

Zadaci za samostalan rad

Uz 1. i 2. tutorijal

Prvu verziju pripremili: Enil Pajić, Ivona Ivković

Ovi zadaci su namijenjeni da studenti utvrde gradivo prva dva tutorijala i steknu potrebne vještine i programersku logiku za rješavanje kompleksnijih programerskih problema koji će biti tematika u nastavku kursa. Zadaci su ciljano pisani da obuhvate što je više moguće aspekata predenog gradiva te je jako korisno da ih studenti samostalno prerade. Dijelom je obuhvaćena tematika koja će se kasnije koristiti kao dio zadatka (npr. pretvaranje sekundi u sate, minute i sekunde).

Također, postoje zadaci tipa „šta će program ispisati?“ gdje studenti trebaju da pažljivo analiziraju kôd i predvide šta će program ispisati. Ispisi su fundament razumijevanja rada programa i u ispisima se „kriju“ neki trikovi zbog kojih se dešavaju najčešće greške koje pravimo.

Nastavni ansambl toplo preporučuje preradu što više ovih zadataka i obeshrabruje mišljenja tipa „znam ja ovo“, jer se uvijek nađe neka sitnica koju ne znate. Ako znate, svejedno uradite zadatke i dokažite sebi da znate.

#1 Napisati program koji „precizno“ računa 4. korijen unesenog broja. Potrebno je osigurati preciznost na preko 12 decimalnih mjesta. Korisnik unosi broj, program računa 4. korijen i ispisuje ga na 10 decimalnih mjesta.

Primjer ulaza i izlaza:

```
Unesite broj: 123456
4. korijen iz 123456 je 18.7446808481
```

Savjet: koristiti `double` tip podataka, a za izračunavanje 4. korijena koristiti dva sukcesivna poziva funkcije za računanje običnog (drugog) korijena.

#2 Napisati program koji od korisnika traži unos ocjena iz 5 predmeta te računa prosjek tih ocjena. Ocjene su u intervalu [6, 10] i to nije potrebno provjeravati. Potrebno je ispisati zbir unesenih ocjena i prosjek zaokružen na 2 decimalna mjesta.

Primjer ulaza i izlaza:

```
Unesite 5. ocjena: 6 6 7 9 8
Zbir unesenih ocjena je 36, a prosjek je: 7.20
```

#3 Napisati program koji računa stepene unesenog (realnog) broja i ispisuje ih tako da je je prvih 5 stepena poravnato udesno, a drugih 5 ulijevo. Program računa

stepene broja (do 10. stepena) i ispisuje ih na 20 mjesta širine (ne računajući „^“ prefiks), zaokruženo na 5 decimalnih mjesta, poravnatih kako je iznad opisano.

Primjer ulaza i izlaza:

```
Unesite broj: 17.777
^ 1:          17.77700
^ 2:          316.02173
^ 3:          5617.91828
^ 4:          99869.73320
^ 5:         1775384.24710
^ 6: 31561005.76068
^ 7: 561059999.40761
^ 8: 9973963609.46904
^ 9: 177307151085.53107
^10: 3151989224847.48584
```

#4 Napisati program koji će pitati korisnika da unese neki broj u rasponu od 0 do 9. Od korisnika se očekuje da unese jednocifren prirodan broj, tako da vi ne trebate vršiti provjeru. Nakon toga program treba ispisati taj broj u sredini okružen zvjezdicama (5 zvjezdica u širini, 5 u visini osim u srednjem redu). Primjer ispisa za uneseni broj 7:

```
Unesite prirodan broj u intervalu [0, 9]: 7
*****
*****
* 7 *
*****
*****
```

#5 Prepraviti prethodni zadatak tako da korisnik može birati koji broj će biti ispisan u sredini, ali isto tako da može birati znak kojim će taj broj biti okružen. Primjer ispisa za broj 7 i znak "=" :

```
Unesite znak kojim ce broj biti okruzen: =
Unesite prirodan broj u intervalu [0, 9]: 7
=====
=====
= 7 =
=====
=====
```

Savjet: uneseni znak smjestiti u varijablu znakovnog tipa (tj. tipa `char`), unositi ga i ispisivati pomoću „%c“ formatnog stringa. Također, jako je važno da se prvo unosi znak pa onda broj (a ne obrnuto, broj pa znak), jer će znak novi red (kojeg „unesemo“

kada pritisnemo tipku ENTER prilikom unosa broja) biti unesen u varijablu pa nas program uopšte neće pitati za unos znaka.

Za razmisliti: kako riješiti ovaj problem? Uvesti novu varijablu u koju ćemo smjestiti novi znak novi red („\n“ znak)? Poziv `getchar()` funkcije? Ili neki drugi pristup?

#6 Napišite program koji računa vrijednost korijena diskriminante $K = \sqrt{b^2 - 4ac}$. Unose se koeficijenti a , b i c . Također se pretpostavlja da su svi uneseni koeficijenti pozitivni (nije potrebno provjeravati) i da diskriminanta ima rješenje u skupu realnih brojeva (drugim riječima pod korijenom se neće naći negativan broj, tj. pretpostavimo da uvijek vrijedi $b^2 \geq 4ac$). Primjeri unosa i ispisa:

```
Unesite koeficijente a, b i c: 2 10 4
Iznos za koeficijente 2, 10 i 4 je: 8.24621
```

```
Unesite koeficijente a, b i c: 2.03 15.7 3
Iznos za koeficijente 2.03, 15.7 i 3 je: 14.904
```

Primijetite kako su „fino“ zaokruženi brojevi u ispisu, npr. ako smo unijeli cijele brojeve (primjer 1), onda su oni ispisani kao cijeli brojevi, bez decimala a rezultat je zaokružen na 5 decimalnih mjesta. U drugom primjeru smo miješali unos cijelih i realnih brojeva, i ispis je bio upravo takav, dok je rezultat zaokružen na 3 mjesta. Potrebno je da program ima upravo ovakav ispis gdje se brojevi „pametno“ zaokružuju u zavisnosti od unosa.

#7 Prepraviti prethodni program tako da je ovaj put potrebno izvršiti provjeru vrijednosti diskriminante. Ako je vrijednost diskriminante nenegativna, program ima ispis identičan kao u prethodnom zadatku, u suprotnom (ako je vrijednost diskriminante negativna), potrebno je ispisati „*Korijen negativnog broja nema realnih rjesenja!*“.

Primjeri ispisa:

```
Unesite koeficijente a, b i c: 7 15.3 2.11
Iznos za koeficijente 7, 15.3 i 2.11 je: 13.2291
```

```
Unesite koeficijente a, b i c: 2 5.4 14
Korijen negativnog broja nema realnih rjesenja!
```

#8 Napisati program koji (okvirno) računa koliko osoba ima godina, te dodatno klasificira osobu u neku od sljedećih kategorija „prije škole“ (do 6 godina), „osnovna škola“ (od 6 do 14 godina), „srednja škola“ (od 14 do 18 godina), „fakultet“ (od 18 do 23 godine i „ostalo“ (stariji od 23 godine).

Korisnik unosi godinu rođenja osobe i njen spol (slovo „m“ ili „M“ ako je osoba muškog spola, slovo „z“ ili „Z“ ako je osoba ženskog spola; u slučaju je uneseno nešto osim ova 4 znaka, program treba ispisati „Pogresan unos spola osobe!“ i završiti sa radom, pri čemu ćemo zanemariti pojmove „oba spola“ ili „srednjeg“ roda).

Pomoć: potrebno je uvesti varijablu (ili simboličku konstantu pomoću `#define`) koja označava trenutnu godinu i inicijalizirati je na 2015.

Ako je kao godina rođenja unesena godina veća od trenutne godine (npr. veća od 2015), onda program treba ispisati „Neispavna godina rođenja!“ i prekinuti izvršavanje.

Program treba da izračuna koliko osoba ima godina, da je razvrsta u neku od kategorija i dopiše „musko“ ili „zensko“ u zavisnosti od unesenog spola. Za više detalja, pogledati ispis:

```
Unesite godinu rodjenja osobe: 1993
Unesite spol osobe (m/M ili z/Z): M
Osoba ima 22 godina/e/u, klasificirana je u kategoriju:
'fakultet', musko je.
```

```
Unesite godinu rodjenja osobe: 1997
Unesite spol osobe (m/M ili z/Z): m
Osoba ima 18 godina/e/u, klasificirana je u kategoriju:
'fakultet', musko je.
```

```
Unesite godinu rodjenja osobe: 2000
Unesite spol osobe (m/M ili z/Z): z
Osoba ima 15 godina/e/u, klasificirana je u kategoriju:
'srednja skola', zensko je.
```

```
Unesite godinu rodjenja osobe: 1953
Unesite spol osobe (m/M ili z/Z): Z
Osoba ima 62 godina/e/u, klasificirana je u kategoriju:
'ostalo', zensko je.
```

```
Unesite godinu rodjenja osobe: 1995
Unesite spol osobe (m/M ili z/Z): S
Pogresan unos spola osobe!
```

```
Unesite godinu rodjenja osobe: 2017
Neispravna godina rodjenja!
```

#9 Napisati program koji od korisnika traži unos 10 prirodnih brojeva, te parne brojeve ispisi desno poravnate, a neparne lijevo poravnate. Ispis je potrebno vršiti na 15 mjesta širine, pri čemu kod parnih brojeva prazna mjesta treba ispuniti nulama. Primjer ispisa:

Unesite 10 prirodnih brojeva: 102 7374 0 123 -14 727 7349272 73
111 82824
000000000000102
0000000000007374
0000000000000000
123
-000000000000014
727
000000007349272
73
111
000000000082824

Savjet: parnost broja ispitivati pomoću modulo operatora (operator %), koji daje ostatak pri dijeljenju. Npr. „ $n \% 2$ “ je 0 ako i samo ako je n paran broj!

#10 Napisati program koji računa zbir, razliku, proizvod i količnik posljednje dvije cifre unesene cijelog broja. Ukoliko broj ima manje od dvije cifre, cifru koja nedostaje (tj. prvu cifru) zamijeniti nulom. U slučaju dijeljenja s nulom, ispisati „Nemoguće!“. Primjeri ispisa:

Unesite cijeli broj: -123
Zbir zadnje dvije cifre (2 i 3) je: 5
Razlika zadnje dvije cifre (2 i 3) je: -1
Proizvod zadnje dvije cifre (2 i 3) je: 6
Kolicnik zadnje dvije cifre (2 i 3) je: 0.666667

Unesite cijeli broj: 75
Zbir zadnje dvije cifre (7 i 5) je: 12
Razlika zadnje dvije cifre (7 i 5) je: 2
Proizvod zadnje dvije cifre (7 i 5) je: 35
Kolicnik zadnje dvije cifre (7 i 5) je: 1.400000

Unesite cijeli broj: 123456
Zbir zadnje dvije cifre (5 i 6) je: 11
Razlika zadnje dvije cifre (5 i 6) je: -1
Proizvod zadnje dvije cifre (5 i 6) je: 30
Kolicnik zadnje dvije cifre (5 i 6) je: 0.833333

Unesite cijeli broj: 12300
Zbir zadnje dvije cifre (0 i 0) je: 0
Razlika zadnje dvije cifre (0 i 0) je: 0
Proizvod zadnje dvije cifre (0 i 0) je: 0
Kolicnik zadnje dvije cifre (0 i 0) je: Nemoguće!

Unesite cijeli broj: 7

Zbir zadnje dvije cifre (0 i 7) je: 7
Razlika zadnje dvije cifre (0 i 7) je: -7
Proizvod zadnje dvije cifre (0 i 7) je: 0
Kolicnik zadnje dvije cifre (0 i 7) je: 0.000000

#11 Napisati program koji sekunde (koje se unose kao cijeli broj), pretvara u dane, sate, minute i sekunde. Zanimariti predznak unesenog broja. Primjeri ispisa:

Unesite broj sekundi: 65
Pretvoreno: 0d 0h 1m 5s

Unesite broj sekundi: 3600
Pretvoreno: 0d 1h 0m 0s

Unesite broj sekundi: 105235
Pretvoreno: 1d 5h 13m 55s

Unesite broj sekundi: 175924
Pretvoreno: 2d 0h 52m 4s

Unesite broj sekundi: -346
Pretvoreno: 0d 0h 5m 46s

Unesite broj sekundi: 0
Pretvoreno: 0d 0h 0m 0s

Savjet: koristiti modulo operator kako biste sveli broj na određeni opseg, npr. računanje minuta:

```
int m = (ukupne_sekunde / 60) % 60;
```

#12 Napisati program sličan prethodnom, samo se ovaj put pretvaraju stepeni iz decimalnog formata u cjelobrojne stepene, minute i sekunde. Korisnik unosi stepene kao realan broj. Iako izgleda jako slično prethodnom zadatku, pristup za računanje je potpuno drugačiji. Primjeri ispisa:

Unesite stepene: 112.435
112.435 = 112° 26' 6''

Unesite stepene: 0
0 = 0° 0' 0''

Unesite stepene: 180.05
180.05 = 180° 3' 0''

Unesite stepene: 90
90 = 90° 0' 0''

```
Unesite stepene: 45.5624
45.5624 = 45^ 33' 44''
```

Savjet: za dobivanje cijelog dijela broja koristiti operator konverzije u `int`. Npr. ako želimo da od broja $x = 123.456$ odsiječemo decimale, tj. da dobijemo broj 123, to će izgledati ovako: `cijeli = (int)x;`

#13 Napisati program koji „pametno“ ispisuje godine unesene kao cijeli broj iz intervala $[0, 100]$. Ukoliko broj nije iz intervala, program treba da ispiše (proizvoljnu) grešku i završi sa radom. Program treba, u zavisnosti od unesenog broja, da ispiše „godina“, „godinu“ ili „godine“. Pogledati primjere ispisa:

```
Unesite godine: 12
Imate 12 godina
```

```
Unesite godine: 77
Imate 77 godina
```

```
Unesite godine: 2
Imate 2 godine
```

```
Unesite godine: 14
Imate 14 godina
```

```
Unesite godine: 22
Imate 22 godine
```

```
Unesite godine: 4
Imate 4 godine
```

```
Unesite godine: 21
Imate 21 godinu
```

#14 Prepraviti zadatak #8 kombinirajući ga sa prethodnim zadatkom (#13) tako da sada ne ispisuje „godina/e/u“ nego da odluči šta da upotrijebi prilikom ispisa. Ako ste zadatak pametno uradili, onda ćete izmjenju raditi **samo na jednom** mjestu. Cilj ovoga zadatka je da vas pokuša usmjeriti da zadatke pišete što modularnije i skalabilnije, pa kada je potrebno nešto mijenjati ili dodavati, to uradite na što manje mjesta jer time smanjujete mogućnost grešaka. Što manje pišete kôda koji se ponavlja, to je bolje!

#15 Napisati program koji od korisnika traži unos cijelog broja te ispituje da li se taj broj nalazi u nekom od ispod navedenih intervala. Ako se uneseni broj ne nalazi u nekom od intervala, potrebno je ispisati „Ne nalazi“, u suprotnom treba ispisati „Nalazi“. Ispitati pripadnost broja sljedećim intervalima:

A: $[-3, 11)$, $(5, 15)$, $(17, 31]$, $(19, 33)$

B: $(-1, 1]$, $[2, 12]$, $[5, 56]$, $[9, 10)$

C: $(-11, -6)$, $(15, 16]$, $[6, 9]$

Broj pripada intervalu ako pripada nekom od intervala iz skupine **A** i ako pripada nekom od intervala iz skupine **B**, te ako ne pripada nekom od intervala iz **C** skupine.

Napomena: cilj ovog zadatka jeste da naučite kombinirati više uslova istovremeno, povezujući ih (ispravnim) logičkim operatorima, bez miješanja I (&&) i ILI (||).

#16 Napisati program koji od korisnika traži da unese 3 cijela broja, te provjerava da li su ti brojevi u nekom od sljedećih intervala: [-20, -8], [-1, 1], [40, 50], [77, 101]. Program za svaki broj treba da ispiše kojem intervalu pripada odnosno da ne pripada niti jednom intervalu. Na kraju, program treba da ispiše da li svi uneseni brojevi pripadaju jednom te istom intervalu. Primjeri:

```
Unesite 3 cijela broja: 3 1 -14
Broj 3 ne pripada niti jednom intervalu
Broj 1 pripada intervalu [-1, 1]
Broj -14 pripada intervalu [-20, -8]
Svi brojevi ne pripadaju jednom te istom intervalu.
```

```
Unesite 3 cijela broja: 80 77 90
Broj 80 pripada intervalu [77, 101]
Broj 77 pripada intervalu [77, 101]
Broj 90 pripada intervalu [77, 101]
Svi brojevi pripadaju jednom intervalu
```

Ukoliko pametno smislite logiku rješavanja zadatka, ova posljednja provjera za sve brojeve može biti trivijalna, stoga dobro razmislite da, kada već pojedinačno ispitujete pripadnost brojeva, možete i zapisati kojem je intervalu pripadao, te će finalna provjera biti lagana. Savjet: imajte dodatne varijable koje će „čuvati“ pripadnost pojedinih brojeva određenim intervalima.

#17 Napisati program koji od korisnika traži da unese tri broja koji predstavljaju *dan*, *mjesec* i *godinu*. Nakon toga će program ispisati datum u sljedećem formatu „Uneseni datum je X. mjesec Y. godine“. Primjer:

```
Unesite dan, mjesec i godinu (razdvojite razmakom): 21 9 2008
Uneseni datum je 21. septembar 2008. godine
```

Program treba da izvrši provjeru ulaznih podataka (*mjesec* ne smije biti manji od 1 niti veći od 12, *dan* mora biti u određenim granicama [1, 31] – nije potrebno da provjeravate broj dana za određeni mjesec npr. februar da ima 28/29 dana, april 30 i slično, a *godina* ne smije biti manja od 1 niti veća d 4000). U slučaju da se unese pogrešan podatak, program treba ispisati „Pogrešni ulazni podaci!“ i završiti sa radom.

#18 Napisati program koji će koristiti kao jednoupotrebnii jednostavni kalkulator. Program najprije izlistava operacije koje korisnik može koristiti, a zatim, nakon što je korisnik odabrao operaciju, unosi jedan ili dva broja/operanda (u zavisnosti od

operacije) i vrši tu operaciju nad njima. Potrebno da je se među ponuđenim operacijama nađu:

Operacija	Znak	Broj operanada
Sabiranje	+	2
Oduzimanje	-	2
Množenje	*	2
Dijeljenje	/	2
Kvadriranje	#	1
Korjenovanje	&	1

Primjer 1:

--- Moj mini kalkulator ---

Sabiranje: +

Oduzimanje: -

Množenje: *

Dijeljenje: /

Kvadriranje: #

Korjenovanje: &

Odaberite operaciju: *

Unesite prvi broj: 11

Unesite drugi broj: 15.5

Rezultat: $11 * 15.5 = 170.5$

Primjer 2:

--- Moj mini kalkulator ---

Sabiranje: +

Oduzimanje: -

Množenje: *

Dijeljenje: /

Kvadriranje: #

Korjenovanje: &

Odaberite operaciju: #

Unesite broj: 23

Rezultat: $23^2 = 529$

Primjer 3:

```
--- Moj mini kalkulator ---  
Sabiranje: +  
Oduzimanje: -  
Množenje: *  
Dijeljenje: /  
Kvadriranje: #  
Korjenovanje: &
```

```
Odaberite operaciju: %  
GRESKA:Nepostojeca  
operacija!
```

Primjer 4:

```
--- Moj mini kalkulator ---  
Sabiranje: +  
Oduzimanje: -  
Množenje: *  
Dijeljenje: /  
Kvadriranje: #  
Korjenovanje: &
```

```
Odaberite operaciju: -  
Unesite prvi broj: 2311  
Unesite drugi broj: 6003
```

```
Rezultat: 2311 - 6002 =  
-3692
```

Primjer 5:

```
--- Moj mini kalkulator ---  
Sabiranje: +  
Oduzimanje: -  
Množenje: *  
Dijeljenje: /  
Kvadriranje: #
```

Korjenovanje: &

```
Odaberite operaciju: /  
Unesite prvi broj: 51  
Unesite drugi broj: 0  
GRESKA: Dijeljenje sa nulom!
```

Za operaciju korjenovanja, tj. operaciju „&“ bi trebalo ispisati: „Rezultat: $100^{(1/2)} = 10$ “, za uneseni broj 100.

Za dodatnu vježbu studenti mogu dodati i veći broj operacija ukoliko žele, npr. *sinus*, *cosinus*, n-ti korijen, stepenovanje i slično.

#19 Napišite program koji će omogućiti prodavnicu računanje cijena akcijskih artikala.

Ukoliko se pokuša računati cijena van radnog vremena prodavnice (radno vrijeme je 8:00 - 17:00), program neće odobriti računanje cijene. U suprotnom, program od

korisnika traži da unese prijašnju cijenu artikla, te se nova cijena računa na sljedeći način:

Ukoliko je cijena bila manja od 15KM, cijena artikla je povećana za 15%

Ukoliko je cijena bila u intervalu od 15KM do 45KM, cijena artikla je smanjena za 4KM

Ukoliko je cijena bila veća od 45KM, smanjena je za 12%.

Potrebno je da program ispiše prigodnu poruku o tome koja korekcija nad cijenom je izvršena te da ispiše novu cijenu.

#20 Pravite program za jednu poznatu teretanu u Sarajevu. Program treba da pravi kalkulacije na osnovu parametara koje unosi korisnik, a koji će korisniku biti korisni u konstrukciji treninga. Tačnije, na osnovu pola i visine, program treba da korisnika obavijesti o tome da li je potrebno da smrša ili se udeblja, i za koliko kilograma (u odnosu na idealnu masu). Odstupanje od manje od 3kg u odnosu na idealnu tjelesnu masu je potrebno zanemariti.

Formula za računanje idealne tjelesne mase je:

za muškarce: idealna masa[kg] = $50 + 2.3 (\text{visina[inč]} - 60)$

za žene: idealna masa[kg] = $45.5 + 2.3 (\text{visina[inč]} - 60)$

Korisnik unosi visinu u centimetrima, tako da ćete morati vršiti konverziju visine u inče, kako bi se mogla napraviti kalkulacija. 1 inč = 2,54cm

#21 Gospođa Betty ima tri kućna ljubimaca: psa, mačku i kornjaču. Veterinar je propisao posebnu (čudnu) dijetu za svakog kućnog ljubimca. Uputstvo je sljedeće:

- pas treba da jede $\frac{\text{zbir sati i minuta u trenutku jela} + \text{redni broj dana u sedmici}}{34} \text{ kg}$ hrane, od toga 1/3 mesa i ostatak hrane po izboru. Poslije 18h pas jede samo kreker, i to $\text{redni broj dana u sedmici} \cdot 7 \text{ kreker}$. Npr, ukoliko je danas srijeda, 17:45, pas treba da jede $\frac{17+45+3}{34} = 1,91 \text{ kg}$, od toga 0,64kg mesa i 1,27kg ostale hrane. Ako gospođa Betty odluči da hrani psa 20 minuta kasnije, pas treba da dobije samo $3 \cdot 7 = 21$ kreker.
- Mačka treba da jede $\frac{2 \cdot \text{broj sati kada jede}}{40} + \frac{\text{redni broj dana u sedmici}}{7} \text{ kg}$ mačije hrane. U srijedu u 17:45 ona treba da dobije $\frac{2 \cdot 17}{40} + 3/7 = 1,28 \text{ kg}$ mačije hrane
- Kornjača treba da jede $\frac{7 + \text{broj sati}}{\text{redni broj dana u sedmici} + 154} \text{ kg}$ hrane za kornjače. Ako je četvrtak potrebno joj je dodati i kalcij u obrok. Dakle, kornjača u srijedu u 17:45 treba da dobije $\frac{7+17}{3+154} = 0,15 \text{ kg}$ hrane za kornjače.

Također, veterinar je strogo napomenuo da ljubimci ne smiju jesti prije 8:00h.

S obzirom da je gospođa Betty starija, napravite program koji će za nju računati koliko hrane treba da dâ svakom ljubimcu.

#22 (Zadaća 1, 2016/17) Tri prijatelja - Tarik, Bojan i Mirza - odlučili su da uporede svoj uspjeh na predmetu Osnove računarstva. Njima je potreban program koji omogućuje unos bodova po kategorijama: I parcijalni ispit (max. 20 bodova), II

parcijalni ispit (max. 20 bodova), Prisustvo (max. 10 bodova), Zadaće (max. 10 bodova) i Završni ispit (max. 40 bodova). Broj bodova može imati i decimalni dio. U slučaju da se unese negativan broj bodova ili veći od dozvoljenog, program treba ispisati poruku "Neispravan broj bodova" (bez navodnika) i prekinuti rad programa.

Ako su svi bodovi uneseni ispravno, nakon unosa treba odrediti ocjene za svakog studenta po važećoj skali:

- do 55 bodova - student pao
- [55, 65) bodova - ocjena 6
- [65, 75) bodova - ocjena 7
- [75, 85) bodova - ocjena 8
- [85, 92) bodova - ocjena 9
- [92, 100] bodova - ocjena 10

Potrebno je ispisati jednu od sljedećih poruka ovisno o situaciji:

- Nijedan student nije položio.
- Jedan student je položio.
- Dva studenta su položila.
- Sva tri studenta su položila.

Te ako su sva tri studenta položila, u sljedećem redu ispisati jednu od poruka:

- Sva tri studenta imaju istu ocjenu.
- Dva od tri studenta imaju istu ocjenu.
- Svaki student ima različitu ocjenu.

Primjer ulaza i izlaza:

```
Unesite bodove za Tarika:
I parcijalni ispit: 18
II parcijalni ispit: 15
Prisustvo: 9
Zadace: 5
Zavrsni ispit: 30
Unesite bodove za Bojana:
I parcijalni ispit: 12
...
Sva tri studenta su polozila.
Svaki student ima razlicitu ocjenu.
```

Napomena: Pri rješavanju zadatka **nije dozvoljeno** koristiti nizove niti petlje!

#23 (Zadaća 1, 2016/17) Date su dvije linearne jednačine:

$$y_1 = a_1x + b_1$$
$$y_2 = a_2x + b_2$$

Ove dvije jednačine opisuju dvije prave u koordinatnom prostoru. Potrebno je napisati program koji određuje tačku u kojoj se te dvije prave sijeku (x za koje je $y_1=y_2$) ili, ako su prave paralelne, ispisati poruku "Paralelne su", a ako se prave poklapaju treba

ispisati poruku "Poklapaju se". Program najprije omogućuje unos četiri realna koeficijenta a_1 , b_1 , a_2 , b_2 , a zatim ispisuje koordinate tačke sjecišta zaokružene na jednu decimalu.

Primjer ulaza i izlaza:

```
Unesite a1,b1,a2,b2: 3.1,5.2,2.1,1.0
Prave se sijeku u tacni (-4.2,-7.8)
```

Analiza programskog kôda

#1 Analizirajte sljedeći isječak kôda. Koji će biti njegov izlaz (šta će program ispisati)?

```
int a = 5, b = 4;
int r1 = a/b;
float r2 = a/b;
float p = a/b * 4;
printf("%d %.2f %.f %+d\n", r1, r2, r2, r1);
printf("%.2f %.2f %.2f \n", 4/3., 4./3, 4./3.);
printf("%.2f\n", p);
printf("%.2f %7.2f %.12f\n", 51.5456, 51.5456, 51.5456);
printf("%.2f, %.2f \n", 12.112121, 12.11742);
printf("%.2f %.2f %.f\n", 2.2f, .2f, 2.f);
```

Potrebno je uzeti u obzir svaki karakter, uključujući razmake.

#2 Analizirajte sljedeći isječak koda. Koji će biti njegov izlaz?

```
char e = '*';
char i = '.';
printf("%.10d.\n", 5);
printf("%.10d.\n", 5);
printf("%07.2f %04d %01.4f \n", 0.0, 0, 0.0);
printf("%c*%5c*%-5c* \n" , e, e, e);
printf("%c \n %c\n  %c\n", i, i, i);
```

#3 Analizirajte sljedeći isječak koda. Koji će biti njegov izlaz?

```
int a = 5, b = 4, c = 2;
float d = 7.25, e = 5, f = 54.43;

if(a / b % c < a / b) printf("a");
else printf("b");
```

```
if(c / a % b == c / (a % b)) printf("a");  
else printf("b");
```

```
if(a % c / b == (a % c) / b) printf("a");  
else printf("b");
```

```
if(d - a / b <= b + c) printf("a"); else printf("b");  
if(d - e / b <= b + c) printf("a"); else printf("b");
```