



operatori

prof. Kadić Neđad

operatori

- Operatori su ***simboli*** koji predstavljaju određene ***funkcije***. Možemo ih podijeliti na:
 - ***aritmetički*** operatori,
 - ***logički*** operatori,
 - **relacijske** operatore,
 - ***ostali*** operatori.

Binarni aritmetički operatori

simbol	funkcija
+	zbrajanje
-	oduzimanje
*	množenje
/	dijeljenje
%	modul (za tip <code>int</code> cjelobrojni ostatak dijeljenja)

Primjer 1

- Treba izračunati zbroj, razliku, proizvod i količnik dvaju realnih brojeva.
- Ispis neka bude oblika:

```
Unesi prvi broj:  
unesi drugi broj:  
..... + ..... = .....  
..... - ..... = .....  
..... * ..... = .....  
..... / ..... = .....
```



Primjer 1

- Deklarirati dvije ili šest realnih varijabli!
- Aritmetički operatori i znak = mogu se unijeti kao znakovni nizovi!

```
#include<iostream>
using namespace std;
int main()
{

    return 0;
}
```

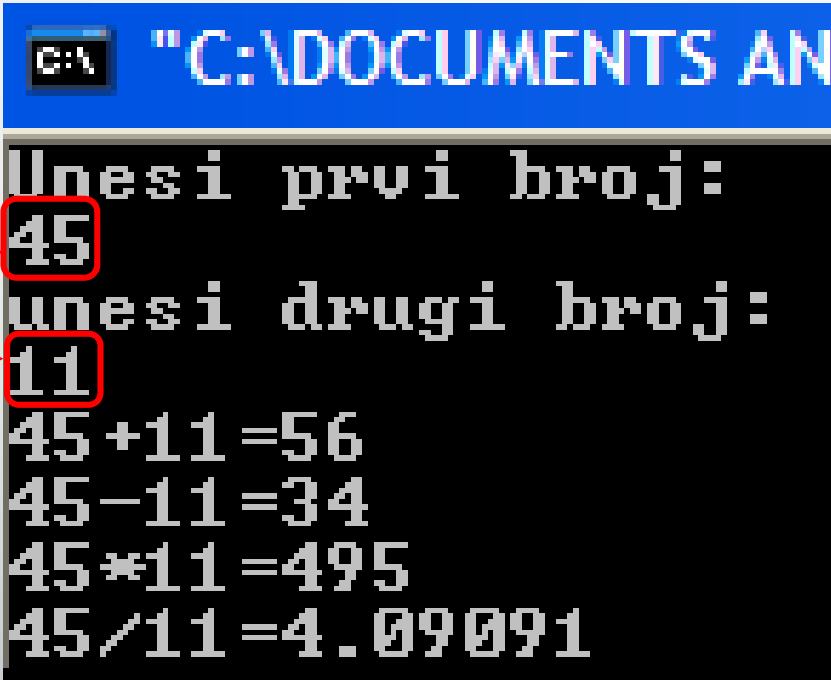
```
#include<iostream>
using namespace std;
int main()
{
    float a,b,c,d,e,f;
    cout<<"Unesi prvi broj:"<<endl;
    cin>>a;
    cout<<"unesi drugi broj:"<<endl;
    cin>>b;
    c=a+b;
    d=a-b;
    e=a*b;
    f=a/b;
    cout<<a<<"+"<<b<<"="<<c<<endl;
    cout<<a<<"-"<<b<<"="<<d<<endl;
    cout<<a<<"*"<<b<<"="<<e<<endl;
    cout<<a<<"/"<<b<<"="<<f<<endl;
    return 0;
}
```

- Zadatak je moguć uraditi koristeći samo dvije varijable:

```
#include<iostream>
using namespace std;
int main()
{
    float a,b;
    cout<<"Unesi prvi broj:"<<endl;
    cin>>a;
    cout<<"unesi drugi broj:"<<endl;
    cin>>b;
    cout<<a<<"+"<<b<<"="<<a+b<<endl;
    cout<<a<<"-"<<b<<"="<<a-b<<endl;
    cout<<a<<"*"<<b<<"="<<a*b<<endl;
    cout<<a<<"/"<<b<<"="<<a/b<<endl;
    return 0;
}
```

Primjer 1

- Provjera programa s podacima za koje je unaprijed poznat krajnji rezultat:



```
C:\DOCUMENTS AN
unesi prvi broj:
45
unesi drugi broj:
11
45+11=56
45-11=34
45*11=495
45/11=4.09091
```


Primjer 2



- Potrebno je izračunati cjelobrojni količnik i cjelobrojni ostatak dijeljenja dvaju cijelih brojeva.
- Ispis neka bude oblika:

Upisi djeljenik:

Upisi djelitelj:

... : ... = ... cijelih, a ostatak je ...

Primjer 2

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,ostatak,kvoc;
    cout<<"Upisi djeljenik:";
    cin>>a;
    cout<<"Upisi djelitelj:";
    cin>>b;
    kvoc=a/b;
    ostatak=a%b;
    cout<<a<<" : " <<b<<" = " <<kvoc
        <<" cijelih, a ostatak je " <<ostatak<<endl;
    return 0;
}
```

Primjer 2 – moguća pogreška

- Ako se varijable za pohranu *ulaznih podataka* deklariraju kao *realne*, javit će se *pogreška!*

```
5 float a,b,ostatak, kolicnik;
6 cout<<"Upisi dielienik:";
7 cin>>a;
8 cout<<"Upisi djelitelj:";
9 cin>>b;
10 kolicnik=a/b;
11 ostatak=a%b;
12 cout<<a<<" "<<b<<"="<< kolicnik
13 <<" cijelih, a ostatak je "<<ostatak;
14 return 0;
```

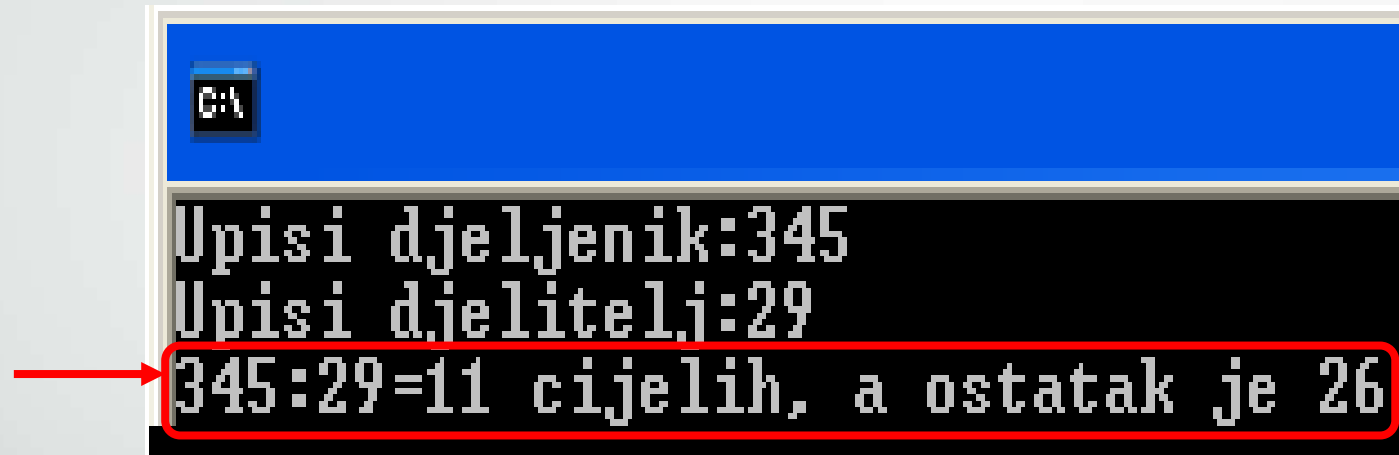
Logs & others

Code::Blocks Search results Cccc Build log Build messages CppCheck

File	Line	Message
C:\Users\kadic...		In function 'int main()':
C:\Users\kadic...	11	error: invalid operands of types 'float' and 'float' to binary 'operator%'
=== Build failed: 1 error(s), 0 warning(s) (0 minute(s), 1 second(s)) ===		

Primjer 2

- Provjera programa s podacima za koje je unaprijed poznat krajnji rezultat.



```
G:\nUpisi djeljenik:345\nUpisi djelitelj:29\n345:29=11 cijelih, a ostatak je 26
```

Primjer 3



- Potrebno je izračunati srednju vrijednost po volji izabrana četiri realna broja.
- Ispis neka bude oblika:

```
Unesi cetiri realna broja:
```

```
Srednja vrijednost brojeva:
```

```
....., ....., ..... i ..... je .....
```

Primjer 3

```
#include<iostream>
using namespace std;
int main()
{
    float a,b,c,d, srednja_vrijednost;
    cout<<"Unesi cetiri realna broja"<<endl;
    cin>>a>>b>>c>>d;
    srednja_vrijednost=(a+b+c+d)/4;
    cout<<endl<<"Srednja vrijednost brojeva:";
    cout<<endl<<a<<" , "<<b<<" , "<<c<<" i "<<d
        <<" je "<< srednja_vrijednost <<endl;
    return 0;
}
```

Primjer 3 – moguća pogreška

- Ako se srednja vrijednost računa prema ovoj formuli, rezultat nije ispravan (prioriteti!)

srednja_vrijednost=a+b+c+d/4;

```
"C:\DOCUMENTS AND SETTINGS"
Unesi cetiri realna broja.
45 6 34 105

Srednja vrijednost brojeva:
45, 6, 34 i 105 je 111.25
```

Računa se:
 $a+b+c+(d/4)$

Primjer 3 – moguća pogreška

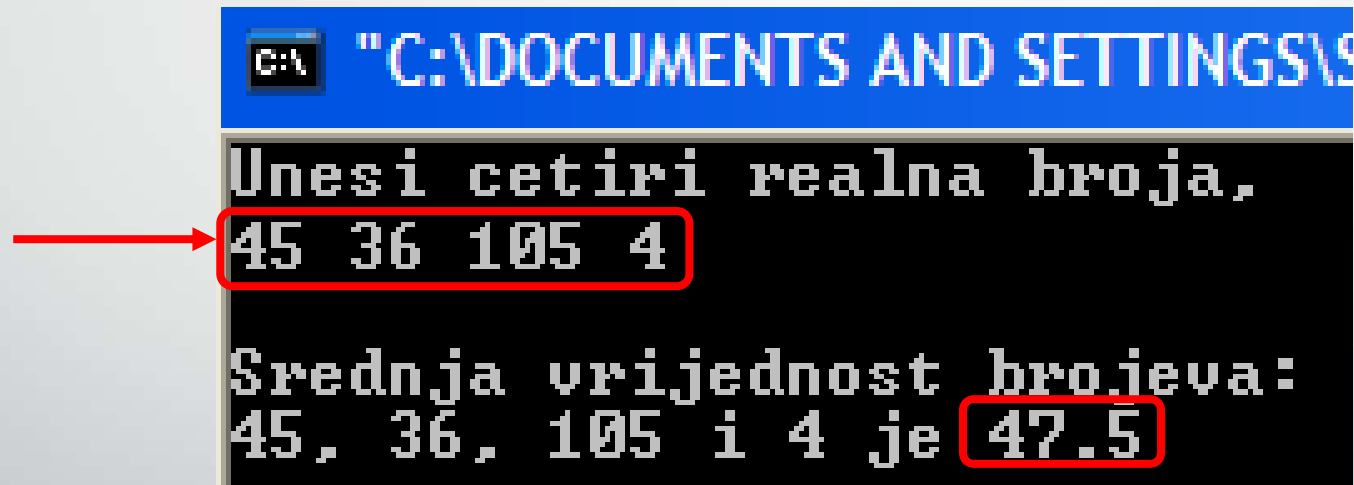
- Ako ispis nije pregledan, između riječi i brojeva treba staviti razmake (potrebno ih je predvidjeti unutar znakova navodnika)!

```
Srednja vrijednost brojeva  
45,6,34i105je47.5
```

```
Srednja vrijednost brojeva:  
45, 6, 34 i 105 je 47.5
```


Primjer 3

- Provjera programa s podacima za koje je unaprijed poznat krajnji rezultat.



```
C:\> "C:\DOCUMENTS AND SETTINGS\S...
Unesi cetiri realna broja,
45 36 105 4
Srednja vrijednost brojeva:
45, 36, 105 i 4 je 47.5
```

Unarni aritmetički operatori

simbol	funkcija	svrha
- a	<i>unarni minus</i>	Mijenja predznak broja.
++a	operator <i>uvećanja</i> <i>za 1</i>	Uvećava sadržaj varijable, a potom vraća promijenjenu vrijednost.
a++		Vraća postojeću vrijednost, a potom uvećava sadržaj varijable.
--a	operator <i>umanjenja</i> <i>za 1</i>	Umanjuje sadržaj varijable, a potom vraća promijenjenu vrijednost.
a--		Vraća postojeću vrijednost, a potom umanjuje sadržaj varijable.

Unarni operatori prije varijable

- Ako je operator *ispred varijable* (npr. $++a$) tada se u izrazu računa *s uvećanom/umanjenom* vrijednošću varijable. U primjeru:

a=1;

b=++a +5;

po izvršenju naredbi, sadržaj varijable **b=7, a=2.**

Unarni operatori poslije varijable

- Ako je operator *iza varijable* (npr. $a++$) tada se vrijednost varijable uveća/umanji *tek nakon izračunavanja* izraza. U primjeru:

$a=1;$

$b=a++ +5;$

po izvršenju naredbi, sadržaj varijable $b=6$, $a=2$.

Primjer 4



- Potrebno je unijeti cijeli broj i pohraniti ga u varijablu A. U varijablu X pohraniti vrijednost koja se dobije kada se na varijablu A primijeni unarni operator za uvećanje:
 - a) nakon dobave iz memorije (A++),
 - b) prije dobave iz memorije (++A).
- Ispisati sadržaj varijabli X i A.

Primjer 4a



- Ispis neka bude oblika:

```
Unesi broj za operaciju A++:  
Kad operacija zavrsi X= ...  
Kad operacija zavrsi A= ...
```

Primjer 4a

```
#include<iostream>
using namespace std;
int main()
{
    int X,Y,A;
    cout<<"Unesi broj za operaciju A++: ";
    cin>>A;
    X=A++;
    cout<<"Kad operacija završi X= "<<X<<endl;
    cout<<"Kad operacija završi A= "<<A<<endl;
    return 0;
}
```

```
#include<iostream>
using namespace std;
int main()
{
    int X,A;
    cout<<"Unesi broj za operaciju A++: ";
    cin>>A;
    X=A++;
    cout<<"Kad operacija zavrshi X= "<<X<<endl;
    cout<<"Kad operacija zavrshi A= "<<A<<endl;
    return 0;
}
```


Primjer 4a

- Provjera programa s podacima za koje je unaprijed poznat krajnji rezultat.

```
C:\n
Unesi broj za operaciju A++: 5
Kad operacija završi X= 5
Kad operacija završi A= 6
```

Primjer 4b

- Ispis neka bude oblika:

```
Unesi broj za operaciju ++A:  
Kad operacija završi X= ...  
Kad operacija završi A= ...
```

Primjer 4b

```
#include<iostream>
using namespace std;
int main()
{
    int X,Y,A;
    cout<<"Unesi broj za operaciju ++A: ";
    cin>>A;
    X=++A;
    cout<<"Kad operacija zavrsi X= "<<X<<endl;
    cout<<"Kad operacija zavrsi A= "<<A<<endl;
    return 0;
}
```

```
#include<iostream>
using namespace std;
int main()
{
    int X,A;
    cout<<"Unesi broj za operaciju ++A: ";
    cin>>A;
    X=++A;
    cout<<"Kad operacija zavrshi X= "<<X<<endl;
    cout<<"Kad operacija zavrshi A= "<<A<<endl;
    return 0;
}
```

Primjer 4b

- Provjera programa s podacima za koje je unaprijed poznat krajnji rezultat.

```
C:\nUnesi broj za operaciju ++A: 5  
Kad operacija završi X= 6  
Kad operacija završi A= 6
```

Tip podatka operanada i rezultata

- 1) ***Tip podatka rezultata aritmetičkog izraza određuju tipovi podataka operanada iz izraza.***

(Ako su operandi u izrazu tipa float i rezultat aritmetičkog izraza je također tog tipa.)

- 2) ***Kada se u izrazu nađe više različitih tipova podataka operanada, podaci se svode na zajednički tip prije zadane operacije.***

(Pravila pretvorbe su usmjerena prema višem tipu podataka.)

Primjer 5



- Treba izračunati količnik dva broja i rezultat spremiti u zasebnu varijablu. Kombinirati razne tipove podataka operanada i rezultata pa promatrati posljedice. Ispis neka bude oblika:

```
Unesi prvi broj:  
unesi drugi broj:  
kolicnik iznosi: ....
```

Primjer 5a

```
#include<iostream>
using namespace std;
int main()
{
    int a, kolicnik;
    float b;
    cout<<"Unesi prvi broj \n";
    cin>>a;
    cout<<"Unesi drugi broj \n";
    cin>>b;
    kolicnik=a/b;
    cout<<"Kolicnik iznosi: "<<kolicnik<<endl;
    return 0;
}
```


Primjer 5a

```
Unesi prvi broj  
5  
Unesi drugi broj  
2  
Kolicnik iznosi: 2
```

- Zbog toga što su *operandi različitog tipa*, prvo se pretvaraju u viši tip, a to je *float*. Po izračunu, rezultat je tipa *float* (2.5).
- Rezultat se pridružuje varijabli *količnik* koja je tipa *int* (2)

Primjer 5b

```
#include<iostream>
using namespace std;
int main()
{
    int a,b ;
    float kolicnik;
    cout<<"Unesi prvi broj \n";
    cin>>a;
    cout<<"Unesi drugi broj \n";
    cin>>b;
    kolicnik=a/b;
    cout<<"Kolicnik iznosi: "<<kolicnik<<endl;
    return 0;
}
```

Primjer 5b

```
Unesi prvi broj
43
Unesi drugi broj
11
Kolicnik iznosi: 3
```

- Zbog toga što su *operandi* tipa *int* i *rezultat* izraza je tipa *int* ($43/11=3$).
- Po izračunu, *rezultat* (3) se pridružuje varijabli kolicnik koja je po tipu *float* (3.).

Primjer 6



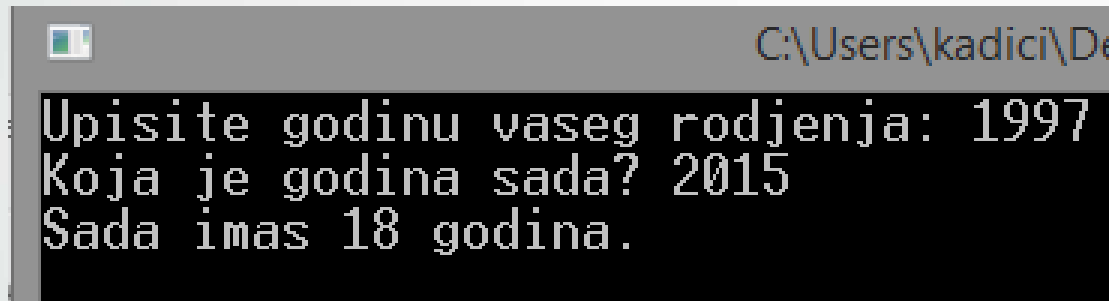
- Treba unijeti godinu rođenja i tekuću godinu. Računar treba da izračuna godine starosti.
- Ispis neka bude oblika:

```
Upisi godinu svog rođenja:  
Koja je godina sada?  
Sada imas .... godina.
```

Primjer 6

```
#include<iostream>
using namespace std;
int main() {
    int g_rodjenja, g_sada, g_starosti;
    cout<<"Upisite godinu vaseg rodjenja: ";
    cin>>g_rodjenja;
    cout<<"Koja je godina sada? ";
    cin>>g_sada;
    g_starosti=g_sada-g_rodjenja;
    cout<<"Sada imas "<<g_starosti<<" godina.\n";
    return 0;
}
```

Primjer 6



```
C:\Users\kadic\De
Upisite godinu vaseg rođenja: 1997
Koja je godina sada? 2015
Sada imas 18 godina.
```

- Provjera programa s podacima za koje je unaprijed poznat krajnji rezultat.

Komentari

- U datoteci *izvornog koda nekada je* korisno opisati što program radi, s ciljem ***lakšeg razumijevanja programa.***
- Takvi pomoćni opisi se nazivaju ***komentari.***
- Komentar služi samo programeru da bi objasnio pojedine dijelove programa i ***računar ga "ne razumije".***

Komentari

- To je tekst koji **započinje** s dvostrukom kosom crtom `//`, a završava **krajem reda**.
- Može biti napisan u **istom redu s naredbom** ili u **zasebnom redu**.

```
int god, god1, god2; //deklaracija cjelobrojnih varijabli
cout<<"Upisi godinu svog rođenja:";
//unos godine rođenja
cin>>god1;
```

- Komentari su obično prikazani **u drugoj boji**.

Komentari

- Pri prevođenju *izvornog koda* komentar se ***ne prevodi***.
- Osim za opis programa komentar se ponekad koristi ***za privremeno isključivanje dijelova izvornog koda***.

```
int X, A;  
cout<<"Unesi broj za operaciju ++A: ";  
cin>>A;  
//X=++A;  
//cout<<"Kad operacija završi X= "<<X<<endl;  
cout<<"Kad operacija završi A= "<<A<<endl;
```

Primjer 6b



- U prethodnom zadatku u izvornom kodu privremeno isključi red:

`g_starosti=g_sada-g_rodjenja;`

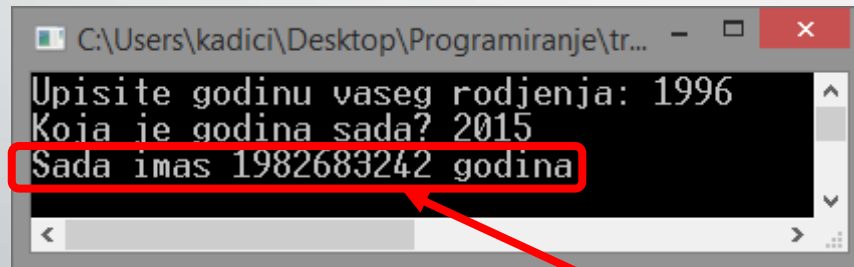
- Pokrenuti program pa uočiti posljedice.

Primjer 6b

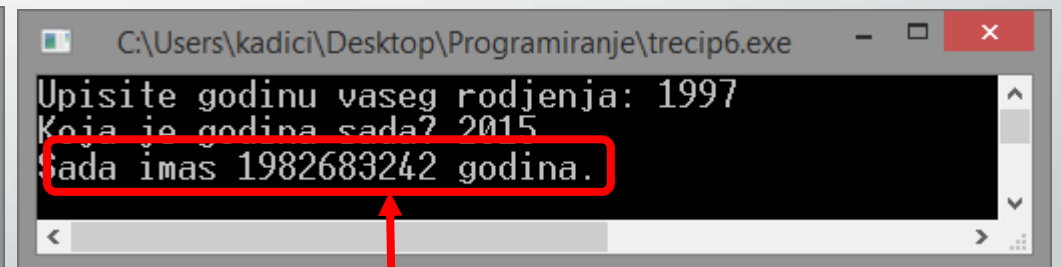
- Dakle izmjenjeni dio koda bi trebao izgledati:

```
cout<<"Koja je godina sada? ";  
cin>>g_sada;  
//g_starosti=g_sada-g_rodjenja;  
cout<<"Sada imas "<<g_starosti<<" god
```

- Posljedica je lako uočljiva, vrijednost varijable ***g_starosti*** nije izračunata.



```
C:\Users\kadic\\Desktop\Programiranje\tr...  
Upisite godinu vaseg rodjenja: 1996  
Koja je godina sada? 2015  
Sada imas 1982683242 godina
```



```
C:\Users\kadic\\Desktop\Programiranje\trecip6.exe  
Upisite godinu vaseg rodjenja: 1997  
Koja je godina sada? 2015  
Sada imas 1982683242 godina.
```

Operatori obnavljajućeg pridruživanja

- Omogućavaju *kraći zapis* nekih aritmetičkih izraza.
- Sastoje se od *znaka jednakosti* i odgovarajućeg *aritmetičkog operatora*.
- Npr. izraz:

$a=a+8,$

može se zapisati kao:

$a+=8.$

Operatori obnavljajućeg pridruživanja

izraz	operator obnavljajućeg pridruživanja
$a = a + \dots$	$a += \dots$
$a = a - \dots$	$a -= \dots$
$a = a * \dots$	$a *= \dots$
$a = a / \dots$	$a /= \dots$
$a = a \% \dots$	$a \% = \dots$

Primjer 7



- Potrebno je unijeti realni broj i pridružiti ga varijabli A.
- Sadržaj varijable A prvo uvećati za 5, pa umanjiti za 8, na kraju pomnožiti s 3.

(Koristiti operatore obnavljajućeg pridruživanja i varijablu A, ne uvoditi pomoćne varijable.)

Primjer 7

- Ispis neka bude oblika:

```
Upisi zeljeni broj:  
Sadržaj varijable A se uvecava za 5.  
Sada A iznosi: ....  
Od trenutnog sadržaja varijable A se  
oduzima 8. Sada A iznosi: ....  
Trenutni sadržaj varijable A se  
mnozi sa 3. Sada A iznosi: ....
```

Primjer 7

```
#include<iostream>
using namespace std;
int main()
{
    float A;
    cout<<"Upisi zeljeni broj:";
    cin>>A;
    A+=5;
    cout<<"Sadržaj varijable A se uvecava za 5. Sada A iznosi: "<<A<<endl;
    A-=8;
    cout<<"Od trenutnog sadržaja varijable A se oduzima 8. Sada A iznosi:"<<A<<endl;
    A*=3;
    cout<<"Trenutni sadržaj varijable A se množi sa 3. Sada A iznosi:"<<A<<endl;
    return 0;
}
```


Primjer 7

- Provjera programa s podacima:

```
Upisi zeljeni broj:25
Sadržaj varijable A se uvecava za 5. Sada A iznosi:30
Od trenutnog sadržaja varijable A se oduzima 8. Sada A iznosi:22
Trenutni sadržaj varijable A se mnozi sa 3. Sada A iznosi:66
```

Logički podaci (tip podataka – bool)

- Mogu poprimiti *samo jednu od dvije vrijednosti* (true/false, istina/laž, 1/0).
- **Varijabla** koja je pogodna za pohranu logičkog podatka je **tipa bool**.
- Takva varijabla može poprimiti vrijednosti **true** ili **false** (pri ispisu se true pretvara u **1**, a false u **0**).

Logički operatori

- Za rad s logičkim podacima, postoje *logičke funkcije*.
- Logičke se funkcije zapisuju *logičkim operatorima*.

simbol	funkcija
!	negacija (unarni operator: 1 pretvara u 0 i obratno)
&&	logički I - AND
	logički ILI – OR (Alt Gr + W)

Primjer 8



- Potrebno je unijeti dva logička podatka A i B. Treba ispisati: negaciju od A, vrijednost $(A \text{ I } B)$ te $(A \text{ ILI } B)$.
- Ispis neka bude oblika:

```
Vrijednost logickog podatka A=  
Vrijednost logickog podatka B=  
Ako je logicki podatak A= ... tada je  
negacija od A ....  
Za A= .... i B= ....  $(A \text{ I } B) = \dots$   
Za A= .... i B= ....  $(A \text{ ILI } B) = \dots$ 
```

Primjer 8

```
#include<iostream>
using namespace std;
int main()
{
    bool A, B, k, d, n;
    cout<<"Vrijednost logickog podatka A= ";
    cin>>A;
    cout<<"Vrijednost logickog podatka B= ";
    cin>>B;
    n=!A;
    k=A&&B;
    d=A||B;
    cout<<"Ako je logicki podatak A= "<<A<<" tada je negacija od A "<<n<<endl;
    cout<<"Za A= "<<A<<" i B= "<<B<<" (A i B)= "<<k<<endl;
    cout<<"Za A= "<<A<<" i B= "<<B<<" (A ili B)= "<<d<<endl;
    return 0;
}
```

Primjer 8

- Provjera programa s podacima za koje je unaprijed poznat krajnji rezultat.

```
C:\> "C:\Documents and Settings\Sanda\Desktop\slik\Debug\x.exe"  
Urijednost logickog podatka A=0  
Urijednost logickog podatka B=1  
Ako je logicki podatak A=0 tada je suprotno od A 1  
Za A=0 i B=1 (A I B)=0  
Za A=0 i B=1 (A ILI B)=1
```

Primjer 8 – moguća pogreška

- Ispravna ***bool varijabla*** može imati vrijednost samo **0** ili **1**.
- Kod nekih se kompajlera sve različito od 0 smatra podatkom true, a kod nekih ne.
- Da bi se izbjegla moguća pogreška treba koristiti ***samo vrijednosti 0 i 1***.

Poređenje podataka (relacijski operatori)

- Kada se dva podatka **porede**, *rezultat* je *podatak tipa bool*.

```
int a,b;
```

```
bool c;
```

```
c = a > b;
```

- Poredimo koristeći *operatore poredjenja*.

Operatori poređenja

simbol		funkcija
<	>	manje / veće
<=	=>	manje ili jednako / veće ili jednako
==		jednako
!=		različito

Primjer 9



- Potrebno je unijeti dva cijela broja. Nakon toga, treba ih usporediti pa ispisati rezultate poređenja.
- Ispis neka bude oblika:

```
Vrijednost prvog broja je  
Vrijednost drugog broja je  
Je li.... < .... odgovor: ....  
Je li.... > .... odgovor: ....  
Je li.... == .... odgovor: ....  
Je li.... != .... odgovor: ....
```

Primjer 9

```
#include<iostream>
using namespace std;
int main()
{
    int A, B;
    bool m, v, j, r;
    cout<<"Vrijednost prvog broja je A= ";
    cin>>A;
    cout<<"Vrijednost logickog podatka B= ";
    cin>>B;
    m=A<B;
    v=A>B;
    j=A==B;
    r=A!=B;
    cout<<"Je li "<<A<<"<"<<B<<" odgovor: "<<m<<endl;
    cout<<"Je li "<<A<<">"<<B<<" odgovor: "<<v<<endl;
    cout<<"Je li "<<A<<"=="<<B<<" odgovor: "<<j<<endl;
    cout<<"Je li "<<A<<"!="<<B<<" odgovor: "<<r<<endl;
    return 0;
}
```

Ovdje je moguće algoritam zapisati i korištenjem samo dvije varijable. Kako?

Primjer 9

- Provjera programa s podacima za koje je unaprijed poznat krajnji rezultat.

```

C:\
Urijednost prvog broja=234
Urijednost drugog broja=123
Da li je 234<123 odgovor: 0 ← false (neistina)
Da li je 234>123 odgovor: 1 ← true (istina)
Da li je 234==123 odgovor: 0 ← false (neistina)
Da li je 234!=123 odgovor: 1 ← true (istina)

```

String

- Za lakši rad sa **znakovnim nizovima** promjenjive dužine u standardnoj je biblioteci jezika C++ definiran tip podataka **string**.
- Tip podataka **string** "brine" o prostoru kojeg treba predvidjeti u memoriji, te **podržava funkcije** vezane uz znakovne nizove.

String

- Za rad s podacima tipa string, potrebno je uključiti ***biblioteku string*** (sadrži niz funkcija za rukovanje sa znakovnim nizovima).

```
#include<iostream>  
→ #include<string>  
using namespace std;
```

- Podatak tipa string može se pohraniti u ***varijablu deklariranu*** riječju ***string***, npr.:

```
string a;
```

Primjer 10



- Nadopuniti Primjer 7 tako da se unosi i ime korisnika programa.
- Ispis neka bude oblika:

```
Upisi svoje ime:  
Upisi godinu svog rođenja:  
Koja je godina sada?  
..... ima .... godina.
```

Primjer 10

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string ime;
    int g_rodjenja, g_sada;
    cout<<"Upisi svoje ime: ";
    cin>>ime;
    cout<<"Upisi godinu svog rodjenja: ";
    cin>>g_rodjenja;
    cout<<"Koja je godina sada? ";
    cin>>g_sada;
    cout<<ime<<" ima "<<g_sada-g_rodjenja<<" godina.";
    return 0;
}
```



```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    int god, god1, god2;
    string ime;
    cout<<"Upisi svoje ime:";
    cin>>ime;
    cout<<endl<<"Upisi godinu svog rođenja:";
    cin>>god1;
    cout<<endl<<"Koja je godina sada? ";
    cin>>god2;
    god=god2-god1;
    cout<<endl<<ime<<" ima"<<god<<"godina."<<endl;
    return 0;
}
```

Primjer 10

- Provjera programa s podacima za koje je unaprijed poznat krajnji rezultat.

```
C:\Documents and Settings\Sanda\De
Upisi svoje ime: Ivo
Upisi godinu svog rođenja: 1990
Koja je godina sada? 2004
Ivo ima 14 godina.
```