

## PROGRAMIRANJE 1



**Milena Vujošević Jančić, Jovana Kovačević,  
Danijela Simić, Anđelka Zečević**

# **PROGRAMIRANJE 1**

## **Zbirka zadataka**

**Beograd  
2017.**

Autori:

*dr Milena Vujošević Jančić*, docent na Matematičkom fakultetu u Beogradu

*dr Jovana Kovačević*, docent na Matematičkom fakultetu u Beogradu

*Danijela Simić*, asistent na Matematičkom fakultetu u Beogradu

*Anđelka Zečević*, asistent na Matematičkom fakultetu u Beogradu

PROGRAMIRANJE 1

Zbirka zadataka

# Sadržaj

<b>1</b>	<b>Uvodni zadaci</b>	<b>1</b>
1.1	Naredba izraza . . . . .	1
1.2	Rešenja . . . . .	12
<b>2</b>	<b>Kontrola toka</b>	<b>29</b>
2.1	Naredbe grananja . . . . .	29
2.2	Rešenja . . . . .	43



# 1

## Uvodni zadaci

### 1.1 Naredba izraza

**Zadatak 1.1.1** Napisati program koji na standardni izlaz ispisuje tekst Zdravo svima!.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Zdravo svima!
```

[Rešenje 1.1.1]

**Zadatak 1.1.2** Napisati program za uneti ceo broj ispisuje njegov kvadrat i njegov kub.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: 4  
|| Kvadrat: 16  
|| Kub: 64
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite ceo broj: -14  
|| Kvadrat: 196  
|| Kub: -2744
```

[Rešenje 1.1.2]

**Zadatak 1.1.3** Napisati program koji za uneta dva cela broja  $x$  i  $y$  ispisuje njihov zbir, razliku, proizvod, ceo deo pri deljenju prvog broja drugim brojem i

## 1 Uvodni zadaci

---

ostatak pri deljenju prvog broja drugim brojem. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednost promenljive x: 7
Unesite vrednost promenljive y: 2
7 + 2 = 9
7 - 2 = 5
7 * 2 = 14
7 / 2 = 3
7 % 2 = 1
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednost promenljive x: -3
Unesite vrednost promenljive y: 8
-3 + 8 = 5
-3 - 8 = -11
-3 * 8 = -24
-3 / 8 = 0
-3 % 8 = -3
```

[Rešenje 1.1.3]

**Zadatak 1.1.4** Napisati program koji pomaže kasirki da izračuna ukupan račun ako su poznate cene dva kupljena artikla. Cene artikala su pozitivni celi brojevi. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite cenu prvog artikla: 173
Unesite cenu drugog artikla: 2024
Ukupna cena iznosi 2197
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite cenu prvog artikla: 384
Unesite cenu drugog artikla: 555
Ukupna cena iznosi 939
```

[Rešenje 1.1.4]

**Zadatak 1.1.5** Napisati program koji za unetu količinu jabuka u kilogramima i unetu cenu po kilogramu ispisuje ukupnu vrednost date količine jabuka. Obe ulazne vrednosti su pozitivni celi brojevi. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite kolicinu jabuka (u kg): 6
Unesite cenu (u dinarima): 82
Molimo platite 492 dinara.
```

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite kolicinu jabuka (u kg): 10
Unesite cenu (u dinarima): 93
Molimo platite 930 dinara.
```

[Rešenje 1.1.5]

**Zadatak 1.1.6** Napisati program koji pomaže kasirki da obračuna kusur koji treba da vrati kupcu. Za unetu cenu artikla, količinu artikla i iznos koji je



kupac dao, program treba da ispiše vrednost kusura. Sve ulazne vrednosti su pozitivni celi brojevi. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite cenu, kolicinu i iznos:
132 2 500
Kusur je 236 dinara.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite cenu, kolicinu i iznos:
59 6 2000
Kusur je 1646 dinara.
```

[Rešenje 1.1.6]

**Zadatak 1.1.7** Napisati program koji za uneta vremena poletanja i sletanja aviona ispisuje dužinu trajanja leta. NAPOMENA: *Pretpostaviti da su poletanje i sletanje u istom danu kao i da su sve vrednosti ispravno unete.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme poletanja: 8 5
Unesite vreme sletanja: 12 41
Duzina trajanja leta je 4 h i 36 min
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite vreme poletanja: 13 20
Unesite vreme sletanja: 18 45
Duzina trajanja leta je 5 h i 25 min
```

[Rešenje 1.1.7]

**Zadatak 1.1.8** Date su dve celobrojne promenljive  $x$  i  $y$ . Napisati program koji razmenjuje njihove vrednosti.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti x i y: 5 7
Pre zamene: x=5, y=7
Posle zamene: x=7, y=5
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti x i y: 237 -592
Pre zamene: x=237, y=-592
Posle zamene: x=-592, y=237
```

[Rešenje 1.1.8]

**Zadatak 1.1.9** Date su dve celobrojne promenljive  $a$  i  $b$ . Napisati program koji promenljivoj  $a$  dodeljuje njihovu sumu, a promenljivoj  $b$  njihovu razliku. NAPOMENA: *Ne koristiti pomoćne promenljive.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti a i b: 5 7
Nove vrednosti su: a=12, b=-2
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite vrednosti a i b: 237 -592
Nove vrednosti su: a=-355, b=829
```

[Rešenje 1.1.9]

**Zadatak 1.1.10** Napisati program koji za uneti pozitivan trocifreni broj ispisuje njegove cifre jedinica, desetica i stotina. NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite trocifreni broj: 697  
jedinica 7, desetica 9, stotina 6
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite trocifreni broj: 504  
jedinica 4, desetica 0, stotina 5
```

[Rešenje 1.1.10]

**Zadatak 1.1.11** Napisati program koji za unetu cenu proizvoda ispisuje najmanji broj novčanica koje je potrebno izdvojiti prilikom plaćanja proizvoda. Na raspolaganju su novčanice od 5000, 2000, 1000, 500, 200, 100, 50, 20, 10 i 1 dinar. Cena proizvoda je pozitivan ceo broj. NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite cenu proizvoda: 8367  
8367 = 1*5000 + 1*2000 + 1*1000 + 0*500 + 1*200 + 1*100 + 1*50 + 0*20 + 1*10 + 7*1
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite cenu proizvoda: 934  
934 = 0*5000 + 0*2000 + 0*1000 + 1*500 + 2*200 + 0*100 + 0*50 + 1*20 + 1*10 + 4*1
```

[Rešenje 1.1.11]

**Zadatak 1.1.12** Napisati program koji učitava pozitivan trocifreni broj i ispisuje broj dobijen obrtanjem njegovih cifara. NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite trocifreni broj: 892  
Obrnuto: 298
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite trocifreni broj: 230  
Obrnuto: 32
```

[Rešenje 1.1.12]

**Zadatak 1.1.13** Napisati program koji za uneti pozitivan četvorocifreni broj:

- (a) izračunava proizvod cifara
- (b) izračunava razliku sume krajnjih i srednjih cifara
- (c) izračunava sumu kvadrata cifara
- (d) izračunava broj koji se dobija ispisom cifara u obrnutom poretku
- (e) izračunava broj koji se dobija zamenom cifre jedinice i cifre stotine

NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite cetvorocifreni broj: 2371
Proizvod cifara: 42
Razlika sume krajnjih i srednjih: -7
Suma kvadrata cifara: 63
Broj u obrnutom poretku: 1732
Broj sa zamenjenom cifrom jedinica i stotina: 2173
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite cetvorocifreni broj: 3570
Proizvod cifara: 0
Razlika sume krajnjih i srednjih: -9
Suma kvadrata cifara: 83
Broj u obrnutom poretku: 753
Broj sa zamenjenom cifrom jedinica i stotina: 3075
```

[Rešenje 1.1.13]

**Zadatak 1.1.14** Napisati program koji ispisuje broj koji se dobija izbacivanjem cifre desetica u unetom pozitivnom celom broju. NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 1349
Rezultat je: 139
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:
Unesite broj: 825
Rezultat je: 85
```

[Rešenje 1.1.14]

## 1 Uvodni zadaci

---

**Zadatak 1.1.15** Napisati program koji učitava pozitivan ceo broj  $n$  i pozitivan dvocifreni broj  $m$  i ispisuje broj dobijen umetanjem broja  $m$  između cifre stotina i cifre hiljada broja  $n$ . NAPOMENA: *Za neke ulazne podatke može se dobiti neočekivan rezultat zbog prekoračenja, što ilustruje test primer broj 2.*

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite pozitivan ceo broj: 12345  
|| Unesite pozitivan dvocifreni broj: 67  
|| Novi broj je 1267345
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite pozitivan ceo broj: 50000000  
|| Unesite pozitivan dvocifreni broj: 12  
|| Novi broj je 705044704
```

[Rešenje 1.1.15]

**Zadatak 1.1.16** Napisati program koji učitava realnu vrednost izraženu u inčima, konvertuje tu vrednost u centimetre i ispisuje je zaokruženu na dve decimalne. UPUTSTVO: *Jedan inč ima 2.54 centimetra.*

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj inča: 4.69  
|| 4.69 in = 11.91 cm
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj inča: 71.426  
|| 71.43 in = 181.42 cm
```

[Rešenje 1.1.16]

**Zadatak 1.1.17** Napisati program koji učitava dužinu izraženu u miljama, konvertuje tu vrednost u kilometre i ispisuje je zaokruženu na dve decimalne. UPUTSTVO: *Jedna milja ima 1.609344 kilometara.*

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj milja: 50.42  
|| 50.42 mi = 81.14 km
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj milja: 327.128  
|| 327.128 mi = 526.46 km
```

[Rešenje 1.1.17]

**Zadatak 1.1.18** Napisati program koji učitava težinu izraženu u funtama, konvertuje tu vrednost u kilograme i ispisuje je zaokruženu na dve decimalne. UPUTSTVO: *Jedna funta ima 0.45359237 kilograma.*

## Primer 1

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj funti: 2.78
|| 2.78 lb = 1.26 kg

```

## Primer 2

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite broj funti: 89.437
|| 89.437 lb = 40.57 kg

```

[Rešenje 1.1.18]

**Zadatak 1.1.19** Napisati program koji učitava temperaturu izraženu u farenhajtima, konvertuje tu vrednost u celzijuse i ispisuje je zaokruženu na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.* UPUTSTVO: *Veza između farenhajta i celzijusa je zadata narednom formulom  $F = \frac{9 \cdot C}{5} + 32$*

## Primer 1

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite temperaturu u F: 100.93
|| 100.93 F = 38.29 C

```

## Primer 2

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite temperaturu u F: 25.562
|| 25.562 F = -3.58 C

```

[Rešenje 1.1.19]

**Zadatak 1.1.20** Napisati program koji za unete realne vrednosti  $a_{11}$ ,  $a_{12}$ ,  $a_{21}$ ,  $a_{22}$  ispisuje vrednost determinante matrice:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Pri ispisu vrednost zaokružiti na 4 decimale.

## Primer 1

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 1 2 3 4
|| Determinanta: -2.0000

```

## Primer 2

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: -1 0 0 1
|| Determinanta: -1.0000

```

## Primer 3

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 1.5 -2 3 4.5
|| Determinanta: 12.7500

```

## Primer 4

```

|| INTERAKCIJA SA PROGRAMOM:
|| Unesite brojeve: 0.01 0.01 0.5 7
|| Determinanta: 0.0650

```

[Rešenje 1.1.20]

**Zadatak 1.1.21** Napisati program koji za unete realne vrednosti dužina stranica pravougaonika ispisuje njegov obim i površinu. Ispisati tražene vrednosti zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.*

## 1 Uvodni zadaci

---

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite duzine stranica: 4.3 9.4
|| Obim: 27.40
|| Povrsina: 40.42
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite duzine stranica: 10.756 36.2
|| Obim: 93.91
|| Povrsina: 389.37
```

[Rešenje 1.1.21]

**Zadatak 1.1.22** Napisati program koji za unetu realnu vrednost dužine poluprečnika kruga ispisuje njegov obim i površinu zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite poluprecnik: 4.2
|| Obim: 26.39
|| Povrsina: 55.42
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite poluprecnik: 14.932
|| Obim: 93.82
|| Povrsina: 700.46
```

[Rešenje 1.1.22]

**Zadatak 1.1.23** Napisati program koji za unetu realnu vrednost dužine stranice jednakostraničnog trougla ispisuje njegov obim i površinu zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.* UPUTSTVO: *Za računanje korena broja koristiti funkciju `sqrt` čija se deklaracija nalazi u zaglavlju `math.h`.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite duzinu stranice trougla: 5
|| Obim: 15.00
|| Povrsina: 10.82
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite duzinu stranice trougla: 2
|| Obim: 6.00
|| Povrsina: 1.73
```

[Rešenje 1.1.23]

**Zadatak 1.1.24** Napisati program koji za unete realne vrednosti dužina stranica trougla ispisuje njegov obim i površinu zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite duzine stranica trougla:
3 4 5
Obim: 12.00
Povrsina: 6.00

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite duzine stranica trougla:
4.3 9.7 8.8
Obim: 22.80
Povrsina: 18.91

```

[Rešenje 1.1.24]

**Zadatak 1.1.25** Pravougaonik čije su stranice paralelne koordinatnim osama zadan je svojim realnim koordinatama suprotnih temena (gornje levo i donje desno teme). Napisati program koji ispisuje njegov obim i površinu zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.*

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite koordinate gornjeg levog temena: 4.3 5.8
Unesite koordinate donjeg desnog temena: 6.7 2.3
Obim: 11.80
Povrsina: 8.40

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite koordinate gornjeg levog temena: -3.7 8.23
Unesite koordinate donjeg desnog temena: -0.56 2
Obim: 18.74
Povrsina: 19.56

```

[Rešenje 1.1.25]

**Zadatak 1.1.26** Napisati program koji za tri uneta cela broja ispisuje njihovu aritmetičku sredinu zaokruženu na dve decimale.

*Primer 1*

```

INTERAKCIJA SA PROGRAMOM:
Unesite tri cela broja: 11 5 4
Aritmeticka sredina: 6.67

```

*Primer 2*

```

INTERAKCIJA SA PROGRAMOM:
Unesite tri cela broja: 3 -8 13
Aritmeticka sredina: 2.67

```

[Rešenje 1.1.26]

**Zadatak 1.1.27** Napisati program koji pomaže moleru da izračuna površinu zidova prostorije koju treba da okreči. Za unete celobrojne vrednosti dimenzije

## 1 Uvodni zadaci

---

sobe u metrima (dužinu, širinu i visinu), program treba da ispiše površinu zidova za krečenje pod pretpostavkom da na vrata i prozore otpada oko 20%. Omogućiti i da na osnovu unete celobrojene cene usluge po kvadratnom metru program izračuna ukupnu cenu krečenja. Sve realne vrednosti ispisati zaokružene na dve decimale. NAPOMENA: *Pretpostaviti da je unos ispravan.*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenzije sobe: 4 4 3
Unesite cenu po m2: 500
Moler treba da okreći 51.20 m2
Cena krecenja je 25600.00
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dimenzije sobe: 13 17 3
Unesite cenu po m2: 475
Moler treba da okreći 320.80 m2
Cena krecenja je 152380.00
```

[Rešenje 1.1.27]

**Zadatak 1.1.28** Napisati program koji za unete pozitivne cele brojeve  $x$ ,  $p$  i  $c$  ispisuje broj koji se dobija ubacivanjem cifre  $c$  u broj  $x$  na poziciju  $p$ . Pretpostaviti da numeracija cifara počinje od nule, odnosno da se cifra najmanje težine nalazi se na nultoj poziciji. NAPOMENA: *Pretpostaviti da je unos ispravan.* UPUTSTVO: *Koristiti funkciju  $\text{pow}$  čija se deklaracija nalazi u zaglavlju  $\text{math.h}$ .*

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite redom x, p i c: 140 1 2
Rezultat je: 1420
```

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite redom x, p i c: 12345 2 9
Rezultat je: 123945
```

[Rešenje 1.1.28]

**Zadatak 1.1.29** Napisati program koji za uneta dva cela broja  $a$  i  $b$  dodeljuje promenljivoj *rezultat* vrednost 1 ako važi uslov:

- a)  $a$  i  $b$  su različiti brojevi
- b)  $a$  i  $b$  su parni brojevi
- c)  $a$  i  $b$  su pozitivni brojevi, ne veći od 100

U suprotnom, promenljivoj *rezultat* dodeliti vrednost 0. Ispisati vrednost promenljive *rezultat*.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva cela broja: 4 8
a) rezultat=1
b) rezultat=1
c) rezultat=1
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite dva cela broja: 3 -11
a) rezultat=1
b) rezultat=0
c) rezultat=0
```



[Rešenje 1.1.29]

**Zadatak 1.1.30** Napisati program koji za uneta dva cela broja ispisuje njihov maksimum.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva cela broja: 19 256
|| Maksimum je 256
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva cela broja: -39 57
|| Maksimum je 57
```

[Rešenje 1.1.30]

**Zadatak 1.1.31** Napisati program koji za uneta dva cela broja ispisuje njihov minimum.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva cela broja: 4 8
|| Minimum je 4
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva cela broja: -3 -110
|| Minimum je -110
```

[Rešenje 1.1.31]

**Zadatak 1.1.32** Napisati program koji za unete realne vrednosti promenljivih  $x$  i  $y$  ispisuje vrednost sledećeg izraza:

$$rez = \frac{\min(x, y) + 0.5}{1 + \max^2(x, y)}$$

zaokruženu na dve decimale.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva realna broja: 5.7 11.2
|| Rezultat je: 0.05
```

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite dva realna broja: -9.34 8.99
|| Rezultat je: -0.11
```

[Rešenje 1.1.32]

### 1.2 Rešenja

#### Rešenje 1.1.1

```
1 #include<stdio.h>
2
3 int main()
4 {
5     /* Ispisuje se trazena poruka. Na kraju poruke se ispisuje i
6        novi red. */
7     printf("Zdravo svima!\n");
8
9     /* Povratna vrednost 0 se obicno koristi da oznaci da je prilikom
10        izvršavanja programa sve proslo u redu. */
11     return 0;
12 }
```

#### Rešenje 1.1.2

```
1 #include <stdio.h>
2
3 int main()
4 {
5     /* Deklaracija celobrojne promenljive. */
6     int n;
7
8     /* Ucitava se vrednost celog broja. */
9     printf("Unesite ceo broj: ");
10    scanf("%d", &n);
11
12    /* Ispis kvadratne vrednosti unetog broja. */
13    printf("Kvadrat: %d\n", n * n);
14
15    /* Ispis kubne vrednosti unetog broja. */
16    printf("Kub: %d\n", n * n * n);
17
18    return 0;
19 }
```

#### Rešenje 1.1.3

```
1 #include<stdio.h>
2
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     int x, y, rezultat;
```

```

8  /* Ucitava se vrednost broja x. */
   printf("Unesite vrednost promenljive x: ");
10  scanf("%d", &x);

12  /* Ucitava se vrednost broja y. */
   printf("Unesite vrednost promenljive y: ");
14  scanf("%d", &y);

16  /* I nacin ispisa: dodela zbira x+y promenljivoj rezultat i
      ispis vrednosti promenljive rezultat. */
18  rezultat = x + y;
   printf("%d + %d = %d\n", x, y, rezultat);

20  /* II nacin ispisa: direktan ispis vrednosti izraza, bez njegovog
      dodeljivanja posebnoj promenljivoj. */
22  printf("%d - %d = %d\n", x, y, x - y);
24  printf("%d * %d = %d\n", x, y, x * y);

26  /* Kada se operator / primeni na dva celobrojna argumenta x i y,
      kao rezultat se dobije ceo deo pri deljenju broja x brojem y,
      a ne kolicnik. Na primer, rezultat primene operatora / na 7 i 2
      je 3, a ne 3.5. */
30  printf("%d / %d = %d\n", x, y, x / y);

32  /* Operator % izracunava ostatak pri celobrojnem deljenju dve
      celobrojne promenljive.
      Da bi se odstampao karakter %, u naredbi printf se pise %%. */
34  printf("%d %% %d = %d\n", x, y, x % y);

36  return 0;
38 }

```

### Rešenje 1.1.4

Rešenje ovog zadatka svodi se na rešenje zadatka 1.1.3, na deo koji se odnosi na izračunavanje zbira dva broja. Zbog pretpostavke da su cene artikala pozitivni celi brojevi, tip promenljivih za artikle treba da bude `unsigned int`.

### Rešenje 1.1.5

Rešenje ovog zadatka svodi se na rešenje zadatka 1.1.3, na deo koji se odnosi na izračunavanje proizvoda dva broja. Zbog pretpostavke da su cene artikala pozitivni celi brojevi, tip promenljivih za artikle treba da bude `unsigned int`.

### Rešenje 1.1.6

```

2  #include <stdio.h>

```

## 1 Uvodni zadaci

---

```
int main()
{
    /* Deklaracija promenljivih cija je vrednost neoznacena ceo broj. */
    unsigned int cena, kolicina, iznos;
    unsigned int kusur;

    /* Ucitavaju se vrednosti cene, kolicine i iznosa. */
    printf("Unesite cenu, kolicinu i iznos:\n");
    scanf("%u%u%u", &cena, &kolicina, &iznos);

    /* Izracunava se kusur. */
    kusur = iznos - kolicina * cena;

    /* Ispis vrednosti kusura. */
    printf("Kusur je %u dinara.\n", kusur);

    return 0;
}
```

### Rešenje 1.1.7

```
#include <stdio.h>

int main()
{
    /* Deklaracija potrebnih promenljivih. */
    unsigned int poletanje, poletanje_sat, poletanje_minut;
    unsigned int sletanje, sletanje_sat, sletanje_minut;
    unsigned int duzina, duzina_sat, duzina_minut;

    /* Ucitavaju se sat i minut vremena poletanja. */
    printf("Unesite vreme poletanja: ");
    scanf("%u%u", &poletanje_sat, &poletanje_minut);

    /* Ucitavaju se sat i minut vremena sletanja. */
    printf("Unesite vreme sletanja: ");
    scanf("%u%u", &sletanje_sat, &sletanje_minut);

    /* Obe vrednosti se pretvaraju u sekunde,
       kako bi se lakse izracunala razlika. */
    poletanje = poletanje_sat * 3600 + poletanje_minut * 60;
    sletanje = sletanje_sat * 3600 + sletanje_minut * 60;

    /* Racunanje razlike u sekundama izmedju sletanja i poletanja. */
    duzina = sletanje - poletanje;

    /* Razlika u sekundama se pretvara u razliku u satima i minutima.
       Razlika u satima se dobija celobrojn timer deljenjem broja sekundi
       sa 3600.
       Preostali broj minuta se dobija deljenjem preostalog broja
    */
}
```

```

32     sekundi sa 60. */
    duzina_sat = duzina / 3600;
    duzina_minut = (duzina - duzina_sat * 3600) / 60;
34
    /* II nacin: duzina_minut = (duzina % 3600) / 60; */
36
    /* Ispis rezultata. */
38    printf("Duzina trajanja leta je %u h i %u min\n", duzina_sat,
        duzina_minut);
40
    return 0;
42 }

```

### Rešenje 1.1.8

```

#include<stdio.h>
2
int main()
4 {
    /* Deklaracija potrebnih promenljivih. */
6    int x, y;
    int p;
8
    /* Ucitavaju se vrednosti x i y. */
10    printf("Unesite vrednosti x i y: ");
    scanf("%d%d", &x, &y);
12
    /* Ispis vrednosti promenljivih pre zamene. */
14    printf("Pre zamene: x=%d, y=%d\n", x, y);
16
    /* Pomocna promenljiva p je potrebna da sacuva vrednost
       promenljive x pre nego sto se ona izmeni i dobije vrednost
       promenljive y. */
18    p = x;
20    x = y;
    y = p;
22
    /* Ispis vrednosti promenljivih nakon zamene. */
24    printf("Posle zamene: x=%d, y=%d\n", x, y);
26
    return 0;
}

```

### Rešenje 1.1.9

```

1  #include<stdio.h>
3  int main()
   {

```

## 1 Uvodni zadaci

---

```
5  /* Deklaracija potrebnih promenljivih. */
   int a, b;

7

   /* Ucitavaju se vrednosti a i b. */
9  printf("Unesite vrednosti a i b: ");
   scanf("%d%d", &a, &b);

11

   /* U promenljivu a se smesta suma a+b. */
13  a = a + b;

15  /* U promenljivu b se smesta izraz a - 2*b, cija je vrednost (nakon
      promene promenljive a) jednaka a + b - 2*b = a - b. */
   b = a - 2*b;

17

   /* Ispis rezultata. */
19  printf("Nove vrednosti su: a=%d, b=%d\n", a, b);

21  return 0;
}
```

### Rešenje 1.1.10

```
1  #include <stdio.h>

3  int main()
   {
5     /* Deklaracija neoznacnog broja. */
     unsigned int x;

7

     /* Promenljive koje cuvaju cifre treba da budu najmanjeg
9      celobrojnog tipa jer nece sadrzati druge vrednosti osim
      jednocifrenih celih brojeva. Zbog toga se koristi tip char. */
11    char cifra_jedinice, cifra_desetice, cifra_stotine;

13    /* Ucitava se trocifren broj. */
     printf("Unesite trocifreni broj: ");
15    scanf("%u", &x);

17    /* Izdvajaju se cifre jedinice, desetice i stotine. */
     cifra_jedinice = x % 10;
19     cifra_desetice = (x / 10) % 10;
     cifra_stotine = x / 100;

21

     /* Ispis rezultata.
23      NAPOMENA: Kada se stampa numericka vrednost promenljive tipa
      char koristi se %d. Kada se stampa karakter ciji je ASCII
25      kod jednak vrednosti te promenljive, tada se koristi %c.
      U ovom slucaju je potrebno stampati numericku vrednost. */
27     printf("jedinica %d, desetica %d, stotina %d\n", cifra_jedinice,
        cifra_desetice, cifra_stotine);

29 }
```

```

31  /* II nacin: Ispis rezultata bez uvođenja dodatnih promenljivih
    cifra_jedinice, cifra_desetice i cifra_stotine:

33      printf("Cifre unetog broja su %d,%d,%d\n", x%10, (x/10)%10, x
        /100); */

35      return 0;
}

```

### Rešenje 1.1.11

```

1  #include <stdio.h>

3  int main()
{
5      /* Deklaracija i učitavanje cene proizvoda. */
    unsigned int x;
7      printf("Unesite cenu proizvoda: ");
    scanf("%u", &x);

9

11     /* Vrednost x/5000 predstavlja maksimalan broj novčanica od 5000
        dinara koje je moguće iskoristiti za plaćanje računa.
        Na primer, neka je uneta cena 8367 dinara, vrednost izraza
13     8367/5000 je jednaka 1. */
    printf("%u = %u*5000 + ", x, x / 5000);

15

17     /* Da bi se isti postupak primenio i na ostale novčanice, potrebno
        je izračunati preostali iznos. Jedan način da se to uradi je
        računanje ostatka pri deljenju unete vrednosti x
19     (u primeru 8367) sa 5000. On iznosi 3367. Ovu vrednost
        dodeljujemo promenljivoj x. */
21     x = x % 5000;

23     /* Postupak se ponavlja i za za ostale novčanice. */
    printf("%u*2000 + ", x / 2000);
25     x = x % 2000;
    printf("%u*1000 + ", x / 1000);
27     x = x % 1000;
    printf("%u*500 + ", x / 500);
29     x = x % 500;
    printf("%u*200 + ", x / 200);
31     x = x % 200;
    printf("%u*100 + ", x / 100);
33     x = x % 100;
    printf("%u*50 + ", x / 50);
35     x = x % 50;
    printf("%u*20 + ", x / 20);
37     x = x % 20;
    printf("%u*10 + ", x / 10);
39     x = x % 10;
    printf("%u*1\n", x);
}

```

## 1 Uvodni zadaci

---

```
41     return 0;
43 }
```

### Rešenje 1.1.12

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      unsigned int x;
7      unsigned int obrnuto_x;
8      char cifra_jedinice;
9      char cifra_desetice;
10     char cifra_stotine;
11
12     /* Ucitava se neoznaceni trocifreni broj. */
13     printf("Unesite trocifreni broj: ");
14     scanf("%u", &x);
15
16     /* Izdvajaju se pojedinačne cifre broja. */
17     cifra_jedinice = x % 10;
18     cifra_desetice = (x / 10) % 10;
19     cifra_stotine = x / 100;
20
21     /* Formira se rezultujući broj. */
22     obrnuto_x = cifra_jedinice * 100 + cifra_desetice * 10 +
23         cifra_stotine;
24
25     /* Ispis rezultata. */
26     printf("Obrnuto: %u\n", obrnuto_x);
27
28     return 0;
29 }
```

### Rešenje 1.1.13

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      unsigned int n, broj_obrnuto, broj_zamena;
7      char jedinice, desetice, stotine, hiljade;
8      int proizvod_cifara, razlika_cifara, suma_kvadrata;
9
10     /* Ucitava se jedan neoznaceni broj. */
11     printf("Unesite četvorocifreni broj: ");
```



```

scanf("%u", &n);

13
/* Izdvajaju se cifre ucitanog broja. */
15
jedinice = n % 10;
desetice = (n / 10) % 10;
17
stotine = (n / 100) % 10;
hiljade = n / 1000;
19

/* Izracunava se proizvod cifara. */
21
proizvod_cifara = jedinice * desetice * stotine * hiljade;
printf("Proizvod cifara: %d\n", proizvod_cifara);
23

/* Izracunava se razlika sume krajnjih i srednjih cifara. */
25
razlika_cifara = (hiljade + jedinice) - (stotine + desetice);
printf("Razlika sume krajnjih i srednjih: %d\n", razlika_cifara);
27

/* Izracunava se suma kvadrata cifara. */
29
suma_kvadrata = jedinice * jedinice + desetice * desetice +
                stotine * stotine + hiljade * hiljade;
31
printf("Suma kvadrata cifara: %d\n", suma_kvadrata);

33
/* Izracunava se broj zapisan istim ciframa ali u obrnutom
   redosledu. */
35
broj_obrnuto = jedinice * 1000 + desetice * 100 + stotine * 10 +
               hiljade;
printf("Broj u obrnutom poretku: %u\n", broj_obrnuto);
37

/* Izracunava se broj u kojem su cifra jedinica i cifra stotina
   zamenile mesta. */
39
broj_zamena = hiljade * 1000 + jedinice * 100 + desetice * 10 +
              stotine;
41
printf("Broj sa zamenjenom cifrom jedinica i stotina: %u\n",
       broj_zamena);

43
return 0;
}

```

### Rešenje 1.1.14

```

1 #include <stdio.h>

3 int main()
{
5     /* Deklaracija potrebnih promenljivih. */
    unsigned int broj, novibroj;
7     unsigned int levo, desno;

9     /* Ucitava se neoznaceni broj. */
    printf("Unesite broj: ");
11    scanf("%u", &broj);

```

## 1 Uvodni zadaci

---

```
13  /* Desni deo rezultata je cifra jedinice unetog broja.
    Na primer, za broj 1234, desni deo je cifra 4. */
15  desno = broj%10;

17  /* Levi deo rezultata su sve cifre levo od cifre desetice.
    Na primer, za broj 1234, levi deo je broj 12 i dobija se
19  deljenjem unetog broja sa 100. */
    levo = broj/100;

21

23  /* Rezultat se dobija spajanjem levog i desnog dela.
    U datom primeru: 12*10 + 4 = 124. */
    novibroj = levo*10 + desno;

25

27  /* Ispis rezultata. */
    printf("Rezultat je: %u\n", novibroj);

29  return 0;
}
```

### Rešenje 1.1.15

```
#include <stdio.h>

2  int main()
3  {
4      /* Deklaracija potrebnih promenljivih. */
5      unsigned int n, novibroj;
6      unsigned int levi, desni, m;
7
8      /* Ucitavaju se brojevi n i m. */
9      printf("Unesite pozitivan ceo broj: ");
10     scanf("%u", &n);
11     printf("Unesite pozitivan dvocifreni broj: ");
12     scanf("%u", &m);
13
14     /* Levi deo rezultata su sve cifre levo od cifre stotina.
        Na primer, ako je n=12345, levi deo rezultata je 12.
        On se dobija deljenjem unetog broja sa 1000. */
16     levi = n / 1000;
17
18     /* Desni deo rezultata su sve cifre desno od cifre hiljada.
        Za n=12345, desni deo rezultata je 345. */
20     desni = n % 1000;
21
22     /* Srednji deo rezultata je broj m.
        U navedenom primeru, rezultat se dobija nadovezivanjem
24     brojeva 12, 67 i 345. Ovo se radi mnozenjem delova sa
        odgovarajucim stepenom broja 10 i njihovim sabiranjem. */
26     novibroj = levi * 100000 + m * 1000 + desni;
27
28     /* Ispis rezultata. */
29
30 }
```

```
printf("Novi broj je %u\n", novibroj);
return 0;
}
```

### Rešenje 1.1.16

```
#include <stdio.h>

int main()
{
    /* Deklaracija potrebnih promenljivih. */
    float in, cm;

    /* Ucitava se realna vrednost koja predstavlja broj inca. */
    printf("Unesite broj inca: ");
    scanf("%f", &in);

    /* Izracunava se rezultat (1 in = 2.54 cm) */
    cm = in * 2.54;

    /* Ispis rezultata (na dve decimale). */
    printf("%.2f in = %.2f cm\n", in, cm);

    return 0;
}
```

### Rešenje 1.1.17

Zadatak se rešava analogno zadatku 1.1.16.

### Rešenje 1.1.18

Zadatak se rešava analogno zadatku 1.1.16.

### Rešenje 1.1.19

Zadatak se rešava analogno zadatku 1.1.16.

### Rešenje 1.1.20

```
#include <stdio.h>

int main()
{
    /* Deklaracija potrebnih promenljivih. */
    float a11, a12, a21, a22;
    float determinanta;
```

## 1 Uvodni zadaci

---

```
9  /* Ucitavaju se elementi matrice. */
   printf("Unesite brojeve: ");
11  scanf("%f%f%f%f", &a11, &a12, &a21, &a22);

13  /* Izracunava se determinanta matrice. */
   determinanta = a11*a22 - a12*a21;

15

17  /* Ispis rezultata na cetiri decimala. */
   printf("Determinanta: %.4f\n", determinanta);

19  return 0;
   }
```

### Rešenje 1.1.21

```
#include <stdio.h>

2
int main()
{
4  /* Deklaracija potrebnih promenljivih. */
   float a, b;
   float obim, povrsina;

8

10  /* Ucitavaju se duzine stranica pravougaonika. */
   printf("Unesite duzine stranica pravougaonika: ");
   scanf("%f%f", &a, &b);

12

14  /* Izracunava se obim pravougaonika. */
   obim = 2 * (a + b);

16  /* Izracunava se povrsina pravougaonika. */
   povrsina = a * b;

18

20  /* Ispis rezultata na dve decimala. */
   printf("Obim: %.2f\n", obim);
   printf("Povrsina: %.2f\n", povrsina);

22  return 0;
24 }
```

### Rešenje 1.1.22

```
#include <stdio.h>
2 #include <math.h>

4
int main()
{
6  /* Deklaracija potrebnih promenljivih. */
   float r, obim, povrsina;
```

```

8
/* Ucitava se poluprecnik kruga. */
10 printf("Unesite poluprecnik: ");
scanf("%f", &r);

12
/* Racunaju se obim i povrsina.
14     M_PI je konstanta koja se nalazi u zaglavlju math.h
    i njena vrednost odgovara približnoj vrednosti broja pi. */
16 obim = 2 * r * M_PI;
povrsina = r * r * M_PI;

18
/* Ispis rezultata na dve decimale. */
20 printf("Obim: %.2f\nPovrsina: %.2f\n", obim, povrsina);

22 return 0;
}

```

### Rešenje 1.1.23

```

#include <stdio.h>
2 #include <math.h>

4 int main()
{
6     /* Deklaracija potrebnih promenljivih. */
    float a, povrsina, obim;

8
    /* Ucitava se dužina stranice. */
10 printf("Unesite dužinu stranice trougla: ");
scanf("%f", &a);

12
    /* Racunaju se obim i povrsina. */
14 obim = 3 * a;
povrsina = (a * a * sqrt(3)) / 4;

16
    /* Ispis rezultata na dve decimale. */
18 printf("Obim: %.2f\n", obim);
printf("Povrsina: %.2f\n", povrsina);

20 return 0;

22 }

```

### Rešenje 1.1.24

```

1 #include <stdio.h>
#include <math.h>

3
int main()
5 {

```

## 1 Uvodni zadaci

---

```
/* Deklaracija potrebnih promenljivih. */
7 float a, b, c;
  float obim, s, površina;

9 /* Ucitavaju se duzine stranica. */
11 printf("Unesite duzine stranica trougla:\n");
   scanf("%f%f%f", &a, &b, &c);

13
15 /* Racuna se obim. */
   obim = a + b + c;

17 /* Racuna se površina koriscenjem Heronovog obrasca. */
   s = obim / 2;
19 površina = sqrt(s * (s - a) * (s - b) * (s - c));

21 /* Ispis rezultata. */
   printf("Obim: %.2f\n", obim);
23 printf("Površina: %.2f\n", površina);

25 return 0;
}
```

### Rešenje 1.1.25

Nakon ispravnog izračunavanja dužina stranica, zadatak se rešava analogno zadatku 1.1.21.

### Rešenje 1.1.26

```
1 #include<stdio.h>

3 int main()
  {
5     /* Deklaracija potrebnih promenljivih. */
      int a, b, c;
      float as;

7
9     /* Ucitavaju se tri cela broja. */
      printf("Unesite tri cela broja:");
11     scanf("%d%d%d", &a, &b, &c);

13     /* Pogresan nacin: as = (a+b+c)/3;
       * Kada se operacija / koristi nad celim brojevima,
       * deljenje je celobrojno.
       * Na primer, (1+1+3)/3 ima vrednost 1.*/

15
17     /* Ispravan nacin je da se bar jedan operand
       * pretvori u realan broj. */
19     as = (a + b + c) / 3.0;

21
```

```
23  /* Drugi ispravni nacini:
    as=1.0*(a+b+c)/3;
    as=(0.0+a+b+c)/3;
25  as=((float)(a+b+c))/3; */

27  /* Ispis rezultata. */
    printf("Aritmeticka sredina: %.2f\n", as);
29
    return 0;
31 }
```

## Rešenje 1.1.27

```
1  #include <stdio.h>

3  int main()
{
5  /* Deklaracija potrebnih promenljivih. */
    unsigned int duzina, sirina, visina;
7  unsigned int cena;
    float površina_za_krecenje;
9  float ukupna_cena;

11 /* Ucitavaju se vrednosti duzine, sirine i visine sobe. */
    printf("Unesite dimenzije sobe: ");
13 scanf("%u%u%u", &duzina, &sirina, &visina);

15 /* Ucitava se cena krecenja */
    printf("Unesite cenu po m2: ");
17 scanf("%u", &cena);

19 /* Povrsina za krecenje odgovara površini kvadra
    umanjena za površinu poda jer se on ne kreci. */
21 površina_za_krecenje = 0.8 * (duzina * sirina +
                                2 * duzina * visina +
23                                2 * sirina * visina);

25 /* Racuna se ukupna cena. */
    ukupna_cena = površina_za_krecenje * cena;
27

29 /* Ispis rezultata. */
    printf("Moler treba da okreći %.2f m2\n", površina_za_krecenje);
    printf("Cena krecenja je %.2f\n", ukupna_cena);
31
    return 0;
33 }
```

## Rešenje 1.1.28

## 1 Uvodni zadaci

---

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main()
5 {
6     /* Deklaracija potrebnih promenljivih. */
7     unsigned int x, p, c;
8     unsigned int levo, desno;
9     unsigned int novo_x;
10
11     /* Ucitavaju se broj, pozicija i cifra. */
12     printf("Unesite redom x, p i c: ");
13     scanf("%u%u%u", &x, &p, &c);
14
15     /* Racuna se deo broja koji se nalazi desno od pozicije p.
16        Funkcija pow kao povratnu vrednost vraca realan broj dvostruke
17        tacnosti, a operacija % ocekuje celobrojne operande. Iz tog
18        razloga je neophodno izvršiti pretvaranje povratne vrednosti
19        u tip unsigned int. */
20     desno = x % (unsigned int) pow(10, p);
21
22     /* Racuna se deo broja koji se nalazi levo od pozicije p. */
23     levo = x / (unsigned int) pow(10, p);
24
25     /* Rezultat se racuna nadovezivanjem levog dela, cifre c
26        i desnog dela. */
27     novo_x = levo * (unsigned int) pow(10, p + 1) +
28             c * (unsigned int) pow(10, p) + desno;
29
30     /* Ispis rezultata. */
31     printf("Rezultat je: %u\n", novo_x);
32
33     return 0;
34 }
35
```

### Rešenje 1.1.29

```
1 #include <stdio.h>
2
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     int a, b;
7     int rezultata, rezultatb, rezultatc;
8
9     /* Ucitavaju se dva cela broja. */
10    printf("Unesite dva cela broja: ");
11    scanf("%d%d", &a, &b);

```



```

13  /* Izraz a != b ima vrednost 1 ako je ova relacija tacna, a 0 ako
    je netacna. */
15  rezultata = a != b;

17  /* Izraz a%2==0 && b%2==0 je konjunkcija koja se sastoji od dve
    relacije poredjenja jednakosti. Izraz a%2==0 ima vrednost 1 ako
19  je ova relacija tacna, a 0 u suprotnom. */
    rezultatb = (a % 2 == 0 && b % 2 == 0);

21  /* Izraz a>0 && a<=100 && b>0 && b<=100 je konjunkcija koja se
    sastoji od cetiri konjunkata. Svaki od konjunkata je izraz
23  koji sadrzi relacioni operator i ima vrednost 1 ako relacija
    vazi, a 0 ako ne vazi. */
25  rezultatc = (a > 0 && a <= 100 && b > 0 && b <= 100);

27  /* Ispis rezultata. */
29  printf("a) rezultat=%d\n", rezultata);
    printf("b) rezultat=%d\n", rezultatb);
31  printf("c) rezultat=%d\n", rezultatc);

33  return 0;
}

```

### Rešenje 1.1.30

```

1  #include <stdio.h>

3  int main()
{
5  /* Deklaracija potrebnih promenljivih. */
    int a, b, max;

7  /* Ucitavaju se dve celobrojne vrednosti. */
    printf("Unesite dva cela broja: ");
9  scanf("%d%d", &a, &b);

11  /* Racuna se maksimum koriscenjem ternarnog operatora uslova. */
13  max = (a > b) ? a : b;

15  /* Ispis rezultata. */
    printf("Maksimum je %d\n", max);

17  return 0;

19 }

```

### Rešenje 1.1.31

Zadatak se rešava analogno zadatku 1.1.30

### Rešenje 1.1.32

```
1 #include <stdio.h>
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     float a, b, rez;
7     float min, max;
8
9     /* Ucitavaju se dva realna broja. */
10    printf("Unesite dva realna broja: ");
11    scanf("%f%f", &a, &b);
12
13    /* Racunaju se minimalna i maksimalna vrednost unetih brojeva. */
14    min = (a < b) ? a : b;
15    max = (a > b) ? a : b;
16
17    /* Racuna se vrednost rezultata. */
18    rez = (min + 0.5) / (1 + max * max);
19
20    /* Ispis rezultata. */
21    printf("Rezultat je %.2f\n", rez);
22
23    return 0;
24 }
```

## 2

# Kontrola toka

## 2.1 Naredbe grananja

**Zadatak 2.1.1** Napisati program koji ispisuje najmanji od tri uneta cela broja.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite tri cela broja: 5 18 -1  
| Najmanji: -1
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite tri cela broja: 0 43 16  
| Najmanji: 0
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite tri cela broja: 3 3 3  
| Najmanji: 3
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite tri cela broja: -5 -5 -5  
| Najmanji: -5
```

[Rešenje 2.1.1]

**Zadatak 2.1.2** Napisati program koji za uneti realan broj ispisuje njegovu apsolutnu vrednost zaokruženu na dve decimale.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite jedan realan broj: 7.42  
| Apsolutna vrednost: 7.42
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite jedan realan broj: -562.428  
| Apsolutna vrednost: 562.43
```

## 2 Kontrola toka

---

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan realan broj: 0  
|| Apsolutna vrednost: 0.00
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan realan broj: 52  
|| Apsolutna vrednost: 52.00
```

[Rešenje 2.1.2]

**Zadatak 2.1.3** Napisati program koji za uneti ceo broj ispisuje njegovu recipročnu vrednost zaokruženu na četiri decimale. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan ceo broj: 22  
|| Recipročna vrednost: 0.0455
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan ceo broj: -9  
|| Recipročna vrednost: -0.1111
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan ceo broj: 0  
|| Greška: nedozvoljeno deljenje nulom.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan ceo broj: 57298  
|| Recipročna vrednost: 0.0000
```

[Rešenje 2.1.3]

**Zadatak 2.1.4** Napisati program koji učitava tri cela broja i ispisuje zbir pozitivnih.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja: 1 3 -6  
|| Zbir pozitivnih: 4
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja: -15 81 0  
|| Zbir pozitivnih: 81
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja: -719 -48 -123  
|| Zbir pozitivnih: 0
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite tri cela broja: 16 2 576  
|| Zbir pozitivnih: 594
```

[Rešenje 2.1.4]

**Zadatak 2.1.5** U prodavnici je organizovana akcija da svaki kupac dobije najjeftiniji od tri artikla za jedan dinar. Napisati program koji za unete cene

tri artikla izračunava ukupnu cenu, kao i koliko dinara se uštedi zahvaljujući popustu. Cene artikala su celi brojevi. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite tri cene: 35 125 97  
Cena sa popustom: 223 din  
Usteda: 34 din
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:  
Unesite tri cene: 1034 15 25  
Cena sa popustom: 1060 din  
Usteda: 14 din
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:  
Unesite tri cene: 500 500 500  
Cena sa popustom: 1001 din  
Usteda: 499 din
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:  
Unesite tri cene: 247 -133 126  
Greska: neispravan unos cene.
```

[Rešenje 2.1.5]

**Zadatak 2.1.6** Napisati program koji za uneto vreme u formatu *sat:minut* ispisuje koliko je sati i minuta ostalo do ponoći. Broj sati treba da bude iz intervala  $[0, 24)$ , a broj minuta iz intervala  $[0, 60)$ . U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite vreme: 18:19  
Do ponoci: 5 sati i 41 minuta
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:  
Unesite vreme: 23:7  
Do ponoci: 0 sati i 53 minuta
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:  
Unesite vreme: 24:20  
Greska: neispravan unos vremena.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:  
Unesite vreme: 14:0  
Do ponoci: 10 sati i 0 minuta
```

[Rešenje 2.1.6]

**Zadatak 2.1.7** Napisati program koji za unetu godinu ispisuje da li je prestupna. Godina je neoznačen ceo broj.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite godinu: 2016  
Godina je prestupna.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:  
Unesite godinu: 1997  
Godina nije prestupna.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite godinu: 2000  
|| Godina je prestupna.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite godinu: 1900  
|| Godina nije prestupna.
```

[Rešenje 2.1.7]

**Zadatak 2.1.8** Napisati program koji za učitani karakter ispisuje uneti karakter i njegov ASCII kod. Ukoliko je uneti karakter malo (veliko) slovo, ispisati i odgovarajuće veliko (malo) slovo i njegov ASCII kod.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karakter: 0  
|| Uneti karakter: 0  
|| ASCII kod: 48
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karakter: ?  
|| Uneti karakter: ?  
|| ASCII kod: 63
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karakter: A  
|| Uneti karakter: A  
|| ASCII kod: 65  
|| Odgovarajuće malo slovo: a  
|| ASCII kod: 97
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karakter: v  
|| Uneti karakter: v  
|| ASCII kod: 118  
|| Odgovarajuće veliko slovo: V  
|| ASCII kod: 86
```

[Rešenje 2.1.8]

**Zadatak 2.1.9** Napisati program koji učitava 3 karaktera od kojih prvi mora biti cifra i ispisuje zbir ASCII kodova svih unetih karaktera, pomnožen sa vrednoscu unete cifre. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: *Karakteri koji se unose su razmaknuti belinama.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karaktere: 1 A c  
|| Rezultat: 213
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karaktere: 2 a _  
|| Rezultat: 484
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karaktere: 4 3 9  
|| Rezultat: 640
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite karaktere: B 6 (  
|| Greska: prvi karakter nije cifra.
```

[Rešenje 2.1.9]

**Zadatak 2.1.10** Napisati program koji učitava 3 karaktera i ispisuje proizvod svih karaktera koji su cifre. Ukoliko među unetim karakterima nema cifara, program treba da ispise odgovarajuću poruku. NAPOMENA: *Karakter koji se unose su razmaknuti belinama.*

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: A 5 3  
| Proizvod cifara: 15
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: k ! m  
| Medju unetim karakterima nema cifara.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: 9 9 9  
| Proizvod cifara: 729
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: a 8 0  
| Proizvod cifara: 0
```

[Rešenje 2.1.10]

**Zadatak 2.1.11** Kasir unosi šifru artikla koja se zadaje kao 3 spojena karaktera koji mogu biti mala slova, velika slova ili cifre. U kasi, sve šifre su zapisane malim slovima i ciframa. Napisati program koji unos kasira konvertuje u unos koji je odgovarajući za kasu, tj. koji sva velika slova pretvara u odgovarajuća mala, a ostale karaktere ne menja. U slučaju neispravnog unosa šifre, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite sifru: aBc  
| abc
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite sifru: a?!  
| Greška: ? je neispravan karakter.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: 5A5  
| 5a5
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite karaktere: 123  
| 123
```

[Rešenje 2.1.11]

**Zadatak 2.1.12** Napisati program koji za uneti četvorocifreni broj ispisuje njegovu najveću cifru. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 6835  
|| Najveca cifra je: 8
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 7777  
|| Najveca cifra je: 7
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 238  
|| Greska: niste uneli cetvorocifreni broj.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: -2002  
|| Najveca cifra je: 2
```

[Rešenje 2.1.12]

**Zadatak 2.1.13** Broj je Armstrongov ako je jednak zbiru kubova svojih cifara. Napisati program koji za dati trocifren broj proverava da li je Armstrongov. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite trocifreni broj: 153  
|| Broj je Armstrongov.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite trocifreni broj: 111  
|| Broj nije Armstrongov.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite trocifreni broj: 84  
|| Greska: niste uneli trocifreni broj.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite trocifreni broj: 371  
|| Broj je Armstrongov.
```

[Rešenje 2.1.13]

**Zadatak 2.1.14** Napisati program koji ispisuje proizvod parnih cifara unetog četvorocifrenog broja. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 8123  
|| Proizvod parnih cifara: 16
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 3579  
|| Nema parnih cifara.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: 288  
|| Greska: niste uneli cetvorocifreni broj.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite cetvorocifreni broj: -1234  
|| Proizvod parnih cifara: 8
```



[Rešenje 2.1.14]

**Zadatak 2.1.15** Napisati program koji učitava četvorocifreni broj i ispisuje broj koji se dobija kada se unetom broju razmene najmanja i najveća cifra. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci. NAPOMENA: *U slučaju da se najmanja ili najveća cifra pojavljuju na više pozicija, uzeti prvo pojavljivanje sa, gledajući sa desna na levo.*

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: 2863
|| Rezultat: 8263
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: 1192
|| Rezultat: 1912
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: 247
|| Greska: niste uneli četvorocifreni broj.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: -4239
|| Rezultat: -4932
```

[Rešenje 2.1.15]

**Zadatak 2.1.16** Napisati program koji za uneti četvorocifreni broj proverava da li su njegove cifre uređene neopadajuće, nerastuće ili nisu uređene i štampa odgovarajuću poruku. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: 1389
|| Cifre su uredjene neopadajuće.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: -9622
|| Cifre su uredjene nerastuće.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: 88
|| Greska: niste uneli četvorocifreni broj.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:
|| Unesite četvorocifreni broj: 6792
|| Cifre nisu uredjene.
```

[Rešenje 2.1.16]

**Zadatak 2.1.17** Napisati program koji ispituje da li se tačke  $A(x_1, y_1)$  i  $B(x_2, y_2)$  nalaze u istom kvadrantu i ispisuje odgovor DA ili NE. Koordinate tačaka su realni brojevi jednostruke tačnosti.

## 2 Kontrola toka

---

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 1.5 6
Unesite koordinate tacke B: 2.33 9.8
DA
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: -3 6
Unesite koordinate tacke B: 0.33 -5
NE
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 0 -6
Unesite koordinate tacke B: -1 -99.66
DA
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 3 -6
Unesite koordinate tacke B: -0.33 0
NE
```

[Rešenje 2.1.17]

**Zadatak 2.1.18** Napisati program koji ispituje da li se tačke  $A(x_1, y_1)$ ,  $B(x_2, y_2)$  i  $C(x_3, y_3)$  nalaze na istoj pravoj i ispisuje odgovor DA ili NE.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 1.5 6
Unesite koordinate tacke B: -2.5 -10
Unesite koordinate tacke C: 3 12
DA
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: -1.5 3
Unesite koordinate tacke B: -0.4 9.8
Unesite koordinate tacke C: 2 3
NE
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 1.55 6
Unesite koordinate tacke B: -8.4 9.8
Unesite koordinate tacke C: 5 4.682412
DA
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 5.5 3.5
Unesite koordinate tacke B: 5.5 3.5
Unesite koordinate tacke C: 5.5 3.5
DA
```

### Primer 5

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 1 2
Unesite koordinate tacke B: 1 2
Unesite koordinate tacke C: -56 1.3
DA
```

### Primer 6

```
INTERAKCIJA SA PROGRAMOM:
Unesite koordinate tacke A: 3.4 3.5
Unesite koordinate tacke B: -10 -1
Unesite koordinate tacke C: -10 -1
DA
```

[Rešenje 2.1.18]

**Zadatak 2.1.19** Napisati program za rad sa intervalima. Za dva celobrojna intervala  $[a1, b1]$  i  $[a2, b2]$ , program treba da odredi:

- a) dužinu preseka datih intervala
- b) presečni interval datih intervala
- c) dužinu prave koju pokrivaju dati intervali
- d) najmanji interval koji sadrži date intervale.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite a1, b1, a2 i b2: 2 9 4 11
Duzina preseka: 5
Presecni interval: [4,9]
Duzina koju pokrivaju: 9
Najmanji interval: [2, 11]
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite a1, b1, a2 i b2: 1 2 10 13
Duzina preseka: 0
Presecni interval: prazan
Duzina koju pokrivaju: 4
Najmanji interval: [1, 13]
```

[Rešenje 2.1.19]

**Zadatak 2.1.20** Napisati program koji za unete koeficijente kvadratne jednačine ispisuje koliko realnih rešenja jednačina ima i ako ih ima, ispisuje ih zaokružene na dve decimale.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite koeficijente A, B i C: 1 3 2
Jednacina ima dva razlicita realna resenja:
-1.00 i -2.00
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite koeficijente A, B i C: 1 1 1
Jednacina nema resenja.
```

[Rešenje 2.1.20]

**Zadatak 2.1.21** U nizu 12345678910111213....9899 ispisani su redom brojevi od 1 do 99. Napisati program koji za uneti ceo broj  $k$  ( $1 \leq k \leq 189$ ) ispisuje cifru koja se nalazi na  $k$ -toj poziciji datog niza. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:
Unesite k: 13
Na 13-toj poziciji je broj 1.
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:
Unesite k: 105
Na 105-toj poziciji je broj 7.
```

### Primer 3

```
INTERAKCIJA SA PROGRAMOM:
Unesite k: 200
Greska: neispravan unos pozicije.
```

### Primer 4

```
INTERAKCIJA SA PROGRAMOM:
Unesite k: 10
Na 10-toj poziciji je broj 1.
```

[Rešenje 2.1.21]

**Zadatak 2.1.22** Data je funkcija  $f(x) = 2 \cdot \cos(x) - x^3$ . Napisati program koji za učitane vrednosti realne promenljive  $x$  i vrednosti celobrojne promenljive  $k$  koje može biti 1, 2 ili 3 izračunava vrednost funkcije  $F(x, k) = f(f(f(\dots f(x))))$  gde je funkcija  $f$  primenjena  $k$ -puta i ispisuje je zaokruženu na dve decimale. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite redom x i k: 2.31 2  
|| F(2.31, 2)=2557.52
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite redom x i k: 12 1  
|| F(12, 1)=-1726.31
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite redom x i k: 2.31 0  
|| Greska: nedozvoljena vrednost za k.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite redom x i k: 1 3  
|| F(1, 3)=-8.74
```

[Rešenje 2.1.22]

**Zadatak 2.1.23** Napisati program koji za uneti redni broj dana u nedelji ispisuje ime odgovarajućeg dana. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 4  
|| U pitanju je: cetvrtak
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 7  
|| U pitanju je: nedelja
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 8  
|| Greska: neispravan unos dana.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite broj: 2  
|| U pitanju je: utorak
```

[Rešenje 2.1.23]

**Zadatak 2.1.24** Napisati program koji za uneti karakter ispituje da li je samoglasnik ili ne.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan karakter: A  
|| Uneti karakter je samoglasnik.
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan karakter: i  
|| Uneti karakter je samoglasnik.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan karakter: f  
|| Uneti karakter nije samoglasnik.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite jedan karakter: 4  
|| Uneti karakter nije samoglasnik.
```

[Rešenje 2.1.24]

**Zadatak 2.1.25** Napisati program koji učitava dva cela broja i jedan od karaktera +, -, \*, / ili % i ispisuje vrednost izraza dobijenog primenom date operacije na date argumente. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite izraz: 8 - 11  
|| Rezultat je: -3
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite izraz: 14 / 0  
|| Greska: deljenje nulom.
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite izraz: 5 ? 7  
|| Greska: nepoznat operator.
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite izraz: 19 / 5  
|| Rezultat je: 3
```

[Rešenje 2.1.25]

**Zadatak 2.1.26** Napisati program koji za uneti datum u formatu *dan.mesec.* ispisuje godišnje doba kojem pripadaju. NAPOMENA: *Podrazumevati da je unos korektan.*

### Primer 1

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 14.10.  
|| jesen
```

### Primer 2

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 2.8.  
|| leto
```

### Primer 3

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 27.2.  
|| zima
```

### Primer 4

```
|| INTERAKCIJA SA PROGRAMOM:  
|| Unesite dan i mesec: 19.5.  
|| prolece
```

[Rešenje 2.1.26]

**Zadatak 2.1.27** Napisati program koji za unetu godinu i mesec ispisuje naziv meseca kao i koliko dana ima u tom mesecu te godine. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite godinu: 2018  
Unesite mesec: 1  
Januar, 31 dan
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite godinu: 2000  
Unesite mesec: 2  
Februar, 29 dana
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite godinu: 2018  
Unesite mesec: 13  
Greska: neispravan unos meseca.
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite godinu: 1998  
Unesite mesec: 2  
Februar, 28 dana
```

[Rešenje 2.1.27]

**Zadatak 2.1.28** Napisati program koji za uneti datum u formatu *dan.mesec.godina*. proverava da li je korektan.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite datum: 25.11.1983.  
Datum je korektan.
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite datum: 1.17.2004.  
Datum nije korektan.
```

[Rešenje 2.1.28]

**Zadatak 2.1.29** Napisati program koji za korektno unet datum u formatu *dan.mesec.godina*. ispisuje datum prethodnog dana.

*Primer 1*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite datum: 30.4.2008.  
Prethodni datum: 29.4.2008.
```

*Primer 2*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite datum: 1.12.2005.  
Prethodni datum: 30.11.2005.
```

*Primer 3*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite datum: 1.1.2019.  
Prethodni datum: 31.12.2018.
```

*Primer 4*

```
INTERAKCIJA SA PROGRAMOM:  
Unesite datum: 10.12.2015.  
Prethodni datum: 9.11.2015.
```

[Rešenje 2.1.29]

**Zadatak 2.1.30** Napisati program koji za korektno unet datum u formatu *dan.mesec.godina.* ispisuje datum narednog dana.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite datum: 30.4.2008.  
| Naredni datum: 1.5.2008.
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite datum: 1.12.2005.  
| Naredni datum: 2.12.2005.
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite datum: 31.12.2008.  
| Naredni datum: 1.1.2009.
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite datum: 5.5.2005.  
| Naredni datum: 6.5.2005.
```

[Rešenje 2.1.30]

**Zadatak 2.1.31** Korisnik unosi tri cela broja  $P$ ,  $Q$  i  $R$  i dva karaktera,  $op1$  i  $op2$ . Ovi karakteri predstavljaju operacije nad unetim brojevima i imaju naredno značenje:

- karakter **k** predstavlja logičku konjukciju
- karakter **d** predstavlja logičku disjunkciju
- karakter **m** predstavlja relaciju manje
- karakter **v** predstavlja relaciju veće

Program treba da sračuna vrednost izraza  $P \text{ op1 } Q \text{ op2 } R$  i da ga ispiše. U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

*Primer 1*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite izraz: 0 k 1 m 2  
| Vrednost izraza: 1
```

*Primer 2*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite izraz: -3 d -1 k 0  
| Vrednost izraza: 0
```

*Primer 3*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite izraz: -3 k 1 d 0  
| Vrednost izraza: 1
```

*Primer 4*

```
|| INTERAKCIJA SA PROGRAMOM:  
| Unesite izraz: -3 m -1 m 100  
| Vrednost izraza: 1
```

[Rešenje 2.1.31]

\* **Zadatak 2.1.32** Program učitava jedan karakter i osam realnih brojeva koji predstavljaju koordinate četiri tačke:  $A(x_1, y_1)$ ,  $B(x_2, y_2)$ ,  $C(x_3, y_3)$ ,  $D(x_4, y_4)$ . Na osnovu unetog karaktera ispisuje se odgovarajuća poruka na standardni izlaz:

- ukoliko je uneti karakter  $k$  - proverava da li su date tačke temena pravougao-  
nika čije su stranice paralelne koordinatnim osama i u slučaju da jesu, is-  
pisuje vrednost obima datog pravougaoika. Možemo podrazumevati da će  
korisnik koordinate tačaka unosi redom  $A, B, C, D$ , pri čemu  $ABCD$  opisuje  
pravougaoik čije su stranice  $AB, BC, CD, DA$ , a dijagonale  $AC$  i  $BD$ . Na  
primer, tačke  $(1, 1), (2, 1), (2, 2), (1, 2)$  čine pravougaoik čije su stranice pa-  
ralelne koordinatnim osama i čiji je obim 4 a tačke  $(1, 1), (2, 2), (3, 3), (4, 4)$   
ne čine pravougaoik.
- ukoliko je uneti karakter  $h$  - proverava da li su unete tačke kolinearne i  
ukoliko jesu, ispisuje jednačinu prave kojoj pripadaju. Na primer, tačke  
 $(1, 2), (2, 3), (3, 4), (4, 5)$  su kolinearne i pripadaju pravoj  $y = x + 1$ , tačke  
 $(1, 1), (1, 2), (1, 3), (1, 4)$  su kolinearne i pripadaju pravoj  $x = 1$ , a tačke  
 $(1, 1), (2, 1), (2, 2), (1, 2)$  nisu kolinearne.
- ukoliko je uneti karakter  $j$  - Kramerovim pravilom proverava da li je sistem  
jednačina  $x_1 * p + x_2 * q = x_4 - x_3$ ,  $y_1 * p + y_2 * q = y_4 - y_3$  određen, neodređen  
ili nema rešenja, i u slučaju da je određen ispisuje rešenja.

[Rešenje 2.1.32]

\* **Zadatak 2.1.33** Polje šahovske table se definiše parom prirodnih brojeva  
ne većih od 8: prvi se odnosi na red, drugi na kolonu. Ako su dati takvi parovi,  
napisati program koji proverava:

- a) da li su polja  $(k, l)$  i  $(m, n)$  iste boje
- b) da li kraljica sa  $(k, l)$  ugrožava polje  $(m, n)$
- c) da li konj sa  $(k, l)$  ugrožava polje  $(m, n)$

Pretpostaviti da je polje  $(1,1)$  crno i da predstavlja donji levi ugao sahovske table.  
U slučaju neispravnog unosa, ispisati odgovarajuću poruku o grešci.

### Primer 1

```
INTERAKCIJA SA PROGRAMOM:  
Unesite (k,l): 1 1  
Unesite (m,n): 2 2  
Polja su iste boje.  
Kraljica sa (1,1) ugrozava (2,2).  
Konj sa (1,1) ne ugrozava (2,2).
```

### Primer 2

```
INTERAKCIJA SA PROGRAMOM:  
Unesite (k,l): 1 1  
Unesite (m,n): 3 2  
Polja su razlicite boje.  
Kraljica sa (1,1) ne ugrozava (3,2).  
Konj sa (1,1) ugrozava (3,2).
```



*Primer 3*

```

INTERAKCIJA SA PROGRAMOM:
Unesite (k,l): 5 4
Unesite (m,n): 3 3
Polja su razlicite boje.
Kraljica sa (5,4) ne ugrozava (3,3).
Konj sa (5,4) ugrozava (3,3).

```

*Primer 4*

```

INTERAKCIJA SA PROGRAMOM:
Unesite (k,l): 0 1
Unesite (m,n): 3 9
Greska: neispravna pozicija.

```

[Rešenje 2.1.33]

## 2.2 Rešenja

### Rešenje 2.1.1

```

1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int a, b, c, najmanji;
7
8      /* Ucitavaju se ulazne vrednosti. */
9      printf("Unesite tri cela broja: ");
10     scanf("%d%d%d", &a, &b, &c);
11
12     /* Najmanji broj se inicijalizuje na vrednost prvog broja. */
13     najmanji = a;
14
15     /* Ako je vrednost drugog broja manji od vrednosti tekuceg
16        minimuma, vrednost minimuma se azurira. */
17     if (b < najmanji)
18         najmanji = b;
19
20     /* Postupak se ponavlja za treci broj. */
21     if (c < najmanji)
22         najmanji = c;
23
24     /* Ispis rezultata. */
25     printf("Najmanji: %d\n", najmanji);
26
27     return 0;
28 }

```

### Rešenje 2.1.2

```
1 #include<stdio.h>
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     float x, apsolutno_x;
7
8     /* Ucitava se vrednost broja. */
9     printf("Unesite jedan realan broj:");
10    scanf("%f", &x);
11
12    /* Racuna se apsolutna vrednost unetog broja. */
13    apsolutno_x = x;
14    if (x < 0)
15        apsolutno_x = -x;
16
17    /* Ispis rezultata. */
18    printf("Apsolutna vrednost: %.2f\n", apsolutno_x);
19
20    /* II nacin: koriscenjem funkcije fabs cija se deklaracija nalazi
21       u zaglavlju math.h: apsolutno_x=fabs(x); */
22
23    return 0;
24 }
```

### Rešenje 2.1.3

```
1 #include <stdio.h>
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     int x;
7     float reciprocno_x;
8
9     /* Ucitavanje vrednosti broja x. */
10    printf("Unesite jedan ceo broj:");
11    scanf("%d", &x);
12
13    /* Vrsi se provera ispravnosti ulaznih podataka. Napomena: za
14       razliku od izlaza iz programa sa kodom 0 (return 0;) koji
15       služi kao indikator da se program završio uspešno, izlaz iz
16       programa sa izlaznim kodom koji se razlikuje od nule služi
17       kao indikator da je pri izvršavanju programa doslo do neke
18       greske. */
19    if (x == 0) {
20        printf("Greska: nedozvoljeno deljenje nulom.\n");
21        return -1;
22    }
23 }
```

```
25  /* Racuna se recipročna vrednost. */  
    reciprocno_x = 1.0 / x;  
  
27  /* Ispis rezultata. */  
    printf("Recipročna vrednost: %.4f\n", reciprocno_x);  
  
29  return 0;  
31 }
```

### Rešenje 2.1.4

```
1  #include<stdio.h>  
  
3  int main()  
{  
5  /* Deklaracija potrebnih promenljivih. */  
    int a, b, c, suma;  
  
7  /* Ucitavaju se ulazne vrednosti. */  
9  printf("Unesite tri cela broja:");  
    scanf("%d%d%d", &a, &b, &c);  
  
11 /* Pocetna vrednost sume se postavlja na 0. */  
13 suma = 0;  
  
15 /* Na sumu se dodaju vrednosti onih brojeva cija je vrednost  
    pozitivna. Uvecavanje je moguće uraditi na dva načina:  
17     I način: suma = suma + vrednost;  
     II način: suma += vrednost; */  
19 if (a > 0)  
    suma = suma + a;  
  
21 if (b > 0)  
23     suma += b;  
  
25 if (c > 0)  
    suma += c;  
  
27 /* Ispis rezultata. */  
29 printf("Zbir pozitivnih: %d\n", suma);  
  
31 return 0;  
}
```

### Rešenje 2.1.5

```
1  #include <stdio.h>  
  
3  int main()
```

```
{
5  /* Deklaracija potrebnih promenljivih. */
   int a, b, c;
7  int najjeftiniji;
   int cena_bez_popusta, cena_sa_popustom;

9

   /* Ucitavaju se vrednosti cena. */
11 printf("Unesite tri cene: ");
   scanf("%d%d%d", &a, &b, &c);

13

   /* Vrsi se provera ispravnosti ulaznih podataka. */
15 if (a <= 0 || b <= 0 || c <= 0) {
       printf("Greska: neispravan unos cene.");
17     return -1;
   }

19

   /* Racuna se vrednost najjeftinijeg artikla. */
21 najjeftiniji = a;

23 if (b < najjeftiniji)
       najjeftiniji = b;
25
27 if (c < najjeftiniji)
       najjeftiniji = c;

29 /* Racunaju se cene sa i bez popusta. */
   cena_bez_popusta = a + b + c;
31 cena_sa_popustom = cena_bez_popusta - najjeftiniji + 1;

33 /* Ispis rezultata. */
   printf("Cena sa popustom: %d din\n", cena_sa_popustom);
35 printf("Usteda: %d din\n", cena_bez_popusta - cena_sa_popustom);

37 return 0;
}
```

### Rešenje 2.1.6

```
1 #include<stdio.h>

3 int main()
{
5  /* Deklaracija potrebnih promenljivih. */
   int sati, minuti;
7  int preostali_sati, preostali_minuti;

9

   /* Ucitavaju se podaci o vremenu. Napomena: Vreme se zadaje u
       formatu sat:minut. Iz tog razloga je i odgovarajuci format u
11   funkciji scanf %d:%d. */
   printf("Unesite vreme: ");
13   scanf("%d:%d", &sati, &minuti);
}
```

```

15  /* Vrsi se provera ispravnosti ulaznih podataka. */
16  if (sati > 24 || sati < 0 || minuti > 59 || minuti < 0) {
17      printf("Greska: neispravan unos vremena.\n");
18      return -1;
19  }

21  /* Racuna se preostalo vreme. */
22  preostali_sati = 24 - sati - 1;
23  preostali_minuti = 60 - minuti;

25  if (preostali_minuti == 60) {
26      /* Uvecavanje vrednosti broja za 1 se moze uraditi na vise
27         nacina. Neki od njih su:
28         broj = broj + 1;
29         broj += 1;
30         broj++; */
31      preostali_sati++;
32      preostali_minuti = 0;
33  }

35  /* Ispis rezultata. */
36  printf("Do ponoci: %d sati i %d minuta\n",
37         preostali_sati, preostali_minuti);

39  return 0;
}

```

### Rešenje 2.1.7

```

1  #include <stdio.h>

3  int main()
4  {
5      /* Deklaracija potrebne promenljive. */
6      unsigned int x;

7      /* Ucitava se vrednost godine. */
8      printf("Unesite godinu:");
9      scanf("%u", &x);

11     /* Proverava se da li je godina prestupna ili ne i ispisuje se
12        odgovarajuca poruka. */
13     if ((x % 4 == 0 && x % 100 != 0) || x % 400 == 0)
14         printf("Godina je prestupna.\n");
15     else
16         printf("Godina nije prestupna.\n");

17     return 0;
18 }

```

### Rešenje 2.1.8

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija karakterske promenljive. */
6      char c;
7
8      /* Ucitava se jedan karakter. */
9      printf("Unesite karakter: ");
10     scanf("%c", &c);
11
12     /* Ispis karaktera i vrednosti njegovog ASCII koda. */
13     printf("Uneti karakter: %c\n", c);
14     printf("ASCII kod: %d\n", c);
15
16     /* Karakteri koji odgovaraju velikim slovima su u ASCII tablici
17        smesteni sekvencijalno. Na primer, ASCII kod karaktera 'A' je
18        65, 'B' je 66, ..., 'Z' je 90. Isto vazi i za mala slova: 'a'
19        je 97, 'b' je 98, ..., 'z' je 122.
20
21        Oдавде, ako se vrsi provera da li je neki karakter veliko
22        slovo, dovoljno je proveriti da li se njegov ASCII kod nalazi
23        izmedju ASCII kodova slova 'A' i slova 'Z'.
24
25        Dodatno, moze se primetiti da je razlika izmedju ASCII koda
26        svakog malog i odgovarajuceg velikog slova konstanta koja ima
27        vrednost 'a'-'A', sto je isto sto i 'b'-'B', itd. Zbog toga,
28        ako je potrebno od velikog slova dobiti malo, onda je
29        dovoljno ASCII kodu velikog slova dodati pomenutu konstantu.
30        Za mala slova, vazi obrnuto - da bi se dobilo veliko slovo,
31        ova konstanta se oduzima. */
32
33     if (c >= 'A' && c <= 'Z') {
34         printf("Odgovarajuće malo slovo: %c\n", c + ('a' - 'A'));
35         printf("ASCII kod: %d\n", c + ('a' - 'A'));
36     }
37
38     if (c >= 'a' && c <= 'z') {
39         printf("Odgovarajuće veliko slovo: %c\n", c - ('a' - 'A'));
40         printf("ASCII kod: %d\n", c - ('a' - 'A'));
41     }
42
43     return 0;
44 }
```

### Rešenje 2.1.9

```
1  #include <stdio.h>
```

```
#include <ctype.h>

3
int main()
5
{
    /* I nacin ucitavanja ulaza: koriscenjem funkcije getchar()
    Funkcija getchar cita jedan karakter sa ulaza i vraca njegov
    ASCII kod. Napomena: beline su takodje karakteri i nece
    7
    automatski biti preskocene. Iz tog razloga se getchar poziva 5
    puta u ovom primeru. Posto je poznato da su drugi i cetvrti
    9
    karakter beline, nema potrebe da se cuva povratna vrednost tih
    poziva. */
    11
    int c1, c2, c3;
    printf("Unesite karaktere: ");
    13
    c1 = getchar();
    getchar();
    15
    c2 = getchar();
    getchar();
    17
    c3 = getchar();

    /* II nacin ucitavanja ulaza: koriscenjem funkcije scanf()
    Beline se navode kao deo ocekivanog formata ulaza.
    21
    char c1, c2, c3;
    scanf("%c %c %c", &c1, &c2, &c3); */
    23

    /* Pogresan nacin ucitavanja ulaza:
    scanf("%c%c%c", &c1, &c2, &c3);
    27
    U ovom slucaju ce u c1 biti upisan prvi karakter, u c2
    belina i u c3 drugi karakter. */
    29

    /* Karakteri koji predstavljaju cifre su u ASCII tablici takodje
    smesteni sekvencijalno. Na primer, '0' ima ASCII kod 48, '1'
    33
    49, ..., '9' ima ASCII kod 57.

    Oдавде, ako se proverava da li je karakter cifra, dovoljno je
    37
    proveriti da li se njegov ASCII kod nalazi izmedju '0' i '9'.

    Dodatno, ako je potrebno izracunati dekadnu vrednost karaktera
    41
    koji je cifra, dovoljno je od ASCII koda tog karaktera,
    oduzeti ASCII kod karaktera '0'. Na primer, '4'-'0' = 52 - 48
    43
    = 4. */
    if (c1 < '0' || c1 > '9') {
    45
        printf("Greska: prvi karakter nije cifra.");
        return -1;
    47
    }

    /* Racuna se zbir ASCII kodova. */
    49
    int zbir = c1 + c2 + c3;

    /* Racuna se vrednost prve cifre. */
    51
    int vrednost_prve_cifre = c1 - '0';
    53
```

## 2 Kontrola toka

---

```
55  /* Ispis rezultata. */
    printf("Rezultat: %d\n", vrednost_prve_cifre * zbir);
57
    return 0;
59 }
```

### Rešenje 2.1.10

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      char c1, c2, c3;
7      unsigned int broj_cifara = 0;
8      unsigned int proizvod_cifara = 1;
9
10     /* Ucitavaju se tri karaktera. */
11     printf("Unesite karaktere: ");
12     scanf("%c %c %c", &c1, &c2, &c3);
13
14     /* Racuna se proizvod onih karaktera koji su cifre. */
15     if (c1 >= '0' && c1 <= '9') {
16         proizvod_cifara *= (c1 - '0');
17         broj_cifara++;
18     }
19
20     if (c2 >= '0' && c2 <= '9') {
21         proizvod_cifara *= (c2 - '0');
22         broj_cifara++;
23     }
24
25     if (c3 >= '0' && c3 <= '9') {
26         proizvod_cifara *= (c3 - '0');
27         broj_cifara++;
28     }
29
30     /* Ispis rezultata. */
31     if (broj_cifara == 0)
32         printf("Medju unetim karakterima nema cifara.\n");
33     else
34         printf("Proizvod cifara: %u\n", proizvod_cifara);
35
36     return 0;
37 }
```

### Rešenje 2.1.11



```
1 #include <stdio.h>
2 #include <ctype.h>
3
4 int main()
5 {
6     /* Deklaracija potrebnih promenljivih. */
7     int c1, c2, c3;
8
9     /* Ucitava se sifra artikla. */
10    printf("Unesite sifru: ");
11    c1 = getchar();
12    c2 = getchar();
13    c3 = getchar();
14
15    /* Funkcije islower, isupper i isdigit proveravaju da li je
16       prosledjeni karakter malo slovo, veliko slovo ili cifra.
17       Deklaracije ovih funkcija se nalaze u zaglavlju ctype.h.
18
19       Ukoliko prvi karakter nije ni malo slovo ni veliko slovo, ni
20       cifra, ispisuje se odgovarajuca poruka o gresci i izlazi se
21       iz programa. */
22    if (!islower(c1) && !isupper(c1) && !isdigit(c1)) {
23        printf("Greska: %c je neispravan karakter.\n", c1);
24        return -1;
25    }
26
27    /* Postupak se ponavlja za druga dva karaktera. */
28    if (!islower(c2) && !isupper(c2) && !isdigit(c2)) {
29        printf("Greska: %c je neispravan karakter.\n", c2);
30        return -1;
31    }
32
33    if (!islower(c3) && !isupper(c3) && !isdigit(c3)) {
34        printf("Greska: %c je neispravan karakter.\n", c3);
35        return -1;
36    }
37
38    /* Funkcija tolower(c) radi sledece: ako je c veliko slovo, kao
39       povratnu vrednost vraca odgovarajuce malo slovo, u suprotnom
40       vraca c. Dakle, tolower('A') je 'a', a tolower('6') = '6',...
41
42       Slicno, samo obrnuto, radi i funkcija toupper(c). Deklaracije
43       ovih funkcija se takodje nalaze u zaglavlju ctype.h. */
44    c1 = tolower(c1);
45    c2 = tolower(c2);
46    c3 = tolower(c3);
47
48    printf("%c%c%c\n", c1, c2, c3);
49
50    return 0;
51 }
```

```
}

```

### Rešenje 2.1.12

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     /* Deklaracija potrebnih promenljivih. */
7     int n;
8     char jedinica, desetica, stotina, hiljada, najveca_cifra;
9
10    /* Ucitava se cetvorocifreni broj. */
11    printf("Unesite cetvorocifreni broj: ");
12    scanf("%d", &n);
13
14    /* Da bi program radio ispravno i za negativne brojeve, uzima se
15       apsolutna vrednost broja n. */
16    n = abs(n);
17
18    /* Vrsi se provera ispravnosti ulaznih podataka. */
19    if (n < 1000 || n > 9999) {
20        printf("Greska: niste uneli cetvorocifreni broj.\n");
21        return -1;
22    }
23
24    /* Izdvajaju se cifre broja n. */
25    jedinica = n % 10;
26    desetica = (n / 10) % 10;
27    stotina = (n / 100) % 10;
28    hiljada = n / 1000;
29
30    /* Racuna se najveca cifra broja n. */
31    najveca_cifra = jedinica;
32
33    if (desetica > najveca_cifra)
34        najveca_cifra = desetica;
35
36    if (stotina > najveca_cifra)
37        najveca_cifra = stotina;
38
39    if (hiljada > najveca_cifra)
40        najveca_cifra = hiljada;
41
42
43    /* Ispis rezultata */
44    printf("Najveca cifra je: %d\n", najveca_cifra);
45
46    return 0;
47 }
```

## Rešenje 2.1.13

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     /* Deklaracija potrebnih promenljivih. */
7     int n;
8     char jedinica, desetica, stotina;
9
10    /* Ucitava se broj n. */
11    printf("Unesite trocifreni broj: ");
12    scanf("%d", &n);
13
14    /* Da bi program radio ispravno i za negativne brojeve, uzima se
15       apsolutna vrednost broja n. */
16    n = abs(n);
17
18    /* Vrsi se provera ispravnosti ulaznih podataka. */
19    if (n < 100 || n > 999) {
20        printf("Greska: niste uneli trocifreni broj.\n");
21        return -1;
22    }
23
24    /* Izdvajaju se cifre broja n. */
25    jedinica = n % 10;
26    desetica = (n / 10) % 10;
27    stotina = n / 100;
28
29    /* Ispis rezultata. */
30    if (n == jedinica * jedinica * jedinica +
31        desetica * desetica * desetica + stotina * stotina * stotina)
32        printf("Broj je Armstrongov.\n");
33    else
34        printf("Broj nije Armstrongov.\n");
35
36    return 0;
37 }
```

## Rešenje 2.1.14

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
```

```
/* Deklaracija potrebnih promenljivih. */
7 int n;
char jedinica, desetica, stotina, hiljada;
9 int broj_parnih, proizvod_parnih;

/* Ucitava se broj n. */
11 printf("Unesite cetvorocifreni broj: ");
13 scanf("%d", &n);

/* Da bi program radio ispravno i za negativne vrednosti, uzima
   se apsolutna vrednost broja n. */
15 n = abs(n);
17

/* Vrsi se provera ispravnosti ulaznih podataka. */
19 if (n < 1000 || n > 9999) {
21     printf("Greska: niste uneli cetvorocifreni broj.\n");
    return -1;
23 }

/* Izdvajaju se cifre broja n. */
25 jedinica = n % 10;
27 desetica = (n / 10) % 10;
stotina = (n / 100) % 10;
29 hiljada = n / 1000;

/* Inicijalizacija brojac i rezultata. */
31 broj_parnih = 0;
33 proizvod_parnih = 1;

/* Za svaku cifru se vrsi provera da li je parna i ukoliko jeste
   tekuci rezultat se mnozi sa tekucim cifrom. */
35 if (jedinica % 2 == 0) {
37     proizvod_parnih = proizvod_parnih * jedinica;
39     broj_parnih++;
}

41 if (desetica % 2 == 0) {
43     proizvod_parnih = proizvod_parnih * desetica;
    broj_parnih++;
45 }

47 if (stotina % 2 == 0) {
    proizvod_parnih = proizvod_parnih * stotina;
49     broj_parnih++;
}

51 if (hiljada % 2 == 0) {
53     proizvod_parnih = proizvod_parnih * hiljada;
    broj_parnih++;
55 }

57 /* Ispis rezultata. */
```

```

59     if (broj_parnih == 0) {
        printf("Nema parnih cifara.\n");
    } else {
61         printf("Proizvod parnih cifara: %d\n", proizvod_parnih);
    }
63
65     return 0;
}

```

### Rešenje 2.1.15

```

1  #include <stdio.h>
   #include <stdlib.h>
3
   int main()
5  {
   /* Deklaracija potrebnih promenljivih. */
7  int n, n_abs;
   char jedinica, desetica, stotina, hiljada;
9  int najveca, najmanja, stepen_najvece, stepen_najmanje;
   int rezultat;
11
   /* Ucitava se broj n. */
13  printf("Unesite cetvorocifreni broj: ");
   scanf("%d", &n);
15
   /* Da bi program radio ispravno i za negativne vrednosti, uzima
      se apsolutna vrednost broja n. */
17  n_abs = abs(n);
19
   /* Vrsi se provera ispravnosti ulaznih podataka. */
21  if (n_abs < 1000 || n_abs > 9999) {
       printf("Greska: niste uneli cetvorocifreni broj.\n");
23     return -1;
   }
25
   /* Izdvajaju se cifre broja n. */
27  jedinica = n_abs % 10;
   desetica = (n_abs / 10) % 10;
29  stotina = (n_abs / 100) % 10;
   hiljada = n_abs / 1000;
31
   /* Racunaju se najveca i najmanja cifra, kao i pozicija na kojoj
      se one nalaze. Radi lakseg izracunavanja, pozicija se pamti
      kao stepen broja 10. Na primer, pozicija cifre jedinica je 1,
35     cifre desetica 10, itd... */
   najveca = jedinica;
37  najmanja = jedinica;
   stepen_najvece = 1;
39  stepen_najmanje = 1;

```

```
41  if (desetica > najveca) {
43      najveca = desetica;
      stepen_najvece = 10;
  }
45  if (desetica < najmanja) {
      najmanja = desetica;
47      stepen_najmanje = 10;
  }
49
51  if (stotina > najveca) {
      najveca = stotina;
      stepen_najvece = 100;
53  }
55  if (stotina < najmanja) {
      najmanja = stotina;
      stepen_najmanje = 100;
57  }
59
61  if (hiljada > najveca) {
      najveca = hiljada;
      stepen_najvece = 1000;
  }
63  if (hiljada < najmanja) {
      najmanja = hiljada;
65      stepen_najmanje = 1000;
  }
67
69  /* Ideja: U broju 4179, najmanja cifra je 1 i njen stepen je 100,
      a najveca cifra je 0 i njen stepen je 1. Zamena mesta se vrši
      tako što se oduzme 9 i doda 1, a zatim oduzme 100 i doda 900. */
71  if (n > 0){
      rezultat = n - najveca * stepen_najvece
73                  + najmanja * stepen_najvece
                  - najmanja * stepen_najmanje
75                  + najveca * stepen_najmanje;
  }
77  else {
      rezultat = n + najveca * stepen_najvece
79                  - najmanja * stepen_najvece
                  + najmanja * stepen_najmanje
81                  - najveca * stepen_najmanje;
  }
83
85  /* Ispis rezultata. */
      printf("Rezultat: %d\n", rezultat);
87
      return 0;
  }
```

### Rešenje 2.1.16

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     /* Deklaracija potrebnih promenljivih. */
7     int n;
8     char jedinica, desetica, stotina, hiljada;
9
10    /* Ucitava se broj n. */
11    printf("Unesite cetvorocifreni broj: ");
12    scanf("%d", &n);
13
14    /* Da bi program radio ispravno i za negativne vrednosti, uzima
15       se apsolutna vrednost broja n. */
16    n = abs(n);
17
18    /* Vrsi se provera ispravnosti ulaznih podataka. */
19    if (n < 1000 || n > 9999) {
20        printf("Greska: niste uneli cetvorocifreni broj.\n");
21        return -1;
22    }
23
24    /* Izdvajaju se cifre broja n. */
25    jedinica = n % 10;
26    desetica = (n / 10) % 10;
27    stotina = (n / 100) % 10;
28    hiljada = n / 1000;
29
30    /* Ispis rezultata. */
31    if (hiljada <= stotina && stotina <= desetica
32        && desetica <= jedinica)
33        printf("Cifre su uredjene neopadajuce. \n");
34    else if (hiljada >= stotina && stotina >= desetica
35            && desetica >= jedinica)
36        printf("Cifre su uredjene nerastuce. \n");
37    else
38        printf("Cifre nisu uredjene.\n");
39
40    return 0;
41 }
```

### Rešenje 2.1.17

```
#include<stdio.h>
2
3 int main()
4 {
5     /* Deklaracija potrebnih promenljivih. */
6     float x1, y1, x2, y2;
```

```
8  /* Ucitavaju se koordinate tacaka A i B. */
   printf("Unesite koordinate tacke A: ");
10  scanf("%f%f", &x1, &y1);

12  printf("Unesite koordinate tacke B: ");
   scanf("%f%f", &x2, &y2);

14

16  /* Proverava se da li su obe tacke u istom kvadrantu. */
   if ((x1 >= 0 && y1 >= 0 && x2 >= 0 && y2 >= 0) ||
18       (x1 <= 0 && y1 >= 0 && x2 <= 0 && y2 >= 0) ||
       (x1 >= 0 && y1 <= 0 && x2 >= 0 && y2 <= 0) ||
       (x1 <= 0 && y1 <= 0 && x2 <= 0 && y2 <= 0)) {
20     printf("DA\n");
   } else {
22     printf("NE\n");
   }

24   return 0;
26 }
```

### Rešenje 2.1.18

```
1  #include<stdio.h>

3  int main()
   {
5     /* Deklaracija potrebnih promenljivih. */
       float x1, y1, x2, y2, x3, y3;
7     float k, n;

9     /* Ucitavaju se koordinate tacaka A, B i C. */
       printf("Unesite koordinate tacke A: ");
11    scanf("%f%f", &x1, &y1);

13    printf("Unesite koordinate tacke B: ");
       scanf("%f%f", &x2, &y2);

15    printf("Unesite koordinate tacke C: ");
       scanf("%f%f", &x3, &y3);

17

19    /* Ako su bilo koje dve tacke jednake, onda se sigurno sve tri
       nalaze na jednoj pravoj. */
21    if ((x1 == x2 && y1 == y2) ||
        (x1 == x3 && y1 == y3) || (x2 == x3 && y2 == y3)) {
23        printf("DA\n");
        return 0;
25    }

27    /* Odredjuju se koeficijent pravca k i odsecak na y osi n, prave
       y = k*x + n koja prolazi kroz tacke A i B. Napomena: u
```



```

29     slucaju kada je x2 jednako x1, ova prava je paralelna sa y
30     osom i k ima vrednost beskonacno, a n ima vrednost 0, tj.
31     jednacina prave je x = x1 (sto je isto sto i x = x2). Da bi se
32     izbeglo deljenje nulom (x2-x1), ovaj slucaj se posebno
33     obradjuje. */
34     if (x2 != x1) {
35         k = (y2 - y1) / (x2 - x1);
36         n = y1 - k * x1;
37         /* Proverava se da li tacka C pripada pravoj y=k*x + n na
38            kojoj se vec nalaze tacke A i B. */
39         if (y3 == k * x3 + n) {
40             printf("DA\n");
41         } else {
42             printf("NE\n");
43         }
44     } else {
45         /* Proverava se da li se i tacka C nalazi na pravoj x = x2. */
46         if (x3 == x2) {
47             printf("DA\n");
48         } else {
49             printf("NE\n");
50         }
51     }

52
53     /* II nacin: 3 tacke su kolinearne ako je:
54
55     |x1 y1 1 |
56     |x2 y2 1 | = 0
57     |x3 y3 1 |
58     odnosno, ako je:
59     x1*y2 + y1*x3 + x2*y3 - y1*x2 - x1*y3 - y2*x3 = 0
60
61     if(x1*y2 + y1*x3 + x2*y3 - y1*x2 - x1*y3 - y2*x3 == 0)
62         printf("DA\n");
63     else
64         printf("NE\n"); */
65
66     return 0;
67 }

```

### Rešenje 2.1.19

```

#include<stdio.h>

2
int main()
4 {
    /* Deklaracija potrebnih promenljivih. */
6     int a1, a2, b1, b2;

8     /* Ucitavaju se granice intervala. */
    printf("Unesite a1, b1, a2 i b2: ");

```

```

10  scanf("%d%d%d%d", &a1, &b1, &a2, &b2);

12  /* U zavisnosti od razlicitih polozaia dva intervala, racunaju se
    i ispisuju trazene vrednosti. */
14  if (a1 <= a2 && b1 >= a2) {
16      /* I slucaj: intervali se seku i [a1,b1] je pre [a2,b2]. */
17      printf("Duzina preseka:: %d\n", b1 - a2);
18      printf("Presecni interval: [%d, %d]\n", a2, b1);
19      printf("Duzina koju pokrivaju: %d\n", b2 - a1);
20      printf("Najmanji interval: [%d, %d]\n", a1, b2);
21  } else if (a2 <= a1 && b2 >= a1) {
22      /* II slucaj: intervali se seku i [a2,b2] je pre [a1,b1]. */
23      printf("Duzina preseka:: %d\n", b2 - a1);
24      printf("Presecni interval: [%d, %d]\n", a1, b2);
25      printf("Duzina koju pokrivaju: %d\n", b1 - a2);
26      printf("Najmanji interval: [%d, %d]\n", a2, b1);
27  } else if (a1 >= a2 && b1 <= b2) {
28      /* III slucaj: interval [a1,b1] se nalazi unutar [a2,b2]. */
29      printf("Duzina preseka:: %d\n", b1 - a1);
30      printf("Presecni interval: [%d, %d]\n", a1, b1);
31      printf("Duzina koju pokrivaju: %d\n", b2 - a2);
32      printf("Najmanji interval: [%d, %d]\n", a2, b2);
33  } else if (a2 >= a1 && b2 <= b1) {
34      /* IV slucaj: interval [a2,b2] se nalazi unutar [a1,b1]. */
35      printf("Duzina preseka:: %d\n", b2 - a2);
36      printf("Presecni interval: [%d, %d]\n", a2, b2);
37      printf("Duzina koju pokrivaju: %d\n", b1 - a1);
38      printf("Najmanji interval: [%d, %d]\n", a1, b1);
39  } else {
40      /* V slucaj: intervali su disjunktni. */
41      printf("Duzina preseka:: 0\n");
42      printf("Presecni interval: prazan\n");
43      printf("Duzina koju pokrivaju: %d\n", b1 - a1 + b2 - a2);
44      if (a1 < a2)
45          printf("Najmanji interval: [%d, %d]\n", a1, b2);
46      else
47          printf("Najmanji interval: [%d, %d]\n", a2, b1);
48  }

49  return 0;
50  }

```

### Rešenje 2.1.20

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      /* Deklaracija potrebnih promenljivih. */
7      float a, b, c;

```

```

9      float D;

11     /* Ucitavaju se koeficijenti kvadratne jednacine. */
12     printf("Unesite koeficijente A, B i C:");
13     scanf("%f%f%f", &a, &b, &c);

14     /* Racunaju se resenja jednacine u zavisnosti od vrednosti
15        koeficijenata a, b i c i ispisuje se odgovarajuci rezultat. */
16     if (a == 0) {
17         if (b == 0) {
18             if (c == 0) {
19                 /* Slucaj a==0 && b==0 && c==0: beskonacno mnogo resenja. */
20                 printf("Jednacina ima beskonacno mnogo resenja\n");
21             } else {
22                 /* Slucaj a==0 && b==0 && c!=0: nema resenja. */
23                 printf("Jednacina nema resenja\n");
24             }
25         } else {
26             /* Slucaj a=0 && b!=0: jedinstveno resenje. */
27             printf("Jednacina ima jedinstveno realno resenje %.2f\n",
28                   -c / b);
29         }
30     } else {
31         /* Slucaj a != 0: racuna se diskriminanta. */
32         D = b * b - 4 * a * c;

33         /* U zavisnosti od vrednosti diskriminante, ispisuje se
34            rezultat. */
35         if (D < 0) {
36             printf("Jednacina nema realnih resenja\n");
37         } else if (D > 0) {
38             printf("Jednacina ima dva realna resenja %.2f i %.2f\n",
39                   (-b + sqrt(D)) / (2 * a), (-b - sqrt(D)) / (2 * a));
40         } else {
41             printf("Jednacina ima jedinstveno realno resenje %.2f\n",
42                   -b / (2 * a));
43         }
44     }
45 }

46 return 0;
47 }

```

### Rešenje 2.1.21

```

1  #include <stdio.h>

3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int k, broj;
7

```

```
/* Ucitava se trazena pozicija. */
9 printf("Unesite k: ");
scanf("%d", &k);

11

/* Vrsi se provera ispravnosti ulaznih podataka. */
13 if (k < 1 || k > 189) {
    printf("Greska: neispravan unos pozicije.\n");
15     return -1;
}

17

/* Racuna se rezultat. */
19 if (k < 10) {
    /* I slucaj: trazi se jednocifreni broj. */
21     printf("Na %d-toj poziciji je broj %d.\n", k, k);
} else {
23     /* II slucaj: trazi se dvocifreni broj. */

25     /* Ideja: izracunati broj na koji pokazuje pozicija k. Zatim,
        ako je k parno, uzeti cifru desetica tog broja, a ako je k
27     neparno, uzeti cifru jedinica tog broja.

29     Na primer, za k=14 i k=15, broj koji se nalazi na ovim
        pozicijama je 12, pa u slucaju da je k=14, treba ispisati 1,
31     a u slucaju da je k=15, treba ispisati 2. */

33     /* Odredjivanje odgovarajuceg broja: Kada bi niz izgledao
        10111213...9899, za dato k, broj bi se dobio kao  $9 + k/2 + 1$ 
35     za neparne vrednosti k, odnosno  $9 + k/2$  za parne (dodaje se
        vrednost deset jer je prvi broj u nizu desetica.) Na primer:
37     k=1, broj =  $9 + 1/2 + 1 = 9 + 0 + 1 = 10$  k=2, broj =  $9 + 2/2$ 
        = 10 k=3, broj =  $9 + 3/2 + 1 = 9 + 1 + 1 = 11$  k=4, broj =  $9$ 
39     +  $4/2 = 11$  ... Posto ovde postoji i 9 pozicija ispred,
        potrebno je i njih uzeti u obzir - odatle: broj =  $9 +$ 
41      $(k-9)/2 + 1$  za neparne vrednosti k, odnosno broj =  $9 +$ 
         $(k-9)/2$  za parne vrednosti k. */
43     if (k % 2 != 0) {
        broj =  $9 + (k - 9) / 2$ ;
45     printf("Na %d-toj poziciji je broj %d.\n", k, broj % 10);
    } else {
47     broj =  $9 + (k - 9) / 2 + 1$ ;
        printf("Na %d-toj poziciji je broj %d.\n", k, broj / 10);
49     }
    }

51     return 0;
53 }
```

### Rešenje 2.1.22

```
1 #include <stdio.h>
#include <math.h>
```

```

3  int main()
5  {
    /* Deklaracija potrebnih promenljivih. */
7   float x, Fx;
    int k;

9   /* Ucitavaju se vrednosti x i k. */
11  printf("Unesite redom x i k: ");
    scanf("%f %d", &x, &k);

13  /* Vrsi se provera ispravnosti ulaznih podataka. */
15  if (k < 1 || k > 3) {
        printf("Greska: nedozvoljena vrednost za k.\n");
17     return 0;
    }

19  /* U zavisnosti od vrednosti k, data funkcija ce se izracunati
21     jednom, dva puta ili tri puta. */
    Fx = 2 * cos(x) - x * x * x;
23     if (k > 1)
        Fx = 2 * cos(Fx) - Fx * Fx * Fx;
25     if (k > 2)
        Fx = 2 * cos(Fx) - Fx * Fx * Fx;

27  /* Ispis rezultata. Napomena: ispis realnih brojeva sa %g
29     rezultuje ispisom na onaj broj decimala koliko sam broj ima.
        Dakle, broj 1 ce se ispisati kao 1, broj 2.33 kao 2.33, broj
31     0.9999 kao 0.9999. */
    printf("F(%g,%d)=%.2f\n", x, k, Fx);

33     return 0;
35 }

```

### Rešenje 2.1.23

```

1  #include <stdio.h>

3  int main()
4  {
5     /* Deklaracija potrebnih promenljivih. */
        int dan;

7     /* Ucitava se redni broj dana u nedelji. */
9     printf("Unesite broj: ");
        scanf("%d", &dan);

11

13     switch (dan) {
        case 1:
15         /* Ako dan ima vrednost 1, ispisuje se ponedeljak. */

```

```
    printf("ponedeljak\n");
17
    /* Ako se naredba break ne navede, izvršice se i sledeca
19    naredba, tj. ispis ce biti "ponedeljak utorak". */
    break;
21 case 2:
    /* Postupak se ponavlja i za ostale dane. */
23    printf("utorak\n");
    break;
25 case 3:
    printf("sreda\n");
27    break;
    case 4:
29    printf("cetvrtak\n");
    break;
31 case 5:
    printf("petak\n");
33    break;
    case 6:
35    printf("subota\n");
    break;
37 case 7:
    printf("nedelja\n");
39    break;
    default:
41    /* Ako vrednost promenljive dan nije ni jedna od vrednosti
        izmedju 1 i 7, onda je uneta vrednost neispravna. */
43    printf("Greska: neispravan unos dana.\n");
    }
45
    return 0;
47 }
```

### Rešenje 2.1.24

```
1  #include <stdio.h>
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
        char c;
7
8      /* Ucitava se jedan karakter. */
9      printf("Unesite jedan karakter:");
        scanf("%c", &c);
11
12     /* Proverava se da li je karakter c samoglasnik, tj. da li
13     odgovara nekom od sledecih karaktera: A,E,I,O,U,a,e,i,o,u. */
        switch (c) {
15         case 'A':
            case 'E':
```

```
17     case 'I':
18     case 'O':
19     case 'U':
20     case 'a':
21     case 'e':
22     case 'i':
23     case 'o':
24     case 'u':
25         printf("Uneti karakter je samoglasnik.\n");
26         break;
27     default:
28         printf("Uneti karakter nije samoglasnik.\n");
29         break;
30 }
31
32 return 0;
33 }
```

### Rešenje 2.1.25

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      char op;
7      int x, y;
8
9      /* Ucitava se izraz. */
10     printf("Unesite izraz: ");
11     scanf("%d %c %d", &x, &op, &y);
12
13     /* U zavisnosti od unete operacije, racuna se vrednost izraza. */
14     switch (op) {
15     case '+':
16         printf("Rezultat je: %d\n", x + y);
17         break;
18     case '-':
19         printf("Rezultat je: %d\n", x - y);
20         break;
21     case '*':
22         printf("Rezultat je: %d\n", x * y);
23         break;
24     case '/':
25         if (y == 0)
26             printf("Greska: deljenje nulom.\n");
27         else
28             printf("Rezultat je: %d\n", x / y);
29         break;
30     case '%':
31         printf("Rezultat je: %d\n", x % y);
```

```
        break;
33 default:
    printf("Greska: nepoznat operator.\n");
35 }

37 return 0;
}
```

### Rešenje 2.1.26

```
1  #include <stdio.h>

3  int main()
{
5     /* Deklaracija potrebnih promenljivih. */
    int dan, mesec;

7     /* Ucitava se vrednost datuma koji je zadat u formatu:
9     dan.mesec. */
    printf("Unesite dan i mesec");
11    scanf("%d.%d.", &dan, &mesec);

13    /* Odredjuje se godisnje doba. */
    switch (mesec) {
15        case 1:
16        case 2:
17            /* Ako je mesec januar ili februar, onda je sigurno u pitanju
                zima. */
19            printf("zima\n");
                break;
21        case 3:
17            /* Ako je mesec mart, onda se godisnje doba odredjuje u
                zavisnosti od dana u mesecu. */
23            if (dan < 21)
25                printf("zima\n");
            else
27                printf("prolece\n");
                break;
29        case 4:
30        case 5:
31            /* Ako je mesec april ili maj, onda je sigurno u pitanju
                prolece. */
33            printf("prolece\n");
                break;
35        case 6:
17            /* Ako je mesec jun, onda se godisnje doba odredjuje u
                zavisnosti od dana u mesecu. */
37            if (dan < 21)
39                printf("prolece\n");
            else
41                printf("leto\n");
    }
```



```

43     break;
44 case 7:
45 case 8:
46     /* Ako je mesec jul ili avgust, onda je sigurno u pitanju
47        leto. */
48     printf("leto\n");
49     break;
50 case 9:
51     /* Ako je mesec septembar, onda se godisnje doba odredjuje u
52        zavisnosti od dana u mesecu. */
53     if (dan < 23)
54         printf("leto\n");
55     else
56         printf("jesen\n");
57     break;
58 case 10:
59 case 11:
60     /* Ako je mesec oktobar ili novembar, onda je sigurno u pitanju
61        jesen. */
62     printf("jesen\n");
63     break;
64 case 12:
65     /* Ako je mesec decembar, onda se godisnje doba odredjuje u
66        zavisnosti od dana u mesecu. */
67     if (dan < 22)
68         printf("jesen\n");
69     else
70         printf("zima\n");
71 }
72 return 0;
73 }

```

### Rešenje 2.1.27

```

2  #include <stdio.h>
3
4  int main()
5  {
6      /* Deklaracija potrebnih promenljivih. */
7      int godina;
8      int mesec;
9      int prestupna;
10
11     /* Ucitava se vrednost godine. */
12     printf("Unesite godinu: ");
13     scanf("%d", &godina);
14
15     /* Vrsi se provera ispravnosti ulaznih podataka. */
16     if (godina < 0) {
17         printf("Greska: neispravan unos godine.\n");
18     }
19 }

```

```
    return -1;
18 }

20 /* Vrsi se proverava da li je godina prestupna, zbog februara */
21 if ((godina % 4 == 0 && godina % 100 != 0) || godina % 400 == 0)
22     prestupna = 1;
23 else
24     prestupna = 0;

26 /* Ucitava se redni broj meseca. */
27 printf("Unesite redni broj meseca: ");
28 scanf("%d", &mesec);

30 /* U zavisnosti od vrednosti meseca, ispisuje se odgovarajuci
31    rezultat. */
32 switch (mesec) {
33     case 1:
34         printf("Januar, 31 dan\n");
35         break;
36     case 2:
37         if (prestupna)
38             printf("Februar, 29 dana\n");
39         else
40             printf("Februar, 28 dana\n");
41         break;
42     case 3:
43         printf("Mart, 31 dan\n");
44         break;
45     case 4:
46         printf("April, 30 dana\n");
47         break;
48     case 5:
49         printf("Maj, 31 dan\n");
50         break;
51     case 6:
52         printf("Jun, 30 dana\n");
53         break;
54     case 7:
55         printf("Jul, 31 dan\n");
56         break;
57     case 8:
58         printf("Avgust, 31 dan\n");
59         break;
60     case 9:
61         printf("Septembar, 30 dana\n");
62         break;
63     case 10:
64         printf("Oktobar, 31 dan\n");
65         break;
66     case 11:
67         printf("Novembar, 30 dana\n");
68         break;
```

```
70     case 12:
71         printf("Decembar, 31 dan\n");
72         break;
73     default:
74         printf("Greska: neispravan unos meseca.\n");
75         return -1;
76     }
77
78     return 0;
79 }
```

### Rešenje 2.1.28

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int dan, mesec, godina, dozvoljeni_broj_dana;
7
8      /* Ucitava se datum. */
9      printf("Unesite datum: ");
10     scanf("%d.%d.%d", &dan, &mesec, &godina);
11
12     /* Vrsi se provera korektnosti vrednosti unete godine. */
13     if (godina < 0) {
14         printf("Datum nije korektan.\n");
15         return 0;
16     }
17
18     /* Vrsi se provera korektnosti vrednosti unetog meseca. */
19     if (mesec < 1 || mesec > 12) {
20         printf("Datum nije korektan.\n");
21         return 0;
22     }
23
24     /* Vrsi se provera korektnosti vrednosti unetog dana. */
25     switch (mesec) {
26     case 1:
27     case 3:
28     case 5:
29     case 7:
30     case 8:
31     case 10:
32     case 12:
33         /* Dozvoljeni broj dana za januar, mart, maj, jul, avgust,
34            oktobar i decembar je 31 */
35         dozvoljeni_broj_dana = 31;
36         break;
37     case 2:
38         /* Dozvoljeni broj dana za februar je 28 ili 29 u zavisnosti od
```

```
39     toga da li je godina prestupna ili ne. */
40     if ((godina % 4 == 0 && godina % 100 != 0) || godina % 400 == 0)
41         dozvoljeni_broj_dana = 29;
42     else
43         dozvoljeni_broj_dana = 28;
44     break;
45 case 4:
46 case 6:
47 case 9:
48 case 11:
49     /* Dozvoljeni broj dana za april, jun, septembar i novembar je
50        30. */
51     dozvoljeni_broj_dana = 30;
52     break;
53 }
54
55 if (dan < 0 || dan > dozvoljeni_broj_dana) {
56     printf("Datum nije korektan.\n");
57     return 0;
58 }
59
60 /* Kako su sve provere korektnosti prosle, datum se smatra
61    korektnim. */
62 printf("Datum je korektan.\n");
63
64 return 0;
65 }
```

### Rešenje 2.1.29

```
1  #include <stdio.h>
2
3  int main()
4  {
5      /* Deklaracija potrebnih promenljivih. */
6      int dan, mesec, godina;
7      int prethodni_dan, prethodni_mesec, prethodni_godina;
8
9      /* Ucitava se vrednost datuma. */
10     printf("Unesite datum: ");
11     scanf("%d.%d.%d.", &dan, &mesec, &godina);
12
13     /* Racunaju se dan, mesec i godina prethodnog dana. */
14     prethodni_dan = dan - 1;
15     prethodni_mesec = mesec;
16     prethodni_godina = godina;
17
18     /* Ako je potrebno, vrse se korekcije. */
19     if (prethodni_dan == 0) {
20         prethodni_mesec = mesec - 1;
21         if (prethodni_mesec == 0) {
```

```
23     prethodni_mesec = 12;
24     prethodni_godina = godina - 1;
25 }
26
27 switch (prethodni_mesec) {
28     case 1:
29     case 3:
30     case 5:
31     case 7:
32     case 8:
33     case 10:
34     case 12:
35         prethodni_dan = 31;
36         break;
37     case 2:
38         if ((prethodni_godina % 4 == 0 && prethodni_godina % 100 != 0)
39             || prethodni_godina % 400 == 0)
40             prethodni_dan = 29;
41         else
42             prethodni_dan = 28;
43         break;
44     case 4:
45     case 6:
46     case 9:
47     case 11:
48         prethodni_dan = 30;
49 }
50
51 /* Ispis rezultata. */
52 printf("Prethodni datum: %d.%d.%d.\n",
53        prethodni_dan, prethodni_mesec, prethodni_godina);
54
55 return 0;
56 }
```

### Rešenje 2.1.30

Rešenje je analogno rešenju zadatka 2.1.29.

### Rešenje 2.1.31

Rešenje je analogno rešenju zadatka 2.1.25.

### Rešenje 2.1.32

```
1 #include<stdio.h>
2 #include<math.h>
3
4 int main()
```

```
5 {
6     /* Deklaracija potrebnih promenljivih. */
7     char c;
8     float x1, y1, x2, y2, x3, y3, x4, y4;
9     float dab, dad;
10    float delta, deltap, deltaq;
11    float 0;
12    float k, n;
13
14    printf("Unesi jedan karakter:");
15    scanf("%c", &c);
16
17    printf("Unesi realne koordinate 4 tacke:");
18    scanf("%f%f%f%f%f%f%f", &x1, &y1, &x2, &y2, &x3, &y3, &x4, &y4);
19
20    switch (c) {
21    case 'k':
22        if (y1 == y2 && y3 == y4 && x1 == x4 && x2 == x3) {
23            dab = sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2));
24            dad = sqrt(pow(x1 - x4, 2) + pow(y1 - y4, 2));
25            0 = 2 * dab + 2 * dad;
26            printf("Obim pravougaonika je %f\n", 0);
27        } else
28            printf("Tacke ne cine pravougaonik sa stranicama "
29                "koje su paralelne koordinatnim osama\n");
30        break;
31    case 'h':
32        /*
33         * Ukoliko se tacke A(x1,y1) i B(x2,y2) ne nalaze na pravoj
34         * koja je paralelna x osi, izracunamo k,n za pravu odredjenu
35         * tackama A(x1,y1) i B(x2,y2) */
36        if ((x1 - x2) != 0) {
37            k = (y1 - y2) / (x1 - x2);
38            n = y1 - k * x1;
39            /*
40             * Proverimo da li tacke C(x3,y3) i D(x4,y4) nalaze na toj
41             * pravoj */
42            if (y3 == x3 * k + n && y4 == x4 * k + n)
43                printf("Tacke su kolinearne, pripadaju pravoj "
44                    "y=%f*x+%f\n", k, n);
45            else
46                printf("Tacke nisu kolinearne\n");
47        }
48        /*
49         * Ukoliko se A i B nalaze na pravoj koja je paralelna x osi,
50         * proverimo da li tacke C(x3,y3) i D(x4,y4) nalaze na toj
51         * pravoj */
52        else if (x3 == x1 && x4 == x1)
53            printf("Tacke su kolinearne, pripadaju pravoj " "x=%f\n", x1);
54        else
55            printf("Tacke nisu kolinearne\n");
56        break;
```

```

57  case 'j':
    delta = x1 * y2 - x2 * y1;
59  deltap = x2 * (y4 - y3) - y2 * (x4 - x3);
    deltaq = x1 * (y4 - y3) - y1 * (x4 - x3);
61  if (delta != 0)
    printf("Sistem ima jedinstveno resenje %.2f, %.2f\n",
63         deltap / delta, deltaq / delta);
    else if (deltap == 0 && deltaq == 0)
65         printf("Sistem je neodredjen ili nema resenja.\n");
    else
67         printf("Sistem nema resenja\n");
        break;
69  default:
    printf("Nekorektan unos\n");
71  }

73  return 0;
}

```

### Rešenje 2.1.33

```

#include<stdio.h>
2  #include<stdlib.h>

4  int main()
{
6  /* Deklaracija potrebnih promenljivih. */
    int k, l, m, n;

8

    /* Ucitavaju se vrednosti pozicija na tabli. */
10   printf("Unesite (k,l): ");
    scanf("%d%d", &k, &l);

12

    printf("Unesite (m,n): ");
14   scanf("%d%d", &m, &n);

16

    /* Vrsi se provera korektnosti ulaznih podataka. */
    if (k < 1 || k > 8 || l < 1 || l > 8 ||
18         m < 1 || m > 8 || n < 1 || n > 8) {
        printf("Greska: neispravna pozicija.\n");
20         return -1;
    }

22

    /* Proverava se da li su (k,l) i (m,n) iste boje. Polja su iste
24     boje ako su: 1) oba reda parna i obe kolone parne ILI 2) oba
        reda neparna i obe kolone neparne. */
26   if (((k % 2 == m % 2) && (l % 2 == n % 2))
        || ((k % 2 != m % 2) && (l % 2 != n % 2)))
28       printf("Polja su iste boje.\n");
    else
30       printf("Polja su razlicite boje.\n");
}

```

```
32  /* Proverava se da li kraljica sa (k,l) napada polje (m,n).
34  Kraljica napada polje u sledecim situacijama:
36  1) Ako se nalaze u istom redu (k==m)
38  2) Ako se nalaze u istoj koloni (l==n)
38  3) Ako se nalaze na istoj dijagonali. Dijagonala moze biti:
38  a) paralelna glavnoj dijagonali (abs(k-l) == abs(m-n))
38  b) paralelna sporednoj dijagonali (k+l == m+n) */
40  if ((k == m) || (l == n) || (abs(k - l) == abs(m - n))
40  || (k + l == m + n)){
42  printf("Kraljica sa (%d, %d) ugrozava polje (%d, %d).\n",
42  k, l, m, n);
44  }
46  else {
46  printf("Kraljica sa (%d, %d) ne ugrozava polje (%d, %d).\n",
46  k, l, m, n);
48  }

50  /* Proverava se da li konj sa (k, l) napada polje (m, n). Postoji
52  8 mogucih vrednosti za polja koja konj napada. Vrsi se
52  provera da li je (m,n) jednako nekom od tih polja. */
54  int uslov = 0;
54  if (((m == k + 2) || (m == k - 2)) &&
54  ((n == l - 1) || (n == l + 1))){
56  uslov = 1;
56  }
58  else if (((n == l - 2) || (n == l + 2)) &&
58  ((m == k + 1) || (m == k - 1))){
60  uslov = 1;
60  }
62
62  if (uslov)
64  printf("Konj sa (%d, %d) ugrozava polje (%d, %d).\n",
64  k, l, m, n);
66  else
66  printf("Konj sa (%d, %d) ne ugrozava polje (%d, %d).\n",
68  k, l, m, n);
70
70  return 0;
}
```