

## Zadaci – 6b (Logički operatori)

### Zadatak 45:

Napravite algoritam i predstavite ga dijagramom toka za sljedeći zadatak, a zatim napišite C++ kod koristeći dijagram toka:

Napravite program u kome ćete od korisnika zahtijevati da unese broj indeksa nekog studenta FIT-a. Program treba ispisati da li je to validan broj indeksa (neka to budu brojevi između 1 i 1600, uključujući i njih).

Program riješite pomoću dva *if-else*-iskaza.

Program treba ispisati poruku "Kraj programa" pomoću naredbe `cout` koja se ne nalazi u *if-else*-iskazima.


Rješenje se nalazi na stranici 57.

Napomena:

**if-else**-iskaz predstavlja dvostruki izbor, dok **if**-iskaz predstavlja jednostruki izbor

Nakon što riješite neki zadatak pogledajte ponuđeno rješenje i usporedite!

### Zadatak 46:

Operator	Simbol	Primjer u <i>if</i> -iskazu
I (AND)	&&	<pre>if ((a==5) &amp;&amp; (b==9))     cout &lt;&lt; "Oba broja su tacna";</pre>
ILI (OR)		<pre>if ((a==5)    (b==9))     cout &lt;&lt; "Mislim da ste pogodili barem jedan broj";</pre>
NE (NOT)	!	<pre>if (!(a==5))     cout &lt;&lt; "Broj nije jednak 5";</pre> 

Koristeći tabelu, prepravite prethodni program tako da umjesto dva *if-else*-iskaza koristite jedan *if-else*-iskaz, ali da ne mijenjate uslove nego ih samo spojite.

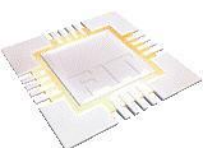
Da li ćete koristiti AND, OR ili/i NOT operator?

Rješenje se nalazi na stranici 58.

### Zadatak 47:

Prepravite program iz zadatka 45 tako da koristite dva if-else-iskaza sa sljedećim uslovima:

- A. `(i >= 1601)`
- B. `(i <= 0)`



Rješenje se nalazi na stranici 58.

#### Zadatak 48:

Koristeći datu tabelu logičkih operatora, prepravite prethodni program tako da umjesto dva *if-else*-iskaza koristite jedan *if-else*-iskaz, ali da ne mijenjate prethodne uslovne izraze nego ih spojite.

Da li ćete koristiti AND, OR ili/i NOT operator?

Rješenje se nalazi na stranici 59.

*Pomoć:*

Pogledajmo tok dijagram za prethodni program. Koji uslovi moraju biti zadovoljeni da bi se ispisala poruka "Broj indeksa je validan"? Ili: 'Koji uslovi ne moraju biti zadovoljeni?'

– Ne, nego pitanje glasi: 'Koji uslovi ne smiju biti zadovoljeni da bi se ispisala poruka?'

– Da bi se ispisala poruka, uslov A ne smije biti zadovoljen i (AND) uslov B ne smije biti zadovoljen.

Koje ćete logičke operatore koristiti? AND, OR ili NOT?

Da li ćete koristiti jedan AND i dva NOT operatora.

#### Zadatak 49:

Koristeći tabelu logičkih operatora, prepravite program iz zadatka 46 tako da umjesto jednog *if-else*-iskaza koristite dva *if*-iskaza, kao na slici 1. Koristite logičke AND, OR ili NOT operatore?

Rješenje se nalazi na stranici 60.

*Pomoć (rješenje):*

Prvi način:

Za ispis poruke "... je validan" koristit ćemo isti uslov kao u programu iz zadatka 46 (pogledajte rješenje na stranici br. 58).

A koji ćemo (složeni) uslov koristiti za ispis poruke "... nije validan"?

Odgovor se krije u sljedećem pitanju:

Koji uslov ne smije biti ispunjen da bi se ispisala poruka "... nije validan"?

Drugi način:

Uslov ...

$! ((i \geq 1) \ \&\& \ (i \leq 1600))$

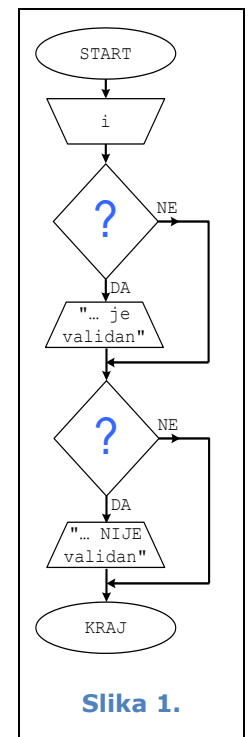
... koji služi za ispis poruke "... nije validan" glasi govornim jezikom:

"Ako **nije**: (broj *i* strogo veći od 1 **AND** broj *i* strogo manji od 1600)"

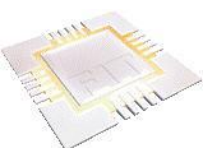
Podsjetite se DeMorganovih pravila iz matematike. Pravila glase:

$$a) (\neg A \wedge \neg B) \Leftrightarrow \neg(A \vee B)$$

$$b) \neg(A \wedge B) \Leftrightarrow (\neg A \vee \neg B)$$



Slika 1.



Uz pomoć *DeMorganovih* pravila uslov  $!(i \geq 1 \ \&\& \ i \leq 1600)$  možemo napisati u obliku...

$$(\ ! (i \geq 1) \ || \ ! (i \leq 1600) )$$

A pošto znamo da vrijedi...

- "*i nije veće ili jednako od 1*"  $!(i \geq 1)$  je isto kao "*i manje od 1*" ( $i < 1$ )
- "*i nije manje ili jednako od 1600*"  $!(i \leq 1600)$  je isto kao "*i veće od 1600*" ( $i > 1600$ )

...onda možemo napisati prethodno dobijeni uslov i na sljedeći način:

$$((i < 1) \ || \ (i > 1600))$$

Novi krajnji uslov se može opisati i riječima:

Ispiši poruku "...nije validan" ako je "*i manje od 1 ili i veće od 1600*"

### ZAKLJUČAK:

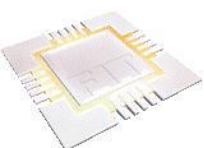
Dva (dvostruka) *if-else*-iskaza (iz zadatka 45) koja su pretvorena iz jednog složenog *if-else*-iskaza (iz zadatka 46) su **ugniježđena** jer uslov  $(i \geq 1 \ \&\& \ i \leq 1600)$  sadrži logičko **AND**.

Dva (jednostruka) *if*-iskazi koja su nastala iz jednog složenog *if*-iskaza (koji sadrži logičko **OR**) moraju biti **jedan ispod drugog**, a **ne** smiju biti ugniježđena. Primjer *if*-iskaza sa logičko **OR** je *if*-iskaz (za ispis poruke "...nije validan" iz rješenja zadatka 49c) čiji uslov glasi:  $(i < 1 \ || \ i > 1600)$ .

### Zadatak 51:

Prepravite, na osnovu prethodnog zaključka, program iz zadatka 49c tako da u *if*-iskazima ne koristite ni jedno logičko **OR** i logičko **AND**.

Rješenje se nalazi na stranici 61.



## Koju vrijednost ima logički operator?

### Zadatak 52:

Isprobajte sljedeći program. On vam daje odgovor na pitanje koju vrijednost imaju lšta je uslov i čemu je on jednak?

```
4: void main()
5: {
6:     int a = 5;
7:     int b = 0;
8:
9:     cout << "Zadovoljen uslov je isto kao " << (a>b) << endl;
10:
11:     cout << "Nezadovoljen uslov je isto kao " << (a<b) << endl;
12: }
```

Šta možete zaključiti?

*Odgovor se nalazi na stranici 61.*

### Zadatak 53:

Napravite neki program u kojem ćete koristiti dvije varijable čija će vrijednost biti neki uslov.

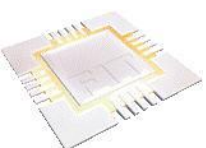
*Odgovor se nalazi na stranici 61.*

## Vrijednosti izraz umjesto logičkog operatora

Na osnovu onog što ste do sada pročitali možete zaključiti da su sljedeća dva programa potpuno ista:

```
4: void main()
5: {
6:     if (5>0)
7:         cout << "uslov je zadovoljen" << endl;
8:     else
9:         cout << "uslov NIJE zadovoljen" << endl;
10: }
```

```
4: void main()
5: {
6:     if (1)
7:         cout << "uslov je zadovoljen" << endl;
8:     else
9:         cout << "uslov NIJE zadovoljen" << endl;
10: }
```



**Zadatak 54:**Šta će ispisati sljedeći *if-else*-iskazi:

<b>a)</b> if (0) cout << "DA"; else cout << "NE";	<b>b)</b> if (1) cout << "DA"; else cout << "NE";
<b>c)</b> int a = 0; if (a = 1) cout << "DA"; else cout << "NE";	<b>d)</b> int a = 0; if (a == 1) cout << "DA"; else cout << "NE";

*Pomoć:*Pravilo za *if*-iskaze glasi:Ako je **mjesto uslova RAZLIČITO OD NULE** uslov je zadovoljen i biće izvršen slučaj „DA”.Prema tom pravilu u zadatku **a** će biti ispisano „NE” jer mesto uslova (tj. **0**) NIJE RAZLIČITO OD NULE. ( $\tau(0 \neq 0) = \perp$ ).U zadatku **b** će biti ispisano „DA”, jer mesto uslova (tj. **1**) JE RAZLIČITO OD NULE. ( $\tau(1 \neq 0) = T$ )Po istom principu ako bi koristili *if*(**3**) uslov bi bio zadovoljen, jer je mesto uslova RAZLIČITO OD NULE. ( $\tau(3 \neq 0) = T$ )Zašto će po ovom pravilu pod u zadatku *d* biti ispisano „Ne”?*Odgovor:* - Da se podsjetimo: Ako je neki uslov (npr.  $3 > 1$ ) zadovoljen, program će na to mesto postaviti vrijednost 1, a ako nije postaviti će vrijednost 0. U to smo se uvjerali naredbama kao:

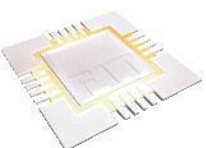
```
if (3 > 1)
    cout << "DA";
else
    cout << "NE";
```

```
cout << (3 > 1);           // ispis je 1
cout << (3 < 1);           // ispis je 0
```

U zadatku **c** na mjestu uslova stoji **a=1**. To nije uslov, već izraz, jer nemamo **a==1** nego **a=1**. Kompajler će prvo izvršiti sve izraze na mjestu uslova a potom će vršiti provjeru da li je **mjesto uslova** različito od nule. U zadatku **c** varijabla **a** će dobiti novu vrijednost 1, a zatim će se ispisati „DA”, jer je *mjesto uslova* različito od nule. To može, inače, biti veliki problem. Ako mi nesvjesno izostavimo znak "=" (tj. umjesto '==' stavimo '=') kompajler **neće** prijaviti grešku, i izvršiti će se pogrešne naredbe. A pored takve fatalne greške, vrijednost varijable **a** će se promijeniti, i ako to nismo željeli. Iz tog razloga kad god koristite u uslovu '==' provjerite više puta da niste slučajno stavili '='.

Još da znate:

```
if (cout << "Da li ovo moguće?")
    cout << "JESTE, jer izraz ima vrijednost 1";
else
    cout << "Ko ne vjeruje neka isproba";
```



### Zadatak 55:

Prepravite program 49c tako da koristite što više varijabli za uslove.

*Rješenje se nalazi na stranici 62.*

### Zadatak 56:

...zadatak sa ispita iz predmeta "Uvod u programiranje" – 24.01.2006.

Napravite program koji će ispisati da li je uneseni broj prihvaćen. Broj je prihvaćen ako je zadovoljio sljedeće nabrojane uslove:

1. pozitivan
2. neparan (*da nije djeljiv sa 2*)
3. da nije trocifren (*da nije u rasponu od 100 do 999*)
4. da nije djeljiv sa 7

Sljedeći brojevi spadaju u izuzetke, i oni se takođe prihvaćaju bez provjere prethodno navedenih uslova:

5. 114
6. -99 do -33
7. brojevi djeljivi sa 19

Preporučujemo da koristite što više varijabli tipa `bool`.

*Naprimjer:*

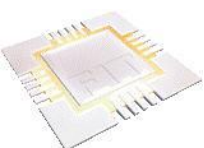
Broj -64 je prihvaćen iako on ne zadovoljava niti prvi uslov (jer je paran broj), niti drugi uslov (jer je negativan broj), ali zadovoljava šesti uslov-slučaj izuzetka (jer je broj -64 u rasponu -99 do -33).

Broj +65 je prihvaćen jer zadovoljava prvi, drugi, treći i četvrti uslov.

Broj +380 je prihvaćen iako on ne zadovoljava dva od prva četiri uslova, ali zadovoljava sedmi uslov izuzetka (jer je broj 380 djeljiv sa 19).

Da bi broj bio prihvaćen on mora zadovoljiti sva četiri uslova ili on mora zadovoljiti neki od tri moguća slučaja izuzetaka.

*Rješenje se nalazi na stranici 62.*



## true i false

U Visual C++-u možete koristiti i ključne riječi `true` i `false` umjesto brojeva 1 i 0. To vam dokazuje sljedeći program:

```

1: #include <iostream>
2: using namespace std;
3:
4: void main()
5: {
6:     cout << "true = " << true << endl;      //ispisat će se 'true = 1'
7:     cout << "false = " << false << endl;    //ispisat će se 'false = 0'
8:     cout << "2*true = " << 2*true << endl;  //ispisat će se '2*true = 2'
9: }

```

Upotreba varijabli za uslove je moguća na dva načina:

1. kao u zadatku 57:

```
niz = (i>=-99 && i<=-33);
```

2. korištenjem `true` i `false`:

```

if (i>=-99 && i<=-33)
    niz = true;
else
    niz = false;

```

odnosno:

```

if (i>=-99 && i<=-33)
    niz = 1;
else
    niz = 0;

```

### Zadatak 57:

Prepravite program iz zadatka 57 tako da varijable za uslove upotrebljavate na gore spomenuti način pod 2.

Rješenje se nalazi na stranici 63.

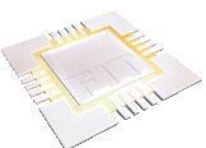
## Prioriteti operatora

### Zadatak 58:

Napravite program koji ima samo jednu liniju koda i to `cout`-naredbu za ispis vrijednosti:

$$\frac{20}{2 \cdot 5}$$

Rješenje se nalazi na stranici 63.



**Zadatak 59:**

Koji od sljedeća dva programa ispisuje ispravnu poruku? Kojem *if*-iskazu pripada *else*?

```

4: void main()
5: {
6:     int c;
7:
8:     cout << "Unesite broj: ";
9:     cin >> c;
10:
11:     if (c >= 5)
12:         if (c >= 10)
13:             cout << "Broj je (strogo) veci od 10 \n";
14:         else
15:             cout << "Broj je izmedju 5 i 10 \n";
16: }
```

```

4: void main()
5: {
6:     int c;
7:
8:     cout << "Unesite broj: ";
9:     cin >> c;
10:
11:     if (c >= 5)
12:         if (c >= 10)
13:             cout << "Broj je (strogo) veci od 10 \n";
14:     else
15:         cout << "Broj je manji od 5 \n";
16: }
```

**Zadatak:**

1. Kojem *if*-iskazu pripada *else*? Kako glasi pravilo?
2. Dodajte neispravnom programu vitičaste zagrade tako da ispisuje ispravna poruku!
3. Dodajte ispravnom programu vitičaste zagrade tako da se lakše može raspoznati koja će se poruka ispisati

Rješenje se nalazi na stranici 64.

**Zadatak 60:**

Napravite program koji će zahtijevati unos dva broja (*a* i *b* tipa *int*) i ispisuje vrijednost *y*.

Trebate izračunati vrijednost  $y = \frac{a}{b}$  ako su zadovoljeni sljedeći uslovi:

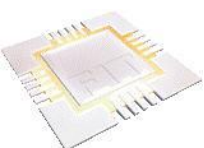
- $a > 0$
- broj *a* ne smije biti djeljiv sa 19
- $a \notin [111, 555]$
- $b \neq 0$
- $a > 3b$

Izuzetak gore navedenim uslovima su sljedeći slučajevi:

- $b = 13$
- $b \in (127, 255]$
- $b = a$

Za slučajeve koji ne zadovoljavaju nabrojane uslove ili ne spadaju u izuzetke potrebno

zahtijevati unos broja *c* (tipa *int*) i izračunati vrijednost  $y = \frac{ab}{\sqrt{c}}$ .

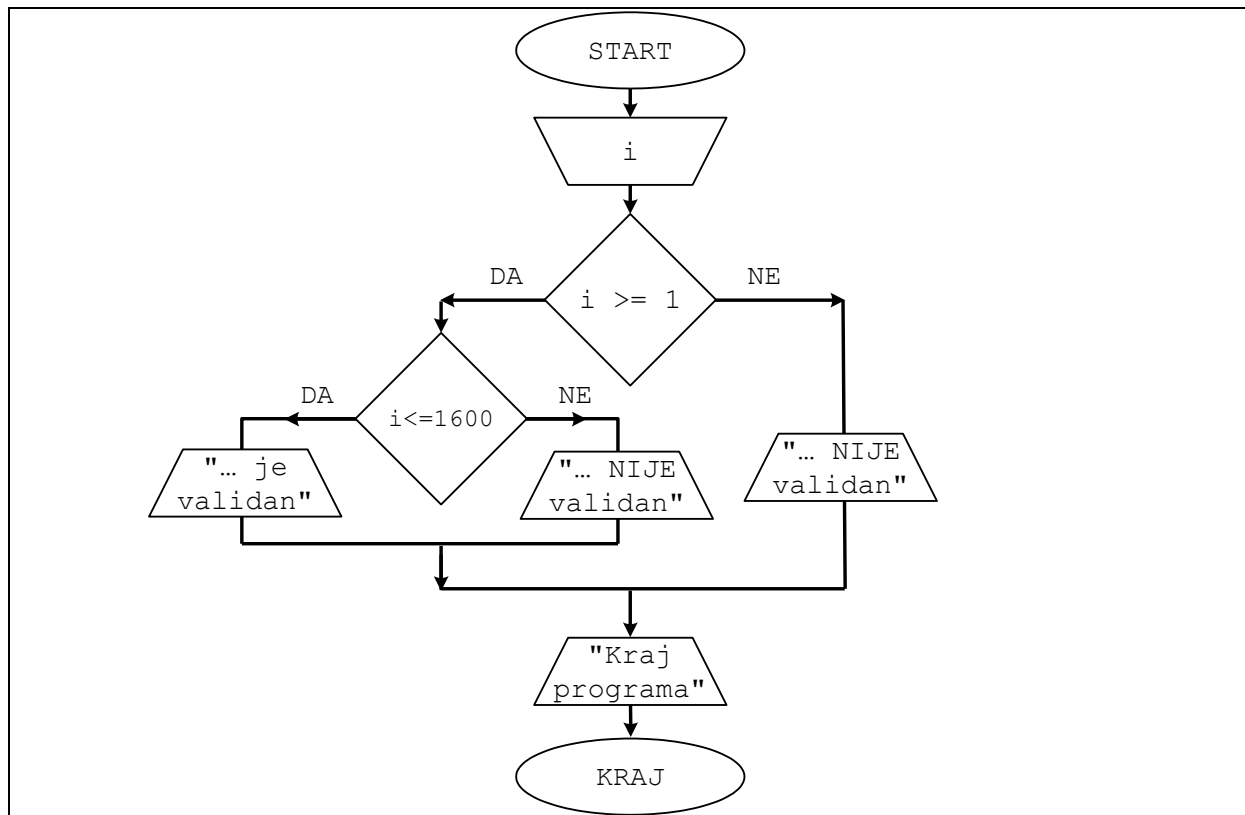




Rješenje se nalazi na stranici 64.

## Rješenja

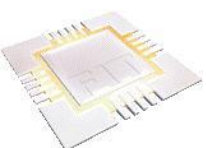
Rješenje zadatka br. **45**:



```

1:  #include <iostream>
2:  using namespace std;
3:
4:  void main()
5:  {
6:      int i;
7:      cout << "Unesi broj indeksa: ";
8:      cin >> i;
9:
10:     if (i >= 1)
11:     {
12:         if (i <= 1600)
13:             cout << "To je validan broj indeksa \n";
14:         else
15:             cout << "To NIJE validan broj indeksa \n";
16:     }
17:     else
18:     {
19:         cout << "To NIJE validan broj indeksa \n";
20:     }
21:
22:     cout << "Kraj programa \n";
23: }
  
```

Broj indeksa je validan ako je ' $i \geq 1$ ' i ako je ' $i \leq 1600$ '.

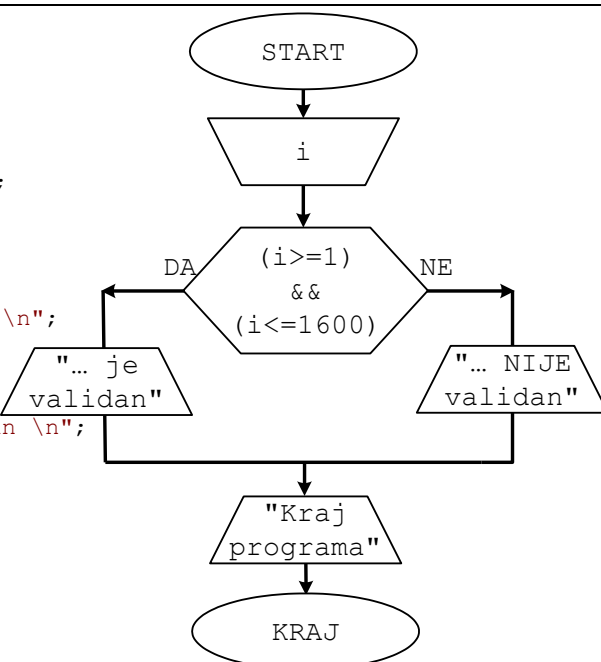


## Rješenje zadatka br. 46:

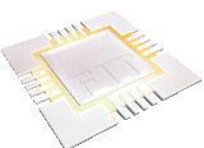
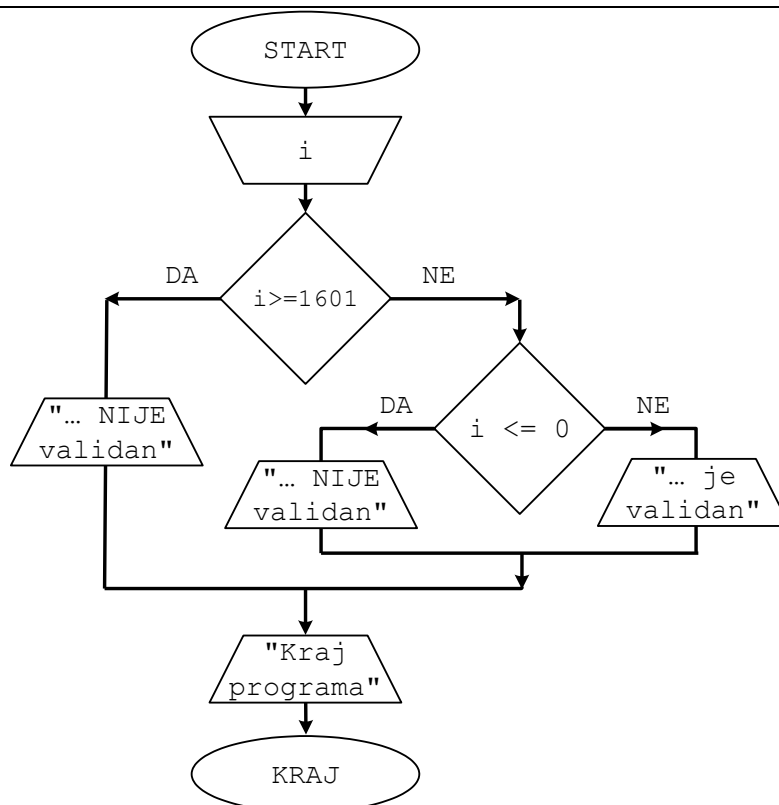
```

1:  #include <iostream>
2:  using namespace std;
3:
4:  void main()
5:  {
6:      int i;
7:      cout << "Unesi broj indeksa: ";
8:      cin >> i;
9:
10:     if ((i >= 1) && (i <= 1600))
11:     {
12:         cout << "... je validan \n";
13:     }
14:     else
15:     {
16:         cout << "... NIJE validan \n";
17:     }
18:
19:     cout << "Kraj programa \n";
20: }

```



## Rješenje zadatka br. 47 :



```
1:  #include <iostream>
2:  using namespace std;
3:
4:  void main()
5:  {
6:      int i;
7:      cout << "Unesi broj indeksa: ";
8:      cin >> i;
9:
10:     if (i >= 1601)
11:     {
12:         cout << "To NIJE validan broj indeksa \n";
13:     }
14:     else
15:     {
16:         if (i <= 0)
17:             cout << "To NIJE validan broj indeksa \n";
18:         else
19:             cout << "To je validan broj indeksa \n";
20:     }
21:
22:     cout << "Kraj programa \n";
23: }
```

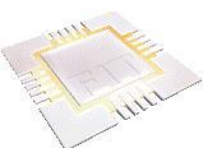
#### Rješenje zadatka br. 48 :

```
1:  #include <iostream>
2:  using namespace std;
3:
4:  void main()
5:  {
6:      int i;
7:      cout << "Unesi broj indeksa: ";
8:      cin >> i;
9:
10:     if ( !(i >= 1601) && !(i <= 0) )
11:     {
12:         cout << "To je validan broj indeksa \n";
13:     }
14:     else
15:     {
16:         cout << "To NIJE validan broj indeksa \n";
17:     }
18:
19:     cout << "Kraj programa \n";
20: }
```

#### Napomena:

Logičko NOT (!) ima veći prioritet od logičkog AND (&&)

Logičko NOT (!) ima veći prioritet od logičkog OR (||)



## Rješenje zadatka br. 49:

## Prvi način: a)

```
4: void main()
5: {
6:     int i;
7:     cout << "Unesi broj indeksa: ";
8:     cin >> i;
9:
10:    if ((i >= 1) && (i <= 1600))
11:    {
12:        cout << "Broj indeksa je validan \n";
13:    }
14:
15:    if (! ((i >= 1) && (i <= 1600)) )
16:    {
17:        cout << "Broj indeksa NIJE validan \n";
18:    }
19:
20:    cout << "Kraj programa \n";
21: }
```

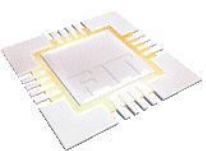
Zbog većeg prioriteta kod logičko NOT (!) u odnosu na logičko AND (&&), znak '!' mora se nalaziti ispred dodatne zagrade (u gore prikazanom rješenju te zagrade su boldirane), inače bi se negacija vezala samo za prvi uslov ( $i \geq 1$ ).

## Drugi način: b)

```
4: void main()
5: {
6:     int i;
7:     cout << "Unesi broj indeksa: ";
8:     cin >> i;
9:
10:    if ((i >= 1) && (i <= 1600))
11:    {
12:        cout << "Broj indeksa je validan \n";
13:    }
14:
15:    if ( !(i >= 1) || !(i <= 1600) )
16:    {
17:        cout << "Broj indeksa NIJE validan \n";
18:    }
19:
20:    cout << "Kraj programa \n";
21: }
```

## ili c)

```
4: void main()
5: {
6:     int i;
7:     cout << "Unesi broj indeksa: ";
8:     cin >> i;
9:
10:    if ((i >= 1) && (i <= 1600))
11:    {
12:        cout << "Broj indeksa je validan \n";
13:    }
14:
15:    if (i < 1 || i > 1600)
16:    {
17:        cout << "Broj indeksa NIJE validan \n";
18:    }
19:
20:    cout << "Kraj programa \n";
21: }
```

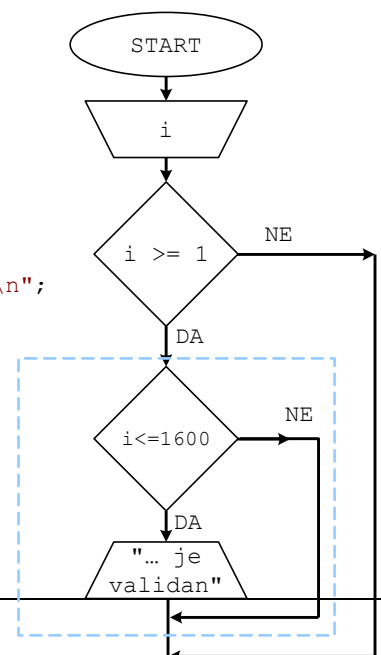


## Rješenje zadatka br. 51:

```

4: void main()
5: {
6:     int i;
7:     cout << "Unesi broj indeksa: ";
8:     cin >> i;
9:
10:    if (i >= 1)
11:    {
12:        if (i <= 1600)
13:        {
14:            cout << "Broj indeksa je validan \n";
15:        }
16:    }
17:
18:    if (i < 1)
19:        cout << "Broj indeksa NIJE validan \n";
20:
21:    if (i > 1600)
22:        cout << "Broj indeksa NIJE validan \n";
23:
24:    cout << "Kraj programa \n";
25: }

```



## Rješenje zadatka br. 52:

Ako je uslov zadovoljen, program na njegovo mjesto stavlja vrijednost 1, a ako nije onda vrijednost 0.

Zbog toga možete vrijednost nekog uslova dodijeliti varijabli tipa *int*.

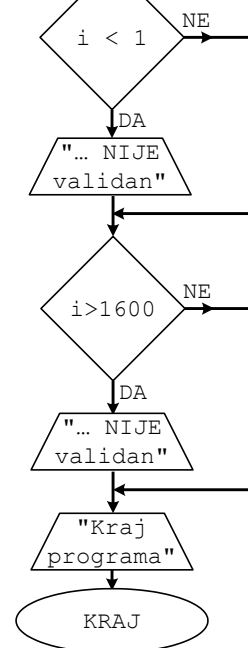
U mnogim kompajlerima (kao Visual C++) možete koristiti i posebnu varijablu tipa *bool*, koja može imati vrijednost 0 ili 1.

## Rješenje zadatka br. 53:

```

4: void main()
5: {
6:     int a = 5;
7:     int b = 0;
8:     int uslov1 = (a > b);
9:     int uslov2 = (a < b);
10:
11:     cout << "Zadovoljen uslov: " << uslov1 << endl;
12:     cout << "Nezadovoljen uslov: " << uslov2 << endl;
13: }

```

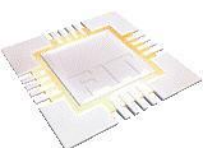


*Ili:*

```

8: bool uslov1 = (a > b);
9: bool uslov2 = (a < b);

```

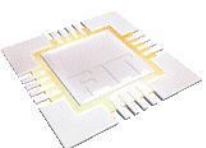


## Rješenje zadatka br. 55:

```
1:  #include <iostream>
2:  using namespace std;
3:
4:  void main()
5:  {
6:      int i;
7:      int a, b, c, d, e, f;
8:      //bool a, b, c, d, e, f;
9:
10:     cout << "Unesi broj indeksa: ";
11:     cin >> i;
12:
13:     a = (i >= 1);
14:     b = (i <= 1600);
15:     c = (a && b);
16:
17:     if (c)
18:         cout << "Broj indeksa je validan \n";
19:
20:     d = !a;
21:     e = !b;
22:     f = (d || e);
23:
24:     if (f)
25:         cout << "Broj indeksa NIJE validan \n";
26:
27:     cout << "Kraj programa \n";
28: }
```

## Rješenje zadatka br. 56:

```
1:  #include <iostream>
2:  using namespace std;
3:
4:  void main()
5:  {
6:      int i;
7:      int kontrola, niz1, niz2, zlatni;
8:      cout << "Unesi broj ";
9:      cin >> i;
10:
11:      niz1 = (i>=100 && i<=999);
12:      kontrola = (i>=0) && (i%2!=0) && !niz1 && !(i%7==0);
13:
14:      niz2 = (i>=-99) && (i<=-33);
15:      zlatni = (i==114) || niz2 || (i%19==0);
16:
17:      if (kontrola || zlatni)
18:          cout << "Broj je prihvacen \n";
19:      else
20:          cout << "Broj NIJE prihvacen \n";
21: }
```



Rješenje zadatka br. **57**:

```

1:  #include <iostream>
2:  using namespace std;
3:
4:  void main()
5:  {
6:      int i;
7:      int kontrola, niz1, niz2, zlatni;
8:      cout << "Unesi broj ";
9:      cin >> i;
10:
11:      if (i>=100 && i<=999)
12:          niz1 = true;
13:      else
14:          niz1 = false;
15:
16:
17:      if ((i>=0) && (i%2!=0) && !niz1 && !(i%7==0))
18:          kontrola = true;
19:      else
20:          kontrola = false;
21:
22:
23:      if (i>=-99 && i<=-33)
24:          niz2 = true;
25:      else
26:          niz2 = false;
27:
28:
29:      if ((i==114) || niz2 || (i%19==0))
30:          zlatni = true;
31:      else
32:          zlatni = false;
33:
34:
35:      if (kontrola || zlatni)
36:          cout << "Broj je prihvacen \n";
37:      else
38:          cout << "Broj NIJE prihvacen \n";
39:  }

```

Rješenje zadatka br. **58** :

Ako ste riješili zadatak ovako...

```

4:  void main()
5:  {
6:      cout << 20/2*5 << endl;    // ispisat će se 50
7:  }

```

... onda ste pogrešno riješili zadatak, jer će program ispisati vrijednost 50 a ne 2 koju očekujemo. *Zašto?*

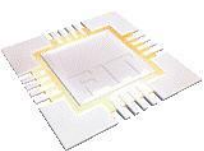
C++ računa operatore istog prioriteta sa lijeva na desno, tako da će prvo izračunati izraz  $20/2 = 10$  a zatim izraz  $10*5 = 50$ .

Stoga je neophodno izraz  $2*5$  staviti u zagrade kao...

```

4:  void main()
5:  {
6:      cout << 20/(5*2) << endl;    // ispisat će se 2
7:  }

```



## Rješenje zadatka br. 59 :

'else' pripada prvom slobodnom if-iskazu!

```
13:         if (c >= 5)
14:         {
15:             if (c >= 10)
16:                 cout << "Broj je (strogo) veci od 10 \n";
17:             else
18:                 cout << "Broj je izmedju 5 i 10 \n";
19:         }
20:     }
```

```
13:         if (c >= 5)
14:         {
15:             if (c >= 10)
16:                 cout << "Broj je (strogo) veci od 10 \n";
17:         }
18:         else
19:             cout << "Broj je manji od 5 \n";
20:     }
```

*Preporuka:*

Nemojte u ovakvim slučajevima izostavljati dodatne vitičaste zagrade.

## Rješenje zadatka br. 60:

```
1:  #include <iostream>
2:  #include <math.h>
3:  using namespace std;
4:
5:  void main()
6:  {
7:      int a, b, c;
8:      float y;
9:      bool niz1, kontrola, niz2, izuzetak;
10:
11:      cout << "Unesite broj a: ";
12:      cin >> a;
13:
14:      cout << "Unesite broj b: ";
15:      cin >> b;
16:
17:      niz1 = (a >= 111) && (a < 555);
18:      kontrola = (a>0) && !(a%19==0) && !niz1 && (a!=0) && (a>3*b);
19:
20:      niz2 = (b > 127) && (b <= 255);
21:      izuzetak = (b==13) || niz2 || (a==b);
22:
23:      if (kontrola || izuzetak)
24:          y = a / float(b);
25:      else
26:      {
27:          cout << "Unesite broj c: ";
28:          cin >> c;
29:          y = a*b / sqrt(float(c)); //zasto float? pogledajte W4, zadatak27
30:      }
31:
32:      cout << "y = " << y << endl;
33:  }
```

