97:



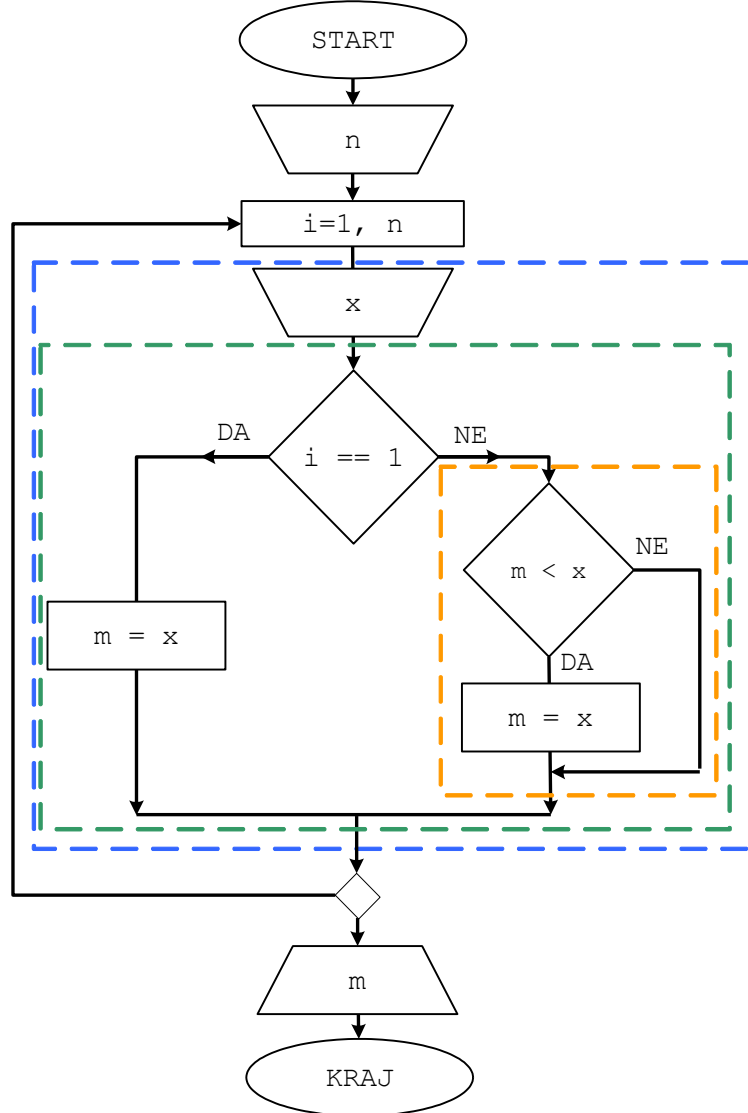
Prva mogućnost:



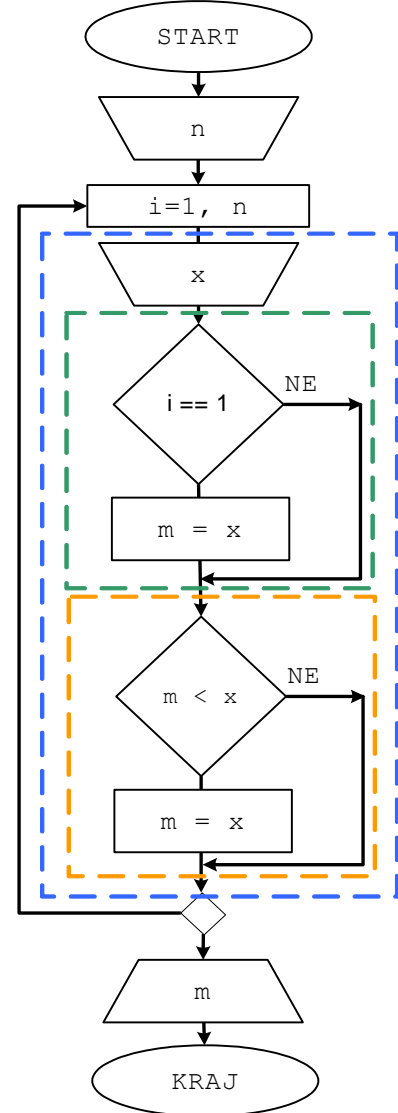
Druga mogućnost (a):



Druga mogućnost (b1):



Druga mogućnost (b2):



Prva mogućnost:

```

4: void main()
5: {
6:     int m, n, x;
7:
8:     cout << "Unesite n: ";
9:     cin >> n;
10:
11:
12:     m = -10000;
13:
14:     for (int i=1; i<=n; i++)
15:     {
16:         cout << i << ": ";
17:         cin >> x;
18:
19:         if (m < x)
20:         {
21:             m = x;
22:         }
23:     }
24:
25:     cout << "\nNajveci broj je " << m;
26: }
    
```

Druga mogućnost (a):

```

void main()
{
    int m, n, x;

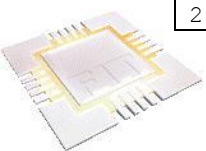
    cout << "Unesite n: ";
    cin >> n;

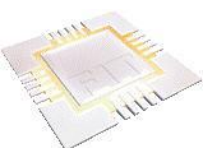
    cout << "1: ";
    cin >> m;

    for (int i=2; i<=n; i++)
    {
        cout << i << ": ";
        cin >> x;

        if (m < x)
        {
            m = x;
        }
    }

    cout << "\nNajveci broj je " << m;
}
    
```

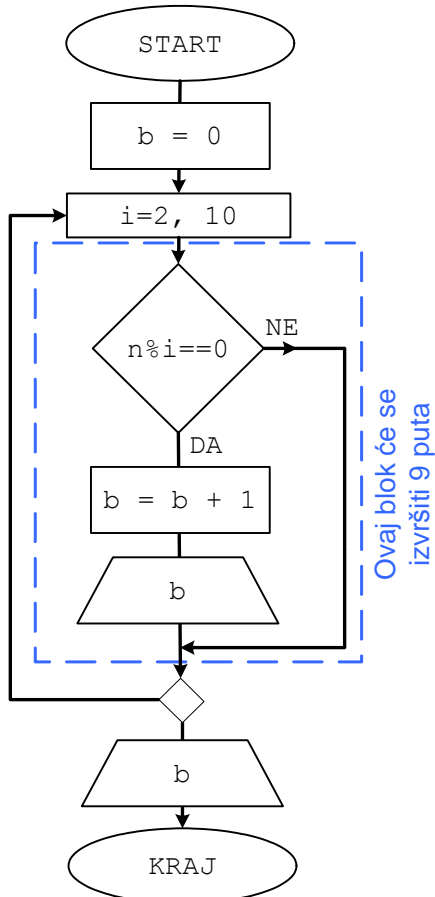




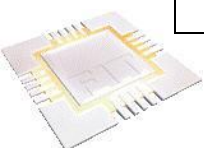
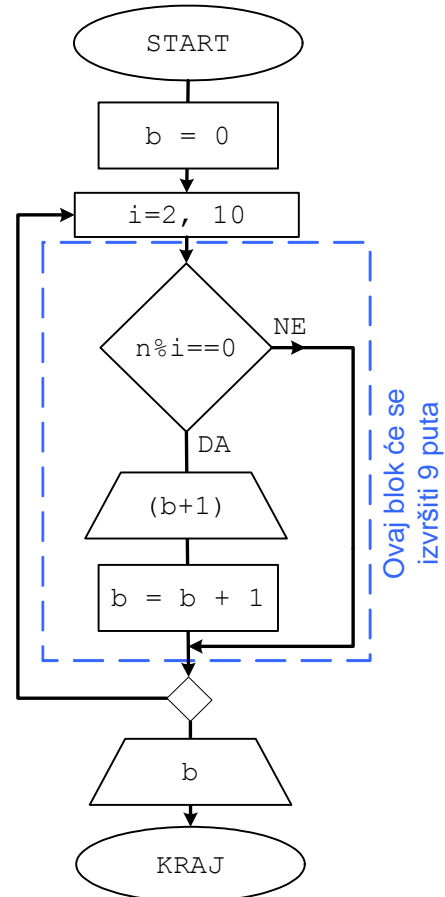
Druga mogućnost (b1):	Druga mogućnost (b2):
<pre> 4: void main() 5: { 6:     int m, n, x; 7: 8:     cout &lt;&lt; "Unesite n: "; 9:     cin &gt;&gt; n; 10: 11:     for (int i=1; i&lt;=n; i++) 12:     { 13:         cout &lt;&lt; i &lt;&lt; ": "; 14:         cin &gt;&gt; x; 15: 16:         if (i==1) 17:         { 18:             m = x; 19:         } 20:         else 21:             // ako se petlja ne izvršava prvi put 22: 23:             if (m &lt; x) 24:             { 25:                 m = x; 26:             } 27:             // ako se petlja nije izvršavala prvi put 28:         } // kraj for-petlje 29: 30:     cout &lt;&lt; "\nNajveci broj je " &lt;&lt; 31:     m; 32: }</pre>	<pre> void main() {     int m, n, x;      cout &lt;&lt; "Unesite n: ";     cin &gt;&gt; n;      for (int i=1; i&lt;=n; i++)     {         cout &lt;&lt; i &lt;&lt; ": ";         cin &gt;&gt; x;          if (i==1)         {             m = x;         }          if (m &lt; x)         {             m = x;         }     }      cout &lt;&lt; "\nNajveci broj je " &lt;&lt;     m; }</pre> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>/* prilikom izvršavanja ove linije koda varijabla m ima neku vrijednost; ako se petlja izvršava prvi put, varijabla m je tek u gornjoj liniji koda dobila neku vrijednost */</p> </div>

### Rješenje zadatka br. 98:

prvi način:



drugi način:





**prvi način:**

```

4: void main()
5: {
6:     int n, b;
7:     b = 0;
8:
9:     cout << "Unesi broj n: ";
10:    cin >> n;
11:
12:    cout << endl;
13:
14:    for (int i=2; i<=10; i++)
15:    {
16:        if (n%i == 0)
17:        {
18:            b = b + 1;
19:            cout << b << ". Broj " << n << " je djeljiv sa " << i << endl;
20:        }
21:    }
22:
23:    cout << "\nBroj " << n << " je djeljiv sa ukupno puta: " << b << endl;
24: }

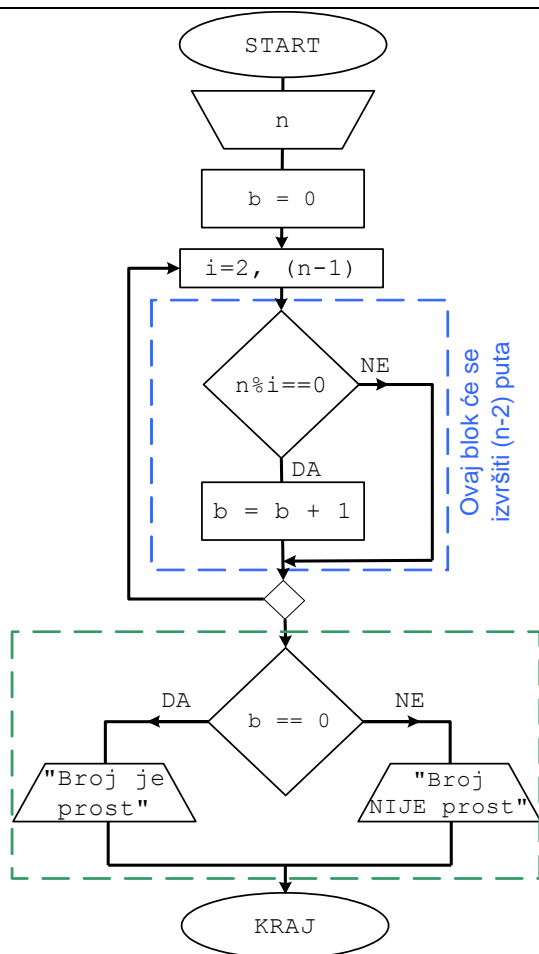
```

**drugi način:**

```

16:         if (n%i == 0)
17:         {
18:             cout << b+1 << ". Broj " << n << " je djeljiv sa " << i << endl;
19:             b = b + 1;
20:         }

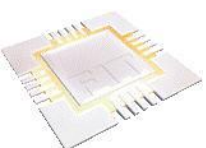
```

**Rješenje zadatka br. 99:**

```

4: void main()
5: {
6:     int n, b;
7:     b = 0;
8:
9:     cout << "Unesi broj n: ";
10:    cin >> n;
11:
12:    for (int i=2; i<=n-1; i++)
13:    {
14:        if (n%i == 0)
15:        {
16:            b = b + 1;
17:        }
18:    }
19:
20:    if (b==0)
21:        cout << "\nBroj je prost. \n";
22:    else
23:        cout << "\nBroj NIJE prost \n";
24: }

```



## Rješenje zadatka br. 100:

```
1:  #include <iostream>
2:  using namespace std;
3:
4:  void main()
5:  {
6:      int x, y, b;
7:
8:      cout << "Unesite brojeve x i y \n";
9:      cin >> x >> y;
10:     cout << "=====\n";
11:
12:     for (int n=x; n<=y; n++)
13:     {
14:         b = 0;
15:
16:         for (int i=2; i<=n-1; i++)
17:         {
18:             if (n%i == 0)
19:             {
20:                 b = b + 1;
21:             }
22:         }
23:
24:         if (b==0)
25:             cout << "Broj " << n << " je prost. \n";
26:         else
27:             cout << "Broj " << n << " NIJE prost \n";
28:     } // kraj petlje n
29: }
```

