

## Zadaci – 12. dio, Dodatak

### Zadatak 126:

Napravite program koji će provjeravati da li je uneseni broj  $n$  prost broj. Koristite:

- a) *while*-petlju
- b) *do-while*-petlju

Za pomoć pogledajte ponovo zadatak 99.

Rješenje pomoću *do-while*-petlje se nalaze na stranici 160.

### Zadatak 127:

Optimizirajte prethodni program (*nije bitno u kojoj vrsti petlje je riječ*):

- a) Kao što vidite, zadatak smo riješili tako što smo brojali sa koliko je brojeva djeljiv broj  $n$ . Međutim, program ne nudi optimalno rješenje. Moguće je smanjiti korištenje resursa za dva puta, tako što ćemo provjeravati djeljivost broja  $n$  samo sa brojevima iz intervala od **2** do  $n/2$ . Primjer:

Za  $n=36597$ , provjeravat ćemo djeljivost broja  $n$  samo u intervalu od 2 do 18298, jer broj 36597 nije sigurno djeljiv ni sa jednim brojem iz intervala 18299 do 36596.

- b) Dodatno optimiziranje programa:

Ako smo u toku provjeravanja (izvršavanja petlje) našli da je broj  $n$  djeljiv sa jednim brojem (brojem većim od 1) možemo odmah prekinuti petlju - jer broj  $n$  sigurno nije prost.

Poslije petlje slijedi *if-else*-iskaz (dvostruki izbor) koji treba, na osnovu nekog uslova, ispisati "...je prost" ili "...nije prost".

Pošto, *if-else*-iskaz 'ne zna' da li se petlja prekinula, moramo prethodno u *if*-iskazu ( $n\%i==0$ ) promijeniti vrijednost neke varijable tako da *if-else*-iskaz koji ispisuje poruku može 'znati' da li se *if*-iskaz ( $n\%i==0$ ) bio istinit ili neistinit izraz.

Promjena neke vrijednosti u *if*-iskazu ( $n\%i==0$ ) može da bude: `brojac=1` ili `prost=false` ili sl.

(*Napomena*: varijable `brojac` ili `prost` moraju da imaju neku početnu vrijednost prije izvršavanja petlje. Ta početna vrijednost mora biti različita od vrijednosti `1` za varijablu `brojac` i različita od vrijednosti `false` za varijablu `prost`.)

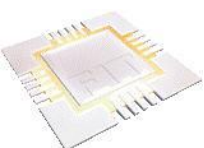
Prekidanje petlje možete vršiti na dva načina:

1. Pomoću dodatnog uslova za prekid petlje.

Ako koristite varijablu `brojac`, uslov bi glasio:

- `i<=n/2 && brojac!=1` (čita se: ponavlja ako je ...)  
dovoljno je da jedan uslov **ne** bude zadovoljen i petlja se **prekida**
- `!(i>n/2 || brojac==1)` (čita se: ponavlja ako je ...)  
dovoljno je da jedan uslov **bude** zadovoljen i petlja se **prekida**

(Ako su vam ovi logički uslovi nejasni pročitajte ponovo *Workshop – 6. dio - logički operatori!*)



2. Pomoću naredbe `break`: Kada se prvi put u petlji izvrši naredba `break`, petlja se prekida i nastavlja se sa izvršavanjem naredne linije koda koja slijedi nakon petlje.

Rješenje pomoću while-petlje i varijable prost sa oba načina prekida petlje se nalazi na str. 160.

### Zadatak 128:

Napraviti program koji će ispisati sve trocifrene brojeve koji su djeljivi sa sumom svojih cifara, te utvrditi koliko ukupno ima takvih brojeva.

Program riješiti u skladu sa fazama programiranja!

Rješenje se nalaze na stranici 162.

Pomoć:

- postaviti ćemo vrijednost za varijablu `brojac` na 0
- ispitati ćemo sve brojeve od 100 do 999, koristeći brojač petlje `i`
- izračunat ćemo stotice tako što ćemo podijeliti trenutni broj koji ispitujemo (broj `i`) sa 100 i odbaciti decimalnu vrijednost  
(npr.  $467:100 = 4.67$ ,  $\text{int}(4.67) = 4$ )
- izračunat ćemo desetice tako što ćemo, pomoću operatora za ostatak dijeljenja `%`, modularno dijeliti trenutni broj (`i`) sa 100 i dobit ćemo zadnje dvije cifre, a taj rezultat ćemo cijelo-brojno podijeliti sa 10, kao što smo to činili u prethodnom koraku  
(npr.  $467\%100 = 67$ ,  $67/10 = 6.7$ ,  $\text{int}(6.7) = 6$ )
- izračunat ćemo jedinice tako što ćemo trenutni broj (`i`) modularno dijeliti sa 10  
(npr.  $467\%10 = 7$ )
- izračunat ćemo sumu cifara trenutnog broja (`i`)
- ako je trenutni broj (`i`) djeljiv sa sumom svojih cifara ispisat ćemo ga na ekran i povećat ćemo `brojac` za 1
- poslije petlje ispisat ćemo vrijednost varijable `brojac`

Program treba ispisati ukupno 180 trocifrenih brojeva koji su djeljivi sa sumom svojih cifara.

### Zadatak 129:

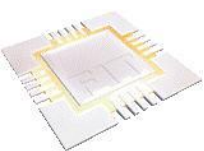
Riješite prethodni program pomoću tri ugniježdene *for*-petlje! Nacrtajte dijagram toka i na osnovu njega napišite C++ kôd.

Rješenje se nalaze na stranici 163.

### Zadatak 130:

Napravite program u kome ćete inicijalizirati dvije varijable (`a`, `b`) tipa integer, a zatim te vrijednosti varijabli zamijeniti i ispisati na ekran prvo u hexadecimalnom, pa u oktalnom, a zatim u decimalnom brojnom sistemu.

Rješenje se nalaze na stranici 163.



### Zadatak 131:

Napravite program u kome će te deklarirati tri varijable (*a*, *b* i *c*) tipa *integer*.

- varijabli *a* dodijelite neku vrijednost hexadecimalnom obliku
- varijabli *b* dodijelite neku vrijednost oktalanom obliku
- varijabli *c* dodijelite neku vrijednost decimalnom obliku
- ispišite vrijednost varijabli *a*, *b* i *c* u decimalnom obliku

Rješenje se nalaze na stranici 163.

### Zadatak 132:

(zadatak sa ispita: UPR - 29.01.2005.)

Izraditi program u programskom jeziku C++ koji izračunava izraz:

$$z = \frac{ZbirKubova(m,n)}{x^2}$$

Povratna vrijednost funkcije *ZbirKubova* i parametri *m* i *n* su tipa *int*. Zadatak riješiti u skladu sa fazama programiranja.

Omogućiti ponavljanje unosa ulaznih podataka (čitavog programa) ako korisnik to želi.

Rješenje se nalaze na stranici 164.

### Zadatak 133:

U zadatku 149 je korištenja funkcija *ZbirKubova* sa povratnom vrijednošću i sa pozivom *pass-by-value*. Prepravite funkciju u *ZbirKubova2*, tako da funkcija nema povratnu vrijednost i da za izlaznu vrijednost iz funkcije koristite referencu (*pass-by-reference*).

Objasnite razliku između ovog programa i program iz zadatka 149.

Riješen kôd se nalazi na stranici 166.

Pomoć:

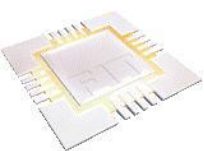
- u oba zadatka (149 i 150) ispis na ekran treba da se vrši u glavnom programu
- izlaz iz funkcije *ZbirKubova* (zadatak 149) je proslijeđen pomoću naredbe *return*
- izlaz iz funkcije *ZbirKubova2* treba da se smjesti u referentnu varijablu
- u funkciji *ZbirKubova2* nećemo koristiti naredbu *return*, s toga će ona biti tipa *void*
- u funkciji *ZbirKubova2* koristit ćemo tri parametra: treći parametar će biti referentna varijabla (izlazni parametar)

### Zadatak 134:

(zadatak sa ispita: UPR - 16.04.2005.)

Napišite program sa izbornikom (menî) za izračunavanje površine geometrijskih likova: kvadrat, pravougaonik i krug. Za izračunavanje površine koristiti funkcije. Menî treba da sadrži i izbor za kraj izvršenja programa.

Riješen kôd se nalazi na stranici 167.



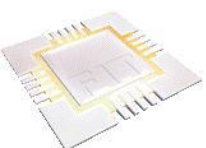
## Rješenje:

### Rješenje zadatka br. 126 (pomoću while-petlje):

```
4: void main()
5: {
6:     int n, brojac;
7:
8:     cout << "Unesi broj: ";
9:     cin >> n;
10:
11:     brojac = 0;
12:
13:     int i=2;
14:     while (i<=n/2)
15:     {
16:         if (n%i == 0)
17:             brojac++;
18:         i++;
19:     }
20:
21:     if (brojac == 0)
22:         cout << "Broj je prost\n";
23:     else
24:         cout << "Broj nije prost\n";
25: }
```

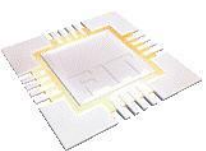
### Rješenje zadatka br. 127 a (optimiziran program uz pomoć dodatnog uslova za prekid):

```
4: void main()
5: {
6:     int n;
7:     bool prost;
8:
9:     cout << "Unesi broj: ";
10:    cin >> n;
11:
12:    prost = true; // broj je prost (nevin) dok mu se ne dokaže da je složen (kriv)
13:
14:    int i=2;
15:    while (i<=n/2 && prost==true)
16:    {
17:        if (n%i == 0)
18:            prost=false; // dokazano da je broj slozen - dokazana je krivica
19:        i++;
20:    }
21:
22:    if (prost) // if (prost ==false)
23:        cout << "Broj je prost\n";
24:    else
25:        cout << "Broj nije prost\n";
26: }
```



**Rješenje zadatka br. 127 b** (optimiziran program uz pomoć naredbe *break*):

```
4: void main()
5: {
6:     int n;
7:     bool prost;
8:
9:     cout << "Unesi broj: ";
10:    cin >> n;
11:
12:    prost = true;
13:
14:    int i=2;
15:    while (i<=n/2)
16:    {
17:        if (n%i == 0)
18:        {
19:            prost=false; // ova se naredba mora prije prekida izvršiti
20:            break;       // prekid - izlaz iz petlje
21:        }
22:        i++;
23:    }
24:
25:    if (prost) //ili if (prost ==false)
26:        cout << "Broj je prost\n";
27:    else
28:        cout << "Broj nije prost\n";
29: }
```



Rješenje zadatka br. **128** (u skladu sa fazama programiranja):

### 1. Analiza i specifikacija:

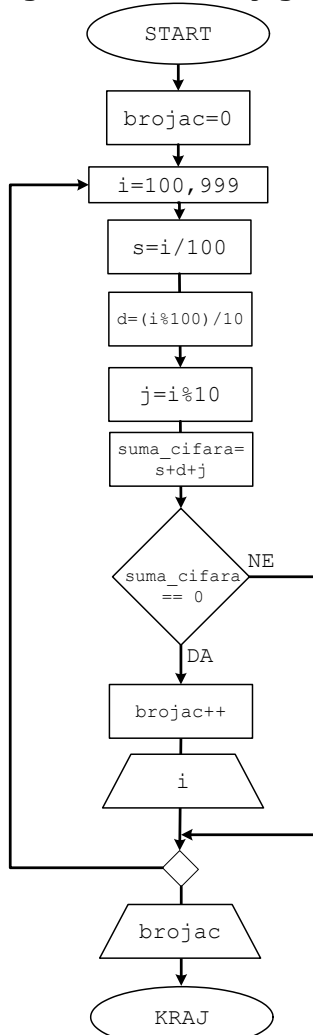
Input	Output
-	svi trocifreni brojevi djeljivi sa sumom svojih cifara
	brojač ispisanih brojeva

Ograničenja: *nema*

### 2. Dizajn

Podatkovni objekt	Vrsta vrijednosti	Tip objekta	Ime objekta
brojač ispisanih brojeva	varijabla	int	brojac
stotice	varijabla	int	s
desetice	varijabla	int	d
jedinice	varijabla	int	j
suma cifara	varijabla	int	suma_cifara

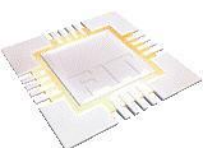
### 3. Algoritam – blok dijagram



### 4. Kôd (code)

```

1:  #include <iostream>
2:  using namespace std;
3:
4:  void main()
5:  {
6:      int brojac, s, d, j, suma_cifara;
7:      brojac = 0;
8:
9:      for (int i=100; i<=999; i++)
10:     {
11:         s = i / 100; // int/int = int
12:         d = (i % 100) / 10;
13:         j = (i % 10);
14:
15:         suma_cifara = s + d + j;
16:
17:         if (i%suma_cifara == 0)
18:         {
19:             brojac++;
20:             cout << i << " ";
21:         }
22:     }
23:     cout << "\nUkupno: " << brojac << endl;
24: }
    
```



## Rješenje zadatka br. 129:

```

1: void main()
2: {
3:     int brojac;
4:     brojac = 0;
5:
6:     for (int s=1; s<=9; s++)
7:     {
8:         for (int d=1; d<=9; d++)
9:         {
10:            for (int j=1; j<=9; j++)
11:            {
12:                if ((s*100+d*10+j)%(s+d+j)==0)
13:                {
14:                    brojac++;
15:                    cout <<s<<d << j << " ";
16:                }
17:            }
18:        }
19:    }
20:    cout << "\nUkupno: " << brojac << endl;
21: }

```

Nakon što riješite zadatak, pročitajte sljedeću napomenu:

Ovaj program ima jednu sitnu grešku. Pokušajte je otkriti. Usporedite ispis ovog programa s ispisom prethodnog programa. Ovaj program će ispisati ukupno 108 brojeva, dok će prethodni program ispisati ispravan rezultat, tj. 180.

Rješenje ove 'zagonetke' se nalazi na stranici br. 167.

## Rješenje zadatka br. 130:

```

1: void main()
2: {
3:     int a = 23;
4:     int b = 24;
5:     int temp;
6:
7:     temp = a;
8:     a = b;
9:     b = temp;
10:
11:     cout << "a: " << hex << a << ", " << oct << a << ", " << dec << a << endl;
12:     cout << "b: " << hex << b << ", " << oct << b << ", " << dec << b << endl;
13: }
14: }

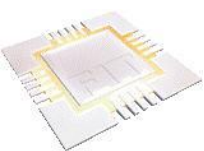
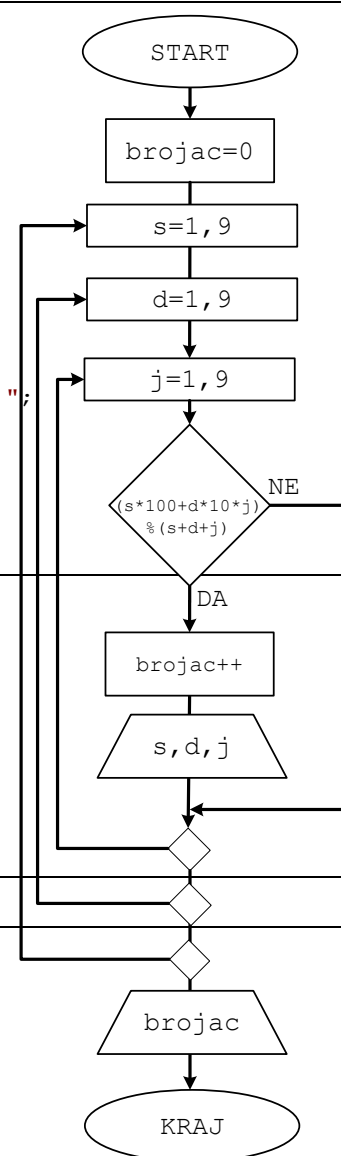
```

## Rješenje zadatka br. 131:

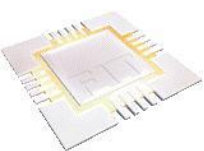
```

1: void main()
2: {
3:     int a,b,c;
4:
5:     a = 0x123;           //hexadecimalno '123'
6:     b = 0123;           //oktalno '123'
7:     c = 123;            //decimalno '123'
8:
9:     cout << "a: " << a << endl;
10:    cout << "b: " << b << endl;
11:    cout << "c: " << c << endl;
12: }

```



Rješenje zadatka br. **132:** *(u skladu sa fazama programiranja):*





### 1. Analiza i specifikacija:

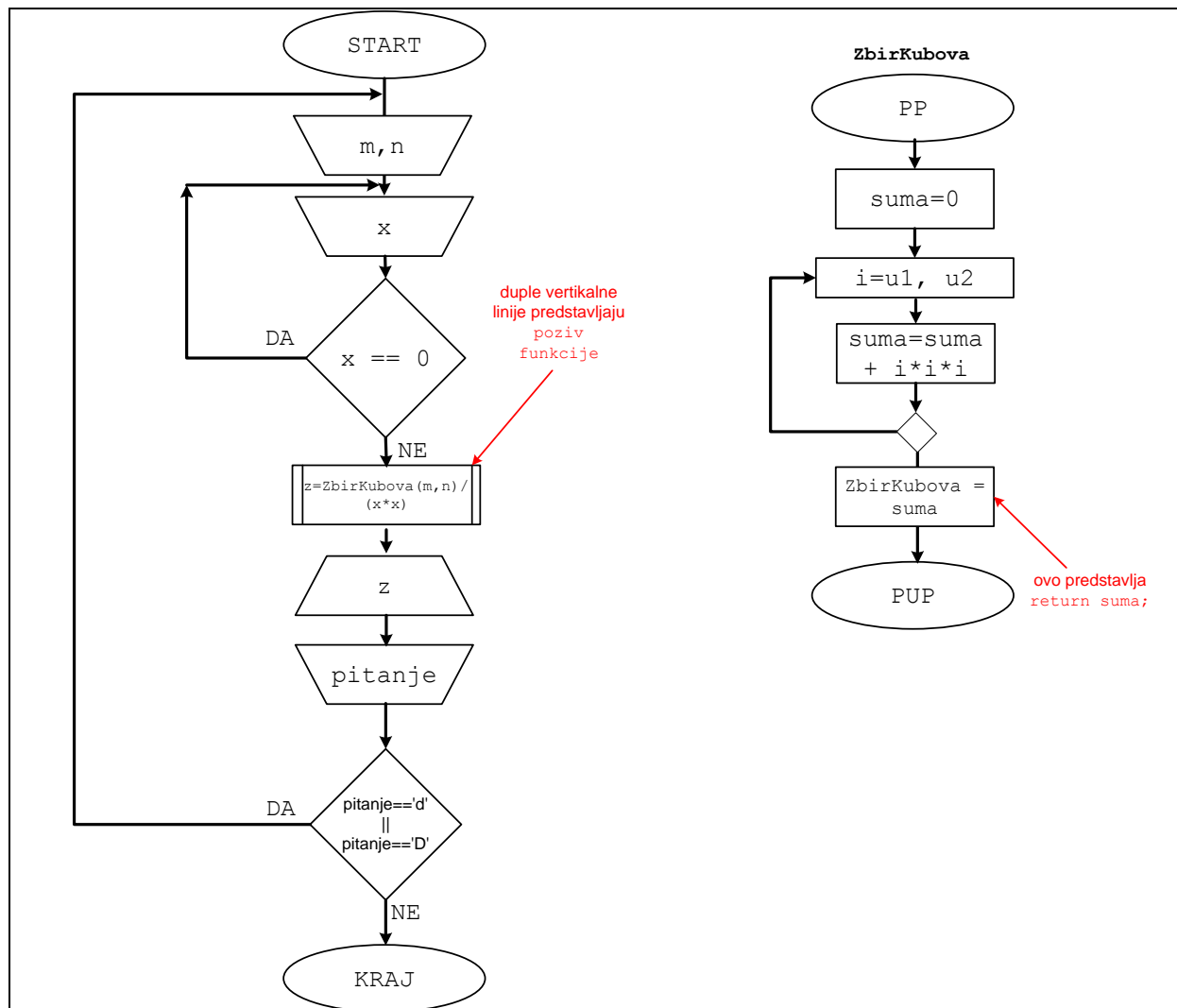
Input	Output
Broj za početak - parametar (m)	Razlomak (z)
Broj za kraj parametar (n)	
Nazivnik (x)	

Ograničenja: nazivnik **x mora biti različit od nule**

### 2. Dizajn

Podatkovni objekt	Vrsta vrijednosti	Tip objekta	Ime objekta
Broj za parametar	varijabla	int	m
Broj za parametar	varijabla	int	n
Nazivnik	varijabla	int	x
Razlomak	varijabla	<b>float</b>	z
Suma kubova	varijabla	int	suma

### 3. Algoritam – blok dijagram



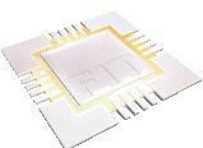
### 4. Kôd (code)

```

1: #include <iostream>
2: using namespace std;
3:
4: int ZbirKubova(int, int); // deklaracija (prototip) funkcije
5:
6: void main()
7: {
8:     int m, n, x;

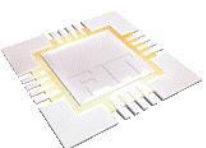
```

*Programiranje 1 :: Vježbe*  
<http://student.fit.ba/>



## Rješenje zadatka br. 133:

```
1: #include <iostream>
2: using namespace std;
3:
4: void ZbirKubova(int, int, int& );
5:
6: void main()
7: {
8:     int m, n, x, zbir;
9:     float z;
10:    char pitanje;
11:
12:    do
13:    {
14:        cout << "Unesite m, m \n";
15:        cin >> m >> n;
16:
17:        do
18:        {
19:            cout << "Unesite x (x mora biti razlicito od 0) \n";
20:            cin >> x;
21:
22:        }while(x==0);
23:
24:        ZbirKubova(m, n, zbir);    //funkcija je tipa void
25:        z = zbir / float(x*x);
26:        cout << "z = " << z << endl;
27:
28:        cout << "Ponoviti? (D)a...\n";
29:        cin >> pitanje;
30:
31:    }while (pitanje=='d' || pitanje=='D');
32: } // kraj funkcije main - kraj programa
33:
34: void ZbirKubova(int u1, int u2, int& Izlaz)
35: {
36:     int suma;
37:     suma = 0;
38:
39:     for (int i=u1; i<=u2; i++)
40:     {
41:         suma = suma + i*i*i;
42:     }
43:
44:     Izlaz = suma; //izlaz iz funkcije je suma kubova
45: }
```



## Rješenje zadatka br. 134:

```
1:  #include <iostream>
2:  using namespace std;
3:
4:  const float Pi = 3.141;    // konstantna globalna varijabla
5:
6:  void f1(float a)
7:  {
8:      cout << " Povrsina kvadrata = " << a*a << endl;
9:  }
10:
11: void f2(float a, float b)
12: {
13:     cout << " Povrsina pravougaonika = " << a*b << endl;
14: }
15:
16: void f3(float r)
17: {
18:     cout << " Povrsina kruga = " << r*r*Pi << endl;
19: }
20:
21: void main()
22: {
23:     int izbor;
24:     float a, b;
25:
26:     do
27:     {
28:         cout << "\n\n=====\n";
29:         cout << "IZBORNIK \n";
30:         cout << "1: kvadrat \n";
31:         cout << "2: pravougaonik \n";
32:         cout << "3: krug \n";
33:         cout << "4: izlaz \n>>>";
34:         cin >> izbor;
35:
36:         switch(izbor)
37:         {
38:             case 1:
39:                 cout << "Unesite a: ";
40:                 cin >> a;
41:                 f1(a);
42:                 break;
43:             case 2:
44:                 cout << "Unesite a i b \n";
45:                 cin >> a >> b;
46:                 f2(a, b);
47:                 break;
48:             case 3:
49:                 cout << "Unesite r: ";
50:                 cin >> a;
51:                 f3(a);
52:                 break;
53:             case 4:
54:                 break;
55:             default:
56:                 cout << "Pogresan izbor \n";
57:                 }; // kraj switch-iskaza
58:         }while(izbor != 4);
59:     }
```

