

Zadaci – 8a, Petlja „FOR” (prvi dio)

Za *for*-petlju vrijede sljedeća pravila:

- *for*-petlja je petlja sa eksplicitnim brojačem
- *for*-petlju koristimo za ponavljanje jedne naredbe ili više naredbi određenim brojem puta
- u *for*-petlji koristimo jednu varijablu kao brojač ponavljanja naredbi (ime te varijable je obično 'i' ili 'j')
- brojač *for*-petlje ima svoju početnu i krajnju vrijednost
- brojač se obično nakon svakog izvršavanja naredbi povećava za 1
- naredbe unutar *for*-petlje smiju koristiti trenutnu vrijednost brojača (varijable i)
 - npr. `cout << i << endl;`
- razlika (između krajnje i početne vrijednosti brojača) uvećana za 1 predstavlja broj izvršavanja petlje, tj. broj ponavljanja naredbi unutar *for*-petlje:

Primjer 1:

- početna vrijednost brojača: 1
- krajnja vrijednost brojača: 5
- razlika: $5 - 1 = 4$
- broj ponavljanja: $4 + 1 = 5$

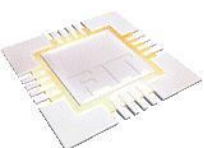
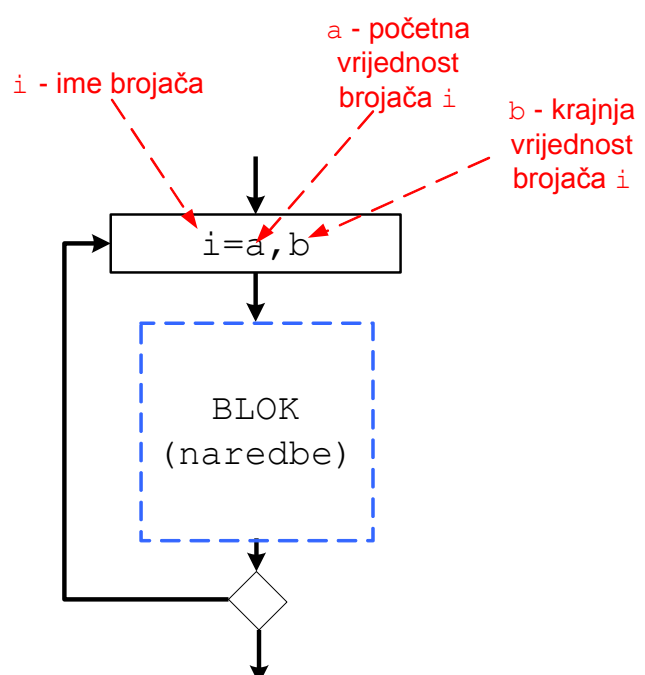
Primjer 2:

- početna vrijednost brojača: 3
- krajnja vrijednost brojača: 10
- razlika: $10 - 3 = 7$
- broj ponavljanja: $7 + 1 = 8$

Potrebno je obratiti pažnju na sljedeće napomene:

- ako želimo ponavljati više naredbi moramo koristiti dodatne vitičaste zagrade
- ako želimo ponavljati jednu naredbu možemo izostaviti vitičaste zagrade
- bilo bi fino da naredbu (ili blok naredbi) koja se nalazi u *for*-petlji uvučemo (poravnamo) pomoću tabulatora

Opšti prikaz dijagram toka *for*-petlje je dat na slici:



Zadatak 77:

Napravite dijagram toka za sljedeći zadatak (tj. ne pišite kôd programa):

Napravite program koji će ispisati sljedeće poruke na ekran:

```
Pocetak programa!  
Broj neuspjesnih pokusaja je 0  
Broj neuspjesnih pokusaja je 1  
Broj neuspjesnih pokusaja je 2  
Broj neuspjesnih pokusaja je 3  
Broj neuspjesnih pokusaja je 4  
Broj neuspjesnih pokusaja je 5  
Kraj programa!
```

Riješen dijagram toka se nalazi na kraju dokumenta.

Zadatak 78:

Napravite dijagram toka za sljedeći zadatak (tj. ne pišite kôd programa):

Napravite petlju koja će ispisati na ekran parne trocifrene brojeve.

Riješen dijagram toka se nalazi na kraju dokumenta.

Pomoć:

- trocifreni brojevi su brojevi od 100 do 999

Prvi način:

- vrijednost brojača i treba se mijenjati od 100 do 999
- u *for*-petlju treba dodati *if*-iskaz koji će se izvršavati 900 puta
- svaki put kad se *if*-iskaz bude izvršavao on mora provjeriti da li je broj i paran broj
- (broj je paran ako je djeljiv sa 2, tj. ako je ostatak dijeljena jednak 0)
- samo ako je i paran broj (ako je uslov zadovoljen) treba se na ekran ispisati vrijednost i
- ako je i nije paran broj (ako uslov nije zadovoljen) ne treba ispisivati na ekran nikakvu vrijednost (nećemo imati *else*-slučaja)

Drugi način:

- vrijednost brojača i treba se mijenjati od 50 do 499
- u *for*-petlji treba ispisivati vrijednost ($2*i$) na ekran

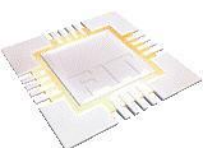
Zadatak 79:

Napravite dijagram toka za sljedeći zadatak (tj. ne pišite kôd programa):

Napravite petlju koja će ispisati na ekran kvadratnu vrijednost svih neparnih brojeva od 10 do 30 i vrijednost korijena svih parnih brojeva od 10 do 30, na sljedeći način:

```
korijen broja 10 je 3.333  
kvadrat broja 11 je 121  
korijen broja 12 je 3.464  
kvadrat broja 13 je 169  
korijen broja 14 je 3.741  
...
```

Riješen dijagram toka se nalazi na kraju dokumenta.



Zadatak 80:

Napravite dijagram toka za sljedeći zadatak:

Prepravite prethodni program tako da korisnik unosi početak i kraj niza brojeva umjesto fiksno određenog niza od 10 do 30:

Riješen dijagram toka se nalazi na kraju dokumenta.

Pomoć:

Korisnik unosi vrijednosti za dvije varijable (a i b). Treba voditi računa o tome da korisnik može unijeti negativne brojeve, tako da treba dodati još jedan *if-else*-iskaz za provjeru da li je brojač i trenutno negativan broj.

Unesi pocetak i kraj niza:

-4
20

```
korijen broja -4 je nemoguć
kvadrat broja -3 je 9
korijen broja -2 je nemoguć
kvadrat broja -1 je 1
korijen broja 0 je 0
kvadrat broja 1 je 1
korijen broja 2 je 1.414
kvadrat broja 3 je 9
korijen broja 4 je 2
...
```

Zadatak 81:

Napišite nekoliko linija kôda koristeći samo izraze u kojima ćete upotrebljavati matematički operator sabiranja (+) da bi ste izračunali vrijednost $y = 7 \cdot 6$:

- bez korištenja operatora za množenje (*) i bez korištenja petlje
- od brojeva smijete upotrebljavati samo broj 0 i broj 7
- u svakom izrazu smijete samo jednom koristiti matematički operator sabiranja (+)
- bit će potrebno deklarirati varijablu y i ispisati izračunatu vrijednost y na ekran

Riješen dijagram toka se nalazi na kraju dokumenta.

Zadatak 82:

Napravite dijagram toka za sljedeći zadatak:

Prepravite prethodni program tako da koristite *for*-petlju.

Riješen dijagram toka se nalazi na kraju dokumenta.

Zadatak 83:

Napravite dijagram toka za sljedeći zadatak:

Prepravite prethodni program tako da umjesto konstantnog broja 7 koristite neki broj b i umjesto konstantnog broja 6 da koristite broj a . Korisnik treba unijeti vrijednosti a i b .

Riješen dijagram toka se nalazi na kraju dokumenta.

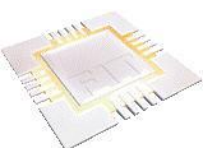
Zadatak 84:

Napravite dijagram toka za sljedeći zadatak:

Napravite program za računanje sume kvadrata brojeva od a do b . Korisnik treba unijeti vrijednosti a i b .

Riješen dijagram toka se nalazi na kraju dokumenta.

Pomoć: Ovdje nećemo računati sumu, kao u prethodnom zadatku gdje smo varijablu za sumu (y) uvećavali za vrijednost b (tj. $y=y+b$), nego ćemo varijablu za sumu (ime



varijable je ovdje `suma`) uvećavati za vrijednost kvadrata brojača `i*i` (tj. `suma=suma+i*i`), jer se `i` mijenja od `a` do `b`.

Zadatak 85:

Napravite dijagram toka za sljedeći zadatak:

Napravite program za računanje sume kvadrata brojeva koji su djeljivi sa 19 ili nisu djeljivi sa 17, od `a` do `b`. Korisnik treba unijeti vrijednosti `a` i `b`.

Riješen dijagram toka se nalazi na kraju dokumenta.

Zadatak 86:

Napravite dijagram toka za sljedeći zadatak:

Napravite program za računanje sume kvadrata parnih brojeva i sumu kubova neparnih brojeva od `a` do `b`. Korisnik treba unijeti vrijednosti `a` i `b`.

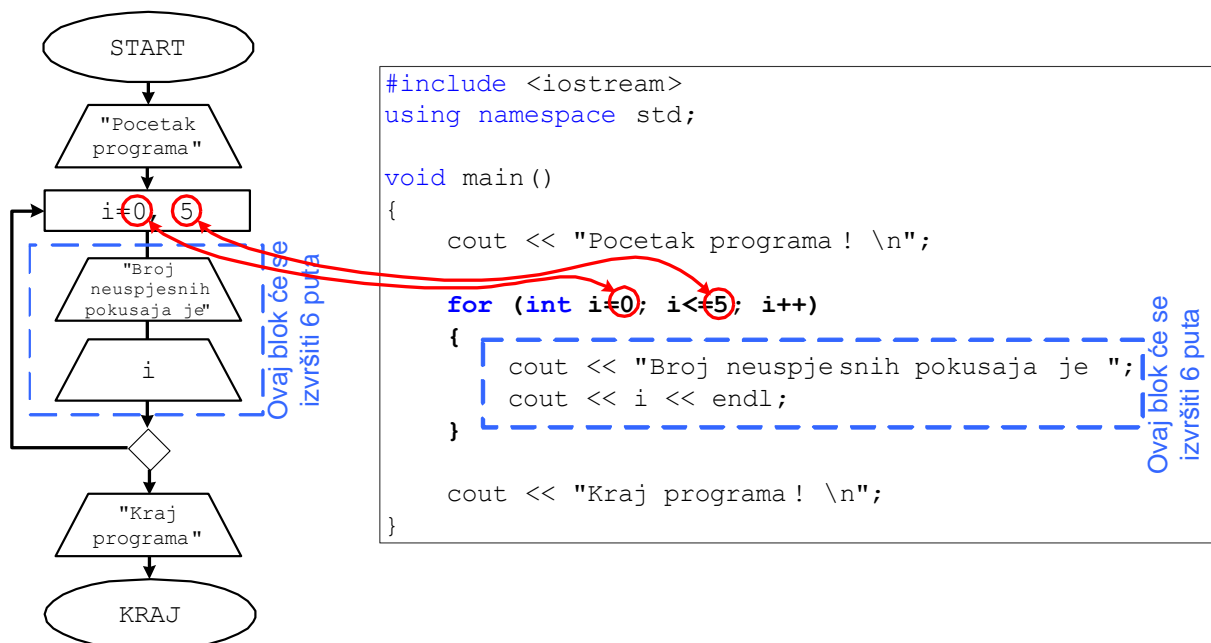
Riješen dijagram toka se nalazi na kraju dokumenta.

Pomoć: Ovdje ćemo koristiti dvije posebne varijable za sumu (`s1` i `s2`).

Iz dijagrama toka u C++ kôd

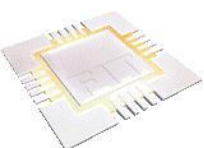
Kako pretvoriti dijagram toka u C++ kôd?

- To je veoma jednostavno, pogledajte:



Objašnjenje:

- `int i` predstavlja deklaraciju brojača `i` kojeg ćemo koristiti u `for`-petlji
- `i=0` predstavlja inicijalizaciju brojača (varijable) na početnu vrijednost `0`
- `i<=5` predstavlja krajnju vrijednost brojača `i`, a čita se: ponavljaj dok je `i<=5`



- `i++`, svaki put kad se izvrši tijelo petlje (naredbe koje se nalaze u petlji) vrijednost brojača se mijenja (povećava) pomoću naredbe za inkrement (`i++`)

Zadatak 87:

Pretvorite sve dijagrame toka u C++ kôd.

Riješen C++ kod se nalazi na kraju dokumenta.

Zadatak 88:

Napravite dijagram toka i napišite C++ kôd za sljedeći zadatak:

Napravite program za računanje aritmetičke sredine kvadrata brojeva od `a` do `b`. Korisnik treba unijeti vrijednosti `a` i `b`.

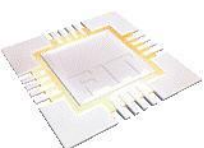
Riješen C++ kôd (ali bez dijagrama toka) se nalazi na kraju dokumenta.

Pomoć:

Aritmetičku sredinu kvadrata brojeva možete izračunati po formuli:

$$s_r = \frac{\text{suma kvadrata}}{n}$$

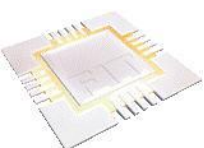
Razmislite: $n = ?$



Rješenja

Rješenje zadatka br. 77:	Rješenje zadatka br. 78 (87) prvi način:	Rješenje zadatka br. 78 drugi način:

Rješenje zadatka br. 79 :	Rješenje zadatka br. 80:

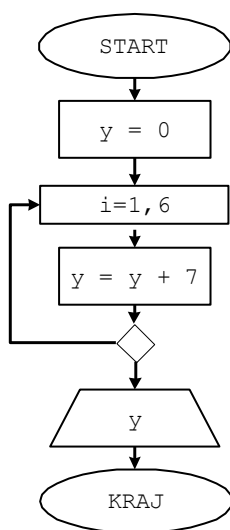


Rješenje zadatka br. 81:

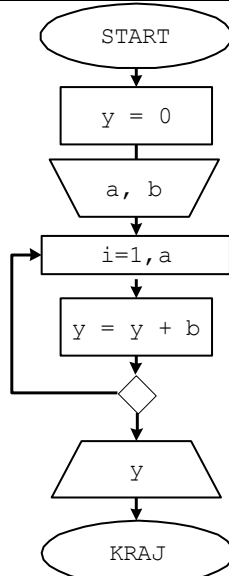
```

4:  void main()
5:  {
6:      int y;
7:      y = 0;
8:      // varijabla y mora imati pocetnu vrijednost 0
9:      y = y + 7; // ovo nije matematicka jednacina, ovo treba citati kao:
10:     y = y + 7; // novo y = staro y + 7
11:     y = y + 7;
12:     y = y + 7;
13:     y = y + 7;
14:     y = y + 7;
15:
16:     cout << "y = 7 * 6 = " << y << endl;
17: }
    
```

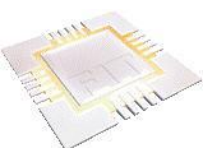
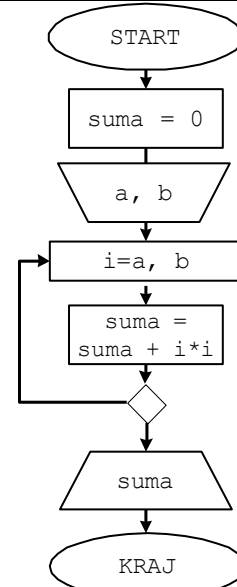
Rješenje zadatka br. 82:

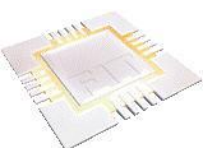
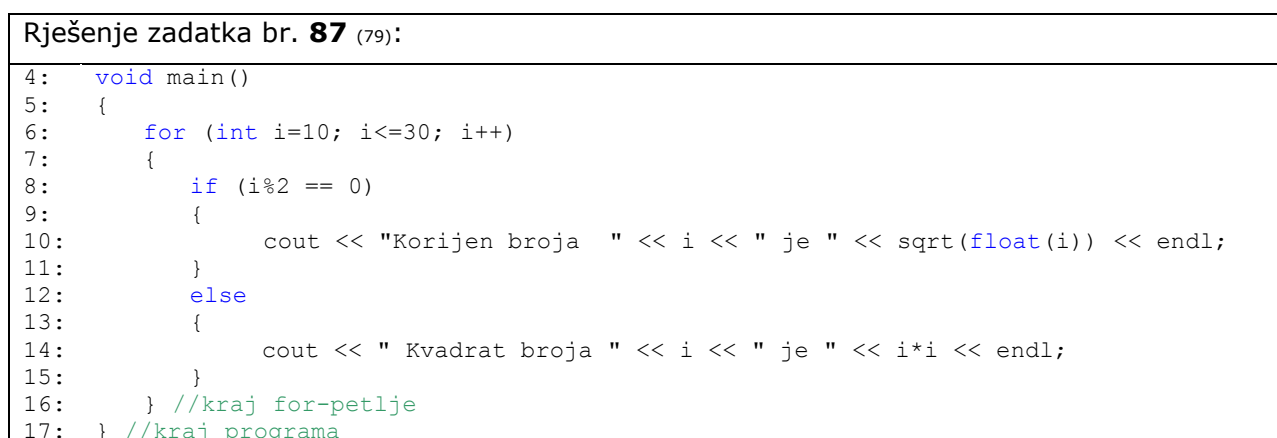
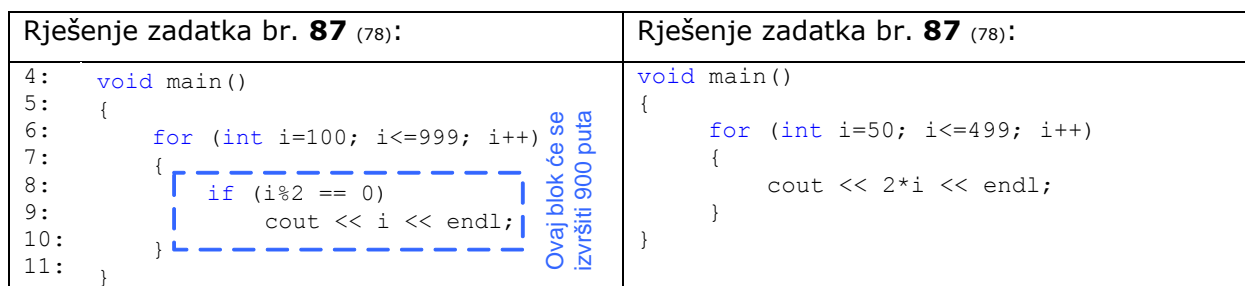
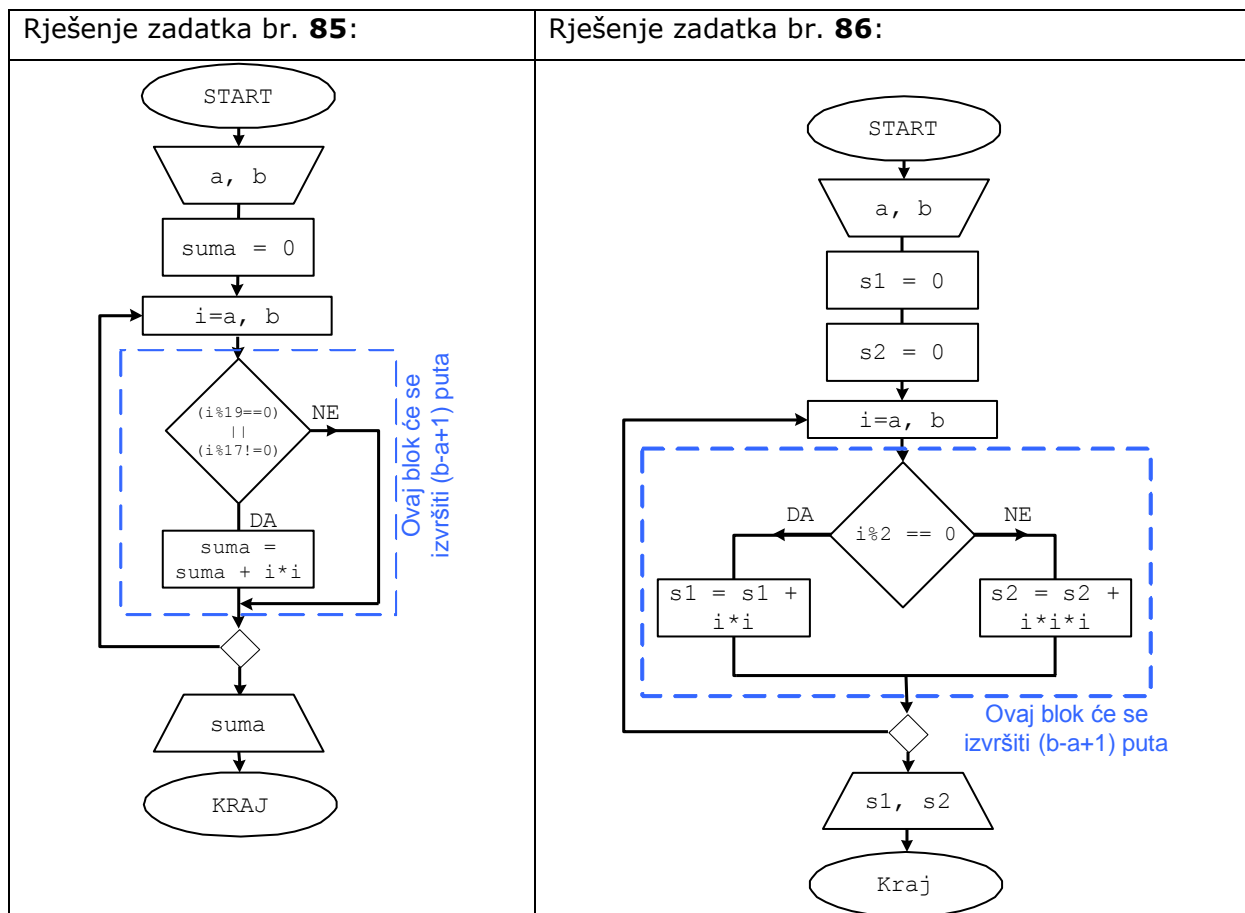


Rješenje zadatka br. 83:



Rješenje zadatka br. 84:





Rješenje zadatka br. 87 (80):

```

4: void main()
5: {
6:     int a, b;
7:     cout << "Unesi pocetak i kraj niza: \n";
8:     cin >> a >> b;
9:
10:    for (int i=a; i<=b; i++)
11:    {
12:        if (i%2 == 0)
13:        {
14:            if (i>=0)
15:                cout << "Korijen broja " << i << " je " << sqrt(float(i)) << endl;
16:            else
17:                cout << "Korijen broja " << i << " je nemoguc \n";
18:        }
19:        else
20:        {
21:            cout << " Kvadrat broja " << i << " je " << i*i << endl;
22:        }
23:    }
24: }

```

Rješenje zadatka br. 87 (81):

```

4: void main()
5: {
6:     int y;
7:     y = 0;
8:
9:     for (int i=1; i<=6; i++)
10:    {
11:        y = y + 7;
12:    }
13:
14:    cout << "y = " << y << endl;
15: }
16:
17:
18:

```

Rješenje zadatka br. 87 (82):

```

void main()
{
    int suma, a, b;
    suma = 0;

    cout << "Unesi pocetak i kraj niza: \n";
    cin >> a >> b;

    for (int i=a; i<=b; i++)
    {
        suma += i*i; //suma = suma + i*i;
    }

    cout << "suma = " << suma << endl;
}
/* za a=2 i b=5, suma iznosi
22 + 32 + 42 + 52 = 4 + 9 + 16 + 25 = 54 */

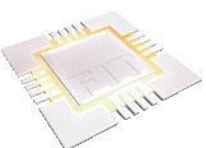
```

Rješenje zadatka br. 87 (83):

```

4: void main()
5: {
6:     int y, a, b;
7:     y = 0;
8:
9:     cout << "Unesi pocetak i kraj niza: \n";
10:    cin >> a >> b;
11:
12:    for (int i=1; i<=a; i++)
13:    {
14:        y = y + b; //ili y += b;
15:    }
16:
17:    cout << "y = " << a << " * " << b << " = " << y << endl;
18: }

```



Rješenje zadatka br. **87** ⁽⁸⁴⁾:

```

4: void main()
5: {
6:     int suma, a, b;
7:     suma = 0;
8:
9:     cout << "Unesi pocetak i kraj niza: \n";
10:    cin >> a >> b;
11:
12:    for (int i=a; i<=b; i++)
13:    {
14:        if ((i%19 == 0) || (i%17 != 0))
15:            suma = suma + i*i; //ili suma = suma + i*i;
16:    }
17:
18:    cout << "suma = " << suma << endl;
19: }

```

Rješenje zadatka br. **87** ⁽⁸⁵⁾:

```

4: void main()
5: {
6:     int s1, s2, a, b;
7:     s1 = 0;
8:     s2 = 0;
9:
10:    cout << "Unesi pocetak i kraj niza: \n";
11:    cin >> a >> b;
12:
13:    for (int i=a; i<=b; i++)
14:    {
15:        if (i%2 == 0)
16:            s1 += i*i; // s1 = s1 + i*i;
17:        else
18:            s2 += i*i*i; // s1 = s1 + i*i*i;
19:    }
20:
21:    cout << "suma kvadrata parnih brojeva je " << s1 << endl;
22:    cout << "suma kubova neparnih brojeva je " << s2 << endl;
23: }

```

za a=5 i za b=9 vrijedi: $s_1 = 6^2 + 8^2 = 36 + 64 = 100$, $s_2 = 5^3 + 7^3 + 9^3 = \dots = 1197$

Rješenje zadatka br. **87** ⁽⁸⁶⁾:

```

4: void main()
5: {
6:     int suma, a, b, n;
7:     suma = 0;
8:
9:     cout << "Unesi pocetak i kraj niza: \n";
10:    cin >> a >> b;
11:
12:    for (int i=a; i<=b; i++)
13:    {
14:        suma += i*i;
15:    }
16:
17:    n = b - a + 1;
18:    float sr = suma / float(n); // zasto float(n)? pogledajte W4, zadatak 27
19:
20:    cout << "prosjecna vrijednost kvadrata je " << sr << endl;
21: }

```

za a=2 i za b=3 vrijedi: $\text{suma} = 2^2 + 3^2 = 4 + 9 = 13$, $\text{pr} = \text{suma} / n = 13 / 2 = 6,5$

