

## Zadaci – 7. dio, *do-while* petlja

### Zadatak 71:

Napravite program koji će ispisati poruke kao na slici koristeći *do-while*-petlju.

- a) brojač (varijabla) *i* bi se trebao mijenjati od 0 do 9
- b) brojač (varijabla) *i* bi se trebao mijenjati od 10 do 1

Riješen C++ kôd se nalaze na stranici 107.

```
Pocetak programa!
1. Brojim unazad: 10
2. Brojim unazad: 9
3. Brojim unazad: 8
4. Brojim unazad: 7
5. Brojim unazad: 6
6. Brojim unazad: 5
7. Brojim unazad: 4
8. Brojim unazad: 3
9. Brojim unazad: 2
10. Brojim unazad: 1
```

### Zadatak 72:

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: 17

Unesi broj n: 18

Rješenje se nalazi 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 da je broj djeljiv ili nije djeljiv sa tim i tim i tim... S toga koristiti ćemo drugi definiciju koja ne sadrži u sebi izraz SAMO:

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

Ista definicija ali bez riječi 'osim':

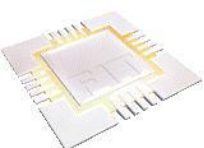
Broj **n** je prost ako NIJE djeljiv ni sa jednim brojem u intervalu od **2** do (**n-1**).

Kako riješiti zadatak?

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

Koristit ćemo 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 doraditi program, tako da umjesto da ispisivanja svih



brojeva iz datog intervala (i odgovarajuće poruke "djeljiv" ili "nije djeljiv"), ispisuje jedna poruka poruka: '...je prost' ili '...NIJE prost'. Iz prethodnog program trebat ćemo još izbaciti nepotrebnu *cout*-naredbu za ispis poruke "Broj ... je djeljiv sa ...".

### Zadatak 73:

Optimizirajte prethodni program:

- a) Kao što vidite, zadatak smo riješili tako što smo brojali sa koliko brojeva je 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 (morali smo) u *if*-iskazu ( $n\%i==0$ ) promijeniti neku vrijednost neke varijable tako da *if-else*-iskaz koji ispisuje poruku može 'znati' da li se *if*-iskaz ( $n\%i==0$ ) izvršio.

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. 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 ...)
 

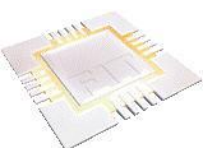
dovoljno je da jedan uslov ne bude zadovoljen i petlja se prekida
- `!(i>n/2 || brojac==1)` (čita se: ponavlja ako je ...)
 

dovoljno je da jedan uslov bude zadovoljen i petlja se prekida

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 se nalazi na kraju dokumenta.

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



#### Zadatak 74:

Napraviti program koji će ispisati sva cjelobrojna rješenja jednačine:  
 $3x - 10y = 5$  za  $y = 1$  do  $60$

Rješenje se nalazi na kraju dokumenta.

Pomoć:

- trebate varijablu  $y$  mijenjati pomoću petlje od 1 do 60
- izračunajte vrijednost  $x$ :
- $3x - 10y = 5 \Rightarrow x = \frac{5 + 10y}{3}$
- ako je  $x$  cijeli broj, ispišite  $x$  i  $y$  (ako vrijednost u brojniku djeljiva sa nazivnikom)

#### Zadatak 75:

(zadatak sa ispita: UPR - 14.02.2005.)

Napišite program koji izračunava zbir parnih brojeva od 0 do 1000 koji su djeljivi sa 7 i 3 ili koji su djeljivi sa 8. Program treba, osim rezultata, da ispisuje i brojeve koji zadovoljavaju uslov. Ispis širine 5 mjesta.

Rješenje se nalazi na kraju dokumenta.

Pomoć:

- a)  $i \% 3 == 0$  (djeljiv sa 3) }  
 b)  $i \% 7 == 0$  (djeljiv sa 7) }  $\&\& \Rightarrow (i \% 3 == 0) \&\& (i \% 7 == 0)$  }  
 c)  $i \% 8 == 0$  (djeljiv sa 8) }  $|| \Rightarrow ((i \% 3 == 0) \&\& (i \% 7 == 0)) || (i \% 8 == 0)$   
 d)  $i \% 2 == 0$  (parni brojevi)

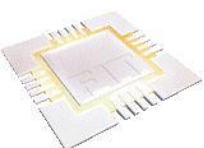
#### Zadatak 76:

Napraviti program za skraćivanje razlomka. Program riješiti u skladu sa fazama programiranja!

Rješenje se nalazi na kraju dokumenta.

Pomoć:

- input: brojnik i nazivnik
- provjerit ćemo koji ima manju vrijednost brojnik ili nazivnik: manju vrijednost ćemo pridružiti varijabli manji (zbog optimizacije programa)
- moramo provjeriti djeljivost varijable brojnik i nazivnik sa brojevima iz opsega [manji, 2] (ako ne želimo optimizirati programa, onda je svejedno da li ćemo koristiti opseg [brojnik, 2] ili opseg [nazivnik, 2])
- ako su brojnik i nazivnik djeljivi sa trenutnom vrijednošću brojača petlje (i), podijeliti ćemo ih sa brojem i
- output (skraćeni razlomak): varijabla brojnik i varijabla nazivnik



## Rješenja

### Rješenje zadatka br. 71 - a:

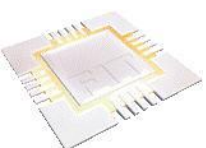
```
4:  #include <iostream>
5:  using namespace std;
6:
7:  void main()
8:  {
9:      cout << "Pocetak programa! \n";
10:     int i=0;
11:
12:     do
13:     {
14:         cout << i+1 <<" . Brojim unazad: " << 10-i << endl;
15:         i++;
16:     }while (i<=9);
17:
18:     cout << "Kraj programa! \n";
19: }
```

### Rješenje zadatka br. 71 - b:

```
4:  void main()
5:  {
6:      cout << "Pocetak programa! \n";
7:      int i=10;
8:
9:      do
10:     {
11:         cout << 11-i <<" . Brojim unazad: " << i << endl;
12:         i--;
13:     }while (i>=1);
14:
15:     cout << "Kraj programa! \n";
16: }
```

### Rješenje zadatka br. 72 (pomoću do-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:
15:      do
16:      {
17:          if (n%i == 0)
18:              brojac++;
19:          i++;
20:      }while (i<=n/2);
21:
22:      if (brojac == 0)
23:          cout << "Broj je prost\n";
24:      else
25:          cout << "Broj nije prost\n";
26:  }
27:
```

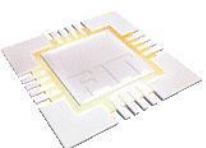


**Rješenje zadatka br. 73 (1)** (optimiziran program uz pomoć dodatnog uslova za prekid):

```
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:    do
16:    {
17:        if (n%i == 0)
18:            prost=false; // dokazano da je broj slozen – dokazana je krivica
19:        i++;
20:    }while (i<=n/2 && prost==true);
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. 73 (2)** (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:
16:    do
17:    {
18:        if (n%i == 0)
19:        {
20:            prost=false; // ova se naredba mora prije prekida izvršiti
21:            break; // prekid - izlaz iz petlje
22:        }
23:        i++;
24:    }while (i<=n/2);
25:
26:    if (prost) // if (prost ==false)
27:        cout << "Broj je prost\n";
28:    else
29:        cout << "Broj nije prost\n";
30: }
```

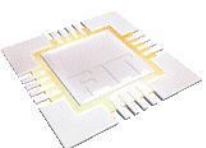


## Rješenje zadatka br. 74:

```
1: void main()
2: {
3:     int brojnik;
4:
5:     int y=1;
6:     do
7:     {
8:         brojnik = 5 + 10*y;
9:         if (brojnik%3 == 0)
10:        {
11:            int x = brojnik/3;
12:            cout << "x = " << x;
13:            cout << " za y = " << y << endl;
14:        }
15:        y++;
    }while(y<=60);
```

## Rješenje zadatka br. 75:

```
1: #include <iostream>
2: #include <iomanip> // biblioteka za funkciju setw, koja omogućava ispis širine 5
3: mjesta
4: using namespace std;
5:
6: void main()
7: {
8:     int zbir;
9:     zbir = 0;
10:    int i=0;
11:    do
12:    {
13:        if ( (i%2==0) && ((i%3==0) && (i%7==0)) || (i%8 == 0) )
14:        {
15:            cout << i << setw(5);
16:            // koja je razlika između setw(5) i pet space-razmaka?
17:
18:            zbir = zbir + i;
19:        }
20:        i++;
21:    }while(i<=1000);
22:    cout << "\nZbir: " << zbir << endl;
23: }
```



## Rješenje zadatka br. 76:

```
4: void main()
5: {
6:     int brojnik, nazivnik, manji;
7:
8:     cout << "Unesi brojnik i nazivnik \n";
9:     cin >> brojnik >> nazivnik;
10:
11:     if (nazivnik == 0)
12:         cout << "Greska...\n";
13:     else
14:     {
15:         if (brojnik < nazivnik)
16:             manji = brojnik;
17:         else
18:             manji = nazivnik;
19:
20:         int i=manji;
21:         do
22:         {
23:             if ((brojnik%i==0) && (nazivnik%i==0))
24:             {
25:                 brojnik = brojnik / i;
26:                 nazivnik = nazivnik / i;
27:             }
28:             i--;
29:         }while(i>=2);
30:
31:         cout << brojnik << " / " << nazivnik << endl;
32:     }
33: }
```

