

**Zadatak 1: Za dva učitana broja sa tastature odrediti manji (t.j. odrediti  $\min(a,b)$ ).**

(**Napomena:** Ovaj se trik često koristi. Prvo postavimo da je jedan od brojeva manji u neku varijablu **minimum** a zatim provjerimo da li je drugi broj manji od **minimum** ako jeste drugi broj postaje minimum a ako nije ostavimo sve kako jeste. Ista se strategija primjenjuje za računanje maksimuma)

Testni primjeri:

```
D:\Skola\5_s_m_z\15-16\dodatna\struktura_grananja_zadaci\g21.exe
Unesite dva broja: 7
-9
Manji od brojeva 7 i -9 je -9
Process returned 0 (0x0)   execution time : 24.415 s
Press any key to continue.
```

```
D:\Skola\5_s_m_z\15-16\dodatna\struktura_grananja_zadaci\g21.exe
Unesite dva broja: -14
5
Manji od brojeva -14 i 5 je -14
Process returned 0 (0x0)   execution time : 6.364 s
Press any key to continue.
```

**Zadatak 2: Za učitane a i b izračunati z prema izrazu  $z := \max(a^2, b^2)$ .**

(**Napomena:** Isto kao u prethodnom primjeru. Samo treba izračunati kvadrate.)

Testni primjer:

```
D:\Skola\5_s_m_z\15-16\dodatna\struktura_grananja_zadaci\g22.exe
Unesite dva broja: -2.45 3
Veci od dva kvadrata unesenih brojeva -2.45 i 3 je 9
Process returned 0 (0x0)   execution time : 10.900 s
Press any key to continue.
```

**Zadatak 3: Za učitane a i b izračunati z prema izrazu  $z = \frac{1-\min(a,b)}{1+\max(a^2,b^2)}$ .**

(**Napomena:** Koristite prethodne zadatke)

Testni primjeri:

```
D:\Skola\5_s_m_z\15-16\dodatna\struktura_grananja_zadaci\g23.exe
Unesite dva broja: 1
9
Za unesene vrijednosti 1 i 9 vrijednost z je 0
Process returned 0 (0x0)   execution time : 2.335 s
Press any key to continue.
```

```
D:\Skola\5_s_m_z\15-16\dodatna\struktura_grananja_zadaci\g23.exe
Unesite dva broja: -1
3
Za unesene vrijednosti -1 i 3 vrijednost z je 0.2
Process returned 0 (0x0)   execution time : 5.031 s
Press any key to continue.
```

**Zadatak 4:** Potrebno je unijeti koeficijente linearne jednačine  $kx + n = 0$ . U ovisnosti od koeficijenata  $k$  i  $n$  izračunati i ispisati rješenja jednačine.

(**Napomene:** Zadatak treba riješiti koristeći višestruki uvjetnog grananja:

- Ako je  $k=0$  i  $n=0$  jednačina ima beskonačno rješenja.
- Ako je samo  $k=0$ , to linearna jednačina nema rješenja.
- Za ostale vrijednosti koeficijenata,  $x=-n/k$ .

Testni primjeri:

```
D:\Skola\5_s_m_15-16\dodatna_struktura_grananja_zadaci\sg24.exe
Unesite koeficijent smjera k: 0
Unesite koeficijent n: 0
Ako je k=0 i n=0. Jednacina 0*x=0 ima beskonacno rjesenja!
Process returned 0 (0x0)   execution time : 4.165 s
Press any key to continue.
```

```
D:\Skola\5_s_m_15-16\dodatna_struktura_grananja_zadaci\sg24.exe
Unesite koeficijent smjera k: 0
Unesite koeficijent n: 5
Ako je k!=0 i n=0. Jednacina 0*x=5 nema rjesenja!
Process returned 0 (0x0)   execution time : 5.381 s
Press any key to continue.
```

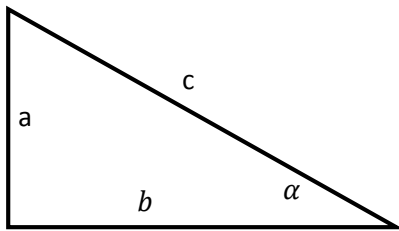
**Zadatak 5:** Napraviti program za rješavanje kvadratne jednačine. Korisnik treba da unese koeficijente kvadratne jednačine. Ako postoje realna rješenja program treba da ih ispiše a ako ne postoje treba da vrati poruku da su rješenja kompleksna.

(**Napomena:** Iz matematike znamo da je opšti oblik kvadratne jednačine dat sa  $ax^2 + bx + c = 0$  kao i da rješenja kvadratne jednačine zavise od diskriminante  $D = b^2 - 4ac$ . Ako je diskriminanta  $D$  veća ili jednaka od 0 tada su rješenja reala i dobijamo ih po formuli  $x_{1,2} = \frac{-b \pm \sqrt{D}}{2a}$ . Naravno mi u c++ moramo računati jedno po jedno rješenje recimo za  $x_1$  bi bilo:  $x_1 = (-b + \text{sqrt}(D))/2 * a$  rješenje  $x_2$  bi bilo slično samo bi ispred korijena imali minus. Ako je  $D$  manje od nule samo treba ispisati sa su rješenja kompleksna)

```
D:\Skola\5_s_m_15-16\dodatna_struktura_grananja_zadaci\sg25.exe
Unesite koeficijente kvadratne jednacine:
a=1
b=-5
c=6
Rjesenja jednacine su
x1=3
x2=2
Process returned 0 (0x0)   execution time : 10.342 s
Press any key to continue.
```

```
D:\Skola\5_s_m_15-16\dodatna_struktura_grananja_zadaci\sg25.exe
Unesite koeficijente kvadratne jednacine:
a=3
b=2
c=1
Rjesenja kvadratne jednacine su kompleksna!
Process returned 0 (0x0)   execution time : 4.705 s
Press any key to continue.
```

**Zadatak 6: U pravouglom trouglu je poznat ugao  $\alpha$  i stranica a. Treba odrediti hipotenuzu c.**



(Napomena: Iz oblasti trigonometrije znamo da je  $\sin \alpha = \frac{a}{c}$ .

Odavde možemo odrediti hipotenuzu c. Međutim ugao je dan u stepenima a znamo da u c++ ako primjenjujemo funkciju **sin** ili bilo koju drugu trig. funkciju moramo koristiti uglove izražene radijanima a za to imamo formulu konverzije uglova:

$$\alpha_{rd} = \frac{\pi}{180^\circ} \alpha_{st}$$

Testni primjeri:

```

D:\Skola\5_s_m_15-16\dodatna\struktura_grananja_zadaci\q26.exe
Unesite stranicu a pravouglog trougla:
a=5
Unesite ugao alfa:
45
Ako je vrijednost ugla alfa 45 stepeni i stranica a=5cm hipotenuza ima vrijednos
t c=7.07107cm.
Process returned 0 (0x0)   execution time : 13.055 s
Press any key to continue.
    
```

```

D:\Skola\5_s_m_15-16\dodatna\struktura_grananja_zadaci\q26.exe
Unesite stranicu a pravouglog trougla:
a=9
Unesite ugao alfa:
30
Ako je vrijednost ugla alfa 30 stepeni i stranica a=9cm hipotenuza ima vrijednos
t c=18cm.
Process returned 0 (0x0)   execution time : 6.459 s
Press any key to continue.
    
```

(Malo da se prisjetimo linearnog programiranja. Primjetimo da nije bilo strukture grananja)

**Zadatak 7: Treba unijeti dva realna broja pa izračunati i ispisati njihov cjelobrojni koeficijent i ostatak dijeljenja. Potrebno je zapisati izraz za provjeru pa provjeriti rezultat. Ako je drugi uneseni broj nula treba ispisat upozorenje „Djeljenje nulom“ i prekinuti program.**

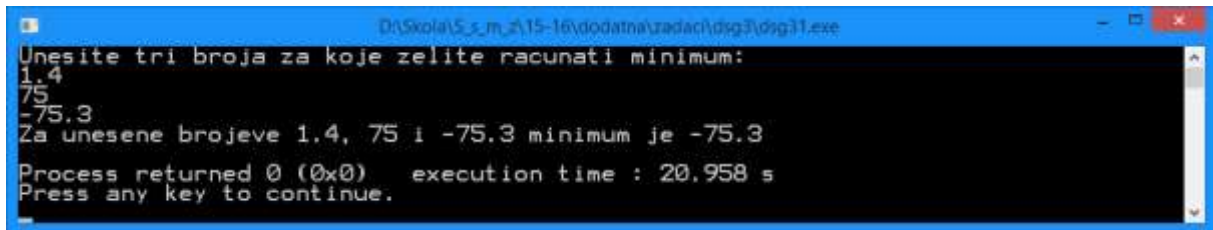
```

#include<iostream>
#include<cmath>
using namespace std;
int main(){
    double a,b,c,o;
    cout<<"Unesite djeljenik:\n";
    cin>>a;
    cout<<"Unesite djelitelj:\n";
    cin>>b;
    if(b){
        c=floor(a/b);
        o=fmod(a,b);
        cout<<a<<"/"<<b<<"= "<<c<<" cijelih i "<<o<<"ostatka"<<endl;
        cout<<"Provjera: ("<<b<<"*"<<c<<")+ "<<o<<"="<<a<<endl;
    }
    else
        cout<<"Djeljenje nulom!";
    return 0;
}
    
```

(**Napomena:** Primjetimo da djelimo realne brojeve a ne cijele što znači da za cijeli dio realnog broja treba koristiti funkciju **floor** za koju smo vidjeli da zaokružuje realan broj na prvi manji cijeli broj kao i funkciju **fmod** koja vraća ostatak djeljenja realnih brojeva. Ako je drugi broj koji korisnik unese 0 program treba da ispiše da je to djeljenje sa 0 )

**Zadatak 8:** Napraviti program koji za unesena tri realna broja računa koji je najmanji tj.  $\min(a,b,c)$ .

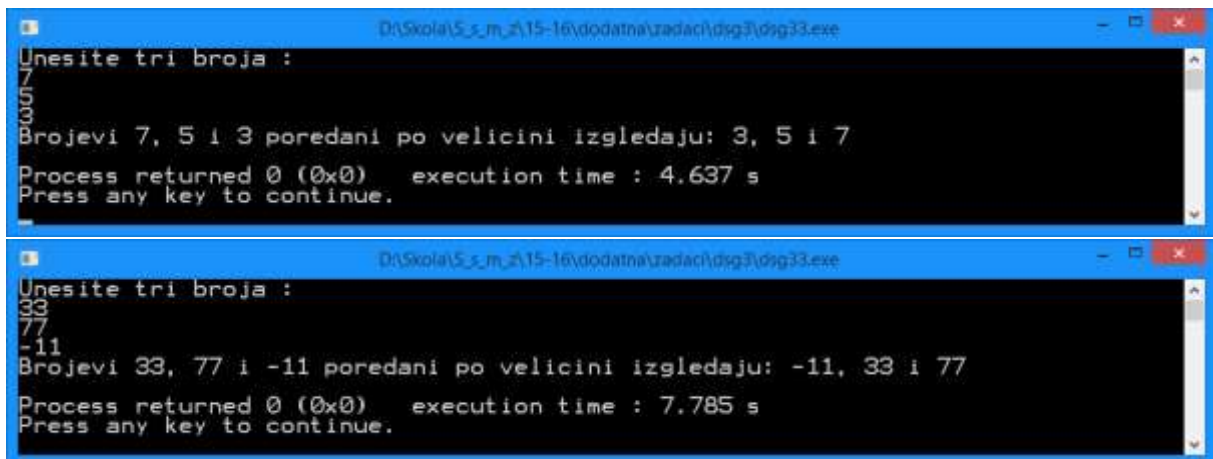
Testni primjer:



```
D:\Skola\5_3_m_2\15-16\dodatna\zadaci\dig3\dig33.exe
Unesite tri broja za koje zelite racunati minimum:
1.4
75
-75.3
Za unesene brojeve 1.4, 75 i -75.3 minimum je -75.3
Process returned 0 (0x0)   execution time : 20.958 s
Press any key to continue.
```

**Zadatak 9:** Treba unijeti tri realna broja pa ih ispisati od najmanjeg ka najvećem.

(**Napomena:** Iz prethodnog primjera imamo način da odredimo mini (skoro isto odredimo i maxi samo se koristi > umjesto <). Srednji broj odrediti kao broj koji je niti mini niti maxi)



```
D:\Skola\5_3_m_2\15-16\dodatna\zadaci\dig3\dig33.exe
Unesite tri broja :
7
5
3
Brojevi 7, 5 i 3 poredani po velicini izgledaju: 3, 5 i 7
Process returned 0 (0x0)   execution time : 4.637 s
Press any key to continue.

D:\Skola\5_3_m_2\15-16\dodatna\zadaci\dig3\dig33.exe
Unesite tri broja :
33
77
-11
Brojevi 33, 77 i -11 poredani po velicini izgledaju: -11, 33 i 77
Process returned 0 (0x0)   execution time : 7.785 s
Press any key to continue.
```

**Zadatak 10:** Napisati program koji na ulazu prihvata punu cijenu karte, kao i godine starosti putnika koji želi da putuje, a na izlazu daje cijenu karte koju putnik treba da plati.

(**Napomena:** Koristite if- else if – else strukturu grananja)

The image shows three screenshots of a C++ program execution. Each screenshot displays the program's output for different user inputs. The first screenshot shows an input of 10 KM and 15 years, resulting in a 50% discount to 5 KM. The second screenshot shows an input of 25 KM and 45 years, resulting in a full price of 25 KM. The third screenshot shows an input of 10 KM and 84 years, resulting in a free ticket.

```

D:\Skola\5_s_m_2\15-16\Dodatna\zadaci\big7\big32.exe
Unesite punu cijenu karte u KM:
10
Unesite vasu starost:
15
Placate pola cijene, tj. 5KM.
Process returned 0 (0x0)   execution time : 29.985 s
Press any key to continue.

D:\Skola\5_s_m_2\15-16\Dodatna\zadaci\big7\big32.exe
Unesite punu cijenu karte u KM:
25
Unesite vasu starost:
45
Placate punu cijenu, tj. 25KM.
Process returned 0 (0x0)   execution time : 17.070 s
Press any key to continue.

D:\Skola\5_s_m_2\15-16\Dodatna\zadaci\big7\big32.exe
Unesite punu cijenu karte u KM:
10
Unesite vasu starost:
84
Mozete putovati besplatno!
Process returned 0 (0x0)   execution time : 16.881 s
Press any key to continue.

```

Interesantno je da se kao uvjet unutar naredbe “if” može upotrijebiti *objekat ulaznog toka* “cin”. U tom slučaju se smatra da je objekat ulaznog toka “tačan” ukoliko je tok u *ispravnom stanju*, a “netačan” ukoliko je tok u *neispravnom stanju*. Ulazni tok može dospjeti u neispravno stanje na primjer u slučaju da se očekuje unos broja sa tastature, a umjesto broja korisnik unese znakove koji ne predstavljaju broj. Ovo je ilustrirano sljedećim primjerom:

**Primjer 1: Napraviti program koji od korisnika traži da unese broj a zatim provjerava da li je broj zaista unesen. Ako je unesen znak to treba da ispiše.**

```

int broj;
cout << "Unesite cijeli broj: ";
cin >> broj;
if(cin)
    cout << "Unijeli ste broj " << broj << endl;
else
    cout << "Varate, niste uopće unijeli broj!\n";

```

Također, nad objektom ulaznog toka može se primijeniti i operator negacije “!”. Tako je izraz “!cin” tačan ukoliko je tok u neispravnom stanju, a netačan u suprotnom. Stoga smo prethodni primjer mogli napisati i ovako:

```

int broj;
cout << "Unesite cijeli broj: ";
cin >> broj;
if(!cin) cout << "Varate, niste uopće unijeli broj!\n";
else cout << "Unijeli ste broj " << broj << endl;

```

O testiranju ulaznog toka na ispravnost govorićemo detaljnije kada upoznamo naredbe ponavljanja, koje će omogućiti da u slučaju neispravnog unosa zahtijevamo od korisnika da *ponovi unos*.

**Zadatak 11: Napraviti program koji na osnovu unesenog rednog broja mjeseca ispisuje koliko taj mjesec ima dana. U slučaju unosa broja koji nije iz raspona od 1 do 12 treba ispisati upozorenje.**

(Napomena: Prisjetimo se da 1.,3.,5.,7.,8.,10. i 12. mjesec imaju po 31 dan dok 2,4,6,9 i 11 imaju po 30 dana naravno 2. mjesec ima 28 (29 ako je prestupna) dana. Koristiti isključivo naredbu switch case)

```
D:\Skola\5_s_m_2\15-16\dodatna\zadaci\093\09g34.exe
Unesite redni broj mjeseca :
7
Mjesec ima: 31 dan
Process returned 0 (0x0)   execution time : 2.820 s
Press any key to continue.
```

```
D:\Skola\5_s_m_2\15-16\dodatna\zadaci\093\09g34.exe
Unesite redni broj mjeseca :
11
Mjesec ima: 30 dana
Process returned 0 (0x0)   execution time : 4.890 s
Press any key to continue.
```

```
D:\Skola\5_s_m_2\15-16\dodatna\zadaci\093\09g34.exe
Unesite redni broj mjeseca :
2
Mjesec ima: 28 dana (ili 29 ako je godina prestupna)
Process returned 0 (0x0)   execution time : 1.226 s
Press any key to continue.
```

**Primjer 2: Napisati program koji simulira jednostavan kalkulator, koji omogućava korisniku da unese broj, operator “+”, “-”, “\*” ili “/”, i drugi broj (npr. “2+3”). Program tada računa i prikazuje rezultat tražene operacije nad traženim operandima:**

```
#include <iostream>
using namespace std;
int main() {
double prvi, drugi;
char operacija;
cin << prvi << operacija << drugi;
switch(operacija) {
case '+':
cout << prvi + drugi << endl; break;
case '-':
cout << prvi - drugi << endl; break;
case '*':
cout << prvi * drugi << endl; break;
case '/':
cout << prvi / drugi << endl; break;
default:
cout << "Nedozvoljena operacija!\n ";
}
return 0;
}
```

Da bismo shvatili kako radi ovaj program, moramo se podsjetiti da se izdvajanje broja pomoću operatora “>>” iz ulaznog toka prekida na prvom znak koji sigurno ne pripada broju. Tako, ako zadamo naredbu poput

```
cin >> prvi;
```

a sa tastature unesemo nešto poput “321\*443”, tada će u promjenljivu “prvi” biti smješten broj “321”, jer se izdvajanje pomoću operatora “>>” prekida na znaku “\*” koji nije sastavni dio broja.

Druga činjenica bitna za razumijevanje je da se izdvajanje sljedećeg elementa nastavlja na mjestu gdje sezavršilo izdvajanje prethodnog elementa. Zbog toga će nakon naredbe

```
cin >> prvi >> operacija >> drugi;
```

za slučaj da je unos sa tastature bio također “321\*443”, u promjenljivu “prvi” biti smješten broj “321”, u promjenljivu “operacija” znak ‘+’ (zapravo njegova ASCII šifra ako hoćemo da budemo posve precizni), a u promjenljivu “drugi” broj “443”. Uz ovo objašnjenje ostatak programa je dovoljno jasan.