

Samoevaluacijski test

Da biste znali da li ste spremni za predmet "Programiranje 1" uradite slijedećih 10 zadataka. Zadaci su obrađeni u sklopu workshopa 1 do 15 (Uvod u programiranje).

Zadatak 1:

Napravite funkciju `void ispisi_rjesenja(float, float, float)` koja na osnovu tri ulazna parametra a , b , c treba ispisati na ekran rješenja (x_1 i x_2) kvadratne funkcije ($ax^2+bx+c=0$). U funkciji `main` zahtijevati od korisnika unos članova kvadratne jednačine (a , b , c).

U slučaju da kvadratna jednačina postane linearna (za $a=0$), nemojte ispisati rješenja.

Pomoć:

- funkciji izračunajte vrijednost diskriminante $D = b^2 - 4ac$.
- na osnovu vrijednosti D i vrijednosti a možete zaključiti da li kvadratna jednačina ima rješenja:

$D \geq 0$		$D < 0$
$a \neq 0$	$a = 0$	Nema realnih rješenja
Rješenja kvadratne jednadžine: $x_1 = \frac{-b + \sqrt{D}}{2a}$ $x_2 = \frac{-b - \sqrt{D}}{2a}$	Postaje linearna jednačina: $bx + c = 0 \Rightarrow x = -\frac{c}{b}$ ali u ovom slučaju ćemo ispisati da nema rješenja	

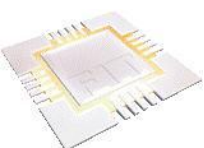
- uslov za računanje rješenja x_1 i x_2 glasi: $(D \geq 0 \ \&\& \ a \neq 0)$
- testirajte program, unesite barem jednu negativnu vrijednost:
- za unesene vrijednosti **1,3,2** program treba ispisati:

$x_1 = -2$
 $x_2 = -1$

- za unesene vrijednosti 0,8,2 i 1,2,3 program treba ispisati:

Greska: Nije moguće izračunati rješenja

Rješenje se nalazi u dokumentu Workshop 13 – zadatak br. 137 – UPR.



Zadatak 2:

Kad bi htjeli napraviti funkciju sa povratnom vrijednošću za računanje rješenja kvadratne jednačine (samo računanje a ne ispisivanje) imali bi poteškoća. Problem je u tome što kvadratna jednačina ima dva rješenja. Funkcije u C++ mogu imati samo jednu povratnu vrijednost. (Povratna vrijednost predstavlja izlaz iz funkcije pomoću naredbe `return`, a ne izlaz pomoću reference ili sl.)

U tom slučaju mogli bi napraviti dvije slične funkcije, jednu koja računa rješenje kvadratne jednačine **x1** a drugu za **x2**.

Napravite program za računanje rješenja kvadratne jednačine, tako da koristite funkcije `fx1` i `fx2` (funkcija `fx1` računa **x1**, a funkcija `fx2` računa **x2**).

Rješenje se nalazi u dokumentu Workshop 13 – zadatak br. 138 – UPR.

Ove funkcije ne bi trebale ispisivati rješenja na ekran, nego rješenja trebaju biti izlaz iz funkcije, a funkcija `main` bi trebala ispisivati rješenja kvadratne jednačine **x1** i **x2**. Ovdje nam se javlja još jedan matematički problem: Šta ako jednačina nema rješenja? (npr. za vrijednosti 0,8,2 i 1,2,3)

Kako će nam funkcija koja ne ispisuje ništa, koja samo vraća gotovu vrijednost (broj), javiti da ne postoji rješenje. Taj problem ne možemo lahko riješiti pa ćemo se dogovoriti, u slučaju da ne postoje rješenja, da funkcija ispiše poruku o grešci i vrati vrijednost 0, jer ova funkcija mora vratiti (kroz `return`) neki broj – isto kao sve matematičke funkcije (`sqrt`, `pow`, `sin`, ...).

Zadatak 3:

Riješite prethodni zadatak tako da koristite jednu funkciju sa pozivom po referenci.

Rješenje se nalazi u dokumentu Workshop 13 – zadatak br. 139 – UPR.

Zadatak 4:

Isprobajte sljedeći program. Šta će se ispisati i zašto?

```
4: void main()
5: {
6:     float f = 5/3;
7:     cout << "5/3 = " << f << endl;
8: }
```

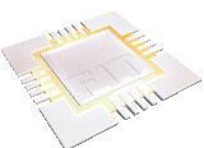
Rješenje se nalazi na stranici 5.

Zadatak 5:

Šta će ispisati sljedeći program? Obrazložite! Da li je sljedeći program ispravan? Ako nije, ispravite greške!

```
4: void main()
5: {
6:     cout << "@ + # = " << char('@' + '#') << endl;
7:     cout << "Ovaj program zna racunati!" << char(2) << endl;
8: }
```

Rješenje se nalazi na stranici 5.



Zadatak 6 :

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).

a) *Workshop 6 - zadatak br. 45 – UPR:*

Program riješite pomoću dva *if-else*-iskaza (dvostruki izbor), koristeći uslove $(i \geq 1)$ i $(i \leq 1600)$.

b) *Workshop 6 - zadatak br. 47 – UPR:*

Program riješite pomoću dva *if-else*-iskaza (dvostruki izbor), koristeći uslove $(i \leq 0)$ i $(i \geq 1601)$.

c) *Workshop 6 - zadatak br. 48 – UPR:*

Program riješite pomoću jednog *if-else*-iskaza (dvostruki izbor), koristeći uslove $(i \leq 0)$ i $(i \geq 1601)$. Koristite neke od logičkih operatora (AND, OR i NOT).

d) *Workshop 6 - zadatak br. 49a – UPR:*

Program riješite pomoću dva *if*-iskaza (jednostruki izbor). Koristite neke od logičkih operatora (AND, OR i NOT).

e) *Workshop 6 - zadatak br. 49c – UPR:*

Program riješite pomoću dva *if*-iskaza (jednostruki izbor), bez korištenja logičkog operatora NOT.

Koristite dijagram toka kao pomoć.

Zadatak 7:

...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:

1. 114
2. -99 do -33
3. brojevi djeljivi sa 19

Koristite što više varijabli tipa `bool`.

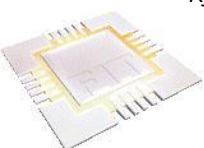
Rješenje se nalazi u na stranici 5.

Pomoć za ovaj zadatak pogledajte u dokumentu *Workshop 6 – zadatak br. 56 – UPR*.

Zadatak 8:

Napravite program koji će pitati korisnika koliko želi unijeti brojeva. Zatim program treba zahtijevati unos toliko cijelih brojeva. Nakon što je korisnik unio sve te brojeve, program treba ispisati najveći uneseni broj.

Rješenje se nalazi u dokumentu *Workshop 9 – zadatak br. 97 – UPR*.



Zadatak 9:

Zašto će sljedeći kompajler prijaviti u liniji br. 14 grešku?

```
1:  #include <iostream>
2:  using namespace std;
3:
4:  void main()
5:  {
6:      if (10 > 1)
7:      {
8:          int b = 4;
9:      }
10:     else
11:     {
12:         int b = 5;
13:     }
14:     cout << "b = " << b << endl; //greska
15: }
```

Rješenje se nalazi u dokumentu Workshop 10 – str. 93 – UPR.

Zadatak 10:

a) Napravite prototip i definiciju funkcije `uvecaj_r` i testirajte je u sljedećem programu:

```
1:  void main()
2:  {
3:      int x;
4:      x = 5;
5:
6:      uvecaj_r(x);          // vrijednost varijable x će se povećati sa 5 na 6
7:      uvecaj_r(x);          // vrijednost varijable x će se povećati sa 6 na 7
8:
9:      cout << x << endl;    // ispisat će se vrijednost 7
10: }
```

Rješenje se nalazi u dokumentu Workshop 13 – str. 140 – UPR.

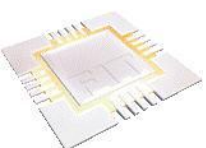
b) Prepravite samo linije br. 6 i 7 iz prethodne funkciju `main` tako da koristite datu funkciju `uvecaj_v` umjesto funkcije `uvecaj_r`:

```
1:  int uvecaj_v(int a)
2:  {
3:      return a+1;
4:  }
5:
```

Rješenje se nalazi u dokumentu Workshop 13 – str. 140 – UPR.

Za prolaz je potrebo uraditi 100%.

Ako ste uspješno položili test spremni ste za predmet "Programiranje 1".



Rješenja

Rješenje zadatka br. 4:

Količnik $5/3$ predstavlja broj 1 (jer se radi o dijeljenju: **int** kroz **int**), tako da se varijabli ϵ dodjeljuje broj 1, bez obzira što je varijabla ϵ tipa *float*.

Za detaljnije objašnjene pročitajte dokument Workshop 2 - str. 12 (*Uvod u programiranje*).

Rješenje zadatka br. 5:

```
6:    cout << "@ + # = " << char('@' + '#') << endl;
```

Operator '+' odnosi na brojčane podatke a ne na podatke tipa `char`. Stoga će se vrijednost karaktera pretvoriti u brojčanu vrijednost kao da koristimo naredbe `int('@')` i `int('#')`:

```
int('@') = 64    (ASCII broj znaka '@' je 64)
```

```
int('#') = 35
```

```
'@' + '#' = 99
```

Ispisat će se slovo 'c', jer ono ASCII karakter za broj 99.

Više sličnih zadataka možete naći u dokumentu Workshop 3 (*Uvod u programiranje*).

Rješenje zadatka br. 7:

```
1:    #include <iostream>
2:    using namespace std;
3:
4:    void main()
5:    {
6:        int i;
7:        bool kontrola, niz1, niz2, zlatni; //U Borland C++-u morat će koristiti int umjesto bool
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:    }
```

